



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

PLÁNOVÁNÍ CESTY ROBOTU PŘI KOLABORACI S ČLOVĚKEM

ROBOT PATH PLANNING FOR HUMAN COLLABORATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vojtěch Válek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Filip Radil

BRNO 2025

Zadání bakalářské práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Vojtěch Válek**
Studijní program: Mechatronika
Studijní obor: bez specializace
Vedoucí práce: **Ing. Filip Radil**
Akademický rok: 2024/25

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Plánování cesty robotu při kolaboraci s člověkem

Stručná charakteristika problematiky úkolu:

Kolaborativní roboty jsou na trhu dostupné již řadu let a počet jejich implementací neustále roste. Jejich schopnost pracovat v bezprostřední blízkosti člověka přináší do průmyslových provozů zcela nové možnosti, ale zároveň také nové výzvy a potenciální komplikace. Tato práce se zaměří na plánování pohybu průmyslového robota na kolaborativním pracovišti.

Cíle bakalářské práce:

- Proved'te řešerši z oblasti algoritmů pro plánování cesty
- Proved'te řešerši z oblasti bezpečnosti kolaborativních robotů
- Implementujte vhodný algoritmus na robot UR5e
- Ověřte algoritmus na laboratorní úloze
- Navrhněte bezpečnostní opatření pro potenciální pracoviště

Seznam doporučené literatury:

Robot Operating System (ROS). 707. Cham: Springer International Publishing. ISBN 9783319549262. ISSN 1860-949X. Dostupné z: doi:10.1007/978-3-319-54927-9

CORKE, Peter. Robotics, Vision and Control. 118. Cham: Springer International Publishing, 2017. ISBN 9783319544120.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2024/25

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jiří Hlinka, Ph.D.
děkan fakulty

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací systému pro bezpečné plánování cesty kolaborativního robota UR5e v prostředí sdíleném s člověkem. Na základě rešerše v oblasti kolaborativní robotiky a metod plánování pohybu byl zvolen algoritmus RRT-Connect, který byl implementován v kloubovém prostoru robota. Pracovní prostředí je modelováno pomocí diskrétní mřížky, která slouží k vyhodnocování kolizí. Do tohoto modelu prostředí je integrován model člověka na základě informací získaných prostřednictvím hloubkové kamery. Funkčnost systému byla ověřena na laboratorní úloze typu pick and place, ve které robot dynamicky reaguje na přítomnost člověka a přizpůsobuje svou cestu. Výsledky testování potvrzují, že systém umožňuje bezpečný a plynulý provoz a může sloužit jako základ pro další vývoj i nasazení v reálném provozu.

Klíčová slova

Kolaborativní robot, plánování cesty, RRT-Connect, UR5e, hloubková kamera

Abstract

This bachelor thesis deals with the design and implementation of a system for safe path planning of a collaborative robot UR5e in a workspace shared with a human. Based on a research in collaborative robotics and motion planning methods, the RRT-Connect algorithm was chosen and implemented in the robot's joint space. The working environment is modeled using a discrete grid to evaluate collisions. A human model is integrated into this environment representation based on the information obtained through the depth camera. The functionality of the system was verified on a pick and place laboratory task in which the robot dynamically reacts to the presence of a human and adapts its path. The test results confirm that the system enables safe and smooth operation and can serve as a foundation for further development and real-world deployment.

Key words

Collaborative robot, path planning, RRT-Connect, UR5e, depth camera

Bibliografická citace

VÁLEK, Vojtěch. *Plánování cesty robotu při kolaboraci s člověkem*. Online, bakalářská práce. Filip RADIL (vedoucí práce). Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2025. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/169477>. [cit. 2025-05-20].

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „Plánování cesty robotu při kolaboraci s člověkem“ vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu, který tvoří přílohu této práce.

23. května 2025

.....
Vojtěch Válek

Poděkování

Tímto děkuji panu Ing. Filipu Radilovi za cenné připomínky a rady při vypracování této bakalářské práce.

Obsah:

1	Úvod	9
2	Rešerše.....	10
2.1	Bezpečnost kolaborativních robotů	10
2.1.1	Bezpečnostní normy a pravidla	10
2.1.2	Ochranné prvky a technologie	14
2.2	Algoritmy pro plánování cesty	17
2.2.1	Plánování cesty v diskrétním prostoru	17
2.2.2	Vzorkovací metody.....	17
2.2.3	Reaktivní a hybridní přístupy	21
2.2.4	Optimalizační metody.....	23
2.2.5	Metody využívající strojové učení	23
3	Postup a výsledky řešení	24
3.1	Algoritmus.....	24
3.1.1	Pracovní prostředí.....	24
3.1.2	Zpracování překážek.....	25
3.1.3	Implementace algoritmu	27
3.2	Laboratorní úloha	29
3.2.1	CAD model pracoviště	29
3.2.2	Hloubková kamera a zpracování dat	30
3.2.3	Umístění kamery	33
3.2.4	Integrace hardwaru a implementace řídicí logiky	34
3.2.5	Testy a hodnocení výsledků	39
3.3	Návrh bezpečnostních prvků	44
3.3.1	Tlačítka nouzového zastavení (E-STOP)	44
3.3.2	Detekce přítomnosti člověka	45
3.3.3	Vizuální označení kolaborativní zóny	45
4	Závěr.....	46
	Seznam použitých zdrojů.....	47
	Seznam obrázků.....	51

1 Úvod

V současné době čelí průmyslová výroba stále vyšším nárokům na flexibilitu, efektivitu a bezpečnost. Výrobní procesy se musí přizpůsobovat rychle se měnícím požadavkům trhu, zkracujícím se výrobním cyklům i potřebě častější spolupráce mezi člověkem a strojem. Tato situace přispívá k rostoucímu zájmu o nasazování kolaborativních robotů, kteří na rozdíl od klasických průmyslových robotů neoperují v uzavřených zónách oddělených fyzickými bariérami, ale vykonávají své úlohy ve společném pracovním prostoru s člověkem.

Takové uspořádání přináší mnoho výhod, včetně úspory místa, rychlého nasazení do provozu a větší flexibility při rozdělení pracovních úkolů. Zároveň však klade vysoké nároky na bezpečnostní aspekty celého systému. Odpovědnost za bezpečný průběh spolupráce se v tomto případě přesouvá z fyzické infrastruktury přímo na řídicí logiku robota. Schopnost reagovat na okolí a přizpůsobit pohyb aktuální situaci se tak stává klíčovým předpokladem bezpečného a plynulého provozu.

Tato práce se zaměřuje na návrh systému, který umožňuje kolaborativnímu robotu bezpečně plánovat a upravovat svou cestu v prostředí sdíleném s člověkem. Cílem je, aby robot dokázal v reálném čase vyhodnocovat situaci v pracovním prostoru a přizpůsobit svou činnost tak, aby se vyhnul rizikovým interakcím s člověkem. Důraz je kladen nejen na samotné zajištění bezpečnosti, ale také na zachování plynulého provozu a minimalizaci zbytečných zastavení tam, kde je možné pokračovat v práci bez ohrožení člověka.

V teoretické části práce je představena problematika bezpečnosti v kolaborativní robotice, včetně přehledu norem a přístupů k ochraně člověka. Následuje popis vybraných metod plánování pohybu v prostředí s překážkami a detailní rozbor navrženého řešení. Praktická část se věnuje implementaci systému využívajícího hloubkovou kameru pro detekci osoby a výpočtu cesty v kloubovém prostoru tak, aby robot bezpečně vykonával zadanou úlohu. Na závěr jsou výsledky testování i návrh bezpečnostních prvků zhodnoceny pro možné nasazení systému do reálného provozu.

2 Rešerše

Rešeršní část práce představuje výchozí krok pro návrh systému, který umožní bezpečnou a efektivní spolupráci člověka s robotem. Zaměřuje se na dvě klíčové oblasti: bezpečnost kolaborativních robotů a plánování cesty. Obě témata spolu úzce souvisejí, protože způsob pohybu robota musí odpovídat jeho technickým možnostem i bezpečnostním požadavkům.

2.1 Bezpečnost kolaborativních robotů

Při práci robota v těsné blízkosti člověka je zásadní dodržovat jasně definovaná bezpečnostní pravidla. Tato pravidla stanovují mezinárodní standardizované normy ISO 10218-1 a ISO 10218-2, které určují požadavky na konstrukci robotů a bezpečnou integraci robotických systémů. Jejich obsah dále rozšiřuje technická specifikace ISO/TS 15066, která upřesňuje požadavky na bezpečný provoz robotů ve sdíleném pracovním prostoru s člověkem. [1]

Tato kapitola se proto zaměřuje na přehled uvedených norem a souvisejících bezpečnostních pravidel, stejně jako na ochranné prvky a technologie, které zajišťují bezpečný provoz kolaborativních robotických systémů.

2.1.1 Bezpečnostní normy a pravidla

ISO 10218-1 – Bezpečnostní požadavky na průmyslové roboty

Norma stanovuje základní bezpečnostní požadavky pro konstrukci a provoz průmyslových robotů a je určena především jejich výrobcům. Tyto požadavky platí i pro kolaborativní roboty. Vybrané klíčové požadavky jsou shrnuty níže. [2]

Konstrukce robota musí zajistit, že ani v případě výpadku napájení, ztráty energie v pohonech nebo poruchy součástí nedojde k nekontrolovanému pohybu. Po ukončení činnosti musí být možné veškerou zbytkovou energii bezpečně odpojit nebo zablokovat její další využití. V praxi může být robot vybaven například brzdami, které při ztrátě napájení automaticky zablokují kloubové pohony, aby se zabránilo samovolnému pohybu. [2] [3]

Ovládací prvky používané ke spuštění pohybu nebo provozu robota musí být navrženy tak, aby se zabránilo neúmyslné aktivaci (např. pomocí krytu přes tlačítko nebo umístění uzamykatelného voliče). V případě použití více ovládacích zařízení musí být zajištěno, že robot nemůže být ovládán z více míst současně, což by mohlo vést k nebezpečné situaci. Stav zařízení by měl být zároveň jasně indikován, například světelným signálem aktivní režim nebo přítomnost napájení. [2]

Robot musí být vybaven funkcí ochranného zastavení a nezávislou funkcí nouzového zastavení. **Nouzové zastavení** slouží k okamžitému ukončení všech pohybů robota a k odpojení všech zdrojů energie. Aktivuje se ručně (např. tlačítkem) a má přednost před všemi ostatními řídicími příkazy. Po aktivaci zůstává systém v tomto stavu, dokud není funkce ručně resetována (uvolněním tlačítka). Takto je zajištěno, že opětovné spuštění robota proběhne pouze vědomě a bezpečně. Ovládací prvky určené k zahájení pohybu robota musí obsahovat tlačítko aktivující tuto funkci. [2]

Ochranné zastavení se využívá v méně kritických situacích a po aktivaci nezpůsobí nouzový stav systému. Může být aktivováno automaticky nebo ručně a je určeno zejména k reakci na podněty z vnějších bezpečnostních zařízení (např. světelná závora detekující osobu vstupující do pracovního prostoru). Po aktivaci robot okamžitě zastaví svůj pohyb, avšak zůstává napájen a připraven k opětovnému spuštění. Na rozdíl od nouzového zastavení není nutné zařízení resetovat, což umožňuje rychlejší návrat do provozu. Funkce zastavení musí být navržena tak, aby byla nezávislá na běžném řízení a nebylo možné ji obejít nebo deaktivovat. [2]

Rychlost pohybu koncového efektoru (TCP) musí být volitelně nastavitelná a řízená. V režimu snížené rychlosti nesmí překročit hodnotu 250 mm/s, přičemž je možné nastavit

i nižší limit podle konkrétních požadavků. Pokud je nastavený limit překročen, musí se automaticky aktivovat ochranné zastavení. [2]

Provozní režimy musí být volitelně přepínatelné (např. přepínačem s klíčem), přičemž přepnutí je možné pouze tehdy, když je robot zcela zastaven. V automatickém režimu robot pracuje podle předem naprogramované sekvence a během provozu musí být aktivní všechny požadované bezpečnostní funkce. Norma dále definuje dva ruční režimy (se sníženou a se zvýšenou rychlostí), které se využívají pro pomalé nastavování, učení, programování nebo údržbu. V těchto režimech lze robotem pohybovat pouze při nepřetržitém stisku ovládacího tlačítka umístěného na ovládacím panelu. Uvolnění tlačítka vede k okamžitému zastavení, což zajišťuje přímou kontrolu operátora nad pohybem a minimalizuje riziko zranění. [2]

Norma specifikuje **režim provozní spolupráce**, kdy člověk sdílí pracovní prostor s robotem. Aktivace tohoto režimu musí být jasně vizuálně označena. Pokud režim zahrnuje ruční vedení robota, musí být jeho maximální rychlost stanovena na základě posouzení rizik konkrétní situace. Pro rychlost i polohu platí, že při překročení stanovených limitů se musí aktivovat ochranné zastavení. Současně je nutné omezit výkon a sílu robota, aby se v případě kontaktu zabránilo zranění. Pro podrobnější návrh norma výslovně odkazuje na normu ISO 10218-2 a technickou specifikaci ISO/TS 15066. [2]

Všechny pohyby a výstupy robota musí být plně řízeny, aby nemohlo dojít k nežádoucím pohybům. Současně nesmí být překročen předem definovaný pracovní prostor. Pokud robot překročí předem definovaný pracovní prostor, jeho pohyb musí být automaticky zastaven. [2]

ISO 10218-2 – Bezpečnostní požadavky na integraci robotických systémů

Druhá část normy ISO 10218 se zaměřuje na bezpečnostní požadavky spojené s instalací, integrací a provozem kompletních robotických systémů. Je určena především integrátorům a koncovým uživatelům. Níže jsou shrnuty vybrané klíčové oblasti, které jsou zásadní pro bezpečné uvedení robotů do provozu v konkrétních pracovních prostředích. [4]

Robotická buňka nebo linka musí být navržena a integrována tak, aby minimalizovala rizika osob v jejím okolí. Přístup do nebezpečných oblastí musí být omezen vhodnými ochrannými zařízeními, která brání nežádoucímu kontaktu. Provozní ovladače a zařízení, která vyžadují přístup během automatického provozu, musí být umístěna mimo zabezpečený prostor. [4]

V této části normy je zdůrazněno, že funkce **nouzového a ochranného zastavení** musí být integrovány do celého robotického systému včetně přidružených aplikací. Nouzové zastavení musí být dostupné z více míst v chráněné zóně, aby byla zajištěna rychlejší reakce. Po aktivaci musí dojít k vypnutí všech souvisejících zařízení a procesů, čímž se zamezí dalším potenciálně nebezpečným situacím. Ochranné zastavení bývá aktivováno prostřednictvím externích bezpečnostních prvků, jako jsou světelné závory nebo laserové skenery, které efektivně pokrývají přechod mezi chráněnou a nebezpečnou zónou. [4]

Koncový efektor musí být navržen tak, aby nedocházelo k náhodnému uvolnění nástrojů, a zároveň musí odolávat předpokládanému zatížení během provozu. Odnímatelné nástroje musí být bezpečně uchyceny a jejich uvolnění možné pouze za bezpečných podmínek. Důležitá je také pravidelná kontrola jejich stavu. U rizikových aplikací, jako je svařování či manipulace s nebezpečnými materiály, je nezbytné zohlednit specifická provozní rizika. [4]

Pro bezpečný provoz robotického systému je nutné jednoznačně **vymezit pracovní prostor** robota tak, aby se minimalizovala rizika. Pohyb robota musí být omezen mechanickými prostředky (např. dorazy, zarážky) nebo softwarově (omezení rozsahu os). K ochraně prostoru se využívají fyzické nebo optoelektronické prvky. Dynamické omezení navíc umožňuje

automaticky upravovat mezní prostor robota během provozu (např. pomocí výsuvných zářezů nebo světelných clon). Tato zařízení musí zajistit bezpečné zastavení robota i za běžného zatížení. [4]

Součástí bezpečného uspořádání pracoviště je rovněž ochrana jeho okolí, kterou lze zajistit fyzickými kryty nebo optoelektronickými prvky. Zároveň je nezbytné vytvořit bezpečný a snadno přístupný prostor pro obsluhu robota a provádění údržby. [4]

V této části jsou požadavky na **provozní spolupráci** dále rozvedeny, a to zejména z hlediska návrhu pracoviště, charakteru interakce a podmínek, za kterých může člověk bezpečně sdílet pracovní prostor s robotem. Tyto aspekty jsou nedílnou součástí celkového hodnocení rizik a musí být zohledněny při návrhu konkrétní aplikace. [4]

V rámci provozní spolupráce jsou definovány tři základní režimy interakce mezi člověkem a robotem, které se liší mírou sdílení prostoru a typem vykonávané činnosti:

- **Ruční vedení:** Tento režim byl již zmíněn v předchozí části. Operátor přímo manipuluje s robotem, přičemž musí být zajištěna kontrola pohybu a omezení rychlosti.
- **Monitorování rychlosti a separace:** V tomto režimu se člověk může pohybovat v pracovním prostoru robota, avšak bez přímého kontaktu. Vzdálenost mezi člověkem a robotem je neustále monitorována a robot na její změnu reaguje úpravou rychlosti, případně úplným zastavením.
- **Omezení výkonu a síly:** Tento režim připouští fyzický kontakt mezi člověkem a robotem, avšak nesmí při něm dojít ke zranění. Parametry, jako síla, tlak a rychlost, musí být omezeny na základě důkladného posouzení rizik konkrétní aplikace.

Podrobnější popis jednotlivých režimů spolupráce je obsažen v technické specifikaci ISO/TS 15066, na kterou se zde norma odkazuje. [4]

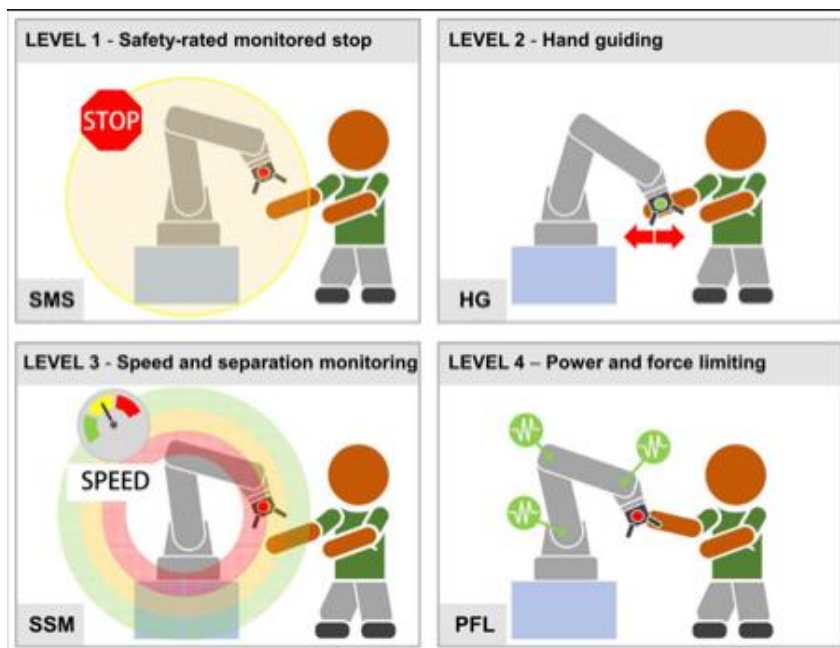
ISO/TS 15066 – Technická specifikace pro kolaborativní roboty

Tato technická specifikace se zaměřuje na bezpečný provoz kolaborativních robotických systémů, ve kterých dochází k přímé spolupráci člověka a robota ve sdíleném pracovním prostoru. Je určena především výrobcům, integrátorům a uživatelům kolaborativních aplikací. V této části jsou dále rozebrány vybrané klíčové požadavky, které mají zásadní význam pro návrh bezpečného kolaborativního pracoviště. [5]

Kolaborativní pracovní prostor představuje vymezenou oblast, ve které může probíhat současná činnost robota a člověka. V tomto režimu zůstává robot plně aktivní i během přítomnosti obsluhy, což klade zvýšené nároky na bezpečnost celého systému. Již při návrhu je proto nezbytné zohlednit možnost fyzického kontaktu a zajistit, aby pohyb robota probíhal výhradně za předem definovaných a kontrolovaných podmínek. [5]

Návrh kolaborativní aplikace musí vycházet z komplexního hodnocení prostředí, ve kterém bude systém provozován, a z charakteru interakce mezi člověkem a robotem. Nedílnou součástí tohoto procesu je analýza rizik, která mimo jiné zohledňuje přístupnost pracovního prostoru, ergonomii obsluhy, rozmístění zařízení a pravděpodobnost fyzického kontaktu. Zvláštní pozornost je třeba věnovat uspořádání okolního prostředí, aby se předešlo situacím, kdy by mohlo dojít k přiskřípnutí nebo sevření částí těla mezi robotem a ostatními konstrukčními prvky. Tato rizika lze snížit vhodným uspořádáním pracovního prostoru, dostatečnými odstupy nebo případně softwarovým omezením rozsahu pohybu robota. Cílem je navrhnout uspořádání, které umožní bezpečný a spolehlivý provoz ve sdíleném prostoru a současně podpoří efektivní spolupráci obou stran. [5]

Pro bezpečný provoz kolaborativních robotických systémů byly definovány čtyři základní režimy spolupráce člověka s robotem. Každý z nich stanovuje odlišné podmínky a omezení, za kterých mohou spolupracovat ve společném pracovním prostoru. Tyto režimy doplňují obecná bezpečnostní pravidla a umožňují přizpůsobit chování robota konkrétním požadavkům dané aplikace. Následující část je popisuje podrobněji. [5]



Obr. 2.1: Čtyři režimy kolaborativní spolupráce. [6]

- **Bezpečnostní monitorované zastavení (SMS)**

Robot se automaticky zastaví, jakmile do pracovního prostoru vstoupí člověk, a po celou dobu jeho přítomnosti zůstává nehybný. Napájení a ovládání zůstávají po tuto dobu aktivní, po opuštění prostoru proto robot může pokračovat v pohybu bez zásahu obsluhy.

Tento způsob se využívá zejména tam, kde je nutný krátký zásah do prostoru robota (např. výměna dílu nebo vizuální kontrola). Přítomnost osoby je detekována bezpečnostními snímači, jako jsou světelné závory nebo laserové skenery. Výhodou je snadné nasazení, zároveň ale platí, že robot a člověk nikdy nepracují současně. [5]

- **Ruční vedení (HG)**

Pohyb robota je přímo řízen člověkem, a pohybuje se tedy pouze na základě záměrných pohybů obsluhy. Ruční vedení se využívá především při výuce dráhy nebo nastavování konkrétních pozic. [5]

- **Monitorování rychlosti a vzdálenosti (SSM)**

Jedná se o režim, který umožňuje současný pohyb člověka a robota ve stejném prostoru, přičemž robot reaguje na změnu jejich vzájemné vzdálenosti. Technická specifikace rozšiřuje tento princip o přesně definovaný výpočet bezpečné vzdálenosti, která zohledňuje rychlost, reakční dobu systému a přesnost senzorů. Je vhodný zejména pro aplikace, kde se lidé pohybují v blízkosti robota často a nepředvídatelně, protože umožňuje plynulý provoz bez nutnosti opakovaného zastavování. [5]

- **Omezení síly a výkonu (PFL)**

Jde o jediný režim, který připouští přímý kontakt člověka s robotem za předpokladu, že nedojde k překročení bezpečných silových limitů. Technická specifikace tento režim dále rozvádí s ohledem na to, jak lze snížit rizika, a to buď pasivně například zaoblením tvaru robota, použitím měkkých materiálů nebo omezením jeho hmotnosti, nebo aktivně prostřednictvím senzorů síly, řízeného zpomalení nebo softwarového omezení pracovního prostoru.

Současně rozlišuje dva typy fyzického kontaktu: kvazistatický (např. skřípnutí těla mezi robotem a překážkou) a přechodový (krátkodobý náraz). Pro oba případy jsou stanoveny maximální přípustné síly a tlaky na jednotlivé části lidského těla. [5]

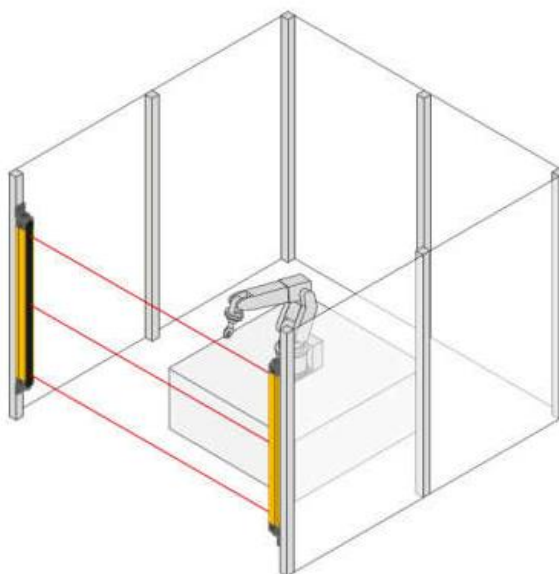
2.1.2 Ochranné prvky a technologie

Při návrhu a provozu robotických systémů se bezpečnosti dosahuje prostřednictvím různých ochranných prvků, které snižují riziko kontaktu s pohyblivými částmi nebo minimalizují následky případné kolize. Některá z těchto řešení již byla zmíněna v předchozím textu. Tato část se jim však věnuje podrobněji a doplňuje je o další příklady prvků běžně využívaných v praxi.

Fyzické ochranné prostředky se používají k zabránění vstupu osob do nebezpečné zóny nebo k omezení pohybu robota. Patří mezi základní a pasivní bezpečnostní opatření. Nejčastěji se využívají pevné kryty a oplocení, které bývají doplněny o bezpečnostní spínače, jež přeruší provoz robota při otevření přístupových dveří. Pro omezení rozsahu pohybu robota se uplatňují mechanické dorazy a zarážky, které zabráňují kolizi s okolím nebo průniku do zakázaných oblastí. [2] [3] [4]

Snímací a detekční systémy slouží k tomu, aby systém dokázal včas zareagovat na přítomnost osoby. Používají se zejména v případech, kdy nelze pracovní prostor fyzicky oddělit a je potřeba zajistit bezpečnost pomocí aktivního sledování okolí.

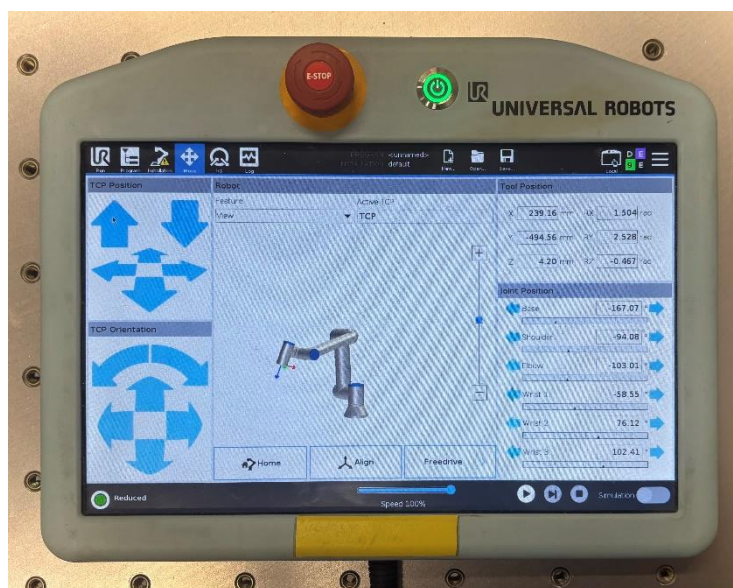
Mezi základní prvky patří světelné závory a mříže, které zaznamenají vstup osoby do vymezené zóny, a laserové skenery schopné rozlišovat více úrovní přiblížení osoby. Příklad umístění světelných závor u vstupu do robotického pracoviště je znázorněn na obrázku 2.2. Pro přesnější lokalizaci pohybu se využívají hloubkové a 3D kamery nebo LIDAR senzory, které umožňují určit konkrétní polohu osoby v prostoru. V některých případech se používají také kapacitní nebo magnetické senzory. Ty jsou vhodné tam, kde nelze spolehlivě použít optické systémy. [7] [39] [40]



Obr. 2.2: Vizualizace světelných závor u vstupu do prostoru s robotem. [39]

Řídící a ovládací rozhraní umožňují operátorovi přímo ovlivňovat chování robota během provozu i při jeho nastavování. Základním bezpečnostním prvkem je tlačítko nouzového zastavení (E-STOP), které musí být snadno dosažitelné, vhodně označené (tlačítko červené na žlutém podkladu) a při aktivaci musí okamžitě přerušit všechny pohyby robota. [2] [4] [34]

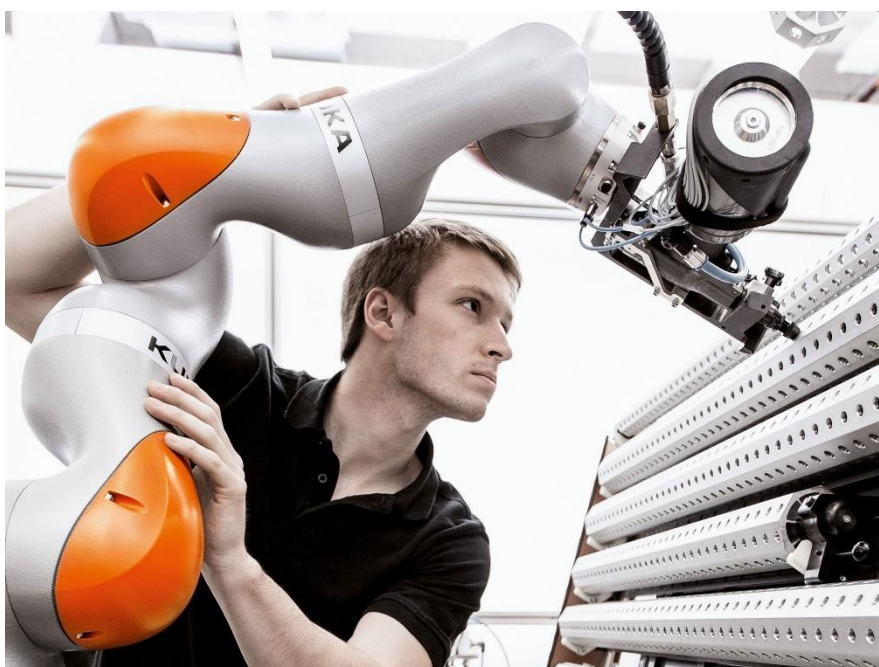
Standardní součástí výbavy robotických systémů je ruční ovladač (teach pendant), pomocí kterého lze robota ovládat, programovat a měnit provozní režimy. Důraz je kladen na přehledné rozhraní a snadnou dostupnost klíčových funkcí, včetně vestavěného tlačítka E-STOP pro nouzové zastavení. Konkrétní podoba ručního ovladače je zachycena na obrázku 2.3. Nedílnou součástí těchto rozhraní bývají také softwarová bezpečnostní omezení, která umožňují nastavit maximální rychlost, zrychlení a pracovní rozsah robota podle požadavků konkrétní aplikace. [2] [4]



Obr. 2.3: Fotografie ručního ovladače robota UR5e.

Senzory síly a momentu slouží k detekci fyzického kontaktu mezi robotem a okolím. Bývají přímo integrovány v konstrukci robota, nejčastěji v kloubech nebo koncovém efektoru, a umožňují rozpoznat, zda došlo ke kolizi nebo dotyku. Použití těchto senzorů je zásadní pro režim omezení síly a, jak je definováno v technické specifikaci. Díky moderním technologiím umožňují tyto senzory přesnou a rychlou detekci kontaktu, na kterou může systém okamžitě reagovat. Tím se výrazně snižuje riziko zranění a minimalizují se následky případného kontaktu. [5] [41]

Vedle samotné detekce kontaktu má na bezpečnost výrazný vliv i **konstrukční provedení** robota. Zaoblené tvary, měkké povrchy a omezená hmotnost pomáhají snížit následky případného kontaktu robota s překážkou. Tyto konstrukční prvky jsou důležité zejména při přímé spolupráci člověka s robotem a tvoří doplňkovou ochranu k senzorickým řešením. Příkladem robota navrženého s ohledem na tyto konstrukční požadavky je KUKA LBR iiwa, jehož zaoblené tvary a hladké povrchy minimalizují riziko poranění při kontaktu s člověkem. Zmíněný robot je zobrazen na obrázku 2.4. [5] [41] [42] [43]



Obr. 2.4: Robot KUKA LBR iiwa s hladkým, zaobleným designem určeným pro bezpečnou spolupráci s člověkem. [43]

Nositelná bezpečnostní zařízení mohou být založena na aktivních prvcích, které komunikují s okolními roboty. Příkladem je bezpečnostní vesta používaná ve skladech Amazonu, která pomocí signálu informuje roboty o přítomnosti člověka, což vede k adekvátní úpravě jejich pohybu. Tento přístup zvyšuje bezpečnost zejména v prostředích, kde by jinak bylo obtížné pracovníka spolehlivě detekovat. Technologie byla úspěšně nasazena v reálném provozu a ukazuje potenciál pro širší využití v oblasti kolaborativní automatizace. [44] [45]

Volba konkrétních ochranných prvků závisí na typu aplikace, provozním režimu, míře interakce s člověkem a prostorovém uspořádání pracoviště. V praxi se jednotlivé prvky často kombinují s cílem zajistit maximální úroveň bezpečnosti a efektivní provoz robotického systému. Fyzické zábrany, snímací systémy, softwarová omezení i konstrukční řešení se vzájemně doplňují, čímž se zvyšuje celková spolehlivost a flexibilita systému. Tento přístup založený na principu **redundance** umožňuje lépe reagovat na různé provozní situace a přispívá k bezpečné spolupráci člověka s robotem. [7] [40]

2.2 Algoritmy pro plánování cesty

Plánování cesty slouží k nalezení bezpečné cesty robota mezi výchozí a cílovou konfigurací v prostředí, kde se vyskytují překážky. Každá konfigurace vyjadřuje konkrétní stav robota daný jeho polohou a orientací v prostoru. Pohyb robota lze přitom popsat buď jako geometrickou cestu, tedy sled konfigurací, nebo jako trajektorii, která k těmto konfiguracím přiřazuje i časové informace, jako je rychlost a zrychlení. [8]

Tato kapitola si klade za cíl poskytnout přehled různých přístupů k plánování cesty a vytvořit tak výchozí základ pro výběr vhodného algoritmu pro řešení dané úlohy. Jsou zde obecně popsány různé skupiny metod, přičemž větší prostor je věnován těm, které se vzhledem k povaze řešeného problému jeví jako potenciálně nejvhodnější.

2.2.1 Plánování cesty v diskrétním prostoru

Plánování cesty v diskrétním prostoru je založeno na systematickém prohledávání předem definované mřížky nebo grafu, kde každý bod (buňka) reprezentuje možný stav prostředí a hrany popisují možné přechody mezi těmito stavy. V rámci takto vytvořeného grafu se k nalezení cesty mezi výchozím a cílovým stavem využívají algoritmy grafového prohledávání, které prohledávají prostor podle určitých pravidel a kritérií.

Mezi nejčastěji používané algoritmy pro plánování cesty v diskrétním prostoru patří Dijkstraův algoritmus, který systematicky prohledává graf a nachází nejkratší cestu bez použití heuristiky. Jeho rozšířením je algoritmus A*, který zavádí heuristickou funkci odhadující vzdálenost k cíli a tím urychluje proces hledání. Pro prostředí s proměnlivou strukturou je pak vhodný algoritmus D*, který umožňuje dynamickou aktualizaci nalezené cesty při změnách v okolí.

Díky své robustnosti a jednoduché implementaci se grafově založené algoritmy hojně využívají pro plánování pohybu v autonomním řízení, mobilní robotice a průmyslové automatizaci, kde umožňují efektivní navigaci v předem známých nebo strukturovaných prostředích. [9]

2.2.2 Vzorkovací metody

Vzorkovací metody (sampling-based motion planning) představují skupinu algoritmů pro plánování cesty, které jsou založeny na generování náhodných vzorků v konfiguračním prostoru a hledání spojení mezi nimi. Není nutné znát nebo explicitně modelovat celý prostor, ale stačí pracovat pouze s informacemi z okolí generovaných vzorků, čímž se snižuje výpočetní náročnost.

S rostoucím počtem stupňů volnosti robota roste i dimenze konfiguračního prostoru, což u metod založených na mřížce vede k exponenciálnímu nárůstu výpočetní náročnosti. Vzorkovací metody tento problém obcházejí tím, že prostor definují pouze lokálně v místech, kde jsou náhodně generované vzorky, mezi nimiž je testována dosažitelnost a případné kolize.

Přestože nezaručují nalezení řešení v konečném čase, pravděpodobnost jeho nalezení (pokud existuje) se s rostoucím počtem vzorků asymptoticky blíží jedné. [10]

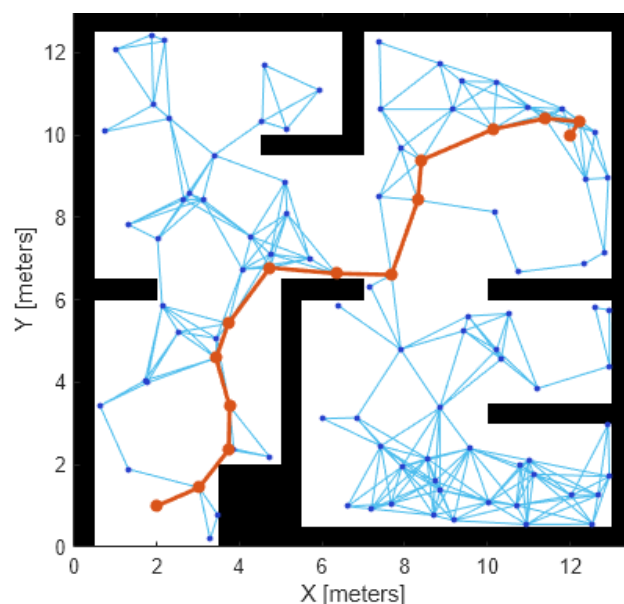
Metoda pravděpodobnostní roadmapy (Probabilistic Roadmap Method, PRM)

Algoritmus pracuje ve dvou fázích. V první fázi vytváří roadmapu, která vzniká náhodným vzorkováním bezkolizních konfigurací a jejich vzájemným propojováním. Ve druhé fázi se do roadmapy přidá startovní a cílová konfigurace, která se na vzniklou roadmapu napojí a cesta mezi nimi je nalezena pomocí grafových algoritmů, jako je například A* algoritmus.

Princip konstrukce roadmapy metodou PRM je ilustrován na obrázku 2.5, kde jsou znázorněny náhodně generované uzly, jejich propojení do grafové struktury a výsledná nalezená cesta v prostředí obsahujícím překážky.

Hlavní výhodou je možnost rychlého vyhledávání cest po vytvoření roadmapy a opakované použití při různých počátečních a cílových konfiguracích. Nevýhodou je nutnost dostatečného počtu vzorků pro pokrytí prostoru a nevhodnost pro dynamicky se měnící prostředí, protože změny v rozmístění překážek vyžadují opětovné vytvoření celé roadmapy.

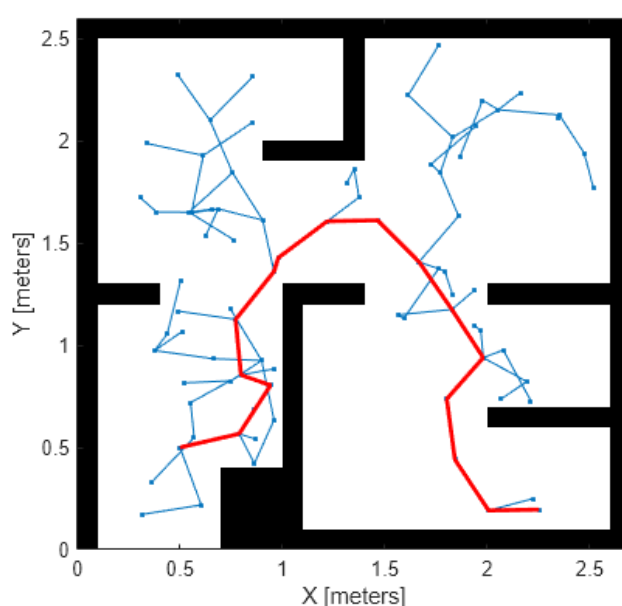
PRM se proto nejlépe uplatňuje v úlohách, kde je prostředí v čase neměnné, což umožňuje opakovaně využívat již vytvořenou roadmapu. Typickým příkladem je plánování pohybu mobilních robotů ve skladech, kde jsou překážky pevně dané a nemění se během provozu. [8] [10] [11]



Obr. 2.5: Roadmapa vytvořená metodou PRM s připojeným startem a cílem a vyznačenou výslednou cestou. [11]

Rapidly-Exploring Random Tree (RRT)

Algoritmus je založen na postupném rozšiřování stromové struktury v konfiguračním prostoru. V každém kroku je vygenerován náhodný bod, který je připojen k nejbližšímu existujícímu uzlu ve stromě. Pokud toto spojení nevede přes překážku, tak je tento bod přidán do stromové struktury. V opačném případě se bod nepřidá a generuje se nový. Tento proces se opakuje, dokud strom nedosáhne oblasti cílové konfigurace. Na obrázku 2.6 je znázorněno rozrůstání stromu (modře) a nalezená cesta od startu do cíle (červeně) v prostředí s překážkami. [8] [10] Hlavní výhodou tohoto algoritmu je jednoduchá implementace a schopnost rychle nalézt cestu. Výsledné cesty však nebývají optimální a mohou obsahovat nadbytečné odchytky, což často vyžaduje jejich dodatečné vyhlazení. Algoritmus nachází uplatnění zejména při plánování pohybu mobilních robotů a robotických manipulátorů. [8] [10]

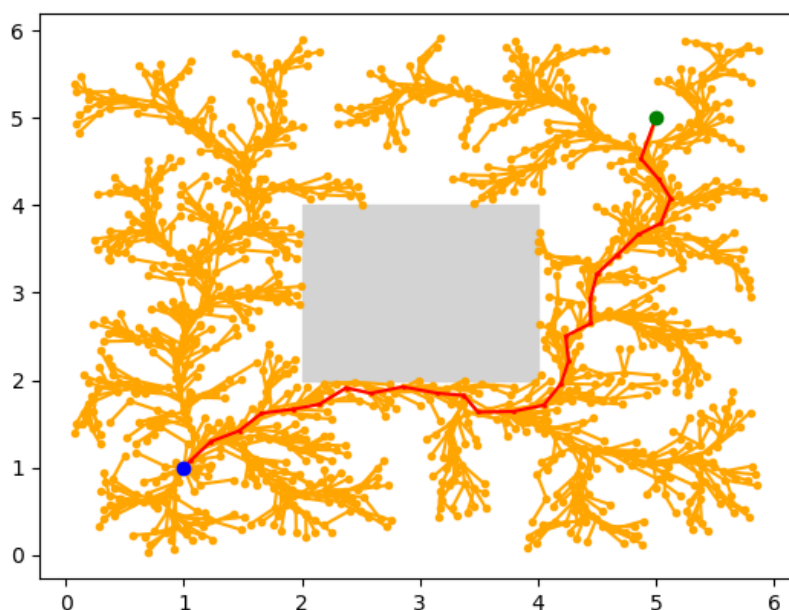


Obr. 2.6: Plánování cesty metodou RRT v prostředí s překážkami. [12]

RRT*

Metoda RRT* představuje vylepšení základního algoritmu RRT, přičemž hlavní změnou je způsob připojování nových bodů do stromové struktury. Nově generované body se nepřipojují k nejbližším uzlům, ale k těm, které umožňují nejkratší (nejlevnější) cestu od startu. Navíc algoritmus přepočítává okolní uzly (rewiring), pokud nové spojení zkracuje jejich cestu k počátečnímu bodu. Průběh rozrůstání stromu (žlutou) a výslednou optimalizovanou cestu (červeně) znázorňuje obrázek 2.7. [13]

Hlavní výhodou algoritmu je dosažení asymptotické optimality – při dostatečném počtu vzorků se nalezená cesta blíží globálnímu minimu. Výsledné cesty jsou tak kratší a efektivnější než u klasického RRT. To je však vykoupeno vyšší výpočetní náročností způsobenou výpočtem nákladů a přepojováním sousedních uzlů. Algoritmus nachází uplatnění zejména v aplikacích, kde je kladen důraz nejen na nalezení cesty, ale také na její kvalitu. Například při plánování pohybu mobilních robotů a autonomních vozidel. [13]

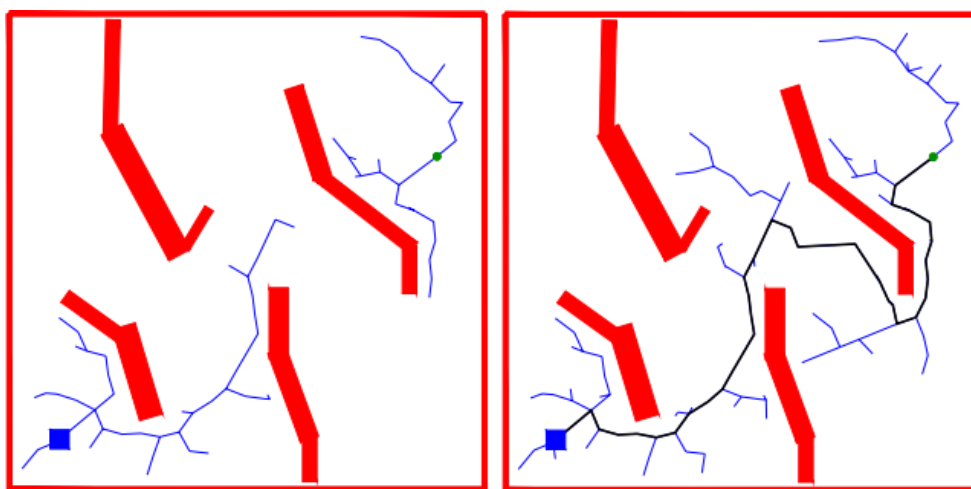


Obr. 2.7: Rozrůstání stromu a výsledná cesta nalezená algoritmem RRT*. [14]

RRT-Connect

RRT-Connect je varianta algoritmu pro plánování cesty, která využívá dva samostatné stromy, kde jeden vyrůstá od startovní a druhý od cílové konfigurace. Každý z nich se střídavě rozšiřuje, co nejdále ve směru ke generovanému bodu, dokud nenarazí na překážku nebo nedojde k propojení obou stromů. Tento princip vede k výrazně rychlejšímu nalezení cesty ve srovnání se základním RRT. Průběh rozrůstání a spojení stromů je znázorněn na obrázku 2.8. [15]

Stejně jako základní varianta RRT neposkytuje ani RRT-Connect optimalizované řešení a rovněž tak je nutné výslednou cestu dodatečně vyhladit. Algoritmus se osvědčuje při plánování pohybu mobilních robotů a robotických manipulátorů v prostředích s překážkami, kde je důležité rychle nalézt realizovatelnou cestu. [15]



Obr. 2.8: Rozrůstání dvou stromů a nalezení cesty metodou RRT-Connect. [15]

Kromě zde popsaných metod existuje celá řada dalších rozšíření algoritmu RRT, která cílí na zlepšení výpočetní efektivity, kvality plánovaných cest nebo přizpůsobení dynamickým změnám v prostředí. Vývoj těchto algoritmů zůstává aktivní oblastí výzkumu, která posouvá možnosti jejich praktického nasazení v různých oblastech robotiky. [8]

2.2.3 Reaktivní a hybridní přístupy

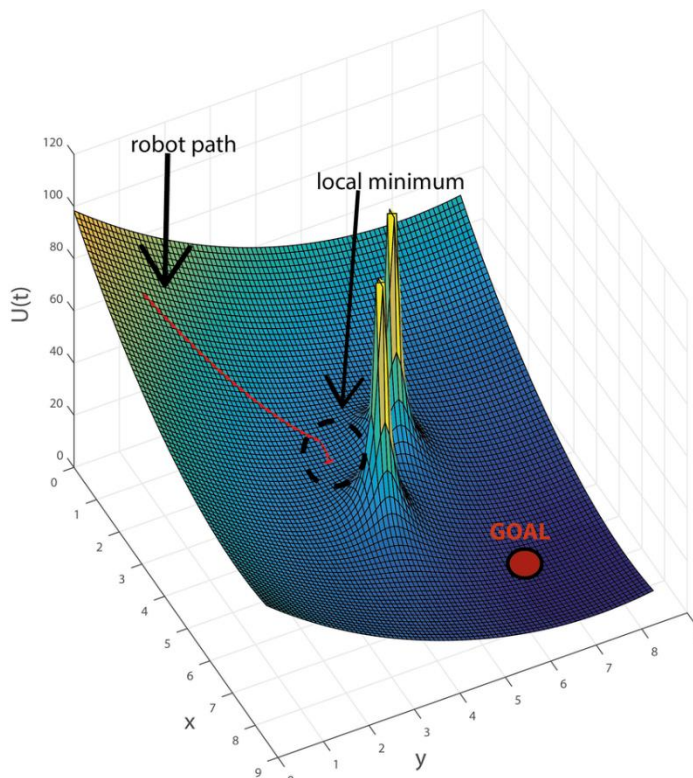
Tyto metody umožňují robotům přizpůsobit pohyb v reálném čase na základě aktuálního stavu prostředí bez nutnosti znalosti celého prostoru. Uplatňují se především v dynamických nebo neznámých prostředích.

Umělá potenciálová pole (Artificial Potential Fields, APF)

Princip metody APF spočívá v tom, že na základě aktuálního stavu jsou definovány dvě složky působících sil: přitažlivá síla směrem k cílovému bodu a odpudivé síly od okolních překážek. Přitažlivá složka táhne systém k požadovanému cíli, zatímco odpudivé složky zabráňují kolizím s překážkami. Výsledná síla v daném místě určuje směr dalšího postupu, kterým se systém posouvá směrem k cílové pozici. Výsledné trajektorie vznikají postupným sledováním směru výsledné síly v potenciálovém poli. Metoda patří mezi reaktivní přístupy, protože směr pohybu je určován pouze na základě aktuální informace o cíli a překážkách bez potřeby znalosti celého prostoru. [16]

Metoda se vyznačuje jednoduchou implementací, nízkou výpočetní náročností a schopností okamžité reakce na změny v prostředí. Výsledkem je plynulá, lokálně určená trajektorie směrem k cíli. Nevýhodou je riziko uvíznutí v lokálních minimech, jak je vidět na obrázku 2.9, což bývá často doprovázeno oscilacemi v blízkosti překážek. S rostoucím počtem

stupňů volnosti robota se navíc zvyšuje složitost potenciálového pole, což zvyšuje náročnost nalezení trajektorie. Využívá se nejčastěji při lokálním plánování trajektorie mobilních robotů v prostředích s menším počtem překážek. [16] [17] [18]



Obr. 2.9: Potenciálové pole s vyznačenou cestou robota, včetně uvíznutí v lokálním minimu. [38]

Elastické pásy (Elastic Bands)

Princip metody elastických pásů spočívá v úpravě již existující globální cesty tak, aby se přizpůsobila překážkám v okolí. Vstupní cesta je reprezentována jako elastický pás, kde jednotlivé body jsou propojeny virtuálními pružinami. Při výskytu překážky v blízkosti cesty dochází k posunutí bodů působením vnějších sil, zatímco vnitřní síly pružin zachovávají spojitost a hladkost průběhu. Výsledkem je deformovaná cesta, která se plynule vyhýbá překážkám a udržuje směr k cíli. [19]

Tato metoda se vyznačuje nízkou výpočetní náročností a schopností přizpůsobit se aktuálním změnám v prostředí. Nevýhodou je závislost na kvalitě původně naplánované cesty a citlivost na nastavení jednotlivých parametrů algoritmu. Využívá se při lokálním plánování pohybu mobilních robotů, přičemž její rozšířená časovaná varianta umožňuje optimalizaci trajektorie s ohledem na dynamiku pohybu. [19] [20]

Vector Field Histogram (VFH)

VFH převádí okolní prostředí na histogram vzdáleností a podle něj vybírá směr s nejnižší hustotou překážek. Tím umožňuje plynulé a rychlé přizpůsobení pohybu aktuálním podmínkám. Existují také rozšířené varianty, například VFH+ zohledňující kinematická omezení robota a VFH* kombinující lokální plánování s globální metodou A^* . [21]

Dynamic Window Approach (DWA)

DWA umožňuje volbu optimální trajektorie na základě aktuální polohy, rychlosti a dynamických omezení robota. Algoritmus generuje množinu možných trajektorií v prostoru rychlostí a hodnotí je podle kritérií jako vzdálenost k cíli, vzdálenost od překážek a preferovaná rychlost. Z těchto možností vybírá takovou, která směřuje k cíli a zároveň se bezpečně vyhýbá překážkám. Díky tomu zajišťuje rychlou a plynulou navigaci v měnícím se prostředí. [22]

Metody VFH a DWA se využívají pro lokální plánování pohybu v mobilní robotice. Díky rychlé reakci na okolní podmínky jsou vhodné pro navigaci v reálném čase. [21] [22]

2.2.4 Optimalizační metody

Optimalizační metody v plánování trajektorie se zaměřují na vylepšení existujících cest tak, aby byly plynulé, krátké a energeticky efektivní. Využívají matematickou optimalizaci s omezeními a často navazují na hrubě nalezené cesty, které dále zpřesňují a přizpůsobují reálnému prostředí. Typickými příklady těchto přístupů jsou algoritmy CHOMP, STOMP a TrajOpt, které hledají trajektorie minimalizující pravděpodobnost kolize při zachování přirozeného průběhu pohybu. Tyto metody se uplatňují například při plánování trajektorií robotických ramen v blízkosti překážek nebo při navigaci mobilních robotů v omezených prostorech. [23]

2.2.5 Metody využívající strojové učení

Metody využívající strojové učení umožňují plánovat trajektorie robotů na základě předchozích zkušeností nebo znalostí o rozložení překážek v prostředí. Uplatňují se zejména tam, kde je klasické plánování obtížné – například v prostředích s vysokou dimenzionalitou, neznámými překážkami nebo tam, kde je potřeba rychlá reakce. Nejčastěji se využívají neuronové sítě, jako je MPNet, metody posilovaného učení typu DDPG či grafové neuronové sítě, které dokážou modelovat složité vztahy v prostředí. Přestože tyto přístupy nabízejí značný potenciál, jejich využití je zatíženo vysokými nároky na výpočetní výkon a kvalitu trénovacích dat. V praxi se proto zatím objevují spíše výjimečně, a to především kvůli požadavkům na spolehlivost a robustnost systémů. [24]

3 Postup a výsledky řešení

Tato kapitola se věnuje řešení kolaborativní úlohy, při které se robot UR5e pohybuje v prostoru sdíleném s člověkem a automaticky se mu vyhýbá. Cílem je zajistit, aby robot dokázal bezpečně vykonávat svou práci i v případě, že se v jeho okolí objeví překážka, zejména člověk. K tomu slouží navržený algoritmus, který plánuje cestu robota na základě aktuální situace v okolí a dokáže cestu za chodu přepočítávat. Součástí kapitoly je popis principu plánování, konkrétní kolaborativní úloha, ověření funkčnosti celého systému a na závěr návrh bezpečnostních prvků.

3.1 Algoritmus

S rostoucím počtem stupňů volnosti robota se výrazně zvětšuje rozsah jeho konfiguračního prostoru. Robot UR5e má šest stupňů volnosti, a tedy pracuje v šesti-dimenzionálním prostoru kloubových konfigurací. Pro takto dimenzované úlohy se dobře uplatňují vzorkovací metody, které zvládají plánování i ve složitějších prostorech bez nutnosti jejich explicitního popisu. Proto se v této práci volí plánování přímo v kloubovém prostoru. [10]

Z dostupných vzorkovacích metod byl pro tuto práci zvolen algoritmus RRT-Connect, jehož hlavní výhodou je rychlejší nalezení cesty oproti klasickému RRT. Tento přístup staví dva stromy (ze startovní a cílové konfigurace), které se snaží co nejrychleji propojit. Díky této strategii dosahuje algoritmus kratší doby výpočtu, což je klíčové pro aplikaci v prostředí s dynamickými překážkami. Vzhledem k tomu, že hlavním cílem není nalezení globálně optimální cesty, ale především jde o rychlé nalezení bezpečně realizovatelné cesty, představuje RRT-Connect vhodnou volbu pro plánování cesty v reálném čase. Algoritmy zaměřené na optimalizaci cesty, jako je například RRT*, by mohly nabídnout vhodnější cestu, ale za cenu delšího výpočtu, což je pro tuto aplikaci nežádoucí. [13]

Na tuto volbu navazuje detailní popis reprezentace prostředí, překážek a samotné implementace algoritmu přizpůsobeného pro kolaborativního robota UR5e.

3.1.1 Pracovní prostředí

Předpokládá se, že robotické rameno zůstává po celou dobu na pevném místě. Díky tomu je pracovní prostor kolem robota jasně definovaný a jeho rozměry zůstávají konstantní.

Pro účely kolizní kontroly je tento třídimenzionální pracovní prostor rozdělen na pravidelnou diskrétní mřížku, která je reprezentována trojrozměrnou binární maticí. Tato struktura slouží výhradně k detekci kolizí a není využívána při samotném plánování cesty. Každá buňka obsahuje binární informaci: hodnota 1 označuje přítomnost překážky, hodnota 0 značí volný prostor.

Takto zvolená reprezentace prostoru je vhodná nejen díky své jednoduchosti, ale i z hlediska efektivity výpočtu. Vzhledem k tomu, že mřížka je tvořena pevně definovanými buňkami, lze k ní přistupovat lokálně. Díky tomu není nutné procházet celou mřížku, což výrazně snižuje výpočetní náročnost i při komplexnějších operacích. Rychlé vyhodnocení překážek spolu s nízkými výpočetními nároky zajišťuje, že celý systém dokáže včas reagovat na změny v okolí, což je zásadní požadavek při nasazení v kolaborativním prostředí. Ve srovnání s jinými přístupy, které vyžadují přesné geometrické modely nebo výpočetně náročné simulace, se tento přístup ukazuje jako spolehlivý a praktické řešení pro dynamické a časově citlivé úlohy.

Velikost jednotlivých buněk mřížky představuje snadno nastavitelný parametr, který ovlivňuje úroveň detailu při kolizní kontrole. Menší buňky umožňují přesnější zachycení tvaru překážek, zatímco větší snižují paměťové nároky a zrychlují výpočet. Tento parametr lze flexibilně upravit podle požadavků konkrétní aplikace, a tím dosáhnout rovnováhy mezi přesností a rychlostí. Příklad vytvoření této mřížky s definovaným rozsahem os je ilustrován na obrázku 3.1, který zachycuje klíčovou část kódu pro její inicializaci.


```
function [range, voxel_grid] = initialize_voxel_grid(voxel_size)

    % Definice rozsahů os
    x_range = -0.85:voxel_size:0.85;
    y_range = -0.85:voxel_size:0.85;
    z_range = -0.15:voxel_size:1.55;

    % Inicializace prázdné voxel mřížky
    voxel_grid = zeros(length(x_range), length(y_range), length(z_range));
    range = [x_range; y_range; z_range];

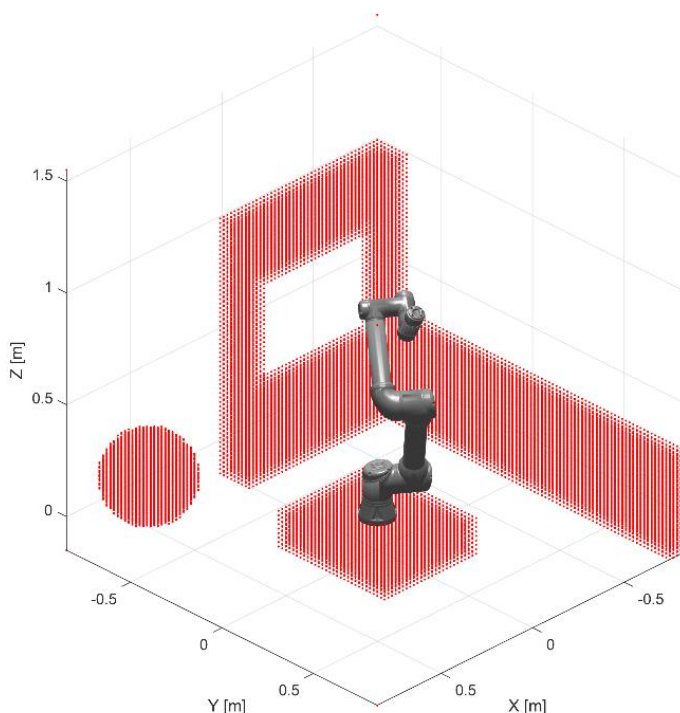
end
```

Obr. 3.1: Ukázka kódu pro vytvoření a inicializaci diskrétní mřížky kolaborativního prostoru.

3.1.2 Zpracování překážek

Překážky v pracovním prostoru jsou zaznamenávány do již vytvořené diskrétní mřížky, která slouží jako základ pro detekci kolizí.

Statické překážky, jako jsou například stoly, konstrukce nebo stěny vycházející z CAD modelu pracoviště, představují neměnné prvky prostředí. Proto je vhodné tyto objekty zapsat do mřížky již při samotné inicializaci, čímž se nejen sníží výpočetní náročnost, ale zároveň se urychlí samotný provoz systému. Ukázka výstupu diskrétní mřížky s již zanesenými statickými překážkami je uvedena na obrázku 3.2. Robot je v tomto případě umístěn uprostřed pracovního prostoru, jak je zmíněno v kapitole 3.1.1. Tento výstup slouží jako vizuální ověření správného zápisu překážek do prostoru ještě před zahájením samotného provozu systému.



Obr. 3.2: Diskrétní mřížka reprezentující pracovní prostor robota obsahující statické překážky.

Při kolaborativní spolupráci robota s člověkem je nezbytné, aby robot dokázal bezpečně reagovat na pohyb osoby v pracovním prostoru. Člověk v tomto případě představuje dynamickou překážku, jejíž aktuální poloha se může neustále měnit. Aby bylo možné v každém okamžiku vyhodnotit potenciální kolizi a zachovat bezpečnou vzdálenost, je nutné jeho přítomnost promítnout do prostředí, kde probíhá kolizní kontrola.

Pro účely kolizní kontroly je třeba z dostupných dat o poloze klíčových částí těla vytvořit objemovou reprezentaci člověka. Zvolený model je záměrně jednoduchý a skládá se ze základních geometrických tvarů (válců a sfér), které umožňují rychlé výpočty a efektivní zápis do mřížky. Každému segmentu těla je přiřazen konkrétní tvar s definovaným poloměrem, který byl stanoven na základě orientačního experimentálního měření:

Hlava – sféra o poloměru 200 mm
Paže – válec o poloměru 100 mm
Předloktí – válec o poloměru 80 mm
Zápěstí – sféry o poloměru 150 mm
Trup – válec o poloměru 200 mm
Stehna – válec o poloměru 150 mm

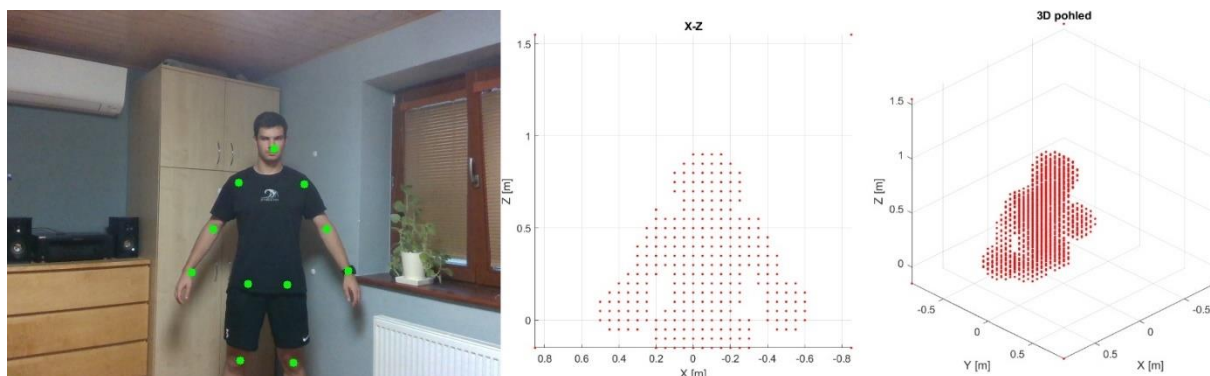
Takto vytvořený objemový model je následně promítnut do diskrétní mřížky, přičemž každá buňka nacházející se uvnitř některého z definovaných segmentů je v mřížce označena jako překážka. Možný způsob detekce klíčových bodů lidského těla je patrný z obrázku 3.3 vlevo, kde je znázorněn snímek z kamery s vyznačenými body. Napravo je pak zobrazena diskrétní mřížka obsahující odpovídající geometrický model člověka, připravená k použití pro následnou kolizní kontrolu.

Vzhledem k tomu, že člověk představuje dynamickou překážku, je nutné jeho model pravidelně aktualizovat podle aktuální polohy. Každý segment je do mřížky doplňován samostatně a lokálně pouze v okolí svého rozložení, což výrazně urychluje celý proces. Díky tomuto přístupu není nutné zpracovávat celou mřížku, ale pouze vybrané oblasti.

Celková doba potřebná pro přijetí dat, vytvoření modelu a jeho zanesení do diskrétní mřížky závisí na velikosti buňky této mřížky. V rámci testování byly zvoleny tři odlišné velikosti buňky a měřena odpovídající doba výpočtu:

- 8 cm – $\Delta t = (0,002 - 0,004)$ s
- 5 cm – $\Delta t = (0,003 - 0,007)$ s
- 3 cm – $\Delta t = (0,006 - 0,015)$ s
- 1 cm – $\Delta t = (0,14 - 0,2)$ s

Jako optimální byla zvolena velikost buňky 5 cm, která představuje nejhrubší možné rozlišení umožňující stále dostatečně přesně reprezentovat postavu člověka. Zajišťuje tak potřebnou rychlost výpočtu bez ztráty funkčnosti detekce. Výsledný model při tomto nastavení je zobrazen na obrázku 3.3.



Obr. 3.3: Detekce člověka a vizualizace jeho geometrického modelu v diskrétní mřížce.

3.1.3 Implementace algoritmu

Pro plánování cesty v kloubovém prostoru jsem implementoval vlastní rozšířenou verzi algoritmu RRT-Connect, která je přizpůsobena specifickým robota UR5e a využívá informace o aktuálním stavu prostředí reprezentovaného diskrétní mřížkou.

Algoritmus pracuje se dvěma stromy – aktivním stromem, který se v daném okamžiku rozrůstá, a pasivním stromem, který čeká na případné přiblížení. Aktivní strom vyrůstá ze startovní konfigurace (q_{start}), pasivní z cílové konfigurace (q_{goal}). V každé iteraci je generována nová kandidátní konfigurace (q_{rand}), která je buď náhodná v rámci kloubových rozsahů robota, nebo s určitou pravděpodobností přímo odpovídá cíli. Tato strategie zvyšuje šanci na rychlé propojení obou stromů.

Zvolená konfigurace (q_{rand}) je porovnána s nejbližším uzlem v aktivním stromu a vypočítá se nový bod (q_{new}) směrem k (q_{rand}), s omezenou velikostí kroku. Nový bod je následně testován z hlediska kolizí.

Nejprve je pro (q_{new}) vypočítána dopředná kinematika a určeny souřadnice všech kloubů robota v kartézském prostoru. Následně probíhá dvoufázová kolizní kontrola. V první fázi se pro každý kloub zjistí, zda jeho pozice odpovídá obsazené buňce v diskrétní mřížce. Pokud některá z těchto buněk obsahuje hodnotu 1 (překážka), je konfigurace zahozena. Tento přístup umožňuje velmi rychlé vyhodnocení kolizí, protože se kontrolují pouze konkrétní buňky, nikoliv celé okolí nebo celý prostor. Ve druhé fázi kolizní kontroly se ověřuje, zda některá část objemového modelu robota nezasahuje do překážky. Tento model je tvořen válcovými segmenty, které propojují dvojice sousedních kloubů. Pro každý válec se nejprve vypočítá jeho prostorová poloha a orientace na základě kloubových souřadnic. Následně se do mřížky lokálně doplní objem tohoto válce a ověří se, zda příslušné buňky neobsahují překážku. Stejně jako v předchozím kroku se přistupuje pouze k relevantním buňkám mřížky, což zajišťuje efektivní průběh kontroly bez nutnosti procházet celý prostor. Pokud je detekována kolize s překážkou, konfigurace je rovněž vyřazena.

Pokud konfigurace (q_{new}) úspěšně projde oběma fázemi kolizní kontroly, je přidána do aktivního stromu jako nový uzel. Zároveň je uložena informace o jejím rodiči, tedy uzlu, ze kterého byla vygenerována.

Následně algoritmus vyhledá nejbližší bod (q_{near}) v pasivním stromu a zahájí snahu o jeho přiblížení směrem ke konfiguraci (q_{new}). Tento pohyb je realizován postupnými kroky ve směru mezi oběma body, přičemž každý mezikrok představuje novou potenciální konfiguraci, která je stejně jako (q_{new}) podrobena dvoufázové kolizní kontrole. Pokud v některém z kroků dojde ke kolizi, nebo je překročen maximální počet povolených kroků, proces rozšiřování se přeruší a algoritmus pokračuje další iterací, při které si stromy navzájem prohodí role.

Po každém přidání nového bodu se navíc vyhodnocuje, zda se oba stromy nepřiblížily na vzdálenost menší než zadaná prahová hodnota (*goal_tolerance*). Pokud je tato podmínka splněna, přechází se k sestavení výsledné cesty.

Výsledná cesta je vytvořena zpětným sledováním rodičovských vazeb v obou stromech, od propojeného bodu zpět ke konfiguracím (q_{start}) a (q_{goal}). Obě části jsou následně spojeny do jedné sekvence, která tvoří úplnou cestu v kloubových souřadnicích

Pro dosažení plynulejšího a přirozenějšího průběhu cesty je výsledná cesta v kloubovém prostoru dále zjemněna pomocí spline interpolace. Nejprve je vytvořena parametrická časová osa, na jejímž základě se pro každou kloubovou osu samostatně provede interpolace pomocí kubického spline. Tím vznikne hladký průběh natočení jednotlivých kloubů, který eliminuje trhaný nebo nepřirozený pohyb mezi jednotlivými konfiguracemi a zajišťuje plynulý přechod podél celé cesty.

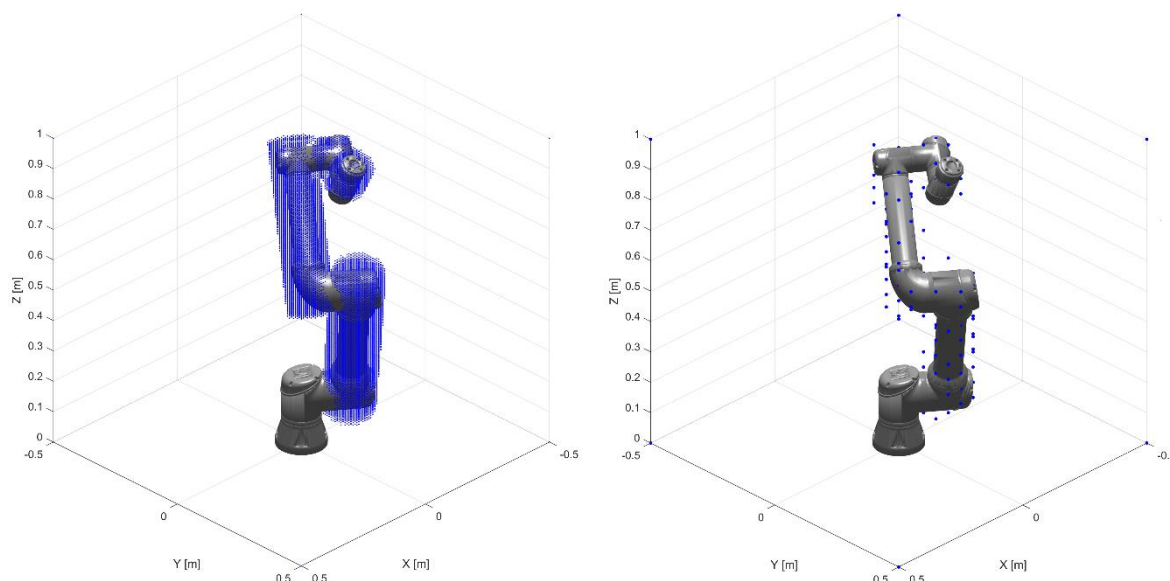
Implementace celého algoritmu až po sestavení kolizně bezpečné cesty v kloubovém prostoru je obsažena ve funkci *rrt_connect.m*, jejíž úplné znění je uvedeno v příloze této práce.

Kolizní model robota a výpočet kloubových pozic

Při vytváření kolizního modelu robota je nezbytné znát přesné prostorové pozice všech jeho kloubů v kartézském prostoru. Vestavěné funkce pro výpočet dopředné kinematiky v Malabu však nebylo možné použít, protože výpočet vyžaduje i specifický směrový vektor, který určuje směr offsetu prvního segmentu kolizního modelu.

Kolizní model robota je tvořen soustavou válcových segmentů, přičemž každý válec propojuje dvojici sousedních kloubů. Tato aproximace umožňuje jednoduchou a efektivní reprezentaci objemu jednotlivých článků ramene robota, jak je znázorněno na obrázku 3.4.

Kolizní model je dále snadno rozšiřitelný o další segment reprezentující koncový nástroj robota. Tento prvek je realizován jako přídatný válec navazující na poslední kloub, přičemž jeho rozměry lze upravit podle konkrétního typu použitého nástroje. V případě potřeby může být válec nahrazen sférickým objemem. Díky této variabilitě je možné kolizní model přizpůsobit konkrétním podmínkám nasazení robota bez zásahu do základní struktury výpočtu.



Obr. 3.4: Vizualizace kolizního modelu robota promítnutého do diskrétní mřížky pro dvě různé velikosti buněk, s překrytím přesného 3D modelu.

3.2 Laboratorní úloha

Navržený algoritmus a princip kolizní kontroly byly následně ověřeny v rámci laboratorní úlohy, která byla navržena za účelem testování celého systému v reálném prostředí.

V rámci laboratorních podmínek byla navržena úloha, která představuje základní kolaborativní činnost typu pick and place, při které robot manipuluje s objekty v prostoru sdíleném s člověkem. Tento scénář zároveň reprezentuje běžné typy spolupráce, jako je doplňování polotovarů do strojů, odebrání výrobků nebo asistence při montážních činnostech.

3.2.1 CAD model pracoviště

Pro potřeby návrhu a testování byl vytvořen CAD model, který slouží k vizuálnímu a prostorovému znázornění celé úlohy a reprezentuje rozvržení pracoviště. Model zároveň umožňuje vymezit pracovní prostor robota, ve kterém probíhá manipulace s objekty. Člověk má v rámci úlohy přístup ke všem částem pracoviště, a sdílí tak s robotem společný prostor.

CAD model zahrnuje všechny klíčové prvky pracoviště potřebné k testování (viz obrázek 3.5). Níže následuje stručný popis jednotlivých komponent.

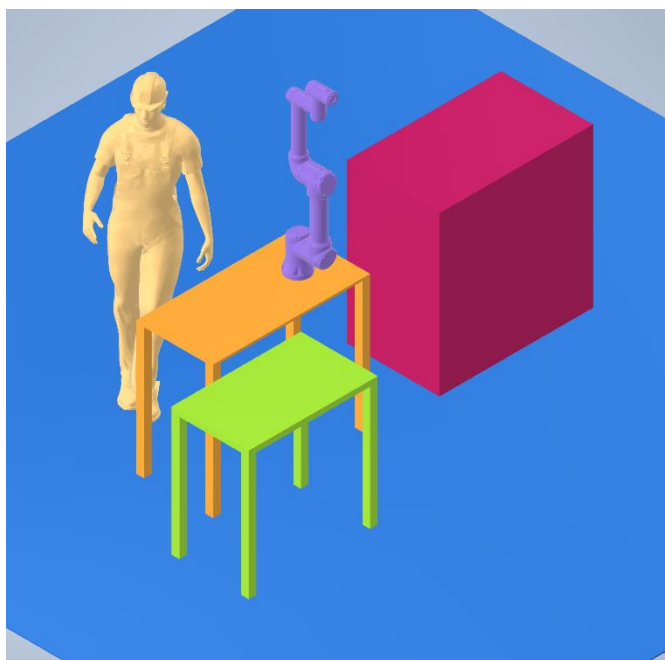
Podstavec robota (oranžová) zajišťuje stabilní ukotvení ramene UR5e vůči pracovnímu prostředí a určuje jeho výchozí výšku a orientaci v prostoru. Jeho rozměry odpovídají reálnému vozíku, ke kterému je robot připevněn.

Robot UR5e (fialová) je v modelu umístěn ve středu mezi pracovními stoly, které v rámci úlohy obsluhuje. [28]

Vstupní pracovní stůl (zelená) je umístěn těsně vedle podstavce robota a slouží jako výchozí místo, na kterém jsou umístěny objekty k manipulaci.

Výstupní pracovní stůl (červená) je v modelu umístěn na druhé straně robota než vstupní. Od podstavce je odsazen tak, aby zachoval volný prostor umožňující průchod člověka a přístup ke všem potřebným částem pracoviště. Stůl slouží jako cílové místo, kam robot umísťuje předměty.

Postava člověka (béžová) je v modelu zahrnuta především za účelem znázornění výškového a prostorového vztahu vůči robotu. Je umístěna do pozice, která odpovídá kolaborativní situaci, kdy člověk částečně zasahuje do pracovního prostoru robota. [29]

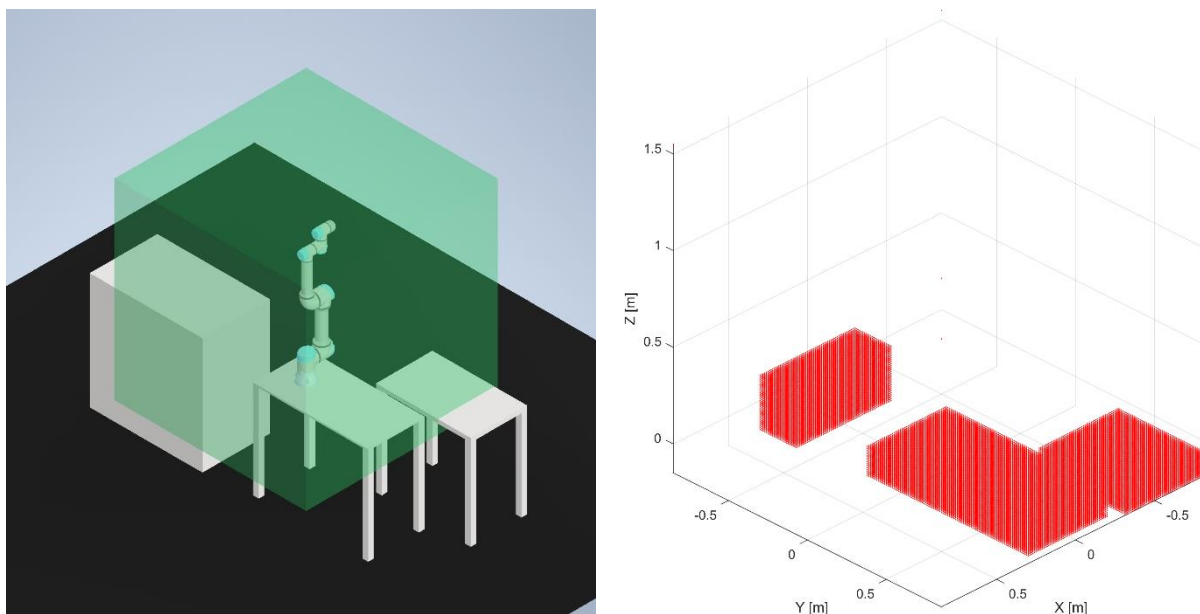


Obr. 3.5: CAD model pracoviště laboratorní úlohy.

Maximální rozsah robota UR5e je podle technické dokumentace výrobce přibližně 850 mm. [30] Pro integraci do navrženého systému diskrétní mřížky byla kolaborativní zóna definována jako krychle o rozměru strany 1700 mm. Půdorysně je tento prostor vymezen symetricky kolem počátku souřadného systému robota, zatímco ve výšce má rozsah -150 mm až 1550 mm od tohoto počátku.

Volba krychlového tvaru je záměrná, protože se s diskrétní mřížkou této podoby počítalo už při samotné implementaci algoritmu. Rozsah výšky byl stanoven tak, aby pokrýval reálně využívaný prostor pro pohyb robota. Takhle vymezený pracovní prostor je vizuálně vyznačen v modelu na obrázku 3.6 vlevo.

Všechny objekty, které se nacházejí uvnitř vymezeného prostoru, jsou na začátku běhu algoritmu zaneseny do diskrétní mřížky ve formě statických překážek, jak je popsáno v kapitole 3.1.2. Tyto překážky jsou definovány pomocí souřadnicových rozsahů ve třech osách, které dohromady tvoří kvádry. Pro tento účel byla vytvořena funkce, která umožňuje takto definované překážky vložit do diskrétní mřížky. Výsledná podoba diskrétní mřížky odpovídající prostoru vymezenému na obrázku 3.6 vlevo je znázorněna na obrázku 3.6 vpravo.



Obr. 3.6: Vymezení pracovního prostoru v CAD modelu a diskrétní mřížka obsahující statické překážky pracovního prostoru.

3.2.2 Hloubková kamera a zpracování dat

Hloubková kamera Intel RealSense D435 umožňuje současné snímání barevného RGB obrazu a hloubkových dat v reálném čase. Díky hloubkovým datům je možné určit prostorové souřadnice objektů, což je zásadní pro následné zpracování. Vzhledem k tomu, že kamera byla již dostupná v laboratoři, byla z praktických důvodů zvolena jako senzor pro tento projekt. [25]

K získání informací o vzdálenosti jednotlivých pixelů kamera využívá principu aktivní stereoskopie. Samotná konstrukce kamery zahrnuje dvě infračervené kamery, které jsou od sebe vodorovně posunuté a snímají scénu ze dvou mírně odlišných pohledů. Tento posun způsobuje, že se stejný bod ve scéně promítne do různých pozic na obrazech obou kamer. Vzniklý rozdíl v poloze tohoto bodu se označuje jako disparita. Na základě tohoto rozdílu je pak pomocí triangulace dopočítána vzdálenost každého bodu od kamery. [26]

Součástí kamery je také infračervený projektor, který promítá na scénu jemný strukturovaný vzor a zvyšuje tak kontrast v infračerveném obraze. Díky tomu jsou infračervené kamery schopné lépe detekovat i hladké nebo málo strukturované povrchy. Projektor tím výrazně přispívá ke zlepšení kvality získaných hloubkových dat. [26]

Detekce člověka v obraze

Pro účely sledování osoby v pracovním prostoru bylo potřeba zajistit spolehlivou detekci člověka v obraze. Zvažovány byly dvě běžně používané knihovny: OpenPose (OP) a MediaPipe (MP). OP se vyznačuje detekcí více osob s vysokou přesností, ale klade vyšší nároky na výpočetní výkon a její implementace je náročnější. Oproti tomu MP je vhodná pro sledování jedné osoby a vyžaduje pouze minimální systémové požadavky. Cílem bylo otestovat obě knihovny, ovšem při implementaci OP se objevily komplikace spojené s nestabilitou při napojení na živý obraz z kamery, což představovalo zásadní problém pro reálné nasazení systému. Z těchto důvodů byla zvolena knihovna MP, která fungovala bez komplikací. [27]

MP standardně detekuje 31 klíčových bodů na lidském těle, avšak pro tuto aplikaci stačí využít pouze 11 z nich, které slouží k následném vytvoření modelu dynamické překážky popsaného v kapitole 3.1.2. Výstup detekce je zobrazen na snímcích z kamery na obrázku 3.7.



Obr. 3.7: RGB a hloubkový snímek s detekovanými klíčovými body osoby pomocí knihovny MediaPipe.

Zpracování a přenos dat z kamery

Zpracování dat z hloubkové kamery a jejich následné odesílání do prostředí Matlab bylo realizováno pomocí skriptu v jazyce Python. Python byl zvolen zejména díky snadné implementaci potřebných knihoven:

- pyrealsense2 [31] – přístup k RGB a hloubkovým datům z kamery RealSense
- MediaPipe [32] – detekce osoby a určení klíčových bodů lidského těla v RGB obraze
- OpenCV [33] – zpracování a vizualizaci obrazových dat
- json – formátování výstupních dat do struktury JSON
- socket – komunikace přes TCP/IP, propojení Pythonu a Matlabu

Aby bylo možné správně přiřadit každému detekovanému bodu v RGB obraze odpovídající hloubku, je nutné nejprve zarovnat hloubkový a barevný obraz tak, aby měly shodné souřadnice a rozlišení.

Pro zlepšení kvality hloubkových dat jsou aplikovány filtry, které jsou přímo dostupné v knihovně `pyrealsense2`:

- `Spatial_filter` – prostorové vyhlazení obrazu, redukce náhodných šumů v okolí hran
- `Temporal_filter` – zamezení náhodných výkyvů hloubky mezi snímky
- `Hole_filling` – doplnění chybějící informace o hloubce v místech, kde senzor nezaznamenal odraz

Po předzpracování hloubkových dat následuje detekce osoby pomocí knihovny MP, která v RGB obraze identifikuje vybrané klíčové body lidské postavy. MP vrací pozice těchto bodů ve formě normalizovaných 2D souřadnicích (x , y), vyjádřených jako poměr vůči rozměrům obrazu. Tyto souřadnice jsou následně převedeny na pixelové souřadnice (tj. vynásobením normalizovaných hodnot skutečnou šířkou a výškou obrazu) a ke každému bodu je z hloubkového obrazu přiřazena odpovídající vzdálenost od kamery D v metrech.

K určení 3D kartézských souřadnic jednotlivých bodů je dále zapotřebí znát vnitřní parametry kamery, a to konkrétně ohniskové vzdálenosti a souřadnice hlavního bodu obrazu (tj. bodu, kam dopadá optická osa snímáče). Samotný výpočet prostorových souřadnic odpovídá rovnicím (1) – (3).

$$X = \frac{(x_{pixel} - c_x) \cdot D}{f_x} \quad (1)$$

$$Y = -\frac{(y_{pixel} - c_y) \cdot D}{f_y} + y_{offset} \quad (2)$$

$$Z = \sqrt{D^2 - X^2 - Y^2} - z_{offset} \quad (3)$$

Kde:

x_{pixel} , y_{pixel} jsou souřadnice bodu v obraze (v pixelech)

D je vzdálenost bodu od kamery v metrech

f_x , f_y jsou ohniskové vzdálenosti kamery

c_x , c_y jsou souřadnice optického středu v pixelech

y_{offset} , z_{offset} jsou korekce pro zarovnání souřadnic se soustavou robota

Takto vypočítané souřadnice detekovaných klíčových bodů jsou následně zpracovány do struktury ve formátu JSON a odeslány pomocí TCP/IP komunikace do prostředí Matlab, kde slouží jako vstup pro další výpočty (viz kapitola 3.1.2). Komunikace je zajištěna pomocí soketového spojení. Na straně Pythonu je vytvořen klient, který se připojí k definovanému portu a odesílá JSON řetězec (viz obrázek 3.8). Na straně Matlabu běží TCP server, který data postupně přijímá, převádí je z formátu JSON do struktury a dále je zpracovává (viz obrázek 3.9).

Kompletní skript `pose_to_matlab.py`, který tuto funkcionalitu zajišťuje, je uveden v příloze této práce.


```
# TCP/IP připojení k MATLAB serveru
HOST = "127.0.0.1"
PORT = 5051
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((HOST, PORT))

# Odeslání dat ve formátu JSON
data_str = json.dumps(pose_data) + "\n"
sock.sendall(data_str.encode())
```

Obr. 3.8: Odesílání dat ve formátu JSON z Python skriptu do Matlabu.

```
% Inicializace TCP/IP serveru v MATLABu
port = 5051;
server = tcpserver("127.0.0.1", port, "Timeout", 10);

% Čekání na první zprávu
while server.NumBytesAvailable == 0
    pause(0.1);
end

% Načtení přijaté zprávy
latest_data = readline(server);
json_data = jsondecode(latest_data);
```

Obr. 3.9: Načtení a dekódování dat v Matlabu pomocí TCP serveru.

3.2.3 Umístění kamery

Prvním kritériem umístění kamery bylo správné detekování člověka na obraze. Na základě série experimentálních měření byla stanovena výška kamery na 1350 mm nad úrovní podlahy. Při této výšce je kamera schopná spolehlivě detekovat osobu, která se nachází již ve vzdálenosti 500 mm od kamery. Kamera se v této výšce zaměřuje především na horní polovinu lidského těla, kde se nejčastěji odehrává potenciální interakce s ramenem robota. Dolní část těla však není opomíjena, protože i s ní se může robot dostat do interakce. Diskrétní mřížka pokrývá prostor až 150 mm pod úrovní základny robota a kamera je v tomto rozsahu schopná spolehlivě detekovat přítomnost člověka.

Osa Y v souřadném systému kamery reflektuje výšku obrazu, proto je nutné doplnit hodnotu y_{offset} jako rozdíl výšky kamery (1350 mm) a počátku robota (865 mm). Výsledná hodnota y_{offset} v rovnici (2) je rovna 0,485 m.

Druhou důležitou veličinou při návrhu umístění kamery byla její vodorovná vzdálenost od pracovního prostoru robota, která je závislá jak na kvalitě hloubkových dat, tak na schopnosti správně detekovat osobu. Cílem bylo zajistit, aby kamera spolehlivě pokrývala celý pracovní prostor robota UR5e, jehož maximální dosah činí 1700 mm.

Nejmenší možná vzdálenost byla dána především zorným polem, které je nutné pro pokrytí pracovního prostoru v šířce alespoň 1700 mm. Příliš blízké umístění kamery může omezit záběr a způsobit, že osoba na okraji zóny nebude detekována. Z tohoto důvodu byla stanovena minimální vzdálenost kamery na 1200 mm od okraje sledované oblasti.

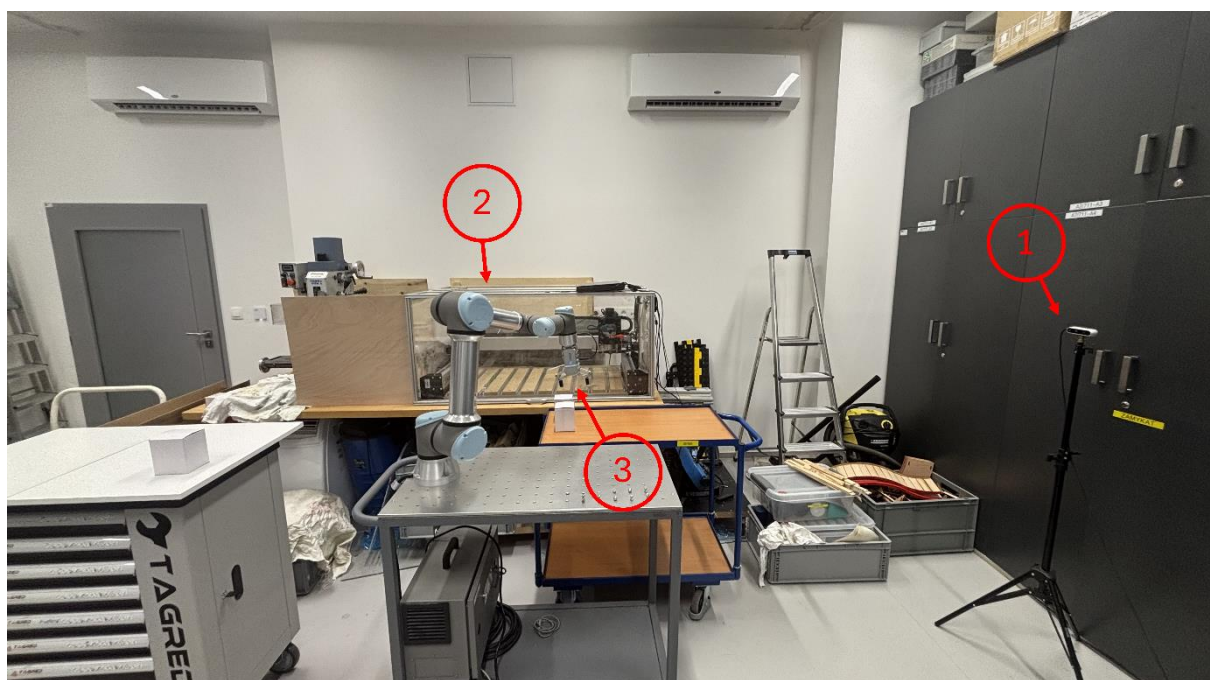
S rostoucí vzdáleností od kamery dochází ke zhoršení přesnosti měření hloubky, což by mohlo negativně ovlivnit detekci polohy osoby. Na základě testování byla proto stanovena maximální vzdálenost osoby od kamery na 3300 mm, při které jsou data stále dostatečně spolehlivá.

Na základě provedených testů byla kamera umístěna do vzdálenosti 2250 mm od počátku souřadného systému, čímž je zajištěno pokrytí pracovního prostoru v rozsahu 1400 mm až 3100 mm. Takto zvolená pozice zároveň poskytuje dostatečnou rezervu vzhledem ke stanovené minimální a maximální použitelné vzdálenosti. Uvedená hodnota zároveň odpovídá parametru z_{offset} z rovnice (3), který udává horizontální vzdálenost kamery od počátku robota.

Kamera byla umístěna z bočního pohledu na pracovní prostor, tedy kolmo na směr vstupu člověka na pracoviště. Toto umístění bylo zvoleno s cílem minimalizovat počet situací, kdy by se člověk nacházel za ramenem robota, což by mohlo vést k nevhodné detekci nebo úplné ztrátě detekce. Umístění kamery je zachyceno na obrázku 3.10 v kapitole 3.2.4.

3.2.4 Integrace hardwaru a implementace řídicí logiky

Pracoviště bylo fyzicky sestaveno na základě návrhu zpracovaného v CAD modelu. Zahrnuje hloubkovou kameru Intel RealSense D435 (1), kolaborativního robota UR5e (2) a koncový efektor OnRobot RG2 gripper (3) určený k manipulaci s objekty. Rozmístění jednotlivých komponent je zachyceno na obrázku 3.10.



Obr. 3.10: Reálné testovací pracoviště odpovídající rozložení popsané úlohy.

Kolaborativní robot UR5e je vybaven dotykovým ovládacím rozhraním, tzv. teach pendantem (obr. 2.3), pomocí kterého lze robota ovládat v režimu Local Control. Tento režim byl využit výhradně pro manuální nastavení konfigurace robota. Kloubové úhly bylo možné upravovat buď ručně, nebo softwarově prostřednictvím přímého posunu jednotlivých kloubů, případně posunem koncového efektoru v kartézském prostoru. Takto nastavené konfigurace byly následně použity jako výchozí a cílové body pro plánování cesty.

V režimu Remote Control byl robot řízen z prostředí Matlabu prostřednictvím síťového připojení přes TCP/IP. Pro tento účel byla využita již vytvořená knihovna *UR_robot* [37], která

umožňuje řízení robota z Matlabu bez nutnosti programování v jazyce URScript. V rámci této knihovny jsem využil především funkce pro navázání spojení s robotem, zastavení pohybu a odesílání kloubových konfigurací, které má robot následovat. Robot si mezi těmito konfiguracemi automaticky vypočítá trajektorii pomocí svého interního řídicího systému a provádí plynulý pohyb. Pokud není zadán žádný parametr pro zaoblení trajektorie, robot se u každého bodu krátce zastaví. V případě zadání této hodnoty si robot naplánuje trajektorii tak, aby všechny konfigurace projel plynule bez zastavení. [35]

Pro manipulaci s objekty byl robot vybaven koncovým efektem OnRobot RG2 gripper, který slouží k uchopování a přenášení předmětů v rámci úlohy. Maximální rozevření čelistí je 100 mm, což umožňuje uchopit i větší předměty používané v této úloze. Gripper je znázorněn na obrázku 3.11.

Ovládání gripperu bylo zajištěno pomocí funkcí obsažených ve stejné knihovně jako řízení robota. V rámci úlohy byly využity základní funkce pro otevření a zavření čelistí.

Kolizní model robota byl rozšířen o segment reprezentující připojený gripper s objektem. Na základě měření byla délka koncového efektoru nastavena na 265 mm a doplněna do funkce pro výpočet dopředné kinematiky. Na základě této hodnoty byl přidán odpovídající válec pokrývající objem gripperu včetně uchopeného objektu, jak je popsáno v části 3.1.3.



Obr. 3.11: OnRobot RG2 gripper použitý jako koncový efektor robota UR5e.

Řídicí logika systému

Celá řídicí logika systému byla implementována v hlavním skriptu *main.m*. Struktura tohoto skriptu je přehledně znázorněna v blokovém schématu na obrázku 3.14.

V úvodní části probíhá inicializace všech klíčových komponent systému. Nejprve je navázáno spojení s robotem UR5e prostřednictvím zmíněné knihovny *UR_robot*, která umožňuje nastavení parametrů pohybu, jako je poloměr oblouku, maximální rychlost a zrychlení kloubů. Následně je aktivován gripper, jehož výchozí stav je nastaven na úplné otevření čelistí. Posledním krokem inicializace je spuštění TCP/IP serveru, který zajišťuje komunikaci s Python skriptem připojeným k hloubkové kameře. V této fázi skript čeká na první příchozí data z kamery a až poté přechází do další části.

Po úvodní inicializaci je vytvořena prázdná diskretní mřížka kolaborativního prostoru, do které jsou následně zaneseny statické překážky podle popisu uvedeného v kapitolách 3.1.2 a 3.2.1. Součástí těchto překážek jsou také místa, kde se v průběhu úlohy nacházejí manipulované objekty, aby se zamezilo jejich kolizi s robotem během pohybu.

Následně systém vstupuje do „vnější smyčky“, ve které robot opakovaně vykonává předem definované úkoly. Tyto úkoly jsou reprezentovány maticí *waypoints*, ve které každý řádek obsahuje dvě kloubové konfigurace:

- Startovní konfigurace – výchozí pozice, ze které probíhá plánování cesty
- Operační konfigurace – pozice, ve které robot provede manipulaci s objektem

Po přesunu do startovní konfigurace robot nejprve vykoná požadovanou akci v operační poloze a následně se vrací zpět na výchozí pozici. Přehled této části implementace je uveden na obrázku 3.12. Ta slouží jako počáteční bod pro plánování cesty k dalšímu cíli, který je definován jako startovní konfigurace v následujícím řádku matice.

```
% waypoints: každý řádek = 1 bod [start (1-6), oper (7-12)]
waypoints = [5.9 -2.3 -0.6 -1.6 1.5 -0.3 5.9 -2.4 -1.0 -1.2 1.5 -0.3;
             1.5 -2.1 -0.7 -1.8 1.5  0.0 1.5 -2.1 -1.1 -1.4 1.5  0.0];

% výběr aktuální dvojice [start -> goal] a start_oper
[q_start, q_goal, q_start_oper, way_point_index] = ...
    get_next_waypoints(waypoints, way_point_index);

ur.set_config(q_start');           % přesun do startovní konfigurace
ur.set_config(q_start_oper');      % přesun do operační konfigurace

if gripper_loaded == 1
    ur.RG2(100,20,0,false,true);   % otevření gripperu
    gripper_loaded = 0;
else
    ur.RG2(80,8,0.163,true,true);  % uchopení objektu
    gripper_loaded = 1;
end

ur.set_config(q_start');           % návrat do startovní konfigurace
```

Obr. 3.12: Výběr konfigurací z matice *waypoints* a následná manipulace pomocí gripperu.

Poté je pomocí algoritmu RRT-Connect naplánována výchozí cesta v prostředí se statickými překážkami. V okamžiku, kdy se robot podle ní začne pohybovat, systém přechází do „vnitřní smyčky“.

V první části „vnitřní smyčky“ dochází ke zpracování aktuálních dat z kamery. Pokud jsou data dostupná, je do diskretní mřížky doplněn model člověka. V opačném případě, kdy žádná nová data nepříjdou, systém předpokládá, že se člověk v kolaborativní zóně nenachází a pokračuje se pouze s mřížkou obsahující statické překážky.

Aktuální naplánovaná cesta je následně zkontrolována pomocí dvoustupňové kolizní kontroly vůči aktuální podobě diskretní mřížky. Ověřuje se, zda v jakémkoliv bodě cesty nedochází ke kolizi. Pokud je cesta vyhodnocena jako bezpečná, robot pokračuje v pohybu.

V případě, že cesta zasahuje do překážky, je nutné provést její přeplánování. Nejprve robot obdrží příkaz k zastavení. Následně je pomocí algoritmu RRT-Connect vypočítána nová cesta ze současné konfigurace robota (načtené přímo ze zařízení) do původního cílového bodu. Ukázka zjednodušeného kódu této části je uvedena na obrázku 3.13.

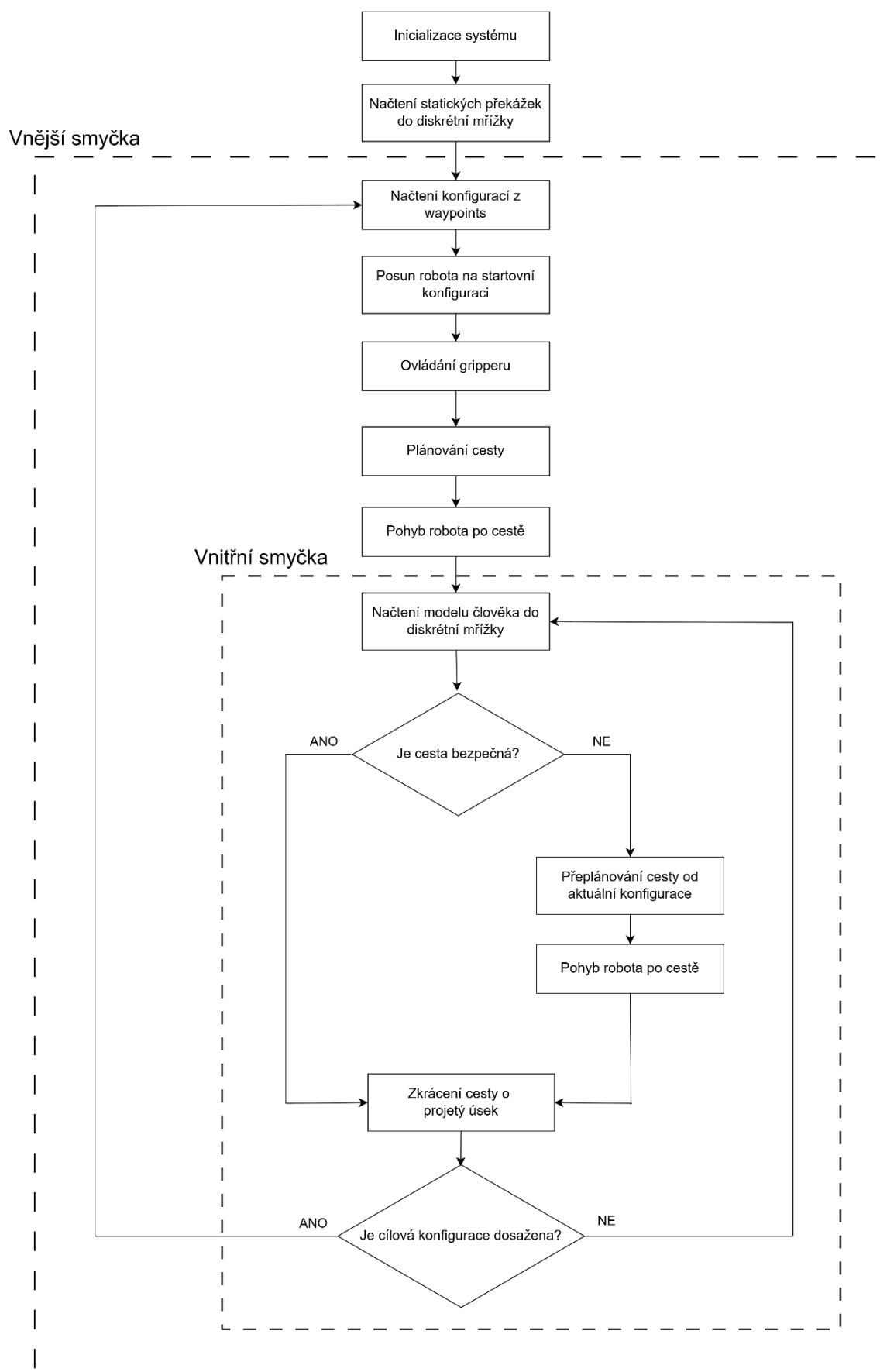
Jakmile je nová cesta nalezena, je odeslána robotovi k vykonání. V případě, že se novou cestu nepodaří naplánovat (kvůli blokujícím překážkám), robot zůstává zastaven a aktuální cesta je označena jako nevalidní. V dalším cyklu, po novém načtení dat o překážkách, je proveden opětovný pokus o nalezení bezpečné cesty.

Na konci každého průchodu smyčkou je aktuální cesta zkrácena podle aktuální konfigurace robota. Zůstává tak pouze ta část cesty, kterou má robot ještě projet. Díky tomu se při další validaci nevyhodnocují úseky, které už byly bezpečně projety. Pokud by se například objevil člověk v místech, kudy robot už projel, nemá to na další průběh žádný vliv.

V každé iteraci vnitřní smyčky se zároveň kontroluje, zda se robot již nenachází v cílové konfiguraci s určitou tolerancí. Pokud je tato podmínka splněna, smyčka se ukončí a řízení pokračuje ve vnější smyčce, kde je načten další řádek z matice *waypoints*, čímž se celý proces opakuje.

```
is_valid = is_path_valid(actual_q_path, ~, voxel_grid, range, ~);  
  
if is_valid  
    % cesta je bezpečná - robot pokračuje v pohybu  
else  
    % cesta není bezpečná  
    q_now = ur.config; % Aktuální konfigurace robota  
    ur.stop;          % Zastavení robota  
    % naplánování nové cesty od aktuální konfigurace  
    [actual_q_path, ~, ~, goal_flag] = rrt_6dof_connect_02(q_now, q_goal', voxel_grid, range);  
end
```

Obr. 3.13: Zjednodušený kód pro ověření bezpečnosti cesty a reakce systému.



Obr.3.14: Blokové schéma logiky řízení systému.

3.2.5 Testy a hodnocení výsledků

Za účelem ověření funkčnosti a spolehlivosti navrženého systému byly provedeny testy, které simulují různé scénáře přítomnosti člověka v kolaborativním prostoru. Testování se zaměřilo především na detekci osoby v reálném čase, rychlost a spolehlivost reakce systému na změny v okolí a ověření, zda nedošlo k přímé interakci mezi robotem a člověkem.

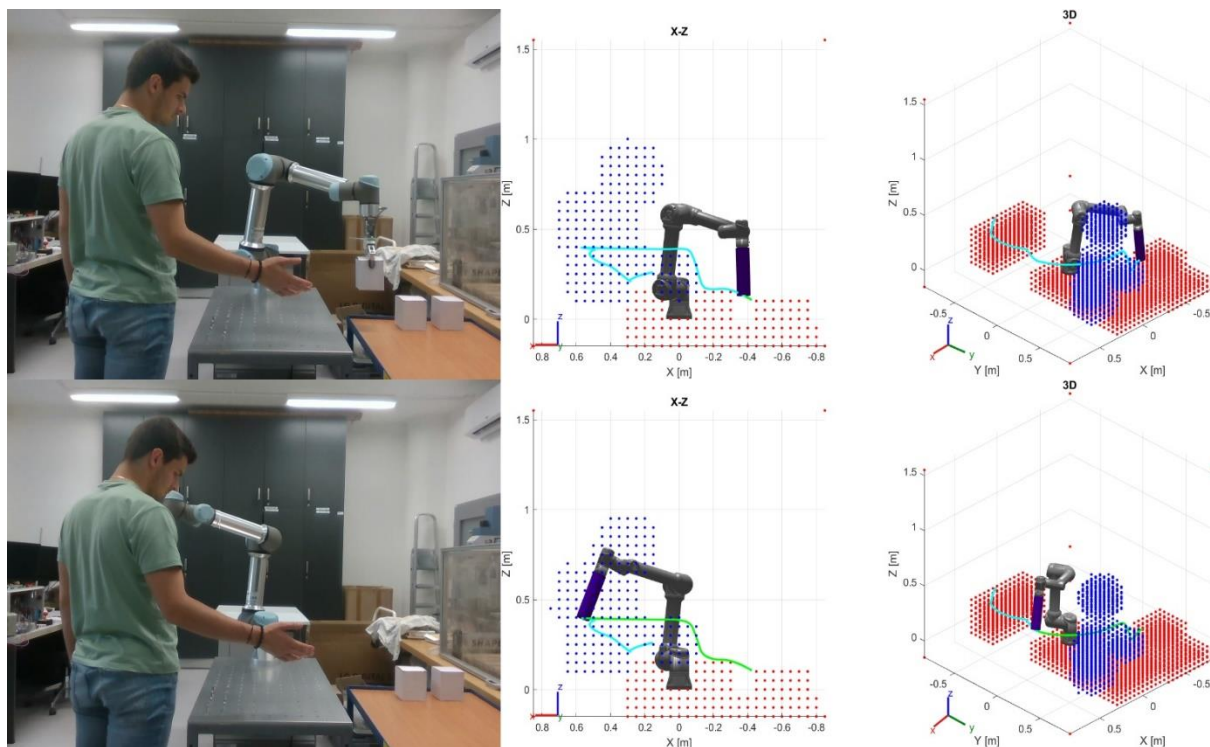
Videozáznam zachycující průběhy všech tří testovacích scénářů je dostupný na odkazu: <https://www.youtube.com/watch?v=4gGTHiZyHUA>

Člověk stojí na místě

V tomto testu se člověk po celou dobu nacházel na jednom místě v kolaborativní zóně. Účelem testu bylo ověřit, že systém v takové situaci správně provede přeplánování cesty na začátku pohybu, a poté robot projede celou trasu plynule, bez nutnosti dalších zásahů.

Výsledky testu potvrzují, že kamera správně detekovala přítomnost osoby a tato překážka byla stabilně zahrnuta v diskretní mřížce (na výstupech znázorněna modře). Systém reagoval očekávaným způsobem a cestu přeplánoval pouze na začátku a robot po celou dobu vykonával pohyb plynule a bez zastavení.

Z doprovodných snímků (viz obrázek 3.15) je patrné, že naplánovaná cesta (pro jednoduchou vizualizaci zobrazena jako cesta TCP – modrá křivka) zůstala po celou dobu neměnná. Vykrešená je také skutečně projetá cesta TCP (zelená křivka), která naplánovanou cestu téměř přesně kopíruje. Menší odchylky jsou způsobeny vědomě nastaveným parametrem pro zaoblení trajektorie (tzv. blend radius), který určuje, jak blízko se musí robot přiblížit k jednotlivým bodům cesty, aby je mohl plynule projet bez zastavení. Tento princip odpovídá plánování trajektorie v interním systému robota a nemá negativní vliv na bezpečnost výsledné cesty.



Obr. 3.15: Průběh testu s člověkem stojícím v kolaborativní zóně.

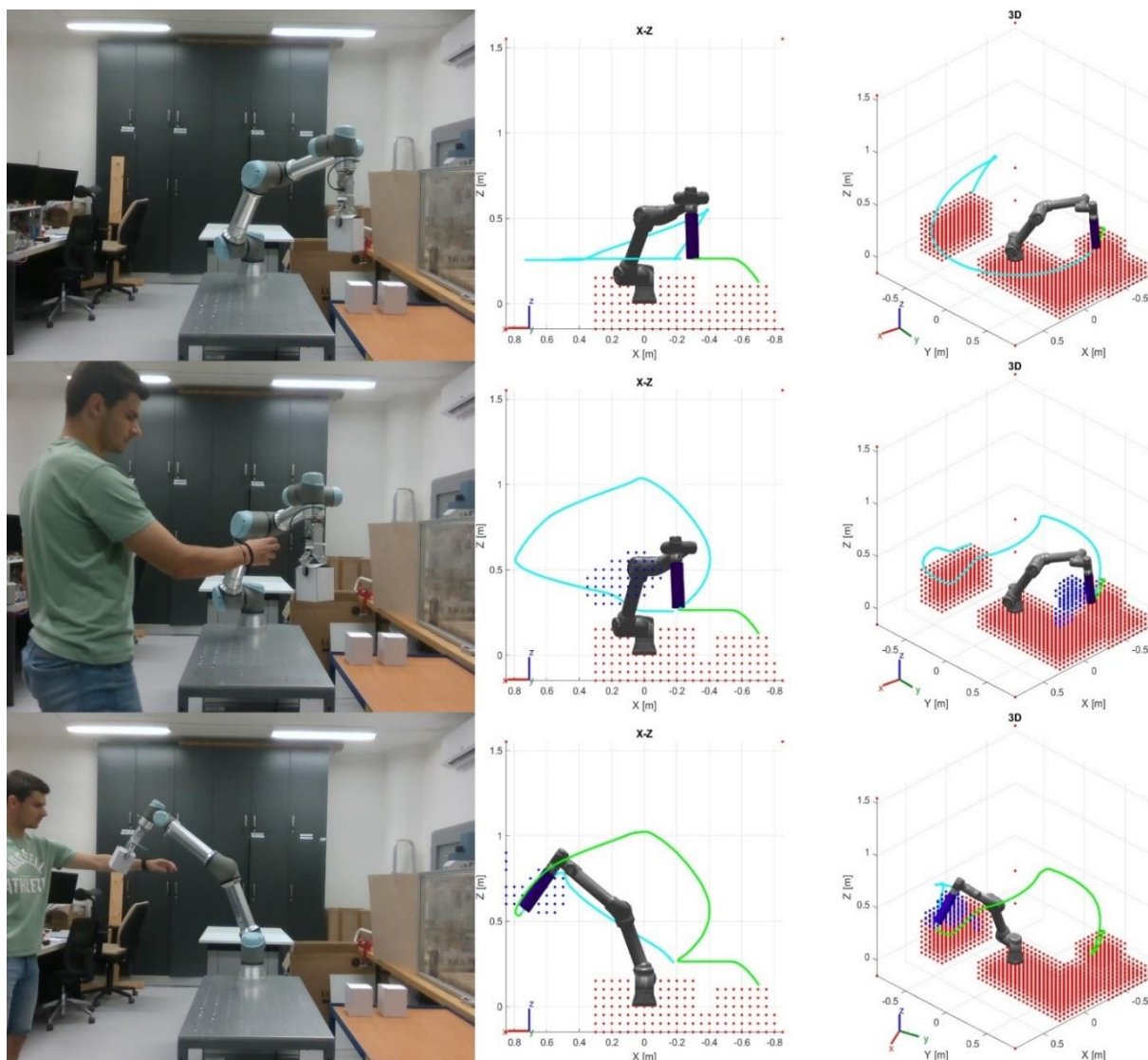
Člověk se pohybuje v kolaborativní zóně a zasahuje do plánované cesty

Tento test ověřil chování systému v situaci, kdy se člověk pohyboval v blízkosti robota a opakovaně zasahoval do jeho plánované cesty. Cílem testu bylo prověřit schopnost systému v reálném čase reagovat na dynamickou překážku a upravovat cestu tak, aby nedošlo ke kolizi.

Na začátku testu robot zahájil pohyb podél cesty naplánované v prostředí se statickými překážkami. Tato situace odpovídá prvnímu řádku obrázku 3.16, kdy se člověk ještě nacházel mimo kolaborativní zónu. Jakmile však vstoupil do oblasti plánované cesty, systém překážku rozpoznal a provedl přeplánování. Robot následně pokračoval v pohybu po nové cestě, která se překážce vyhýbala, jak je vidět na druhém řádku obrázku 3.16.

V další fázi testu (třetí řádek obrázku 3.16) robot pokračoval po cestě, která byla naplánována v reakci na první výskyt překážky. Ve chvíli, kdy člověk opět vstoupil do prostoru cesty, došlo k opětovné detekci kolize a dalšímu přeplánování.

Test prokázal, že systém dokáže rychle a spolehlivě reagovat na změny v okolí robota. Jakmile se člověk postupně přibližoval a zasahoval do plánované cesty, docházelo k opakovanému přeplánování. Cesta byla vždy upravena podle aktuální polohy překážek tak, aby zůstala bezpečná.



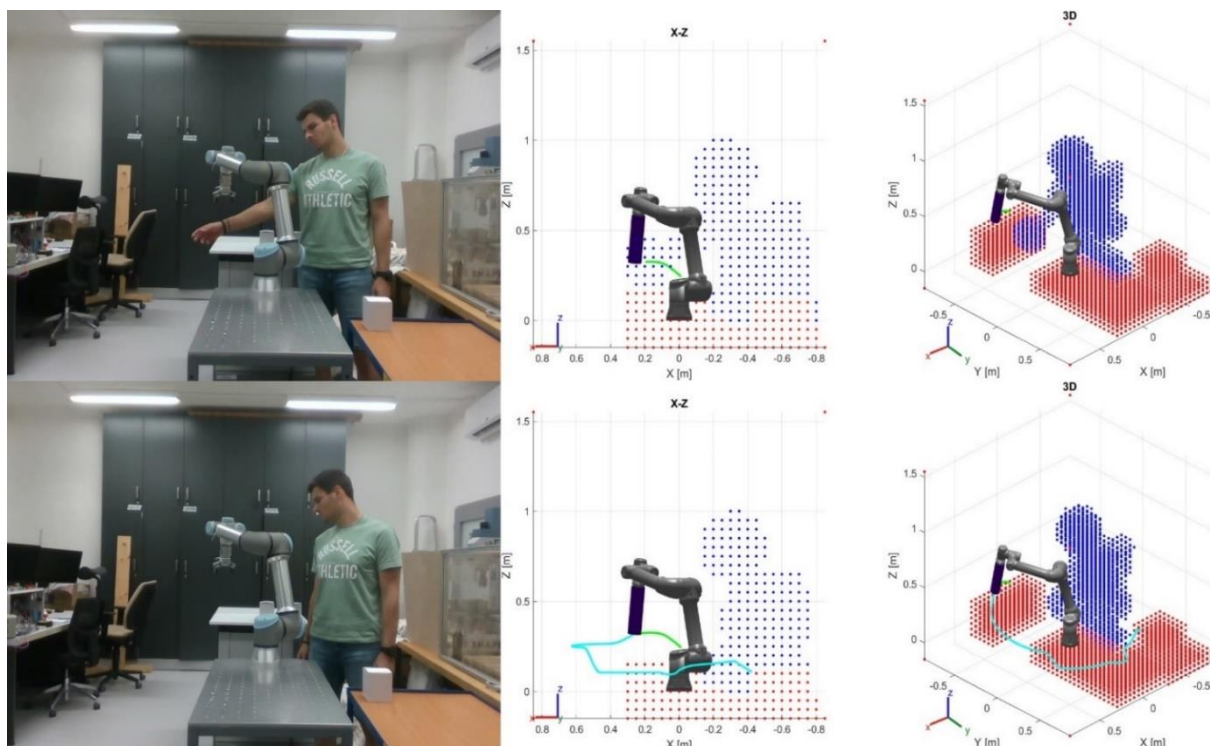
Obr. 3.16: Průběh testu s člověkem pohybujícím se v kolaborativní zóně.

Člověk blokuje přístup k cíli

Test ověřoval chování systému v situaci, kdy člověk zcela blokoval všechny možné cesty vedoucí k cíli. Cílem bylo zjistit, zda robot zůstane bezpečně stát a zda systém bude opakovaně provádět pokusy o nalezení nové cesty.

Na začátku testu se robot pohyboval po dříve vypočítané cestě. Ve chvíli, kdy člověk zablokoval všechny možné cesty vedoucí k cílové konfiguraci, došlo k opakovaným pokusům o nalezení nové cesty. Jelikož však žádné řešení v daný moment neexistovalo, robot se správně zastavil a zůstal na místě, jak je zachyceno na prvním řádku obrázku 3.17.

Jakmile člověk uvolnil prostor a bylo možné naplánovat bezpečnou cestu, robot na ni navázal a pokračoval v pohybu, jak je patrné z druhém řádku obrázku 3.17.



Obr. 3.17: Průběh testu s člověkem blokujícím pohyb.

Hodnocení výsledků

Všechny provedené testy byly opakovány vícekrát a výsledky zůstávaly konzistentní. Systém se v jednotlivých scénářích choval podle očekávání a reagoval na přítomnost překážky.

Rychlost jednotlivých částí systému byla dostatečná pro včasnou reakci. Výstup z kamery byl na TCP/IP server poslán v intervalu 10 až 15 snímků za sekundu (0,1-0,067 s). Následné zpracování detekovaných dat do diskrétní mřížky probíhalo řádově v milisekundách a nezpůsobovalo tak výrazné zpoždění.

Validace cesty probíhala v rozmezí 0,02 až 0,06 sekundy, zatímco výpočet nové cesty pomocí algoritmu trval dle složitosti prostředí 0,05 až 0,5 sekundy. Pokud nebylo možné najít bezpečné řešení, robot se automaticky zastavil. Výsledná reakční doba systému se tak pohybovala mezi 0,14 až 0,66 sekundy.

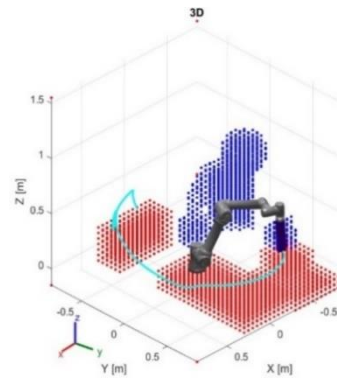
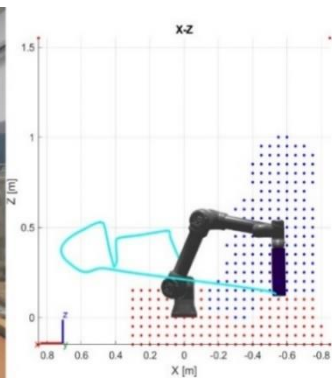
Ve všech případech bylo zajištěno, že nedošlo ke kontaktu mezi člověkem a žádnou aktivní částí systému (robot, gripper a přenášený objekt). Byla tak splněna klíčová podmínka bezpečného provozu kolaborativní aplikace.

Během testů se objevily situace, kdy se osoba nacházela za robotem z pohledu kamery. V těchto případech byla postava systémem sice správně detekována, ale některé klíčové body těla byly hloubkově promítnuty na povrch samotného robota. To vedlo k tomu, že část modelu člověka byla chybně přiřazena přímo k robotu.

Výsledkem byla umělá překážka v bezprostředním okolí robota, což vedlo k jeho zastavení. Z hlediska bezpečnosti je tento výsledek akceptovatelný, protože nedocházelo k pohybu robota. Avšak z hlediska správné funkce detekčního systému tato situace představuje zásadní nedostatek.

Tato situace je zachycen na obrázku 3.18. Na snímku z kamery je vidět, že osoba stála za robotem, přičemž jeden z detekovaných bodů (levé zápěstí) byl přiřazen přímo na tělo robota. V důsledku toho systém vytvořil umělou překážku v blízkosti koncového efektoru. Tím došlo k zablokování všech možných cest vedoucích k cíli a k následnému zastavení robota.

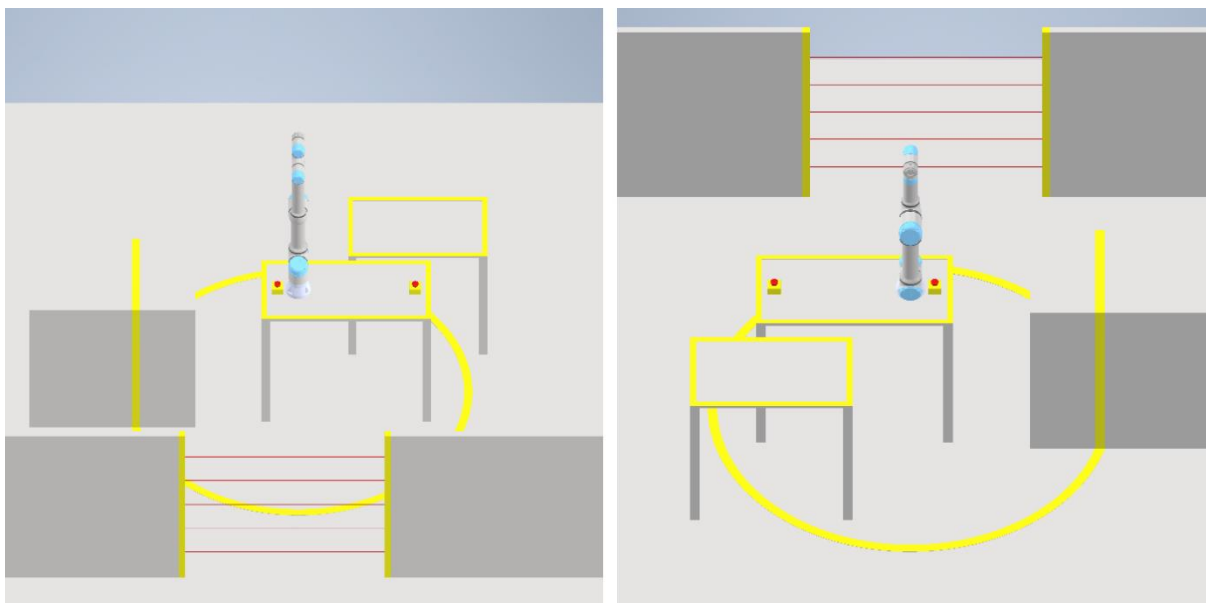
Tento problém by bylo možné řešit použitím druhé hloubkové kamery umístěné z jiného pohledu. Fúze dat z více kamer by umožnila přesněji odlišit skutečnou polohu osoby od jiných částí a tím zvýšit spolehlivost systému při zachování bezpečnosti.



Obr. 3.18: Situace, kdy chybná detekce osoby způsobila vytvoření falešné překážky a přerušení pohybu robota.

3.3 Návrh bezpečnostních prvků

Tato část představuje návrh konkrétních bezpečnostních prvků pro případ, že by bylo kolaborativní pracoviště navrženo v této práci skutečně uvedeno do provozu. Jedná se o hypotetický návrh, který navazuje na teoretické shrnutí bezpečnostních principů a ochranných prvků v kapitole 2.1.



Obr. 3.19: Vizualizace návrhu bezpečnostních prvků v CAD modelu pracoviště.

3.3.1 Tlačítka nouzového zastavení (E-STOP)

Tlačítko nouzového zastavení představuje základní bezpečnostní prvek, jehož cílem je umožnit obsluze rychlou reakci v krizových situacích. Tento prvek je definován normou ISO 13850, která stanovuje požadavky na jeho funkčnost, viditelnost, dostupnost a konstrukci. Tlačítko musí být červené, umístěné na žlutém podkladu a po aktivaci musí zajistit bezpečné zastavení robota. Opětovné spuštění vyžaduje ruční restartování, což zajišťuje bezpečnou obnovu provozu. [34]

Na navrženém pracovišti jsou umístěna dvě nouzová tlačítka, přičemž každé se nachází na jedné straně podstavce robota. Jsou ve výšce zajišťující snadnou dosažitelnost pro stojící osobu. Díky své poloze, předepsanému barevnému provedení a otevřenému prostoru kolem jsou snadno vizuálně identifikovatelná během běžné činnosti na pracovišti. Jejich umístění je patrné z celkového pohledu na pracoviště na obrázku 3.19, detailní provedení tlačítka je pak zobrazeno na obrázku 3.20.



Obr. 3.20: Detailní pohled na nouzové tlačítko.

3.3.2 Detekce přítomnosti člověka

Pro bezpečný provoz kolaborativního pracoviště je zásadní, aby systém včas detekoval přítomnost osoby na pracovišti, kde může dojít k interakci s robotem. Hlavním prvkem, který tuto detekci zajišťuje, je světelná závora umístěná při vstupu na pracoviště. V režimu bez přítomnosti člověka není pohyb robota nijak omezován a může se pohybovat vyššími rychlostmi bez bezpečnostních omezení. Ve chvíli, kdy člověk přeruší paprsek světelné závory a vstoupí na pracoviště, systém automaticky přejde do režimu aktivního vyhýbání se člověku, jak je popsáno v předchozích kapitolách.

Světelná závora je umístěna v dostatečné vzdálenosti od kolaborativního prostoru. Tím je systému poskytnuta potřebná časová rezerva pro přechod do režimu vyhýbání, a je tak umožněna bezpečná reakce ještě před vstupem osoby do kolaborativního prostoru. Její poloha je zachycena v celkovém pohledu na pracoviště na obrázku 3.19.

Redundantním prvkem pro detekci osoby na pracovišti je v tomto případě hloubková kamera, která je již v systému implementována. Ve spojení se zpracováním obrazu pomocí knihovny MediaPipe zajišťuje detekci přítomnosti člověka. Pokud robot pracuje v režimu bez omezení a kamera detekuje osobu v pracovním prostoru, tak systém robota ihned zastaví. Zároveň systém ohlásí selhání hlavního bezpečnostního prvku, jehož funkčnost je nutné před dalším spuštěním ověřit a obnovit.

3.3.3 Vizuální označení kolaborativní zóny

Kolaborativní prostor je na pracovišti vyznačen pomocí žluté kružnice aplikované na podlahu. Její poloměr je 1000 mm od základny robota, čímž je zajištěno pokrytí celého kolaborativního prostoru. Vyznačení dále navazuje i na podstavec robota a okolní pracovní stoly, aby bylo z různých pohledů zřetelné, jak je kolaborativní prostor vymezen. Toto označení je rovněž patrné v celkovém pohledu na pracoviště na obrázku 3.19.

Účelem tohoto označení je předejít nechtěnému vstupu osoby do kolaborativního prostoru. Jedná se o pasivní bezpečnostní prvek, který podporuje bezpečné chování osob na pracovišti. Už při vstupu by mělo být zároveň jednoznačně a viditelně uvedeno, že se jedná o pracoviště s kolaborativním robotem.

4 Závěr

V rámci této bakalářské práce byl navržen a realizován systém pro plánování cesty kolaborativního robota UR5e, který umožňuje jeho bezpečný provoz v prostředí sdíleném s člověkem. Robot dokáže aktivně přizpůsobovat svůj pohyb aktuálním podmínkám a vyhýbat se tak detekovaným překážkám, což umožňuje plynulou a nepřerušovanou spolupráci člověka a stroje bez potřeby fyzických zábran.

V rámci implementace byl navržen vlastní algoritmus RRT-Connect v šestidimenzionálním kloubovém prostoru, optimalizovaný pro rychlé vyhodnocování kolizí na základě diskretní mřížky a objemového modelu robota. Klíčovým prvkem je právě diskretní mřížka reprezentující pracovní prostor, která umožňuje efektivní kolizní kontrolu na základě binární informace o výskytu překážek. Do této mřížky jsou postupně doplňovány jak statické překážky reprezentující rozložení pracoviště, tak model člověka pohybujícího se v pracovním prostoru. Aktuální poloha člověka je určována na základě dat z hloubkové kamery s využitím modelu pro detekci osoby v obraze.

Navržený systém byl ověřen na laboratorní úloze typu pick and place, která simuluje běžnou kolaborativní činnost v prostředí sdíleném s člověkem. Během testování bylo provedeno několik scénářů s různým výskytem osoby v pracovním prostoru. Ve všech případech se robot úspěšně vyhnul překážkám, nedošlo ke kontaktu s člověkem a úloha byla vždy bezpečně dokončena. Systém tak prokázal spolehlivou funkci i v dynamických situacích. Díky své modulární struktuře je systém snadno přizpůsobitelný různým typům úloh i odlišným konfiguracím pracoviště, což umožňuje jeho flexibilní nasazení v různých provozních podmínkách.

U testovací úlohy byly zároveň navrženy doplňující bezpečnostní prvky pro případné nasazení do reálného provozu. Ty splňují příslušné bezpečnostní požadavky a lze je přímo začlenit do stávajícího systému.

Během testování bylo identifikováno určité omezení při detekci osoby nacházející se za ramenem robota, kdy docházelo ke vzniku falešné překážky. Bezpečnost provozu tím sice nebyla narušena, avšak došlo ke zbytečnému přerušení pohybu robota. Toto omezení by bylo možné vyřešit nasazením vícekamerového systému a datové fúze, což by vedlo k přesnější a spolehlivější detekci osoby v celém pracovním prostoru.

Tato práce ukazuje, že kolaborativní robot může aktivně reagovat na změny v okolí a přizpůsobovat své chování aktuální situaci v pracovním prostoru. Díky tomu se posouvají dosavadní hranice mezi klasickým automatizovaným řízením a skutečně bezpečnou spoluprací člověka a robota.

Seznam použitých zdrojů

- [1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *New ISO technical specification for collaborative robot safety*. [online]. 2016 [cit. 2025-05-20]. Dostupné z: <https://www.iso.org/news/2016/03/Ref2057.html>
- [2] ČSN EN ISO 10218-1. *Roboty a robotická zařízení – Požadavky na bezpečnost průmyslových robotů – Část 1: Roboty*. Praha: Česká agentura pro standardizaci, 2012.
- [3] TECHNIKA A TRH. *Roboty a automatizace s brzdami značky Mayr – technologie brzd, která inspiruje*. [online], 2024 [cit. 2025-05-20]. Dostupné z: <https://www.technikaatrh.cz/pohony/roboty-a-automatizace-s-brzdami-znacky-mayr-technologie-brzd-ktera-inspiruje>
- [4] ČSN EN ISO 10218-2. *Roboty a robotická zařízení – Požadavky na bezpečnost průmyslových robotů – Část 2: Systémy robotů a integrace*. Praha: Česká agentura pro standardizaci, 2011.
- [5] ISO/TS 15066. *Robots and robotic devices – Collaborative robots*. Ženeva: International Organization for Standardization, 2016.
- [6] VOJÁČEK, Antonín. *Problematika bezpečnosti kolaborativních robotů – ISO/TS 15066*. In: Automatizace.hw.cz [online]. 2019 [cit. 2025-05-21]. Dostupné z: <https://automatizace.hw.cz/problematika-bezpecnosti-kolaborativnich-robotu-isots-15066.html>
- [7] SICK. *Safety for collaborative robot applications*. [online]. SICK Sensor Blog, 2022 [cit. 2025-05-21]. Dostupné z: <https://www.sick.com/ag/en/sick-sensor-blog/safety-for-collaborative-robot-applications/w/blog-safety-collaborative-robot-applications>
- [8] LAVALLE, Steven M. *Motion Planning: Part I – The Essentials*. IEEE Robotics & Automation Magazine. [online]. 2011, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/5751929>
- [9] GONZÁLEZ, David, PÉREZ, Joshué, MILANÉS, Vicente, NASHASHIBI, Fawzi. *A review of motion planning techniques for automated vehicles*. IEEE Transactions on Intelligent Transportation Systems. [online]. 2016, [cit. 2025-05-21]. Dostupné z: https://www.researchgate.net/publication/284766879_A_Review_of_Motion_Planning_Techniques_for_Automated_Vehicles
- [10] LAVALLE, Steven M. *Planning Algorithms*. [online]. 2006, [cit. 2025-05-21]. Dostupné z: <https://lavalle.pl/planning/bookbig.pdf>
- [11] MATHWORKS. *Probabilistic Roadmaps. (PRM)*. [online]. [cit. 2025-05-21]. Dostupné z: <https://www.mathworks.com/help/robotics/ug/probabilistic-roadmaps-prm.html>
- [12] MATHWORKS. *plannerRRT*. [online]. [cit. 2025-05-21]. Dostupné z: <https://www.mathworks.com/help/nav/ref/plannerrrt.html>

-
- [13] KARAMAN, Sertac, FRAZZOLI, Emilio. *Sampling-based Algorithms for Optimal Motion Planning*. [online]. 2011, [cit. 2025-05-21]. Dostupné z: <https://arxiv.org/pdf/1105.1186>
 - [14] JM9176. *RRT_RRTstar_Bi-RRT*. [online]. GitHub. [cit. 2025-05-21]. Dostupné z: https://github.com/jm9176/RRT_RRTstar_Bi-RRT?tab=readme-ov-file
 - [15] KUFFNER, James J., LAVALLE, Steven M. *RRT-Connect: An Efficient Approach to Single-Query Path Planning*. [online]. 2000, [cit. 2025-05-21]. Dostupné z: https://www.cs.cmu.edu/afs/cs/academic/class/15494-s14/readings/kuffner_icra2000.pdf
 - [16] KHATIB, Oussama, *Real-time obstacle avoidance for manipulators and mobile robots*. [online]. 1985, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/1087247>
 - [17] PETROVIČ, Luka, *Motion planning in high-dimensional spaces*. [online]. 2018, [cit. 2025-05-21]. Dostupné z: <https://arxiv.org/abs/1806.07457>
 - [18] WU, Zhengtian, DAI, Jinyu, JIANG, Baoping, KARIMI, Hamid Reza. *Robot Path Planning Based on Artificial Potential Field with Deterministic Annealing*. [online] ISA Transactions, 2023, [cit. 2025-05-21]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0019057823000769>
 - [19] QUINLAN, Sean, KHATIB, Oussama. *Elastic Bands: Connecting Path Planning and Control*. [online], 1993, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/291936>
 - [20] MAGYAR, Bence, TSIOGKAS, Nikolaos, DERAY, Jérémie, PFEIFFER, Sammy, LANE, David. *Timed-Elastic Bands for Manipulation Motion Planning*. [online], 2019, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/8758897>
 - [21] PAPPAS, Pantelis, CHIOU, Manolis, ESPIMOS, Georgios-Theofanis, NIKOLAOU, Grigoris, STOLKIN, Rustam. *VFH+ Based Shared Control for Remotely Operated Mobile Robots*. [online], 2020, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/9292585>
 - [22] CHANG, Lu, SHAN, Liang, JIANG, Chao, DAI, Yuewei. *Reinforcement Based Mobile Robot Path Planning with Improved Dynamic Window Approach in Unknown Environment*. [online], 2021, [cit. 2025-05-21]. Dostupné z: <https://link.springer.com/article/10.1007/s10514-020-09947-4>
 - [23] SCHULMAN, John et al. *Motion Planning with Sequential Convex Optimization and Convex Collision Checking*. [online], 2014, [cit. 2025-05-21]. Dostupné z: https://www.researchgate.net/publication/273222349_Motion_planning_with_sequential_convex_optimization_and_convex_collision_checking
 - [24] MCMAHON, Troy, SIVARAMAKRISHNAN, Aravind, GRANADOS, Edgar, BEKRIS, Kostas E. *A Survey on the Integration of Machine Learning with Sampling-Based Motion Planning*. [online], 2024, [cit. 2025-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/9946979>

-
- [25] Intel RealSense. *Depth Camera D435*. [online], [cit. 2025-05-22]. Dostupné z: <https://www.intelrealsense.com/depth-camera-d435>
- [26] GRUNNET-JEPSEN, Anders, SWEETSER, John N., WINER, Paul, TAKAGI, Akihiro, WOODFILL, John. *Projectors for D400 Series Depth Cameras*. [online], Intel RealSense Developer Documentation, [cit. 2025-05-22]. Dostupné z: <https://dev.intelrealsense.com/docs/projectors>
- [27] VISHNU, J G, DIVYA, S J, *A Comparative Study of Human Pose Estimation*. [online], 2023, [cit. 2025-05-22]. Dostupné z: https://www.researchgate.net/publication/366956503_A_Comparative_Study_of_Human_Pose_Estimation
- [28] GrabCAD. *Robot e-Series UR5e*. [online], 2019, [cit. 2025-05-22]. Dostupné z: <https://grabcad.com/library/robot-eseries-ur5e-1>
- [29] GrabCAD. *Male Worker EU Hand 2023*. [online], 2023, [cit. 2025-05-22]. Dostupné z: https://grabcad.com/library/001_male_worker_eu_hand_2023-1
- [30] UNIVERSAL ROBOTS. *UR5e technical specification*. [online], 2024, [cit. 2025-05-22]. Dostupné z: https://www.universal-robots.com/media/1807465/ur5e_e-series_datasheets_web.pdf
- [31] Intel RealSense. *librealsense – Intel RealSense SDK 2.0*. [online], 2024, [cit. 2025-05-22]. Dostupné z: <https://github.com/IntelRealSense/librealsense>
- [32] GOOGLE. *MediaPipe*. [online], [cit. 2025-05-22]. Dostupné z: <https://github.com/google-ai-edge/mediapipe>
- [33] OPENCV. *Open Source Computer Vision Library*. [online], [cit. 2025-05-22]. Dostupné z: <https://github.com/opencv/opencv>
- [34] ČSN EN ISO 13850. *Bezpečnost strojních zařízení – Funkce nouzového zastavení – Zásady pro konstrukci*. Praha: Česká agentura pro standardizaci, 2017.
- [35] UNIVERSAL ROBOTS. *URScript Manual – Version 5.12*. [online], [cit. 2025-05-22]. Dostupné z: https://s3-eu-west-1.amazonaws.com/ur-support-site/163530/scriptmanual_5.12.pdf
- [36] TULIP TECHNOLOGIES. *RG2 Two Fingers Gripper*. [online], [cit. 2025-05-22]. Dostupné z: <https://www.tuliptechs.com/en/product/rg2-two-fingers-gripper>
- [37] SLADKÝ, Adam. *Knihovna pro ovládání kolaborativního robota UR5e z prostředí Matlab/Simulink*. [online], Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2021, [cit. 2025-05-22]. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=224710
- [38] FEDELE, Giuseppe, D'ALFONSO, Luigi, CHIARAVALLLOTI, Francesco, D'AQUILA, Gaetano. *Obstacles Avoidance Based on Switching Potential Functions*. [online]. Journal of Intelligent & Robotic Systems, 2018, [cit. 2025-05-21]. Dostupné z: https://www.researchgate.net/publication/320174864_Obstacles_Avoidance_Based_on_Switching_Potential_Functions

-
- [39] PILZ. *Zabezpečení otevřených nebezpečných zón a míst zásahu světelnými závorami*. [online], [cit. 2025-05-23]. Dostupné z: <https://www.pilz.com/cs-CZ/products/applications/electrosensitive-protective-equipment>
- [40] SICILIANO, Bruno, KHATIB, Oussama (ed.). *Springer Handbook of Robotics*. 2. vydání. [online]. Springer. 2016, [cit. 2025-05-23]. Dostupné z: <https://mrce.in/ebooks/Robotics-Springer%20Handbook%20of%20Robotics%202nd%20Ed.pdf>
- [41] VIGNOLLES, Arthur, GARCIA, Ricardo. *Improving Safety Performance in Cobots with Torque Sensors*. [online]. TE Connectivity, 2023, [cit. 2025-05-23]. Dostupné z: <https://www.te.com/en/whitepapers/sensors/improving-safety-performance-in-cobots-with-torque-sensors.html>
- [42] DEUTSCHE GESETZLICHE UNFALLVERSICHERUNG. *Collaborative robot systems*. [online]. 2017, [cit. 2025-05-23]. Dostupné z: https://www.dguv.de/medien/fb-holzundmetall/publikationen-dokumente/infoblaetter/infobl_englisch/080_collaborativerobotsystems.pdf
- [43] ROBOTS. *LBR iiwa*. [online]. [cit. 2025-05-23]. Dostupné z: <https://robotsguide.com/robots/lbriiwa>
- [44] LOUISA. *Amazon Safety Wearable for Harmonious Human-Robot Workplace Relation*. [online]. SafetyCulture, 2019, [cit. 2025-05-23]. Dostupné z: <https://blog.safetyculture.com/safetyculture-news/amazon-safety-wearable>
- [45] COHEN, Nancy. *Amazon's safety wearable is for human-robot workspace*. [online], TechXplore, 2019, [cit. 2025-05-23]. Dostupné z: <https://techxplore.com/news/2019-01-amazon-safety-wearable-human-robot-workspace.html>

Seznam obrázků

- Obr. 2.1: Čtyři režimy kolaborativní spolupráce.
Obr. 2.2: Vizualizace světelných závor u vstupu do prostoru s robotem
Obr. 2.3: Fotografie ručního ovladače robota UR5e.
Obr. 2.4: Robot KUKA LBR iiwa s hladkým, zaobleným designem určeným pro bezpečnou spolupráci s člověkem.
Obr. 2.5: Roadmapa vytvořená metodou PRM s připojeným startem a cílem a vyznačenou výslednou cestou.
Obr. 2.6: Plánování cesty metodou RRT v prostředí s překážkami.
Obr. 2.7: Rozrůstání stromu a výsledná cesta nalezená algoritmem RRT*.
Obr. 2.8: Rozrůstání dvou stromů a nalezení cesty metodou RRT-Connect.
Obr. 2.9: Potenciálové pole s vyznačenou cestou robota, včetně uvíznutí v lokálním minimu.
Obr. 3.1: Ukázka kódu pro vytvoření a inicializaci diskrétní mřížky kolaborativního prostoru.
Obr. 3.2: Diskrétní mřížka reprezentující pracovní prostor robota obsahující statické překážky.
Obr. 3.3: Detekce člověka a vizualizace jeho geometrického modelu v diskrétní mřížce.
Obr. 3.4: Vizualizace kolizního modelu robota promítnutého do diskrétní mřížky pro dvě různé velikosti buněk, s překrytím přesného 3D modelu.
Obr. 3.5: CAD model pracoviště laboratorní úlohy.
Obr. 3.6: Vymezení pracovního prostoru v CAD modelu a diskrétní mřížka obsahující statické překážky pracovního prostoru.
Obr. 3.7: RGB a hloubkový snímek s detekovanými klíčovými body osoby pomocí knihovny MediaPipe.
Obr. 3.8: Odesílání dat ve formátu JSON z Python skriptu do Matlabu.
Obr. 3.9: Načtení a dekodování dat v Matlabu pomocí TCP serveru.
Obr. 3.10: Reálné testovací pracoviště odpovídající rozložení popsané úlohy.
Obr. 3.11: OnRobot RG2 gripper použitý jako koncový efektor robota UR5e.
Obr. 3.12: Výběr konfigurací z matice waypoints a následná manipulace pomocí gripperu.
Obr. 3.13: Zjednodušený kód pro ověření bezpečnosti cesty a reakce systému.
Obr. 3.14: Blokové schéma logiky řízení systému.
Obr. 3.15: Průběh testu s člověkem stojícím v kolaborativní zóně.
Obr. 3.16: Průběh testu s člověkem pohybujícím se v kolaborativní zóně.
Obr. 3.17: Průběh testu s člověkem blokujícím pohyb.
Obr. 3.18: Situace, kdy chybná detekce osoby způsobila vytvoření falešné překážky a přerušení pohybu robota.
Obr. 3.19: Vizualizace návrhu bezpečnostních prvků v CAD modelu pracoviště.
Obr. 3.20: Detailní pohled na nouzové tlačítko.