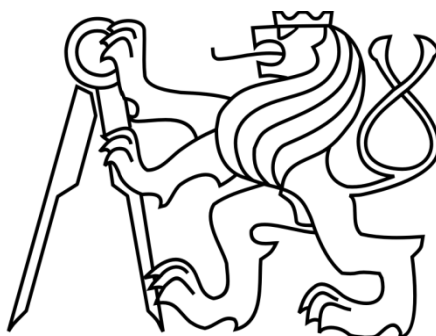


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY

Odbor mechaniky a mechatroniky



Diplomová práce

Pracoviště pro imitační robotické lakování

Praha, 2022

Michael Bryan

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bryan** Jméno: **Michael** Osobní číslo: **466544**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Aplikované vědy ve strojním inženýrství**
Specializace: **Mechatronika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Pracoviště pro imitační robotické lakování

Název diplomové práce anglicky:

Imitation robotic painting workplace

Pokyny pro vypracování:

- 1) Seznamte se s běžnými pracovišti pro robotické lakování v průmyslové praxi
- 2) Navrhněte imitační lakovací pracoviště s průmyslovým robotem a s polohovacím stolem
- 3) Navrhněte a proveďte experiment pro ověření přesnosti senzorického systému
- 4) Navrhněte systémové řešení řídicího SW pro imitační robotické pracoviště
- 5) Proveďte simulaci lakování na imitačním robotickém pracovišti

Seznam doporučené literatury:

- M. Mejdrechová, Návrh robotického lakovacího pracoviště s využitím snímání pohybu a imitace lakýrníka, Diplomová práce, ČVUT v Praze 2020
V. Stejskal, M. Valášek, Kinematics and Dynamics for Machinery, New York, Marcel Dekker 1996

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Jan Pelikán, Ph.D. odbor mechaniky a mechatroniky FS


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


Datum zadání diplomové práce: **22.04.2022**

Termín odevzdání diplomové práce: **15.08.2022**

Platnost zadání diplomové práce: _____


Ing. Jan Pelikán, Ph.D.
podpis vedoucí(ho) práce

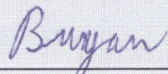

doc. Ing. Miroslav Španiel, CSc.
podpis vedoucí(ho) ústavu/katedry


doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

15. 8. 2022
Datum převzetí zadání


Podpis studenta

Abstrakt

Tématem této práce je návrh robotického lakovacího pracoviště s použitím trackovacího zařízení k programování lakovací trajektorie. Pracoviště je doplněno o otočný polohovací stůl, který umožňuje robotu nalakovat požadovaný předmět ze všech stran. Práce se zabývá řízením stolu a robotu, ale i systémovým řešením pracoviště jako celku. Dále byla ověřena přesnost navržené trackovací kamery Intel RealSense T265. Možnost jejího použití se ukázala být pro tento způsob robotického lakování omezená. Funkčnost celého pracoviště je na závěr ověřena v simulaci.

Klíčová slova

Robotické lakování, sledování pohybů, polohovací stůl, s-křivka, plánování trajektorie, optimalizace, simulace robotu

Abstract

The topic of this thesis is design of a robotic painting workstation using a tracking device to program robotic painting trajectory. The workstation is equipped with rotational positioning table that enables the robot to paint the desired object from all sides. The thesis deals with the control of the table and robot, but also the system design of the workstation as a whole. Furthermore, the accuracy of the proposed Intel RealSense T265 tracking camera was verified. The possibility of its use proved to be limited for this method of robotic painting. Finally, the functionality of the whole workstation is verified in a simulation.

Keywords

Robot painting, motion tracking, positioning table, s-curve, trajectory planning, optimization, robot simulation

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Praze, dne

.....
Podpis

Poděkování

Rád bych poděkoval mému vedoucímu práce Ing. Janu Pelikánovi, Ph.D. za cenné rady a užitečné připomínky k vypracování diplomové práce. Také bych rád poděkoval rodině za neutuchající podporu a trpělivost během studia.

Obsah

1. Úvod	9
2. Robotická lakovací pracoviště	10
2.1. Lakování v malosériové výrobě	11
2.2. Použití trackovací kamery T265	11
2.3. Návrh řešení	14
3. Návrh polohovacího stolu	16
3.1. Požadavky	16
3.2. Pohon polohovacího stolu	16
3.3. Velikost a umístění polohovacího stolu	18
4. Řízení polohovacího stolu	19
4.1. Trajektorie pohybu stolu	22
4.1.1. S-křivka	22
4.1. Ruční ovládaní stolu	26
4.2. Zkušební řízení pohonu	26
5. Výběr robotu a lakovacího příslušenství	27
6. Řízení robotu	29
6.1. Transformace souřadnic	29
6.2. Inverzní kinematika	32
6.3. Optimalizace trajektorie	33
6.3.1. Optimální sekvence dílčích trajektorií	33
7. Systémové řešení imitačního lakovacího pracoviště	36
7.1. Deklarace záměru	36
7.2. Odborný článek	36
7.3. Kontextový diagram	37
7.4. Seznam událostí	38
7.5. Návrh HW a SW řešení	38
7.6. Datový model	39
7.7. Uživatelské rozhraní	41
7.8. Testování	41
8. Simulace celého pracoviště	42
8.1. Volba simulačního prostředí	42
8.1.1. PyBullet	42
8.1.2. ROS	43
8.2. Příprava simulace a prostředí	44
8.3. Výsledná simulace	45
9. Ověření přesnosti trackovací kamery Intel RS T265	47
9.1. Proběhlé studie přesnosti	47
9.2. Vlastní ověření	48
9.3. Vyhodnocení	49
10. Metody kalibrace systému	51
Závěr	53

Budoucí práce	54
Značení.....	57
Seznam zkratk	57
Přílohy	58
Příloha A: Měření přesnosti kamery Intel RealSense T265.....	58
Měření 1.....	58
Měření 2.....	60
Měření 3.....	62
Měření 4.....	64
Příloha B: Záznam simulace	67
Příloha C: Zdrojový kód	67

Seznam obrázků

Obrázek 1: Programování lakovacího robotu Epistolio MRK. [2].....	10
Obrázek 2: Blokové schéma vnitřního uspořádání trackovací kamery T265 [3]	12
Obrázek 3: Pohled z trackovací kamery oproti pohledu z běžného fotoaparátu.	12
Obrázek 4: Souřadnicové systémy trackovací kamery Intel [5].....	14
Obrázek 5: Příklad nastavení řídicích smyček v programu Drive.exe. [10]	17
Obrázek 6: Schéma k výpočtům rychlosti a zrychlení polohovacího stolu.....	19
Obrázek 7: Bezpečná oblast rychlosti a zrychlení polohovacího stolu.....	21
Obrázek 8: Typy trajektorií	22
Obrázek 9: Jednotlivé segmenty trajektorie s-křivka	23
Obrázek 10: Průběhy rychlostí v jednotlivých případech.....	24
Obrázek 11: Průběhy polohy, rychlosti, zrychlení a jeho derivace [13]	26
Obrázek 12: Návrh umístění robotu IRB1600 a polohovacího stolu [14].....	27
Obrázek 13: Automatická stříkací pistole s držákem. [15]	28
Obrázek 14: Zobrazení některých souřadnicových systémů na obrázku ze simulace. Je zde dodržena konvence RGB-xyz.	30
Obrázek 15: Příklad odvození výpočtu IK robotu pomocí Symbolic Math toolbox v Matlabu. 32	
Obrázek 16: Matice vzdáleností a matice rozdílů konfigurací	34
Obrázek 17: Sled jednotlivých tahů, postupně zleva: podle původní trajektorie, metodou simulovaného žíhání a metodou nejbližšího bodu.	35
Obrázek 18: Kontextový diagram	37
Obrázek 19: ER diagram	39
Obrázek 20: Diagram datových toků (DFD), úroveň 1	40
Obrázek 21: Stavový diagram systému, úroveň 1	40
Obrázek 22: Stavové diagramy robotického lakování a optimalizace, úroveň 2.....	41
Obrázek 23: Příklad uspořádání uzlů v systému ROS. [22]	43
Obrázek 24: Struktura robotu v URDF. [21]	44
Obrázek 25: Umístění modelů v simulačním prostředí.	45
Obrázek 26: Robot imitující lakýrníka	46
Obrázek 27: Proces ověřování přesnosti kamery.	48
Obrázek 28: Prototyp kalibračního přípravku se stříkací pistolí a jeho umístění v rámci souřadnicového systému stolu.....	51

1. Úvod

Cílem této práce je navrhnout lakovací pracoviště, které s použitím senzorů zaznamená pohyby profesionálního lakýrníka během lakování. Přesně tyto pohyby jsou poté zopakovány robotem při imitačním lakování. Výhodou tohoto řešení je, že není potřeba žádné složité programování lakovacích trajektorií. Stačí pouze jednou vzorově nalakovat požadovaný předmět a všechny další předměty série už jsou nalakovány robotem. Toto řešení výrazně zlepšuje pracovní podmínky lakýrníka, který nemusí vykonávat stále se opakující činnost v nehostinných podmínkách lakovací kabiny a může se soustředit na jiné činnosti.

Výsledný návrh by měl sloužit lakovnám s variabilními lakovacími technologiemi, u nichž je velikost série malá až střední a klasické robotické lakování se lakovnám nevyplatí.

Tato práce navazuje na práci *Robotic painting cell design using worker's motion tracking and imitation* [1], kterou v mnohých směrech doplňuje, ověřuje některé metody záznamu pohybů a dále rozšiřuje. Robotické lakovací pracoviště je doplněno o senzor spouště a polohovací stůl, který má na starost natáčení lakovaného předmětu tak, aby mohl být robotem nalakován ze všech požadovaných stran.

Důležitou součástí práce je převod záznamu pohybů lakýrníka na lakovací trajektorii, kterou je možné lakovacího robota přímo řídit. Na této trajektorii jsou také vyzkoušeny a popsány různé optimalizační metody. Práce se zabývá i systémovým řešením pracoviště a jeho implementací. Funkčnost celého pracoviště je na závěr ověřena v simulaci.

2. Robotická lakovací pracoviště

Lakování bývá závěrečnou povrchovou úpravou. Lze ho rozdělit podle typu nanášeného media. Práškové laky se nanášejí ve formě prášku díky elektrostatickému náboji, po nanesení se prášek „speče“ v pecích. Tato technologie je omezena rozměry pece a teplotou, kterou musí předmět vystát. Výhodou je, že předmět po opuštění pece je připravený k dalšímu zpracování a nemusí schnout, jako je tomu u mokrého lakování. Tento způsob je i šetrnější k životnímu prostředí, protože téměř všechny prášek se díky náboji usadí na předmětu a dále se z něho neuvolňují žádné těkavé látky. Naproti tomu mokré lakování umožňuje vytvořit velmi kvalitní povrch libovolného odstínu.

V malosériových lakovnách se využívají oba způsoby. Kvalitní mokré lakování má vysoké požadavky na zkušenosti lakýrníka, jeho schopnost nastavit lakovací parametry a samotnou barvu. Zároveň je tento způsob lakování pro lakýrníka nehostinný a méně ekologický.

Nejčastější využití robotického lakování je ve formě automatizovaných lakovacích linek. Tvoří je dopravník a jeden nebo více specializovaných průmyslových robotů. Součástí takové linky jsou předúpravné systémy, robotický lakovací box a také sušící nebo vytvrzovací pece v případě práškových barev.

Celý proces včetně lakování je automatizovaný. Tomu ale předchází velmi složitý proces programování lakovacích trajektorií, které jsou generovány z dat CAD modelů lakovaného výrobku nebo na míru sestavené zkušeným programátorem lakovacích robotů. Toto řešení se ale vyplácí pouze u velkých sérií.

Existují ale i způsoby, jak využít roboty k lakování bez potřeby složitého programování trajektorií. Podobnou úlohu jako bude mít navržený systém, tedy replikaci lakýrníkových pohybů robotem, umějí vykonat i lakovací roboti Epistolio MRK. Ty se programují právě samotným lakováním, kdy obsluha požadovanou součást nalakuje stříkací hlavicí umístěnou na konci robotického ramene. Během prvotního lakování pracovník ovládá pohyby robotu pomocí joysticku na koncovém efektoru. Tyto pohyby se potom uloží a v následující fázi robot lakuje samostatně. I u tohoto robotu je samozřejmě možnost sestavit trajektorii lakování z jednoduchých křivek. [2]



Obrázek 1: Programování lakovacího robotu Epistolio MRK. [2]

Výhodou tohoto řešení je začlenění robotu přímo do fáze učení, tím je zaručena dobrá opakovatelnost celého procesu. Ve fázi automatického lakování navíc nehrozí kolize, protože robot vykonává pouze ověřené pohyby z prvotní fáze. Při prvotním lakování jsou ale pohyby lakýrníka v tomto případě omezeny robotickým ramenem. V tom se liší oproti navrhovanému systému, kdy lakýrník může pracovat s ruční stříkací pistolí, jako při běžném ručním lakování. A v případě využití některých dále popsaných optimalizačních metod se sám nemusí snažit nalakovat předmět během zaznamenávání co neefektivněji, ale může se soustředit spíše na kvalitu výsledných tahů.

Zmíněné řešení často nachází uplatnění v nábytkářském průmyslu, kde je velikost série poměrně malá a předměty bývají tvarově jednodušší, například dvířka skříněk.

2.1. Lakování v malosériové výrobě

V malosériové výrobě stále převládá ruční lakování. Zavádění robotických pracovišť se často ekonomicky nevyplatí nebo je jeho zakomponování do výroby příliš náročné. Navíc na chod takové linky je třeba dostatek proškoleného personálu.

Systém imitačního lakování je cílen právě na série přibližně od 20 do 100 kusů, v závislosti na velikosti dílů. V takové míře se většinou nevyplatí programovat a učení robotu předlohovým lakováním může být výrazně jednodušší a rychlejší.

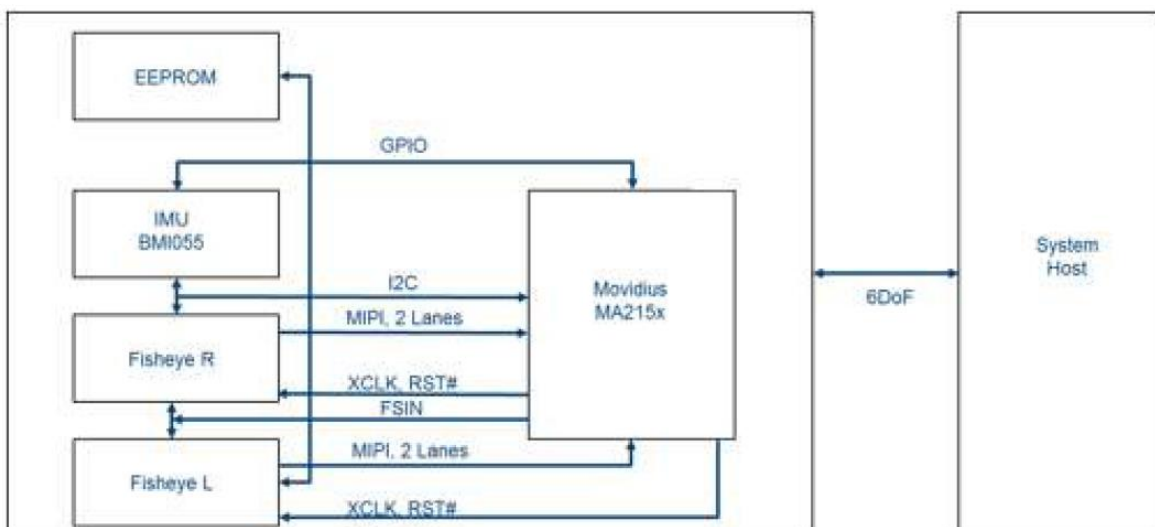
2.2. Použití trackovací kamery T265

Dle návrhu z práce [1] k záznamu trajektorie nejlépe vyšla trackovací kamera T265 od společnosti Intel. Jde o IMU senzor (inertial measurement unit) se dvěma širokoúhlými kamerami, tzv. rybí oko, které slouží ke korekci pozice. Pro zjištění polohy měřením pouze inerciálními senzory se data, konkrétně zrychlení a úhlové rychlosti, postupně integrují. Při tomto procesu vzniká rozdíl mezi odhadovanou skutečnou a polohou. Při delším měření tato chyba významně roste. Proto je senzor doplněn párem kamer, které se snaží tomuto jevu zamezit tím, že mapují vlastní pohyb v prostoru a kameru lokalizují.

Kamera je navržena především pro použití v rozšířené nebo virtuální realitě (AR/VR), v dronech a robotice, kde slouží jako nástroj lokalizace. Dle výrobce je nechtěný posun předpokládané polohy při opakovaných měřeních v různých případech a prostředích menší než 1 %. Využití v AR/VR bylo testováno s kamerou na hlavě ve vnitřních prostorách s běžným osvětlením. Trackování za pomoci kamery bylo také zkoušeno na mobilních robotech v kombinaci s daty z odometrie. Pro správnou funkci je zapotřebí dostatečná viditelnost statických vizuálních předmětů. V případě mlhy, kouře nebo v jiných podmínkách, kdy kamera nebude moci pozorovat referenční body, nebude lokalizace fungovat. Výkon se může v různých podmínkách lišit a případné zhoršení výkonu by měla kamera rozpoznat a oznámit, není to však podmínkou. [3]

Trackovací kamera T265 sestává z několika komponent, které jsou znázorněny na obrázku 2. Klíčovou z nich je IMU senzor BMI055, ten se používá především v robotických aplikacích a systémech s virtuální realitou. Obsahuje tříosý gyroskop se vzorkovací frekvencí 200 Hz, rozlišením 0,004°/s a rozsahem 2000°/s a tříosý akcelerometr se vzorkovací frekvencí 62,5 Hz, rozlišením 0,98 g a rozsahem $\pm 4 \text{ g}^1$. V době psaní této práce je již nahrazen novějšími senzory BMI085 a BMI088 s větším rozlišením nebo rozsahem měřených hodnot. [4]

¹ Uvedené rozsahy jsou použité v kameře T265, programově se však dají změnit, viz. [4]



Obrázek 2: Blokové schéma vnitřního uspořádání trackovací kamery T265 [3]

Dvě černobílé kamery určují název zařízení. Mají rozlišení 848 x 800 a širokoúhlé objektivy se zorným úhlem 173° napříč. V obrázku 3 je ukázka pohledu z kamery v porovnání s pohledem z běžného mobilního telefonu. Procesoru předávají obraz s frekvencí 30 FPS. [3]

Asi největší předností této kamery je integrovaná výpočetní jednotka. Všechna data ze senzorů jsou předávána procesoru Movidius MA215x. Jedná se o VPU (vision processing unit) navržený firmou Intel přímo ke zpracování obrazu. Zpracovaná data jsou pak předávána hostitelskému počítači ve formě pozice a orientace v prostoru, tedy všech šest stupňů volnosti. [3]

V neposlední řadě je to paměť, kam si kamera zjednodušeně ukládá předchozí zaznamenané orientační body, z nichž koriguje odhad polohy.



Obrázek 3: Pohled z trackovací kamery oproti pohledu z běžného fotoaparátu.

V případě použití kamery k vývoji vlastních aplikací je k dispozici soubor softwarových nástrojů pod názvem Intel RealSense SDK 2.0. Nejsnazším způsobem, jakým lze ověřit funkčnost kamery a získat z ní data, je využití programu RealSense Viewer. Program umí vykreslit všechna aktuální data z kamery včetně obrazů. Všechna tato data lze nahrát a následně libovolně přehrávat. Záznam se uloží jako BAG soubor, který se dá využít i v systému ROS.

Dalším jednoduchým způsobem sběru dat z kamery je využití připravených nástrojů, např. *rs-data-collect*. Ten se spouští z příkazového řádku hostitelského počítače spolu s parametry, jako jsou doba nahrávání či počet vzorků a název konfiguračního souboru. V něm lze nastavit parametry požadovaného záznamu, v tomto případě to je pouze záznam polohy. Tento postup byl použit v práci [1]. Nevýhodou je omezená možnost ovládní sběru dat podle potřeb v reálném čase. Například když není dopředu známo, jak dlouho lakování potrvá. [5]

Dalším způsobem ovládní kamery je napsání vlastního skriptu za pomoci jedné z dostupných knihoven pro C++, Python, Matlab nebo případně ROS. [5] V této práci je využita Python knihovna *realsense2* v jednoduchém programu pro záznam trajektorie kamery a stavu spouště stříkací pistole. Program se nachází v příloze C2. Je možné také sbírat data z kamery a ta pak využít k řízení simulace robotu v reálném čase, kdy koncový efektor kopíruje pohyby kamery, což bylo také vyzkoušeno. To dokládá vhodnost využití kamery ve virtuální realitě.

Výstupem z kamery jsou nezpracovaná data z jednotlivých senzorů, data z VPU o poloze zařízení nebo případné snímky z levé a pravé kamery. Všechna data jsou opatřena časovou značkou, tzv. timestamp.

Při spuštění záznamu polohy trackovací kamery T265 se nejprve vytvoří počátek globálního souřadnicového systému, ke kterému jsou pak vztaženy všechny další polohy. Jeho pozice je shodná s pozicí kamery při inicializaci, avšak jeho orientace se liší od orientace kamery. Globální osa y je vždy souběžná s gravitačním zrychlením. Následně se určí osy x a z , kdy globální osa z je promítnutím lokální osy kamery do vodorovné roviny. Osa x je pak dána pravotočivým souřadnicovým systémem. [3]

Jako první bod záznamu trajektorie je tedy zapsán nulový polohový vektor a kvaternion určující orientaci kamery vůči počátku. Kvaterniony se často využívají v počítačové grafice nebo robotice, protože se s nimi z pohledu výpočetní techniky výrazně snadněji pracuje oproti například Eulerovým úhlům. Transformace s nimi jsou jednoznačné a nemůže tedy dojít k takzvanému jevu *gimbal lock*, kdy dojde k zarovnání dvou os rotace. [6] Kvaternion můžeme chápat jako uspořádanou čtveřici čísel. Má tvar:

$$xi + yj + zk + w \tag{2.1}$$

kde w, x, y, z jsou reálná čísla a i, j, k jsou imaginární části. Platí pro ně:

$$\begin{aligned} i^2 = j^2 = k^2 &= -1 \\ ij = -ji &= k \\ jk = -kj &= i \\ ki = -ik &= j \end{aligned} \tag{2.2}$$

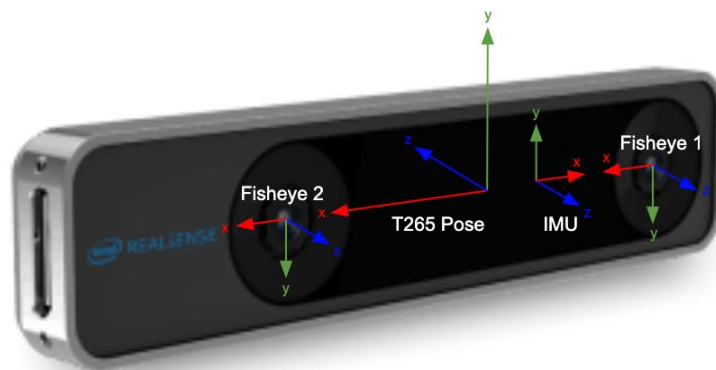
Těmito pravidly se počítač řídí po roznásobení kvaternionů při výpočtu transformací např. v počítačové grafice. Výpočet je jednodušší než například u maticového násobení. Kvaternion v tomto případě popisuje pouze rotaci pravotočivého souřadnicového systému ve 3D prostoru, nikoliv orientaci nebo polohu. Při použití kvaternionu pro popis rotace lze využít tvar:

$$\cos \frac{\phi}{2} + \sin \frac{\phi}{2} (xi + yj + zk) \quad (2.3)$$

Kdy vektor $[x, y, z]$ vyjadřuje osu rotace a ϕ úhel otočení kolem této osy v radiánech. Polovina úhlu je zde proto, aby byla zachována jednotkovost kvaternionu při jeho použití v rotaci bodů, vektorů nebo těles. Nova poloha bodu nebo orientace tělesa vůči jinému se zjistí pomocí transformačního vztahu:

$$p' = qpq^{-1} \quad (2.4)$$

Kde p je bod v prostoru a p' je bod po jeho rotaci. Transformaci výše lze aplikovat i na další kvaternion, kdy výsledek je opět kvaternion, a takto řetězit rotace. [6]



Obrázek 4: Souřadnicové systémy trackovací kamery Intel [5]

Z kamery lze získat i více než pouze data o poloze. Součástí možného streamu mohou být i data z akcelerometru nebo gyroskopu reprezentující 3 stupně volnosti. Hodnoty jsou vztaženy k lokálnímu souřadnicovému systému kamery, viz. Obrázek 4.

2.3. Návrh řešení

Návrh imitačního lakovacího pracoviště je koncipován jako celek skládající se nejen z fyzických objektů, ale především z procesů, které průběh imitačního lakování řídí. První fází je lakování lakýrníkem, které je zaznamenáno. Následuje zpracování těchto dat a případná optimalizace. Ve finále jsou data použita k robotickému imitačnímu lakování.

Trackovací kamera má za cíl sledovat polohu a orientaci lakovací pistole, nebo nástroje obecně, během vzorového lakování. Dále je třeba získat informace o činnosti nástroje, tedy zda je stisknuta spoušť lakovací pistole a barva stříká, nebo ne. Jelikož je využita běžná průmyslová lakovací pistole, je potřeba do ní takové zařízení zakomponovat. K tomu se jeví vhodný Hallův senzor, který sám o sobě má velice malé rozměry (cca 4,2 x 3,3 x 1,5 mm). [7] Naproti němu se ke spoušti přichytí malý dostatečně silný magnet, vhodný je například neodymový magnet.

Toto řešení je bezkontaktní, a proto téměř neovlivňuje síly ve spoušti ani její citlivost. Je však třeba dbát na umístění magnetu v případě, že je tělo lakovací pistole tvořeno slitinou železa, aby nedocházelo k ovlivnění sil na spoušti magnetickou silou. Podmínky pro lakýrníka se při prvotním lakování tímto způsobem řešení nemění. Zároveň je to jednoduché a cenově příznivé řešení.

Součástí lakovacího pracoviště je průmyslový lakovací robot, který má za úkol imitovat pohyby lakýrníka se stříkací pistolí. K tomu, aby zvládl pokrýt celý rozsah pohybů a tím i požadovaný povrch lakovaného předmětu, slouží manipulátor, který daný předmět robotu nastavuje. V tomto případě jím je rotační polohovací stůl se servopohonem.

3. Návrh polohovacího stolu

Hlavní částí této práce je doplnění systému Robopainter z práce [1] o manipulátor, který bude optimálně nastavovat lakovaný předmět lakovacímu robotu tak, aby robot dokázal pokrýt celou požadovanou plochu ze stacionární pozice.

3.1. Požadavky

Manipulátor by měl být přiměřeně robustní s možností přichycení kotvících a kalibračních prvků. Zároveň by neměl být příliš těžký s velkým momentem setrvačnosti, aby při jeho přestavování nedocházelo ke zbytečné ztrátě času nebo energie. Výsledné řešení záleží na typu a velikosti požadovaných lakovaných předmětů.

3.2. Pohon polohovacího stolu

K pohonu pracovního stolu bude využit průmyslový servomotor. K ověření je využit dostupný motor Kollmorgen DDR DH083M². Jedná se o rotační motor určený k přímému pohonu s vysokým kroutícím momentem, tzv. torque motor.

Užitím tohoto typu pohonu odpadá nutnost využití převodovky. S tím souvisí několik výhod. Asi nejvýraznější z nich je přesnost a opakovatelnost polohování. Při použití převodovek je i po úplném zastavení motoru na výstupu z pohonu možný malý pohyb vlivem vůlí v převodech. Použitím přímých pohonů je tento pohyb omezen až na jednu úhlovou vteřinu, uvádí výrobce. [8]

Spojení s převodovkou dále ovlivňuje dobu rozběhu nebo zastavení pohonu a také prodlužuje čas jeho ustálení. V důsledku toho může být omezena frekvenční oblast použití pohonu. Takovýto pohon vyžaduje minimální náklady na údržbu díky minimálnímu počtu dílů, u kterých navíc nedochází k opotřebení, jako například u zubů převodovek, nebo k prodloužení řetězu či řemenu. Zvyšuje se tedy i celková spolehlivost. [8]

Protože je motor napojen na zátěž napřímo a tvoří s ní jednotný moment setrvačnosti, nemusí být momenty setrvačnosti zátěže a motoru párovány, jak se to u průmyslových servopohonů dělá k zamezení vibrací a jiných nežádoucích účinků. [9]

V poslední řadě by měl být provoz přímého pohonu i tišší. [8]

Jedná se o 24 pólový motor, pravděpodobně typu BLDC, princip motoru výrobce neuvádí. Součástí motoru je sinový enkodér EnDat s rozlišením 134 217 728 kroků na otáčku (2²⁷), který poskytuje servozsilovači polohovou zpětnou vazbu. Dokáže dodat konstantní krouticí moment 50,4 Nm a ve špičce až 177 Nm. To je dostatečně velký moment k polohování ocelového stolu o průměru 1 m a tloušťce 2 cm včetně podpůrné konstrukce a lakovaného předmětu s požadovaným zrychlením do 2 rad/s². [8]

Součástí motoru jsou nosná ložiska, která dokážou přenášet síly do rámu. Díky nim je nosnost motoru 3360 kg v axiálním směru a v ohybovém momentu až 53,3 Nm. To znamená, že excentricky uložené těleso o hmotnosti 35 kg motor unese i ve vzdálenosti 15,5 cm od osy rotace.³ V případě použití dvouložiskové varianty motoru je ohybová nosnost výrazně vyšší. Nosnosti jsou dány životností L₁₀ použitých ložisek v motoru. [8]

Maximální rychlost dostupného motoru je 500 ot/min, v případě polohovacího stolu ovšem bude dostačovat rychlost do 20 ot/min. [8]

² DDR – direct drive rotary

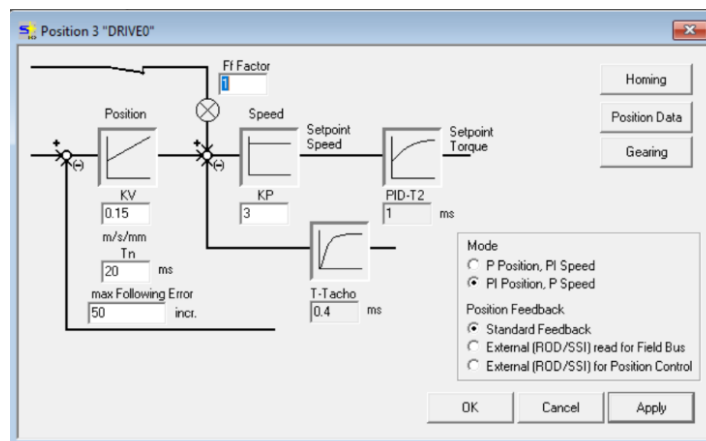
³ Míněna vzdálenost k těžišti tělesa.

Součástí servopohonu je zesilovač Kollmorgen Servostar s600. Ten se stará o veškeré silové řízení motoru. Je napájen třífázovým napětím pro pohon a 24 V je pak napájen řídicí obvod.

Zesilovač v sobě zahrnuje třismyčkové zpětnovazební řízení, tím reguluje výstupní proud, který je úměrný momentu na motoru. Regulační smyčky tvoří PI regulátor a frekvenční filtry. Tyto regulační parametry je třeba nastavit pro konkrétní aplikaci. Závisí především na momentu setrvačnosti stolu a předpokládané zátěži. Některé parametry k řízení daného motoru jsou dostupné přímo v aplikaci Drive.exe, která slouží k nastavení pohonu, nebo jsou dostupné v databázi mbase140.csv. Jedná se například o parametry indukce, proudové charakteristiky atd. [10]

Vstupem pro řízení může být žádaná hodnota polohy, rychlosti nebo momentu, ale i přednastavený pohyb, který se pouze externě spustí. Další možností je motor řídit pulsy podobně jako u krokových motorů, kdy se do ovladače posílají pulzy a směr v závislosti na požadované poloze a rychlosti. Tímto způsobem lze nechat motor sledovat libovolnou trajektorii v reálném čase. Tato funkce je nazvaná „*electronic gearing*“, protože s ní lze synchronně pohybovat několika motory zároveň, kdy je ovládán pouze první motor a ostatní motory jsou napojeny na enkodér toho prvního. Tak se všechny motory pohybují jako ten první, případně v měřítku, pokud je zvoleno jiné rozlišení. [11]

K ovládaní je zvolena tato varianta, protože tak je docíleno pohybu stolu po vlastní trajektorii, pokud je regulace dobře nastavena.



Obrázek 5: Příklad nastavení řídicích smyček v programu Drive.exe. [10]

Servozesilovač umožňuje komunikovat různými způsoby pomocí několika sběrnic. Základní způsob je přes sériovou linku s rozhraním RS-232. Tato linka slouží především k nastavení po propojení s počítačem. Nastavení lze snadno provést v programu Drive.exe s uživatelským rozhraním. Je zde ale možnost využití sériové linky k přímé komunikaci se servozesilovačem pomocí ASCII příkazů. Jimi je možné číst aktuální data, nastavit parametry řízení nebo posílat pohybové příkazy, např. „PFB“, „MOVE“, „INPOS“ apod. [12]

Další možností je využít některou z průmyslových sběrnic: CAN, ProfiBus nebo Ethercat. Těmi lze například ovládat několik pohonů najednou, posílat jimi všechny příkazy dostupné pro sériovou komunikaci nebo posílat požadovanou polohu v reálném čase. [11]

V této práci je navrženo řízení pohonu, které využívá vstup X5, ten může sloužit jako výstup emulovaného enkodéru, nebo pak vstup ve funkci *electronic gearing*. Funguje na 5V rozhraní a

přijímá dva signály po dvou párech vodičů. Jeden udává krok a druhý směr. O převodu trajektorie na pulzy je zmíněno níže v podkapitole 4.1.1.

Pro komunikaci především s PLC jsou zde 24 V digitální vstupy a výstupy. Jejich funkce jsou nastavitelné. Může jimi být pohon uvolněn, dán do pohybu nebo zastaven. Při použití koncových snímačů jsou snímače napojeny právě na tyto vstupy. Také je možné tyto vstupy využít k řízení pomocí pulzů, mají ale nižší frekvenci. Servozesilovač má i dva analogové vstupy a jeden výstup. Jsou také nastavitelné a je jimi například možné přímo řídit rychlost nebo moment pohonu pomocí analogového signálu. [11]

3.3. Velikost a umístění polohovacího stolu

Ideální polohovací stůl je takový, který stabilně drží lakovaný předmět a v případě potřeby ho rychle přestaví. Proto by samotný manipulátor měl být co nejlehčí a zároveň dostatečně pevný. V případě lehčích dobře umístěných předmětů postačí konstrukce namontovaná přímo na motor. Podmínky zatěžování dostupného motoru jsou vypsány v podkapitole 3.2. Z nich vyplývá maximální hmotnost polohovacího stolu spolu s lakovaným předmětem v případě, že je polohovací stůl uchycen k motoru přímo. V opačném případě je potřeba využít podporu polohovacího stolu ve formě axiálních ložisek. Pohon v tomto případě musí být propojen se stolem pomocí spojky, protože není vhodné napřímo propojovat pohon se stolem, pokud mají svá ložiska. [8] Toto řešení může ale dle typu spojky vnášet vůli mezi stůl a enkodér motoru a tím způsobovat mírnou, možná zanedbatelnou nepřesnost. Spojka musí být dostatečně tuhá, aby zaručila vysokou opakovatelnost polohování, která je u imitačního lakování klíčová. Přesnost polohování není oproti opakovatelnosti tak podstatná. Další možností je odečítat polohu polohovacího stolu napřímo.

Manipulátor může mít více podob, zde jsou nastíněné dvě varianty:

1) Vyvýšený polohovací stůl

Vyvýšený stůl usnadní lakýrníkovi práci tím, že bude moci provádět práci a případný záznam trajektorie lakování ve vzpřímené poloze. Takové pracoviště může obsluhovat menší průmyslový nebo dokonce kolaborativní robot.

2) Otočný manipulátor zapuštěný do podlahy

Pracovní plocha manipulátoru je v úrovni okolí. Lakový předmět může být výrazně větší, avšak nesmí přesáhnout výškové možnosti lakýrníka. Tomu bude odpovídat i velikost robotu s dostatečným rozsahem pohybů.

U obou variant může lakýrník předmět lakovat ze stacionární pozice a nechávat si lakovaný předmět natáčet, nebo jej může během záznamu trajektorie obcházet. Pracoviště je sdílené robotem i člověkem, proto musí být kladen velký důraz na bezpečnost. Během pohybů člověka v prostoru například při vzorovém lakování musí být robot nastaven do bezpečné polohy a veškerý jeho pohyb blokován.

Jedním z možných umístění robotu a stanoviště lakýrníka je naproti sobě. Takto je zajištěna velká volnost pohybů lakýrníka. Nevýhodou ale může být zbytečné zanášení robotu barvou při ručním lakování. Proto může být vhodnější umístit stanoviště lakýrníka a základnu robotu okolo stolu pod úhlem 90°. Naproti nim pak zbyde prostor i pro vodní stěnu.

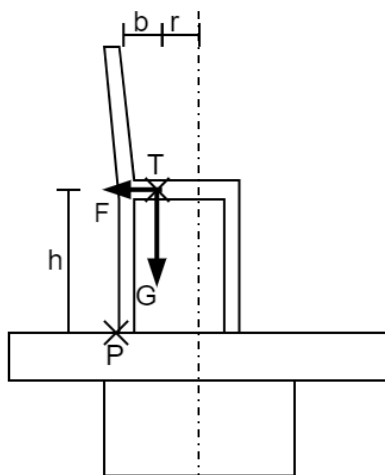
Práce je dále zaměřena na variantu s vyvýšeným polohovacím stolem. Návrh konkrétního umístění robotu vůči polohovacímu stolu je vidět na obrázku 12 v kapitole 5.

4. Řízení polohovacího stolu

Hlavní funkcí stolu je otočit se o zadaný úhel, aby byl lakovaný předmět vhodně natočen vůči robotu. Řídicí systém stolu získá úhel od algoritmu pro lakování. Cílem je potom natočit stůl v co nejkratším čase, a to bez rizika přesunutí, pootočení lakovaného předmětu vůči pracovnímu stolu nebo jeho převrácení. Předpokládá se, že pro lakovaný předmět není k dispozici jeho model, tudíž ani jeho statické vlastnosti, které se hodí pro určení maximálních možných rychlostí a zrychlení manipulátoru. Proto je snaha tyto vlastnosti nahradit co nejmenším počtem parametrů, které jsou snadno odhadnutelné nebo experimentálně určitelné a poslouží k určení optimální bezpečné trajektorie manipulátoru.

Z výše uvedených podmínek vyplývá, že je třeba stanovit maximální rychlost a zrychlení manipulátoru. Ty mají vliv na celkové zrychlení působící na lakovaný předmět, které ho může posunout nebo převrhnout. V opačném případě by bylo třeba využít například upínky nebo jiné fixace předmětu k manipulátoru. Podmínky pohybu jsou stanoveny ze tří možných případů:

- 1) Převrácení



Obrázek 6: Schéma k výpočtům rychlosti a zrychlení polohovacího stolu

K převrácení může dojít, pokud je výslednice sil působících v těžišti směřuje mimo podstavu lakovaného předmětu. Poměr sil vyvolaných od otáčením a gravitační silou tedy musí být menší než poměr vodorovné vzdálenosti těžiště od hrany možného převrácení b a výšky těžiště tělesa h :

$$\frac{F}{G} < \frac{b}{h} \quad (4.1)$$

Z toho po dosazení a nahrazení poměru vyplývá podmínka pro nejvyšší povolené zrychlení v těžišti tělesa:

$$\frac{F}{G} = \frac{m \cdot a_{max}}{m \cdot g} < \frac{b}{h} \quad (4.2)$$

$$a_{max} < \frac{b}{h} \cdot g = K \cdot g \quad (4.3)$$

Z rovnice rovnováhy vyplývá podmínka (4.3), aby nedošlo k převrácení tělesa. Poměr $\frac{b}{h}$ je nahrazen konstantou K .

- 2) K posunutí dojde, pokud síly vyvolané rotací překročí třecí sílu mezi manipulátorem a lakovaným předmětem:

$$F = m \cdot a_{max} < T = \mu_0 \cdot m \cdot g \quad (4.4)$$

Z toho opět vychází nejvyšší dovolené zrychlení pro tento případ:

$$a_{max} < \mu_0 \cdot g \quad (4.5)$$

- 3) Posledním případem je možné pootočení tělesa vůči pracovní desce stolu. Ten nastane ve chvíli, kdy setrvačné síly, přesněji moment, tělesa překročí třecí moment. Následně dojde k prokluzu. Třecí moment vychází po odvození pomocí integrace přes plochu za předpokladu rovnoměrného rozložení tlaku na kontaktní plochy takto:

$$M_T = \mu_0 mg \quad (4.6)$$

Pootočení tělesa vůči pracovní desce stolu nastane ve chvíli, kdy třecí moment mezi stolem a lakovaným předmětem nebude dostačovat a úhlové zrychlení překročí mezní hranici. Z toho vyplývá podmínka (4.7), posléze (x.x), pro úhlové zrychlení:

$$M_t > I_o \alpha \quad (4.7)$$

Dále je zadefinována pomocná konstanta R , která představuje poloměr setrvačnosti. Je to poloměr obruče, která má stejný moment setrvačnosti jako otáčený předmět. Pokud je R odhadnuto jako poloměr předmětu, pak je bezpečnost velmi vysoká.

$$I_o = mR^2 \quad (4.8)$$

Úhlové zrychlení musí splňovat podmínku:

$$\alpha < \frac{\mu_0 g}{R} \quad (4.9)$$

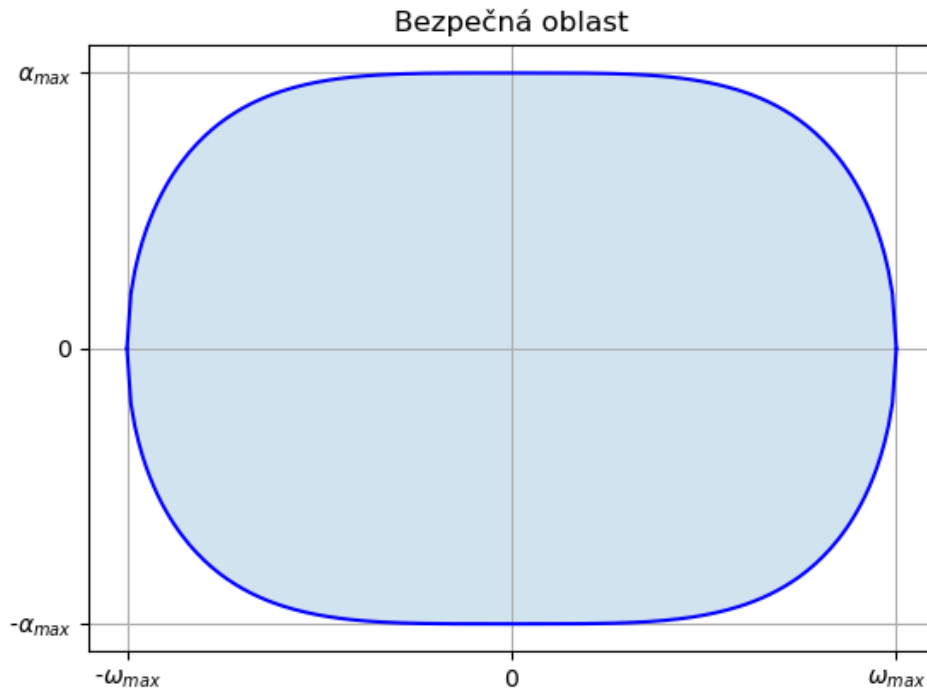
Pro případ 1 a 2 mají vliv rychlost a zrychlení stolu, jelikož rovnice (4.3) a (4.5) mají podobný tvar, budu podmínku maximálního dovoleného zrychlení řešit pouze pro náchylnější z nich. Poměr b/h nebo součinitel tření μ nahradím koeficientem K . Nejvyšší dovolené zrychlení je složeno z tečné a radiální složky, můžu ho tedy rozepsat jako součet funkcí uhlové rychlosti a zrychlení:

$$a_{max} = \sqrt{a_n^2 + a_t^2} = \sqrt{\omega^4 r_T^2 + \alpha^2 r_T^2} \quad (4.10)$$

Dále po úpravě a dosazení vztahu (4.3), případně (4.5), získám následující tvar: (4.11)

$$\omega^4 + \alpha^2 < \left(K \frac{g}{r_t}\right)^2 \quad (4.11)$$

Rychlost a zrychlení polohovacího stolu by se tedy měla pohybovat v rozmezí znázorněné níže.



Obrázek 7: Bezpečná oblast rychlosti a zrychlení polohovacího stolu

Pro ukázkou je uveden výpočet parametrů pro bezpečné polohování lakované židle. Je umístěna excentricky na stole, hrozí její překlacení nebo posunutí. Těžiště má ve výšce $h = 50 \text{ cm}$ a k nejbližší hraně je to vodorovně $b = 15 \text{ cm}$. Těžiště může být až $r = 20 \text{ cm}$ od osy otáčení. Koeficient tření je 0,2. Poloměr setrvačnosti je odhadnut na 30 cm. Výpočet je následující:

$$\frac{b}{h} = \frac{15}{50} = 0,3 > \mu = 0,2 \quad (4.12)$$

Pravděpodobnější tedy je, že se židle posune než, že se překloupí. Další postup je dán vztahem (4.5):

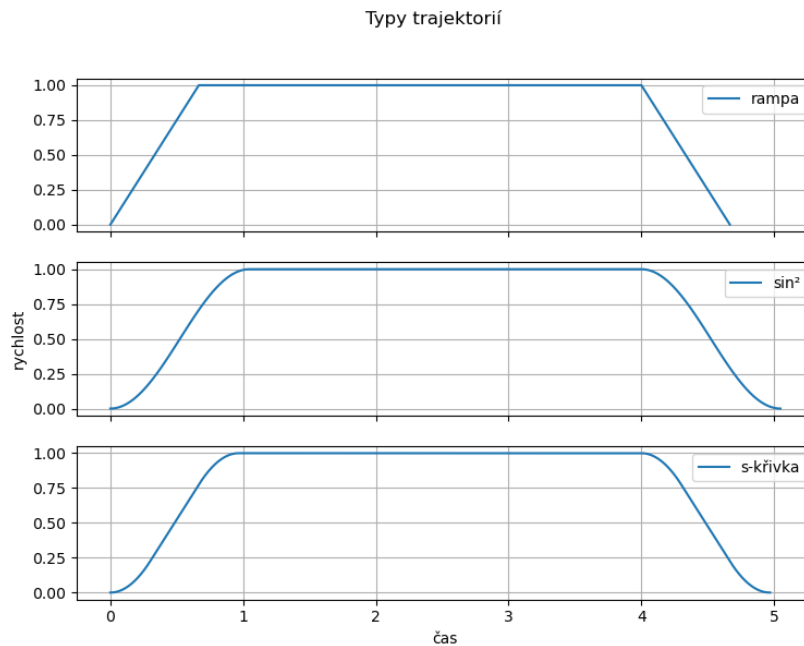
$$a_{max\ 1,2} = 0,2g = 1,96 \frac{m}{s^2} \quad (4.13)$$

Takto je získáno maximální zrychlení, které je využito k výpočtu trajektorie.

4.1. Trajektorie pohybu stolu

Polohovací stůl je řízen v tzv. *position control* módu, tedy polohově. O řízení rychlosti a proudu, který je úměrný síle, se stará servozesilovač. Z předchozích úvah vyplývá nejvyšší možná rychlost a zrychlení stolu. Zde jsou představeny tři možnosti, jak dosáhnout plynulého náběhu rychlosti a nepřesáhnout dovolené zrychlení.

1. Rampa
2. \sin^2
3. S-křivka



Obrázek 8: Typy trajektorií

První možnost je nejjednodušší, rychlost lineárně roste i klesá. Požadované polohy se dosáhne v nejkratším čase. Může se ale projevit trhavý pohyb v důsledku nespojitého zrychlení. U rotačních pohybů je nevýhodou spojení úhlového a dostředivého zrychlení v bodě dosažení požadované rychlosti.

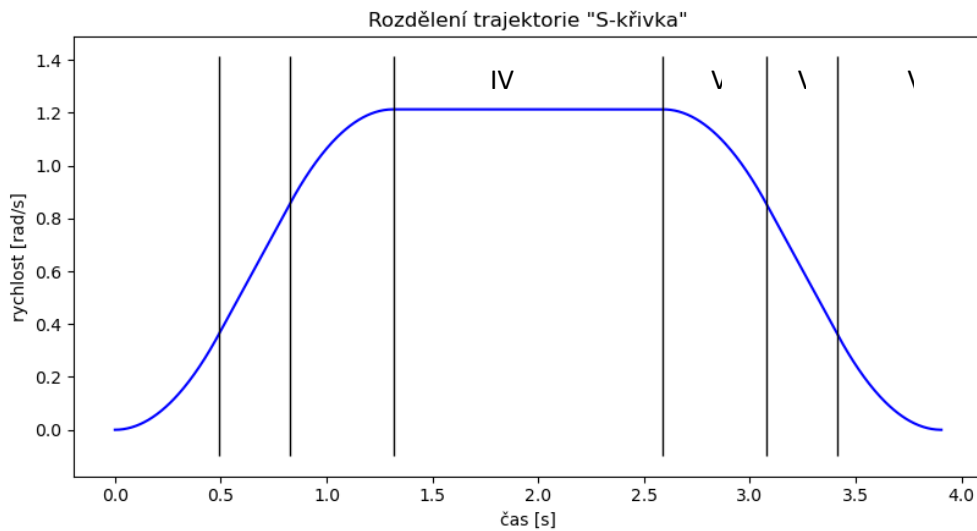
Náběh rychlosti 2. varianty tvoří funkce \sin^2 . Pohyb je plynulejší díky spojitému zrychlení. Výpočet je poměrně jednoduchý, a proto bývá alternativou k lichoběžníkovému průběhu.

Poslední variantou je tzv. s-křivka. Má také spojitě zrychlení a volitelný parametr jerk (derivace zrychlení), který ovlivňuje plynulost pohybu. S rostoucí hodnotou jerk se trajektorie podobá lichoběžníkové trajektorii a celková doba pohybu se zkracuje. Naopak s nízkou hodnotou se trajektorie chová líně. Tento typ trajektorie byl vybrán pro řízení polohovacího stolu.

4.1.1. S-křivka

S-křivka je z výše uvedených trajektorií nejnáročnější na výpočet, ale díky volitelnému parametru jerk umožňuje krátkou dobu přestavení a zároveň plynulost ve formě spojitěho zrychlení. Tato trajektorie bude využita při polohování lakovaného předmětu během fáze robotického lakování. Trajektorii lze rozdělit do 7 segmentů:

- I. Rozběh do maximálního zrychlení
- II. Zrychlování
- III. Doběh k maximální rychlosti
- IV. Pohyb konstantní rychlostí
- V. Počátek zpomalování
- VI. Konstantní zpomalování
- VII. Dojezd



Obrázek 9: Jednotlivé segmenty trajektorie s-křivka.

Trajektorii lze snadno řešit i numericky, výpočet je poměrně jednoduchý, kdy průběh zrychlení, rychlosti a polohy se získá postupnou diskretní integrací. Problémy nastávají, když je časový krok dt příliš velký a trajektorie požadovaný bod přejede, nebo k němu ani nedojede. Též je třeba si dávat pozor na krátké trajektorie, kdy se rychlost nebo zrychlení nestihnou dostat na nejvyšší dovolenou hodnotu. Takové případy je třeba řešit podmínkami. Pro přesnost, rychlost výpočtu a znalost trajektorie ve všech bodech je řešena analyticky. Pro další fázi řízení jsou důležité změny polohy v časových krocích. Výpočet je tedy zaměřen na průběh rychlosti, která je následně převedena na pulzy.

Mohou nastat tři případy průběhu trajektorie:

- I. Zrychlení začne klesat dříve, než dosáhne svého limitu:

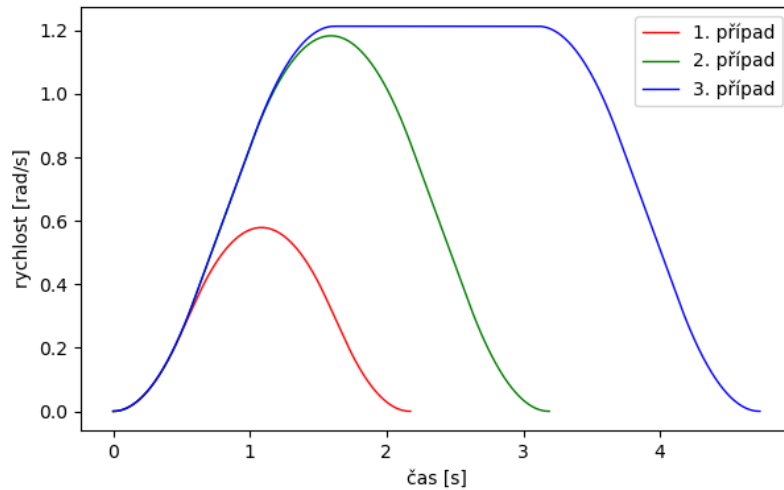
$$a_{max} < a_{lim} \wedge v_{max} < v_{lim} \quad (4.14)$$

- II. Trajektorie začne zpomalovat dříve, než dosáhne maximální dovolené rychlosti:

$$v_{max} < v_{lim} \quad (4.15)$$

- III. Rychlost i zrychlení dosáhnou svých limitů a stůl se ve střední části trajektorie pohybuje konstantní dovolenou rychlostí:

$$a_{max} = a_{lim} \wedge v_{max} = v_{lim} \quad (4.16)$$



Obrázek 10: Průběhy rychlostí v jednotlivých případech.

Pro zjištění, který z výše uvedených případů nastane, je třeba znát časy jednotlivých úseků. Časy úseků 1, 3, 5 a 7 se budou rovnat za předpokladu stejně velkých změn zrychlení.

$$T_1 = T_3 = T_5 = T_7 \quad (4.17)$$

Dále se budou rovnat doby s konstantním zrychlením.

$$T_2 = T_6 \quad (4.18)$$

Který z těchto případů nastane, záleží na požadované míře otočení stolu. Nejprve je odvozena rovnice pro celkovou dráhu třetího případu, který v sobě obsahuje i dráhy prvního a druhého případu.

$$S_{III} = 2S_1 + 2S_2 + 2S_3 + S_4 \quad (4.19)$$

Jednotlivé dráhy byly odvozeny kombinovaně grafickou metodou a postupnou integrací zrychlení, popřípadě rychlosti. Celková dráha vypadá po rozepsání takto:

$$S_{III} = 2j \frac{T_1^3}{6} + 2 \left(v_1 T_2 + a_{max} \frac{T_2^2}{2} \right) + 2 \left(v_{max} T_3 - j \frac{T_3^3}{6} \right) + v_{max} T_4 \quad (4.20)$$

Aby nastal první případ, musely by se vynechat úseky 2, 4 a 6. Zbydou tedy pouze úseky se změnou zrychlení, jejichž doby jsou shodné. Dráha bude kratší nebo rovna dráze s trajektorií dosahující maximálního zrychlení.

$$S_I \leq 2S_1 + 2S_3 = 2j \frac{T_1^3}{6} + 2 \left(v_{max} T_3 - j \frac{T_3^3}{6} \right) \quad (4.21)$$

Příčemž platí:

$$T_1 = T_3 = \frac{a_{lim}}{j} \quad (4.22)$$

$$v_{max} = 2v_1 = 2j \frac{T_1^2}{2} \quad (4.23)$$

Po dosažení se lze dostat k podmínce pro 1. případ:

$$S_I \leq 2 \frac{a_{lim}^3}{j^2} \quad (4.24)$$

Obdobně se lze dopracovat k podmínce pro 2. případ: Kdy je vynechán úsek 4 a časy úseků 2 a 6 se rovnají.

$$S_{II} \leq v_{lim} \left(\frac{a_{lim}}{j} + \frac{v_{max}}{j} \right) \quad (4.25)$$

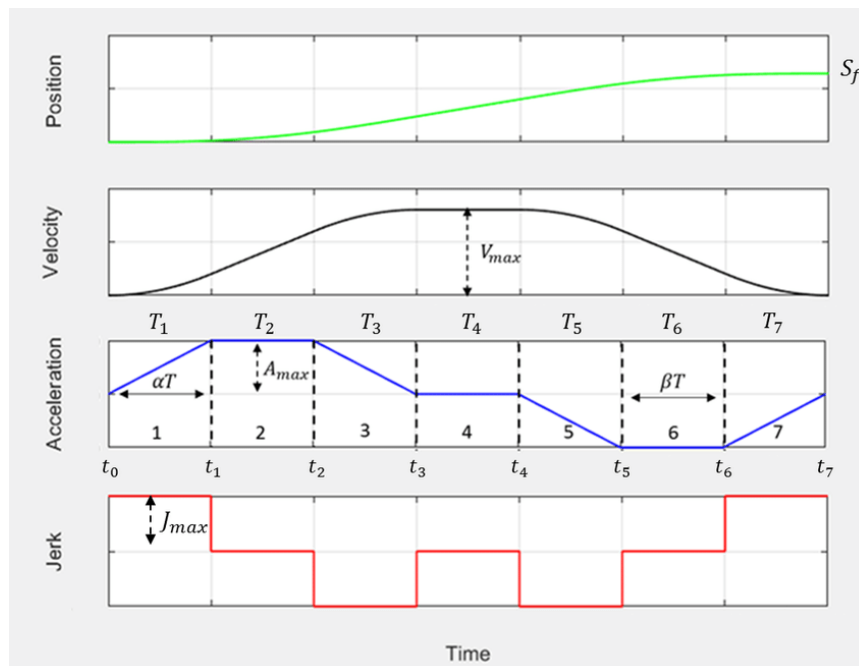
Pokud není splněna ani jedna z podmínek výše, jedná se o 3. případ. Poslední třetí případ už budou všechny zbylé situace. Podle následující tabulky jsou vypočítány průběhy rychlosti.

Fáze	I	II	III	IV	V	VI	VII
Dráha, doba	S_1, T_1	S_2, T_2	S_3, T_3	S_4, T_4	S_5, T_5	S_6, T_6	S_7, T_7
Jerk	j	0	$-j$	0	$-j$	0	j
Zrychlení	jt	jt_1	$\frac{jt_1 - j(t - t_2)}{2}$	0	$-j(t - t_4)$	$-jt_1$	$\frac{-jt_1 + j(t - t_6)}{2}$
Rychlost	$\frac{jt^2}{2}$	$v_1 + jt_1(t - t_1)$	$\frac{v_{max} - j(t_3 - t)^2}{2}$	v_{max}	$\frac{v_{max} - j(t - t_4)^2}{2}$	$v_5 - jt_1(t - t_5)$	$\frac{j(t_7 - t)^2}{2}$

Tabulka 1: Výpočet rychlosti trajektorie

Dále je potřeba převést trajektorii na pulzy, kterými je možné řídit pohon polohovacího stolu. Analytickým řešením je vypočítán průběh rychlosti, stačí jednotlivé vzorky vynásobit časovým krokem dt a tím získat vzdálenost mezi dvěma vzorky. Ta je pak vydělena délkou jednoho řídicího pulzu. Takto získám potřebný počet pulzů v daném časovém kroku. Je zřejmé, že nevyjde celé číslo, proto je každý vzorek zaokrouhlen na nejbližší celočíselný počet pulzů a zbytek, může být i záporný, se převede do dalšího časového kroku.

Kód k výpočtu celé trajektorie včetně převodu na pulzy je v příloze C1 i C4 v programu `trajectory_class.py`.



Obrázek 11: Průběhy polohy, rychlosti, zrychlení a jeho derivace [13]

4.1. Ruční ovládaní stolu

Manipulátor bude potřeba ovládat i ručně, když zrovna nebude v provozu automatického lakování. A to nízkou a bezpečnou rychlostí. Proto bude vybaven párem ovládacích pedálů, popřípadě tlačítky. Jeden pro pohyb vlevo a druhý vpravo. Jednoduchá lichoběžníková trajektorie s lineárními náběhy rychlostí bude postačovat. Jak pro jednodušší výpočet trajektorie v reálném čase, tak pro větší srozumitelnost při ovládaní. Pokud lakýrník sešlápne levý pedál, stůl se začne rozjíždět vlevo a pojede konstantní rychlostí, dokud ho lakýrník nepustí, pak stůl začne plynule zpomalovat. Stejně to bude se směrem vpravo. V případě, že bude třeba náhle změnit směr, stůl plynule zpomalí do nulové rychlosti a opět se plynule rozjede v opačném směru.

Maximální zrychlení je dáno parametry určenými stejně jako v předchozí sekci. Maximální rychlost je volitelná podle potřeb obsluhy. S nižší rychlostí je snazší dosáhnout přesné polohy, ale za cenu delší doby manipulace.

4.2. Zkušební řízení pohonu

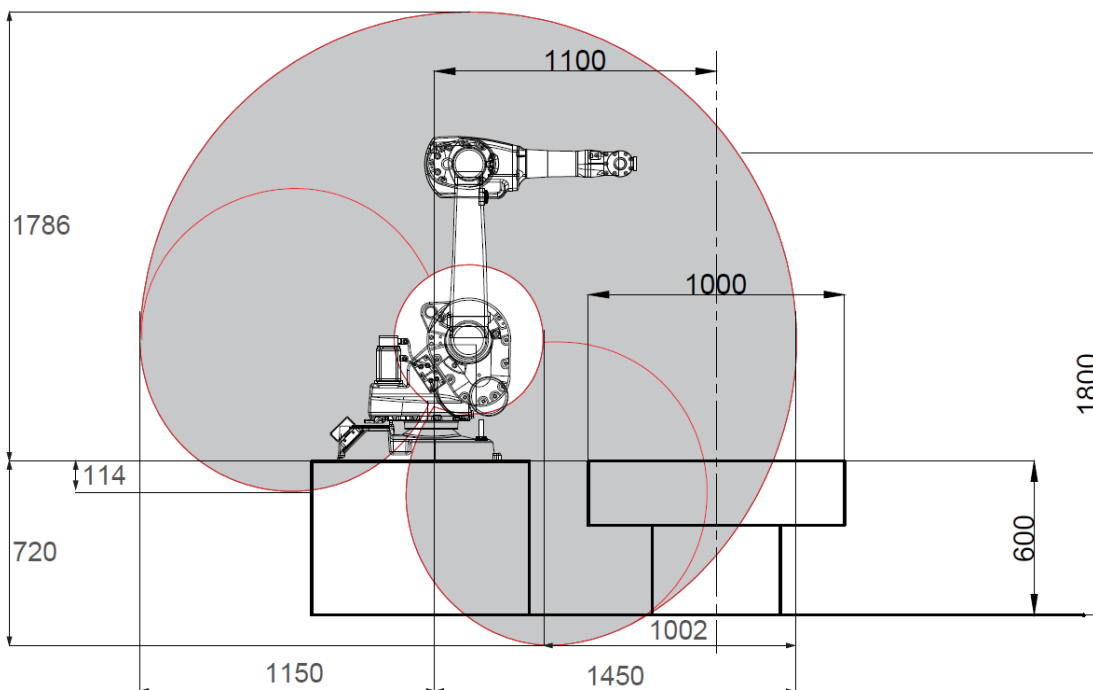
Výše popsaná trajektorie byla aplikována k řízení zmíněného pohonu. Trajektorie byla generována na jednodeskovém počítači Raspberry Pi 4B a následně pomocí knihovny pigpio byla převedena na sekvenci pulzů, které motor ovládaly. Dva signály, krok a směr, byly vyvedeny z GPIO pinů počítače přes pár převodníků RS-485 do servozsilovače. Servozsilovač přijímá kroky v rozlišení od 256 do 65536 kroků na otáčku (2^8 - 2^{16}). Kvůli hardwarovým omezením bylo použito rozlišení 8192 na otáčku. Jemnější krokování by mělo vést k plynulejšímu pohybu. Toto uspořádání dostačuje k ověření systému řízení pohonu, ale protože se nejedná o real-time systém, není zaručeno přesné dodržení trajektorie. Kód ke generování signálu se nachází v příloze C1.

5. Výběr robotu a lakovacího příslušenství

Výběr robotu je v zásadě daný velikostí, případně složitostí tvaru, požadovaného lakovaného dílu. V této práci se zabývám lakováním předmětu dosažitelného člověkem ze stacionární pozice. Pro případ vyvýšeného pracovního stolu byl vybrán řad menších robotů

Pro robotické lakování se využívají speciální lakovací roboty, které jsou odlehčené a přizpůsobené lakování. Často mívají hadice se stříkaným médiem vedené vnitřní částí robotického ramene. Také se využívají univerzální roboty s podobnými vlastnostmi. Těmi jsou nízká hmotnost ramene a dostatečná rychlost pohybů. Zpravidla mívají menší nosnost. Dále pak je důležitá odolnost vůči kapalinám, případně možnost aplikace ochranné folie.

Byl vybrán robot ABB IRB1600-6/1.45, rozsahem pohybů přibližně odpovídá rozsahu pohybů člověka při lakování ze stacionární pozice. Na obrázku níže je vidět manévrovací prostor vybraného robotu a jeho umístění vůči polohovacímu stolu. Ve stejném obrázku kóta vpravo představuje přibližnou výšku lakýrníka. Stejné rozmístění robotu a polohovacího stolu je aplikováno i v simulaci popsané v kapitole 8.2. Výhoda robotů ABB je dostupná dokumentace a pro většinu robotů jsou dostupné modely, které se dají využít například v simulacích.

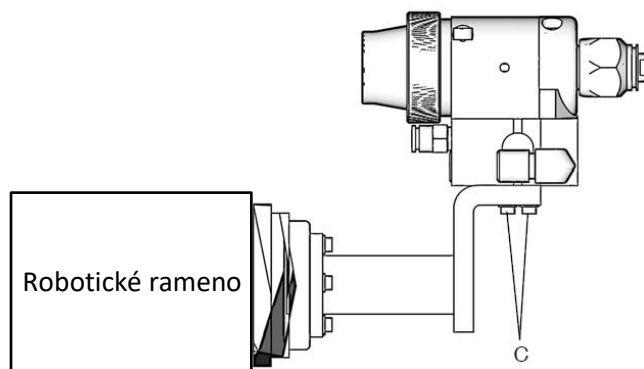


Obrázek 12: Návrh umístění robotu IRB1600 a polohovacího stolu [14]

Konkrétní model má nosnost 6 kg a dosah 1,45 m. Opakovatelnost robotu je při pohybu po požadované trajektorii 0,19 mm. Klidová opakovatelnost je 0,02 mm, ale ta není u lakování podstatná. Vybraný model bude doplněn dodatečnou ochranou FoundryPlus 2 (IP67), která robotu zajistí odolnost vůči podmínkám v lakovacím boxu. [14]

Důležitou součástí lakovacího robotu je lakovací hlavice. Předlohou lakovací pistole k tomuto projektu byla pistole Graco XTR-5, jedná se o stříkáčskou pistoli na bezvzduchovou technologii. Nanášené medium je tedy nanášeno pouze tlakem, nikoli proudem vzduchu. Proto byla vybrána

automatická stříkácí pistole s odpovídajícími parametry. Jde o model 288048 od stejného výrobce Graco k bezvzduchovému stříkání tlakem tekutiny do 35 MPa. [15]



Obrázek 13: Automatická stříkácí pistole s držákem. [15]

6. Řízení robotu

Polohové řízení robotického ramene je vykonáváno samostatnou řídicí jednotkou od výrobce robotu. Úkolem systému Robo-Painter je předat polohová data (trajektorii) s případnými omezeními s cílem zamezit kolizím.

6.1. Transformace souřadnic

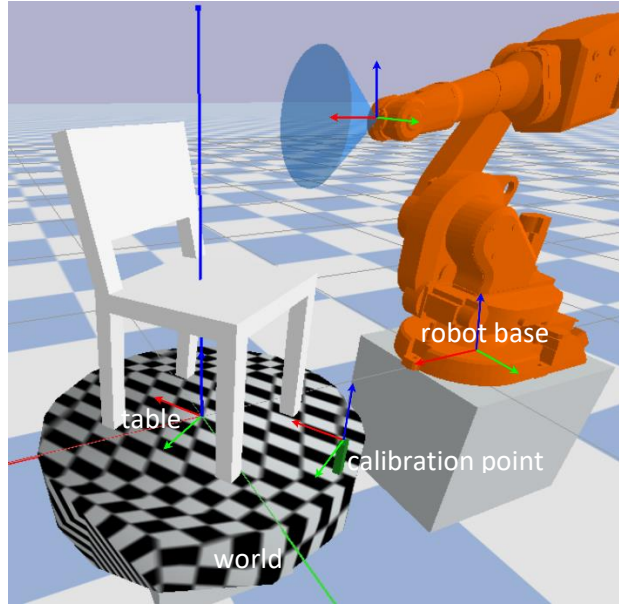
Základem fungování celého systému je komunikace a správná interpretace dat. Systém imitačního lakování je složen z několika podsystémů, které spolu musí spolehlivě fungovat. Cílem této transformace je přenést získaná data ze záznamu lakování k základně robotu, odkud si s nimi již poradí řídicí systém robotu.

Nejprve se celá trajektorie včetně natočení zaznamená trackovací kamerou. Výstupní data z kamery se přes kalibrační bod přetransformují do souřadnicového systému stolu. V závislosti na metodě nahrávání trajektorie, tzn. zda lakýrník lakovaný předmět obcházel nebo si ho nechal polohovacím stolem natáčet, se data rozdělí podle příslušného úhlu, ze kterého má být předmět robotem lakován. Polohová data se orotují o daný úhel kolem počátku stolu a následně se přes základnu a celý kinematický řetězec robotu dostanou do koncového efektoru nebo do lakovací hlavice.

Data vztahená k základně robotu, tedy v jeho souřadnicovém systému, jsou zpracována řídicím systémem robotu, který provede výpočty inverzní kinematiky, popřípadě i dynamiky celé požadované trajektorie.

K vzájemnému popisu poloh má každý člen v kinematickém řetězci od trackovací kamery po robot svůj vlastní souřadnicový systém:

- **Kamery (camera local – cl)**, je lokální s.s. samotné kamery. Jeho poloha v rámci fyzického objektu je popsána v kapitole X.
- **Počátku záznamu trajektorie (camera global – cg)** je s.s., který se vytvoří spuštěním záznamu trajektorie a ke kterému jsou následně vztaheny všechny uložené body trajektorie. Jeho umístění je také popsáno v kapitole 2.2.
- **Kalibračního přípravku (cal)** je umístěn na přesně definované poloze vůči polohovacímu stolu a slouží ke kalibraci trajektorie a s.s. stolu. Jeho praktická užití je znázorněno v kapitole 10.
- **Polohovacího stolu (table)** je s.s., na který je umísťován lakovaný předmět. Všechny další lakované předměty budou během imitačního lakování umísťovány do stejné pozice, jako byl umístěn předmět při vzorovém lakování lakýrníkem. Tento s.s. se natáčí vůči pracovišti stejně s polohovacím stolem.
- **Pracoviště (world)** je s.s. lakovacího boxu nebo pracoviště obecně. Z pohledu kinematiky se jedná o s.s. rámu, k němu je možné vztáhnout všechny body v systému. Jeho umístění je libovolné. Nejjednodušší je jeho umístění pod s.s. stolu.
- **Základna robotu (robot base – rb)** je s.s., od kterého se odvíjí všechny pohyby a kinematické řetězce lakovacího robotu. Je pevně daný vůči s.s. pracoviště.
- **Lakovací pistole (spray tip – st)** je s.s. s počátkem v trysce. Je to cílový s.s. trackování.



Obrázek 14: Zobrazení některých souřadnicových systémů na obrázku ze simulace. Je zde dodržena konvence RGB-xyz.

Souřadnicový systém lakovaného předmětu je ze seznamu vyjat, protože je jeho tvar i umístění neznámé a v tomto systému imitačního lakování je nepodstatný. Předpokládá se, že každý další lakovaný předmět bude umísťován na stejné místo jako předmět vzorově lakovaný. V opačném případě by bylo třeba zavést i jemu souřadnicový systém a umístit jej do souřadnicového systému stolu. Poté by bylo možné lakovat například více dílů najednou během jednoho imitačního cyklu.

K popisu transformací mezi jednotlivými s.s. jsou využity transformační matice. Jejich předností je výstižný popis polohy v prostoru, zahrnuje totiž v jednom operátoru translaci i rotaci jednoho s.s. vůči druhému. Dále je práce s nimi i výpočet poměrně intuitivní a je možné tyto transformace řetězit za sebe. Transformační matice má tvar

$$T_{ab} = \begin{bmatrix} S_{ab} & u_{ab} \\ 0 & 1 \end{bmatrix} \quad (6.1)$$

Kde S_{ab} je rotační matice nebo též matice směrových kosinů a u_{ab} je radius vektor popisující translaci systému b vůči a . V případě transformační matice mezi globálním s.s. kamery a kamerou samotnou $T_{cg,cl}$ se matice směrových kosinů získá z příslušného kvaternionu, který popisuje natočení lokálního souřadnicového systému kamery vůči globálnímu. Převod z kvaternionu na rotační matici je popsán v knize [16] v kapitole 4.3. V této práci je k tomuto přepočtu využita knihovna⁴ v Pythonu. Použitím ověřené knihovny je samotný výpočet transformací po celé trajektorii optimálnější. Případně lze transformační matici získat i následovně.

$$T_{ab} = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_3q_0 & 2q_1q_3 + 2q_2q_0 & x \\ 2q_1q_2 + 2q_3q_0 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_1q_0 & y \\ 2q_1q_3 - 2q_2q_0 & 2q_2q_3 + 2q_1q_0 & 1 - 2q_1^2 - 2q_2^2 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.2)$$

⁴ scipy.spatial.transform

Kde $q_0..q_3$ jsou složky kvaternionu, jak je popsán v kapitole 2.2, a x,y,z jsou složky polohového vektoru. [zdroj]

Samotný záznam trajektorie pro imitační lakování lze chápat jako soubor transformací mezi dvěma s.s. v čase. Z dat o 6 stupních volnosti je vzat polohový vektor popisující translaci a kvaternion popisující rotaci. Z nich je posléze sestavena časová řada transformačních matic $T_{cg,cl}(t)$.

K sestavení kinematického řetězce jsou dále třeba transformace definující polohu robotu, natočení stolu a polohu kalibračního přípravku.

Zkrácený kinematický řetězec popisující pouze transformaci od základny robotu po trysku lakovacího nástroje má tvar:

$$T_{rb,st}(t) = T_{rb,world}T_{world,table}T_{table,cg}T_{cg,cl}(t)T_{cl,st} \quad (6.3)$$

Kde transformace mezi stolem a globálním systémem kamery se určí během kalibrace následovně:

$$T_{table,cg} = T_{table,cal}T_{cal,st}T_{st,cl}T_{cg,cl}(t_{calibration})^{-1} \quad (6.4)$$

Při kalibraci se lakovací pistole vloží do kalibračního přípravku. Tento moment se označí $t_{calibration}$ a takto je propojen globální s.s. kamery a celá zaznamenaná trajektorie s kalibračním přípravkem, posléze polohovacím stolem. Detailněji je proces popsán v kapitole 10. Zde jsou popsány jednotlivé matice z řetězce (6.3) a (6.4):

- $T_{rb,world}$ je inverze transformační matice umístění robotu v rámci pracoviště. V simulovaném pracovišti má tvar:

$$T_{rb,world} = T_{world,rb}^{-1} = E(-x_{rb}, -y_{rb}, -z_{rb})^T \quad (6.5)$$

- $T_{world,table}$ popisuje, jak má být polohovací stůl natočen vůči pracovišti. Je závislá na požadovaném úhlu φ v daný čas:

$$T_{world,table} = T_{\varphi_z}(\varphi(t)) = \begin{bmatrix} \cos(\varphi(t)) & -\sin(\varphi(t)) & 0 & 0 \\ \sin(\varphi(t)) & \cos(\varphi(t)) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.6)$$

Je využita pouze pokud je potřeba nahranou trajektorii natočit čelem k robotu. Pokud byla trajektorie zaznamenaná ze stacionární pozice, je tato matice z kinematického řetězce vypuštěna nebo nahrazena jednotkovou maticí.

- $T_{table,cal}$ popisuje polohu kalibračního přípravku na polohovacím stole.
- $T_{cal,st}$ popisuje polohu trysky v kalibračním přípravku ve chvíli, kdy je do něj lakovací pistole vložena.
- $T_{st,cl}$ je transformační matice ze s.s. lakovací pistole do lokálního s.s. kamery. Je spíše fiktivní, protože jakákoli reálná transformace se stejně během kalibrace ztratí. Trackovací kamera může být na lakovací pistoli umístěna libovolně. Přesto je ale důležitá, protože transformuje s.s. používaný kamerou T265 s osou y směřující vzhůru do s.s. lakovacího pracoviště, kde osa z směřuje vzhůru. V jednoduchosti zaměňuje osy a mění jejich pořadí.

$$T_{st,cl} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.7)$$

Celý kinematický řetězec popisující polohu lakovací pistole vůči základně robotu pak má tvar:

$$T_{rb,st}(t) = T_{world,rb}^{-1} T_{world,table} T_{table,cal} T_{cal,st} \dots \dots T_{st,cl} T_{cg,cl}(t_{calibration})^{-1} T_{cg,cl}(t) T_{st,cl}^{-1} \quad (6.8)$$

Tento řetězec je využit při zpracování dat. Příklad programu pro zpracování nahrané trajektorie se nachází v C3.

6.2. Inverzní kinematika

V předchozí části byl popsán převod trajektorie trysky lakovací pistole do souřadnicového systému robotu. Cílem inverzní kinematiky je nalézt takovou konfiguraci jednotlivých kloubů robotu, aby koncovým efektem byla dosažena požadovaná poloha na trajektorii včetně správné orientace. Lakovací hlavice robotu se musí během imitačního lakování nacházet ve stejné poloze, jako se nacházela stříkací pistole během vzorového lakování.

Téměř všechny průmyslové roboty jsou konstruovány tak, aby bylo možné jejich inverzní kinematiku vyřešit analyticky. Robotické rameno tvoří otevřený kinematický řetězec zakončený efektem. K tomu, aby se dala IK vyřešit analyticky, musí mít 3 po sobě následující osy společný průsečík v jednom bodě. A nezáleží na typu uzlu, může to být lineární osa nebo rotační kloub. [16] Proto většinou poslední tři osy mají společný průsečík, který lze nahradit sférickým kloubem. Při výpočtu se kinematický řetězec v tomto bodě rozdělí a porovnávají se řetězce zleva a zprava, přesněji od základny robotu a od požadované polohy koncového efektoru. Polohové vektory získané oběma cestami se v daném bodě rovnají a konfigurace robotu se získá řešením soustavy rovnic. [16]

$$SL = \begin{pmatrix} L_2 + L_{45} \cos(\Phi_2 + \Phi_3) + h_4 \sin(\Phi_2 + \Phi_3) + L_3 \cos(\Phi_2) \\ 0 \\ h_2 - L_{45} \sin(\Phi_2 + \Phi_3) + h_4 \cos(\Phi_2 + \Phi_3) - L_3 \sin(\Phi_2) \end{pmatrix}$$

$$SR = \begin{pmatrix} s_1 \cos(\Phi_1) - y_{7,st} (s_{12} \cos(\Phi_1) + s_{22} \sin(\Phi_1)) - z_{7,st} (s_{13} \cos(\Phi_1) + s_{23} \sin(\Phi_1)) - x_{7,st} (s_{11} \cos(\Phi_1) + s_{21} \sin(\Phi_1)) + s_2 \sin(\Phi_1) \\ s_2 \cos(\Phi_1) - y_{7,st} (s_{22} \cos(\Phi_1) - s_{12} \sin(\Phi_1)) - z_{7,st} (s_{23} \cos(\Phi_1) - s_{13} \sin(\Phi_1)) - x_{7,st} (s_{21} \cos(\Phi_1) - s_{11} \sin(\Phi_1)) - s_1 \sin(\Phi_1) \\ s_3 - s_{31} x_{7,st} - s_{32} y_{7,st} - s_{33} z_{7,st} \end{pmatrix}$$

Obrázek 15: Příklad odvození výpočtu IK robotu pomocí Symbolic Math toolbox v Matlabu

Analytická metoda umožňuje nalézt přesné řešení. Může být ale zdlouhavé a náročné na výpočet, především pokud by se takto měly počítat všechny konfigurace robotu po dané trajektorii. Numerické řešení obecně nemá zaručenou stabilitu a může mít problém se singulárními polohami robotu.

Pro výpočet IK po celé délce trajektorie může být numerické řešení výrazně jednodušší. Každý bod trajektorie lakování je totiž od toho předchozího nepatrně vzdálen, a tak i natočení kloubů robotu se liší minimálně. Pro výpočet konfigurace kloubů pro danou polohu na trajektorii může předchozí konfigurace sloužit jako velmi dobrý počáteční odhad výpočtu a požadované řešení je nalezeno v jednotkách iterací.

V této práci je IK řešena pomocí funkce v PyBullet. Je to numerická metoda, která využívá tlumenou metodu nejmenších čtverců. Ta by měla být méně náchylná k singularitám. [17]

6.3. Optimalizace trajektorie

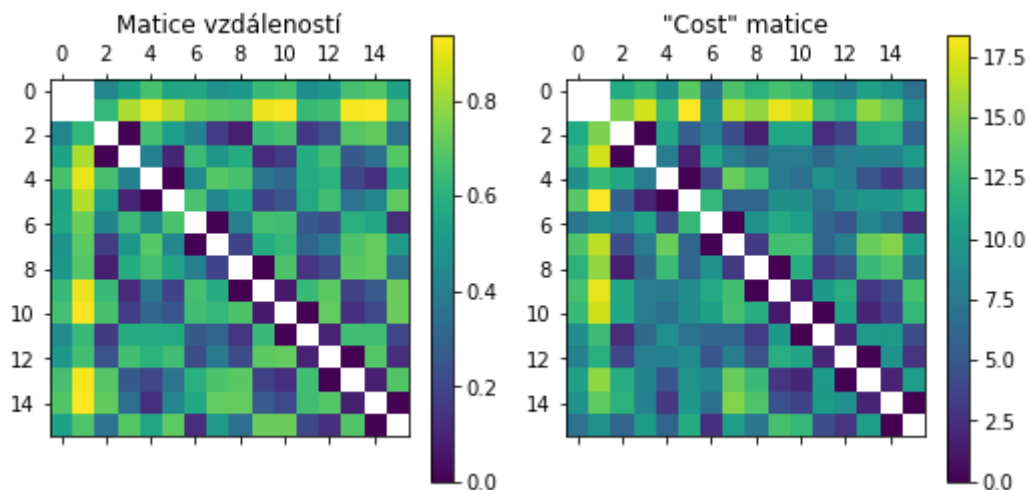
Všechny pohyby lakýrníka jsou během záznamu zachyceny jako jedna dlouhá trajektorie, ve které je uloženo všech šest stupňů volnosti, poloha a orientace, lakovací pistole. Tato trajektorie může ovšem z velké části tvořit plno hluchých míst, kde lakýrník pohybuje manipulátorem nebo se sám pouze přesouvá. Proto jsou tyto části nalezeny a odstraněny, popřípadě nahrazeny optimálnější variantou. Dále je zde velký prostor pro optimalizaci, jelikož trajektorie je záznam lakovacích pohybů člověka. Pohyby přirozené člověku jsou ale odlišné od přirozených, nebo spíše optimálních, pohybů robotu.

6.3.1. Optimální sekvence dílčích trajektorií

Z hlediska imitace lakování jsou pro nás podstatné pouze ty části trajektorie, kdy je stisknutá spoušť a je nanášena barva. Tyto části jsou dále nazývány tahy. Trajektorii je tedy možné rozstříhat na několik tahů. A pokud to technologie dovolí, mělo by být teoreticky možné tyto tahy za sebou poskládat v libovolném pořadí i směru. Případně tahy složit s menším omezením, respektujícím technologický proces, jako například směr vertikálních tahů nebo minimální dobu mezi nanášením překrývajících se vrstev z důvodu stékání barvy. To by však mohlo být algoritmicky ošetřeno. Bez omezení se takto získá celkem $2^n \cdot n!$ možných kombinací, jak lze za sebou tahy poskládat, kde n je počet tahů. Například pro pouhé tři tahy získám 48 možných permutací.

To dává značný prostor pro optimalizaci. Každý tah má jeden začátek a jeden konec. Tento problém se podobá problému obchodního cestujícího. Jeho cílem je najít optimální trasu mezi všemi body, které jsou navštíveny právě jednou. [Wikipedia TSP?]. S tím rozdílem, že pokud je například navštíven bod 4A (původní počátek tahu 4), musí pak následovat bod 4B a naopak. V podstatě jsou města u původního problému obchodního cestujícího nahrazena tunely, které se musí projet v libovolném směru.

Algoritmů pro řešení tohoto problému je několik, od exaktních přes heuristické s náhodou, po metody procházení grafu. Pro všechny metody je společné vyhodnocování vzdálenosti, popřípadě náročnosti cesty mezi dvěma body, v jednotlivých krocích. Proto je výhodné nejprve sestavit matici vzájemných vzdáleností mezi všemi body trajektorie a z té následně při hledání optimální trasy čerpat data. Je to symetrická matice, ve které se podle indexů dají vyhledat vzdálenosti jednotlivých bodů. Proto je diagonála prázdná, protože nemá smysl hledat vzdálenost mezi tím samým bodem. První index připadá počáteční poloze, ze které lakovací robot začíná svou fázi lakování. Podobně je to i s druhým indexem a koncovou bezpečnou polohou, kam se robot opět vrací. Proto i zde jsou prázdná pole mezi prvním a druhým indexem.



Obrázek 16: Matice vzdáleností a matice rozdílů konfigurací

Podobně lze sestavit i tzv. cost matici, která lépe vystihuje náročnost pohybu robotu z jedné polohy do druhé. Místo vzdáleností je zde vážený součet rozdílů v natočení jednotlivých kloubů. Ke každému kloubu je přiřazen váhový koeficient w , kterým lze vyjádřit rychlost kloubu nebo velikost hmoty napojené na daný kloub. Velikost koeficientů závisí na konkrétním robotu. U použitého robotu v simulaci byly koeficienty stanoveny podle nejvyšších rychlostí kloubů. Například je snadnější natočit kloub v zápěstí robotu, než otáčet celý robot v jeho základně. Mimo prázdné pole jsou prvky této matice určeny dle následujícího vztahu:

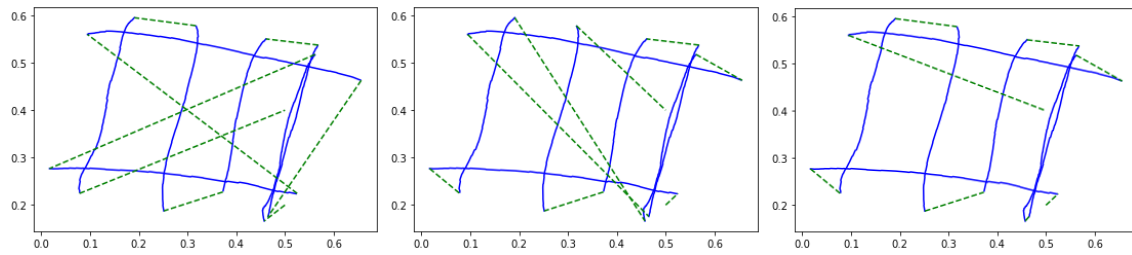
$$C_{ij} = \sum_{k=1}^6 w_k |\varphi_{ik} - \varphi_{jk}| \quad (6.9)$$

Exaktní metodou, kdy se vyhodnotí náročnost průchodu všemi možnými trasami, tzv. metodou „brute force“, lze najít optimální sled 6 až 7 tahů. Dále je náročnost a tím i doba výpočtu neúnosná. Proto se pro větší počet bodů u TSP používají spíše metody heuristické.

Velkou skupinou optimalizačních metod jsou heuristické metody založené na náhodě. Některé z nich lze aplikovat na tuto úlohu, např. genetické algoritmy nebo metodu simulovaného žíhání. Zjednodušená metoda simulovaného žíhání byla vyzkoušena k hledání optimální trajektorie. Nevýhoda této metody je, že při každém jejím spuštění dává díky náhodě jiný výsledek. I přes vysoký počet iterací nebyl výsledek příliš uspokojivý a navržená trajektorie byla jen o něco málo lepší než ta původní. Účinnější by ale měl být sofistikovanější algoritmus, který dokáže sledovat postup optimalizace a případně několikrát opakovat iterační cyklus.

Dalším typem metod, které se používají pro řešení problému obchodního cestujícího, jsou tzv. metody procházení grafu. Ty jsou jednoduše popsány a znázorněny na této stránce [18]. Používají se převážně k nalezení suboptimálního řešení propojení bodů v rovině, většina z nich je ale aplikovatelná i na prostor. Základem je poměrně jednoduchý postup, který se opakuje, dokud nejsou propojeny všechny body. Nejjednodušší algoritmus je tzv. „nejbližší soused“. Z počátečního bodu se jde k nejbližšímu dalšímu a z něj poté k dalšímu nejbližšímu, dokud nezůstane žádný volný bod. Na to, jak je tento algoritmus jednoduchý, dával při optimalizaci velmi dobré výsledky. Další slibnou metodou je „vlození nejbližšího“. Ta spočívá ve vkládání bodů do již hotové řady tahů. U této metody a dalších je potřeba ohlídat, aby počátek i konec tahu byly v sekvenci bodů vždy u sebe.

Výhodou těchto metod je jejich rychlost, proto je vhodné při optimalizaci vyzkoušet více z těchto metod a z výsledků se následně vybere nejlepší řešení.



Obrázek 17: Sled jednotlivých tahů, postupně zleva: podle původní trajektorie, metodou simulovaného žíhání a metodou nejbližšího bodu.

Předložené výsledky jsou pro názornost ukázaný na plochem tělese, algoritmy ale fungují i v prostoru. Tento postup zanedbává možnost kolize, je však mnoho způsobů, jak se jim pokusit vyvarovat. Lze využít původní trajektorii a tu s navrženou porovnat. Nebo by měla být možnost trajektorii ručně upravovat, rozdělovat a libovolně spojovat, např. pokud tam budou zjevné nedostatky.

7. Systémové řešení imitačního lakovacího pracoviště

Systém imitačního robotického lakování je komplexní systém. Skládá se z několika podsystémů, které spolu musí kooperovat. Tato kapitola slouží k popsaní fungování systému jako celku a jako návod pro jeho případnou implementaci.

7.1. Deklarace záměru

Cílem systému Robo-Painter je zaznamenat pohyby profesionálního lakýrníka při lakování a ty poté pomocí robotického ramene a manipulátoru replikovat na dalších dílech z dané série. Cílem této práce je rozšířit současný systém Robo-Painter z práce [Megi] o systém řízení manipulátoru, zpracování dat doplnit o kalibrační metodu a přidat možnost optimalizace trajektorie. Dále systém propojit se simulačním prostředím, kde budou pohyby robotu a manipulátoru ověřeny před samotným procesem lakování. Výsledný lakovaný povrch by měl být stejně kvalitní jako od profesionálního lakýrníka. Systém by měl být téměř samostatný. Hlavním vstupem do systému je trajektorie, která se zaznamená při počátečním měření. V případě složitějších výrobků bude možné do systému zadat doplňující parametry výrobku (např. hmotnost, poloha těžiště nebo kolizní obrys). Pro využívání systému nebudou třeba žádné znalosti programování. Součástí bude uživatelské prostředí a možnost manuálního řízení manipulátoru. Dále bude možnost vizualizace simulovaných pohybů robotu a pracovního stolu s případnou kontrolou kolizí.

7.2. Odborný článek

Současný systém Robo-Painter využívá jako sensorický systém pouze trackovací kameru popsanou v kapitole [asi 2]. Ten je rozšířen o senzor spouště stříkací pistole a prvky manuálního ovládání manipulátoru.

Proces záznamu trajektorie lakování probíhá následovně. Nejprve je vytvořen nový projekt, do něhož se uloží parametry lakování, pohybů manipulátoru apod. Před každým novým záznamem by měla proběhnout i kontrola hardwaru, kontrola datového toku z trackovací kamery a funkčnost Hallova senzoru u spouště. Volitelná důkladnější kontrola umožňuje i ověření trackovací schopnosti kamery v daných podmínkách. Následně je spuštěn samotný záznam a vzorové lakování může začít. Alespoň jednou během záznamu se musí stříkací pistole vložit do kalibračního přípravku a tlačítkem potvrdit její pozici. Systém si tuto polohu uloží a tím se v záznamu vytvoří referenční bod pro kalibraci. Vhodné je to uskutečnit ještě před samotným lakováním.

Lakovaný předmět je v průběhu záznamu možné natáčet pomocí manipulátoru. Ten se ovládá pomocí dvojice pedálů nebo jiných ovládacích prvků. Pokud se lakýrník rozhodne předmět nenatáčet a místo toho ho obcházet, stačí na konci přiřadit jednotlivým úsekům trajektorie úhel, pod kterým bude předmět v automatickém režimu natočen, aby ho robot mohl nalakovat. Data ze senzoru spouště a enkodéru manipulátoru se ukládají během samotného měření a jsou připojeny k datům o trajektorii.

Zaznamenaná trajektorie je v původním programu zpracována a případně filtrována. V rozšířeném systému je dále možnost vyřazení částí trajektorie, kde není stisknuta spoušť a barva tedy nestříká. Dané úseky jsou nahrazeny trajektorií s nejkratší dobou přesunu. Toto je provedeno, pokud je splněna uživatelem stanovená podmínka, např. na vzdálenost koncového bodu předchozího funkčního úseku a počátečního bodu následujícího funkčního úseku trajektorie. To by mělo zabránit kolizím při přesunu pracovní hlavičky na větší vzdálenost. Pokud je zvolena možnost

optimalizace trajektorie, je změněn sled jednotlivých tahů podle zvoleného kritéria. Postup optimalizace a použité algoritmy jsou popsány v kapitole [6.3.1].

Součástí systému je i řízení manipulátoru a robotu. Motor manipulátoru je ovládán servozesilovačem, který lze přímo propojit s řídicím počítačem. Řízení bude polohově, tak aby v co nejkratším čase dosáhl požadovaného úhlu, ale zároveň při tom nepřekročil maximální dovolenou úhlovou rychlost a zrychlení, aby nedošlo k prokluzu nebo překlopení lakovaného předmětu. Tyto dva parametry budou vypočítány z orientační polohy těžiště zadané uživatelem, nebo budou použity výchozí hodnoty.

Předpokládá se, že robot bude mít vlastní řídicí jednotku, a tedy mu budou postačovat data o trajektorii, případně data o natočení kloubů v čase.

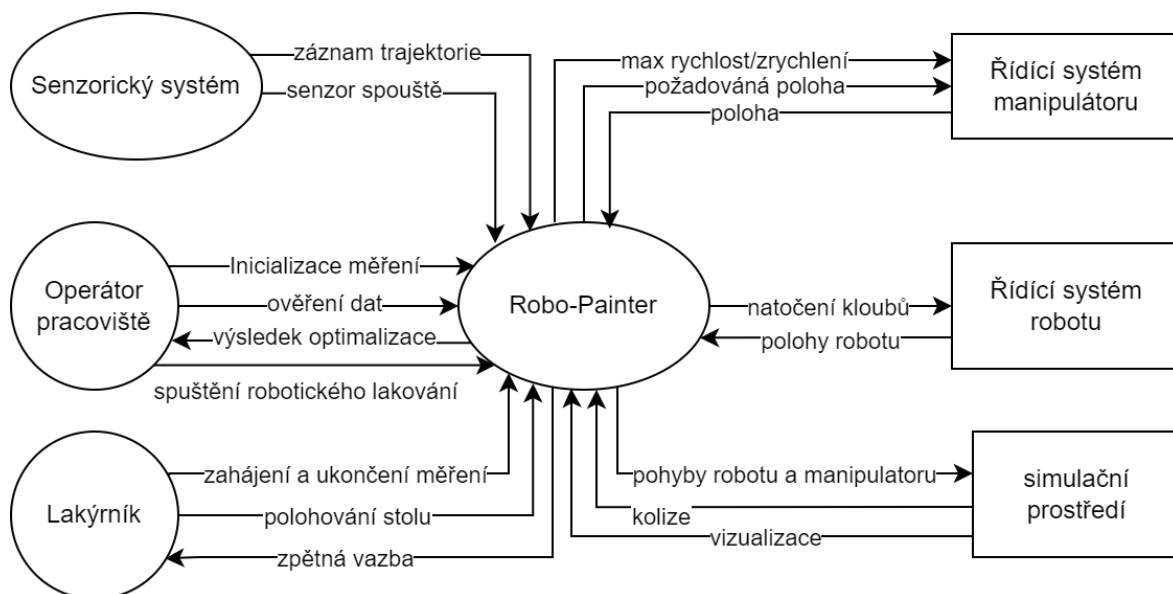
Poslední volitelnou fází je simulace celého lakovacího pracoviště, kde si uživatel může ověřit a vizualizovat pohyby robotu a manipulátoru. Do systému lze nahrát i zjednodušenou geometrii lakovaného výrobku nebo alespoň jeho obrys a program provede kontrolu kolizí.

Systém může být užíván více osobami. Ty jsou rozděleny do uživatelských rolí podle činností, které v souvislosti s imitačním systémem vykonávají. Jedna osoba může mít více rolí.

- **Správce projektu** může začít nový projekt, zapsat do něho parametry lakování a vytvořit konfigurační soubor lakovanému výrobku.
- **Operátor lakovacího pracoviště** inicializuje měření, kontroluje záznam měření a výsledky optimalizace trajektorie.
- **Lakýrník** zahajuje a ukončuje měření trajektorie. Ovládá sensorický systém a manipulátor.

7.3. Kontextový diagram

Propojení jednotlivých částí systému, vztahy mezi nimi a směry informačních toků popisuje kontextový diagram níže.



Obrázek 18: Kontextový diagram

7.4. Seznam událostí

Zde je výčet hlavních událostí v systému:

- Vytvoření projektu
 - Uložení parametrů lakování
 - Uložení parametrů polohování
- Měření
 - Příprava měření
 - Kalibrace
 - Vlastní měření
 - Uložení dat
- Post-processing
 - Transformace souřadnic
 - Plánování trajektorie
 - Optimalizace
- Simulace a kontrola
 - Simulace a ověření kolizí
- Automatický provoz
 - Umístění předmětu
 - Inicializace procesu
 - Lakování
 - Výměna předmětu

Některé události se můžou v rámci jednoho cyklu nebo fáze projektu opakovat, například pokud se při kontrole objeví nedostatky nebo chyba, vrací se zpět k upravení parametrů a zpracování znovu.

7.5. Návrh HW a SW řešení

Jádro systému poběží na běžném osobním nebo industriálním počítači. Jelikož zpracování dat a případná optimalizace probíhá off-line, není třeba nijak vysoký výkon. Specifický hardware slouží k záznamu trajektorie a ukládání dat ze sensorů spouště. Ten může být uzpůsoben buď bezdrátově nebo připojen přímo k hlavnímu počítači.

V bezdrátové variantě je využit jednodeskový počítač typu Raspberry Pi 4B s vlastním zdrojem ve formě powerbanky. K němu je přes rozhraní USB3.0 připojena trackovací kamera. Hallův senzor, indikační diody a ovládací tlačítka jsou připojeny k GPIO pinům Raspberry Pi. V závislosti na použitém typu Hallova senzoru je třeba využít převodník logických úrovní, Raspberry Pi funguje na 3.3 V logické úrovni.

Druhou možností je připojení trackovací kamery kabelem USB přímo k řídicímu počítači. Senzor spouště, indikační diody jsou napojeny na IO vstupy daného počítače, pokud jimi disponuje, v opačném případě je využit mikrokontroler, který tato data serializuje.

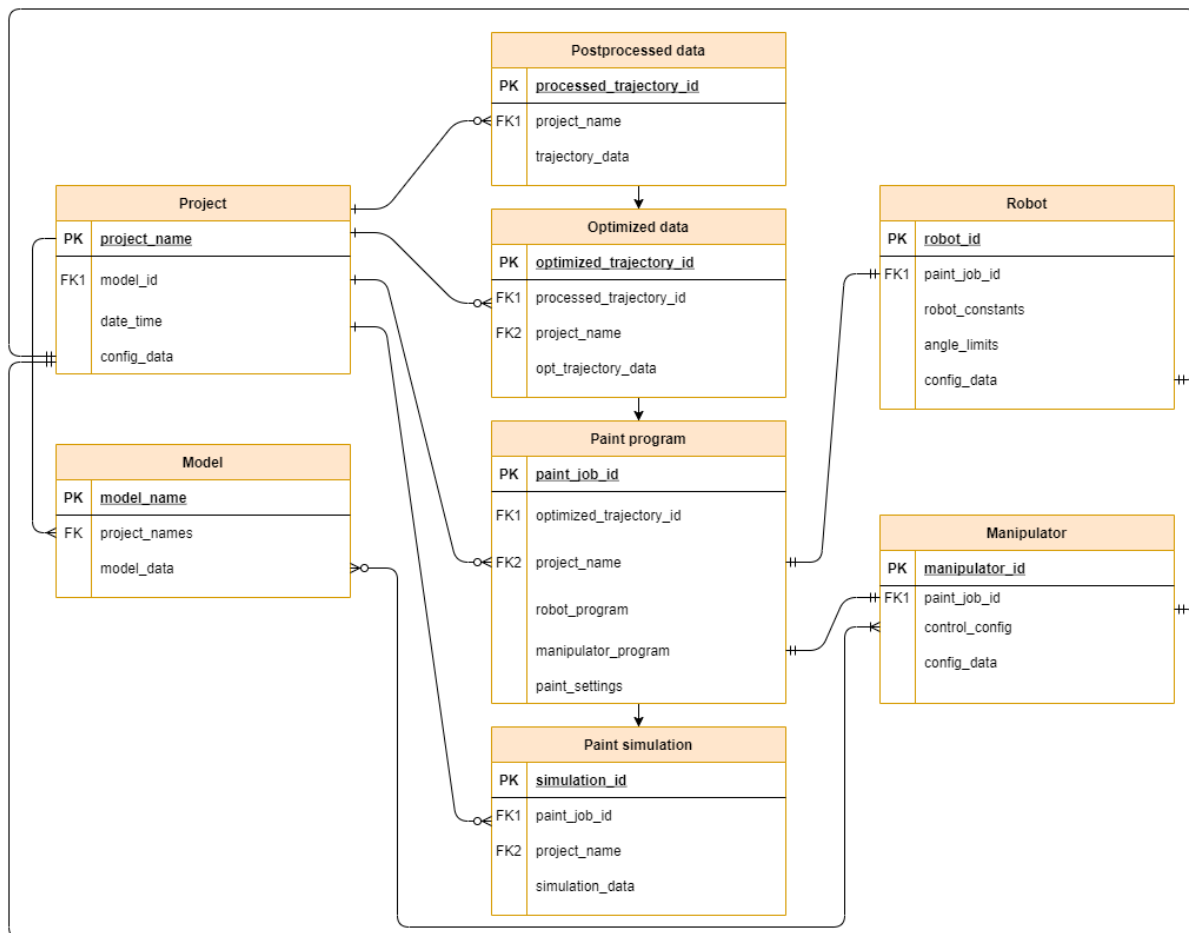
O řízení polohovacího stolu se stará servozesilovač popsán v kapitole 16. Lakovací robot je řízen příslušnou řídicí jednotkou. Obe zařízení získávají příkazy z centrálního počítače.

Současný prototyp je implementován v Python 3. K ovládaní kamery T265 je využita knihovna pyrealsense2 a jako simulační prostředí je použit Pybullet. Finální verze systému bude řízena softwarem Twincat od firmy Beckhoff. O zpracování, optimalizaci a vizualizaci dat se postará program Matlab nebo Python. Software Twincat vyžaduje operační systém Windows 7 nebo

Windows 10. Není nutné, aby měřicí část s řídicí částí systému Robo-Painter pracovaly na stejném stroji jako část systému na zpracování dat a optimalizaci.

7.6. Datový model

Zde jsou základní entity, ze kterých se celý systém skládá.



Obrázek 19: ER diagram

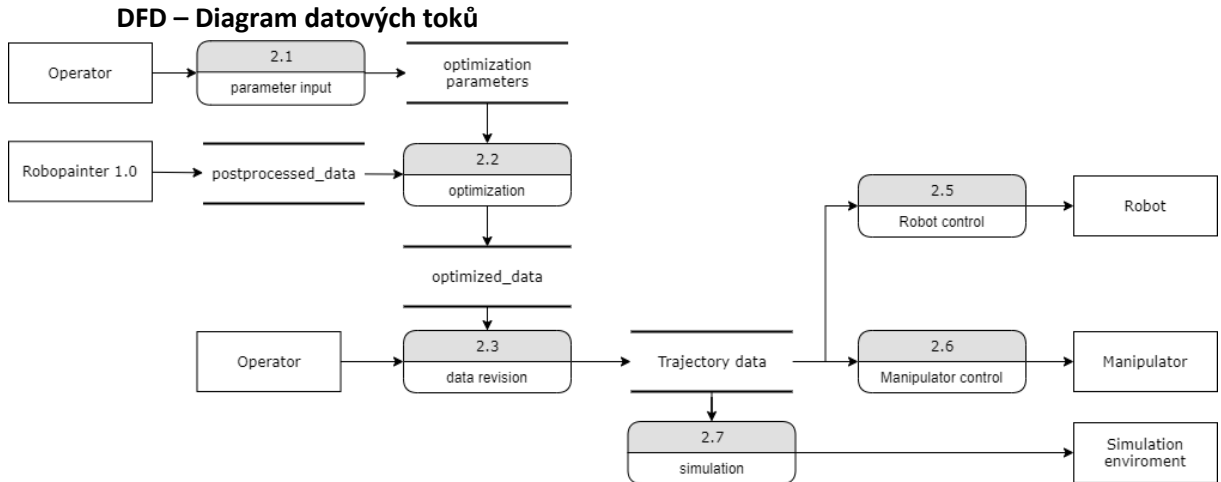
Nejvyšší entitu představuje Model, ten v sobě obsahuje všechna data spojená s lakovaným výrobkem. K modelu mohou být přiřazeny projekty, ty obsahují všechna data trajektorií, řídicí program automatického lakování a případnou simulaci. Každá entita v sobě nese informace, kdy byla vytvořena a z čeho vychází, např. optimalizovaná data vychází ze zpracovaných dat z měření.

Datový slovník

Tabulka 2: Datový slovník

Název	Typ	Popis
raw_trajectory	table	Záznam trajektorie
postprocessed_trajectory	table	Zpracovaný záznam trajektorie
optimized_trajectory	table	Optimalizovaná trajektorie
trigger_on	boolean	Stisknutá spoušť stříkací pistole
table_pos	float	Měřený úhel natočení pracovního stolu (manipulátoru)
alpha_max	float	Maximální dovolené úhlové zrychlení manipulátoru

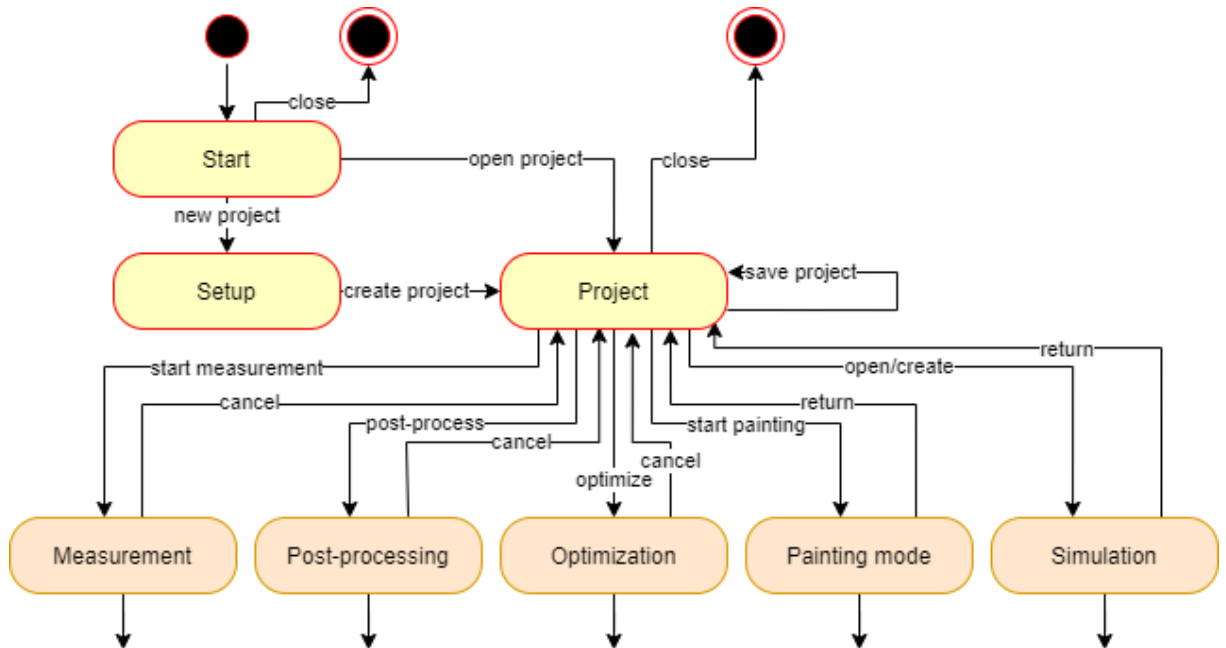
omega_max	float	Maximální dovolená úhlová rychlost manipulátoru
Phi_table	float	Požadovaný úhel natočení pracovního stolu
Phi_1, Phi_2, ... Phi_6	float	Požadované úhly natočení kloubů robotu
robot_consts	dictionary	Konstanty robotu



Obrázek 20: Diagram datových toků (DFD), úroveň 1

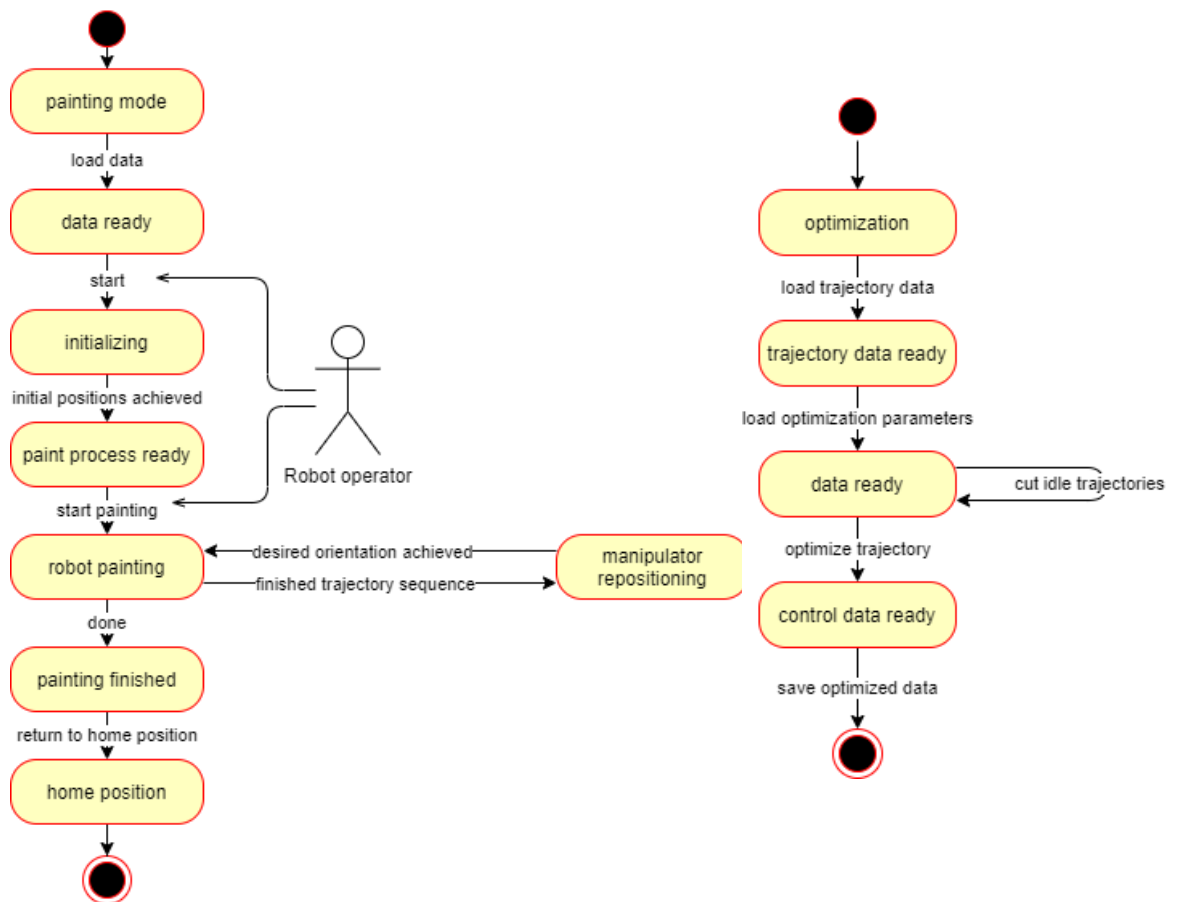
Stavový diagram

Stavový diagram popisuje, jak budoucí software funguje a jaké procesy v něm probíhají.



Obrázek 21: Stavový diagram systému, úroveň 1

Níže jsou dílčí stavové diagramy procesů optimalizace a imitačního lakování.



Obrázek 22: Stavové diagramy robotického lakování a optimalizace, úroveň 2

7.7. Uživatelské rozhraní

Uživatelské rozhraní by měl tvořit systém Twincat, do kterého bude systém implementován. Běžné UI pro tvorbu projektu, ukládání apod. Specifické prvky UI v tomto systému jsou pedály pro ovládaní manipulátoru, tlačítka pro spuštění, přerušení, pokračování a ukončení záznamu, dále tlačítko pro potvrzení kalibrační polohy. Stříkací pistole je navíc doplněna mimo senzor spouště i o pár indikačních diod, které lakýrníkovi poskytují zpětnou vazbu ohledně stavu nahrávání a funkce Hallova senzoru. V uživatelském rozhraní by měla být možnost úpravy trajektorie. Uživatel bude moci trajektorii stříhat na jednotlivé tahy, spojovat nebo měnit pořadí tahů.

7.8. Testování

Systém Robo-Painter bude testován na několika úrovních. Základní test bude zaměřen na chování systému při různých vstupních datech. Dále bude ověřena optimalizační metoda porovnáním různých vstupních a výstupních trajektorií. Také bude třeba otestovat, zda systém dokáže v požadované trajektorii odhalit nedosažitelné a singulární polohy robotického ramene.

8. Simulace celého pracoviště

Simulace pohybů robotu slouží především pro ověření výstupní trajektorie, případně optimalizované trajektorie některým z algoritmů. I přesto, že není k dispozici model lakovaného předmětu, je vhodné vytvořit alespoň velmi zjednodušený obrysový model pro ověření případných kolizí. Ty by mohly nastat například při přejezdu mezi dvěma úseky trajektorie, tzv. point-to-point pohybem.

8.1. Volba simulačního prostředí

Dostupných robotických simulátorů je dnes několik, mezi ty nejčastěji používané patří:

- CoppeliaSim, dříve V-REP, je open-source robotický simulátor dostupný na platformách Linux i Windows s přívětivým uživatelským rozhraním
- Gazebo je open-source simulátor nejčastěji používaný se systémem ROS. Je vhodnější na platformách Linux
- PyBullet je velmi uživatelsky jednoduchý simulátor ovládaný Pythonem využívající fyzikální engine Bullet
- RoboDK je komerční simulátor průmyslových robotů, jeho výhodou je široké spektrum nástrojů
- Webots je další všestranný open-source simulátor

Všechny tyto simulátory mají uživatelské rozhraní s možností přímého zásahu do simulace. Simulátory mimo PyBullet obsahují knihovny ihned použitelných základních senzorů, robotů, aktuátorů a podobně. Řada z nich umožňuje i propojení s Matlab a dalšími programy nebo jazyky. Podrobnější popis simulátorů CoppeliaSim a RoboDK je dostupný v práci *Simulace robotických a výrobních systémů* [19].

Základem těchto simulátorů je fyzikální engine, který zpracovává veškerou dynamiku tuhých těles, kolize a kontakty mezi nimi. Některé simulátory pracují i s dynamikou tekutin, toho se využívá například pro simulaci nautických robotů.

V této práci se nejedná o robota ve smyslu autonomního systému s vlastními senzory, rozhodovací jednotkou a aktuátory, které jsou často cílem testování ve zmíněných simulátorech. Sensorický systém tvořený kamerou T265 sbírá data v reálném světě, ta jsou zpracována a následně jsou aplikována v simulaci lakovacího pracoviště. Úkolem simulátoru v této práci je tedy pouze důvěrně napodobit imitační lakovací pracoviště, především pak fázi automatického robotického lakování. K tomu vystačí PyBullet, který zajistí dodržení fyzikálních principů a detekci případných kolizí v tzv. „digitálním dvojčeti“ lakovacího pracoviště.

Dále v práci je také zmíněn systém ROS, který má potenciál stát se řídicím systémem celého imitačního pracoviště. Může přímo komunikovat se senzory jako výše zmíněná kamera, zpracovávat a optimalizovat data a řídit ostatní členy systému.

8.1.1. PyBullet

PyBullet je modul v Pythonu k ovládní fyzikálního enginu Bullet. Používá se k simulaci robotů, strojovému učení a obecně k testování různých algoritmů.

Jeho předností je jednoduchost, ovládá se kompletně v jazyce Python. Všechny důležité funkce a nástroje jsou popsány v 34stránkovém návodu i s ukázkovým kódem. Pokud je člověk alespoň trochu znalý Pythonu, ocení i jednoduchost instalace pomocí správce balíčků *pip*.

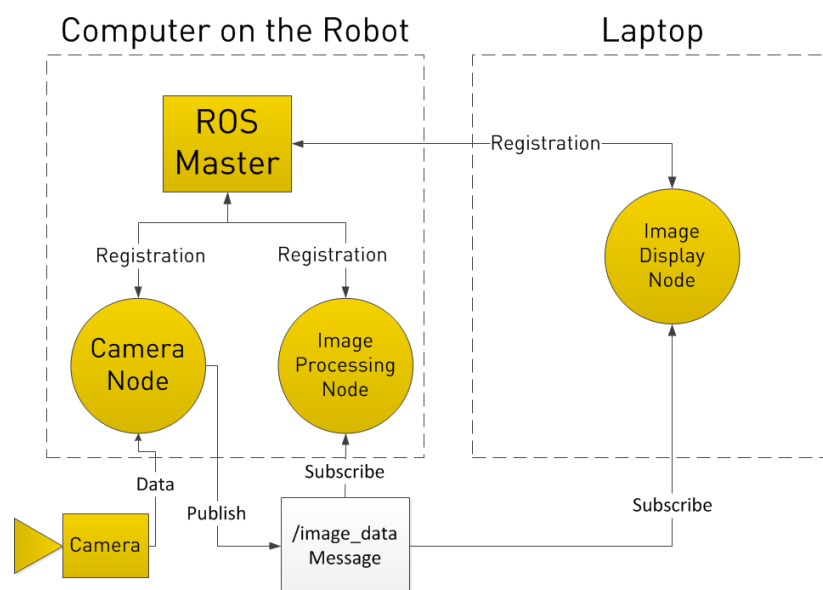
Tak jako většina open-source nástrojů je stále vyvíjen a občas má své nedostatky, často ale lze nalézt řešení na komunitním fóru. Pracuje s modely ve formátech URDF, SDF, MJCF, které jsou kompatibilní i s většinou ostatních simulátorů. To je užitečné při využití knihoven již vytvořených modelů robotů.

Poskytuje dva režimy fungování: GUI nebo DIRECT. V režimu GUI pracuje s klasickým uživatelským rozhraním. Uživatel může přímo ovlivňovat simulované prostředí například myší nebo ovládat simulaci pomocí klávesnice, nastavených tlačítek nebo posuvníků. Může i měnit pohled na simulaci. Naproti tomu mód Direct je připojen přímo k fyzikálnímu enginu bez klasického okna a jediný způsob ovládání simulace je pomocí příkazů. Jeho výpočetní náročnost je výrazně menší a hodí se k náročným simulacím, kdy není potřeba živého grafického výstupu. I tento režim však umožňuje nahrávání, případně generování pohledů. Režim direct je v této práci využit k výpočtu IK při hledání optimálních trajektorií. [20]

8.1.2. ROS

ROS je velmi populární a silný nástroj k vývoji robotů využívaný širokou komunitou vývojářů. Název je zkratka pro Robot Operating System, přestože se nejedná o operační systém v pravém slova smyslu. Jde spíše o sadu nástrojů k řízení robotů. Podobně jako operační systém zajišťuje hardwarovou abstrakci, nízkourovňové řízení, předávání zpráv mezi procesy apod. [21]

Základem je systém uzlů (*node*), které mezi sebou komunikují. Uzlem může být například řídicí algoritmus na počítači, jednotlivý proces, simulace, ale i senzor jako kamera T265. Uzly mohou být i na několika zařízeních zároveň. *ROS Master* všechny tyto uzly spravuje a umožňuje jim se vzájemně najít a komunikovat. Také pod něj spadá server s parametry, kam mají uzly přístup. Ukládají se tam různé konstanty, koeficienty, mapy apod. Uzly mezi sebou komunikují přes témata (*topic*), po kterých si posílají zprávy (*messages*). Organizace zpráv je uskutečněna pomocí služeb: *publish/subscribe*. Uzly posílají zprávy do jednotlivých témat, nebo je z nich čtou. [21]



Obrázek 23: Příklad uspořádání uzlů v systému ROS. [22]

V systému imitačního lakovacího pracoviště by to vypadalo tak, že trackovací zařízení by jako vlastní uzel posílalo zprávy „*PoseStamped Message*“ přes téma *camera_pose*. Zprávy by obsahovaly polohu a orientaci kamery včetně časové značky. Tyto zprávy by přijímal uzel na počítači a ukládal je. Tento uzel by také přijímal zprávy z uzlu, který čte data z enkodéru, a z uzlu, který zapisuje stav senzoru spouště. Tyto uzly by mohly fungovat i na mikrokontrolérech se systémem micro-ROS. Dále by v systému byly uzly pro transformaci, řízení robotu a manipulátoru a uzel simulace. Tím by byl simulátor Gazebo. Nespornou výhodou systému ROS je, že stejné uspořádání systému, které je použité pro simulaci, je možné použít i v reálném světě. Uzel simulace se pouze zamění s uzly reálných akčních členů systému.

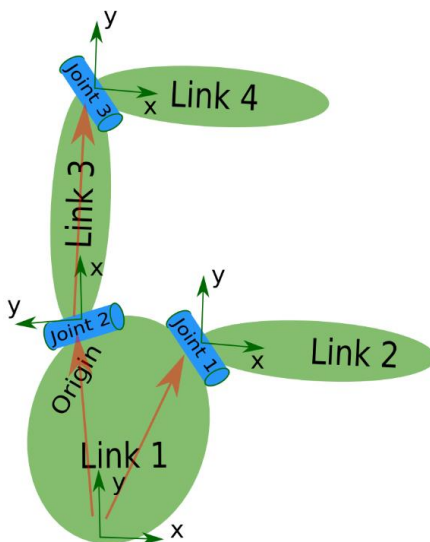
ROS funguje na operačních systémech Linux, verze pro Windows existuje, ale je značně omezena. Výhodnější je využít WSL (Windows Subsystem for Linux), ve kterém ROS funguje, ale s omezeným výkonem. ROS byl v této práci využit pro zpracování modelu robotu a jeho převedení do URDF souboru. Kvůli omezením spojených s operačním systémem a komplexností systému ROS byl k simulaci pracoviště využit pouze PyBullet.

8.2. Příprava simulace a prostředí

Nejprve je potřeba do prostředí simulace nahrát modely všech těles. Každý model je reprezentován vlastním URDF souborem. To je zkratka pro Unified Robot Description Format, který se používá k popisu robotů a je napsán ve stylu XML, takže se dá upravovat v každém textovém editoru.

Model je reprezentován tělesy (link) a vazbami (joint), které tělesa propojují. Každé těleso má vlastní souřadnicový systém, tvar, hmotnost, momenty setrvačnosti i deviační momenty a mnoho dalších volitelných parametrů jako je např. barva. U těles je rozlišován vizuální a kolizní tvar.

Vazby mezi tělesy popisují vztahy podobně jako u klasických robotů nebo mechanických soustav. Základní typy vazeb jsou pevná, rotační, posuvná apod. Jsou navíc doplněny o parametry tlumení a tření, maximální rychlost a rozsah pohybů.



Obrázek 24: Struktura robotu v URDF. [21]

Každý model má jedno základní těleso (*base_link*). Jeho s.s. je pak důležitý při umísťování nebo určování polohy modelu v simulačním prostředí. V simulaci imitačního lakovacího pracoviště jsou tyto modely:

- Model polohovacího stolu je složen ze dvou jednoduchých těles, kvádrů a plochého válce, který představuje pracovní desku stolu. Vazba mezi nimi je rotační se třením i tlumením.
- Model robotu ABB IRB1600 byl použit z knihovny ros-industrial [23]. Konec robotu byl pouze doplněn o kužel, který představuje aerosol barvy.
- Model lakovaného předmětu je vymodelován v CAD programu, uložen ve formátu STL, a potom je jako trojúhelníková síť bodů nahrán do simulace. Každé těleso je v simulaci automaticky převedeno na svou konvexní obálku za účelem snazšího výpočtu kolizí. Proto musela být židle rozdělena na několik těles, která jsou spojena pevnou vazbou.



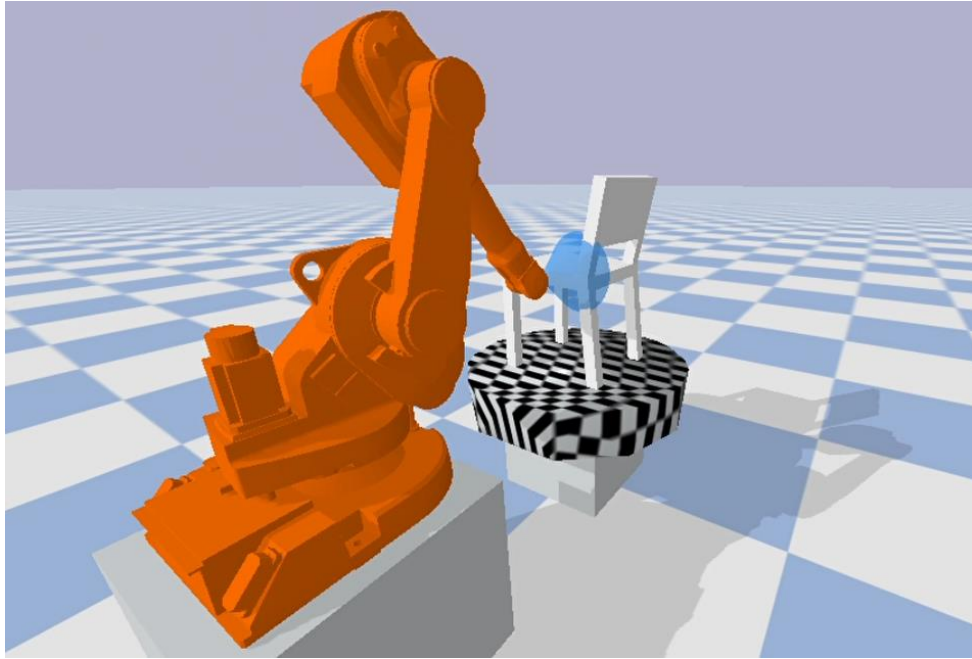
Obrázek 25: Umístění modelů v simulačním prostředí.

Modely jsou v simulaci rozmístěny dle návrhu výše. Dále jsou nastaveny parametry řízení stolu a robotu. Pohyb robotu i klouby robotu jsou řízeny polohově. Všechny pohyblivé vazby v simulátoru mají proporční regulátor rychlosti a polohy. Zesílení regulátoru polohy manipulátoru bylo vyladěno na 0,03 a rychlosti na 0,2. U robotu byly ponechány výchozí parametry.

8.3. Výsledná simulace

Zpracovaná data trajektorie a sled jednotlivých tahů je aplikován v této simulaci. Nejprve je vypočtena celá IK trajektorie lakování. Poté je robot přemístěn do polohy „doma“, kde nehrozí kolize s pohybujícím se stolem. Polohovací stůl se natočí do požadované polohy. Robot začne postupně lakovat předmět z dané strany. Mezi jednotlivými tahy robot přejíždí tzv. „point to point“ pohybem, případně po původní trajektorii, pokud by hrozila kolize. Po dokončení dané strany se robot vrátí „domů“, manipulátor se přestaví a proces se opakuje, dokud nejsou hotovy všechny strany lakovaného předmětu.

Pohyb robotu po trajektorii je realizován tak, že jsou regulátorům jednotlivých kloubů postupně dosazovány požadované polohy s určitým časovým krokem. Tak je docílen plynulý pohyb koncového efektoru po trajektorii lakování. Podobně to je i u manipulátoru, kterému jsou dosazovány body trajektorie, která byla předem vypočítána podle postupu v podkapitole 4.1.



Obrázek 26: Robot imitující lakýrníka

V částech trajektorie, kdy má proudit barva, je zviditelněn kužel na konci robotického ramene pro lepší vizualizaci, poté je opět skryt. V současném stavu se barva při simulovaném lakování na lakovaném předmětu nepromítá.

Změna barvy lakovaného předmětu by ale mohla být provedena editací textury, která je ve formě bitmapy. Ta musí být upravena pomocí UV mapování⁵, aby správně pokryla povrch tělesa. Při průsečíku tělesa s paprskem barvy se zpětně pomocí UV mapy dohledá příslušný bod bitmapy a jeho odstín se může změnit. Takto lze simulovat postupné obarvování tělesa při simulaci lakování.

⁵ UV mapování je metoda k pokrytí prostorového síťového tělesa dvourozměrnou grafikou. UV značí použité souřadnice u 2D obrázku. X, Y, Z jsou rezervovány pro prostorové těleso. [22]

9. Ověření přesnosti trackovací kamery Intel RS T265

Podmínkou funkčnosti celého systému je kvalita vstupních dat, tedy kvalita a přesnost záznamu trajektorie pomocí trackovací kamery Intel RealSense T265. Přesnost uváděná výrobcem je 1 % v uzavřené smyčce. Tedy pokud robot s touto kamerou ujede dráhu dlouhou 10 m a vrátí se opět do počáteční polohy, jeho poloha je kamerou určena s přesností na 0,1 m. Otázkou ale je, jak je to s přesností ve zbytku trajektorie. [3]

9.1. Proběhlé studie přesnosti

Studií na přesnost kamery je několik, ve většině případů se zabývají vhodností využití pro sledování (trackování) autonomních robotů. V této práci [24] je testována kamera T265 souběžně s trackovacím zařízením HTC Vive, které je využito k záznamu referenční trajektorie. Toto zařízení by mělo být velmi přesné do vzdálenosti 5 m, protože využívá systém stanic. Při používání trackovací kamery T265 je možnost vypnutí některých jejích funkcí. Jednou z funkcí je 3D mapování, které umožňuje korigovat odhadovanou polohu a tím zmenšuje naakumulovaný posuv. Její vypnutí mělo zásadní negativní vliv na přesnost. Další funkce je *Pose Jumping*. Pokud je tato funkce povolena, umožňuje to kameře náhle změnit předpokládanou polohu, pokud rozpozná scenerii, kterou už zaznamenala a má ji uloženou v paměti. V určitých případech by tato funkce mohla naopak způsobovat větší chybu, pokud by se vizuální prvky v okolí opakovaly. Poslední funkcí je relokace, kterou kamera využije po ztrátě vizuálního vjemu. Při opakovaném pohybu na prostoru 1x1,5 m byla výsledná směrodatná odchylka polohy 40,5 mm a maximální chyba 93 mm. [24]

Na webové stránce [25] je popsán experiment testování kamery pomocí kolaborativního robotu UR5. Kamera byla uchycena na konci robotického ramene. S tím bylo pohybováno po dvou trajektoriích. První trajektorie byla v kartézských souřadnicích ve směru x a následně ve směru z. Druhá trajektorie byla náhodná s komplikovanou změnou orientací. U obou trajektorií bylo pohybováno výchozí rychlostí robotu UR5, která by měla zhruba odpovídat rychlostem člověka při opakovaných pohybech. U druhé trajektorie navíc proběhl i experiment se zakrytými kamerami pro ověření funkce se samotnými inerciálními senzory. Zjištěná přesnost je podobná experimentům v této práci. V některých úsecích s vysokým zrychlením dosáhla chyba až 10 cm, ale díky VSLAM algoritmu se kamera dokáže zpětně zorientovat a eliminovat tak aktuální chybu, pokud se pohybuje v téměř uzavřených trajektoriích. [25]

Podobný experiment jako v této práci byl proveden v práci [1]. Pomocí klikového mechanismu bylo kamerou pohybováno v jedné ose. Výstup tedy měl mít téměř sinový průběh s amplitudou 68 mm. Data odpovídala předpokladu, avšak nebyla porovnána s přesnou polohou. V jednom záznamu data vykazují i malý posuv v čase. Také byl ověřen výstup z kamery v klidovém stavu, kdy kamera vykazovala stabilní polohu s občasnými milimetrovými výchylkami. [1]

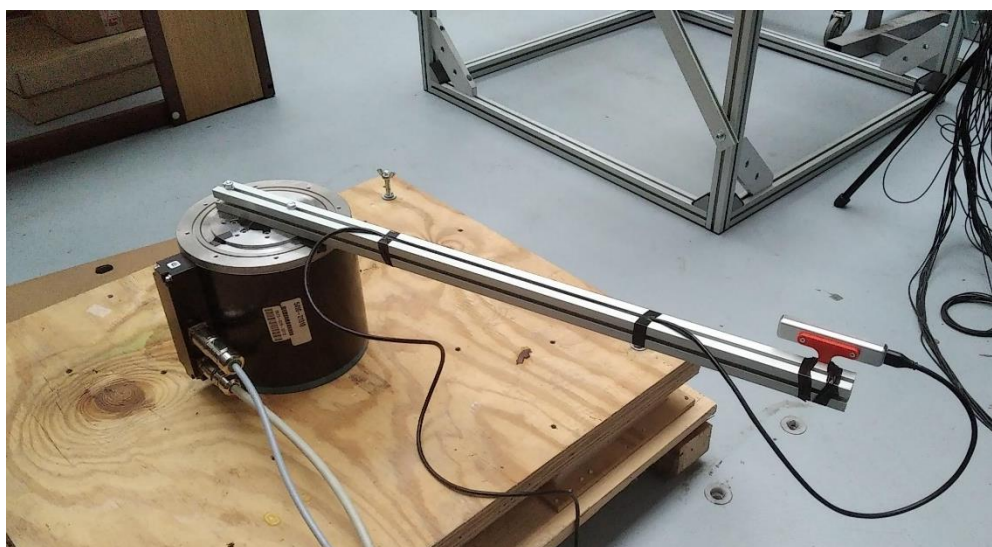
Funkčnost kamery byla testována pouze v jedné ose v rozmezí 136 mm. Naměřená data nebyla porovnána s přesnou referenční polohou, byla spíše pozorována konzistence dat nebo jejich trend. Proto je kamera ověřena znovu s přesným odměřováním polohy a na větším prostoru, který by mohl odpovídat prostoru imitačního lakování a zároveň by měl dostatečný prostor zachytit případný trend ztráty přesnosti v závislosti na vzdálenosti od počátku měření.

I přesto, že přesnost zařízení je poměrně známá, může se lišit na základě okolních podmínek, především osvětlení nebo vizuálních podnětů. Proto je vhodné ověřit funkci kamery již na cíleném místě použití a případně upravit vizuální scénu.

9.2. Vlastní ověření

Měření proběhlo v laboratoři za dobrých světelných podmínek s poměrně pestrou vizuální scénou. K ověření byl využit výše zmíněný motor, který sloužil jako vedení. Kamera byla pevně uchycena na rameni z hliníkovému profilu, viz Obrázek 27. Přesnou polohu motoru, a tedy i kamery na rameni, určoval během měření zabudovaný enkodér. Ramenem bylo pohybováno ručně, aby se napodobovaly pohyby při lakování. Kamera byla na rameni uchycena v různých směrech ve vzdálenosti 500 mm od středu rotace kromě případu c), kdy je uchycena na rameni 700 mm.

- a) Pohled ve směru pohybu po kružnici
- b) Pohled vně kružnice
- c) Pohled ve směru pohybu po kružnici na rameni 700 mm



Obrázek 27: Proces ověřování přesnosti kamery.

Kamerou bylo pohybováno po obloucích i po celých kružnicích. Pevným umístěním na rameni je dán poloměr pohybu a spolu s daty z rotačního enkodéru je následně rekonstruována trajektorie, po které se kamera pohybovala. Tato trajektorie je brána jako referenční. Přesnost této trajektorie je ovlivněna odměřením vzdálenosti na rameni, rozlišením enkodéru a přesností 3D tištěného úchyty kamery. Stanovení přesnosti kamery proběhlo podle následujících kritérií:

- 1) Přesnost polohy v daném čase
- 2) Přesnost orientace kamery

V prvním bodě je v každém časovém kroku porovnávána naměřená poloha s referenční polohou. Jsou měřeny všechny tři souřadnice x , y , z . Výsledná přesnost je stanovena jako kombinovaná odchylka souřadnic x a y . Tedy v rovině, ve které se kamera pohybovala. Kamerou byly naměřené i výchylky

v ose z, ale protože rameno s kamerou není dokonale tuhé a tato souřadnice nebyla přesně odměřována, do chyby není započítána. Výchyly naměřených hodnot všech tří souřadnic jsou vykresleny v příloze A. Zaznamenaná orientace kamery byla stejně jako v prvním bodě porovnána s referenční orientací v každém časovém kroku.

Takto probíhal postup ověřování:

1. Nahrání trajektorie kamery
2. Příprava dat pro zpracování
3. Transformace souřadnic
4. Převzorkování a synchronizace dat
5. Porovnání přesných a naměřených dat

Pro srovnání naměřených a předpokládaných dat je potřeba převést všechna data do stejného souřadnicového systému. Jako referenční s.s. byl zvolen takový, v jehož středu leží střed otáčení ramene a osa z je shodná s osou otáčení. Transformací souřadnic byla data v s.s. kamery převedena do referenčního s.s. podobným způsobem, který je popsán v podkapitole Transformace souřadnic na straně 29.

Dále bylo prověřeno, zda si data odpovídají v čase. Datům z kamery je přiřazena časová značka už během pořizování snímku v kameře. Ale datům o natočení z enkodéru byl přidělen čas až při ukládání v počítači, kde mohla vzniknout určitá časová prodleva. Proto jsou polohová data porovnána a pomocí korelace obou signálů je odhalen případný časový posuv dat, který je následně opraven. U naměřených dat byl tento posuv pouze o jeden vzorek, což odpovídá 5 ms.

Přesnost určení polohy kamerou byla určena v kartézských souřadnicích. Maximální a směrodatná odchylka v rovině xy je v tabulce Tabulka 3. Kompletní naměřené hodnoty a výsledky jsou v příloze A.

Přesnost polohy není jediný důležitý parametr. Důležitá je i správná orientace lakovací pistole. Ta byla stanovena jako úhel mezi naměřenou a předpokládanou orientací v daném čase. Tento úhel je dán vztahem:

$$\phi = 2 \arccos(|\langle p, q \rangle|) \quad (9.1)$$

kde $\langle p, q \rangle$ je skalární součin kvaternionu orientace kamery p a kvaternionu natočení ramene q . [26]

9.3. Vyhodnocení

Měření	1	2	3	4
Směřování kamery	po směru	vně	po směru	dovnitř
Poloměr otáčení	500 mm	500 mm	700 mm	500 mm
Směrodatná odchylka	18,7 mm	25,6 mm	34 mm	34,6 mm
Maximální chyba	58,0 mm	79,8 mm	71 mm	68 mm
Maximální úhlová chyba	1,58 °	1,95 °	1,96 °	2,54 °

Tabulka 3: Výsledky měření.

Z naměřených dat nebyla prokázána žádná závislost chyby na rychlosti. U druhého měření, kdy je natočena kamera směrem od středu otáčení, dosahuje chyba poměrně vysokých hodnot. především ve vzdálenějších polohách od počátku měření. Kamera se pohybuje rychlostí danou

ramenem a poloměrem, na kterém je k němu přichycena. Ale při pohledu kamery vně se orientační body v zorném poli kamery pohybují vůči kameře relativně rychleji. To by mohlo způsobovat naměřenou chybu ve druhém měření. Z grafů závislosti chyby na vzdálenosti od počátku měření lze vyčíst mírně rostoucí trend.

Odchylka naměřené orientace kamery je vůči přesné orientaci poměrně malá. Až na jednu anomálii se chyba pohybuje do 2° . To se zdá jako velmi dobrý výsledek. Nicméně po detailnějším zkoumání průběhů chyb je vidět mírná korelace mezi chybou orientace a chybou polohy. Chyba v odhadu orientace dokonce předchází polohovou chybu. Tento trend je nejlépe vidět na datech z druhého a třetího měření.

Samotná chyba orientace není v trajektorii lakování tak zásadní. I při nepřesnosti orientace do 2° by při lakování ze vzdálenosti 25 cm nebyl posun barvy na výsledném povrchu větší než 9 mm. Špatný odhad orientace způsobuje problém hlavně při odhadu polohy. Data z akcelerometru se po dvojí integraci přičtou k poloze ve směru, který je dán odhadem orientace. Takto můžou vznikat velké nepřesnosti, které musí kamera korigovat. Zjištěná přesnost přibližně odpovídá přesnosti ve výše zmíněných studiích.

10. Metody kalibrace systému

Trackovací kamera T265 si během záznamu trajektorie lakování vytvoří vlastní souřadnicový systém, ve kterém je celá trajektorie lakování uložena. K tomu, aby tato trajektorie mohla být přenesena do s.s. pracoviště, musí být známý vztah mezi těmito s.s. Lakovací pistole, ke které je kamera pevně přichycena, se vloží do kalibračního přípravku. Ten slouží jako referenční bod, jehož poloha je v rámci pracoviště přesně dána. Tím se propojí kinematický řetězec od trysky lakovací pistole přes polohovací stůl k lakovacímu robotu. Přesný popis transformace trajektorie s užitím tohoto referenčního bodu je popsán výše, v podkapitole 6.1.

Kalibrační přípravek je navržen pro běžně používanou lakovací pistoli. Využívá válcových ploch těla pistole, které musí být souosé s tryskou. Její poloha je totiž cílem kalibrace. Přípravek tvoří jedna nebo dvě drážky tvaru V, do kterých se pistole v příslušných místech vloží. Takto je zajištěna správná orientace i poloha v příčných směrech. K zajištění správné polohy v podélném směru je využita přirozená zarážka na těle lakovací pistole.

K ověření algoritmu pro transformaci souřadnic a k nahrání trajektorie simulovaného lakování byl vytvořen prototyp, který je vidět na Obrázek 28. Přestože byla použita lakovací pistole neúplná a do drážky byl vložen pouze vnější zavít, kalibrace fungovala. Výhoda využití V drážky s válcem je její samostředící schopnost. Poloha válce je pak přesně definována a jeho vzdálenost od sedla drážky lze vypočítat:

$$h = \frac{d}{2} \sin\left(\frac{\beta}{2}\right) + h_0 \quad (10.1)$$

kde h je výška trysky nad stolem, d je průměr válcové plochy, β je úhel drážky a h_0 je výška začátku drážky nad stolem. Vzhledem k přesnosti je ale výhodnější naměřit vertikální polohu trysky lakovací pistole vůči polohovacímu stolu napřímo.



Obrázek 28: Prototyp kalibračního přípravku se stříkací pistolí a jeho umístění v rámci souřadnicového systému stolu.

Tímto kalibračním bodem a popisem jeho použití je splněn i jeden z bodů navržených k dalšímu rozpracování v práci *Robotic painting cell design using worker's motion tracking and imitation* [1]. Systém imitačního robotického lakování by tedy v současné podobě měl být aplikovatelný v praxi.

Závěr

Touto prací byl doplněn a dále rozšířen systém robotického lakování, který se učí od pohybů profesionálního lakýrníka. Bylo navrženo uspořádání lakovacího pracoviště a vybrán robot, který bude lakýrníka zastupovat při opakujících se lakovacích procesech.

Systém doplněn o rotační manipulátor, který nastavuje lakovaný předmět robotu, aby jej robot dokázal pokrýt ze všech požadovaných stran. Řízení manipulátoru je navrženo tak, aby v co nejkratším čase, ale zároveň bezpečně, přestavilo lakovaný předmět. Pohyb manipulátoru je uskutečněn po přesně definované trajektorii s-křivky. Trajektorie je v práci do detailu popsána spolu s postupem její aplikace k řízení průmyslového servopohonu. Ten by měl sloužit jako pohon manipulátoru.

Imitační lakovací pracoviště je popsáno i po systémové stránce, jak by mělo fungovat a co s čím v systému komunikuje. Operace v jednotlivých fázích jsou popsány pomocí stavových diagramů. Jsou navrženy i datové struktury systému.

K ověření systému byla provedena simulace celého pracoviště. Záznam trajektorie lakování byl zpracován a pomocí transformačního postupu převeden do souřadnicového systému robotického pracoviště. Následně byl souhrou robotu a manipulátoru v simulačním prostředí nalakován celý lakovaný předmět.

Dále byl ověřen sensorický systém, který má sledovat pohyby lakýrníka a rekonstruovat z nich trajektorii lakování. Přesnost navrženého trackovacího zařízení se ukázala být pro klasické lakování spíše nedostatečná. Stále ale může být tento systém snímání pohybů použitelný v případech, kdy nejsou nároky na přesnost tak vysoké, např. u nanášení ochranných nástřiků apod.

Jednou z možností je zlepšit přesnost záznamu. Kamera všechna data ze sensorů zpracovává online. Pokud by bylo možné získaná data ze sensorů zaznamenat a z nich zpětně dopočítat trajektorii pohybu, mohlo by to vést k vyšší přesnosti. Další možností je využít trackovací systém podobný systému HTC Vive, který využívá systém stanic. Podobně, jako jsou zařízení se systémem GPS lokalizovány družicemi, byl by sensor na lakovací pistoli lokalizován stanicemi v lakovacím boxu.

Tato práce ale není omezena sensorickým systémem, vstupem do systému robotického lakování může být i obecná trajektorie v časoprostoru.

Systém imitačního robotického lakování práci lakýrníka nenahradí, alelepší mu podmínky. Stále bude potřeba dělat přípravné práce pro lakování, jako je odmašťování, maskování, zakrývání závitů atd.

Budoucí práce

Přesnost navrženého trackovacího zařízení není nijak vysoká, proto by další výzkum měl jít směrem hledání lepšího řešení. Trackovací kamera počítá odhad polohy v reálném čase. Pokud by kamera data pouze ukládala, mohla by tato data být později zpracována offline na zařízení s vyšším výkonem a pamětí. To by mělo zlepšit přesnost, nedocházelo by například k jevu *pose jumping*. Poloha by se jednoduše interpolovala mezi body, kde si je kamera svou polohou alespoň trochu jistá. Další možnost k prozkoumání je využití externích kamer k sledování polohy lakovací pistole, třeba v kombinaci s inertními senzory k určení orientace. Inspirací by mohl být i systém trackování pomocí stanic jako využívá HTC Vive.

Dále by bylo třeba ověřit časování trysky u ruční lakovací pistole a u automatické pistole na robotu, případně zajistit časovou korekci záznamu trajektorie.

Literatura a zdroje

- [1] MEJDRECHOVÁ, Megi. *Robotic painting cell design using worker's motion tracking and imitation*. Praha, 2020. Diplomová práce. České vysoké učení technické v Praze. Vedoucí práce Jan Pelikán.
- [2] EPISTOLIO. *MRK ROBOT* [online]. In: . [cit. 2022-06-22]. Dostupné z: <http://www.epistolio.it/en/robot-mrk/>
- [3] Intel® RealSense™ Tracking Camera. *Data sheet* [online]. 2019, (4) [cit. 2022-04-09].
- [4] *BMI055: IMU combining accelerometer and gyroscope* [online]. In: . Bosch Sensortec [cit. 2022-08-05].
- [5] *T265 Tracking Camera* [online]. In: . [cit. 2022-04-04]. Dostupné z: <https://github.com/IntelRealSense/librealsense/blob/master/doc/t265.md>
- [6] EATER, Ben a Grant SANDERSON. *Visualizing quaternions: An explorable video series* [online]. [cit. 2022-04-09]. Dostupné z: <https://eater.net/quaternions>
- [7] Uni- and Bipolar Hall IC Switches for Magnetic Field Applications. *Data Sheet* [online]. 2007 [cit. 2021-12-06]. Dostupné z: <https://www.gme.cz/tle4905l>
- [8] KOLLMORGEN. *Housed Direct Drive Rotary (DDR) Motors: Selection Guide* [online]. Kollmorgen Corporation, 2017 [cit. 2022-08-01]. Dostupné z: <https://www.kollmorgen.com/en-us/products/catalogs/kollmorgen-housed-ddr-motors-selection-guide/>
- [9] *7 Resources For Understanding Inertia and Inertia Mismatch* [online]. In: . [cit. 2022-08-01]. Dostupné z: <https://www.automate.org/blogs/7-resources-for-understanding-inertia-and-inertia-mismatch>
- [10] KOLLMORGEN. Drive.exe. *Software* [online]. (553) [cit. 2021-09-29]. Dostupné z: https://www.kollmorgen.com/en-us/developer-network/downloads/?field_product_family_tid_selective%5B%5D=3027&field_category_tid_selective%5B%5D=106&title=
- [11] *SERVOSTAR 601...620: Instructions Manual* [online]. [cit. 2022-08-01]. Dostupné z: <https://www.kollmorgen.com/en-us/products/drives/servo/serviced-drives/s600/#media>
- [12] KOLLMORGEN. *ASCII Object Reference* [online]. In: . [cit. 2022-03-08]. Dostupné z: <http://www.wiki-kollmorgen.eu/wiki/DanMoBilder/file/s300s700help/ASCII.HTML>
- [13] ASSAD, Fadi, Emma RUSHFORTH, Mus'ab AHMAD, Bilal AHMAD a Robert HARRISON. An Approach of Optimising S-curve Trajectory for a Better Energy Consumption. *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, 98-103. ISBN 978-1-5386-3593-3. Dostupné z: doi:10.1109/COASE.2018.8560587
- [14] ABB. IRB 1600. In: *Data sheet* [online]. [cit. 2022-01-16].
- [15] GRACO. Automatic Airless Spray Guns. *Instructions* [online]. () [cit. 2022-03-16]. Dostupné z: <https://www.graco.com/cz/cs/in-plant-manufacturing/product/288048.html#table-specifications>
- [16] STEJSKAL, Vladimír a Michael VALÁŠEK. *Kinematics and Dynamics of Machinery*. Přeložil Edvard NICHTBURGER. New York: Dekker, 1996. ISBN 0-8247-9731-0.

- [17] CHIAVERINI, S., B. SICILIANO a O. EGELAND. Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator. *IEEE Transactions on Control Systems Technology*. **1994**(2), 123-134. ISSN 10636536. Dostupné z: doi:10.1109/87.294335
- [18] WERU, Lawrence. *11 Animated Algorithms for the Traveling Salesman Problem* [online]. [cit. 2022-08-04]. Dostupné z: <https://stemlounge.com/animated-algorithms-for-the-traveling-salesman-problem/>
- [19] KOKTAN, Jan. *Simulace robotických a výrobních systémů*. Praha, 2021. Bakalářská práce. České vysoké učení technické v Praze. Vedoucí práce Jakub Jura.
- [20] COUMANS, Erwin a Yunfei BAI. *PyBullet, a Python module for physics simulation for games, robotics and machine learning* [software]. 3.2.1. 2016-2021 [cit. 2022-03-21]. Dostupné z: <http://pybullet.org>
- [21] ROS. *Introduction* [online]. [cit. 2022-08-14]. Dostupné z: <http://wiki.ros.org/ROS/Introduction>
- [22] CLEARPATH, Robotics. *Intro to ROS* [online]. [cit. 2022-08-14]. Dostupné z: <https://www.clearpathrobotics.com/assets/guides/melodic/ros/Intro%20to%20the%20RObot%20Operating%20System.html>
- [23] *Model robotu ABB IRB1600* [online]. GitHub [cit. 2022-01-18]. Dostupné z: https://github.com/ros-industrial/abb_experimental/tree/kinetic-devel/abb_irb1600_support
- [24] ALAG, Gunish. *Evaluating the Performance of Intel Realsense T265 Xsens Technologies B.V.* 2020. Internship Report.
- [25] BROWN, Jasper. *Testing the Realsense T265 Accuracy* [online]. In: . 2019 [cit. 2022-08-11]. Dostupné z: <https://jaspereb.github.io/TestingRealsenseT265/>
- [26] BELOUSOV, Boris. *Distance between rotations* [online]. In: . 2016 [cit. 2022-08-01]. Dostupné z: <http://www.boris-belousov.net/2016/12/01/quat-dist/>
- [27] UV mapping. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2022 [cit. 2022-08-10]. Dostupné z: https://en.wikipedia.org/wiki/UV_mapping

Značení

v	Rychlost
a	Zrychlení
j	Jerk (míra změny zrychlení)
p, q	kvaternion
E	Jednotková matice
1A, 1B, 2A, 2B, 3A...	Označení bodů trajektorie
$T_{a,b}$	Transformační matice
$S_{a,b}$	Matice směrových kosinů

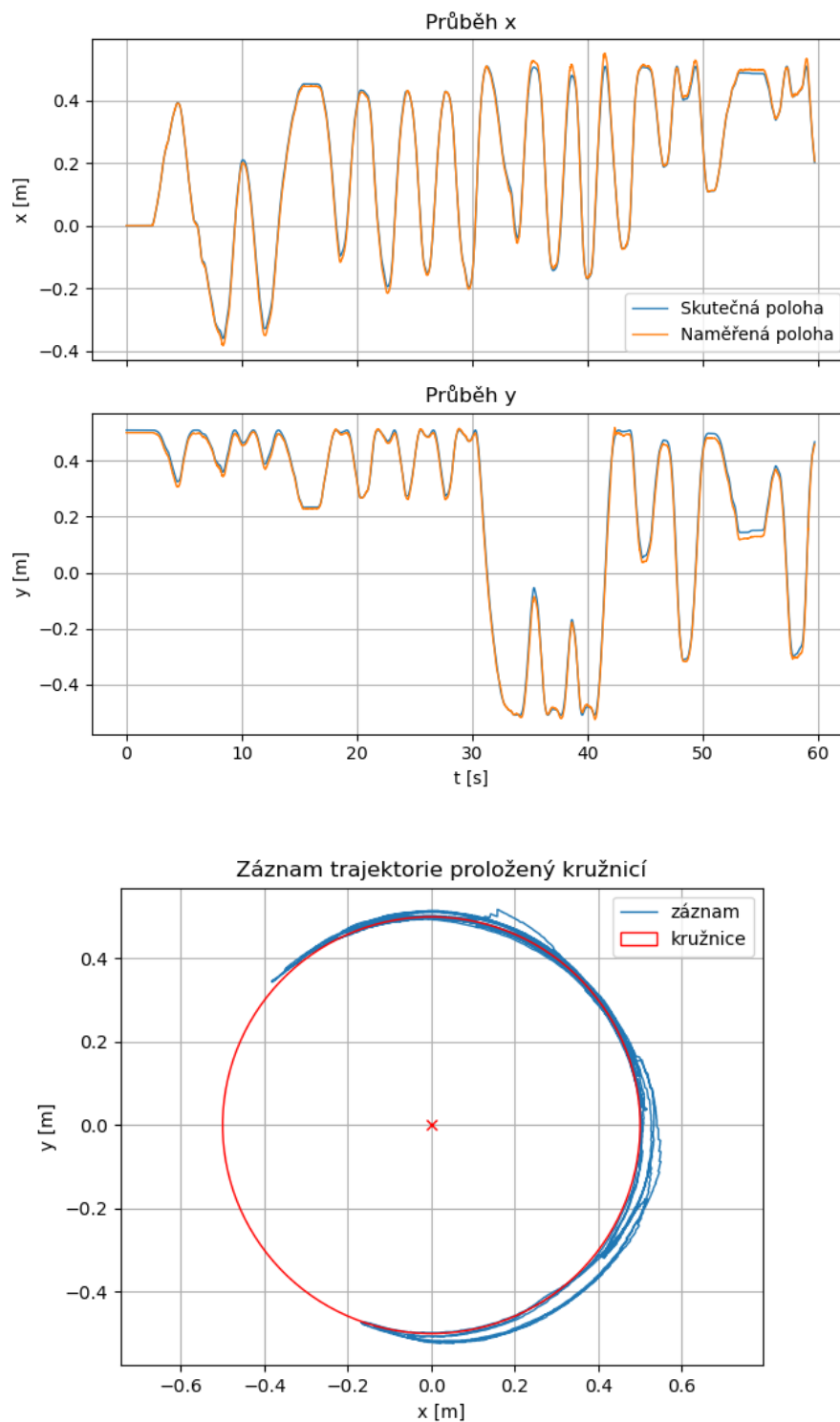
Seznam zkratk

IMU	Inerciální měřící jednotka (Inertial measurement unit)
SDK	Software development kit
FPS	Frames per second (snímků za sekundu)
VPU	Vision processing unit
TSP	Traveling salesman problem
SLAM	Simultaneous localization and mapping
VSLAM	Visual simultaneous localization and mapping
GPIO	General-purpose input/output
s.s.	Souřadnicový systém
RGB	Red, Green, and Blue

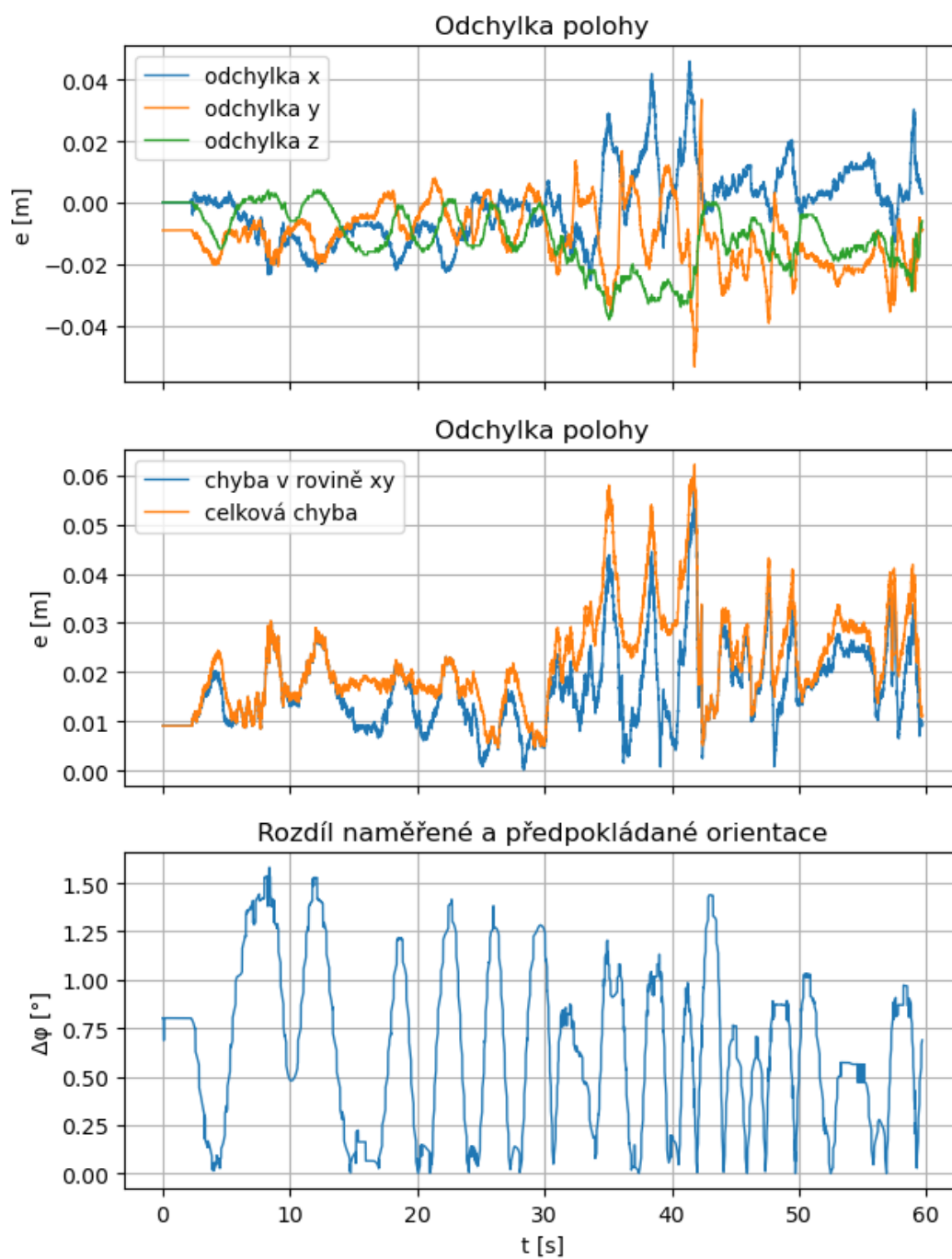
Přílohy

Příloha A: Měření přesnosti kamery Intel RealSense T265

Měření 1

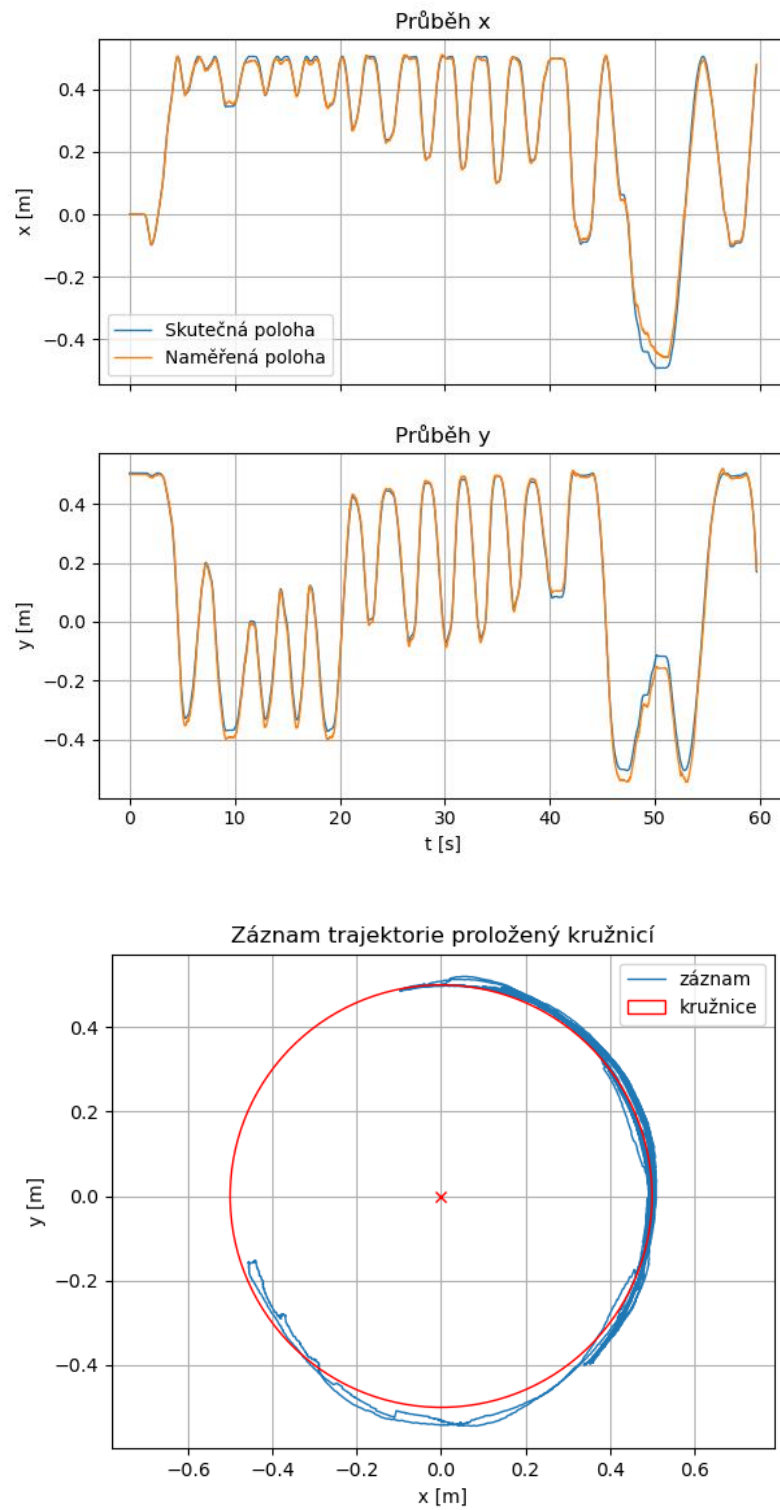


Graf 1: Trajektorie z měření 1, kamera se pohybuje na rameni 500 mm a směřuje po směru pohybu,

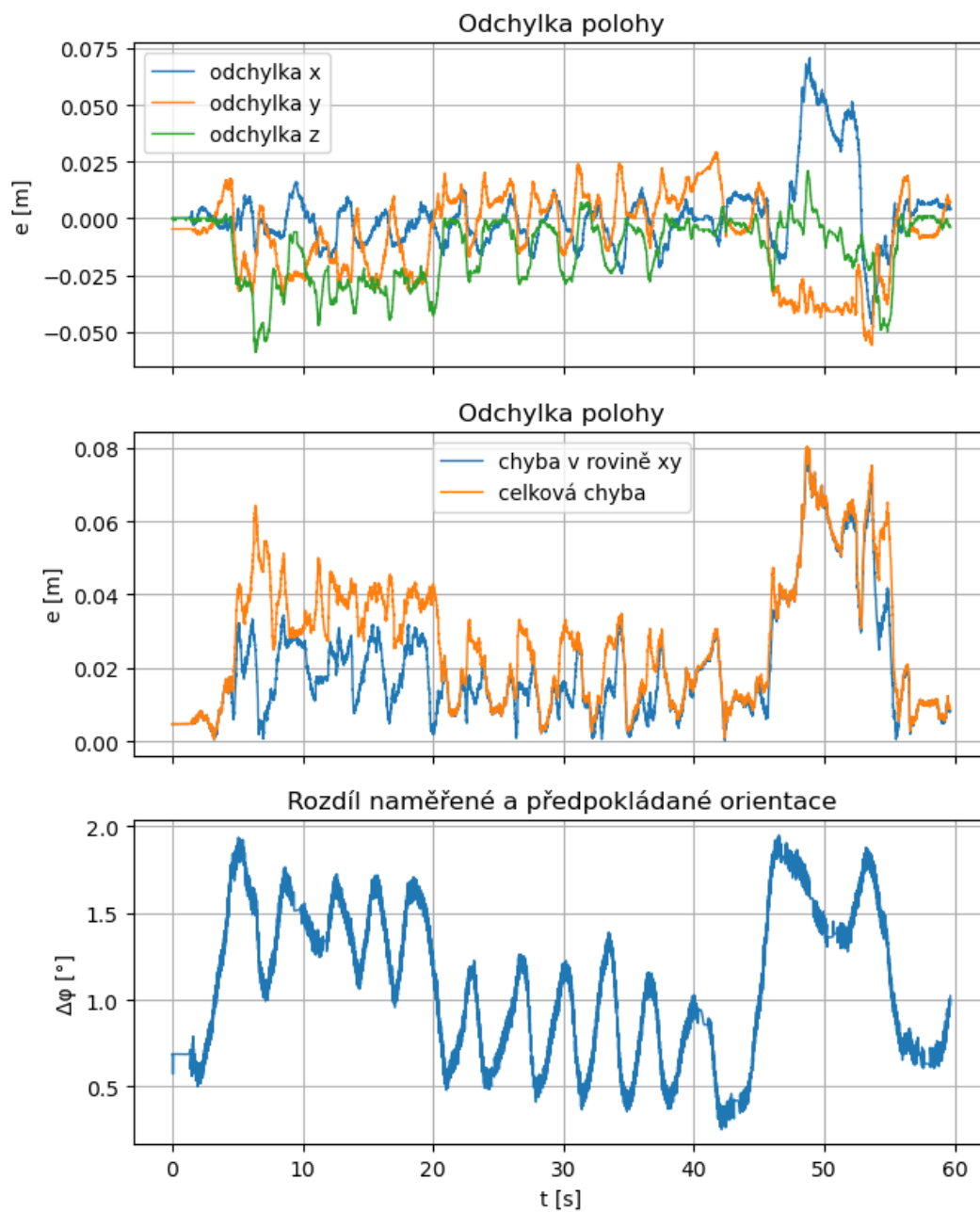


Graf 2: Průběhy jednotlivých odchylek v měření 1.

Měření 2

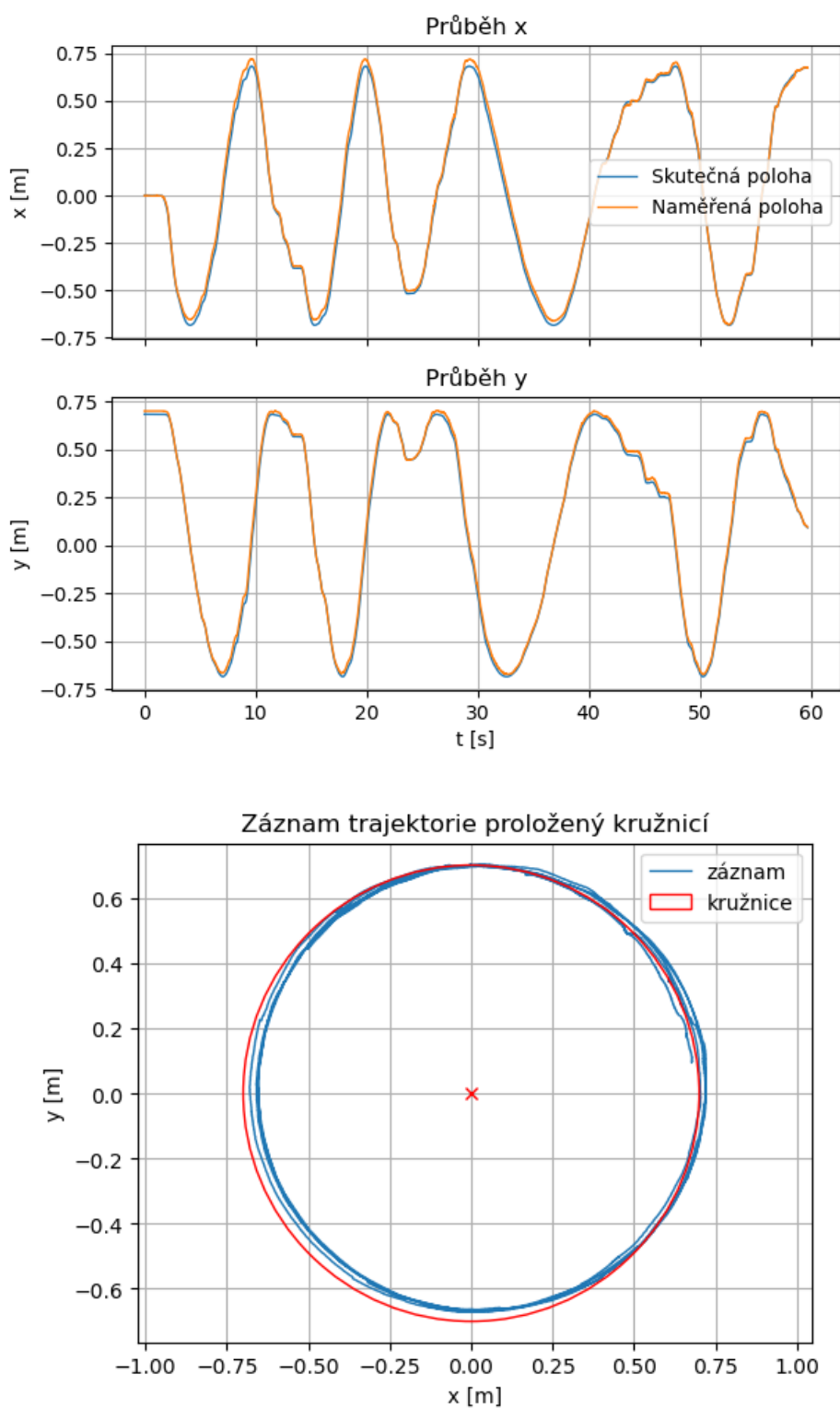


Graf 3: Trajektorie z měření 2, kamera se pohybuje na rameni 500 mm a směřuje ven.

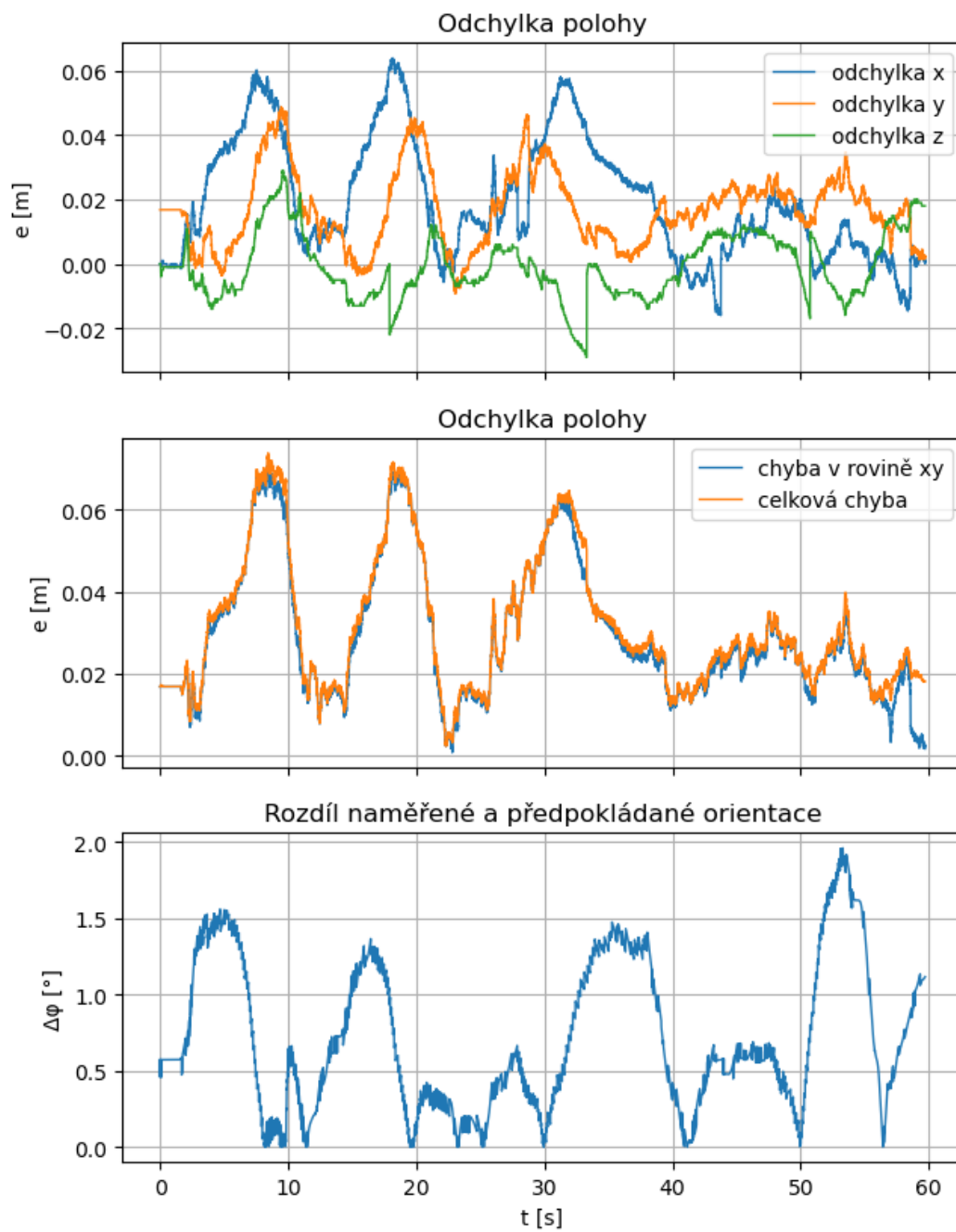


Graf 4: Průběhy jednotlivých odchylek v měření 2.

Měření 3

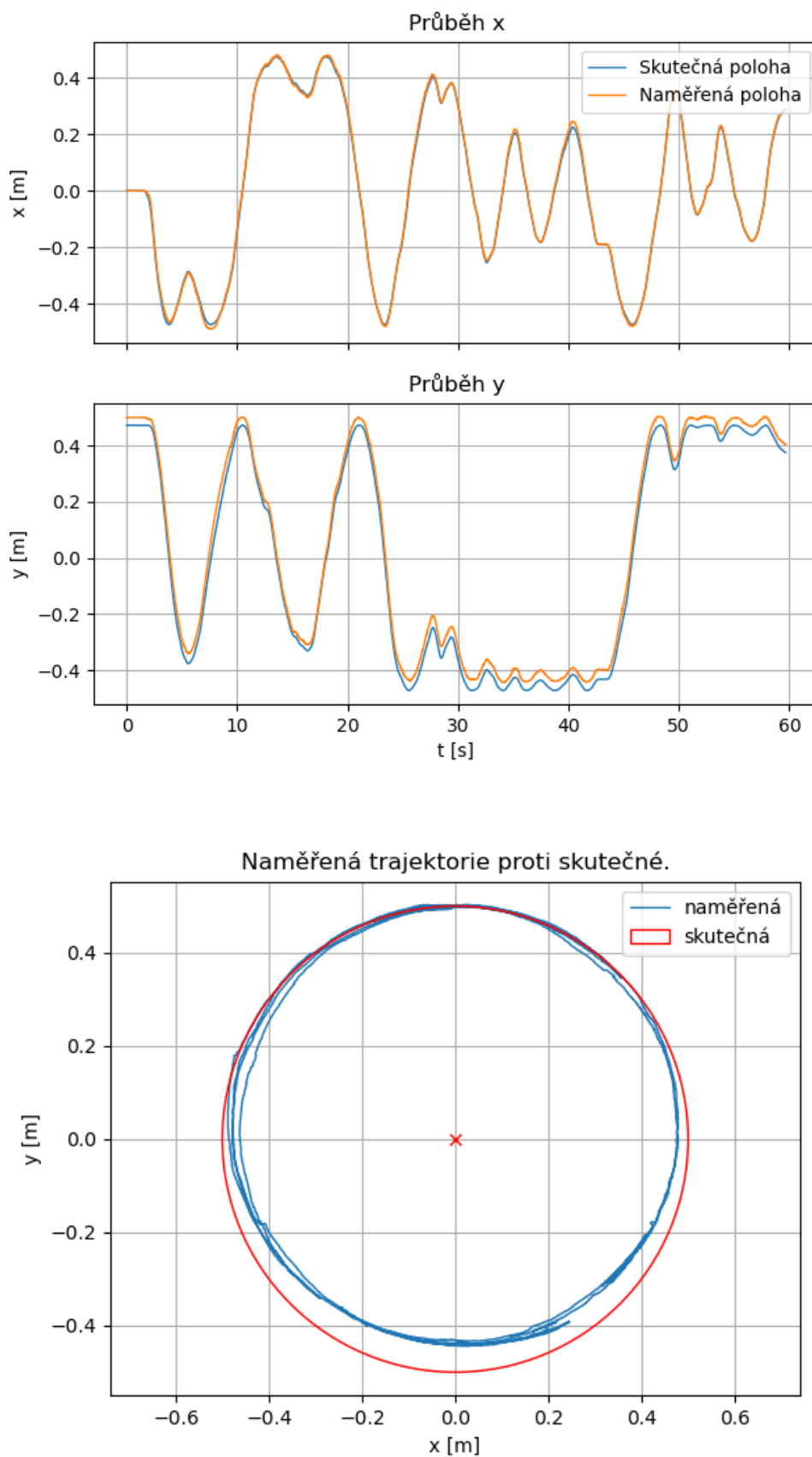


Graf 5: Trajektorie z měření 3, kamera se pohybuje na rameni 700 mm a mm a směřuje po směru pohybu,

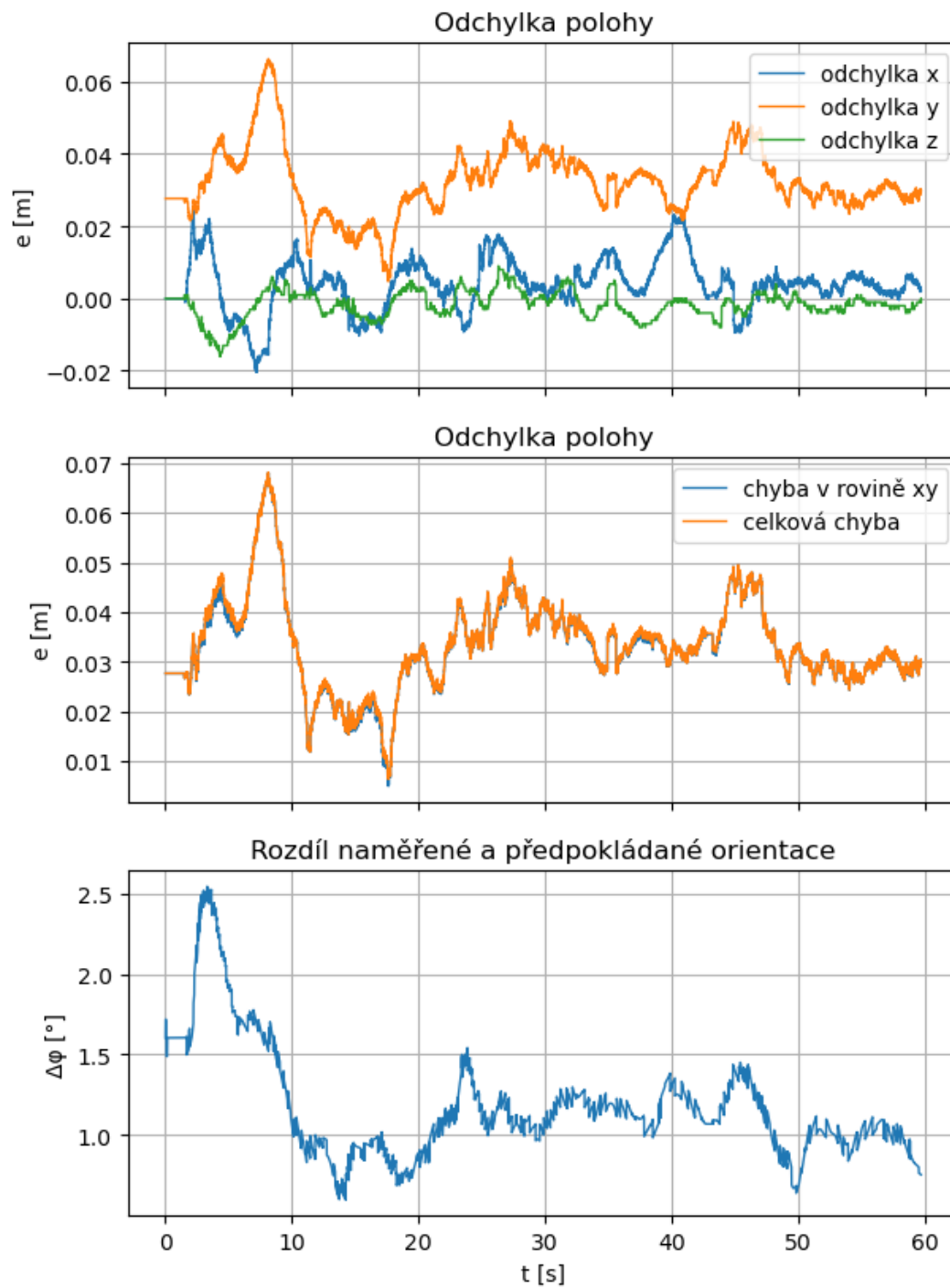


Graf 6: Průběhy jednotlivých odchylek v měření 3.

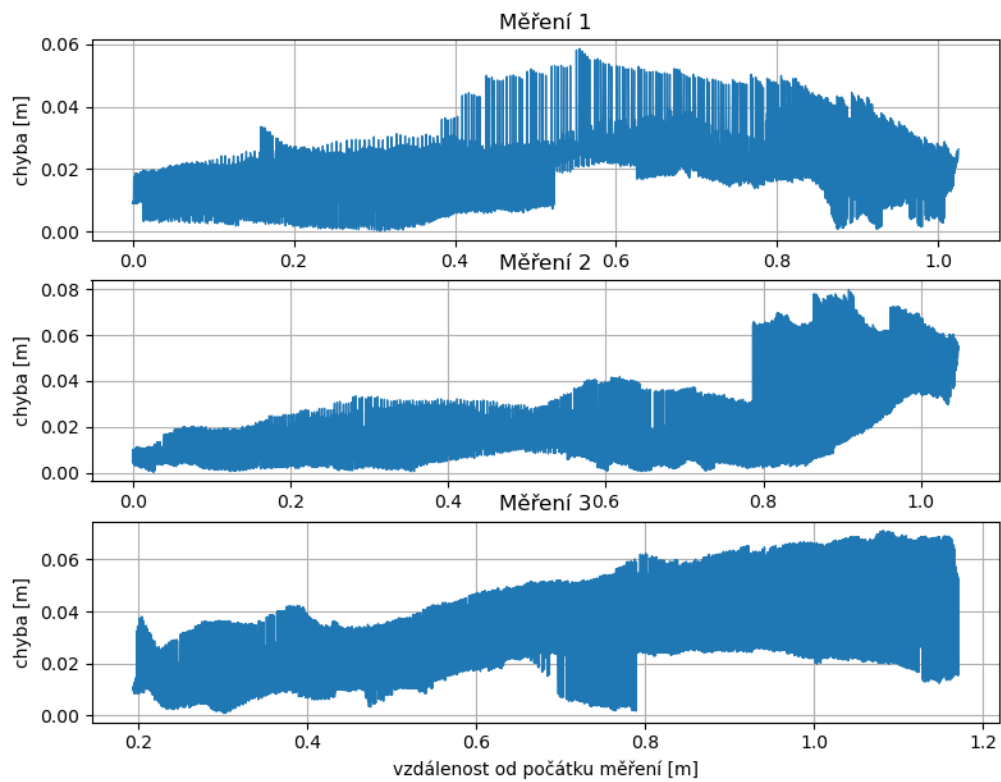
Měření 4



Graf 7: Trajektorie z měření 4, kamera se pohybuje na rameni 500 mm a směřuje dovnitř.



Graf 8: Průběhy jednotlivých odchylek v měření 4.



Graf 9: Závislost chyby polohy na vzdálenosti od počátku měření.

Příloha B: Záznam simulace

Záznam ze simulace imitačního lakování.

Příloha C: Zdrojový kód

- C1 Řízení polohovacího stolu
- C2 Nahrávání trajektorie
- C3 Zpracování dat a optimalizace
- C4 Simulace