



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROBOTICKÉ RAMENO S MODELÁŘSKÝMI SERVY

ROBOTIC ARM WITH MODEL SERVOS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR BOBČÍK

VEDOUcí PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2020

Zadání bakalářské práce



Student: **Bobčík Petr**
Program: Informační technologie
Název: **Robotické rameno s modelářskými servy**
Robotic Arm with Model Servos
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte způsob řízení modelářských serv, možnosti jejich ovládání počítačem a relevantní literaturu
2. Navrhněte jednoduchý strojek - robotické rameno sestavené z modelářských serv; též navrhněte způsob ovládání takového robotického ramene.
3. Popište a diskutujte možnosti navrženého řešení a jeho dosažitelné vlastnosti.
4. Navržený systém implementujte a demonstруйте jeho funkčnost na vhodné úloze.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 28. května 2020

Datum schválení: 27. května 2020

Abstrakt

Hlavním cílem této práce bylo postavit vlastní robotický manipulátor a jemu odpovídající uživatelské rozhraní, které ho bude ovládat. Vytvořil jsem tedy robotické rameno na počítači Raspberry Pi 3B+ s 5 digitálními servy a uchopovací mechanizmem v podobě klepet, ke kterému je přichyceno i laserové ukazovátko. Problematiku samotného uživatelského rozhraní jsem se rozhodl vyřešit vytvořením jednoduché webové stránky (kombinací HTML, PHP a JS), kterou jsem umístil na webový server Apache 2, nainstalovaný právě na Raspberry Pi. Do uživatelského rozhraní se mi podařilo zakomponovat jednak ovládání jednotlivých serv samostatně, tak i ovládání celého ramene pomocí inverzní kinematiky.

Abstract

The main goal of this thesis was to create my own robotic manipulator, which is controlled by its own user interface. I made a robotic arm based on computer Raspberry Pi 3B+ with 5 digital servos and a gripping mechanism in the form of a claw, to which I also attached a laser pointer. The user interface problematics was solved by creating a simple web page (combination of HTML, PHP and JS), which i placed into the Apache 2 web server (installed in Raspberry Pi). I successfully incorporated a way to control the servo one by one and how to control the whole arm with inverse kinematics into the user interface.

Klíčová slova

bakalářská práce, robotické rameno, robotický manipulátor, rameno, 3d tisk, servo, apache 2, server, GUI, inverzní kinematika, ovládání, raspberry pi

Keywords

bachelor's thesis, robotic arm, robotic manipulator, arm, 3d print, servo, apache 2, server, GUI, inverse kinematics, control, raspberry pi

Citace

BOBČÍK, Petr. *Robotické rameno s modelářskými servy*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík

Robotické rameno s modelářskými servy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Petr Bobčík
28. května 2020

Poděkování

Rád bych poděkoval svému vedoucímu práce prof. Dr. Ing. Pavlu Zemčikovi za jeho odborný dohled nad prací a za konzultace, které mi poskytl během vypracovávání této práce. Dále bych chtěl poděkovat své rodině za podporu, kterou mi v průběhu celého studia dávala.

Obsah

1	Úvod	2
2	Existující řešení robotických ramen	3
2.1	Robotické rameno PiArm	3
2.2	Robotické rameno Dexter ER2	5
2.3	Manipulátor Niryo One	7
3	Současné technologické prostředky	9
3.1	Modelářská serva	9
3.2	Řídící desky	12
3.3	Komunikace robota s počítačem	15
3.4	Geometrie manipulátoru	17
3.5	Program pro modelování 3D modelů	19
4	Analýza stavu a návrh řešení	22
4.1	Porovnání existujících ramen	22
4.2	Vymezení konceptu práce a stanovení cílů	23
4.3	Seznam částí k implementaci	24
4.4	Testy potřebné k ověření funkčnosti ramene	25
5	Fyzická realizace robotického ramene	26
5.1	Fyzická realizace robotického ramene	27
5.2	Uživatelské rozhraní pro ovládání manipulátoru	32
5.3	Komunikace uživatelského rozhraní s GUI	35
5.4	Dosažené vlastnosti ramene	37
5.5	Testování robotického manipulátoru	37
6	Závěr	41
	Literatura	44

Kapitola 1

Úvod

V dnešní době je slovo robotické rameno velmi rozšířený pojem. Jedná se o pohyblivé zařízení, které nahrazuje těžkou, ať už fyzicky náročnou práci, kde například umožňuje přesun těžkých předmětů, nebo práci náročnou na čas. Příkladem využití mohou být například výrobní linky, operace na dálku v lékařství, na mezinárodní vesmírné stanici jako robotické chapadlo, práce s pyrotechnikou a mnoho dalších. Ve většině případů, pokud se opomenou automatizované výrobní linky, kde se ramena řídí bez zásahu uživatele, jsou tato ramena ovládána člověkem vzdáleně, například z počítače. Vyskytuje se tedy potřeba jednoduchého a efektivního řízení či ovládání, aby nedošlo k poškození zařízení nebo ještě hůře k poranění či usmrcení člověka.

Cílem této práce je tedy navrhnout si jedno takové zařízení, které by mělo obsahovat alespoň pět modelářských serv, a pokusit se o navržení uživatelského rozhraní, které umožní dané rameno ovládat a nabídne i jednoduché a v rámci možností i intuitivní prostředí. Cílem této práce není pokus překonat již existující řešení, kterých je nepřeberné množství, ale vyzkoušet si postavit jedno takové zařízení sám a udělat si obrázek o náročnosti celého realizačního postupu. Robotické rameno je také navrženo z běžně dostupných zařízení, která je možné koupit v obchodech či na e-shopu, a replikace takového zařízení nebude, v domácím prostředí se základním technickým vybavením, problematické.

Pro toto téma bakalářské práce jsem se rozhodl, protože mám již od mala zálibu v robotech a robotických ramenech. Již delší dobu jsem se chystal jedno takové rameno sestavit sám, ale chybělo mi pro něj využití, které jsem našel v této práci.

Tato práce je rozdělena do čtyř hlavních kapitol, kterými jsou Existující řešení robotických ramen, Současné technologické prostředky, Analýza stavu a návrh řešení a Fyzická realizace robotického ramene. Kapitola Existující řešení robotických ramen popisuje mnou vybrané existující řešení robotických manipulátorů, která mi přišly zajímavá pro tuto práci. V následující kapitole Současné technologické prostředky jsou popsány prvky, ze kterých se robotické rameno skládá, jako je pohon, řídicí deska a tak dále. Po přečtení této kapitoly bude mít čtenář přehled o tom, z čeho se manipulátor skládá. Předposlední kapitola Analýza stavu a návrh řešení poukazuje na to, co vše je nutné udělat pro postavení mého robotického ramene, vymezuje cíle, které jsem si stanovil, a poskytuje také seznam důležitých testů pro to, aby šlo robota použít. Poslední kapitola Fyzická realizace robotického ramene realizuje mnou stanovené body z kapitoly Analýza stavu a návrh řešení. Na závěr této kapitoly je také provedeno testování nad výslednou realizací manipulátora.

Kapitola 2

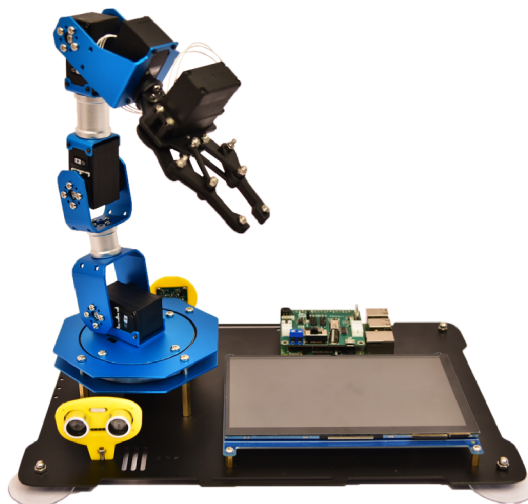
Existující řešení robotických ramen

V této kapitole jsou popsána existující řešení robotických ramen. Existuje jich nepřehledné množství, ale vzhledem k maximálnímu rozsahu bakalářské práce byl prostor k popisu pouze tří robotických ramen. Z tohoto důvodu jsem vybral taková, která mají bezprostřední vztah k zadání mé práce.

2.1 Robotické rameno PiArm

Jedná se o variantu manipulátora ve formě stavebnice, kterou si uživatel může zakoupit a sestavit sám doma. Robot umožňuje pohyb až v šesti různých stupních volnosti [18], kde je každý stupeň tvořen jedním digitálním modelářským servem, s možností otočit se až o 240 stupňů. Je vhodný jak pro uživatele, kteří neumí programovat [18], tak i pro ty, kteří jsou v tomto směru zbláhli.

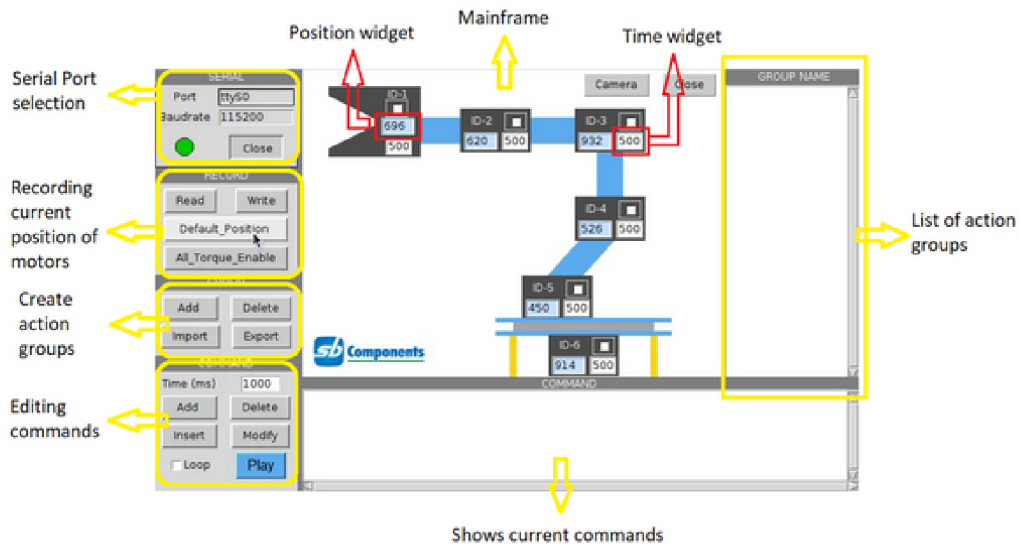
Konstrukce celého ramene je kovová [18] a skládá se ze vzájemně propojených prvků. Kovová struktura poskytuje ramenu velmi robustní strukturu, jak uvádí autor Phil King [38]. Konec ramene tvoří úchopový mechanismus (klepeto), se kterým je opět možné manipulovat. Celé rameno je umístěno na kovové podložce, která je doplněna o gumové přísavky k možnosti uchycení například ke stolu.



Obrázek 2.1: Robotické rameno PiArm.¹

K řízení je využit počítač Raspberry Pi, přesněji model 3B+, který umožňuje ovládat rameno pomocí sériové linky a to jak přes USB [20], tak bezdrátově přes bluetooth [18]. Jednou z možností, jak robota ovládat [20], je uživatelské rozhraní. To je zapotřebí nejprve stáhnout na počítač Raspberry Pi z GitHubu, které je přímo pro PiArm. Komunikace mezi ramenem PiArm a zařízením, na kterém je uživatelské rozhraní stažené, probíhá přes sériovou linku [20]. Ovládání přes GUI je možné buď přes notebook, nebo přes telefon pomocí VPN [18]. Na obrázku 2.2 je toto grafické uživatelské rozhraní znázorněno.

Rozhraní uživateli nabízí možnost ovládat každé servo zvlášť. Uživatel u něj může nastavit jak pozici, do které se má přesunout, tak čas, za který se má do nové pozice přemístit. [20]. Při nastavení serv si uživatel může uložit aktuální nastavení do takzvané skupiny [38] a následně všechny uložené skupiny spustit. Zběhlý programátor si může robota naprogramovat sám [38] pomocí knihoven, které jsou psány v jazyce Python a poskytnuty ve formě open source.



Obrázek 2.2: Uživatelské rozhraní pro PiArm.²

Rameno PiArm je vybaveno senzory, jako [18] ultrazvukový senzor a IR senzor, které umožňují vyšší interaktivitu robota s okolím a s uživatelem. Dále je možné připojit kameru nebo využít dotykový LCD monitor a to v 5 nebo 7 palcovém provedení.

Sumarizace vybraných vlastností robotického manipulátoru PiArm:

- Šest stupňů volnosti
- Digitální modelářská serva
- Hliníková konstrukce robota

¹Obrázek byl převzat z webové stránky: <https://magpi.raspberrypi.org/articles/piarm-review>

²Obrázek byl převzat z webové stránky: <https://shop.sb-components.co.uk/blogs/piarm-lessons/lesson-2-controlling-piarm>

- Možnost přidání senzorů pro interakci s uživatelem
- Vhodný jak pro začátečníky v programování i pro pokročilé
- Poskytuje vlastní grafické rozhraní
- Ovládání je možné pomocí notebooku, joysticku nebo VPN v telefonu
- Zdrojový kód i HW jsou open source
- Robot je ve formě stavebnice

Robotické rameno PiArm je sice dodáváno s řídicí deskou Raspberry Pi 3B+, ale je kompatibilní se všemi existujícími variantami [18].

2.2 Robotické rameno Dexter ER2

Dexter ER2 je, stejně jako PiArm, komerčně dostupný, avšak není možné ho sehnat jako stavebnici, ale pouze jako hotový [41] a složený produkt. Toto řešení přináší výhodu, že je možné manipulátor okamžitě použít a není třeba ztrácet čas jako skládáním.

Konstrukce je stejně jako PiArm kovová [41] a váží 1500 gramů. Konec ramene tvoří uchopovací mechanismus, který dokáže uzvednout objekt široký maximálně 55 mm a s váhou až 50 gramů. S Dexter ER2 je možné manipulovat v pěti osách, kde každou osu tvoří jeden ze dvou různých servo motorů s možnou rotací 180 stupňů. První čtyři serva mají uvedený krouticí moment 15 kg/cm, zatímco krouticí moment serva pro koncový a uchopovací mechanismus je 7 kg/cm. S robotem je možné manipulovat [41] s poloměrem 320 mm.



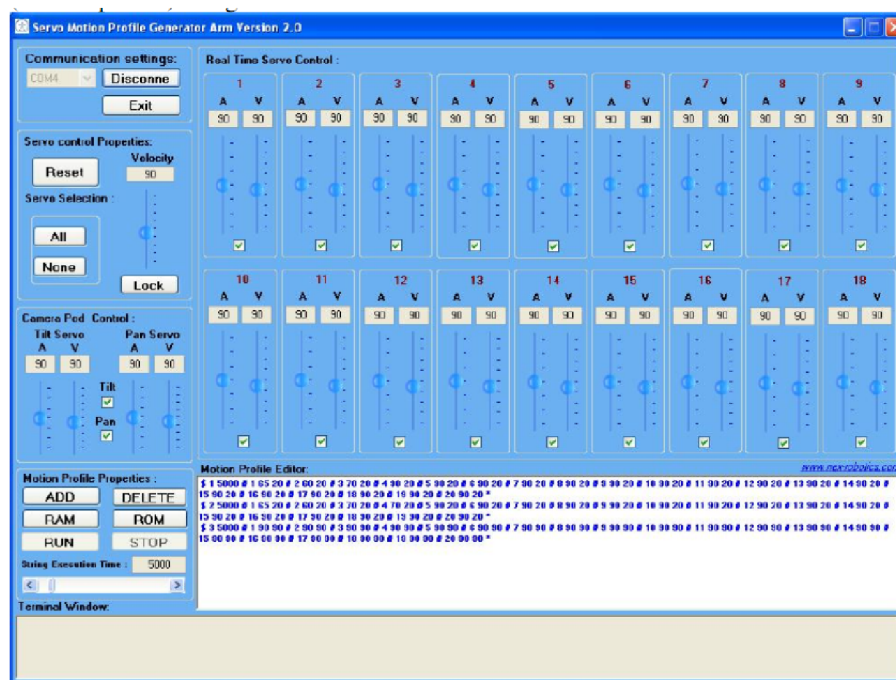
Obrázek 2.3: Robotický manipulátor Dexter.³

Řídicí deska, kterou Dexter ER2 využívá, je založena na ATmega640 [41] a uživateli nabízí piny pro připojení až devíti serv. Pro ovládání robotického ramene uživatelem, je

³Obrázek byl převzat z webové stránky: <http://www.nex-robotics.com/robotic-arm/robotic-arm/pdf.html>

možné využít již předchystané uživatelské rozhraní, které je zapotřebí nainstalovat na koncové zařízení, které ovládá rameno. Tímto zařízením může být [41] počítač nebo notebook. Zařízení, na kterém je spuštěno toto uživatelské rozhraní, komunikuje se samotným robotem pomocí [41] sériové komunikace přes USB.

Uživatelská rozhraní umožňují ovládat až 32 serv [41]. Všechna serva jsou ovládána v reálném čase a je možné nastavit jak polohu samotného serva, tak jeho rychlost. Uživatel si může vygenerovat svoji pohybovou sekvenci a následně i spustit. Součástí rozhraní je také možnost uložit sekvenci na paměť RAM nebo ROM v robotickém rameni. Obrázek 2.4 znázorňuje toto uživatelské rozhraní pro robota Dexter ER2.



Obrázek 2.4: Uživatelské rozhraní pro rameno Dexter ER2.⁴

Obrázek 2.4 znázorňuje uživatelské rozhraní pro robotické rameno Dexter ER2, které je popsáno výše.

Sumarizace vybraných vlastností robotického ramene Dexter ER2:

- Pět stupňů volnosti
- Kovová konstrukce
- Operační rádius 320mm
- Váží 1500g
- Dokáže uchopit předmět 55mm široký a 50g těžký
- Není možné přidat senzory pro interakci s uživatelem

⁴Obrázek byl převzat z webové stránky: <https://dokumen.tips/documents/dexter-er-2-robotic-arm-2011-09-30.html>

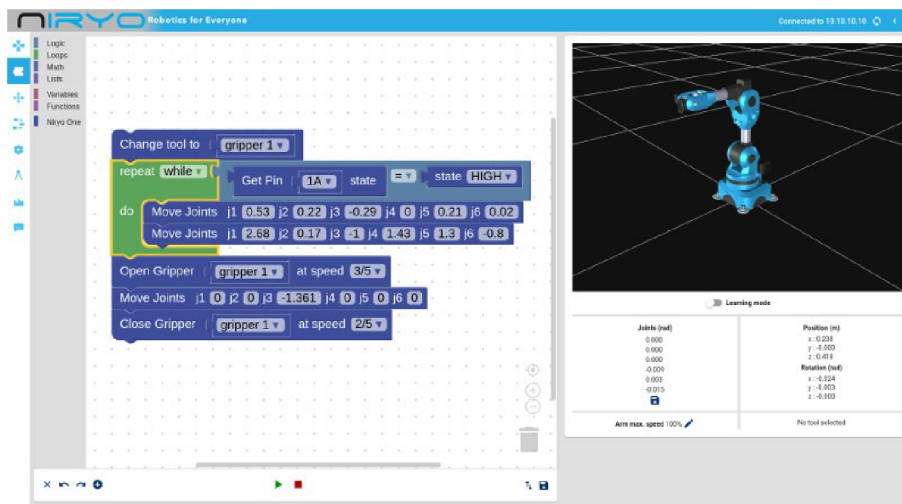
- Poskytuje vlastní grafické rozhraní
- Ovládání pomocí notebooku nebo počítače
- Komunikace s ramenem přes USB
- Nemá ve formě stavebnice

Součástí robotického ramene Dexter ER2 je také ohebný plát [41] s kruhovým a čtvercovým souřadným systémem, na který je možné rameno umístit. Na tyto souřadnice je následně možné do středu vložit robotické rameno a také předměty, se kterými manipuluje.

2.3 Manipulátor Niryo One

Jedná se o variantu, která je dostupná jak v komerční podobě, tak ve formě open source, kde je možné stáhnout modely celého robotického manipulátoru a zhotovit jej doma. Toto rameno umožňuje pohyb až v šesti různých osách a uchopovací mechanismu pro manipulaci s předměty. Uživatel může rameno ovládat ve třech [21] různých režimech:

- **Učící režim:** Cílem tohoto režimu je přesunout manipulátor do určité polohy pomocí vlastních rukou a tím ramenu sdělit, kam se má přesunout.
- **Programovací/kódový režim:** Umožňuje pomocí počítačové aplikace Niryo One Studio naprogramovat jeho chování a pohyby s využitím blokového programování, které se podobá například Scratchi, čili vnořovat bloky (obrázků) do sebe a tím tvořit program. Aplikaci je možné vidět na obrázku 2.5. Komunikace v tomto režimu je možná [17] přes Wi-Fi nebo ethernet.



Obrázek 2.5: Programování ramene Niryo přes Niryo One Studio.⁵

- **Režim ovládní xbox ovladačem:** Varianta, při které uživatel nemusí mít vůbec zkušenosti s programováním, ale pouze pomocí ovladače manipuluje s ramenem.

⁵Obrázek byl převzat z webové stránky: <https://niryo.com/niryo-one/>

Niryo nabízí také API pro vývojáře, kteří si mohou robota naprogramovat sami nebo mohou využít ROS (Robot Operating System), který je možné stáhnout přes git jako open source.

Po konstrukční stránce je robot vytvořený na 3D tiskárně a je k němu možné jako manipulační mechanismus upevnit [21] jedno ze tří klepet (malé, větší a nastavitelné), vakuovou přísavku nebo elektromagnet. Váha předmětu, který může zvednout je různá a je dána typem použitého uchopovacího mechanismu. Například pomocí vakuové pumpy [13] lze uzvednout předmět o váze 350 g a s elektromagnetem až 500 g. Váha robotického ramene je 3300 g [13]. Pro uchycení robota ke stolu je možné využít přichystaných gumových přísavek.



Obrázek 2.6: Robotické rameno Niryo.⁶

Rameno Niryo One je také možné rozšířit o různé rozšiřující sady [21], jako je například přepravní pás, ovládací pult, vakuová pumpa a další. Niryo One je také možné uživatelsky rozšířit pomocí [21] Raspberry Pi nebo Arduino desky a přidat tak možnost souběžného ovládání více Niryo One robotů.

Sumarizace vybraných vlastností robotického ramene Niryo One:

- Šest stupňů volnosti a kostra z 3D tisknutelného materiálu
- Možnost vyměnit uchopovací mechanismus
- Váží 3300 g a hmotnost zvedaného předmětu až 500 g dle uchopovacího mechanismu
- Možnost přidat novou funkcionalitu, přidáním Arduino nebo Raspberry Pi desky
- Možnost ovládání přes joystick, Niryo One Studio a nebo vlastní rukou
- Poskytuje vlastní aplikaci pro programování Niryo One Studio
- Komunikace ramene s Niryo One Studio přes Wi-Fi nebo ethernet

Rameno Niryo One je původně navrženo pro edukativní účel a demonstraci průmyslových úkolů, například ukázka automatizované linky.

⁶Obrázek byl převzat z webové stránky: <https://ozrobotics.com/shop/niryo-one-open-source-collaborative-robot/>

Kapitola 3

Současné technologické prostriedky

Aby bylo možné vytvořit vlastní robotický manipulátor, je zapotřebí provést analýzu již existujících komponent, vývojových programů, možností ovládání robota, designu a dalších. Na základě tohoto průzkumu je poté možné poměrně snadno zvolit, k čemu je daný robot určen, pomocí kterých komponent se bude hýbat, a samotný vývoj se velmi urychlí. Obsahem této kapitoly je tedy popis existujících komponent, metod a programů, které lze pro návrh ramene použít. Z důvodu maximálního rozsahu této práce zde nejsou popsány veškeré programy a komponenty, ale pouze ty, které mají bezprostřední vztah k zadání této práce. Vhodným příkladem může být například pohon robotického ramene, kterých může být spousta. Pohonem může být modelářské servo, krokový motor, hydraulický píst a mnoho dalších, ovšem popsány budou pouze modelářská serva, která jsou popsána v kapitole “Modelářská serva”.

3.1 Modelářská serva

Servo je elektromagnetické zařízení, které podle hodnoty vstupního signálu, nejčastěji PWM, umožňuje otáčet hřídelí. Zjednodušeně se dá říci, že se jedná o motor, který umožňuje kontrolovat přesnou pozici hřídele [27].



Obrázek 3.1: Modelářské servo.¹

Díky této vlastnosti se hojně používá například u RC modelů letadel, vrtulníků a obecně robotů, kde malá akce způsobí velkou reakci. Serva se řadí [27] do takzvaných closed-loop systémů (uzavřený systém), který ovládá zařízení (servo) pomocí kladné zpětné vazby.

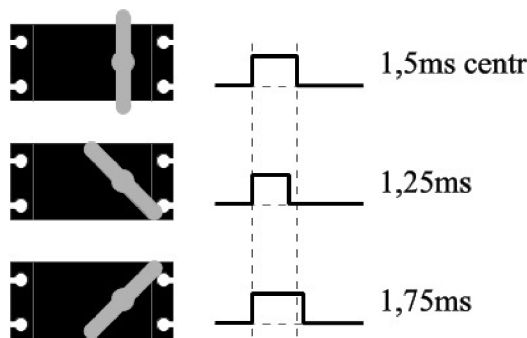
Serva je možné rozdělit do skupin podle typu napájení na **stejnoseměrná (DC) a střídavá (AC)** a podle typu ovládání na **analogová a digitální**. Obecně se serva skládají ze čtyř hlavních bloků, jako je **motor** (AC nebo DC), **zpětná vazba** (potenciometr), **řídící elektronika** a **převod**. Převod napomáhá snížení rychlosti serva [27], ale naopak se zvýší jeho krouticí moment.

Činnost serva

Na vstup serva se přivede signál, který spustí monostabilní klopný obvod, a ten následně vygeneruje signál. Tento signál odpovídá aktuálnímu pootočení hřídele, ovšem je opačné polarity než vstupní signál. Aby se mohlo servo pootočit, dojde k výpočtu rozdílu signálu, který je do serva přiveden, se signálem, který odpovídá aktuálnímu pootočení hřídele. Součástí samotného servomotoru je i zpětnovazební potenciometr, který informuje monostabilní klopný obvod o změně pozice. Z toho vyplývá, že potenciometr se musí pootočit ve stejný okamžik jako samotná hřídel [29].

Vstupní signál

Servo očekává, že na vstupu dostane signál každých 20 ms [42]. Pokud bude hodnota tohoto vstupního signálu korespondovat s aktuálním pootočením hřídele, tak ke změně pootočení nedochází a servo se snaží udržet na aktuální pozici. Délka samotného signálu (nejčastěji PWM) určuje, jak moc se hřídel pootočí. V dnešní době se často používají serva, u nichž se délka vstupního signálu pohybuje od 1 ms do 2 ms, kde 1 ms označuje jednu nejkrajnější polohu hřídele a 2 ms druhou.



Obrázek 3.2: Znázornění řízení serva na základě délky signálu.²

V případě, kdy by se vnější síla snažila působit na servo, které je pod napětím a snaží se udržet nějakou pozici [42], která odpovídá hodnotě vstupního signálu, tak se servo začne bránit. Tato obrana ovšem přetrvá pouze do té doby, dokud není překročena maximální hodnota krouticího momentu.

¹Obrázek byl převzat z webové stránky: <https://www.jazrider.com/tamiya-futaba-rc-truck-operation-set-4ywd-mfc01mfc03-5000mah-nimh-battery-chargerfutaba-s3003-servo-x3-p-90024035.html>

²Obrázek byl převzat z následující webové stránky <http://vlastikd.webz.cz/bast1/serva.htm>

Typy modelářských serv

Modelářská serva je možné sehnat ve dvou základních provedeních. [29] Digitální a analogové.

Analogová

Analogové servo je starší než servo digitální. Pracuje s 50Hz signálem. Typickým problémem, které s sebou přináší, [29] je náchylnost k teplotním změnám a změnám napájecího napětí, kde nemá tu stejnou nulovou pozici. Všechna dnešní velká serva jsou převážně analogová.

Digitální

Oproti výše uvedenému analogovému servu, které pracuje se signálem s frekvencí 50 Hz, pracuje digitální servo s frekvencí 200 Hz-300 Hz, což napomáhá redukci takzvaného mrtvého pásma (v angličtině dead-band), poskytuje rychlejší reakce, stabilnější kroučící moment, [29] přesnost a další. Dalším rozdílem oproti analogovým servům je schopnost některých digitálních serv nechat se naprogramovat podle uživatele, například rychlost, středový bod a další. [43] Ačkoliv jsou serva označována jako “digitální”, tak v sobě obsahují mikrokontroler, který ovládá servo opět pomocí PWM signálu, jak je tomu u analogového serva. Problémem digitálních serv [29] je jeho cena, která může být dvakrát vyšší, než u analogových serv.

Pulzně šířková modulace

Pulzně šířková modulace neboli PWM je metoda, pomocí které je možné generovat analogový signál za použití digitálního signálu. Skládá se ze dvou částí: Střída (anglicky duty cycle) a frekvence. Střída [35] popisuje množství času z celé periody, ve které se na výstupu objevuje logická úroveň 1 oproti logické úrovni 0. Frekvence poté udává, jak často dochází k přepínání mezi logickou úrovní 1 a 0. Střída nabývá hodnoty 0-100 procent, kde 0 znamená, že na výstupu je minimální napětí, a 100 značí maximální napětí. Pokud napájecí zdroj poskytne napětí 12 V, tak pro hodnotu střídy rovnou 20 % bude na výstupu 2,4 V [31], při střídě 50 % získáme 6 V a tak dále. PWM je možné uplatnit při řízení otáček motoru, jasu LED a v mnoho dalších.

Řízení modelářských serv

Aby byl zajištěn pohyb samotných servomotorů, je nezbytné, kromě samotného napájení, připojit pin, který bude obsahovat impulzní signál k otáčení hřídele. Tímto signálem může být například PWM (Pulse Width Modulation) [27], kterému se nastaví odpovídající střída. Pro řízení modelářských serv je podstatná pouze šířka tohoto signálu, tudíž frekvence s periodou signálu nemají na činnost serva žádný vliv. Nejjednodušší cestou, jak servo propojit s PWM signálem je připojení řídicího vstupu servo motoru na pin desky (řídicí desky), který na svém výstupu umožňuje nastavit PWM signál. V některých situacích se při řízení servomotoru může vyskytovat drobné chvění (jittering) servomotoru.

Metoda, která napomáhá eliminovat toto chvění, je generování pwm pomocí dma (Direct memory access). Jedná se o metodu, která umožňuje periferiím přenášet data přímo do nebo z paměti, aniž by byla vyžadována velká asistence CPU [1]. Tato metoda umožňuje

efektivnější využití přerušení, zvýšit propustnost a pomocí eliminace FIFO bufferů také snížit HW nároky. Pokud nastane událost, která vyžaduje DMA přenos, tak upozorní DMA kontrolér, který zašle signál CPU s požadavkem na využití sběrnice. CPU následně dokončí svoji rozpracovanou úlohu [1], umožní použití sběrnice a na konec přenosu si sběrnici opět zabere. CPU musí tedy čekat, dokud nebude sběrnice opět volná. Tento typ řízení je poměrně rychlý a oproti jiným metodám nevyžaduje téměř žádné zdroje CPU. PWM signál, který je pomocí DMA vytvořen, je stabilní a bývá na velkém rozlišení. Knihovna, která umožňuje tento typ řízení, se jmenuje `rpio-pwm`.

Knihovna `rpio-pwm`

`Rpio-pwm` je open source knihovna, psaná v jazyce Python, určená pro řídicí desky Raspberry Pi [33]. Knihovna umožňuje ovládat více servomotorů, připojených k GPIO pinům pomocí DMA, s přesností $1 \mu\text{s}$. Pro ovládání jednotlivých výstupů (například modelářských serv) stačí, aby uživatel zaslal příkaz, jakému servu se má PWM nastavit a jakou hodnotu chce nastavit, tedy jakou chce střídu. Serva pro řízení potřebují typicky pulz od 0,5 ms do 2,5 ms v logické úrovni 1 a jeho perioda je typicky 20 ms. DMA řadič tedy alokuje časový prostor 2,5 ms, kde na začátku každého časového úseku je nastaven výstup serva na logickou hodnotu 1 a na konci se překlápí zpět na hodnotu 0. Tento proces se opakuje každých 20 ms.

Knihovna `pigpio`

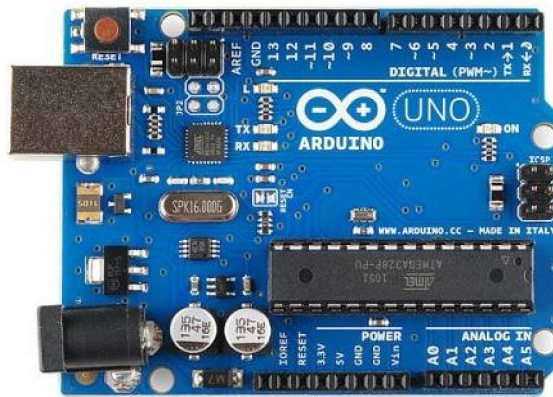
Další knihovnou, kterou je možné pro účely ovládání Raspberry Pi použít, je knihovna `pigpio`. `Pigpio` je napsaná v jazyce C [22] i v jazyce Python a podporuje všechny verze Raspberry Pi. Umožňuje ovládat vstupní i výstupní GPIO piny. Oproti knihovně `rpio-pwm`, která je popsána výše, umožňuje spustit daemona, poskytujícího rozhraní pro roury a sockety. V případě, že například Raspberry Pi neobsahuje dostatek SW PWM pinů, je možné simulovat PWM signál HW cestou. Oproti SW variantě ovšem přestane dané nastavení být aktivní po ukončení běžícího scriptu, který danou změnu provedl. Tento nedostatek může být eliminován [23] pomocí již zmíněného daemona, se kterým `pigpio` komunikuje, na němž se mohou vytvářet běžící scirpty.

3.2 Řídící desky

K tomu, aby bylo možné robotický manipulátor ovládat, je nezbytné zvolit řídicí desku, která to umožní. Možností, jak robotický manipulátor řídit, je spousta, avšak jsem limitován rozsahem této práce. Z toho důvodu jsem se rozhodl popsat Arduino UNO, ESP32S a Raspberry pi.

Arduino UNO

Jedná se o open-source mikrokontrolerovou desku, založenou na mikrokontroléru ATmega328P. Tento mikrokontroler běží na frekvenci 16 MHz a pracuje na 5 V úrovni. Deska nabízí 14 digitálních vstupně-výstupních pinů, z čehož je 6 PWM, 6 analogových pinů a pinhead ICSP. [14]



Obrázek 3.3: Vývojová deska Arduino Uno.³

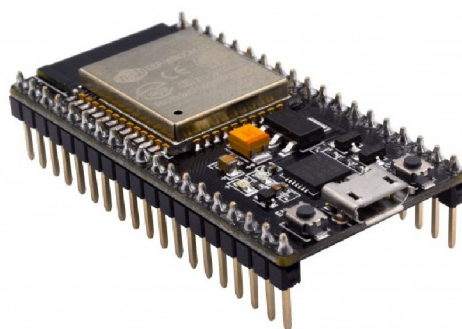
Výhodou této desky je program Arduino IDE, který umožňuje jednoduché programování mikrokontroléru za pomoci velkého množství knihoven a široké komunity, která s nimi pracuje. Další výhodou je 6 PWM [14] výstupů, které je možné využít až na 6 různých motorů. Nevýhodou je absence Wi-Fi modulu i Bluetooth modulu, což znamená, že je nezbytné využít externí modul a tím pádem zabrat některé piny, které by mohly být využity pro jiné účely.

Arduino UNO Wi-Fi Rev 2

Rozšířená základní verze Arduino Uno, které stále zůstává frekvence mikrokontroléru na 16 MHz, operační napětí na 5V úrovni, ale přibyl Wi-Fi modul a interní měřicí jednotka [19], která nabízí tříosý akcelerometr a gyroskop. Nevýhodou oproti základnímu Arduino Uno je absence 1 PWM výstupu [19].

ESP32S

Vývojová deska s mikrokontrolérem Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6 [11] má oproti výše zmíněnému Arduino Uno nižší pracovní napětí 3,3 V a naopak vyšší taktovací frekvenci na 240 MHz.



Obrázek 3.4: Vývojová deska ESP32.⁴

⁴Obrázek byl převzat z: <https://arduino-shop.cz/arduino/1511-original-arduino-uno-rev3.html>

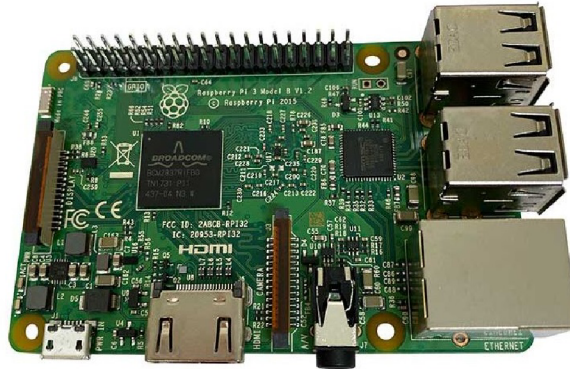
Součástí vývojové desky je i možnost využít až 28 digitálních vstupně-výstupních pinů, 8 analogových pinů a 16kanálový kanálový PWM výstup. Velkou výhodou je i ESP-WROOM-32 modul, který umožňuje komunikovat [11] přes Wi-Fi, Bluetooth i Ethernet. Další výhodou je možnost kompatibility s Arduino ide, kde je možné programovat tuto vývojovou desku prakticky stejně jako v Arduino ide.

Raspberry Pi

Jedná se o malý přenosný počítač od Raspberry PI Foundation. Cílem této společnosti je vzdělávání dětí a dospělých lidí v oblasti počítačů a tomu odvozených věd. Velikost tohoto počítače dosahuje rozměrů kreditní karty [5] a ve srovnání s jinými počítači nebo notebooky je levnější. Pracovat se s ním dá jako s běžným počítačem, takže není problém propojit ho s klávesnicí, myší, monitorem a dalšími zařízeními. Počítače Raspberry Pi nachází uplatnění [8] například jako video jukebox, retro herní automaty, magické zrcadlo, roboti a mnoho dalších.

- Nultá generace
 - Raspberry Pi Zero - Základní a nejmenší verze Raspberry Pi počítačů, s 1 GHz ARMovým, jedno jádrovým CPU, 512 MB RAM [10], 40 uživatelskými piny a bez modulu pro bezdrátovou komunikaci. Napájecí zdroj je udáván alespoň 160 mA, při napájení z microUSB. [25]
 - Raspberry Pi Zero W - Rozšířená verze Zero o moduly pro bezdrátovou komunikaci Bluetooth 4.1, Bluetooth s nízkou spotřebou (BLE) a LAN. Oproti základní verzi se zvýšila proudová spotřeba ze 160 mA na 180 mA. [25]
- První generace
 - Raspberry Pi 1 Model A+ - Základní model první generace přináší slabší 700 MHz ARMový CPU s 1 jádrem a poloviční RAM, která činí 256 MB, bez modulů pro bezdrátovou komunikaci. Oproti nulté generaci (Pi Zero/Pi Zero W) vzrostla proudová spotřeba ze 180 mA na 200 mA.[25]
 - Raspberry Pi 1 Model B - Oproti A+ verzi byla zdvojnásobena RAM na 512 MB, přibyla komunikace přes Ethernet a proudový odběr vzrostl na 700 mA.[25]
- Druhá generace
 - Raspberry Pi 2 Model B - Druhá generace počítače Raspberry Pi, s 900 MHz a čtyř jádrovým ARMovým CPU CORTEX-A7, 1 GB RAM, 40 uživatelských pinů GPIO, 100 Base Ethernetem.
- Třetí generace
 - Raspberry Pi 3 Model B - Nejnížší model třetí generace nabízející 1,2 GHz čtyř jádrový, Broadcom BCM2837 ARMový procesor [7], 1 GB RAM, 40 uživatelských GPIO pinů, obsahuje modul pro bezdrátové komunikace Bluetooth. Odebíraný proud se zvýšil na 2,1 A.
 - Raspberry Pi 3 Model B+ - Rozšířená B verze, kde byla navýšena taktovací frekvence procesoru na 1,4 GHz, nabízí 1 Gb komunikaci Ethernet přes 2.0 USB [25] a změnil se proudový odběr na 1,13 A.

- Raspberry Pi 3 Model A+ - Oproti verzi B+ postrádá komunikaci přes Ethernet a RAM je poloviční, tedy 512 MB.[25]
- Čtvrtá generace
 - Raspberry Pi 4 Model B - Nejvyšší generace, s 1,5 GHz čtyř jádrovým Broadcom BCM2711 CPU, RAM o velikosti 1 GB, 2 GB nebo 4 GB, obsahuje komunikaci přes Ethernet s rychlostí 1 Gb/s[7].



Obrázek 3.5: Přenosný počítač Raspberry Pi 3.⁵

Jednou z nevýhod Raspberry Pi je jednak spotřeba, ale také nedostatek PWM výstupů. Problém s nedostatkem výstupů je možné eliminovat použitím rozšiřujícího modulu s 16kanalovým kanálovým PWM a rozlišením 12 b [30], který je kompatibilní s verzemi Raspberry Pi Model Zero, A+, B+, Pi 2 nebo Pi 3.

3.3 Komunikace robota s počítačem

Ovládání samotných modelářských serv přes řídicí desku vyžaduje připojení daného servomotoru na pin, kde je aktivován PWM signál. Pokud se vyskytne požadavek ovládat robota přes uživatelské rozhraní, které je ovšem umístěno na počítači, bude poté zapotřebí zajistit komunikaci mezi počítačem a samotným robotem. Existují dvě základní metody, kterými toho lze docílit. Jedním z nich je drátové propojení řídicí desky s ramenem. Druhou možností je bezdrátové propojení. Bezdrátová varianta nabízí výhodu, kdy uživatel nemusí mít své rozhraní bezprostředně na počítači, ale může se k němu připojit prakticky z jakéhokoliv zařízení, například tabletu, mobilního telefonu a dalších, které daný typ komunikace podporují. V této práci jsou popsány pouze metody, které jsou bezdrátové.

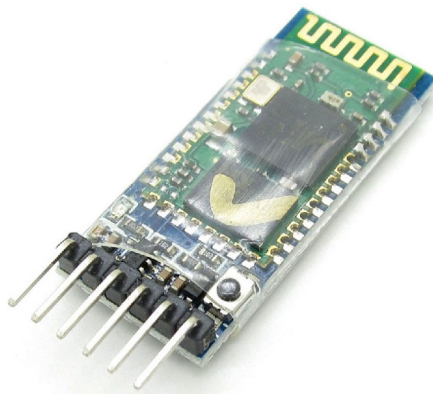
Komunikace přes Bluetooth

Prvním zástupcem z řady bezdrátové komunikace je technologie Bluetooth. Jedná se o standard, který komunikuje přes sériové rozhraní, například přes UART. Základem pro tento typ komunikace je mít alespoň dva moduly, kde jeden bude nastaven jako master,

⁵Obrázek byl převzat z webové stránky: <https://www.active-robots.com/raspberry-pi-3-model-b-sbc.html>

dále jen master, a druhý jako slave, dále jen slave. Master posílá data přímo slave zařízení nebo požádá slave zařízení, aby mu je zaslalo.

Všechna Bluetooth zařízení mají přidělenou adresu, pomocí níž může master komunikovat s konkrétním zařízením. Těto adrese se říká slave adresa a díky tomu nemusí zasílat data všem slave modulům (broadcast), ale pouze jednomu konkrétnímu [37]. Modul, který je pro tento typ komunikace možné použít, je například HC-05 RS232.



Obrázek 3.6: Modul HC-05 s vývody.⁶

HC-05 RS232 je bluetooth modul, který je postavený na základním modulu HC-05, pracující s nízkým operačním napětím od 1,8 V do 3,6 V. S rozšířením RS232 umožňuje pracovat s napětím 4 V až 6 V. Uživateli nabízí šest vývodů, jako je STATUS, RX, TX, GND, VCC a EN. Do modulu je možné přistoupit v takzvaném AT režimu [12], který podporuje až 33 AT příkazů, s nimiž je možné nastavit tento modul, například rychlost komunikace [2] (baud rate) na 460 800. Vstup do tohoto režimu je možný, přivedení logické úrovně jedna na EN pin.

Aby byla zajištěna základní funkcionality, je nezbytné nastavit komunikační rychlost (Baud rate) pomocí AT+UART, následně AT+ROLE, pro nastavené role master nebo slave. Nakonec se musí získat slave adresa zařízení, se kterým komunikujeme pomocí AT+ADDR? [2] a nastavit ji v master modulu pomocí AT+BIND.

Komunikace přes Wi-Fi

Druhou možností bezdrátové komunikace je technologie Wi-Fi. Modulů, které zmíněnou komunikaci umožňují, je vícero, ale rozhodl jsem se popsat ESP8266.

Jedná se o modul, pracující s nízkým napětím od 2,5 V do 3,6 V, podporuje síťové protokoly IPv4, TCP, UDP i HTTP. Podobně jako výše uvedený HC-05 modul umožňuje uživatelskou konfiguraci [36] pomocí speciální instrukční sady. Umožňuje pracovat ve třech různých režimech, jako je Wi-Fi stanice, Wi-Fi přístupový bod (access point) nebo obojí zároveň. Pro nastavení modulu je možné využít software Arduino IDE. Pokud uživatel zmíněný editor používá, stačí použít knihovnu ESP8266WiFi.h.

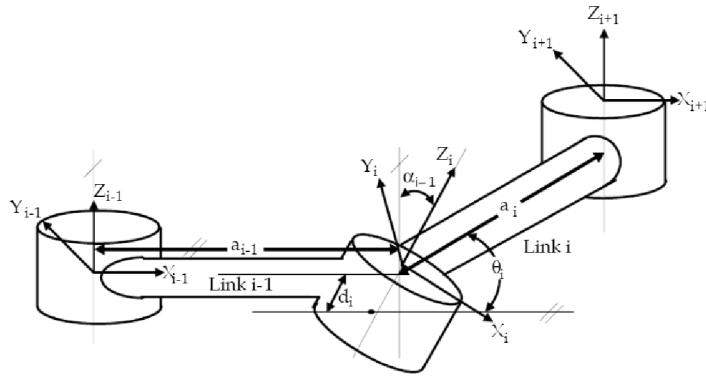
⁶Obrázek byl převzat z webové stránky: <https://www.lelong.com.my/hc-05-bluetooth-wireless-serial-port-tx-rx-hc05-arduino-xyntacpro-F1638059-2007-01-Sale-I.htm>

3.4 Geometrie manipulátoru

Inverzní a dopředná kinematika jsou v dnešní době velmi rozšířené pojmy v oblasti robotiky a robotických ramen. V první řadě je správné klasifikovat manipulátor, respektive určit, kolik bude mít spojů (stupňů volnosti), které spoje budou umožňovat rotace a které posuvy, určit délku jednotlivých spojů (rameno mezi dvěma spoji) a úhly, o které je možné s manipulátorem hýbat.

Dopředná kinematika

Dopředná kinematika využívá pro určení pozice koncového bodu jednotlivých proměnných kloubů. Pro popis dopředné kinematiky je například možné použít [39] metodu Denavit-Hartenberg, která se používá nejčastěji. Využívá celkem čtyři parametry: délka, krut a odsazení spojů a úhel kloubu. Každému kloubu se přidělí souřadnicový rám/kostra (anglicky coordinate frame) v kartézském prostoru k určení DH parametrů (Denavit-Hartenberg parametrů). Obrázek číslo 3.7 zobrazuje přiřazení jednotlivých souřadnicových rámců.



Obrázek 3.7: Manipulátor s přiřazenými souřadnicovými rámcí.⁷

Každý kloub manipulátoru je možné chápat jako [40] transformaci souřadnic z předchozího souřadnicového prostoru na následující. Na základě tohoto poznatku je možné vytvořit rovnici pro dopřednou kinematiku využívající DH parametry ve tvaru [40]:

$${}^n_{1}T = Tr_{Z_{n-1}}(d_n)Ro_{Z_{n-1}}(\theta_n)Tr_{X_n}(r_n)Ro_{Z_n}(\alpha_n)$$

Kde Ro označují rotaci, spodní index určuje, podle jaké osy, o úhel theta nebo alpha, podle toho, co je v závorce. Dále Tr značí posuv, spodní index určí podél jaké osy a hodnota v závorce určí vzdálenost posuvu. Výsledek pro jednotlivá T vypadá následovně:

$$\begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & r_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & r_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

V tento okamžik je definována transformační rovnice a transformační matice, které jdou použít pro jednotlivé klouby. Aby bylo možné určit výsledek pro konkrétní manipulátor,

⁷Obrázek byl převzat z: https://www.researchgate.net/publication/221785964_Robot_Kinematics_Forward_and_Inverse_Kinematics

je nezbytným krokem vytvořit tabulku DH parametrů pro jednotlivé klouby. Dosazením DH parametrů z tabulky do před chystané matice vyjde transformační matice, udávající transformaci z aktuálního kloubu (souřadnicového rámce) do následujícího kloubu.

Inverzní kinematika

Komplikovanějším typem kinematiky je takzvaná inverzní kinematika, která se využívá k ovládání manipulátorů. Pro řešení jsou známy pouze souřadnice počátečního bodu a bodu koncového [28], do kterého se snažíme manipulátor přesunout. Z pohledu manipulátoru známe pouze délky jednotlivých spojů. Na základě těchto údajů dochází k vypočítání úhlů jednotlivých kloubů. Pro pochopení pojmu inverzní kinematika je potřeba vědět [39], že samotná manipulace, čili to, co má manipulátor provádět v počítači, probíhá v rámci kartézského souřadnicového systému (zahrnujícího orientační matici a vektor pozice), zatímco práce samotného ovládání pracuje s prostorem kloubů (pomocí úhlů). Převod mezi těmito prostory se nazývá problém inverzní kinematiky.

Existuje velké množství přístupů, pomocí kterých je možné řešit problémy inverzní kinematiky [32]:

- **Analytické řešení:** Umožňuje nalezení ideálně všech možných řešení. Výsledná pozice kteréhokoliv q (proměnná kloubu) je získána na základě zkoumání projekce jednotlivých segmentů (spojů mezi klouby) do X a Y osy.
- **Numerické řešení:** Řešení je založeno na Jacobianové matici, dále pouze J , které mapuje změny proměnné kloubu q na změny koncového efektoru x . Jakobiho metoda má rozměr $m \times n$, kde m udává dimenzi x a za n se dosadí počet kloubů manipulátoru.

$$J = \frac{\delta f}{\delta q}$$

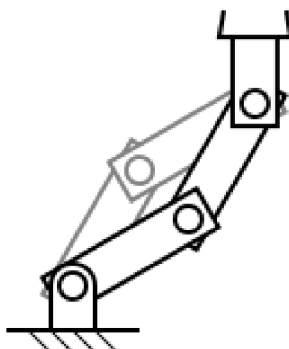
- **Pseudoinverzní řešení** Nejpoužívanější metoda pro řešení inverzní kinematiky, která využívá výše uvedenou Jacobianovou matici, ale v inverzní podobě. Výhodou je nalezení unikátního řešení v případě redundantního systému.

$$\dot{q} = J^+ \dot{x}$$

Za použití iteračního schématu je možné v každé iteraci vypočítat x z aktuálního a žádoucího x a q pomocí pseudoinverzního Jacobiana a následné integrace.

- **Transponovaná matice** Oproti pseudoinverznímu řešení je opakovatelná, což je, v mnoha aplikacích, žádoucí. Pomocí transponované Jacobianové matice je u problémů inverzní kinematiky možné vyřešit uzavřené smyčky. Tato metoda může ale vést k problémům použitelnosti.
- **Lagrangianova metoda:** Napomáhá rozšíření nedostatečně omezeného systému přes Lagrangeové multiplikátory.
- a další ...

Inverzní kinematika je oproti dopředné kinematice složitější [39] kvůli silné závislosti na struktuře manipulátoru. Kromě vyšší složitosti přináší také problém nejednoznačnosti (redundance), hlavně u složitějších modelů.



Obrázek 3.8: Problém více možných řešení inverzní kinematiky.⁸

Nejednoznačností se myslí existence více správných řešení, které vedou ke stejnému koncovému bodu, avšak napozicování jednotlivých spojů je u každého z nich jiné, jak zobrazuje obrázek číslo 3.8.

3.5 Program pro modelování 3D modelů

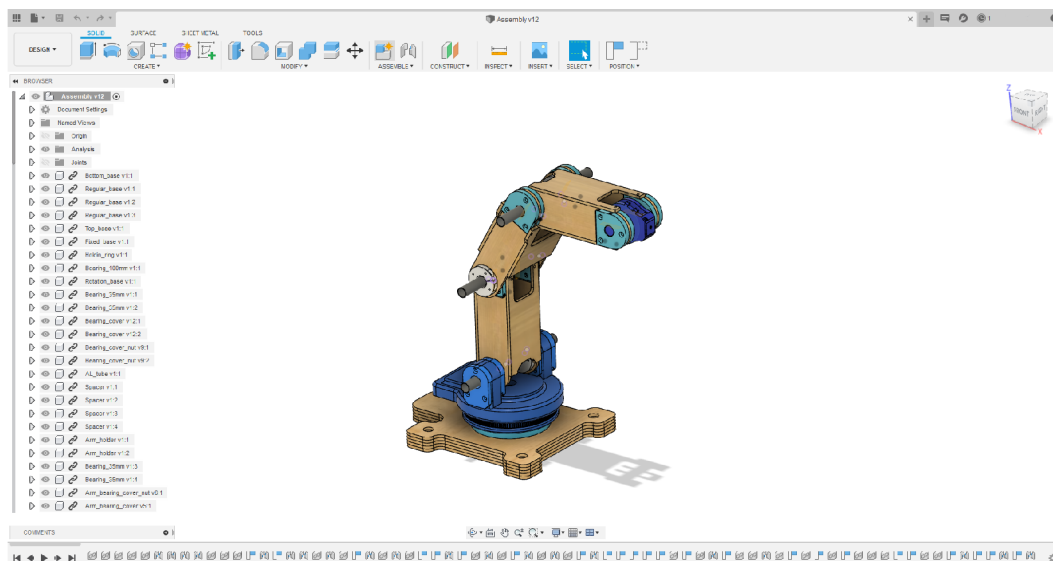
Programů pro vytváření 3D modelů je vícero, ale pro účel této práce jsem zvolil Fusion 360 a Solidworks, které jsou popsány níže.

Fusion 360

Jedním z programů, který je možný pro návrh 3D modelů použít, je Fusion 360. Jedná se o produkt firmy Autodesk Inc.[15], který je možné využít pro nekomerční účely zcela bezplatně. Program nabízí kromě tvorby 3D designu/modelů, také [15] simulace, tvorbu dokumentací, generativní design, práce s elektronikou [16], ve které je kromě designu plošných spojů a schématů, možnost vytvořit 3D model navržené desky. a další. Fusion 360 také umožňuje přidávání poznámek k jednotlivým modelům a kolaborující členové okamžitě vidí, co je například potřeba zkontrolovat.

Na obrázku 3.9 je vidět samotná scéna, ve které jsou modely tvořeny. Fusion 360, vkládá jednotlivé prvky do scény do vertikálního sloupce a časovou osu změn ve spodním řádku, jak znázorňuje obrázek.

⁸Obrázek byl převzat z: <https://commons.wikimedia.org/wiki/File:Inverse-kinematics-multiple-solutions.svg>



Obrázek 3.9: Program pro modelování 3D modelů - Fusion 360.

V rámci Fusion 360 je také podporována práce více uživatelů [4] na projektech, pomocí cloudu. Uživatel si může také stáhnout mobilní aplikaci, ve které může updatovat projekt přímo ze svého telefonu. Při práci v týmu je možné využít verzovací systém [34], který Fusion 360 nabízí. Jakmile uživatel uloží svůj postup, tak se automaticky zaregistruje daná změna a je uložena do verzovacího systému. U každého modelu je umístěna ikona s číslem dané verze. Při problému s modelem je následně možné vrátit se na některou z předešlých verzí.

Minimální požadavky na systém Windows: [26]

- Operační systém: (64b) Windows 8.1 a vyšší
- Procesor: 1.7GHz s alespoň 4 jádry
- Grafická karta: 1GB nebo 6GB integrované, DirectX11 a větší
- Operační paměť RAM: 4GB nebo 6GB integrované
- Uložiště: alespoň 3GB

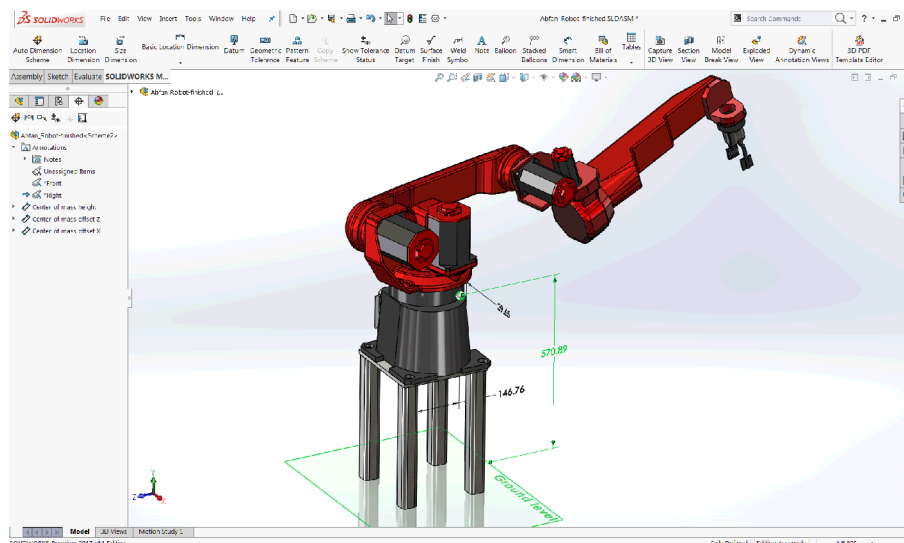
Výše uvedené minimální systémové požadavky jsou uvedeny pro aktuální verzi programu Fusion 360.

SolidWorks

Dalším programem, ve kterém je možné vytvářet 3D modely, je program SolidWorks. Jedná se o program od firmy Dassault Systèmes SolidWorks [3]. Donedávna nejúspěšnější strojírenský 3D CAD systém na Českém trhu. Uživateli umožňuje práci s objemovým a plošným modelováním, práci s plechovými prvky, sestavami a s dalšími prvky. Na základě sestav a jednotlivých modelů je také možné generování výrobních výkresů.

Ovládání tohoto programu je poměrně vizuálně přehledné a snadné [3]. Samotné ovládání je založeno na technologii SWIFT, díky které se redukuje potřeba opakujících se úkonů.

Obrázek 3.10 níže, znázorňuje scénu, ve které samotné modelování 3D modelů i sestav vzniká.



Obrázek 3.10: Program pro modelování 3D modelů - Solidworks.⁹

SolidWorks je možné získat v několika různých balíčcích, které přidávají uživateli například možnost realizovat přesné strukturální a pohybové simulace [3], návrh kabelových svazků, elektrické rozvody, práce s dokumentacemi a další vlastnosti.

Minimální požadavky na systém Windows[24]:

- Operační systém: (64b) Windows 7 nebo Windows 10
- Procesor: alespoň 2 jádrový
- Grafická karta: minimálně 2GB
- Operační paměť RAM: 8GB
- Uložiště: alespoň 2GB

Výše uvedené minimální systémové požadavky jsou uvedeny pro nejnovější verzi programu SolidWorks 2020.

⁹Obrázek byl převzat z: <https://www.engineersrule.com/whats-new-solidworks-2017-mbd/>

Kapitola 4

Analýza stavu a návrh řešení

Na základě předchozích kapitol, kde jsem analyzoval již existující řešení robotických ramen a komponenty, které jsou pro tvorbu robotického ramene možné použít, je důležité si vytyčit základní koncepci mého vlastního manipulátora. V této kapitole jsou popsány vlastnosti, které bude moje robotické rameno splňovat, a také jednotlivé bloky, které musím fyzicky realizovat za účelem vytvoření této práce.

4.1 Porovnání existujících ramen

Ve 2. kapitole jsou popsána již existující řešení robotických ramen. Pro přehlednost, jsem sumarizoval jednotlivé vlastnosti do tabulky níže.

Feature/manipulátor	PiArm	Dexter ER2	Niryo One
Open source	Ne	Ne	Ano
Provedení stavebnice	Ano	Ne	Ano
Bezdrátové ovládání	Ano	Ne	Ano
Modularita a rozšiřitelnost	Ano	Ne	Ano
Více možných klepet	Ne	Ne	Ano
Možnost programovat	Ano	Ne	Ano
Znalost programování	Ne	Ne	Ne
Senzory pro interakci	Ano	Ne	Ne
Počet stupňů volnosti	5	5	6
Komunikace s ramenem	Bluetooth	USB	Wi-Fi
Vlastní GUI	Ano	Ano	Ano
Materiál konstrukce	Hliník	Slitiny hliníku	plast 3D tisk
Přibližná základní cena	£199	295\$	1999\$

Tabulka 4.1: Přehled vlastností existujících ramen

Na základě tabulky výše jsem se rozhodl porovnat jednotlivá ramena z 2. kapitoly, i s výběrem nejlepšího z nich (dle mého názoru) a odůvodnění, proč je zrovna toto rameno dle mého názoru nejlepší.

Vzájemné porovnání ramen

Na základě hodnot, uvedených v tabulce výše, je dle mého názoru nejpříjemnější robotické rameno PiArm. Důvodem je poměr mezi pořizovací cenou a možnostmi, které nabízí. Uživatel má jednoduché a přehledné uživatelské rozhraní, ve kterém ovšem může ovládat pouze jednotlivá serva zvlášť. Dále umožňuje připojit externí senzory, jako jsou IR senzor a ultrazvukový senzor, díky kterým může interagovat s okolím. Pokud uživatel bude chtít, může robota ovládat až třemi různými způsoby, včetně zmíněného GUI.

Dexter ER2 je cenově dražší, než robotické rameno PiArm, ale uživateli neposkytuje prakticky žádný prostor pro kreativitu. Prodává se jako hotový model, neposkytuje možnost rozšíření o senzory nebo jiné uchopovací mechanismy. Komunikace mezi ramenem a GUI je realizována přes USB, nabízí uživatelské rozhraní pro ovládání až 32 serv v reálném čase, pomocí posuvníků.

Robotické rameno Niryo One uživateli poskytuje dle mého názoru diametrálně více prostoru pro kreativitu než Dexter ER2. Uživateli nabízí až 3 různé režimy, jak rameno ovládat, možnost programování, je možné měnit jiné uchopovací mechanismy, jako jsou klasická klepeta, vakuová pumpa nebo elektromagnet. Komunikovat umožňuje jak přes Wi-Fi, tak přes ethernet a nabízí také možnost připojit Arduino nebo Raspberry Pi řídicí desku, pro rozšíření funkcionality. Velkou nevýhodou tohoto ramene je jeho pořizovací cena, která je přibližně 10x vyšší než v případě robota PiArm.

4.2 Vymezení konceptu práce a stanovení cílů

Na základě porovnání jednotlivých robotických ramen z předchozí kapitoly 4.1., jsem si vytyčil základní vlastnosti, které by měl můj manipulátor splňovat. Těmito vlastnostmi jsou:

- **Poskytnutí projektu uživateli zdarma** - Aby si mohl uživatel můj projekt stáhnout, vytisknout a případně modifikovat, je mým cílem poskytnout mu potřebné dokumenty, spojené s touto prací.
- **Jednoduchá replikovatelnost** - Cílem je, aby si prakticky každý uživatel, alespoň s minimálním technickým vybavením, mohl sám doma replikovat tohoto manipulátora sám. Aby to bylo možné, rozhodl jsem se, že veškeré konstrukční prvky mimo spojovací prvky, jako je šroub, budou navrženy tak, aby se daly vytisknout na 3D tiskárně, například z materiálu PLA. Důvodem je, že přístup široké veřejnosti k 3D tiskárnám je v dnešní době poměrně snadný a do jisté míry i levný.
- **Modulární konstrukce (formou stavebnice)** - Důvodů, proč jsem se rozhodl k tomuto bodu, je hned několik. Prvním z nich je možnost dotisknout vadný díl. Pokud by nebyla modularita, tak by si uživatel musel vytisknout celý manipulátor znovu, což je jak časově, tak i finančně náročné. Druhým důvodem jsou zkušenosti uživatele. Jestliže se k mému projektu dostane člověk, který s robotickými rameny nikdy nepracoval, bude moci vytisknout to, co jsem navrhl. V opačném případě si může libovolně prvky poupravit tak, aby splňovaly jeho požadavky.
- **Pět stupňů volnosti** - Jelikož chci realizovat robotické rameno, které by mělo alespoň z části vypadat jako lidské rameno, a zároveň aby těchto stupňů nebylo příliš mnoho, tak jsem se rozhodl pro 5 stupňů volnosti.

- **Interakce robota s okolními objekty** - Aby bylo rameno užitečné, je důležité přidat koncový prvek, který bude moci uchopit nějaký objekt, přesunout ho, zvednout a tak dále. Koncovým tedy mělo být klepeto.
- **Bezdrátové ovládání pomocí Wi-Fi** - Cílem je, aby ovládání ramene nebylo vázáno na konkrétní zařízení, kde je umístěno uživatelské rozhraní. V takovém případě se vyskytuje možnost nahrát uživatelské rozhraní na webový server, na který by se mohl uživatel připojit prakticky odkudkoliv.
- **Poskytnutí vlastního uživatelského rozhraní** - Vytvoření vlastního uživatelského rozhraní pomůže uživateli s tím, že si může jednoduše se základní variantou manipulátoru hrát a nemusí nic zdlouhavě instalovat atd. Pokud se ovšem rozhodne změnit počet serv či radikálně změnit funkčnost celého manipulátoru, bude si muset samotné uživatelské rozhraní programově poupravit sám.

Požadované vlastnosti

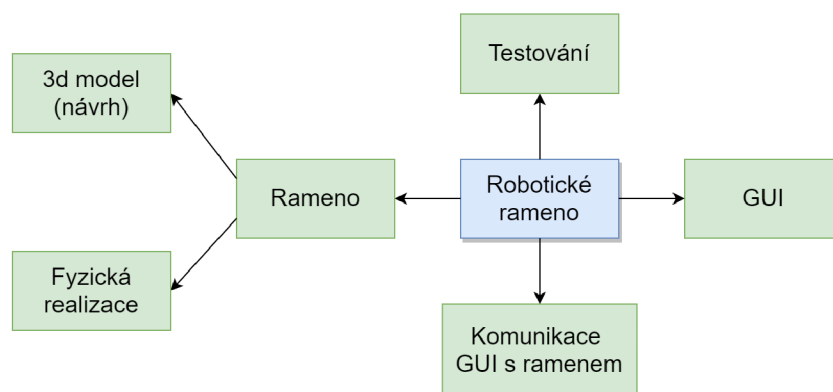
Na základě výše uvedených cílů jsem se rozhodl vytyčit základní vlastnosti, které by měl fyzický model splňovat.

- Bezproblémový pohyb v jednotlivých osách (plynulost, stabilita).
- Jednoduchý design a nenáročné sestavení.
- Schopnost uzvednout alespoň 100g závaží.
- Váhově lehký model, aby se rameno dalo jednoduše přenášet.
- Jednoduše odjímatelné klepeto.
- Poskytnout rozmanité ovládání.

Ověření, zda moje rameno tyto vlastnosti splňuje, bude zapotřebí následně ověřit sadou testů, které jsou popsány v kapitole “Testy potřebné k ověření funkčnosti ramene”.

4.3 Seznam částí k implementaci

Při stanovování koncepce robotického ramene jsem se rozhodl, že bude vhodné, aby struktura díla byla tato:



Obrázek 4.1: Blokové schéma značící strukturu díla.

- **3D model** - Navržení jednotlivých částí ramene tak, aby splňovaly modularitu a zároveň je bylo možné vytisknout bez obtíží na 3D tiskárně.
- **GUI** - Vytvoření intuitivního a jednoduchého uživatelského rozhraní, ve kterém bude uživatel moci rameno ovládat. Tato část zahrnuje nabídnout možnost manipulovat s koncovým zařízením (klepetem), samotnými klouby a případně možnost přepnout se na alternativní způsob ovládání pomocí grafického modelu, který nastaví pozice serv na základě manipulace s tímto modelem.
- **Komunikace** - Tento krok zahrnuje zprovoznění webového serveru, na který se umístí uživatelské rozhraní. Dále je nutné zajistit komunikaci mezi tímto serverem a samotným robotem, aby se změny v uživatelském rozhraní promítly i u reálného modelu. V případě, že bude možné ovládat robotické rameno přes více jak jeden režim, je poté zapotřebí propojit tyto režimy tak, aby oba ukazovaly stejný aktuální stav (změna pozice kloubu bude stejná v obou režimech).

Výše popsané bloky, nezbytné k implementaci, jsou podrobně popsány v 5. kapitole.

4.4 Testy potřebné k ověření funkčnosti ramene

Aby bylo možné ověřit, zda se manipulátor chová správně, a je možné prohlásit výsledek za správný, je nezbytné podrobit ho sadě testů před vypuštěním do světa. Testy, které je potřeba otestovat jsem rozdělil do několika následujících tříd:

1. **Testy kalibrace** - Tato sada by měla obsahovat testy, při nichž se robotické rameno bude uvádět do výchozího stavu při prvním spuštění. Kontrolovat by se měly různé výchozí polohy ramene.
2. **Testy funkčnosti** - Součástí testování funkčnosti je schopnost manipulátoru reagovat na podněty z uživatelského rozhraní, plynulost pohybu a další testy, které se snaží odhalit problémy při normální činnosti ramene.
3. **Testy uživatelského rozhraní** - Má za úkol ověřit samotné uživatelské rozhraní, například zobrazování jednotlivých komponent, jejich funkčnost, kontrolu, že při přepínání režimů v GUI nedojde k poškození zobrazení, test provázanosti jednotlivých komponent a další. Tato sada je úzce spojena s testy funkčnosti.
4. **Testy reakce na zátěž** - Cílem je ověřit reakce robota při manipulaci s předměty, při kterých se bude testovat, jakou váhu je schopen unést, jestli je schopen tuto váhu udržet dlouhodobě, zda je možné s ním plynule pohybovat i když drží nějaký předmět a tak dále. Tato sada je spojena s testy robustnosti.
5. **Testy robustnosti mého řešení** - Pomůže otestovat, zda je můj návrh odolný, pevný, stabilní při manipulaci a další vlastnosti. Tato sada je úzce spojena s konstrukcí ramene.
6. **Testy dlouhodobého užití** - Sada, která zjišťuje, zda při dlouhodobém užití, například několika dní, se robot chová dle specifikace.

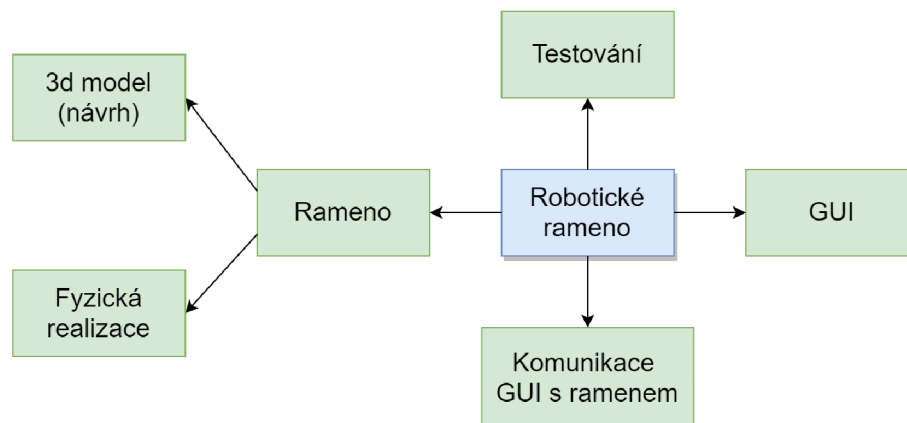
V důsledku časového omezení této práce, nebude možné realizovat všechny třídy testů, obzvlášť ne testy dlouhodobého užití, na které by byl vhodný alespoň 1 měsíc a vícero uživatelů. Tyto testy vřele doporučuji provést všem uživatelům, kteří si mé robotické rameno rozhodnou postavit.

Kapitola 5

Fyzická realizace robotického ramene

Tato kapitola popisuje realizaci jednotlivých bloků, které jsem si stanovil v předešlé kapitole 4.3. Popisuji zde vlastní realizaci uživatelského rozhraní a robotického ramene, ať už po softwarové či hardwarové stránce. V rámci této kapitoly poskytuji také část, která se zabývá testováním robotického ramene, viz kapitola 5.5, na základě mnou stanovených testovacích případů z kapitoly 4.4.

Na obrázku níže poskytuji přehled jednotlivých bloků, které jsou k dokončení této práce nezbytné. Tyto bloky popisuji a implementuji v této kapitole.



Obrázek 5.1: Blokové schéma s částmi, které musím realizovat.

- **Rameno:**

- **3D model:** Robotické rameno je nejprve nutné navrhnut v podobě 3D modelu, abych měl při realizaci z čeho vycházet. Dalším důvodem je možnost vytisknout si tyto modely na 3D tiskárně.
- **Fyzická realizace:** Samotná realizace robotického ramene, která zahrnuje výběr vhodných komponent, vtištění dílů na 3D tiskárně a zkompletování.

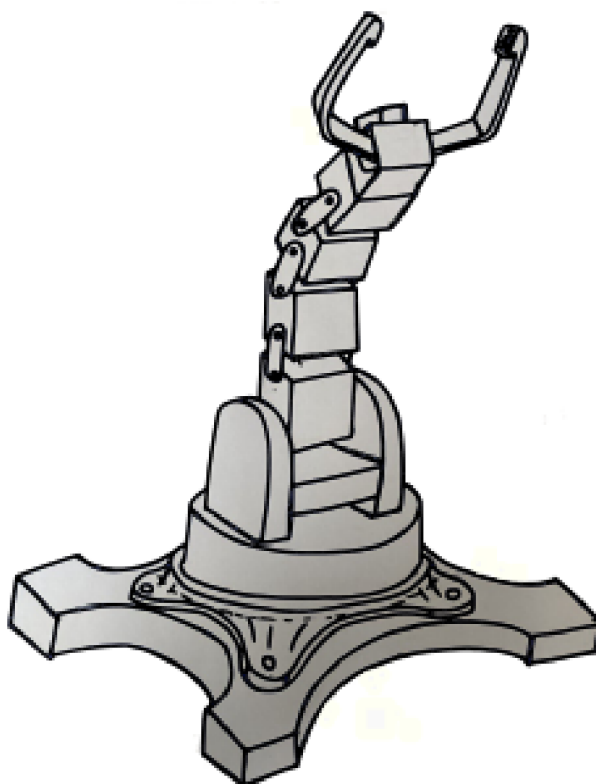
- **GUI:** Vytvoření jednoduchého a intuitivního uživatelského rozhraní, pomocí kterého bude možné robotické rameno ovládat a interagovat s okolím, například pomocí ucho-povacího mechanismu.

- **Komunikace GUI s ramenem:** V rámci tohoto bloku je zapotřebí zprovoznit server, na který bude umístěno samotné GUI. Další důležitou částí je i program, který bude umístěn na řídicí desku robota a bude zajišťovat pohyb jednotlivých serv a funkčnost dalších komponent, které bude možné ovládat přes GUI.
- **Testování:** Na závěr je zapotřebí robotickou ruku otestovat a získat její dosažené vlastnosti. Otestováním se zjistí, zda manipulátor splňuje požadavky, které jsem si definoval v kapitole 4.2.

Pořadí výše uvedených bloků odpovídají pořadí, v jakém jsou popsány kapitoly, které se těmito bloky zabývají.

5.1 Fyzická realizace robotického ramene

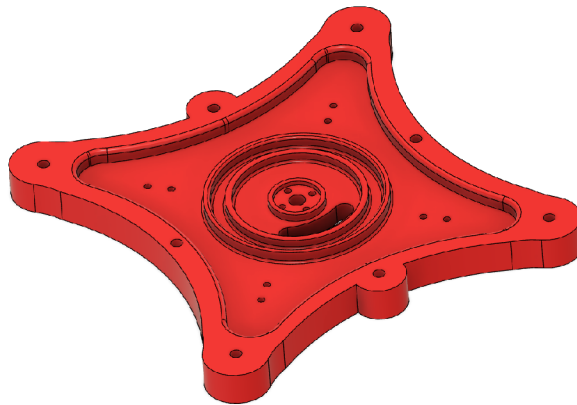
Aby bylo vůbec možné robotické rameno postavit, je poměrně logické začít u samotné konstrukce, nejlépe v podobě 3D modelu. Takový model totiž poslouží jako podklad pro výrobu na 3D tiskárně. Jelikož chci poskytnout veškeré podklady uživateli, rozhodl jsem se koncipovat celý model jako modulární, tedy aby si uživatel mohl libovolnou část dle svých potřeb upravit. Modularita s sebou přináší také výhodu, kdy při změně určitého dílu není nutné, alespoň ne vždy, aby si uživatel musel vytisknout celého robota, ale pouze upravenou část.



Obrázek 5.2: Prvotní návrh robotického ramene.

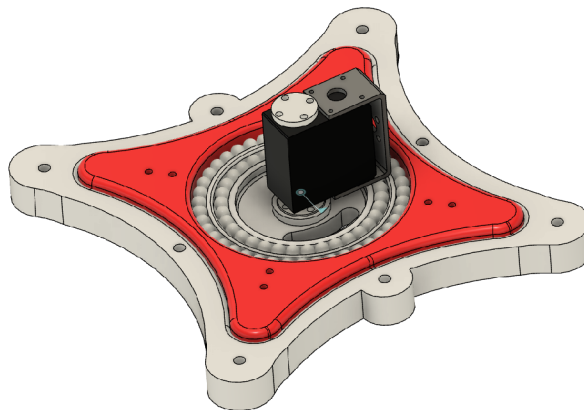
Obrázek 5.2 zobrazuje koncept robotického ramene. Celé rameno jsem rozdělil do pěti různých částí neboli bloků. Těmito bloky jsou:

- **Statická část (báze):** Slouží k uchycení celého manipulátoru k nehybnému objektu, například k pracovnímu stolu. Na tento prvek se poté uchytí další navazující moduly.



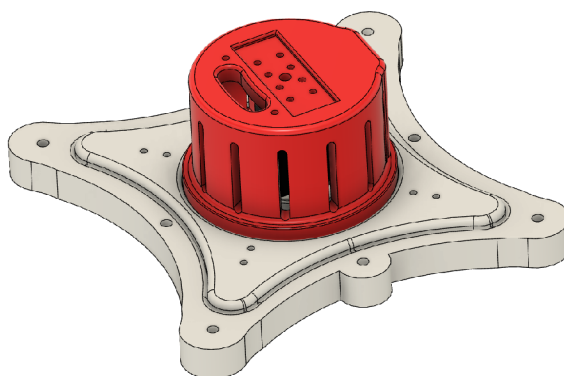
Obrázek 5.3: Statická báze robotického ramene.

- **Vymezovač otočné báze:** Hlavním úkolem je vytvořit pevný přechod mezi otočnou bází a statickou bází. Pro snížení tření těchto částí je možné použít kuličky o průměru 6 mm, ať už kovové, plastové, keramické a další, které do určité míry zastupují kuličková ložiska, viz obrázek 5.4.



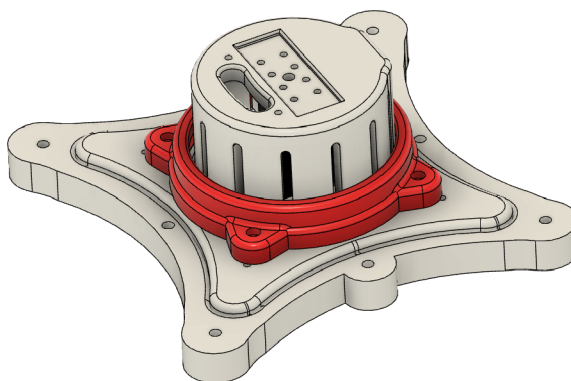
Obrázek 5.4: Vymezovač otočné báze přichycený ke statické bázi.

- **Otočná báze:** Samotná pohyblivá báze, která umožní robotickému rameni otáčet se okolo své osy. Jak zobrazuje obrázek číslo 5.4, ke statické bázi se připevní modelářské servo, které je druhým koncem připevněno k bázi otočné. Ovšem tato realizace s sebou nese úskalí, kdy při pohybu ramene dojde k vyklánění celé báze a může dojít až k vylomení samotného serva. Tento problém řeší další vymodelovaná část “Uchycovací prstenec”. Otvor, který se v otočné bázi nachází, slouží na protáhnutí kabelů od jednotlivých modelářských serv.



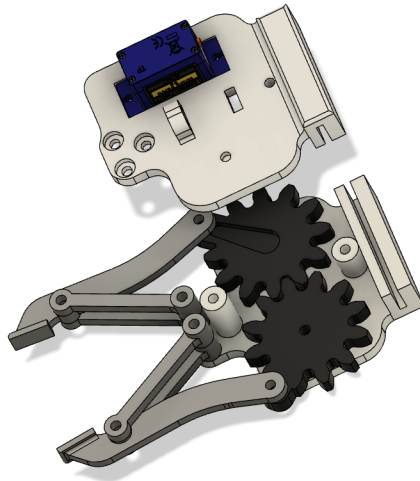
Obrázek 5.5: Otočná báze robotického ramene.

- **Uchycovací prstenec:** Uchytí otočnou bázi k vymezovači otočné báze tak, aby se mohla otáčet, a zároveň zabráňuje vyklánění celé otočné báze, čímž zajistí, aby se společně se servem samovolně nevytrhly. Aby byl zajištěn plynulý pohyb otočné báze a zároveň bázi držel, je v prstenci vymodelovaný schodek, přidržuje bázi, ale zároveň přidržení není silové, tudíž je pohyb plynulý. Pro snížení tření mezi otočnou bází a uchycovacím prstencem je možné vytisknout upravený prstenec, který bude mít schod upravený tak, aby se pod něj dal umístit například prstenec z plexiskla.

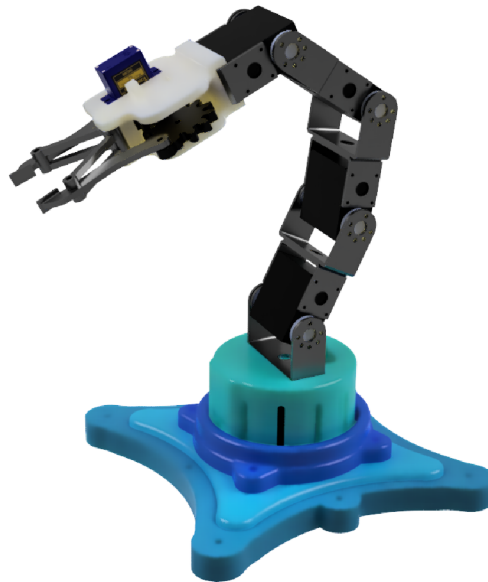


Obrázek 5.6: Uchycovací prstýnek držící bázi.

- **Uchopovací mechanismus (klepeto):** Slouží k interakci s okolím. Umožňuje manipulovat s předměty a díky přidané laserové diodě slouží jako laserové ukazovátko. Klepeto jsem se rozhodl navrhnout tak, aby v zadní části, díky svému žlabu, šlo pouze nasunout na hliníkový rám serva a dotáhnout šrouby.



Obrázek 5.7: Rozložené klepeto k manipulaci s předměty.¹



Obrázek 5.8: Výsledný design robotického manipulátoru.

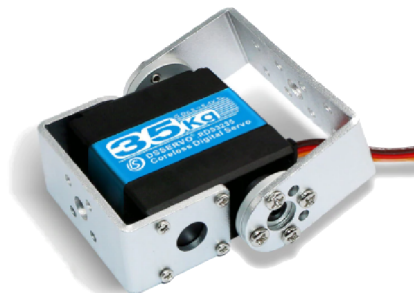
Obrázek 5.8 znázorňuje výsledný model mého robotického ramene, který jsem zhotovil v programu Fusion 360. Výsledný design se od původního návrhu liší pouze v přidání vymezovače otočné báze a změnou designu otočné báze jako takové.

Zvolená modelářská serva

Neodmyslitelnou součástí robotického manipulátoru jsou bezpochyby serva. Můj manipulátor využívá 2 různé typy serv podle toho, v jaké části jsou využívány, a tedy jaké jsou na ně kladeny nároky:

¹Model malého 9g serva byl převzat od autora Matheus Frasson z: <https://grabcad.com/library/sg90-micro-servo-9g-tower-pro-1>

- **Otočná báze a klouby:** Jelikož otočná báze musí otáčet celým robotem a jednotlivé klouby drží poměrnou váhu jak robota, tak samotného objektu, se kterým právě manipuluje, je velmi důležité využít silnější typ. Tím je digitální 35 kg servo RDS3135 s točivým momentem 35 kg.cm, při napájecím napětí 7,4 V. Při uvedeném napětí 7.4 V se jeho klidový proud pohybuje okolo 3,2 A. Jeho mechanický úhel otočení je 270 stupňů.



Obrázek 5.9: Servo, které je využito pro klouby a otočnou bázi.²

- **Klepeto:** Samotný uchopovací mechanismus je umístěn na úplném konci robotického ramene, a tím pádem není nutné, aby byl stejně silný jako serva pro klouby. Spíše by to bylo ke škodě, protože je poměrně těžké a na ostatní serva by tak byla vyvíjena větší zátěž. Druhou nevýhodou je jeho velký rozměr na tak malý prvek, jako je klepeto. Na základě těchto dvou problémů jsem zvolil malé 9g analogové servo [6], které při napětí 6 V poskytne točivý moment až 1,6 kg.cm.



Obrázek 5.10: Servo, které je využito pro uchopovací mechanismus.³

Servo na klepeto je sice výrazně slabší, než to na klouby, ale také nezanedbatelně lehčí a menší. Na otevírání a zavírání mechanického úchopu a manipulování s lehčím objektem je ovšem dostačující.

Řídící deska a napájení

Posledním krokem je výběr správné řídicí desky, která zajistí ovládání ramene, a s tím související napájecí zdroj. Řídící desku jsem zvolil Raspberry Pi 3B+, která mi poskytuje velký výkon a také nabízí spoustu možností pro ovládání svých pinů, serv a dalších prvků.

²Obrázek i popis serva byl převzat z:

<https://www.aliexpress.com/item/32879554214.html?spm=a2g0s.9042311.0.0.35ed4c4dHILfsH>

³Obrázek serva byl převzat z: <https://arduino-shop.cz/arduino/897-eses-servo-motor-9g.html>

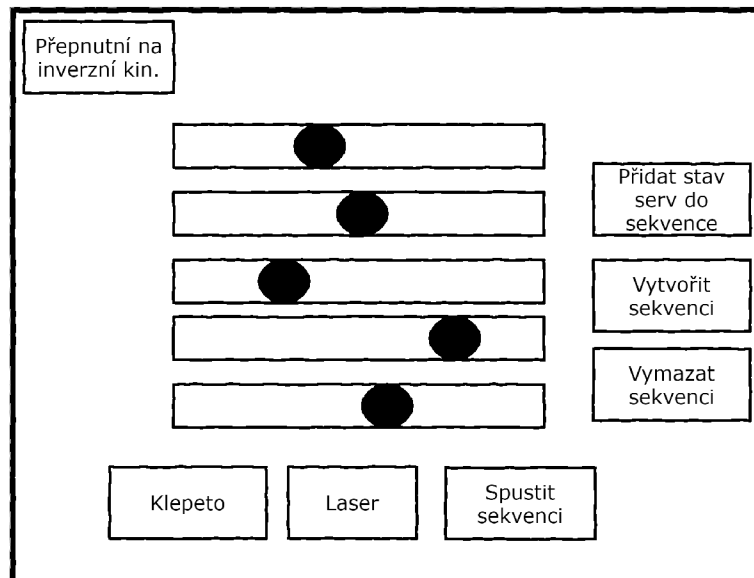
Dalším důvodem, který mě vedl k výběru této řídicí desky, byl můj osobní zájem ji vyzkoušet. Nevýhodou této desky je ovšem její požadovaný proud, který se může pohybovat okolo 1,5 A při 5 V.

Je tedy zapotřebí určit dostatečně silný napájecí zdroj, který poskytne dostatečný výkon k napájení Raspberry Pi 3B+ a všech 5 modelářských serv. Napájecí zdroj, který splnil mé požadavky, je AC/DC CONVERTER 7.5 V 150 W [9] od značky MEAN WELL. Tento zdroj poskytuje 2 sdílené výstupy se 7,5 V, které odpovídají napětí, které vyžadují kloubová serva a servo pro otočnou bázi. Pro Raspberry Pi a servo určené pro klepeta je nezbytné toto napětí pomocí napěťového převodníku snížit na 5 V. Pokud tedy budeme po tomto zdroji vyžadovat napětí 7,5 V, tak do jmenovitého výkonu 150 W je možné čerpat maximálně 20 A. Z těchto 20 A jsem vyhradil 2,5 A pro Raspberry (s rezervou), 5*2,5 A pro jednotlivá serva, což znamená, že i s velkou rezervou zdroj naprosto dostačuje.

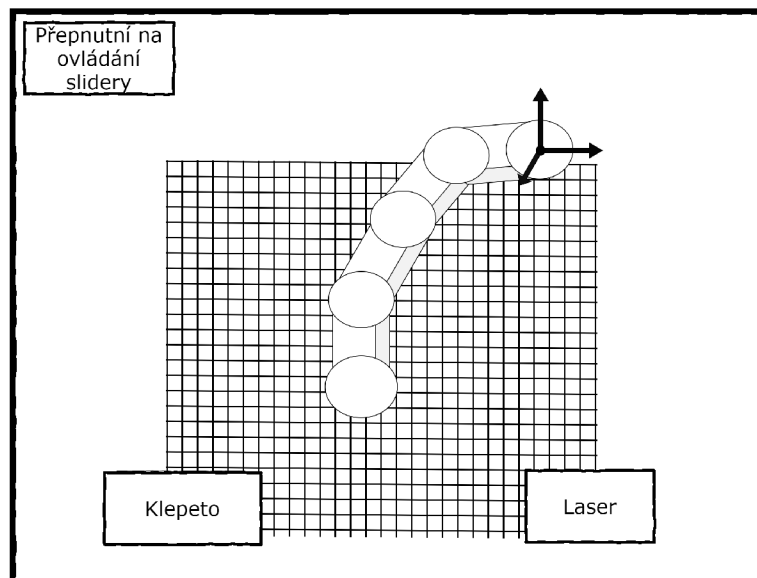
5.2 Uživatelské rozhraní pro ovládání manipulátoru

Mezi cíli, které jsem si stanovil pro samotné GUI, byly bezpochyby bezpochyby intuitivnost a jednoduchost. Uživatelské rozhraní jsem rozdělil do dvou různých režimů. Prvním je ovládání jednotlivých serv pomocí slide barů (horizontální posouvátko). Druhým režimem je poté manipulování s 3D modelem, tedy pohyb využívající inverzní kinematiku. Obě tyto části jsou vzájemně propojené a změna polohy libovolného kloubu, která se provede v jednom z režimů, se po přepnutí do druhého režimu sama aktualizuje. Stejně je tomu i u klepet a laseru. V případě výpadku elektrické energie jsou naposledy nastavené hodnoty opět načteny při spuštění a uživatel se tak nemusí obávat, že o hodnoty přijde.

Cílem bylo vytvořit uživatelské rozhraní, pomocí něhož by uživateli mohl jednak ovládat jednotlivá serva, a také ovládání je pomocí inverzní kinematiky. Prvotní koncept tohoto rozhraní zobrazují obrázky 5.11 a 5.12, kde 5.11 znázorňuje část pro ovládání jednotlivých serv a 5.12 ovládání pomocí inverzní kinematiky.

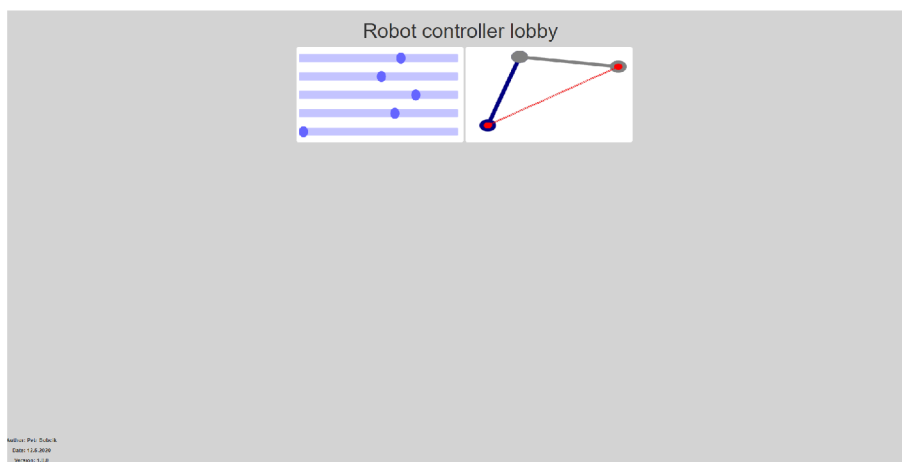


Obrázek 5.11: Koncept grafického rozhraní pro ovládání jednotlivých serv.



Obrázek 5.12: Koncept grafického rozhraní pro inverzní kinematiku.

Obě tyto varianty mají stejné funkce, jako je otevírání a zavírání klepet a zapnutí či vypnutí laserového ukazovátka pomocí tlačítek. Intuitivnost i jednoduchost u obou režimů zajišťuje minimální množství textu, které jsou většinou nahrazeny obrázky, stejné rozmístění interaktivních prvků, jako jsou tlačítka, a minimum prvků, se kterými lze interagovat. Aby se uživatel mohl rozhodnout, do jakého rozhraní se podívá dříve, vytvořil jsem jednoduchý rozcestník, pomocí kterého si může při prvním spuštění vybrat.



Obrázek 5.13: Rozcestník při prvním spuštění GUI.

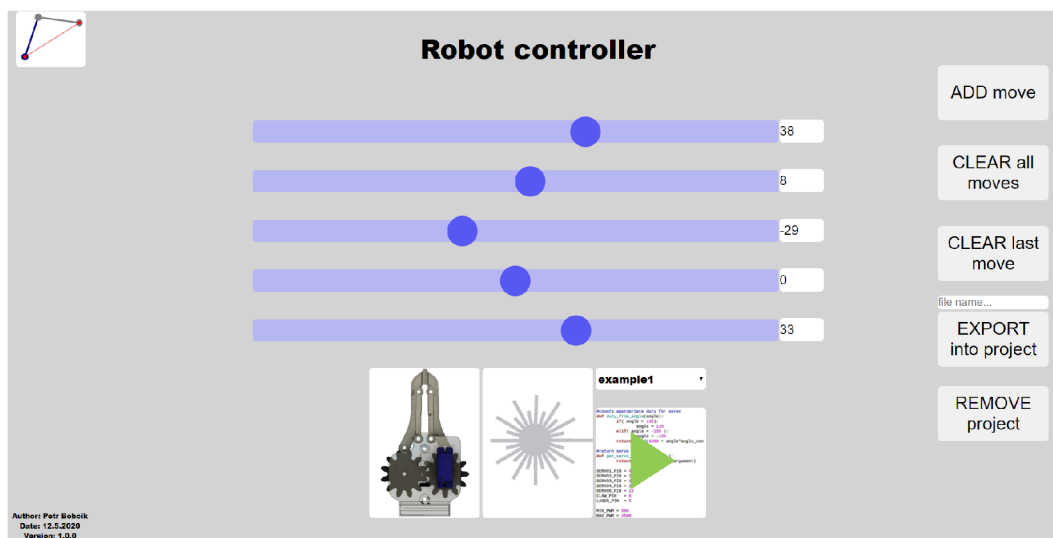
Obrázek 5.13 znázorňuje rozcestník uživatelského rozhraní, které se uživateli ukáže jako první. Jakmile si uživatel zvolí jeden z režimů, tak už není možné navrátit se na tento rozcestník jinak, než ručně zadat do vyhledávače adresu k němu.

Režim ovládání jednotlivých serv

Tento režim obsahuje pět horizontálních posouvačů, umístěných ve středu obrazovky. Každému z těchto posouvačů je přiřazeno právě jedno modelářské servo, a to tak, že nejvyšší posouvač ovládá koncové servo (úplně poslední, ke kterému je přichyceno klepeto) a posouvač úplně na spodku ovládá servo otočné báze. Na pravé straně od jednotlivých serv se nachází aktuálně nastavená hodnota.

Pro možnost větší interakce robota a prostředí jsem připravil několik velkých tlačítek, na kterých je umístěn obrázek, který popisuje, co dané tlačítko dělá a v jakém se nachází stavu. Tlačítko nalevo (obsahující obrázek klepet) umožňuje otevírat a zavírat klepeta. Pokud se na tlačítku nachází ikona otevřeného klepeta, znamená to, že je otevřeno, a naopak. Sousední tlačítko umožňuje rozsvítit a zhasnout laserové ukazovátko. Pokud je obrázek šedý, je laserové ukazovátko zhaslé, v opačném případě, kdy je červené, značí stav rozsvíceno. Tento režim oproti následujícímu obsahuje ještě tlačítko na spuštění základního testovacího příkladu. Uživatel si může vybrat ze select tlačítka chtěný projekt a spustí si jej.

Může nastat situace, kdy si uživatel bude chtít vytvořit vlastní projekt, který bude s robotickým ramenem manipulovat. Pro tento případ jsem, pro režim s posouvači, připravil sloupec s tlačítky, umístěný na pravé straně. Tato tlačítka umožní uchovat aktuální stav serv, klepeta a laseru do souboru, vymazání celého souboru, vymazání posledního řádku v souboru nebo možnost exportovat, která pohybovou sekvenci převede na spustitelný program v jazyce Python. Tento skript se následně uloží do spustitelných projektů, mezi kterými si uživatel může vybrat pomocí select tlačítka.

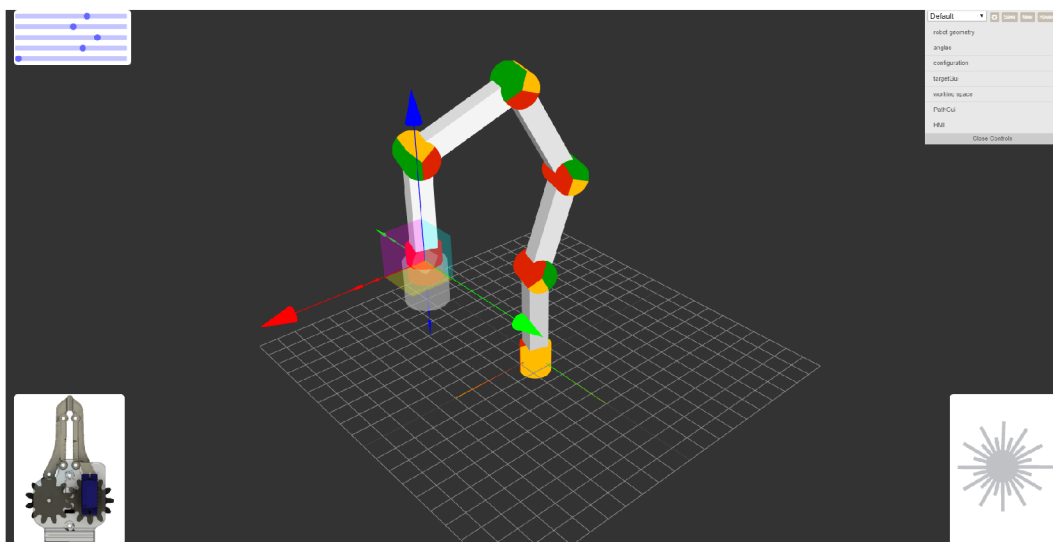


Obrázek 5.14: Uživatelské rozhraní pro ovládání jednotlivých serv.⁴

Aby si uživatel mohl práci s manipulátorem zpestřit, připravil jsem pro něj druhý režim, který mu umožní ovládat rameno pomocí inverzní kinematiky. Aby se do tohoto režimu mohl přepnout, stačí stisknout tlačítko v pravém horním rohu, s ikonou ramene.

Režim inverzní kinematiky

Z důvodu, že zadání nevyžaduje zakomponovat inverzní kinematiku do této práce, rozhodl jsem se převzít již existující řešení, které mi tento typ ovládání umožní. Autorem mnou převzaté části s inverzní kinematikou je Maximilian Beck a je možné jej stáhnout na githubu: <https://github.com/glumb/robot-gui>. Tento projekt jsem lehce poupravil, aby odpovídal mým požadavkům, jako je design manipulátoru, nastavení jednotlivých kloubů, pozicování koncového bodu, aktualizování hodnot z předchozího režimu, komunikace s mým serverem a další...



Obrázek 5.15: Uživatelské rozhraní s inverzní kinematikou.

Pro ovládání stačí pouze najet myší na koncový bod ramene a s myší posouvat. Tento režim nabízí stejná tlačítka pro ovládání klepet a laserového ukazovátka, jako předchozí režim. V případě, kdy se bude chtít uživatel vrátit do režimu s posouvátkem, jsem pro něj připravil opět tlačítko, v levém horním rohu, s ikonou posouvátek.

5.3 Komunikace uživatelského rozhraní s GUI

Dalším cílem, který jsem si vymezil, byla možnost ovládat robotické rameno vzdáleně, pomocí bezdrátové technologie Wi-Fi. Aby mohlo rameno přes Wi-Fi komunikovat, je potřeba, aby i samotná řídicí deska umožňovala tento typ komunikace. Na Raspberry Pi 3B+, kterou je mé robotické rameno ovládáno, tento typ komunikace možný je.

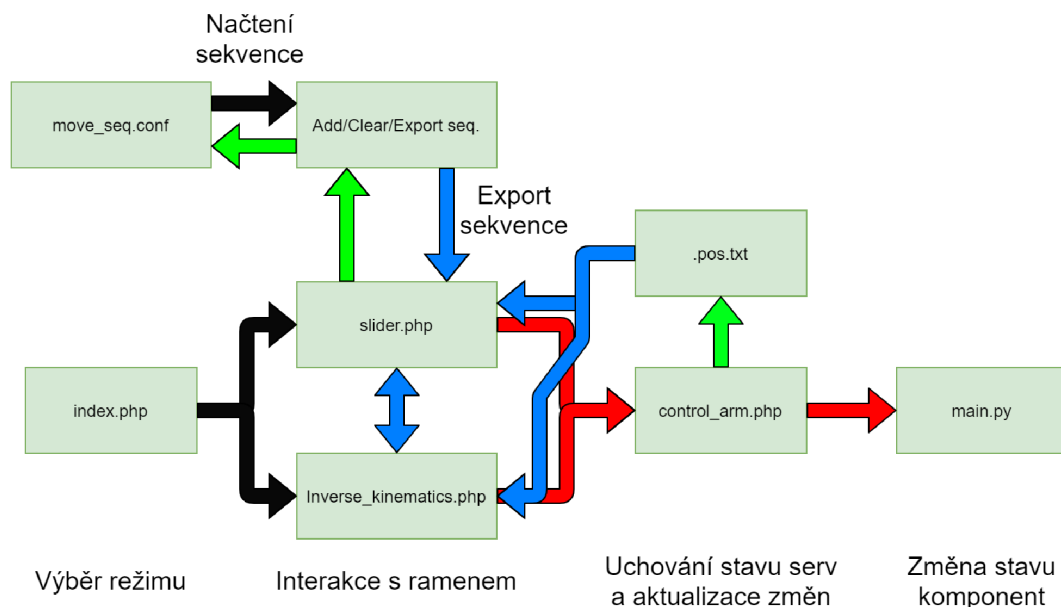
Rozhodl jsem tedy zprovoznit webový server Apache 2, na který se může prakticky kdokoli připojit z lokální sítě. Po připojení se mu zobrazí uživatelské rozhraní v podobě webové stránky, které je popsáno v kapitole 5.2. Ovládání robotického ramene pomocí Apache 2 probíhá v několika krocích:

1. **index.php** - Úvodní stránka (rozcestník), na které uživatel začne při spuštění GUI. Tato stránka mu pomůže inicializovat robota a vybrat si, kterým způsobem chce manipulátor ovládat (slider bary nebo inverzní kinematika).
2. **slider.php** - Uživatel si vybere, zda chce změnit pozici serva pomocí posouvátka, ovládat klepeto, laser nebo zda chce pracovat na vytváření své vlastní pohybové sek-

vence. Po ukončení jedné ze zmíněných akcí dojde k odeslání dat přes javascript do souboru `control_arm.php`. Data jsou zaslána ve formátu: "Akce sekvence" pro práci s pohybovou sekvencí, například: `export`, nebo "Co se ovládá-"hodnota" pro zbylé akce. Pro umístění prvního serva do nulté polohy to bude: `servo1=0`.

3. **control_arm.php** - Úkolem tohoto souboru je převzetí dat ze souboru `index.php`. Načtené hodnoty následně uloží do souboru `.pos.txt`, který slouží k uchování jednotlivých souřadnic všech serv. Jakmile se servo potočí, na základě zásahu uživatele, tak se automaticky uloží do souboru `.pos.txt` a při restartu stránky budou z tohoto souboru načteny. Dalším úkolem je detekce, co se vlastně změnilo (servo, klepato nebo laser) a podle toho spustí script `main.py`. Formát spuštění tohoto scriptu bude vypadat takto: `python main.py -s1 0`.
4. **main.py** - Hlavní script, který umožňuje ovládání jednotlivých komponent robotického ramene, na základě argumentů, se kterými byl spuštěn. Podle kroků, které zde popisují, by se snažil přemístit servo do polohy 0.

Kroky, které zde byly popsány, se vztahují na ovládání se slide bary (posouvátky), ovšem pro možnost s inverzní kinematikou jsou kroky téměř stejné. Obrázek níže ilustruje komunikační tok mezi soubory pro ovládání manipulátora:



Obrázek 5.16: Znázornění toku dat při změně komponenty.

Červenou barvou je znázorněna trasa, která se provede při každé změně komponenty zásahem uživatele. Modré cesty následně znázorňují inicializace při prvním spuštění GUI, přepnutí mezi režimy nebo aktualizování stránky. Zelená barva zobrazuje, kdy dochází k aktualizování hodnoty v souboru s aktuálními hodnotami a daty pro pohybovou sekvenci. Černé šipky znázorňují ostatní akce.

5.4 Dosažené vlastnosti ramene

Po finálním sestavení robotického ramene jsem se rozhodl porovnat výsledné vlastnosti s vlastnostmi, které jsem si vytyčil v kapitole 4.2.

První požadovanou vlastností byl bezproblémový pohyb, a to jak plynulý, tak stabilní. Stabilitou mám na mysli to, aby se rameno nevyklánělo a nehoupalo. Konstrukce je stabilní a pevně drží, zároveň nikde nic nepřekáží v pohybu, tudíž mohu prohlásit tuto vlastnost za splněnou.

Druhou vlastností je jednoduchý design, který si může uživatel doma sám složit. Jelikož stačí vytisknout pouze 4 díly na celou konstrukci ramene, je i tento bod splněn.

Další klíčovou vlastností u manipulátoru je schopnost manipulovat s předměty, proto jsem požadoval, aby byl můj robot schopen, v porovnání s jeho rozměry, uzvednout alespoň 100 g. Pokusy, které jsem s robotem provedl, ukázaly, že je schopen bezproblémové manipulovat až s přibližně 700 g, čímž je i tento bod splněný.

Váha mého manipulátoru je i včetně krabice, ke které je přichycena a ve které se nachází veškerá elektronika, okolo 3160 g. Celá konstrukce je tedy poměrně lehká, není problém s přenosem a díky přidané krabici, která tvoří ze zmíněné váhy celkem velkou část, napomáhá i stabilitě a tím i plynulému pohybu.

Další vlastností, kterou jsem si stanovil, byla schopnost jednoduše odejmout uchopovací mechanismus neboli klepeto. Design, který uživateli poskytuji, lze rozebrat pouhým povolením 3 šroubů. Úchyt samotného klepeta je realizován jednoduchým nasunutím na hliníkový rám serva, takže po odšroubování ho stačí pouze vysunout.

Poslední vlastností bylo poskytnutí rozmanitého ovládání. Jelikož je možné mé robotické rameno ovládat pomocí inverzní kinematiky, ovládání jednotlivých serv zvlášť a také možnost vytvářet si vlastní sekvence, považuji tuto vlastnost za dosaženou.

5.5 Testování robotického manipulátoru

V předchozí kapitole 5.4 jsem diskutoval, zda můj manipulátor splňuje mnou definované požadavky. Pro jejich ověření jsem musel provést řadu testů. Tyto testy spadají do tříd testů, které jsou uvedeny v kapitole 4.4. Jsou jimi:

- **Test funkčnosti 1 -> Testování 100 g závaží:** Rameno jsem uvedl do několika poloh a v každé z nich jsem 100 g objekt jednak položil na uzavřená klepeta, jednak jsem ho uchopil klepety. *Rameno tento test úspěšně zvládlo a dalo se předpokládat, že uzvedne i těžší závaží, proto jsem rozšířil tento experiment o 200 g až 800 g.*
 - 100 g: *Úspěšně*
 - 200 g: *Úspěšně*
 - 300 g: *Úspěšně*
 - 400 g: *Úspěšně*
 - 500 g: *Úspěšně*
 - 600 g: *Úspěšně*
 - 700 g: *Úspěšně*
 - 800 g: *Neúspěšně*

Test byl úspěšný a ukázal se, že rameno je schopno uzvednout i sedminásobnou zátěž. Při zhruba 800 g zátěži jsou však problémy s pohybem značné.

- **Test funkčnosti 2 -> Test synchronizace obou GUI režimů:** Spustil jsem GUI pro ovládání robota. Následně jsem vybral jeden z režimů a zahájil nějakou interakci. Pokud je interakcí laser nebo klepeto, tak jsem si ověřil, že při kliknutí na tlačítko, které při kliku změni svůj obrázek, zůstane tento obrázek i v druhém režimu. V případě pohybu jsem ověřoval, zda hodnoty natočení serv budou vidět i v druhém režimu. *Synchronizace laseru i klepete proběhla úspěšně, aktualizace hodnot z inverzní kinematiky do sliderů také, ovšem aktualizace hodnot ze sliderů do inverzní kinematiky není správná.*
 - Klepeto/laser s přechodem slider -> inverzní kinematika: **Úspěšně**
 - Klepeto/laser s přechodem inverzní kinematika -> slider: **Úspěšně**
 - Hodnoty serv s přechodem slider -> inverzní kinematika: **Neúspěšně** -> Při tomto testu se nevyhodnotila správně dopředná kinematika a 3D model robotického ramene tak není ve správné pozici.
 - Hodnoty serv s přechodem inverzní kinematika -> slider: **Úspěšně**

Výše uvedené výsledky znázorňují, že při testování synchronizace uživatelského rozhraní, nejsou správně hodnoty serv převedeny do inverzní kinematiky. Tento problém může být způsoben nevhodným výpočtem dopředné kinematiky, která se počítá pro počáteční nastavení ramene.

- **Test funkčnosti 3 -> Plynulost pohybu:** Ramenem jsem pohyboval mezi krajními pozicemi a sledoval jsem, zda nedojde k zaseknutí serva či jeho rozhoupání. *Tento test byl úspěšný, rameno prokázalo schopnost plynulého pohybu, bez zasekávání.*
- **Test funkčnosti 4 -> Shoda hodnot slider baru s input polem:** Na slider baru jsem si najel na obě krajní hodnoty. Tyto hodnoty musí být -135 (stupňů) v jednom kraji a 135 v druhém kraji. Tyto hodnoty se musí projevit v input poli. U input pole je dále nezbytné otestovat, zda nelze přesáhnout interval hodnot $\langle -135, 135 \rangle$. *Tento test proběhl úspěšně. V případě nastavení hodnoty na slideru se input pole správně aktualizovalo a naopak také. Při nastavení hodnoty do input pole fungovala i kontrola meze.*
 - Hodnota sliderů změni hodnotu input pole: **Úspěšně**
 - Hodnota input pole změni hodnotu sliderů: **Úspěšně**
 - Hodnoty sliderů se pohybují v intervalu $\langle -135, 135 \rangle$: **Úspěšně**
 - Hodnoty input polí se pohybují v intervalu $\langle -135, 135 \rangle$: **Úspěšně**

Tento test je klíčový a byl úspěšný. V případě, kdy by uživatel tento test neprovedl a chyba by se tam vyskytovala, znamenalo by to nestabilní činnost robota při ovládání.

- **Test kalibrace 1 -> Test střední pozice:** Spustil jsem režim se slidery a nastavil pozici všech serv buď přes slidery, nebo přes input pole na hodnotu 0. Že je test správný, se dá zjistit tak, že veškerá serva ukazují stejným směrem, tedy vzhůru. *Tento test byl úspěšný. V případě, kdyby se uživateli tento test nezdařil, doporučuji provést výše popsany test: Test funkčnosti 4, který je s tímto testem úzce spojený.*

- **Test reakce na zátěž 1 -> Test chování se zátěží a bez ní:** Vybral jsem jeden z režimů pro ovládání ramene (nezáleží, který z nich) a v něm jsem začal pohybovat s ramenem, které nedrželo v klepetech žádný předmět. Tento test jsem provedl znovu, kdy ovšem rameno nějaký předmět drželo. Snahou tohoto testu je zjistit, zda se rameno chová stejně v obou případech. *Tento test je úspěšný, dokud nedojde k překonání maximální váhové hodnoty, která je testována v testu: Test funkčnosti 1.*
- **Test robustnosti 1 -> Test pevnosti modelu:** Spustil jsem uživatelské ovládání v režimu sliderů. S robotem jsem následně pohyboval a prováděl prudké změny, abych otestoval, že je konstrukce dostatečně pevná a při prudké změně se nezlomí. *Pevnost konstrukce mého robota byla otestována úspěšně, kdy robotické rameno prudkým změním odolá. Tento test by bylo lepší realizovat v delším časovém intervalu. V tomto případě by se spíše řadil do testů dlouhodobého užití.*
- **Test uživatelského rozhraní 1 -> Test korektního zobrazení:** Spustil jsem uživatelské rozhraní a vybral jednu z možností ovládání (slider bary nebo inverzní kinematiku). V daném rozhraní jsem změnil stav klepete nebo laseru. Následně jsem přešel do druhého režimu a kontroloval jsem, zda se daný režim spustil s hodnotou klepete a laseru, který jsem nastavil v režimu předchozím. *Tento test prokázal správné zobrazení jednotlivých prvků a také jejich vzájemné propojení mezi režimy v případě režimu se slidery. V režimu s inverzní kinematikou se nesprávně zobrazuje natočení jednotlivých kloubů ramene. Tento test je spojený s testem: Test funkčnosti 2, takže v případě neúspěchu tohoto testu je možné hledat chybu i v tomto testu.*
 - Režim se slidery -> laser/klepeto: **Úspěšně**
 - Režim se slidery -> serva a slidery samotné: **Úspěšně**
 - Režim se slidery -> tlačítko pro vstup do IK režimu: **Úspěšně**
 - Režim s inverzní kinematikou -> laser/klepeto: **Úspěšně**
 - Režim s inverzní kinematikou -> natočení ramene: **Neúspěšně** -> Neúspěch byl způsoben špatným výpočtem dopředné kinematiky při spuštění tohoto rozhraní. *Tento problém nezpůsobuje nefunkčnost uživatelského rozhraní, ale rameno není při prvním pohybu ve správné pozici. Při posunutí s 3D modelem je následně pohyb správný.*
 - Režim s inverzní kinematikou -> tlačítko pro vstup do režimu sliderů: **Úspěšně**

Již při sepisování požadavků mi bylo jasné, že některé testy nebude možné realizovat, jelikož na bakalářskou práci byl pouze rok. Těmito testy jsou zejména testy dlouhodobého užití, u kterých je čas klíčovým faktorem. Testy, které jsem výše uvedl, doporučuji provést všem uživatelům jakmile moje robotické rameno sestojí. Na závěr poskytuji tabulku sumarizující výsledek testu.

Typ testu	Dílčí test	Výsledek
Testování závaží	100 g	Úspěšně
<i>Testování závaží</i>	<i>200 g</i>	Úspěšně
<i>Testování závaží</i>	<i>300 g</i>	Úspěšně
<i>Testování závaží</i>	<i>400 g</i>	Úspěšně
<i>Testování závaží</i>	<i>500 g</i>	Úspěšně
<i>Testování závaží</i>	<i>600 g</i>	Úspěšně
<i>Testování závaží</i>	<i>700 g</i>	Úspěšně
<i>Testování závaží</i>	<i>800 g</i>	Neúspěšně
Synchronizace GUI	Klepeto/laser s přechodem slider -> IK	Úspěšně
Synchronizace GUI	Klepeto/laser s přechodem IK -> slider	Úspěšně
Synchronizace GUI	Hodnoty serv s přechodem slider -> IK	Neúspěšně
Synchronizace GUI	Hodnoty serv s přechodem IK -> slider	Úspěšně
Plynulost pohybu	-	Úspěšně
Shoda slideru s input polem	Hodnota sliderů změní hodnotu Input pole	Úspěšně
Shoda slideru s input polem	Hodnota Input pole změní hodnotu sliderů	Úspěšně
Shoda slideru s input polem	Hodnoty sliderů v intervalu <-135,135>	Úspěšně
Shoda slideru s input polem	Hodnoty input polí v intervalu <-135,135>	Úspěšně
Test střední pozice	-	Úspěšně
Test chování s a bez zátěže	-	Úspěšně
Test pevnosti modelu	-	Úspěšně
Test korektního zobrazení	Režim se slidery -> laser/klepeto	Úspěšně
Test korektního zobrazení	Režim se slidery -> serva a slidery samotné	Úspěšně
Test korektního zobrazení	Režim se slidery -> tlačítko pro vstup do IK režimu	Úspěšně
Test korektního zobrazení	Režim s IK -> laser/klepeto	Úspěšně
Test korektního zobrazení	Režim s IK -> natočení ramene	Neúspěšně
Test korektního zobrazení	Režim s IK -> tlačítko pro vstup do režimu sliderů	Úspěšně

Tabulka 5.1: Přehled vlastností existujících ramen

Výše uvedená tabulka přehledně sumarizuje jednotlivé testy, které jsem v rámci testování mého robotického ramene provedl. Tyto testy jsou přesněji popsány i s postupem realizace testů a dosažených výsledků nad tabulkou.

Kapitola 6

Závěr

Cílem této práce bylo navrhnout a vytvořit mechanickou rameno. Tento cíl byl splněn a výsledkem je robotického ramene s 5 stupni volnosti.

Prostudoval jsem si potřebnou literaturu o řízení modelářských serv i o tom, jak je ovládat počítačem, a výsledek jsem uvedl ve 3. kapitole. Tyto poznatky jsem využil k tvorbě mého robotického ramene a jeho uživatelského rozhraní, pomocí kterého je možné rameno ovládat. Tvorba mého robotického ramene i uživatelského rozhraní je uvedena v kapitole 5. Výsledné vlastnosti mého manipulátoru jsem také diskutoval v 5. kapitole a porovnal s vlastnostmi, které jsem si sám stanovil. Uživateli jsem také poskytl základní testovací příklad, na kterém si může ověřit, že jeho manipulátor funguje dle specifikace.

V rámci zhotovení ramene jsem také ověřil, zda splňuje mnou definované vlastnosti a že se chová tak, jak by měl, pomocí 8 hlavních testů, které doporučuji provést každému, kdo si mého manipulátora postaví. Tyto testy jsou uvedeny v 5. kapitole. Uživateli vřele doporučuji otestovat i případy, které jsem zde nevedl z důvodu rozsahu této práce a omezeného času.

V této práci jsem využil již existující řešení inverzní i dopředné kinematiky. Věděl jsem, že problematika inverzní kinematiky není snadná. Netušil jsem ale, že bude poměrně problematické nalézt funkční řešení, které by bylo možné aplikovat na robotické rameno s více jak 2 stupni volnosti. Řešení, které jsem nakonec našel a použil je nastaveno právě na 5 stupňů volnosti, které má i můj manipulátor. Toto řešení bylo potřeba lehce upravit a samotný výsledek můžete vidět v kapitole 5.

V této práci by bylo možné pokračovat například přidáním senzorů pro detekci aktuální polohy při výpadku elektrické energie a s tím související inicializace ramene při prvním spuštění. Dalším zlepšením by mohlo být například měření proudu jednotlivých serv, aby se dal detekovat kontakt s cizím předmětem a nebyl tak například zraněn člověk.

Seznam obrázků

2.1	Robotické rameno PIarm.	3
2.2	Uživatelské rozhraní pro PiArm.	4
2.3	Robotický manipulátor Dexter.	5
2.4	Uživatelské rozhraní pro rameno Dexter ER2.	6
2.5	Programování ramene Niryo přes Niryo One Studio.	7
2.6	Robotické rameno Niryo.	8
3.1	Modelářské servo.	9
3.2	Znázornění řízení serva na základě délky signálu.	10
3.3	Vývojová deska Arduino Uno.	13
3.4	Vývojová deska ESP32	13
3.5	Přenosný počítač Raspberry Pi 3.	15
3.6	Modul HC-05 s vývody	16
3.7	Manipulátor s přiřazenými souřadnicovými rámci.	17
3.8	Problém více možných řešení inverzní kinematiky.	19
3.9	Program pro modelování 3D modelů - Fusion 360.	20
3.10	Program pro modelování 3D modelů - Solidworks.	21
4.1	Blokové schéma značící strukturu díla.	24
5.1	Blokové schéma s částmi, které musím realizovat.	26
5.2	Prvotní návrh robotického ramene.	27
5.3	Statická báze robotického ramene.	28
5.4	Vymezovač otočné báze přichycený ke statické bázi.	28
5.5	Otočná báze robotického ramene.	29
5.6	Uchycovací prstýnek držící bázi.	29
5.7	Rozložené klepeto k manipulaci s předměty.	30
5.8	Výsledný design robotického manipulátoru.	30
5.9	Servo, které je využito pro klouby a otočnou bázi.	31
5.10	Servo, které je využito pro uchopovací mechanismus.	31
5.11	Koncept grafického rozhraní pro ovládání jednotlivých serv.	32
5.12	Koncept grafického rozhraní pro inverzní kinematiku.	33
5.13	Rozcestník při prvním spuštění GUI.	33
5.14	Uživatelské rozhraní pro ovládání jednotlivých serv.	34
5.15	Uživatelské rozhraní s inverzní kinematikou.	35
5.16	Znázornění toku dat při změně komponenty.	36

Seznam tabulek

4.1	Přehled vlastností existujících ramen	22
5.1	Přehled vlastností existujících ramen	40

Literatura

- [1] *Introduction to direct memory access - Embedded.com* [online]. Cambridge, MA USA: Embedded.com, 2003 [cit. 2020-01-01]. Dostupné z: <https://www.embedded.com/introduction-to-direct-memory-access/>.
- [2] *HC-05-Bluetooth to Serial Port Module* [offline]. Mexiko: iteadstudio, 2010 [cit. 2019-12-20]. Dostupné z: <https://www.electronicaestudio.com/docs/istd016A.pdf>.
- [3] *SolidWorks - 3D CAD navrhování* [online]. Brno, Česká republika: SolidVision, s.r.o., 2011 [cit. 2020-05-22]. Dostupné z: https://www.solidvision.cz/solidworks/?gclid=CjwKCAjw8J32BRBCEiwApQEKgXbzfE01-iH2dLAP0PnqQSXuYxyc1D8gPcYxhfZTYc6_DmQTPioigxoCn4MQAvD_BwE.
- [4] *Cloud Services / Design In The Cloud / Autodesk* [online]. San Rafael, California: Autodesk Inc., 2012 [cit. 2020-05-21]. Dostupné z: <https://www.autodesk.com/360-cloud>.
- [5] *What is a Raspberry Pi?* [online]. Spojené království: Raspberry Pi Foundation, 2014 [cit. 2019-12-29]. Dostupné z: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>.
- [6] *Eses servo motor 9g* [offline]. Česká republika: eses, 2015 [cit. 2020-04-25]. Dostupné z: <https://arduino-shop.cz/docs/produkty/0/741/eses1420669476.pdf>.
- [7] *Buy a Raspberry Pi 3 Model B – Raspberry Pi* [online]. Spojené království: Raspberry Pi Foundation, 2016 [cit. 2019-12-29]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [8] 10 Amazing Pi zero projects. *The MagPi*. Spojené království: Raspberry Pi Press. 2017, č. 61, s. 25–27, [cit. 2019-12-29]. Dostupné z: <https://www.raspberrypi.org/magpi-issues/MagPi61.pdf>.
- [9] *150W Single Output Switching Power Supply* [offline]. BREDA, Nizozemí: Mean Well, 2017 [cit. 2019-12-29]. Dostupné z: <https://www.meanwell-web.com/content/files/pdfs/productPdfs/MW/S-150/S-150-spec.pdf>.
- [10] *Buy a Raspberry Pi Zero W – Raspberry Pi* [online]. Spojené království: Raspberry Pi Foundation, 2017 [cit. 2019-12-28]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>.
- [11] *NodeMCU ESP-32S — Zerynth Docs documentation* [online]. Pisa, Itálie: Zerynth, 2017 [cit. 2019-12-28]. Dostupné z: https://docs.zerynth.com/latest/official/board.zerynth.nodemcu_esp32/docs/index.html.

- [12] *HC-05 Bluetooth Module Pinout, Specifications, Default Settings, Replacements Datasheet* [online]. Components101, 2018 [cit. 2019-12-29]. Dostupné z: <https://components101.com/wireless/hc-05-bluetooth-module>.
- [13] *Niryo One Mechanical Specifications* [offline]. Francie: Niryo, 2018 [cit. 2020-05-21]. Dostupné z: <https://niryo.com/docs/niryo-one/user-manual/mechanical-specifications/>.
- [14] *Arduino Uno Rev3 | Arduino Official Store* [online]. Boston, Massachusetts: Arduino LLC, 2019 [cit. 2019-12-28]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>.
- [15] *Cloud Powered 3D CAD/CAM Software for Product Design | Fusion 360* [online]. San Rafael, California: Autodesk Inc., 2019 [cit. 2019-11-25]. Dostupné z: <https://www.autodesk.com/products/fusion-360/overview#banner>.
- [16] *Fusion 360 Features | Online CAD/CAM Design Software | Autodesk* [online]. San Rafael, California: Autodesk Inc., 2019 [cit. 2019-12-29]. Dostupné z: <https://www.autodesk.com/products/fusion-360/features#electronics>.
- [17] *Niryo One User Manual* [offline]. Francie: Niryo, 2019 [cit. 2020-05-21]. Dostupné z: <https://niryo.com/docs/niryo-one/user-manual/complete-user-manual/>.
- [18] *PiArm: The DIY Robotic Arm for Raspberry Pi* [online]. New York, USA: Kickstarter, 2019 [cit. 2020-01-01]. Dostupné z: <https://www.kickstarter.com/projects/127134527/piarm-the-diy-robotic-arm-for-raspberry-pi>.
- [19] *ARDUINO UNO WiFi REV2 | Arduino Official Store* [online]. Boston, Massachusetts: Arduino LLC, 2020 [cit. 2019-12-29]. Dostupné z: <https://store.arduino.cc/arduino-uno-wifi-rev2>.
- [20] *Lesson 2 - Controlling PiArm with GUI* [online]. Londýn, Spojené království: SB Components Ltd., 2020 [cit. 2020-20-05]. Dostupné z: <https://shop.sb-components.co.uk/blogs/piarm-lessons/lesson-2-controlling-piarm>.
- [21] *Niryo One - Niryo* [online]. Francie: Niryo, 2020 [cit. 2020-01-12]. Dostupné z: <https://niryo.com/niryo-one/>.
- [22] *Pigpio library* [online]. Spojené království: abyz, 2020 [cit. 2020-04-04]. Dostupné z: <http://abyz.me.uk/rpi/pigpio/>.
- [23] *Pigpio library - Python Interface* [online]. Spojené království: abyz, 2020 [cit. 2020-04-04]. Dostupné z: <http://abyz.me.uk/rpi/pigpio/python.html>.
- [24] *Potřebné součásti a systémové požadavky aplikace SOLIDWORKS Visualize* [online]. Paříž, Francie: Dassault Systèmes SE, 2020 [cit. 2020-05-22]. Dostupné z: http://help.solidworks.com/2020/czech/SolidWorks/install_guide/c_viz_prereqs_system_reqs.htm.
- [25] *RaspberryPI models comparison | Comparison tables - SocialCompare* [online]. Francie: SocialCompare, 2020 [cit. 2019-12-29]. Dostupné z: <http://socialcompare.com/en/comparison/raspberrypi-models-comparison>.

- [26] *System requirements for Autodesk Fusion 360 | Fusion 360 | Autodesk Knowledge Network* [online]. San Rafael, California: Autodesk Inc., 2020 [cit. 2020-05-21]. Dostupné z: <https://knowledge.autodesk.com/support/fusion-360/troubleshooting/caas/sfdcarticles/sfdcarticles/System-requirements-for-Autodesk-Fusion-360.html>.
- [27] APOORVE. *Servo Motor: Basics, Theory Working Principle* [online]. Jaipur, India: Circuit Digest, 2015 [cit. 2020-05-06]. Dostupné z: <https://circuitdigest.com/article/servo-motor-basics>.
- [28] ARSLAN, Y. Z. a POPOVIC, M. B. Inverse Kinematics. In: Cambridge, Massachusetts: Academic Press, 2019, s. 305–331 [cit. 2020-03-22]. DOI: <https://doi.org/10.1016/C2016-0-04132-3>. ISBN 978-0-12-812939-5.
- [29] BEHNKE, S. a SCHREIBER, M. *Digital Position Control for Analog Servos* [offline]. Freiburg, Germany: University of Freiburg, 2006 [cit. 2020-03-24]. Dostupné z: <https://pdfs.semanticscholar.org/5a06/88e20f4363f38d9afba65d08bfe6220ce27f.pdf>.
- [30] EARL, B. *Using the Adafruit Library | Adafruit PCA9685 16-Channel Servo Driver | Adafruit Learning System* [online]. New York, USA: Adafruit, 2012 [cit. 2019-12-28]. Dostupné z: <https://learn.adafruit.com/16-channel-pwm-servo-driver/using-the-adafruit-library>.
- [31] FATMA, M. W. a HAMID, M. I. *PWM speed control of dc permanent magnet motor using a PIC18F4550 microcontroller* [offline]. Spojené království, IOP Publishing Ltd: [b.n.], 2019 [cit. 2020-03-21]. Dostupné z: <https://iopscience.iop.org/article/10.1088/1757-899X/602/1/012017/pdf>.
- [32] GE, K. T. *Solving Inverse Kinematics Constraint Problems for Highly Articulated Models*. Waterloo, Ontario, Canada, 2000. [cit. 2020-03-22]. Diplomová práce. University of Waterloo. Dostupné z: <https://cs.uwaterloo.ca/research/tr/2000/19/CS-2000-19.pdf>.
- [33] HAGER, C. *Welcome to RPIO's documentation! — RPIO 0.10.0 documentation* [offline]. pythonhosted.org, 2013 [cit. 2020-01-09]. Dostupné z: <https://pythonhosted.org/RPIO/>.
- [34] HEVENTHAL, B. *Fusion 360 Mastery: How to Use Version Control to Easily Manage Your Design Projects - Fusion 360 Blog* [online]. San Rafael, California: Autodesk Inc., 2012 [cit. 2020-05-21]. Dostupné z: <https://www.autodesk.com/products/fusion-360/blog/fusion-360-mastery-manage-design-projects/>.
- [35] IBRAHIM, D. *Designing Embedded Systems with 32-Bit PIC Microcontrollers and MikroC*. 1. vyd. Butterworth-Heinemann, 2014. ISBN 9780080977867.
- [36] INC., E. *ESP8266EX* [offline]. Shanghai, Čína: [b.n.], 2020 [cit. 2020-04-29]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.
- [37] JIMBLOM. *Bluetooth Basics - learn.sparkfun.com* [online]. Niwot, Colorado: SparkFun Electronics, 2013 [cit. 2019-12-28]. Dostupné z: <https://learn.sparkfun.com/tutorials/bluetooth-basics/all>.

- [38] KING, P. *PiArm review* [online]. Spojené království: The MagPi, 2019 [cit. 2020-20-05]. Dostupné z: <https://magpi.raspberrypi.org/articles/piarm-review>.
- [39] KUCUK, S. a BINGUL, Z. Robot Kinematics: Forward and Inverse Kinematics. In.: 2006, s. 119–135 [cit. 2020-03-21]. DOI: 10.5772/5015. ISBN 3-86611-285-8.
- [40] LE, T. Forward kinematics of a human-arm system and inverse kinematics using vector calculus. In.: červen 2016, s. 1–3 [cit. 2020-03-21]. DOI: 10.1109/ICARCV.2016.7838641.
- [41] MALEWAR, S. *DEXTER ER2 ROBOTIC ARM User Manual* [offline]. Mahape, Indie: NEX Robotics Pvt. Ltd., 2011 [cit. 2020-05-20]. Dostupné z: <https://dokumen.tips/documents/dexter-er-2-robotic-arm-2011-09-30.html>.
- [42] MICHAL ŠUSTEK, P. T. a ÚŘEDNÍČEK, Z. *DC motors and servo-motors controlled by Raspberry Pi 2B* [offline]. Les Ulis, France: EDP Sciences, 2017 [cit. 2020-03-22]. DOI: 10.1051/mateconf/20171250. Dostupné z: https://pdfs.semanticscholar.org/00cf/a6b13303c9215862b4198f17f6b6a369f94f.pdf?_ga=2.24153658.1837385061.1588761153-573100397.1586259322.
- [43] TIRA THOMPSON, E. *Digital Servo Calibration and Modeling* [offline]. Pittsburgh, Pensylvánie: [b.n.], 2009 [cit. 2020-03-24]. Dostupné z: https://www.ri.cmu.edu/pub_files/2009/3/CMU-RI-TR-09-41.pdf.