

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Rozplňování radiofarmak robotem

Lukáš Fanta

Vedoucí: Ing. Vladimír Smutný, Ph.D.

Obor: Robotika

Studijní program: Kybernetika a robotika

Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Fanta** Jméno: **Lukáš** Osobní číslo: **457018**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Rozplňování radiofarmak robotem

Název bakalářské práce anglicky:

Individual Dosing of Radioactive Drugs by Robotic Arm

Pokyny pro vypracování:

1. Seznamte se s problematikou výroby radiofarmak.
2. Prostudujte si robota ABB IRB 1200 a systém ROS.
3. Navrhněte pracoviště robota, zrealizujte maketu pracoviště.
4. Navrhněte a implementujte programové vybavení pro řízení pracoviště.
5. Proveďte experimenty a vyhodnoťte je.

Seznam doporučené literatury:

- [1] K. Kristensen: Preparation and Control of Radiopharmaceuticals in Hospitals, IAEA, Vienna, 1979.
- [2] RADIOPHARMACEUTICALS, WHO Document QAS/08.262/FINAL, 2008.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

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

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

Datum zadání bakalářské práce: **20.12.2017**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **30.09.2019**

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

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

prof. Ing. Pavel Ripka, CSc.
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

Poděkování

Chtěl bych poděkovat vedoucímu práce panu Ing. Vladimíru Smutnému, Ph.D. za jeho cenné rady, trpělivost a vstřícnost při konzultacích a vypracování bakalářské práce. Děkuji také panu Ing. Danielu Seifertovi, Ph.D. za jeho cenné rady z oblasti chemie.

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 dne 24. května 2019

Abstrakt

Radiofarmaka jsou důležitá pro diagnostiku a léčbu některých nemocí. Ionizující záření emitované těmito látkami je škodlivé lidskému organismu. Pro zajištění bezpečnosti osob a optimalizaci času je snahou samotný proces rozplňování alespoň částečně automatizovat, a to převážně u rozplňování velkého množství vzorků. V rámci této bakalářské práce byl proveden návrh a realizace makety automatizované rozplňovací stanice. Součástí této stanice je robot, který provádí manipulaci s lahvičkami. Pro ovládání pohybu tohoto robota byl použit systém ROS. Po návrhu a implementaci rozplňovacího procesu v programovacím jazyce C++ byly provedeny a zhodnoceny experimenty.

Klíčová slova: radiofarmaka, rozplňování radiofarmak, ROS

Vedoucí: Ing. Vladimír Smutný, Ph.D.
Český institut informatiky, robotiky a kybernetiky,
Jugoslávských partyzánů 1580/3,
160 00 Praha 6

Abstract

Radioactive drugs are important for the diagnosis and treatment of some diseases. The ionizing radiation of the emitted by these substances is harmful to the human body. In order to ensure the safety of people and to optimize time, it is an effort to at least partially automate the filling process, in particular by filling a large number of samples. Within this bachelor thesis was made design and realization of the model of the automated dosing station. A part of this station is a robot that moves vials between stations. The ROS system was used to control the movement of this robot. Experiments were performed after the design and implementation of the dosing process in programming language C++.

Keywords: radioactive drugs, dosing of drugs, ROS

Title translation: Individual Dosing of Radioactive Drugs by Robotic Arm

Obsah

1 Úvod	1	3 Vlastní zpracování	23
2 Teoretický základ	3	3.1 Příprava pracoviště	23
2.1 Definice pojmů	3	3.1.1 Kalibrace robota	23
2.2 Popis stavu techniky	6	3.1.2 Chapadlo	24
2.2.1 Míra čistoty prostoru	6	3.2 Propojení počítače s řídicí jednotkou IRC5	27
2.2.2 Polohorká komora	6	3.2.1 Komunikace mezi počítačem a řídicí jednotkou IRC5	28
2.2.3 Rozplňovací proces	6	3.2.2 Formát posílaných dat	29
2.2.4 Existující řešení výroby a rozplňování radiofarmak	7	3.2.3 TCP klient	30
2.2.5 Automatizovaná rozplňovací stanice	7	3.2.4 TCP server	30
2.2.6 Maketa rozplňovací stanice	7	3.3 Rozmístění stanic vůči robotu	32
2.2.7 Popis použitých stanic	8	3.4 Použité vizualizační modely a kolizní modely v ROSu	33
2.3 Popis robota	12	3.4.1 Vizualizační model robota ABB	33
2.3.1 ABB IRB 1200	12	3.4.2 Vizualizační model chapadla	33
2.3.2 Řídicí jednotka IRC5	13	3.4.3 Kolizní modely	34
2.3.3 FlexPendant	14	3.5 Výpočet souřadných soustav	34
2.3.4 Programovací jazyk RAPID	14	3.6 Popis rozplňovacího procesu	38
2.4 Možnosti ovládání robota	16	3.6.1 Přípravný proces prováděný obsluhou	39
2.4.1 RobotStudio	16	3.6.2 Přípravný proces prováděný robotem	39
2.4.2 Robot Operating System (ROS)	16	3.6.3 Rozplňovací proces	40
2.5 Popis vytvořeného grafického uživatelského rozhraní	20	3.7 Softwarová implementace	41

3.7.1 Vytvořené balíčky	41
3.7.2 Použité a vytvořené uzly	43
3.7.3 Vytvořené topics	45
3.7.4 Implementace GUI	48
3.7.5 Implementace rozplňovacího procesu v systému ROS	48
3.8 Uživatelská část	49
3.8.1 Ovládání robota během rozplňovacího procesu	49
3.8.2 Nastavení a ovládání GUI ...	50
3.8.3 Spuštění GUI pomocí systému ROS	51
4 Experimenty	53
4.1 Stanovení maximální rychlosti pohybu robota	53
4.2 Změření času rozplňovacího procesu	54
4.2.1 Zhodnocení naměřených časů	54
5 Závěr	57
A Literatura	59
B Technické výkresy	63
C Seznam použitých programů	65
D Obsah přiloženého CD	67

Obrázky

2.1 Foto makety pracoviště	8	3.6 Formát hlavičky zpráv	30
2.2 Stojan na prázdné/plné lahvičky .	9	3.7 Obrázek vizualizačního modelu chapadla	33
2.3 Vážicí stanice	9	3.8 Obrázek kolizního modelu chapadla	34
2.4 Stojan na dávkované lahvičky ..	10	3.9 Obrázek kolizních modelů robota, chapadla a okolního prostoru robota	35
2.5 Dávkovač roztoku	11	3.10 Obrázek kinematického modelu robota s vyznačenými polohami souřadnicových systémů chapadla a základny robota	36
2.6 Stanice pro měření aktivity	11	3.11 Graf použitých uzlů včetně jejich propojení	44
2.7 Stanice uzavírající lahvičky	11	B.1 Příruba první díl	63
2.8 Řídící jednotka IRC5 s vyznačeným modulem DSQC652 digitálních vstupů a výstupů, zdrojem stejnosměrného napětí (Ucc) a síťovým WAN portem	13	B.2 Příruba druhý díl	64
2.9 Modul digitálních vstupů a výstupů s označením <i>DSQC652</i>	14		
2.10 FlexPendant	15		
2.11 Prostředí Rviz s modelem robota ABB a jeho kolizním modelem	18		
2.12 Obrázek GUI pro ovládání robota ABB	21		
3.1 Foto chapadlo Schunk	24		
3.2 Časové intervaly mezi signály pro otevření a zavření prstů chapadla [2]	25		
3.3 Model prstu chapadla vytvořený panem Ing. Danielem Seifertem, Ph.D.	25		
3.4 Schéma propojení modulu DSQC652 s chapadlem	27		
3.5 Propojení IRC5 a PC	28		

Tabulky

2.1 Tabulka popisující parametry a vlastnosti robota ABB [1]	12
2.2 Tabulka popisující pohyby v jednotlivých kloubech robota ABB [1]	12
2.3 Seznam instrukcí [3]	15
2.4 Popis ovládacích prvků v GUI	20
3.1 Popis portů vytvořených soketů spolu s popisem dat určených pro tyto porty.	29
3.2 Struktura dat pro ovládání digitálních výstupu IRC5	30
3.3 Tabulka souřadnic a rotací souřadnicových soustav jednotlivých stanic v souřadnicové soustavě základny robota	37
3.4 Tabulka souřadnic a rotací souřadnicových soustav poloh pro uchopení lahviček v souřadnicových soustavách jednotlivých stanic	38
3.5 Převod dat z topic /set_DO_ABB (kapitola 3.7.3) do podoby dat odeslaných pomocí TCP klienta (kapitola 3.2)	45
3.6 Data posílaná přes topic /set_DO_ABB	46
3.7 Data posílaná přes topic /abb_rqt_cpp/msg_to_gui	46
3.8 Data posílaná přes topic /abb_rqt_cpp/state_from_gui	49
4.1 Popis časů jednotlivých kroků, které odpovídají krokům rozplňovacího procesu	55

Kapitola 1

Úvod

Roboti jsou čím dál více používáni ve výrobních i nevýrobních odvětvích. Toto stále se zvyšující použití robotů je výhodné pro jejich schopnosti nepřetržitého provozu, možnosti manipulovat s těžkými předměty a možností vykonávat činnosti, které vyžadují vysokou přesnost. Náklady na provoz robotů mají také velmi důležitou roli ovlivňující jejich uplatnění. V současné době jsou tyto náklady často menší než náklady na pracujícího člověka. Z těchto důvodů je lidská práce nahrazována roboty a to převážně ve výrobních odvětvích. Příkladem nahrazení člověka robotem je ve svařovnách a lakovnách. Roboti zde vykonávají okolo 90 až 100 procent výrobní činnosti.

Roboti také nahrazují člověka při vykonávání nebezpečných úkonů. Jedná se například o manipulaci s předměty nebo látkami, které člověka mohou ohrožovat na jeho zdraví nebo dokonce na životě. Za nebezpečné látky můžeme považovat například látky radioaktivní. Zdravotnictví je jedno z odvětví, kde se můžeme setkat s těmito látkami.

Použití těchto radioaktivních látek (radiofarmak) ve zdravotnictví je pro diagnostické a terapeutické účely. Radiofarmakum je látka určená pro aplikaci do těla člověka. Radiofarmakum vzniká zředěním koncentrované radioaktivní látky pomocí několika pomocných látek. Kombinace ředícího a dávkovacího procesu se nazývá proces rozplňování. Proces rozplňování radiofarmak je dosud v naprosté většině případů prováděn lidmi na specializovaných pracovištích. Přes různá bezpečnostní opatření jsou tito lidé i tak vystaveni radioaktivnímu záření, což je jednou z motivací pro nahrazení těchto pracovníků robotem. Další motivací je standardizace celého rozplňovacího procesu.

Jelikož z mnoha důvodů nemáme prostředky k vytvoření reálného pracoviště, bude cílem této práce vytvořit maketu pracoviště. Ta bude demonstrovat jednotlivé části procesu rozplňování. Tento cíl práce rozdělíme na následující části:

- Seznámit se s problematikou výroby radiofarmak.
- Prostudovat robota ABB IRB 1200 a systém ROS.
- Navrhnout pracoviště robota a zrealizovat maketu pracoviště.
- Navrhnout a implementovat programové vybavení pro řízení pracoviště.
- Provést experimenty a vyhodnotit je.

Samotný text bakalářské práce bude rozdělen na několik částí. Nejprve budou uvedeny definice pojmů související s touto prací. Následovat bude analýza současného stavu problematiky, z níž bude vycházet návrh řešení. V dalším kroku bude následovat návrh a sestavení makety pracoviště. V poslední části této práce budou provedeny a vyhodnoceny experimenty. Výsledky těchto experimentů budou použity ke zhodnocení celkového řešení a realizace úkolu.

Kapitola 2

Teoretický základ

2.1 Definice pojmů

Nuklid

„Nuklidem nazýváme množinu atomů“, které mají stejný počet protonů a neutronů a energetický stav [4]. Obecné značení nuklidu X je A_ZX , kde A je hmotnostní (nukleonové) číslo a Z je protonové (atomové) číslo [5].

Radiace

Radiace (záření) je vysílání nebo přenos „energie ve formě vlnění nebo částic skrze prostor nebo hmotu“ [6].

Ionizující záření

Ionizující záření je souhrnné označení pro radiaci, která nese dostatek energie k oddělení elektronů od atomů nebo molekul, čímž je ionizuje [7].

Radioaktivita

„Radioaktivita je proces, při kterém se nestabilní jádro určitého nuklidu samovolně přeměňuje na jádro jiného nuklidu za současného vzniku ionizujícího záření“ [4].

Aktivita

Aktivita daného množství látky udává počet přeměn atomových jader za jednu sekundu a značí se jednotkou *becquerel* (Bq) [4].

Radionuklid

Radionuklid (Radioizotop) je nuklid s nestabilním jádrem atomu, který má přebytečnou jadernou energii [4]. Ta je z jádra atomu uvolňována vyzářováním gama záření nebo vyzářením nové částice (částice alfa, částice beta) [8].

Charakteristické vlastnosti každého radionuklidu jsou poločas rozpadu, množství vyzářené energie a druh přeměny. Poločas rozpadu je čas, za který se přemění polovina jader atomů daného prvku. Energie je vyzářena při radioaktivní přeměně ve formě fotonu. Jednotka energie fotonu je *elektronvolt* (eV). Druh přeměny udává jakým způsobem je přebytečná jaderná energie uvolňována z nestabilního jádra atomu [4].

Existují přírodní a uměle vytvořené radionuklidy. Podle procesu vzniku a původu, přírodní radionuklid dělíme na kosmogenní a terestrální. Kosmogenní radionuklidy vznikají při jaderných reakcích interakcí kosmické záření a stabilních prvků. Mezi nejznámější radionuklidy patří ^{14}C . Terestrální radionuklidy nebo také radionuklidy pozemského původu vznikly již v raných stádiích vzniku Země. Vlivem velkého poločasu rozpadu (řádově 10^8 let) některých radionuklidů (například Uran ^{235}U a ^{238}U) a postupných přeměn mateřských radionuklidů na dceřiné radionuklidy je výskyt radionuklidů na Zemi stále značný [9].

Uměle vytvořené radionuklidy mohou být získávány pomocí urychlovače, jaderného reaktoru nebo radionuklidového generátoru například Mo/Tc. V nukleární medicíně se využívají pouze uměle vytvořené radionuklidy [4].

$^{99m}\text{Technecium}$

Jedná se o nejčastěji používaný radionuklid v nukleární medicíně, sloužící k diagnostickým účelům. Písmeno *m* u jeho hmotnostního čísla označuje, že se jedná o excitovaný stav jádra. Atomová jádra tohoto prvku mají metastabilní stav, což znamená, že mají více energie oproti základnímu stavu jádra. Tato přebytečná energie se uvolní z jádra atomu a vznikne jádro ^{99}Tc v základním stavu. Energie jeho fotonů je 140 keV. Ta způsobuje malou radiační zátěž organismu, ale zároveň je dostatečná k detekci záření v těle pacienta. Poločas rozpadu ^{99m}Tc je 6 hodin [10] a [11].

Ředící roztok

Ředící roztok slouží pro naředění koncentrované látky. Jako ředící roztok se nejčastěji používá fyziologický roztok [12].

Radiofarmaka

Radiofarmakum je směs, která obsahuje jeden nebo více radionuklidů. Hlavními složkami radiofarmaka jsou ředící roztok a koncentrovaná radioaktivní látka. Koncentrovaná radioaktivní látka je vyrobený radionuklid, který obsahuje minimum rozpouštědla [4].

Nukleární medicína

Nukleární medicína je lékařský obor zabývající se diagnostikou a terapií chorob pomocí radiofarmak. Diagnostika spočívá v aplikaci radiofarmak do krve, kostí, kůže, dýchacích a trávících cest pacienta, které lze měřit a následně hodnotit stav zejména srdce, plic, štítné žlázy, jater, žlučníku

a kostry. Terapie spočívá v aplikaci zářiče (radiofarmaka) do místa léčené léze v těle pacienta, kde dochází vlivem vysokých dávek radiace k poškozování a odumírání buněčných struktur tkání. Jedná se například o léčbu nádorových onemocnění [8].

Obsluha

Obsluha je skupina osob, která přijde do styku s radiofarmaky. Radiofarmaka je potřeba vyrobit, nadávkovat, doručit k pacientovi a aplikovat mu je do těla. Všechny tyto úkony jsou prováděny obsluhou. V této práci budeme popisovat pouze výrobu a rozplňování radiofarmak, proto za obsluhu budeme považovat operátory obsluhující nebo vykonávající proces rozplňování [4].

Nebezpečí radiofarmak pro člověka

Obecně můžeme říci, že radioaktivní látky samovolně vyzářují ionizující i neionizující záření. Kde ionizující záření je škodlivé pro lidský organismus. Při nadměrném vystavení tomuto záření pak dochází k poškození tkání, takzvané nemoci z ozáření. Závažnost poškození tkání souvisí s množstvím záření, kterému je člověk vystaven [13].

Nebezpečí pro pacienta

Pacienti jsou vystavováni dávkám ionizujícího záření cíleně za účelem diagnostiky nebo léčby nemocí. V závislosti na závažnosti onemocnění jsou pacienti více či méně vystavováni těmito dávkám, ovšem v porovnání s obsluhou nejsou vystaveni tak pravidelným dávkám záření [4].

Nebezpečí pro obsluhu

Obsluha je díky opakované práci s radiofarmaky vystavena mnohem větším dávkám záření než pacienti. Aby obsluha netrpěla nemocí z ozáření uplatňují se různá bezpečnostní opatření a ochranné principy [4].

ALARA (As Low As Reasonable Achievable) je základní princip ochrany před ionizujícím zářením, který stanovuje, že riziko z ozáření musí být tak nízké, jak je to rozumně dosažitelné z hlediska hospodářského a sociálního. Tento princip ochrany je uplatňován na všech pracovištích, kde obsluha přijde jakýmkoliv způsobem do styku s radiofarmaky. Princip ochrany ALARA předepisuje především tloušťku a typ materiálu, ze kterého je vytvořené pracoviště a časy, které určují, jak dlouho se mohou pracovníci pohybovat na tomto pracovišti [14].

Jedním z bezpečnostních opatření jsou ochranné pomůcky, například ochranné rukavice. Takovéto pomůcky nejčastěji slouží pouze k ochraně přímého kontaktu radiofarmak s tělem či pokožkou obsluhy [4].

■ 2.2 Popis stavu techniky

■ 2.2.1 Míra čistoty prostoru

Míra čistoty prostoru se určuje podle velikosti a počtu prachových částic na daný objem prostoru. Podle konkrétního množství částic prachu o určité maximální velikosti tyto prostory dělíme na třídy A až D, kde třída A je označení pro nejčistší prostor. Toto rozdělení je stanoveno podle evropské normy *EU GMP classification* [15]. Pro otevřené plnění a dávkování léčiv je potřeba čistý prostor splňující třídu čistoty A [12].

■ 2.2.2 Polohorká komora

Polohorká komora je uzavřený prostor, který odděluje radionuklidy (zdroje záření) od obsluhy a tak obsluhu chrání proti ozáření. Je vyrobena z materiálu (železo nebo olovo), který při určité tloušťce dostatečně absorbuje ionizující záření. Dalším důležitým úkolem polohorké komory je udržení čistých prostor uvnitř komory [16]. Pro vstup a výstup zpracovaných materiálů do polohorké komory se používá takzvaná přechodová komora. Tato komora především vyrovnává velké rozdíly míry čistoty v polohorké komoře a mimo ni [12].

Pro účely rozplňování radiofarmak musí polohorká komora splňovat požadavky pro třídu čistoty typu A a přechodová komora požadavky pro třídu čistoty typu B [12].

■ 2.2.3 Rozplňovací proces

Rozplňovací (též plnicí) proces je soubor po sobě jdoucích dílčích úkonů plnění nebo ředění roztoků, které vedou ke vzniku daného množství naředěného roztoku požadovaných vlastností [17]. V našem případě se jedná o ředění roztoků vedoucích ke vzniku radiofarmaka. Nejčastěji se jedná o tyto úkony:

- Manipulace s lahvičkami.
- Otevírání a zavírání lahviček.
- Vážení lahviček.
- Ředění a směšování roztoků.
- Dávkování roztoku do lahviček.

- Měření aktivity.

■ 2.2.4 Existující řešení výroby a rozplňování radiofarmak

V české republice radiofarmaka jako taková nelze vyrobit přímo v nemocnicích na oddělení nukleární medicíny, ale jen u schválených výrobců státní autoritou pro kontrolu léčiv (SÚKL) [18]. Výjimku činí skupina radiofarmak, kterou lze připravit z kitu. Jedná se o kit, který obsahuje certifikovanou lahvičku s prekurzorem a radionuklid (nejčastěji $^{99m}\text{Technecium}$). Smícháním požadovaného množství radionuklidu s prekurzorem pak vznikne požadované radiofarmakum [12].

Obecně lze říci, že výroba a následné rozplňování radiofarmak probíhá na registrovaných výrobních místech. Jeden z nejčastějších typů registrovaných výrobních míst se jmenuje PET centrum [12]. Název PET je také označení pro diagnostické vyšetření pomocí radiofarmak v nemocnicích. Toto vyšetření patří v současné době k jednomu z nejvyužívanějších. Převážná většina produkce radiofarmak těchto PET center je určena právě pro PET vyšetření [4] a [19].

Samotný proces rozplňování musí být kontrolován a schválen. Během procesu musí být na pracovišti zajištěna dostatečná ochrana pro obsluhu z hlediska obdržené radiační dávky. Tento proces podléhá monitorování a může probíhat jen na vymezených místech nazývaných kontrolované pásmo [12].

■ 2.2.5 Automatizovaná rozplňovací stanice

Automatizovaná rozplňovací stanice je zařízení uzpůsobené k vykonávání automatického rozplňovacího procesu bez trvalé asistence obsluhy. Pro zachování čistých prostor při rozplňování je celá stanice umístěna v polohorké komoře, která splňuje požadavky třídy čistoty typu A [17].

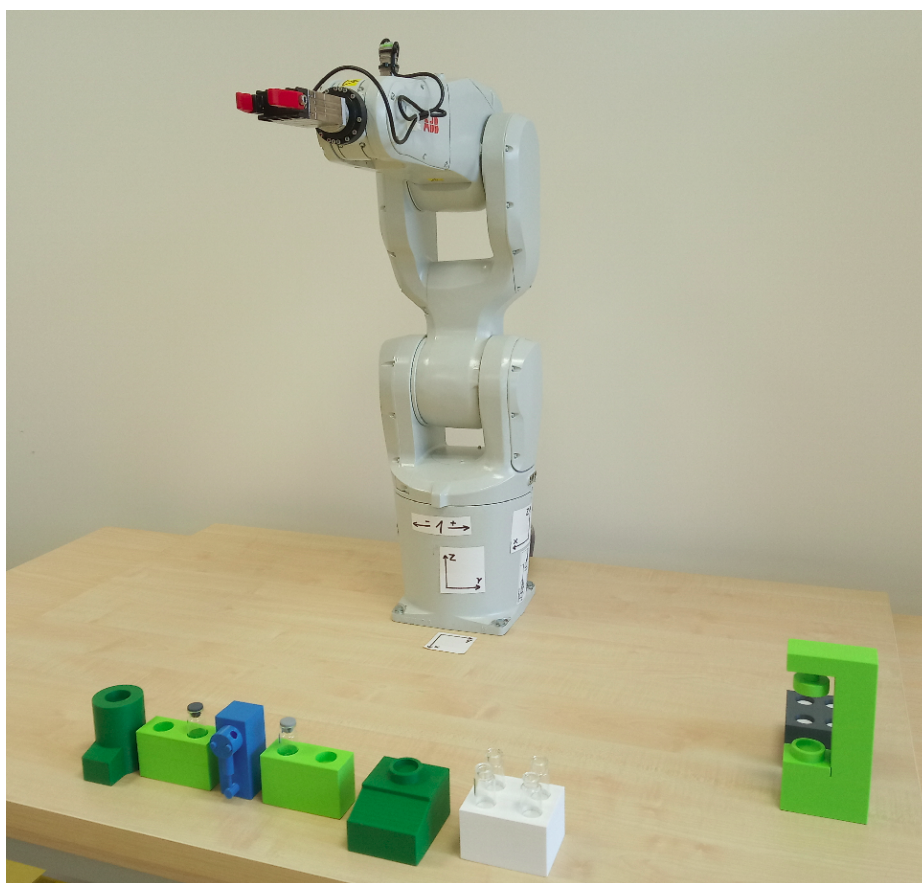
Základními předpoklady této stanice jsou snadná modifikace parametrů rozplňování (změna poměru plnění lahviček a změna aktivity radiofarmaka) a snadná změna jednotlivých kroků rozplňovacího procesu [12].

■ 2.2.6 Maketa rozplňovací stanice

Z mnoha důvodů nebylo možné vytvořit automatizovanou rozplňovací stanici, kterou by bylo možné použít v praxi, proto jsme pro účely demonstrace rozplňovacího procesu vytvořili maketu rozplňovací stanice. Toto pracoviště ovšem nesplňuje základní požadavky na rozplňovací stanici a ani se zde

nepracuje s radiofarmaky nebo radioaktivními látkami. Rozplňovací proces se na této stanici provádí tak, že robot přemísťuje celkem 4 lahvičky mezi vyrobenými 3D modely, které vzdáleně připomínají reálné stanice (popis stanice je uveden v kapitole 2.2.7). Hlavním úkolem této makety rozplňovací stanice je tedy společně s vytvořeným GUI (popis GUI je v kapitole 2.5) ukázat, jak by jednotlivé kroky rozplňovacího procesu mohly vypadat z hlediska pohybu robota a přemísťování lahviček z jedné stanice do druhé. Sestavená maketa pracoviště je vyobrazena na obrázku 2.1.

Robot použit pro manipulaci s lahvičkami nespĺňuje požadavky na míru čistoty při rozplňování a stejně tak i prostor ve kterém je rozplňovací proces prováděn. Rozplňovací proces prováděný touto stanicí je řízen a ovládán pomocí systému ROS (popis systému ROS je v kapitole 2.4.2).



Obrázek 2.1: Foto makety pracoviště

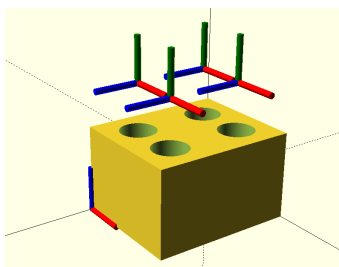
2.2.7 Popis použitých stanic

Jednotlivé stanice slouží k provádění jednotlivých specializovaných úkonů při procesu rozplňování. Při tvorbě makety pracoviště byly použity pouze plastové

3D modely vytisknuté na 3D tiskárně. Jednotlivé soubory s vytvořenými modely jsou ve formátu *STL* k dispozici na přiloženém CD viz příloha D. Jednotlivé modely stanic, které jsou vyobrazeny na obrázcích níže jsou zobrazeny včetně souřadnicových soustav těchto stanic (popis souřadnicových soustav jednotlivých stanic je v kapitole 3.5). Červená barva znázorňuje souřadnicovou osu x , zelená barva znázorňuje souřadnicovou osu y a modrá barva souřadnicovou osu z .

■ Stojan na prázdné lahvičky

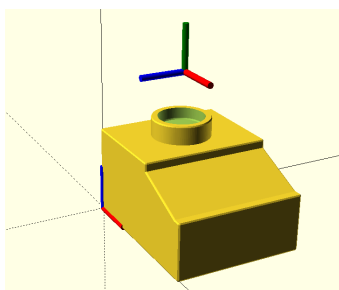
Stojan, který slouží pro umístění prázdných lahviček určených pro rozplňovací proces. Model této stanice je vyobrazen na obrázku 2.2.



Obrázek 2.2: Stojan na prázdné/plné lahvičky

■ Vážicí stanice

Vážicí stanice je určena k vážení prázdných nebo nadávkovaných lahviček. Model této stanice (viz obrázek 2.3) byl vytvořen podle laboratorní váhy s označením *KERN PCB* viz literatura [20].

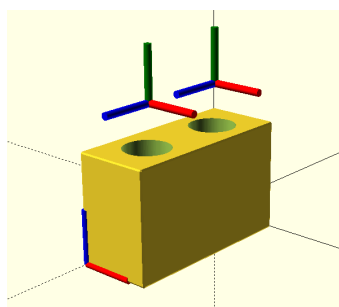


Obrázek 2.3: Vážicí stanice

■ Stanice pro dávkování ředícího roztoku

Jedná se o stanici, která dávkuje stanovené množství ředícího roztoku do lahvičky. Tato stanice se v našem případě skládá ze dvou částí. První část je stojan (obrázek 2.4), ve kterém je na první pozici od začátku rozplňovacího cyklu umístěna lahvička s ředícím roztokem určeným pro rozplnění všech vzorků z dávky. Druhá pozice je určena pro lahvičky do kterých se bude ředící roztok plnit.

Druhá část je dávkovač, který pomocí kapilár dávkuje jednotlivé lahvičky. Model dávkovače (obrázek 2.5) byl vytvořen podle čerpadla nazývaného *Centris Pump* viz literatura [21].



Obrázek 2.4: Stojan na dávkované lahvičky

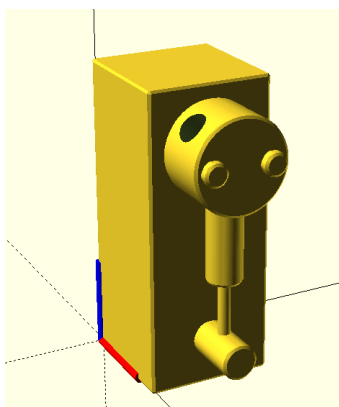
■ Stanice pro dávkování radioaktivní látky

Jedná se o stanici, která dávkuje stanovené množství radioaktivní látky do lahvičky, kde se ředí s ředícím roztokem. Tuto stanici tvoří stojan (obrázek 2.4), ve kterém je na první pozici od začátku rozplňovacího cyklu umístěna lahvička s radioaktivní látkou určenou pro rozplnění všech vzorků z dávky. Druhá pozice je určena pro lahvičky, ve kterých se bude nadávkováná radioaktivní látka ředit s ředícím roztokem. Pro dávkování by mohl být použit stejný typ dávkovače jako u stanice pro dávkování ředícího roztoku.

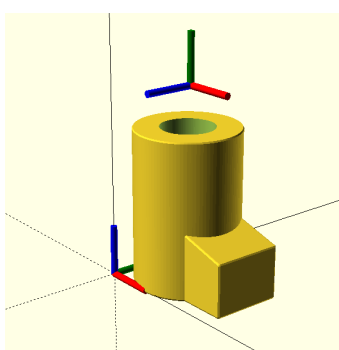
■ Stanice pro měření aktivity

Jedná se o stanici, která provádí měření aktivity roztoku v lahvičce. Pro eliminaci vlivů okolního radioaktivního záření je měřící komora v této stanici vyrobena ze stínících materiálů. Samotné měření pak probíhá v této měřící komoře.

Předlohou pro námi vytvořený model (obrázek 2.6) byla měřící stanice s označením *Měřič aplikované aktivity* viz literatura [22].



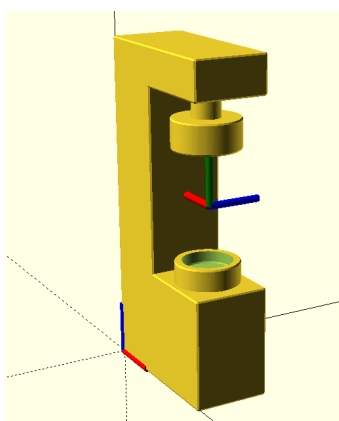
Obrázek 2.5: Dávkoř roztoku



Obrázek 2.6: Stanice pro měření aktivity

■ Stanice uzavírající lahvičky

Stanice uzavírá naplněné lahvičky pomocí silikonového špuntu a hliníkového nebo kovového krytu. Model této stanice je vyobrazen na obrázku 2.7.



Obrázek 2.7: Stanice uzavírající lahvičky

■ Stojan na naplněné lahvičky

Stojan, který slouží pro umístění lahviček s rozplněným radiofarmakem. Model této stanice je vyobrazen na obrázku 2.2.

■ 2.3 Popis robota

■ 2.3.1 ABB IRB 1200

Pro účely vytvoření makety pracoviště byl použit robot s 6 rotačními klouby a 6 stupni volnosti, které robotovi umožňují libovolně manipulovat s předměty v prostoru. Robot je od firmy ABB s označením IRB 1200-7/0.7 (obrázek 2.1). Jedná se o robota, který není uzpůsoben na práci v přítomnosti člověka. Informace o robotu jsou uvedeny v tabulkách 2.1 a 2.2.

Dosah	703 mm
Opakovatelnost pozice	0,02 mm
Užitečné zatížení	7 kg
Hmotnost	54 kg
Řídící jednotka	IRC5

Tabulka 2.1: Tabulka popisující parametry a vlastnosti robota ABB [1]

Pohyb v ose	Pracovní rozsah [°]	Maximální rychlost [°/s]
Kloub 1	+170 až -170	288
Kloub 2	+135 až -100	240
Kloub 3	+70 až -200	300
Kloub 4	+270 až -270	400
Kloub 5	+130 až -130	405
Kloub 6	+360 až -360	600

Tabulka 2.2: Tabulka popisující pohyby v jednotlivých kloubech robota ABB [1]

■ Integrované vodiče v ramenech robota

Skrze jednotlivá ramena robota vede 10 vzájemně izolovaných vodičů s označením CP/CS, což je zkratka pro název *Customer power/Customer signal*. Koncovky těchto vodičů jsou umístěny na prvním (označení R1.CP/CS) a pátém (označení R4.CP/CS podle firmy ABB) rameni robota (schéma propojení koncovek s těmito vodiči je v manuálu [23]). Hlavním účelem těchto vodičů je umožnění elektrického napájení a ovládání externích zařízení připevněných k sedmému rameni robota. V našem případě jsou 4 vodiče využity

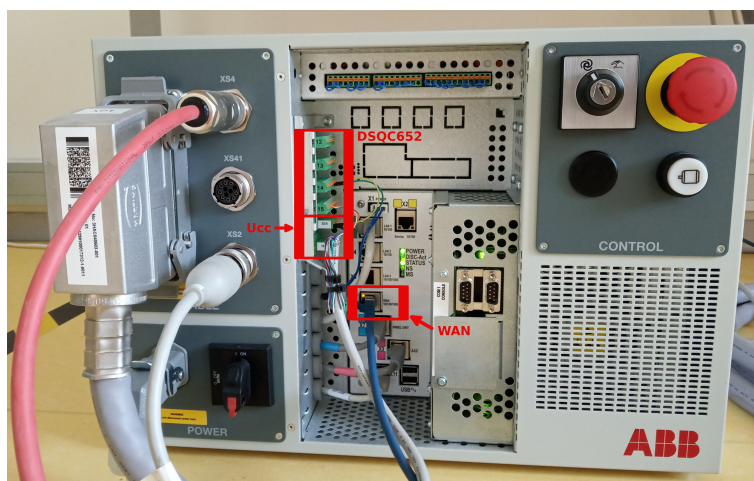
pro napájení elektrického napájení a ovládání chapadla (popis chapadla je v kapitole 3.1.2).

Maximální hodnota elektrického napětí pro jednotlivé vodiče CP/CS je 49 V proti zemi a maximální velikost proudového zatížení jednotlivých vodičů je 500 mA [24].

2.3.2 Řídící jednotka IRC5

Řídící jednotka IRC5 je nedílnou součástí robota, která umožňuje jeho řízení. Řídící jednotka je tvořena ze dvou hlavních modulů, řídicího a pohybového. Řídící modul (anglicky Control module) obsahuje veškerou řídicí elektroniku, například hlavní počítač, vstupně výstupní modul a komunikační modul. Pohybový modul (anglicky Drive module) obsahuje výkonovou elektroniku sloužící k ovládání motorů robota. Operační systém této jednotky se nazývá RobotWare-OS [25].

Obrázek 2.8 vyobrazuje řídicí jednotku IRC5 včetně vyznačených použitých modulů, kterými jsou modul DSQC652 digitálních vstupů a výstupů, zdroj stejnosměrného napětí na 24V proti zemi s označením Ucc a ethernetový WAN port sloužící pro připojení řídicí jednotky do počítačové sítě (popis připojení viz kapitola 3.2.1). Zdroj napětí Ucc obsahuje dva výstupy, které jsou připojeny na konektor označený jako XS16. Podrobnější popis je v manuálu [26].



Obrázek 2.8: Řídící jednotka IRC5 s vyznačeným modulem DSQC652 digitálních vstupů a výstupů, zdrojem stejnosměrného napětí (Ucc) a síťovým WAN portem

■ Modul digitálních vstupů a výstupů

Modul digitálních vstupů a výstupů má u naší řídicí jednotky označení *DSQC652* (obrázek 2.9) a slouží pro ovládání digitálních vstupů a digitálních výstupů. Tento modul umožňuje spínat napětí 24 V proti zemi na šestnácti digitálních výstupech (zkráceně DO) a detekovat napětí 24 V proti zemi na šestnácti digitálních vstupech (zkráceně DI). Digitální vstupy jsou vyvedeny na konektory s označením XS12 a XS13. Digitální výstupy jsou vyvedeny na konektory s označením XS14 a XS15. Schéma vnitřního propojení digitálních vstupů a výstupů je poskytnuto výrobcem v manuálu [26].

Pro možnost spínání napětí na digitálních výstupech je nejprve potřeba přivést napětí 24V proti zemi na piny 9 a 10 konektoru XS14 nebo XS15 [26]. V našem případě jsme toto napětí přivedli na konektor XS15 ze zdroje stejnosměrného napětí U_{cc} a to konkrétně z pinu 1 a 2 konektoru XS16. Schéma tohoto propojení je na obrázku 3.4.



Obrázek 2.9: Modul digitálních vstupů a výstupů s označením *DSQC652*

■ 2.3.3 FlexPendant

FlexPendant (obrázek 2.10) je mobilní ovládací jednotka, která se využívá k provádění řady činností při obsluze systému robota. Jedná se například o spouštění a upravování programů, konfigurace parametrů robota a ruční ovládání robota. FlexPendant je tvořen vlastním HW a SW a jako takový představuje kompletní počítač, který je propojen s řídicí jednotkou IRC5 pomocí integrovaného kabelu s konektorem [25].

■ 2.3.4 Programovací jazyk RAPID

Při implementaci rozplňovacího cyklu pomocí robota bylo potřeba napsat zdrojový kód také v programovacím jazyce RAPID, proto v této kapitole stručně popíšeme funkcionalitu tohoto programovacího jazyka. Detailní popis viz manuál [3].



Obrázek 2.10: FlexPendant

RAPID je vysokoúrovňový programovací jazyk vyvinut firmou ABB pro účely programování robotů. Základem programovacího jazyka RAPID jsou Instrukce (anglicky Instructions), Rutiny (anglicky Routines) a Data. Instrukce programu popisují chování robota pomocí skupiny příkazů, například instrukce pro pohyb robota, pro nastavení výstupů nebo snímání vstupů, atd. Instrukce většinou mají vstupní argumenty, které blíže specifikují provedení dané instrukce [3]. Tabulka 2.3 popisuje nejdůležitější instrukce.

Označení	Popis
MoveL	Pohyb robota po přímce
MoveJ	Pohyb robota
MoveC	Pohyb robota po kružnici
SetDO	Nastavení digitálního výstupu
TPWrite	Výpis zprávy na FlexPendant

Tabulka 2.3: Seznam instrukcí [3]

Rutiny se dělí na 3 typy: procedury (anglicky procedures), funkce (anglicky functions) a trap routines. Z důvodu nepřesného překladu do českého jazyka budeme některé názvy uvádět pouze v anglickém jazyce. Procedury jsou podprogramy, které spouští hlavní program a nemají žádnou návratovou hodnotu. Funkce jsou podprogramy, které po ukončení vrací návratovou hodnotu. Trap routines umožňují zachytit a reagovat na systémová přerušení (anglicky interrupts). Data slouží k ukládání informací. Mohou být ve formě proměnných, konstant nebo takzvaných persistents. Persistents jsou proměnné, které si pamatují poslední uloženou hodnotu při běhu programu a při dalším spuštění programu je použita tato hodnota jako počáteční [3].

Program v tomto programovacím jazyce je tvořen programovými moduly (anglicky program modules) a systémovými moduly (anglicky system modules), které obsahují rutiny a data. Systémový modul je knihovna, jejíž rutiny a data mohou využívat programové moduly. Vstupní bod programu je procedura

`main()`, která musí být implementována právě v jednom programovém modulu [3].

2.4 Možnosti ovládání robota

2.4.1 RobotStudio

RobotStudio je počítačový program určený pro modelování, programování a simulaci robotů od firmy ABB a jejich pracovního prostoru. RobotStudio umožňuje práci ve dvou režimech: online a offline režim. Práce v online režimu je umožněna pouze při fyzickém propojení počítače s řídicí jednotkou IRC5. Uživatel při programování a simulaci nastavuje parametry přímo v řídicí jednotce IRC5. Při práci v offline režimu je uplatněn virtuální řídicí systém, který simuluje funkce řídicí jednotky IRC5. Uživatel upravuje parametry IRC5 pouze lokálně na svém počítači [25].

2.4.2 Robot Operating System (ROS)

Robot Operating System dále jen *ROS* je sbírka programových nástrojů sloužících pro vývoj robotických systémů. Přestože má v názvu *Operating System* nejedná se o operační systém jako takový. Mezi hlavní funkce ROSu patří: poskytování služeb pro různé hardwarové platformy počítačů, nízkourovňové ovládání zařízení, posílání zpráv (anglicky *messages*) mezi procesy a správa balíčků (anglicky *packages*). ROS je určen především pro operační systém Linux a podporuje programovací jazyky C++ a Python [27] a [28].

Software systému ROS je organizován do balíčků. Balíček je základní stavební jednotkou celého systému ROS. Sestavování a správa balíčků probíhá pomocí programu *catkin* [29].

Při běhu systému ROS jsou nejdůležitější prvky takzvané uzly (anglicky *nodes*). Uzel je proces, který provádí výpočetní operace. Jednotlivé uzly spolu vzájemně komunikují pomocí *topics*, *services* nebo *Parameter Server*. Kvůli nejednoznačnému překladu některých slov do českého jazyka, ponecháme tato slova v originálním názvu. *Topic* je název pro virtuální komunikační sběrnici, která realizuje zasílání zpráv (anglicky *messages*) mezi jednotlivými uzly. Tato sběrnice poskytuje jednosměrnou spojovou komunikaci mezi dvěma uzly. Tyto uzly se nazývají *publisher* (uzel, který posílá zprávy) a *subscriber* (uzel, který přijímá zprávy) [30].

Service je taktéž název virtuální sběrnice pro posílání zpráv mezi dvěma uzly. Hlavní rozdíl oproti sběrnici topic je v tom, že komunikace probíhá oboustranně způsobem přijetí zprávy a odeslání potvrzení přijetí zprávy. Tento způsob komunikace se využívá pro vzájemnou synchronizaci prováděných úkonů mezi dvěma uzly. Parametr Server umožňuje ukládat a načítat parametry za běhu programu. Jedná se o server, který je sdílený a přístupný pro uzly přes síťové API [30].

■ Metapackage

Metapackage je balíček, který neobsahuje žádné spustitelné programy nebo zdrojové kódy, ale obsahuje jeden nebo více vzájemně souvisejících balíčků [31]. Slovo metapackage v tomto kontextu lze nejlépe přeložit do českého jazyka jako *projekt* a pod tímto názvem ho budeme v celé práci označovat.

■ ROS-Industrial

ROS-Industrial je program, který obsahuje knihovny, programové vybavení a ovladače určené pro průmyslové roboty [32]. Součástí ROS-Industrial jsou projekty `abb` a `abb_experimental`, které obsahují balíčky týkající se pouze průmyslových robotů od firmy ABB [33]. Zdrojové kódy a jednotlivé balíčky, které byly převzaty z těchto projektů pro účely implementace rozplňovacího cyklu jsou dostupné pomocí webového programu GitHub [34] a [35]. Seznam použitých balíčků z těchto projektů je uveden v kapitole 3.7.1. Veškeré balíčky spadají pod licence BSD nebo Apache 2.0 (popis licencí je uveden v kapitole 2.4.2).

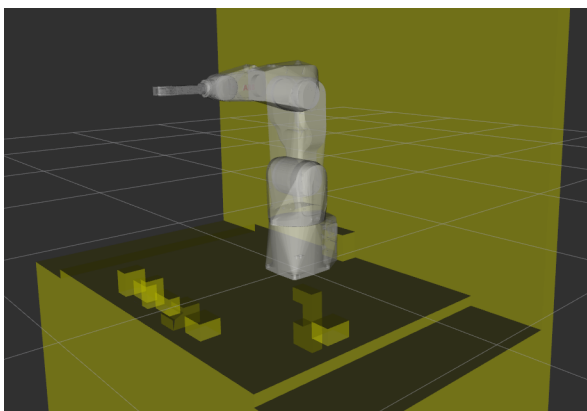
■ MoveIt

MoveIt je soubor programových nástrojů systému ROS určených pro plánování trajektorie, výpočet kinematických úloh a ovládání robotů a jejich orientace v prostoru. MoveIt provádí veškeré výpočty plánování trajektorie nad kinematickým modelem robota a jeho kolizním modelem [28].

Hlavním úkolem plánovacího algoritmu je naplánovat trajektorii, jejíž počátek je v počáteční poloze chapadla a konec je v cílové poloze chapadla. Množina poloh chapadla, které tvoří trajektorii pohybu robota, je určována pomocí prohledávání stavového prostoru, kde každý stav je reprezentován konkrétní polohou chapadla. Při přechodu mezi jednotlivými stavy jsou kontrolovány kolize. Tyto kolize se určují na základě polohy chapadla vůči kolizním modelům [36].

Rviz

Rviz je 3D vizualizační nástroj systému ROS. Slouží především pro vizualizaci modelu robota, modelu pracovního prostředí robota, kolizního modelu robota a vizualizaci stavových proměnných. Rviz lze využívat jako grafickou nadstavbu softwaru MoveIt nebo samostatně [37] a [28]. Obrázek 2.11 zobrazuje prostředí Rviz spolu s modelem robota ABB a jeho kolizním modelem.



Obrázek 2.11: Prostředí Rviz s modelem robota ABB a jeho kolizním modelem

rqt

Součástí naší práce je také vytvořené GUI (popis tohoto GUI je v kapitole 2.5), které bylo vytvořeno pomocí *rqt*. Proto v této kapitole uvedeme základní informace o tomto programu. Podrobný popis programu *rqt* je uveden v manuálu [38].

rqt (někdy také ROS GUI) je sbírka softwarových nástrojů systému ROS určená pro implementaci různých nástrojů GUI ve formě takzvaných pluginů. Pro nejednoznačný překlad tohoto názvu do českého jazyka budeme používat jeho anglický název. Pluginy lze rozdělit na standardní a robotické. Standardní umožňují především vyčítání informací o běžících programech systému ROS pomocí GUI. Jedná se například o plugin *rqt_topic*, který vypisuje informace o všech vytvořených topic mezi danými uzly. Robotické pluginy umožňují interakci s robotem za běhu programu pomocí GUI, například plugin *rqt_rviz* dokáže zobrazit program Rviz a pomocí něj ovládat robota [38]. Hlavní výhodou *rqt* je zobrazení několika pluginů současně, což může zlepšit přehled nad prováděnými programy.

Implementace těchto pluginů probíhá výhradně pomocí programovacího jazyka C++ nebo Python. Pro implementaci je doporučeno použít vývojové prostředí *Qt Creator* [39], které oproti jiným vývojovým prostředím umožňuje

otevřít UI soubor v grafické podobě. UI soubor obsahuje informace o konfiguraci a rozmístění jednotlivých ovládacích prvků (anglicky widgets) GUI uložené ve formátu XML.

■ Moveit Setup Assistant

Moveit Setup Assistant je grafické prostředí sloužící pro vytvoření balíčku systému ROS, který popisuje model robota. V této kapitole je uveden stručný popis tohoto programu. Podrobný popis je uveden v literatuře [28].

Pomocí tohoto programu byl vytvořen jeden z balíčků systému ROS, který je nezbytný pro správné fungování rozplňovacího cyklu. V našem případě se jedná o balíček *abb_irb1200_7_70_moveit_config* (kapitola 3.7.1). Hlavní funkce Moveit Setup Assistant jsou následující:

- Načtení modelu robota včetně modelu chapadla.
- Nastavení interakcí kolizních modelů mezi jednotlivými rameny modelu robota a modelem chapadla. Jedná se o výpočet vzájemných poloh ramen robota a zjištění, které mohou být v kolizi a které naopak nemohou. Tento výpočet vzájemných kolizí je velmi důležitý především pro optimalizaci doby plánování trajektorie pohybu robota.
- Spojení modelu základny robota s kolizním modelem pracoviště.
- Nastavení počáteční polohy robota. Tuto polohu budeme označovat jako *počáteční pozice* (tuto polohu robota vyobrazuje obrázek 2.1).

■ Licence BSD a Apache 2.0

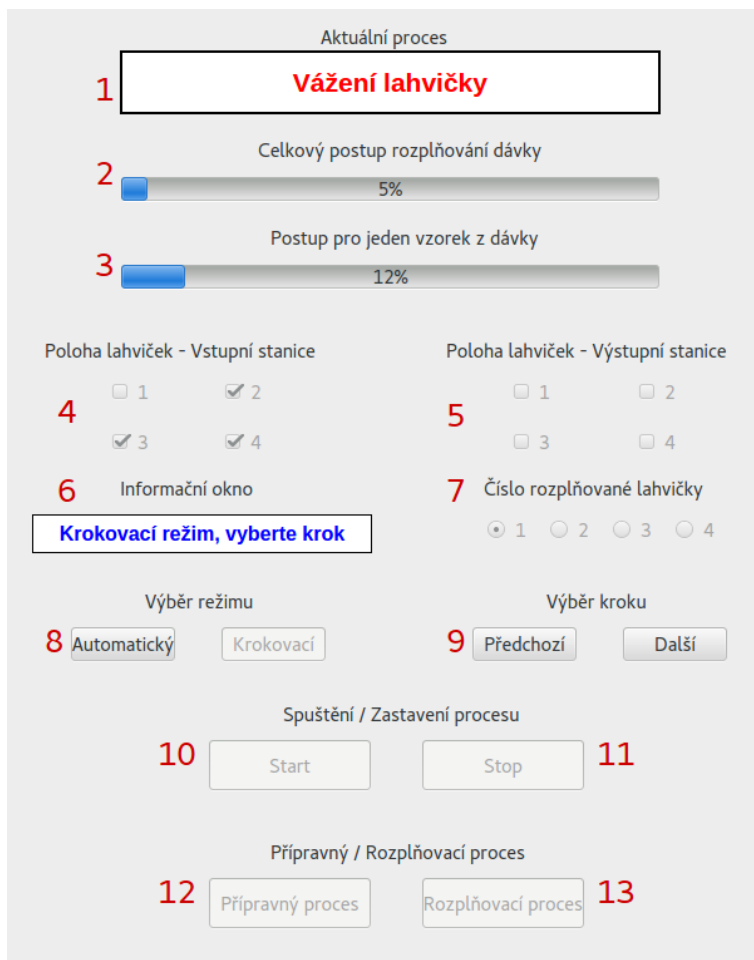
Pod licence BSD nebo Apache 2.0 patří téměř všechny zdrojové kódy a knihovny, které jsme pro účely naší práce použili. Jedná se o licence pro svobodný software, které umožňují volné šíření licencovaného obsahu, pokud jsou uvedeny autor a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo. Tyto informace spolu s autorem jsou uvedeny v každém souboru, který byl použit a spadá pod tyto licence viz literatura [40] a [41].

2.5 Popis vytvořeného grafického uživatelského rozhraní

Hlavním úkolem grafického uživatelského rozhraní (dále jen *GUI*) je ovládání pohybu robota z uživatelsky přívětivého prostředí. Celé GUI je vytvořeno jako plugin programu rqt (popis implementace GUI je uveden v kapitole 3.7.4). Obrázek 2.12 zobrazuje vytvořené GUI, kde funkčnost jednotlivých ovládacích prvků je popsána v tabulce 2.4.

Pozice v GUI	Ovládací prvek/prvky	Popis funkce	Lze ovládat uživatelem ?
1.	Textové pole	Informuje o právě prováděném procesu rozplňování robotem	Ne
2.	Indikátor průběhu	Indikuje (v procentech) postup rozplňování pro celou dávku	Ne
3.	Indikátor průběhu	Indikuje (v procentech) postup rozplňování pro jeden vzorek z dávky	Ne
4.	Zaškrťovací pole	Zobrazuje stav prázdných lahvíček vstupního stojanu na lahvičky	Ne
5.	Zaškrťovací pole	Zobrazuje stav prázdných lahvíček výstupního stojanu na lahvičky	Ne
6.	Textové pole	Informuje o provedené akci v GUI nebo poskytuje návod pro uživatele jak dále pokračovat	Ne
7.	Přepínač	Podle čísla určuje, který vzorek je právě rozplňován	Ne
8.	Tlačítka	Slouží pro výběr režimu rozplňování (Automatický nebo Krokový)	Ano
9.	Tlačítka	Slouží pro výběr kroku v ručním režimu (Předchozí nebo Další)	Ano
10.	Tlačítko	Slouží pro zahájení pohybu robota	Ano
11.	Tlačítko	Slouží pro okamžité zastavení pohybu robota	Ano
12.	Tlačítko	Slouží pro výběr přípravného procesu	Ano
13.	Tlačítko	Slouží pro výběr rozplňovacího procesu	Ano

Tabulka 2.4: Popis ovládacích prvků v GUI



Obrázek 2.12: Obrázek GUI pro ovládání robota ABB

Kapitola 3

Vlastní zpracování

3.1 Příprava pracoviště

3.1.1 Kalibrace robota

Proces kalibrace robota v našem případě znamená nastavení kalibračních parametrů v řídicí jednotce IRC5, které udávají offsety hodnot nastavení motorů ovládající jednotlivé klouby robota. Na základě doporučení od výrobce (formou uživatelského manuálu pro zprovoznění robota [42]) jsme provedli kalibraci ihned po upevnění robota na podstavec.

Po kalibraci robota byla provedena řada úkonů, které souvisejí s uvedením robota do chodu (kapitola 3.1.2). Jeden z úkonů je také vytvoření vizualizačního modelu robota v systému ROS (kapitola 3.4.1). Ihned po propojení vytvořeného modelu robota s reálným robotem jsme zjistili, že vzájemné polohy modelu robota a reálného robota nejsou stejné i přes stejné hodnoty pro natočení kloubů.

Při detailnějším pozorování modelu robota a reálného robota jsme zjistili, že výrobce nám poskytl prostřednictvím nalepovacího štítku špatné informace o modelu robota a tím pádem i špatné kalibrační parametry. Robot měl podle informací na nalepovacím štítku označení IRB 1200/0.9-5, ale náš robot má ve skutečnosti označení IRB 1200/0.7-7 (kapitola 2.3). Hlavní rozdíl je v tom, že náš skutečný robot má o 20 cm kratší dosah [1]. Ihned po odhalení této chyby byla výrobci odeslána zpráva s informací o této chybě a byla provedena ruční metoda kalibrace bez kalibračních nástrojů [42].

Výrobce nám po několika týdnech odeslal informační štítek se správným označením modelu robota a kalibrační parametry, které odpovídaly našemu modelu robota. Po obdržení platného informačního štítku jsme provedli kalibraci znovu podle těchto kalibračních parametrů.

3.1.2 Chapadlo

Popis

Námi použité chapadlo je vyrobeno firmou *Schunk* a označení modelu je *EGP 50*. Prsty chapadla jsou umístěny rovnoběžně proti sobě a posouvají se k sobě a od sebe. Pokud jsou prsty nejdále od sebe budeme tuto polohu nazývat *otevřeno*, pokud jsou prsty jsou nejbližší u sebe budeme tuto polohu nazývat *zavřeno*. Obrázek 3.1 zobrazuje chapadlo ve stavu otevřeno. Vzdálenost vnitřních stěn prstů při stavu zavřeno je 18 mm a při stavu otevřeno 34 mm. Doba procesu zavírání nebo otevírání prstů chapadla je 0,21 sekund [43].

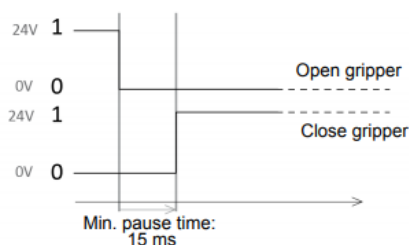


Obrázek 3.1: Foto chapadlo Schunk

Posuv prstů chapadla je realizován pomocí bezkartáčového stejnosměrného servomotoru, který dovoluje sevření prstů až o síle 215 N [43]. Sílu sevření lze měnit přepínačem umístěným na chapadle, který lze přepnout do čtyř poloh pomocí šroubováku. Hodnoty jednotlivých poloh jsou 25 %, 50 %, 75 % a 100 % z celkové síly sevření. Pro uchopení lahvičky pomocí prstů chapadla byly vyzkoušeny všechny hodnoty síly sevření prstů a bylo zjištěno, že hodnota 25 % je plně dostačující pro pevné uchopení lahvičky. Pevné uchopení znamená, že během manipulace s lahvičkou nedochází k samovolnému pohybu lahvičky.

Ovládání posuvu prstů chapadla je pomocí napěťových pulsů o velikosti 24V proti zemi. Výrobce stanovuje minimální časový rozestup mezi pulzy pro ovládání prstů chapadla na 15 ms viz graf 3.2. Pokud nebude dodržen tento časový rozestup dojde k chybovému stavu a pohyb prstů chapadla se okamžitě zastaví. Výrobce také varuje, že pokud tento chybový stav nastane dva a vícekrát, tak může dojít k nenávratnému poškození řídicí elektroniky chapadla. Chybový stav je možné odstranit vypnutím napájení chapadla [2].

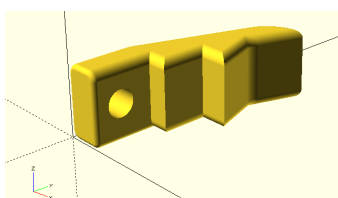
Chapadlo také obsahuje LED diodu (zelenou a červenou). Zelená LED dioda signalizuje připojené napájecí napětí a červená LED dioda signalizuje chybový stav.



Obrázek 3.2: Časové intervaly mezi signály pro otevření a zavření prstů chapadla [2]

■ Mechanické připojení

Chapadlo (obrázek 3.1) je připojeno k 7. rameni robota pomocí příruby. Příruba je složena ze dvou vzájemně spojených částí (technické výkresy obou částí příruby jsou uvedeny v příloze B.1 a B.2). Návrh, zprostředkovávání výroby a montáž obou částí příruby a chapadla k 7. rameni robota byla provedena panem Ing. Danielem Seifertem, Ph.D.. Samotné prsty chapadla nejsou uzpůsobeny pro uchopení lahviček, které jsou válcového tvaru. Pro účely manipulace s lahvičkami byly na konce prstů připevněny pružné nástavce vyrobené na 3D tiskárně z materiálu Flexfill. Model prstu chapadla je na obrázku 3.3. Návrh, výroba a montáž prstů na chapadlo byla provedena panem Ing. Danielem Seifertem, Ph.D..



Obrázek 3.3: Model prstu chapadla vytvořený panem Ing. Danielem Seifertem, Ph.D..

■ Elektrické připojení

Chapadlo jsme připojili pomocí 4-pinového konektoru ke konektoru robota R4.CP/CS (popis konektoru viz kapitola 2.3). Dva piny na konektoru chapadla, které slouží pro připojení napájecího napětí o velikosti 24V proti zemi, byly propojeny pomocí kabelů a skrze robota (vstupní konektor robota je R4.CP/CS a výstupní je R1.CP/CS) s konektorem zdroje stejnosměrného napětí na řídicí jednotce IRC5. Konektor má označení *XS16* a propojení

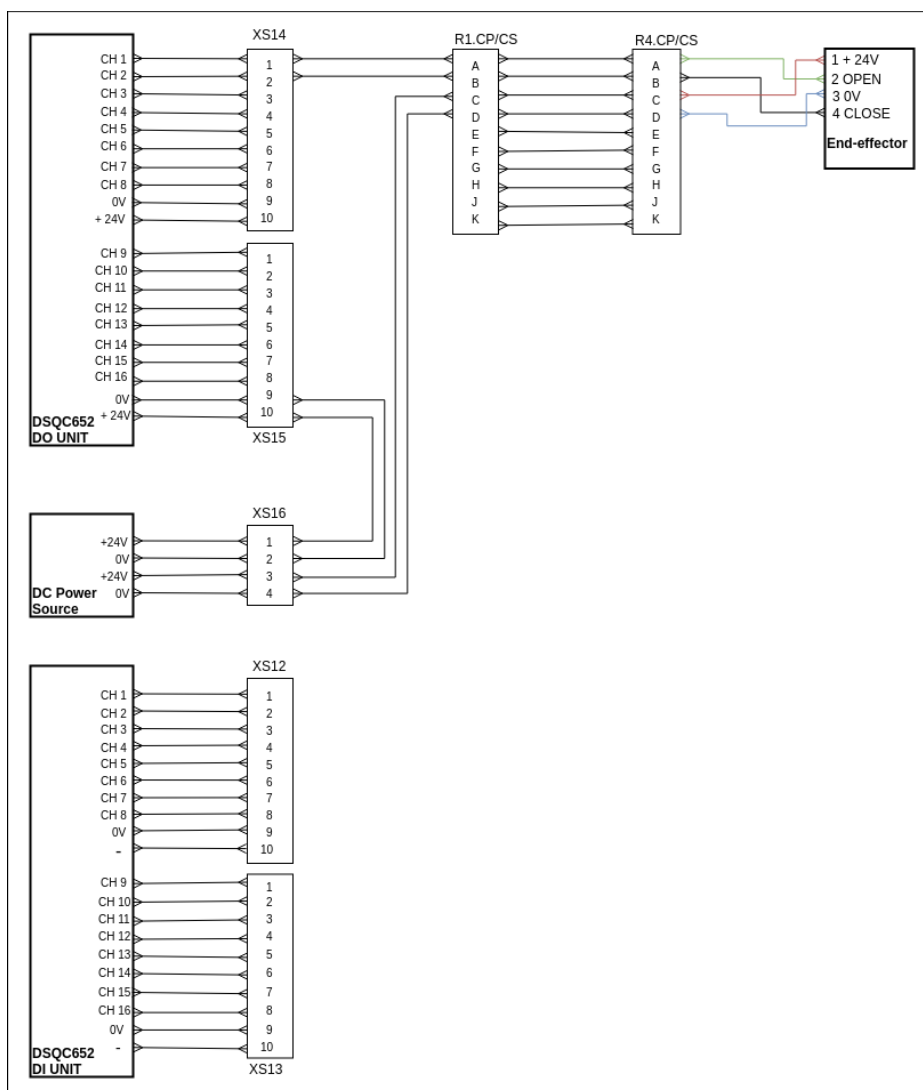
bylo provedeno pomocí pinů 3 a 4. Druhé dva piny, které slouží pro ovládání prstů chapadla, byly propojeny pomocí kabelů a skrze robota s konektorem digitálních výstupů na řídicí jednotce IRC5. Konektor má označení *XS14* a propojení bylo provedeno pomocí pinů 1 a 2. Schéma zapojení 3.4 vyobrazuje toto vzájemné propojení.

Konektor R4.CP/CS pro připojení chapadla je umístěn na pátém rameni robota. Mezi tímto konektorem a chapadlem jsou tedy 2 rotační klouby a vzdálenost je přibližně 30cm při počáteční poloze robota. Kabel, který byl použit pro propojení chapadla s robotem, má délku 1,5 m. Pro zabezpečení poškození kabelu nebo poškození jiných částí pracoviště bylo potřeba tento kabel zkrátit.

Zkrácení bylo provedeno smotáním kabelu do klubka a uchycením na páté rameno robota pomocí nalepovacího úchytu a stahovací pásky (obrázek 2.1 vyobrazuje toto uchycení kabelu na robota). Po zkrácení kabelu bylo potřeba také snížit pracovní rozsah šestého kloubu robota, aby nedošlo k zamotání nebo poškození kabelu robotem. Jeho původní hodnoty jsou -360° až $+360^\circ$. Pro manipulace s lahvičkami stačí otočení chapadla pouze o 90° vzhledem k počáteční poloze šestého kloubu robota. Nové hodnoty pracovního rozsahu šestého kloubu jsou -180° až $+180^\circ$. Změna hodnot pracovního rozsahu 6. kloubu byla provedena pomocí ovládací jednotky FlexPendant (obrázek ovládací jednotky viz 2.10).

■ Softwarové ovládání

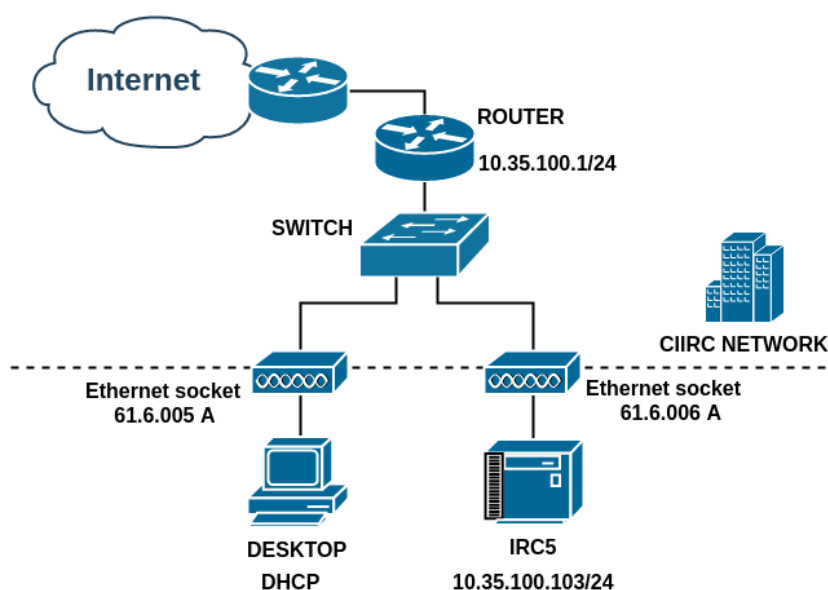
Chapadlo lze ovládat pomocí instrukce SetDO viz tabulka 2.3. Zápis této instrukce v programovacím jazyce RAPID pro otevření chapadla je *SetDO D652_10_OUT0 1*, kde *D652_10_OUT0* je označení modulu, digitálního výstupu a čísla, které po přičtení čísla 1 udává o který DO se jedná. Hodnota 1 na konci instrukce je logická hodnota určující sepnutí nebo rozepnutí zadaného digitálního výstupu. Pro zavření chapadla je použita instrukce *SetDO D65_10_OUT1 1*.



Obrázek 3.4: Schéma propojení modulu DSQC652 s chapadlem

3.2 Propojení počítače s řídicí jednotkou IRC5

Pro komunikaci mezi počítačem a řídicí jednotkou IRC5 byla použita metoda propojení pomocí počítačové sítě 3.5. Připojení IRC5 a počítače do sítě bylo provedeno v budově CIIRC v Praze, kde je počítačová síť spravována školou ČVUT. Podle patra budovy CIIRC a oddělení na tomto patře ve kterém došlo k připojení je síť označována jako ciirc_RMP, kde RMP je zkratka pro název oddělení "Robotika a strojové vnímání".



Obrázek 3.5: Propojení IRC5 a PC

Kromě realizace připojení zařízení do počítačové sítě bylo potřeba zadat parametry sítě do řídicí jednotky IRC5. Tyto parametry jsou následující:

- Statická IPv4: *10.35.100.3*
- IPv4 výchozí brány: *10.35.100.1*
- Maska sítě: *255.255.255.0*

Statická IPv4 adresa řídicí jednotky IRC5 nám byla přidělena správcem sítě CIIRC. Tato adresa je v síti zarezervována pouze pro naši řídicí jednotku a je mimo rozsah IP adres přidělovaných DHCP serverem. Počítač nepotřebuje mít při každém spuštění stejnou IP adresu, proto mu je adresa přidělována pomocí DHCP serveru. Přidělovaná adresa je z rozsahu *10.35.100.101* až *10.35.100.254*.

3.2.1 Komunikace mezi počítačem a řídicí jednotkou IRC5

Komunikace mezi počítačem a řídicí jednotkou IRC5 probíhá po síti pomocí rodiny protokolů TCP/IP. Síťová architektura jejich vzájemného propojení je typu klient-server, kde klient je počítač a server je řídicí jednotka IRC5. Koncovými body síťové komunikace jsou takzvané sokety, které se na začátku komunikace vytvoří a po ukončení vzájemné komunikace mezi počítačem a IRC5 se uzavřou. Hlavními parametry soketu jsou typ komunikace (TCP,

UDP), IP adresa a číslo portu. Typ komunikace je v našem případě TCP protokol, který garantuje odeslání všech dat bez ztráty a ve správném pořadí. Číslo portu rozlišuje jednotlivé služby zařízení. Pro komunikaci počítače s řídicí jednotkou bylo třeba rozlišit celkem tři různé služby, proto byly vytvořeny tři sokety. Čísla portů spolu s jejich popisem jsou uvedeny v tabulce 3.1.

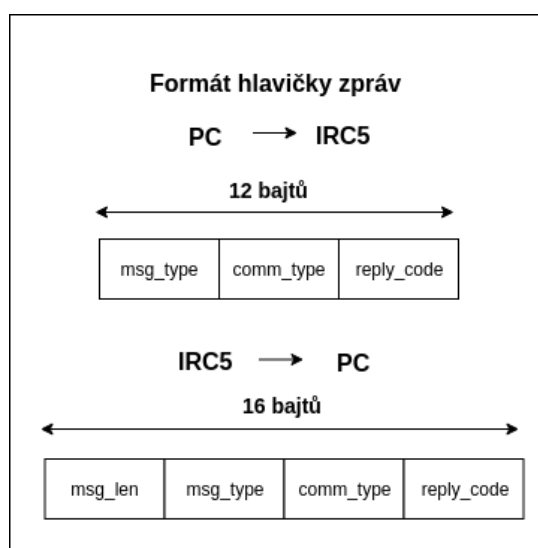
Číslo portu	Popis posílaných dat	Název TCP serveru (řídicí jednotka IRC5)	Název TCP klienta (systém ROS v počítači)
11000	Data s hodnotami požadovaného natočení kloubů robota	ROS_motionServer	/motion_download_interface
11002	Data s hodnotami aktuálního natočení kloubů robota	ROS_stateServer	/robot_state
11003	Data s hodnotami nastavení digitálních výstupů IRC5	ROS_ioUnitServer	/abb_io_unit

Tabulka 3.1: Popis portů vytvořených soketů spolu s popisem dat určených pro tyto porty.

3.2.2 Formát posílaných dat

Nyní se popíšeme formát dat posílaných po síti, které jsou definovány systémem ROS. Tyto data mají dvě hlavní části: hlavičku a samotná data. Pomocí hlavičky probíhá bližší identifikace zprávy na úrovni programu. Formát hlavičky zprávy se liší podle toho, zda je zpráva posílána z PC do IRC5 nebo opačně. Obrázek 3.6 blíže zobrazuje oba typy hlaviček posílaných zpráv, kde *msg_len* udává velikost celé posílané zprávy v bajtech, *msg_type* blíže specifikuje typ zprávy (zpráva pro ovládání polohy robota, zpráva pro ovládání digitálních výstupů, atd.), *comm_type* určuje typ komunikace (topic nebo service) a *reply_code* blíže identifikuje, zda se jedná o potvrzení přijatých dat a v jakém stavu data dorazila (správný nebo špatný formát dat).

Formát dat pro porty 11000 a 11002 včetně kompletní komunikace byly převzaty z projektu ABB programu ROS-industrial (kapitola 2.4.2). Zaměříme se pouze na popis struktury vlastních dat, kterou jsme navrhli a to jsou data pro ovládání digitálních výstupů IRC5 určených pro port 11003. Struktura vlastních dat je popsána v tabulce 3.2, kde *DO_data_1* je jeden bajt (8 bitů), který ovládá digitálních výstupy 1 až 8 a *DO_data_2* je bajt, který ovládá digitálních výstupy 9 až 16. Hodnoty jednotlivých bitů v těchto bajtech odpovídají sepnutí (bit má hodnotu 1) nebo rozepnutí (bit má hodnotu 0) daných digitálních výstupů. Nejméně významný bit (LSB) daného bajtu nastavuje digitální výstup na nejmenší pozici. To znamená, že LSB bajtu *DO_data_1* nastavuje digitální výstup číslo 1.



Obrázek 3.6: Formát hlavičky zpráv

Název	Počáteční pozice v bajtech	Velikost dat v bajtech	Datový typ v IRC5
DO_data_1	1	1	byte
DO_data_2	2	1	byte

Tabulka 3.2: Struktura dat pro ovládání digitálních výstupu IRC5

3.2.3 TCP klient

Hlavním úkolem TCP klienta je připojit se k TCP serveru pomocí jeho IP adresy a daného portu. Pro komunikaci řídicí jednotky IRC5 s počítačem byly použity dva již implementované TCP klienti a jeden bylo potřeba implementovat. Implementace TCP klientů je pomocí systému ROS v programovacím jazyce C++. Každý TCP klient je spouštěn a obsluhován pomocí samostatného uzlu systému ROS. Název TCP klienta je shodný s názvem uzlu, kterým je spouštěn. Jednotlivé názvy TCP klientů včetně čísel portů, pomocí kterých komunikují s daným TCP serverem, jsou uvedeny v tabulce 3.1. Podrobnější popis jednotlivých uzlů vytvořených pomocí systému ROS viz kapitola 3.7.2.

3.2.4 TCP server

Pro komunikaci řídicí jednotky IRC5 s počítačem bylo potřeba použít dva již implementované TCP servery a jeden implementovat pomocí programovacího jazyka RAPID. Jedná se o TCP servery, které jsou spouštěny jako samostatné procesy v řídicí jednotce IRC5 a každý TCP server naslouchá na jiném portu. Názvy serverů včetně hodnot portů na kterých tyto servery naslouchají jsou uvedeny v tabulce 3.1.

■ Implementace TCP serveru

Každý TCP server je vytvořen v samostatném programovém modulu (kapitola 2.3.4), kde každý modul je umístěn v jiném souboru. Kromě programových modulů byly použity také systémové moduly (viz kapitola 2.3.4). Seznam názvů souborů umístěných v IRC5, které obsahují jednotlivé programové a systémové moduly použité pro komunikaci počítače s řídicí jednotkou IRC5 je následující:

- *ROS_socket.sys* (systémový modul) - Obsahuje funkce, které umožňují naslouchání serveru na daném portu, rozbalují/zabalují hlavičku dat a oddělují ji od dat samotných.
- *ROS_messages.sys* (systémový modul) - Obsahuje funkce, které na základě hlavičky určí typ dat a podle toho je rozbalí/zabalí.
- *ROS_common.sys* (systémový modul) - Obsahuje proměnné a struktury popisující trajektorii pohybu robota.
- *ROS_ioUnitServer.mod* (programový modul) - Implementuje server *ROS_ioUnitServer*.
- *ROS_motionServer.mod* (programový modul) - Implementuje server *ROS_motionServer*.
- *ROS_stateServer.mod* (programový modul) - Implementuje server *ROS_stateServer*.
- *ROS_motion.mod* (programový modul) - Jedná se o spustitelný programový modul, obsahující proceduru *main()*, který zpracovává data pro určení trajektorie robota a obsahuje instrukci *MoveJ* (viz 2.3), která provádí samotný pohyb robota.

Programové a systémové moduly *ROS_motionServer.mod*, *ROS_stateServer.mod*, *ROS_motion.mod*, *ROS_messages.sys*, *ROS_socket.sys* a *ROS_common.sys* byly vytvořeny pro účely ovládní průmyslových robotů ABB a jsou součástí balíčků projektu ROS-industrial (kapitola 2.4.2).

Námi vytvořený programový modul *ROS_ioUnitServer.mod* implementuje TCP server *ROS_ioUnitServer*, který naslouchá na daném portu (číslo portu viz tabulka 3.1) po spuštění řídicí jednotky IRC5. Po připojení klienta se čeká na data pro aktivaci nebo deaktivaci jednotlivých digitálních výstupů jednotky DSQ652 (popis jednotky viz kapitola 2.3.2). Pro správnou funkci námi vytvořeného TCP serveru bylo třeba použít funkce systémového modulu *ROS_socket.sys* a implementovat proceduru (popis procedury viz kapitola

2.3.4) *ROS_receive_msg_io_data* v systémovém modulu *ROS_messages.sys*. Tato procedura zpracuje přijatá data a uloží je do námi vytvořené datové struktury *ROS_msg_io_data* (popis dat v této struktuře viz tabulka 3.2).

Proces zpracování dat pomocí námi vytvořeného modulu má dvě odlišné části. První část je implementována ve funkci *set_end_effort_DO* a zpracovává data určená digitální výstupy, které ovládají prsty chapadla. V našem případě jsou to digitální výstupy 1 a 2 (viz schéma zapojení 3.4), proto se v této funkci zpracovávají pouze první a druhý bit prvního bajtu přijatých dat. Druhá část je implementována ve funkci *set_DO* a zpracovává data určená pro ostatní digitální výstupy.

Při implementaci funkce *set_end_effort_DO* bylo potřeba vytvořit algoritmus, který zabrání vzniku chybového stavu při ovládní prstů chapadla (popis chybového stavu viz kapitola 3.1.2). Tento algoritmus provádí následující: nedovolí sepnout oba výstupy najednou, při sepnutí jednoho z výstupů tento výstup po 180 ms znovu rozeptne a pak počká 15 ms, během procesu nastavování těchto digitálních výstupů nelze tento proces nějak ovlivnit a to ani v případě přijetí nových dat. Čas 15 ms vychází z grafu 3.2 a čas 180 ms byl určen experimentálně na základě ovládní prstů chapadla (popis ovládní prstů chapadla viz 3.1.2). Jedná se o nejmenší možnou délku pulzu na daném digitálním výstupu, která umožní plné otevření nebo plné zavření prstů chapadla. Funkce *set_DO* nastaví digitální výstupy 3 až 16 podle přijatých dat bez jakéhokoli časového omezení omezení.

3.3 Rozmístění stanic vůči robotu

Rozmístění stanic bylo provedeno s ohledem na dosah chapadlem robota a co nejkratší čas provádění rozplňovacího cyklu. Vhodný prostor pro umístění stanic na našem pracovišti je před robotem, proto jsou všechny stanice umístěny v části prostoru před robotem vzhledem k základně robota. Stanice jsou umístěny těsně vedle sebe z důvodu úspory času při přechodu robotem od jedné stanice k druhé. Výjimku tvoří výstupní část do které patří stanice uzavírající lahvičky a stanice s naplněnými lahvičkami, které jsou umístěny odděleně.

Jediné možné vzájemné ovlivnění výsledků měření by mohlo nastat u stanice měřící aktivitu a stanice pro dávkování radioaktivní látky. Tyto stanice jsou umístěny těsně vedle sebe, ovšem pro dobré stínící vlastnosti měřící stanice je zkrácení naměřených dat způsobené radioaktivním zářením dávkovací stanice zanedbatelné.

3.4 Použité vizualizační modely a kolizní modely v ROSu

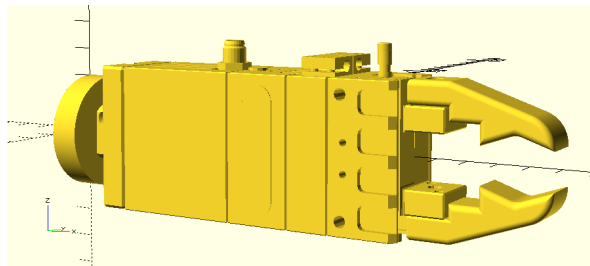
3.4.1 Vizualizační model robota ABB

Vizualizační model robota je součástí balíčku *abb_irt_1200_support* (popis balíčku je v kapitole 3.7.1), který je součástí projektu ROS_industrial (kapitola 2.4.2). Kompletní vizualizační model robota je složen z jednotlivých modelů, které rozměrově odpovídají jednotlivým ramenům robota. Tyto vizualizační modely byly vytvořeny autory: Stephan Wirth, Bianca Homberg a Jonathan Binney. Jednotlivé STL soubory těchto modelů jsou dostupné v rámci projektu projektu *abb_experimental* [35]. Informace o propojení těchto modelů jsou uvedeny v souboru *irt1200_7_70_macro.xacro*.

3.4.2 Vizualizační model chapadla

Vizualizační model chapadla (obrázek 3.7) je sestaven z následujících částí:

- Vizualizační modely obou částí příruby, které byly vytvořeny panem Ing. Danielem Seifertem, Ph.D..
- Vizualizační model chapadla schunk, který byl získán ze stránek výrobce [43].
- Vizualizační modely dvou prstů chapadla, které byly vytvořeny panem Ing. Danielem Seifertem, Ph.D..



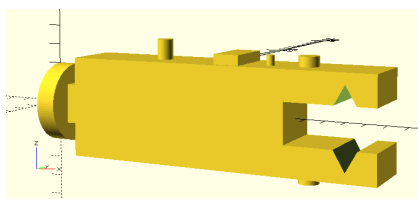
Obrázek 3.7: Obrázek vizualizačního modelu chapadla

3.4.3 Kolizní modely

Kolizní modely stanic, robota a chapadla jsou použity pro výpočet vzájemných kolizí při plánování trajektorie pohybu robota. Pro urychlení tohoto výpočtu je žádoucí tyto modely zjednodušit oproti jednotlivým vizualizačním modelům. Pro správné výpočty kolizí je důležité, aby tyto kolizní modely zcela překrývaly vizualizační modely z hlediska jejich rozměrů.

Kolizní model robota je součástí balíčku *abb_irt_1200_support* a je dostupný ze stejného zdroje jako vizualizační model robota. Kolizní model pracoviště byl vytvořen podle rozmístění jednotlivých stanic na maketě pracoviště.

Kolizní model chapadla (obrázek 3.8) je složen z kolizního modelu příruby, kolizního modelu chapadla a kolizního modelu prstů chapadla. Kolizní modely prstů jsou v systému ROS nastaveny jako nepohyblivé. Pro zamezení nesprávné detekce kolize při otevírání nebo zavírání prstů chapadla byly polohy prstů chapadla nastaveny do polohy, která odpovídá otevřenému chapadlu. Pokud bychom nastavili kolizní modely prstů do polohy zavřeno, tak by mohlo dojít k nesprávné detekci kolize při pohybu chapadla ve stavu otevřeno poblíž jednotlivých stanic. Rozměry našeho kolizního modelu chapadla byly převzaty z technického výkresu dostupného na webových stránkách výrobce chapadla [43].

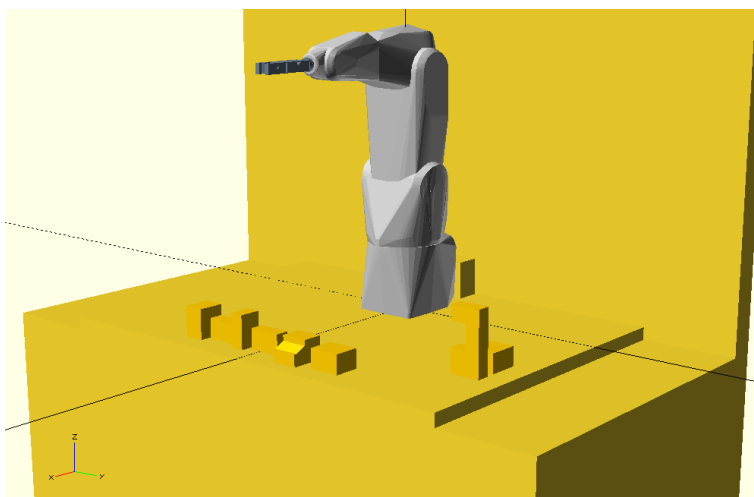


Obrázek 3.8: Obrázek kolizního modelu chapadla

Informace o propojení těchto kolizních modelů jsou uvedeny v souboru *irt1200_7_70_macro.xacro*. Všechny tyto kolizní modely jsou zobrazeny na obrázku 3.9.

3.5 Výpočet souřadných soustav

Popis poloh (tuhých) těles v prostoru je nezbytný pro plánování trajektorie pohybu robota při vykonávání rozplňovacího procesu (popis plánování trajektorie je uveden v kapitole 2.4.2). Polohy těchto těles v prostoru mají 6 stupňů volnosti a pro jejich popis jsme použili transformační matice, které popisují polohu souřadnicového systému jednotlivých těles v nepohyblivém



Obrázek 3.9: Obrázek kolizních modelů robota, chapadla a okolního prostoru robota

souřadnicovém systému svázaným se základnou robota (vyobrazení tohoto souřadnicového systému je na obrázku 3.10).

■ Transformační matice

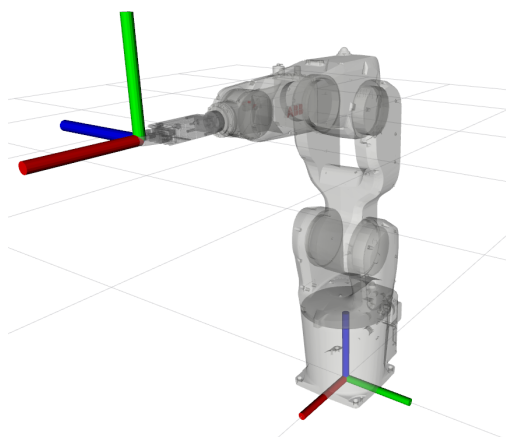
Transformační matice má 4 řádky a 4 sloupce a v této práci ji budeme značit \mathbf{T} . Obecný tvar této transformační matice je následující:

$$\mathbf{T} = \left[\begin{array}{ccc|c} \mathbf{R} & & & \vec{v} \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (3.1)$$

Kde \mathbf{R} je rotační matice, která reprezentuje rotaci souřadného systému a \vec{v} je sloupcový vektor délky tři, který stanovuje posuv souřadného systému. Podrobnější popis je uveden v literatuře [44].

■ Souřadnicové systémy robota

Kinematický model robota s vyznačenými souřadnicovými systémy chapadla a základny robota je zobrazen na obrázku 3.10. Červená barva znázorňuje souřadnicovou osu x , zelená barva znázorňuje souřadnicovou osu y a modrá barva souřadnicovou osu z .



Obrázek 3.10: Obrázek kinematického modelu robota s vyznačenými polohami souřadnicových systémů chapadla a základny robota

■ Souřadnicové systémy stanic

Popis poloh jednotlivých stanic je rozdělena na tři části. První část popisuje transformace souřadnicových systémů stanic, které jsou svázány s polohami levých spodních rohů zadních stěn jednotlivých stanic v souřadnicovém systému základny robota. Druhá část popisuje transformace souřadnicových systémů, které jsou svázány s polohami pro uchopení lahviček na jednotlivých stanicích v souřadnicovém systému dané stanice. Třetí část obsahuje popis celkových transformací souřadnicových systémů poloh pro uchopení lahviček na jednotlivých stanicích v souřadnicovém systému základny robota. Souřadnicové systémy první a druhé části jsou vyobrazeny na obrázcích v kapitole 2.2.7 u jednotlivých stanic.

První část

Transformační matici, která odpovídá transformaci souřadnicového systému stanice v souřadnicovém systému základny robota budeme značit \mathbf{T}_{stat} . Tato transformace je provedena pomocí posuvu a rotace kolem souřadnicové osy z . Hodnoty souřadnic a rotací souřadnicových systémů jednotlivých stanic jsou uvedeny v tabulce 3.3.

Název stanice	Souřadnice x [cm]	Souřadnice y [cm]	Souřadnice z [cm]	Rotace kolem osy z [°]
Stojan na prázdné lahvičky	59	22	0	0
Vážíci stanice	58,9	10	0	0
Stanice s dávkováním ředícího roztoku	60	-3,7	0	0
Stanice s dávkováním radioaktivní látky	60	-20,7	0	0
Dávkovač	60	-9	0	0
Stanice pro měření aktivity	58,9	-27,6	0	0
Stanice uzavírající lahvičky	42	62,8	0	-90
Stojan na naplněné lahvičky	39,5	55	0	-90

Tabulka 3.3: Tabulka souřadnic a rotací souřadnicových soustav jednotlivých stanic v souřadnicové soustavě základny robota

Druhá část

Transformační matici, která odpovídá transformaci souřadnicové soustavy polohy pro uchopení lahvičky v souřadnicovém systému stanice budeme značit \mathbf{T}_{pos} a má tvar 3.2. Transformační matice \mathbf{T}_v provádí posuv, transformační matice $\mathbf{T}_{Rx}(\alpha)$ provádí rotaci kolem souřadnicové osy x o úhel α a transformační matice $\mathbf{T}_{Ry}(\beta)$ provádí rotaci kolem nové souřadnicové osy y o úhel β , která vznikne po rotaci kolem souřadnicové osy x . Hodnoty souřadnic a rotací souřadnicových systémů jednotlivých stanic jsou uvedeny v tabulce 3.4.

$$\mathbf{T}_{pos} = \mathbf{T}_v \mathbf{T}_{Rx}(\alpha) \mathbf{T}_{Ry}(\beta) \quad (3.2)$$

Celková transformace

Výsledná transformační matice \mathbf{T}_{all} má tvar 3.3 a popisuje polohu souřadnicového systému polohy pro uchopení lahvičky dané stanice v souřadnicovém systému základny robota.

$$\mathbf{T}_{all} = \mathbf{T}_{stat} \mathbf{T}_{pos} \quad (3.3)$$

Poloha uchopení lahvičky dané stanice	Souřadnice x [cm]	Souřadnice y [cm]	Souřadnice z [cm]	Rotace kolem osy x [°]	Rotace kolem osy y [°]
1. poloha ve stojanu na prázdné lahvičky	2,2	2,2	8,9	90	0
2. poloha ve stojanu na prázdné lahvičky	2,2	7,25	8,9	90	0
3. poloha ve stojanu na prázdné lahvičky	5,6	2,2	8,9	90	0
4. poloha ve stojanu na prázdné lahvičky	5,6	7,25	8,9	90	0
1. poloha ve vážící stanici	3,1	4,1	9,64	90	0
1. poloha ve stanici pro dávkování ředícího roztoku	2,1	2,6	8,41	90	0
2. poloha ve stanici pro dávkování ředícího roztoku	2,1	7,6	8,41	90	0
1. poloha ve stanici pro dávkování radioaktivní látky	2,1	2,6	8,41	90	0
2. poloha ve stanici pro dávkování radioaktivní látky	2,1	7,6	8,41	90	0
1. poloha v měřící stanici	3,1	3,1	11,4	90	0
1. poloha ve stanici uzavírající lahvičky	6,6	2,1	10,28	90	180
1. poloha ve stojanu na naplněné lahvičky	2,2	2,2	8,9	90	180
2. poloha ve stojanu na naplněné lahvičky	2,2	7,25	8,9	90	180
3. poloha ve stojanu na naplněné lahvičky	5,6	2,2	8,9	90	180
4. poloha ve stojanu na naplněné lahvičky	5,6	7,25	8,9	90	180

Tabulka 3.4: Tabulka souřadnic a rotací souřadnicových soustav poloh pro uchopení lahviček v souřadnicových soustavách jednotlivých stanic

3.6 Popis rozplňovacího procesu

Součástí naší práce je také popis možného řešení rozplňovacího procesu, který je prováděn na maketě pracoviště. Tento popis slouží převážně pro informační účely a informuje, jak přibližně přípravný a rozplňovací proces probíhá. Pro aplikaci tohoto popisu v praxi je potřeba některé kroky přípravného a rozplňovacího procesu více upřesnit podle zadaných požadavků na rozplňování radiofarmak a podle dostupných prostředků a přístrojů na konkrétním pracovišti. Pro demonstraci rozplňovacího procesu na naší maketě pracoviště je tento popis dostačující.

■ 3.6.1 Přípravný proces prováděný obsluhou

Před prováděním přípravného a rozplňovacího procesu robotem je potřeba, aby obsluha provedla následující přípravy:

1. Obsluha vybalí lahvičky z obalu a vloží do vstupního stojanu.
2. Obsluha vloží lahev na vstupní aktivitu do stanice pro dávkování radioaktivní látky.
3. Obsluha vloží lahev s ředícím roztokem do stanice pro dávkování ředícího roztoku.
4. Obsluha potvrdí pro zahájení rozplňovacího procesu. [17]

Potvrzení zahájení rozplňovacího procesu je v našem případě formou GUI.

■ 3.6.2 Přípravný proces prováděný robotem

Přípravný proces prováděný robotem je popsán pomocí následujících kroků:

1. Robot se přemístí z počáteční pozice na stojan dávkovací stanice pro radioaktivní látky a uchopí prázdnou lahev na vstupní aktivitu.
2. Robot přesune lahev na vstupní aktivitu ze stojanu dávkovací stanice pro radioaktivní látky na vážící stanici.
3. Vážící stanice zváží lahev na vstupní aktivitu.
4. Robot přenesla lahev na vstupní aktivitu z vážící stanice zpět do stojanu dávkovací stanice.
5. Proveďte se lahev vstupní aktivitou ze zdroje aktivity.
6. Robot přemístí naplněnou lahev ze stojanu dávkovací stanice pro radioaktivní látky na vážící stanici.
7. Vážící stanice zváží naplněnou lahev.
8. Robot přenesla lahev z vážící stanice do stanice pro měření aktivity.
9. Stanice změří aktivitu radionuklidu v lahvi.
10. Robot přenesla lahev zpět do stojanu dávkovací stanice.
11. Přípravný proces je ukončen, robot se přemístí do počáteční pozice. [17]

Pro vypočtení objemu vstupní aktivity v lahvi je potřeba zvážit nejprve prázdnou a poté znovu naplněnou lahev. Objem roztoku v lahvi se přepočítá z naměřených hmotností.

3.6.3 Rozplňovací proces

1. Robot se přemístí na stojan s prázdnými lahvičkami a pomocí chapadla uchopí prázdnou lahvičku.
2. Robot přenese prázdnou lahvičku ze stojanu s prázdnými lahvičkami na vážící stanici.
3. Vážící stanice zváží lahvičku.
4. Robot přenese zvaženu lahvičku z vážící stanice do stanice pro dávkování ředícího roztoku.
5. Lahvička je naplněna určeným množstvím fyziologického roztoku.
6. Robot přemístí lahvičku ze stanice pro dávkování ředícího roztoku do stanice pro dávkování radioaktivní látky.
7. Stanice pro dávkování radioaktivní látky přidá do lahvičky požadované množství radioaktivní látky.
8. Robot přenese naplněnou lahvičku ze stanice pro dávkování radioaktivní látky do stanice pro měření aktivity.
9. Stanice změří aktivitu radiofarmaka v lahvičce.
10. Robot přemístí lahvičku z měřící stanice na vážící stanici.
11. Vážící stanice zváží lahvičku.
12. Proběhne výpočet zda radiofarmakum v lahvičce odpovídá požadovaným hodnotám z hlediska aktivity a objemu určeného z hmotností naplněné a prázdné lahvičky.
13. Pokud je lahvička naplněna správně, robot ji přenese z vážící stanice do stanice uzavírající lahvičky a skočí se na krok 17.
14. Pokud lahvička není naplněna správně a oprava radiofarmaka v lahvičce pomocí dávkování je možná, tak ji robot přenese z vážící stanice do stanice pro dávkování fyziologického roztoku a proces se opakuje od kroku číslo 5.
15. Pokud lahvička není naplněna správně a oprava radiofarmaka v lahvičce pomocí dávkování není možná, tak ji robot přenese z vážící stanice na stojan pro pokažené vzorky.

16. Lahvička je odstraněna z evidence vzorků v dávce a rozplňovací proces se začne provádět od kroku číslo 1.
17. Stanice uzavírající lahvičky uzavře nadávkovanou lahvičku.
18. Robot přenesení lahvičku ze stanice uzavírající lahvičky do stojanu na naplněné lahvičky a zde ji odloží.
19. Pokud jsou naplněny všechny lahvičky z dávky, rozplňovací proces je ukončen a robot se přemístí do počáteční pozice.
20. Pokud nejsou naplněny všechny lahvičky z dávky, rozplňovací proces začne znovu od kroku číslo 1. [17]

Na naší maketě pracoviště nejsou k dispozici žádná naměřená data z měřících stanic, proto je krok číslo 12 přeskočen a s lahvičkou je zacházeno jakoby byla rozplněna správně.

3.7 Softwarová implementace

3.7.1 Vytvořené balíčky

Pro správnou funkci rozplňovacího cyklu vykonávaného robotem na sestavené maketě pracoviště bylo potřeba vytvořit čtyři balíčky a použít několik již vytvořených balíčků. Všechny použité a vytvořené balíčky lze rozdělit do tří projektů (popis projektů viz kapitola 2.4.2). Názvy těchto projektů jsou *abb*, *abb_experimental* a *industrial_core*. Balíčky, které jsou obsaženy v jednotlivých projektech jsou následující:

```

abb
├── abb_driver
├── abb_hw
├── abb_experimental
│   ├── abb_irb1200_7_70_move
│   ├── abb_irb1200_7_70_moveit_config
│   ├── abb_irb1200_support
│   └── abb_rqt_cpp
├── industrial_core
│   ├── industrial_deprecated
│   ├── industrial_msgs
│   ├── industrial_robot_client
│   ├── industrial_robot_simulator
│   ├── industrial_trajectory_filters
│   └── industrial_utils

```

└─ industrial_msgs

Zaměříme se pouze na detailní popis balíčků, které jsme vytvořili. Ostatní balíčky, které byly v naší práci použity jsou součástí projektu ROS-Industrial (kapitola 2.4.2).

■ **abb_hw**

Balíček *abb_hw* obsahuje soubory umožňující ovládání digitálních výstupů řídicí jednotky IRC5. Tyto soubory můžeme rozdělit na tři kategorie podle jejich využití. První kategorií je zdrojový kód napsán v programovacím jazyce C++, který implementuje uzel */abb_io_unit* (popis uzlu viz kapitola 3.7.2). Název souboru ve kterém je uložen tento zdrojový kód je *CBridge.cpp* (obsah tohoto souboru je na přiloženém CD viz příloha D).

Druhá kategorie zahrnuje skupinu souborů ve kterých jsou implementovány systémové a programové moduly jazyka RAPID (popis jazyka RAPID je uveden v kapitole 2.3.4). Tyto soubory jsou uloženy v řídicí jednotce IRC5. Jejich seznam a popis viz kapitola 3.2.4.

Třetí kategorií je spustitelný soubor s názvem *robot_hw.launch*, který spouští uzel */abb_io_unit*. Tento soubor je použit, protože umožňuje spouštění uzlu */abb_io_unit* z libovolného balíčku v celém projektu.

■ **abb_rqt_cpp**

Balíček *abb_rqt_cpp* obsahuje soubory, ve kterých je implementováno GUI a veškerá funkčnost jednotlivých ovládacích prvků tohoto GUI (popis GUI je uveden v kapitole 2.5 a implementace GUI v kapitole 3.7.4).

■ **abb_irb1200_7_70_move**

Balíček *abb_irb1200_7_70_move* obsahuje zdrojové soubory uzlu */move* (popis uzlu viz kapitola 3.7.2), který provádí rozplňovací cyklus na námi vytvořené maketě pracoviště. Podrobný popis implementace rozplňovacího cyklu pomocí programovacího jazyka C++ je uveden v kapitole 3.7.5.

■ `abb_irt1200_7_70_moveit_config`

Balíček `abb_irt1200_7_70_moveit_config` obsahuje spustitelné a konfigurační soubory, které spouští a předávají vstupní parametry launch souborům v jednotlivých balíčcích, které spouští a nastavují jednotlivé uzly potřebné pro vykonání rozplňovacího cyklu.

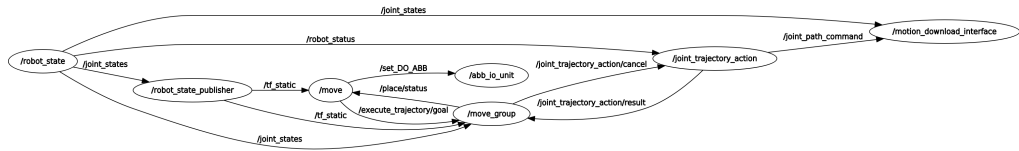
Tento balíček byl vytvořen pomocí Moveit Setup Assistant (viz kapitola 2.4.2). Vygenerované soubory ovšem podporují pouze simulaci pohybů modelu robota pomocí programu Rviz. Pro ovládní skutečného robota bylo potřeba vytvořit soubor s názvem `moveit_planning_execution.launch` uvnitř tohoto balíčku a upravit parametry některých souborů podle následujícího návodu [45].

■ 3.7.2 Použité a vytvořené uzly

V naší práci bylo použito celkem 10 vzájemně komunikujících uzlů. Jejich vzájemné propojení je reprezentováno pomocí orientovaného grafu, kde vrcholy grafu představují jednotlivé uzly včetně jejich názvů a orientované hrany představují komunikaci pomocí topics a zároveň udávají směr komunikace. Programový nástroj 3.11 [38] ve kterém lze vytvořit tento graf umožňuje pouze dvě možnosti zobrazení. Zobrazení všech spuštěných uzlů a jejich vzájemné propojení nebo zobrazení pouze, které přímo souvisí pouze se samotným ovládním pohybu robota a komunikací s řídicí jednotkou robota. V našem případě se jedná o sedm uzlů, které přímo souvisí s robotem (zobrazení těchto uzlů je na obrázku 3.11). Obrázek zobrazující všechny uzly a jejich propojení v grafu je pro své velké rozměry umístěn pouze na přiloženém CD pod názvem `rosgraph_all.png` D.

Kompletní seznam použitých uzlů je následující:

1. `/move`
2. `/abb_io_unit`
3. `/rqt_gui_cpp_node_n`
4. `/motion_download_interface`
5. `/robot_state`
6. `/move_group`
7. `/robot_state_publisher`
8. `/joint_trajectory_action`

9. `/rviz_n`10. `/ros_out`

Obrázek 3.11: Graf použitých uzlů včetně jejich propojení

Uzly `/motion_download_interface` a `/robot_state` byly vytvořeny pro účely komunikace mezi počítačem a řídicí jednotkou IRC5 průmyslových robotů ABB a jsou implementovány v souborech, které jsou součástí balíčku `abb_driver`. Uzly `/move_group`, `/robot_state_publisher` a `/joint_trajectory_action` souvisí s plánováním a ovládáním pohybu robota. Všechny tyto uzly jsou součástí projektu ROS-Industrial (popis je uveden v kapitole 2.4.2). Uzly `/rviz_n` a `/ros_out` jsou základní součástí systému ROS a programu Rviz.

Námi vytvořené uzly jsou `/move`, `/abb_io_unit` a `/rqt_gui_cpp_node_n`, kde n je celé číslo přidělené systémem ROS při spuštění tohoto uzlu. Jedná se o uzly, které jsou implementovány pomocí programovacích jazyků C++ a Python.

■ `/move`

Uzel `/move` řídí rozplňovací cyklus a provádí veškeré výpočetní operace související s rozplňováním. Jedná se například o výpočet celkového postupu rozplňovacího cyklu, výpočet souřadnic bodů v prostoru pro pohyb robotické ruky, atd. Podrobný popis implementace tohoto uzlu je uveden v kapitole 3.7.5

■ `/abb_io_unit`

Uzel `/abb_io_unit` spouští a plní funkci TCP klienta (viz kapitola 3.2.1), který slouží pro ovládání digitálních výstupů modulu DSQ652 (viz kapitola 2.3.2). Tento uzel také spouští subscriber, který naslouchá na topic `/set_DO_ABB` (popis tohoto topic je uveden v kapitole 3.7.3). Data, která jsou přijata subscriberem (viz tabulka 3.6) jsou převedena na data (viz tabulka 3.2), která jsou do řídicí jednotky IRC5 odeslána TCP klientem. Tabulka 3.5 popisuje převod dat z topic `/set_DO_ABB` na data odesílaná do IRC5 pomocí TCP klienta.

Příchozí data z topic /set_DO_ABB	Data odeslaná TCP klientem s hodnotami uvedenými v bitech [DO_data_2, DO_data_1]	Velikost odchozích dat v bajtech
"SetDO1"	[0000 0000, 0000 0001]	2
"SetDO2"	[0000 0000, 0000 0010]	2

Tabulka 3.5: Převod dat z topic /set_DO_ABB (kapitola 3.7.3) do podoby dat odeslaných pomocí TCP klienta (kapitola 3.2)

Uzel /abb_io_unit je implementován v souborech, které jsou součástí balíčku abb_hw a projektu abb.

■ /rqt_gui_cpp_node_n

Uzel /rqt_gui_cpp_node_n spouští a ovládá GUI (popis GUI je v kapitole 2.5). Tento uzel také spouští a obstarává dva subscribers, které přijímají data z daných topics a jeden publisher, který odesílá data na daný topic při jakékoliv změně ovládacího prvku GUI. Jeden subscriber slouží pro příjem dat, která se převádí na text a rovnou se vypisují do textového pole GUI a druhý subscriber přijímá data, která obsahují informaci jak nebo na jaké hodnoty se mají určité ovládací prvky GUI. Informace o posílaných datech a názvech daných topic jsou uvedeny v kapitole 3.7.3.

■ 3.7.3 Vytvořené topics

Zaměříme se pouze na popis topic, které jsme vytvořili ostatní topic jsou součástí projektu ros_industrial (viz literatura [32]). Obsah dat posílaných pomocí všech topics lze zjistit pomocí programu rqt viz kapitola 2.4.2. Veškerá data posílaná přes námi vytvořenými topics jsou datového typu string. Tento datový typ byl zvolen, protože data jsou pomocí stringu lépe čitelná pro uživatele oproti ostatním datovým typům (například datový typ int). Při tvorbě těchto dat byl kladen důraz na formulaci obsahu dat, aby co nejlépe vystihoval co daná data ovládají nebo nastavují. V porovnání s datovým typem int jsou data o řádově jednotky bajtů větší, což je z hlediska rychlosti přenosu dat zanedbatelný rozdíl.

■ /set_DO_ABB

Pomocí topic /set_DO_ABB se posílají data, která slouží pro nastavování digitálních výstupů řídicí jednotky IRC5. Publisher tohoto topic implementován a spouštěn uzlem /move a subscriber uzlem /abb_io_unit .

Konkrétní data posílaná tímto topic jsou uvedeny v tabulce 3.6.

Datový typ	Data	Velikost dat v bajtech	Popis
string	"SetDO1"	6	Data pro nastavení DO 1 v IRC5
string	"SetDO2"	6	Data pro nastavení DO 2 v IRC5

Tabulka 3.6: Data posílaná přes topic /set_DO_ABB

■ /abb_rqt_cpp/msg_to_gui

Pomocí topic /abb_rqt_cpp/msg_to_gui se posílají data, která slouží pro nastavování ovládacích prvků v GUI. Publisher tohoto topic implementován a spuštěn uzlem /move a subscriber uzlem /rqt_gui_cpp_node_n. Konkrétní data posílaná tímto topic jsou uvedeny v tabulce 3.7.

Datový typ	Data	Velikost dat v bajtech	Popis
string	"RequestToVialNum"	16	Po přijetí těchto dat je odeslán počet rozplňovaných lahvíček na topic "/abb_rqt_cpp/state_from_gui"
string	"Init"	4	Nastaví ovládací prvky GUI do počáteční podoby
string	"SetVial_IN_n_i"	14	Aktivuje (i = "1")/Deaktivuje (i = "0") zaškrtačovací pole 1 až 4 (n = "1" až "4") vstupní stanice v GUI
string	"SetVial_OUT_n_i"	15	Aktivuje (i = "1")/Deaktivuje (i = "0") zaškrtačovací pole 1 až 4 (n = "1" až "4") výstupní stanice v GUI
string	"EnablePN"	8	Aktivuje tlačítka "Další" a "Předchozí" v GUI
string	"DisablePBtn"	11	Deaktivuje tlačítko "Předchozí" v GUI

Tabulka 3.7: Data posílaná přes topic /abb_rqt_cpp/msg_to_gui

■ /abb_rqt_cpp/prg_to_gui

Pomocí topic /abb_rqt_cpp/prg_to_gui se posílají data, která nastavují hodnoty indikátorům průběhu v GUI 2.12 (ovládací prvky GUI s čísly 2 a 3) a text v textovém poli (ovládací prvek GUI s číslem 1), který informuje o aktuálním prováděním procesu rozplňování pomocí robota. Publisher tohoto topic je implementován a spouštěn uzlem /move a subscriber uzlem /rqt_gui_cpp_node_n.

Data odesílaná tímto topic jsou složena ze tří různých částí. Protože string je datový typ posílaných dat, je pro odlišení jednotlivých částí použit oddělovač typu bílý znak a to konkrétně běžná mezera. Jednotlivé části posílaných dat jsou následující:

"info_msg prg totalPrg"

Kde *info_msg* je text informující o aktuálním prováděním procesu rozplňování pomocí robota, *prg* je hodnota typu string odpovídající číselné hodnotě v rozsahu 0 až 100, která informuje o postupu rozplňování jednoho vzorku z dávky (jedna lahvička) a *totalPrg* je hodnota typu string odpovídající číselné hodnotě v rozsahu 0 až 100, která informuje o postupu rozplňování celé dávky (všechny lahvičky). Text *info_msg* je často víceslovný. Oddělovačem těchto slov je symbol `_`. Texty použité v naší práci jsou následující:

- *Uchopení_n._lahvičky* - Proces uchopení n-té (n = 1 až 4) lahvičky ve stojanu na prázdné lahvičky.
- *Vážení_lahvičky* - Proces umístění lahvičky na vážící stanici.
- *Dávkování_fyziologického_roztoku* - Proces umístění lahvičky do stanice pro dávkování ředícího roztoku.
- *Dávkování_radionuklidu* - Proces umístění lahvičky do stanice pro dávkování radioaktivní látky.
- *Měření_aktivity* - Proces umístění lahvičky do stanice pro měření aktivity.
- *Vážení_nadávkové_lahvičky* - Proces umístění nadávkové lahvičky na vážící stanici.
- *Uzavření_lahvičky* - Proces umístění lahvičky do stanice uzavírající lahvičky.
- *Odložení_n._lahvičky* - Proces odložení n-té (n = 1 až 4) lahvičky do stojanu na naplněné lahvičky.

- *Příprava prázdné lahvičky* - Proces uchopení prázdné lahvičky ve stojanu pro dávkování aktivity.
- *Plnění lahvičky vstupní aktivitou* - Proces umístění prázdné lahvičky do stojanu pro dávkování aktivity, kde je tato lahvička naplněna vstupní aktivitou.
- *Měření vstupní aktivity* - Proces umístění lahvičky se vstupní aktivitou do stanice pro měření aktivity.
- *Příprava vstupní aktivity* - Proces umístění lahvičky se vstupní aktivitou do stojanu pro dávkování aktivity, kde je provedena příprava na dávkování této aktivity.

Popis jednotlivých stanic je uveden v kapitole 2.2.7.

■ /abb_rqt_cpp/state_from_gui

Pomocí topic /abb_rqt_cpp/state_from_gui se posílají data, která informují o provedené akci v GUI. Publisher tohoto topic je implementován a spuštěn uzlem /rqt_gui_cpp_node_n a subscriber uzlem /move. Konkrétní data posílaná tímto topic jsou uvedeny v tabulce 3.8.

■ 3.7.4 Implementace GUI

Námi vytvořené GUI je součástí programu rqt ve formě pluginu s názvem *Dosage process ABB*. Tento plugin je implementován v programovacím jazyce C++ a rozmístění ovládacích prvků v GUI bylo provedeno pomocí vývojového prostředí *Qt Creator*.

Při implementaci bylo nejprve potřeba vytvořit prázdný plugin a zakomponovat ho do programu rqt. Vytvoření prázdného pluginu umožňuje knihovna, která byla vytvořena Lucasem Walterem [46]. Veškeré soubory se zdrojovými kódy jsou součástí balíčku *abb_rqt_cpp*.

■ 3.7.5 Implementace rozplňovacího procesu v systému ROS

Implementace rozplňovacího procesu byla provedena v programovacím jazyce C++ ve vývojovém prostředí *Clion*. Při implementaci bylo potřeba provádět transformace jednotlivých souřadných soustav v prostoru.

Datový typ	Data	Velikost dat v bajtech	Popis
string	"Start"	5	Informuje o stisknutém tlačítku <i>Start</i> v GUI
string	"Stop"	5	Informuje o stisknutém tlačítku <i>Stop</i> v GUI
string	"End"	3	Informuje o ukončení GUI
string	"Dosage"	6	Informuje o stisknutém tlačítku <i>Rozplňovací proces</i> v GUI
string	"Prep"	4	Informuje o stisknutém tlačítku <i>Přípravný proces</i> v GUI
string	"Auto"	4	Informuje o stisknutém tlačítku <i>Automatický</i> v GUI
string	"Manual"	6	Informuje o stisknutém tlačítku <i>Krokový</i> v GUI
string	"Next"	4	Informuje o stisknutém tlačítku <i>Další</i> v GUI
string	"Prev"	4	Informuje o stisknutém tlačítku <i>Předchozí</i> v GUI
string	n	1	Odešle počet rozplňovaných lahviček (n = "1" až "4")

Tabulka 3.8: Data posílaná přes topic `/abb_rqt_cpp/state_from_gui`

Tato transformace v programovacím jazyce C++ byla provedena pomocí funkcí poskytovaných knihovnou *Eigen* [47]. Všechny soubory se zdrojovými kódy jsou součástí balíčku `abb_irb1200_7_70_move`.

Všechny nástroje systému ROS jsou nastaveny nebo implementovány pro verzi *ROS Kinetic Kame* [48], která byla v době začátku implementace rozplňovacího procesu nejnovější dostupná stabilní verze systému ROS.

3.8 Uživatelská část

3.8.1 Ovládání robota během rozplňovacího procesu

Ovládání pomocí GUI

Pro správné provedení rozplňovacího procesu je potřeba tento proces nastavit a ovládat pomocí vytvořeného GUI (popis GUI je v kapitole 2.5).

■ Ovládání pomocí řídicí jednotky IRC5

Řídicí jednotka IRC5 je po celou dobu provádění rozplňovacího procesu nadřazena systému ROS. Proto je možné zastavit provádění pohybů robota pomocí tlačítek na ovládací jednotce FlexPendant nebo okamžitě zastavit pohyb robota pomocí tlačítek *emergency STOP* umístěných přímo na řídicí jednotce IRC5 (obrázek 2.8) a na ovládací jednotce FlexPendant (obrázek 2.10). Po zastavení pohybu robota jednou z výše uvedených možností dojde k narušení průběhu rozplňovacího procesu prováděného systémem ROS. Pro obnovení pohybu je třeba uvolnit tlačítka *emergency STOP*, restartovat celý program v systému ROS a tím pádem začít od začátku rozplňovacího procesu.

■ 3.8.2 Nastavení a ovládání GUI

Po spuštění GUI je možnost volby mezi stisknutím dvou tlačítek s názvy *Přípravný proces* a *Rozplňovací proces*. Tyto názvy odpovídají procesům, které se budou vykonávat. Po stisknutí jednoho z těchto tlačítek už nelze měnit zvolený proces, dokud se tento proces neprovede. Po provedení zvoleného procesu se všechny ovládací prvky v GUI nastaví do počátečního stavu. Při ukončení GUI dojde k zastavení robota a ukončení provádění rozplňovacího procesu. Po opětovném spuštění GUI jsou ovládací prvky nastaveny do počátečního stavu a znovu se čeká na výběr procesu. Textové pole s názvem *Informační okno* v každém kroku informuje uživatele co je třeba udělat.

■ Přípravný proces

Přípravný proces odpovídá krokům popsaným v kapitole 2.2.3. Tento proces lze provádět pouze v automatickém režimu.

■ Rozplňovací proces

Rozplňovací proces vykonává celý proces rozplňování podle kroků popsaných v kapitole 2.2.3. Tento proces je možný v krokovém nebo automatickém režimu mezi kterými lze kdykoliv přecházet pomocí tlačítek s výběrem režimu. Po změně režimu dojde k automatickému zastavení robota pokud se zrovna pohybuje. Spuštění režimu je pak znovu možné pomocí tlačítka s názvem *Start*.

■ Automatický režim

Automatický režim provádí rozplňovací nebo přípravný proces nepřetržitě dokud nejsou provedeny všechny kroky. Během provádění je možné proces kdykoliv zastavit a poté znovu spustit.

■ Krokový režim

Krokový režim je možný pouze u rozplňovacího procesu. V tomto režimu je celý proces rozdělen na několik kroků mezi kterými lze přecházet dopředu a nazpět pomocí tlačítek s názvy *Předchozí* a *Další*. Během přechodu mezi jednotlivými kroky lze proces zastavit a poté znovu spustit.

■ 3.8.3 Spuštění GUI pomocí systému ROS

Pro spuštění GUI a všech dalších programů souvisejících s vykonáváním rozplňovacího procesu je třeba provést následující kroky ve stanovém pořadí:

1. (IRC5) Zapnout IRC5, aktivovat automatický režim a spustit proceduru `main()` v modulu `ROS_motion`.
2. (Počítač) Spustit launch file s názvem `moveit_planning_execution.launch`, který má dva vstupní parametry `sim` a `robot_ip`. Parametr `sim` je datového typu `bool` a určuje zda se jedná o simulaci nebo o reálného robota. Parametr `robot_ip` je datového typu `string` a specifikuje ip adresu řídicí jednotky IRC5. Příkaz, který spustí tento soubor je následující:

```
„roslaunch abb_irb1200_7_70_moveit_config
moveit_planning_execution.launch
sim:=false robot_ip:=10.35.100.3“
```


Kapitola 4

Experimenty

4.1 Stanovení maximální rychlosti pohybu robota

Při rozplňovacím procesu je kladen důraz na rychlost a s tím související čas rozplňování. Z tohoto důvodu bylo potřeba provést tento experiment, který zjistí maximální možnou rychlost pohybu robota s ohledem na bezpečnost manipulace s naplněnými lahvickami. Rychlost pohybu robota je dána rychlostmi jednotlivých kloubů robota. Maximální rychlosti jednotlivých kloubů robota jsou uvedeny v tabulce 2.2. Tyto rychlosti lze měnit pomocí systému ROS.

Pro zjištění požadovaných rychlostí bylo potřeba provést následující úkony:

1. Rychlosti otáčení jednotlivých kloubů byly nastaveny na polovinu maximálních rychlostí.
2. Lahvička s celkovým objemem 20 ml byla naplněna přibližně 15 ml vody, což odpovídá maximálnímu objemu naplnění lahvičky v praxi.
3. Byl proveden rozplňovací proces a během něho jsme pozorovali, zda nedošlo k vylití vody z lahvičky.
4. Pokud vše proběhlo v pořádku, tak byly zvýšeny rychlosti jednotlivých kloubů o 10 % maximální rychlosti a rozplňovací proces byl spuštěn znovu. V opačném případě byly rychlosti sníženy o 10 % maximální rychlostí a znovu otestovány.

Po provedení těchto úkonů jsme došli k výsledku, že maximální rychlost otáčení všech kloubů robota je při pohybu robota plně dostačující pro bezpečnou manipulaci s lahvickami. Pro stanovení určité rezervy při rychlosti

pohybu robota jsme tuto maximální rychlost otáčení všech kloubů robota snížili o 10 %. Tuto rychlost budeme tedy považovat za maximální možnou při provádění pohybu robota.

4.2 Změření času rozplňovacího procesu

Jedním z nejdůležitějších požadavků na rozplňovací stanici je provedení rozplňování za co nejkratší čas. Důvodem je poločas rozpadu daného radionuklidu, který se bude rozplňovat. Proto jsme provedli tento experiment, ve kterém jsme změřili časy jednotlivých prováděných kroků na maketě pracoviště, které odpovídají krokům rozplňovacího procesu popsaného v kapitole 3.6.

Maximální rychlost pohybu robota při vykonávání rozplňovacího procesu byla určena na základě předchozího experimentu. Jedním z omezujících faktorů, který prodlužuje dobu rozplňování je zpoždění při ovládní prstů chapadla pomocí systému ROS. Při testování nejkratšího času pro otevření a opětovné zavření prstů chapadla bylo zjištěno zpoždění 0,77 s. Toto zpoždění je dáno dobou otevření a zavření prstů chapadla (0,42 s viz kapitola 3.1.2), minimálním časovým rozestupem mezi pulzy pro ovládní prstů (0,015 s viz kapitola 3.1.2) a zpožděním při zpracování dat v systému ROS a v řídicí jednotce IRC5 (0,2 s). Z těchto hodnot můžeme stanovit zpoždění pouze pro otevření nebo pouze pro zavření prstů chapadla. To činí 0,1 s pro zpracování dat pro ovládní prstů chapadla a 0,21 s pro otevření nebo zavření prstů chapadla.

Při provádění rozplňovacího procesu jsme u kroků sovisejících s jednotlivými stanicemi uvažovali toto zpoždění (0,77 s) plus dobu umístění lahvičky v dané stanici (0,22 s). Celkem je tedy tento čas roven 1 s. Naměřené časy jsou uvedeny v tabulce 4.1.

4.2.1 Zhodnocení naměřených časů

Z tabulky 4.1 udávající naměřené časy nám vyšlo, že celkový čas rozplňování jedné lahvičky je 65 s. Tento čas je bez započtení pohybu robota z výstupní stanice do počáteční stanice, protože přímo nesouvisí s rozplňováním. Tento čas také neobsahuje časy provádění měřících a dávkovacích operací jednotlivými stanicemi, protože na maketě pracoviště nejsou tyto stanice k dispozici a v praxi se mohou tyto časy lišit podle jejich způsobů provádění těchto operací na jednotlivých pracovištích.

Použití rozplňovací stanice je výhodné z hlediska bezpečnosti obsluhy a ekonomiky až při rozplňování velkého počtu vzorků (řádově stovky). Pokud budeme uvažovat čas rozplnění jedné lahvičky 65 s a budeme mít například 200 vzorků (lahviček) na rozplnění, tak je celkový čas rozplňovacího procesu 13000 s (3 hodiny a 36 minut). Pro porovnání tento čas je více než polovina poločasu rozpadu radionuklidu ^{99m}Tc . Kromě rozplňování je také potřeba tato radiofarmaka distribuovat do jednotlivých nemocnic a aplikovat je konkrétnímu pacientovi do těla. Tyto úkony zaberou také nějaký čas, který záleží mimo jiné na konkrétní lokalitě, kde je prováděno rozplňování. Z tohoto důvodu nelze přesně říci, zda je tento čas dostačující či nikoliv.

Krok	Čas [s]	Popis
1	8	Přesun robota z počáteční pozice na vstupní stanici nebo z výstupní na vstupní stanici
2	7	Přesun lahvičky
3	1	Vážení
4	6	Přesun lahvičky
5	1	Dávkování
6	7	Přesun lahvičky
7	1	Dávkování
8	7	Přesun lahvičky
9	1	Měření aktivity
10	9	Přesun lahvičky
11	1	Vážení
12	0	Výpočet nelze provést
13	8	Přesun lahvičky
14	0	Nelze provést bez kroku 12
15	0	Nelze provést bez kroku 12
16	0	Nelze provést bez kroku 12
17	1	Uzavírání lahvičky
18	7	Přesun a odložení lahvičky
19	(8)	Přesun robota z výstupní stanice do počáteční pozice
20	0	Pouze skok na krok 1
Celkem	65 (73)	Celkový čas rozplňování jedné lahvičky s nebo bez započtení času kroku 19

Tabulka 4.1: Popis časů jednotlivých kroků, které odpovídají krokům rozplňovacího procesu

Kapitola 5

Závěr

Cíle této bakalářské práce bylo navrhnout a zrealizovat maketu automatizované rozplňovací stanice, implementovat programové vybavení pro řízení tohoto pracoviště a nakonec provést a vyhodnotit experimenty. Všechny tyto části byly provedeny a zdokumentovány.

Maketa rozplňovací stanice je tvořena 3D modely, které vzdáleně připomínají reálné stanice a robotem, který provádí manipulaci s lahvičkami během rozplňovacího procesu. Ovládání tohoto robota bylo implementováno v systému ROS. Pro možnost ovládání a řízení celého rozplňovacího procesu bylo vytvořeno GUI, které signalizuje postup rozplňovacího procesu, informuje uživatele o právě prováděném kroku a také umožňuje měnit režimy pohybu robota. Režimy pohybu jsou automatický, kde robot vykonává celý rozplňovací proces bez nutné asistence obsluhy a krokovací, kde se robot pohybuje po krocích a vyžaduje potvrzení pohybu od uživatele pro každý krok. Velká výhoda implementace programového vybavení pomocí systému ROS spočívá především v univerzálnosti tohoto systému. Důvodem je umožnění použití toto řízení a ovládání pro jiného průmyslového robota s minimálními změnami při implementaci tohoto procesu.

Jedním ze základních požadavků na rozplňovací stanici je čas, který by měl být co nejmenší z důvodu omezené doby působení ionizujícího záření daného radionuklidu. Proto byly po implementaci programového vybavení pracoviště provedeny dva experimenty. První experiment se zaměřuje na určení maximální možné rychlosti pohybu robota provádějícího přesun lahviček. Výsledkem tohoto experimentu je, že rychlost otáčení jednotlivých kloubů robota může být při provádění rozplňování maximální možná, protože při přesunu lahviček robotem nedojde k vylití roztoku z lahvičky. Druhý experiment je zaměřen na měření časů provádění jednotlivých kroků rozplňovacího procesu. Výsledné časy zjištěné tímto experimentem byly porovnány s poločasem rozpadu radionuklidu ^{99m}Tc . Podle porovnání těchto časů můžeme říci, že tyto

časy nejsou úplně dostačující pro provádění rozplňovacího procesu velkého množství vzorků, a proto je potřeba celý proces ještě více optimalizovat z hlediska času.

Příloha A

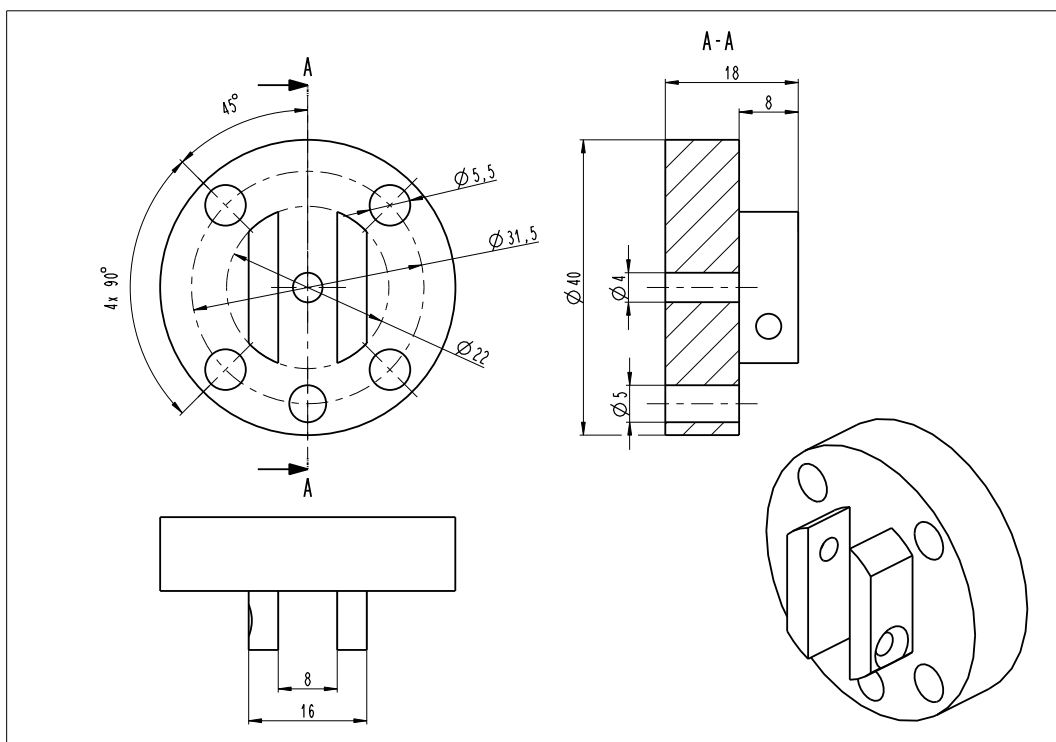
Literatura

- [1] ABB: *IRB 1200 [online]*. [cit. 2018-05-18]. Dostupné z: <http://search.abb.com/library/Download.aspx?DocumentID=9AKK106930A6205&LanguageCode=cs&DocumentPartId=&Action=Launch>
- [2] SCHUNK GmbH & Co. KG: *Assembly and Operating Manual : EGP [online]*. [cit. 2018-05-18]. Dostupné z: <https://schunk.com/fileadmin/pim/docs/IM0010510.PDF>
- [3] ABB: *Technical reference manual: RAPID Instructions, Functions and Data types [online]*. [cit. 2018-05-07]. Dostupné z: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf
- [4] Kupka, K.; Kubinyi, J.; Šámal, M.: *Nukleární medicína: [učební text]*. Praha: P3K, c2007, ISBN 9788090358492.
- [5] Majer, Vladimír: *Základy jaderné chemie*. Praha: Státní nakladatelství technické literatury, druhé vydání, 1981.
- [6] Wikipedie: Záření — Wikipedie: Otevřená encyklopedie [online]. 2019, [cit. 2019-05-21]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Z%C3%A1%C5%99en%C3%AD&oldid=17040059>
- [7] WikiSkripta: Ionizující záření — [online]. 2018, [cit. 2019-05-21]. Dostupné z: https://www.wikiskripta.eu/index.php?title=Ionizuj%C3%ADc%C3%AD_z%C3%A1%C5%99en%C3%AD&oldid=405220
- [8] WikiSkripta: Radionuklid — [online]. 2018, [cit. 2019-05-21]. Dostupné z: <https://www.wikiskripta.eu/index.php?title=Radionuklid&oldid=396200>
- [9] PŘÍRODNÍ RADIONUKLIDY [online]. c2019, [cit. 2019-05-04]. Dostupné z: <http://fbmi.sirdik.org/4-kapitola/42/422.html>

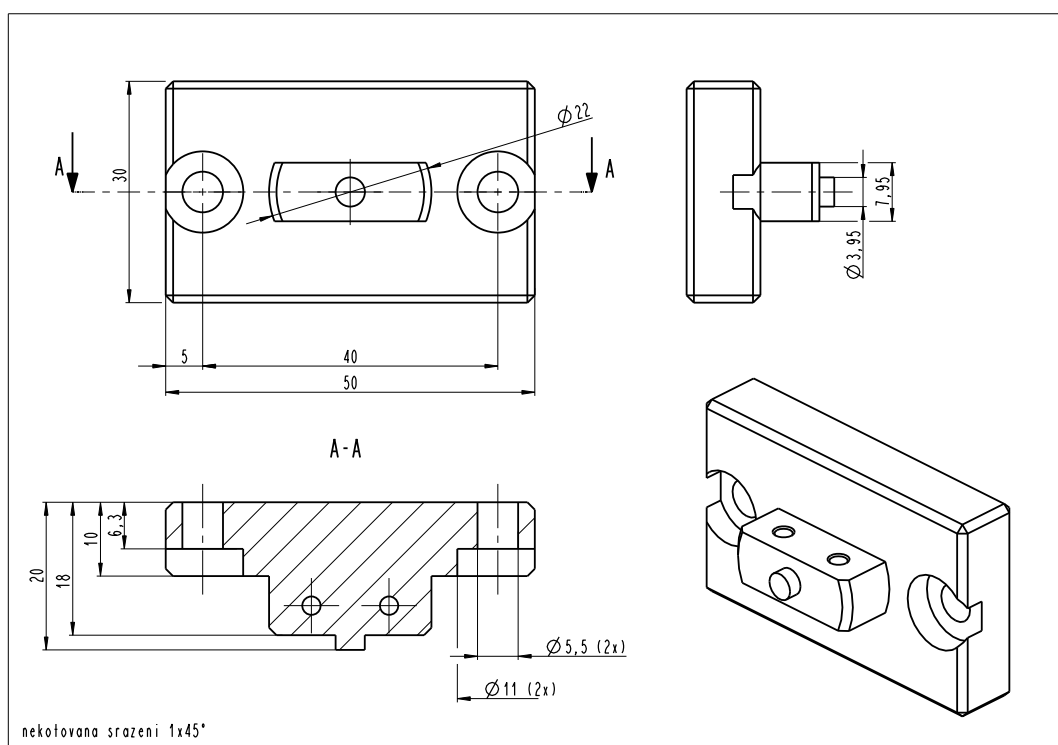
- [23] ABB: *3HAC046307-003_rev06* [online]. [cit. 2018-05-07]. Dostupné z: https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC046307-003_rev06.pdf
- [24] ABB: *Product specification: IRB 1200* [online]. [cit. 2018-05-18]. Dostupné z: <https://library.e.abb.com/public/683ae78194674b809d0362308ebcd5f3/3HAC046982%20PS%20IRB%201200-en.pdf>
- [25] ABB: *Operating manual: IRC5 with FlexPendant* [online]. [cit. 2018-05-18]. Dostupné z: <https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC050941%200M%20IRC5%20with%20FlexPendant%20RW%206-en.pdf>
- [26] ABB: *3HAC049406-003_rev06* [online]. [cit. 2018-05-18]. Dostupné z: https://abb.sluzba.cz/Pages/Public/IRC5UserDocumentationRW6/en/3HAC049406-003_rev06.pdf
- [27] ROS/Introduction [online]. c2018, [cit. 2019-05-21]. Dostupné z: <http://wiki.ros.org/ROS/Introduction>
- [28] Anis, K.: *Robot Operating System (ROS)*. Studies in Computational Intelligence, Cham: Springer International Publishing, 2016, ISBN 978-3-319-26052-5, doi:10.1007/978-3-319-26054-9.
- [29] Catkin [online]. c2018, [cit. 2019-05-04]. Dostupné z: <http://wiki.ros.org/catkin>
- [30] ROS/Concepts [online]. c2018, [cit. 2019-05-21]. Dostupné z: <http://wiki.ros.org/ROS/Concepts>
- [31] Metapackages [online]. c2018, [cit. 2019-05-04]. Dostupné z: <http://wiki.ros.org/Metapackages>
- [32] Industrial [online]. c2018, [cit. 2019-05-04]. Dostupné z: <http://wiki.ros.org/Industrial>
- [33] abb [online]. c2018, [cit. 2019-05-21]. Dostupné z: <http://wiki.ros.org/abb>
- [34] Levi Armstrong a kolektiv: ABB [online]. <https://github.com/ros-industrial/abb>, c2019, [cit. 2019-05-21].
- [35] Levi Armstrong a kolektiv: ABB Experimental [online]. https://github.com/ros-industrial/abb_experimental, c2019, [cit. 2019-05-21].
- [36] Dave Coleman a kolektiv: MoveIt! Tutorials [online]. [cit. 2019-05-22]. Dostupné z: https://ros-planning.github.io/moveit_tutorials/#
- [37] rviz [online]. c2018, [cit. 2019-05-21]. Dostupné z: <http://wiki.ros.org/rviz>

Příloha B

Technické výkresy



Obrázek B.1: Příruha první díl



Obrázek B.2: Příruba druhý díl



Příloha C

Seznam použitých programů

- Schéma zapojení 3.4 a obrázek síťového propojení počítače s řídicí jednotkou IRC5 3.5 byly vytvořeny pomocí online programu `draw.io` [49].
- Veškeré námi vytvořené 3D modely byly vytvořeny pomocí programu `openScad` [50].

Příloha D

Obsah přiloženého CD

src.....	zdrojové kódy implementace rozplňovacího procesu
_ abb	
_ abb_experimental	
_ industrial_core	
src_draw_io.....	zdrojové soubory vytvořené programem draw.io
bp_tex.....	zdrojové soubory této práce ve formátu L ^A T _E X
3d_modely.....	zdrojové soubory vytvořených modelů
PDF	pdf soubory příruby a schéma zapojení
rozplnovani_radiofarmak_robotem.pdf	bakalářská práce ve formátu pdf
rosgraph_all.png	obrázek propojení uzlů v systému ROS