

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F2**

Fakulta strojní  
Ústav přístrojové a řídicí techniky

## Konstrukce a řízení robotického ramene

**Jiří Kasal**

Vedoucí: Ing. Cyril Oswald, Ph. D.  
Obor: Automatizace a průmyslová informatika  
Srpen 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kasal** Jméno: **Jiří** Osobní číslo: **484134**  
Fakulta/ústav: **Fakulta strojn**  
Zadávající katedra/ústav: **Ústav přístrojové a řídicí techniky**  
Studijní program: **Automatizační a přístrojová technika**  
Specializace: **Automatizace a průmyslová informatika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Konstrukce a řízení robotického ramena**

Název diplomové práce anglicky:

**Design and control of robotic arm**

Pokyny pro vypracování:

Navrhněte a realizujte konstrukci 6osého robotického ramena pro přepravu lehkých malých břemen, navrhněte a realizujte systém jeho řízení za využití ROS.

- Proveďte rešerši konstrukčních řešení 6osých robotických ramen včetně možností motorizace
- Proveďte rešerši možností řízení robotických ramen se zaměřením na využití ROS
- Navrhněte a realizujete konstrukci 6osého robotického ramena
- Navrhněte a realizujte řízení

Seznam doporučené literatury:

- [1] L. Zhi and M. Xuesong, 'Navigation and Control System of Mobile Robot Based on ROS,' 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2018, pp. 368-372, doi: 10.1109/IAEAC.2018.8577901. (<https://ieeexplore.ieee.org/document/8577901>)
- [2] Ali, H. M., Hashim, Y., Al-Sakkal, G. A. (2022). Design and implementation of Arduino based robotic arm. International Journal of Electrical and Computer Engineering, 12(2). <https://doi.org/10.11591/ijece.v12i2.pp1411-1418>
- [3] Ting, H. Z., Zaman, M. H. M., Ibrahim, M. F., Moubark, A. M. (2021). Kinematic Analysis for Trajectory Planning of Open-Source 4-DoF Robot Arm. International Journal of Advanced Computer Science and Applications, 12(6). <https://doi.org/10.14569/IJACSA.2021.0120690>

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Cyril Oswald, Ph.D. U12110.3**

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

Datum zadání diplomové práce: **28.04.2023**

Termín odevzdání diplomové práce: **15.08.2023**

Platnost zadání diplomové práce: \_\_\_\_\_

Ing. Cyril Oswald, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Tomáš Vyhlídal, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta





## Poděkování

Děkuji Ing. Cyrilu Oswaldovi, Ph. D. za trpělivost a věcné rady.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 10. srpna 2023

## Abstrakt

Práce se zabývá návrhem konstrukce a řízení robotického ramene s 6 stupni volnosti. Součástí je také návrh cykloidních převodovek pro jednotlivé klouby ramene. Pro řízení je využita vývojová deska STM32 Nucleo, pro kterou byl v práci napsán řídicí program. Dále se zde zabýváme implementací The Robot Operating System (ROS), který zajistí jednoduché plánování pohybu ramene.

**Klíčová slova:** Robotické rameno, ROS, STM32, Cykloidní převodovka, 3D tisk

**Vedoucí:** Ing. Cyril Oswald, Ph. D.

## Abstract

The thesis deals with the design and control of a robotic arm with 6 degrees of freedom. It also includes the design of cycloidal gearboxes for individual joints of the arm. The STM32 Nucleo development board is used for control, for which the control program has been written in the thesis. We also discuss the implementation of The Robot Operating System (ROS) which will provide simple motion planning of the arm.

**Keywords:** robotic arm, ROS, STM32, Cycloidal gearbox, 3D printing

**Title translation:** Design and control of robotic arm

# Obsah

<b>1 Úvod</b>	<b>1</b>		
<b>2 Rešerše</b>	<b>3</b>		
2.1 Historie	3		
2.2 Dělení průmyslových robotů a manipulátorů	5		
2.2.1 Dělení dle autonomie	5		
2.2.2 Dělení dle kinematiky	6		
2.3 Existující aplikace průmyslových robotických ramen	10		
2.4 Jednotlivé příklady 3D tištěných robotů	14		
<b>3 Návrh ramene</b>	<b>19</b>		
3.1 Specifikace požadavků na navrhované robotické rameno	19		
3.2 Volba motorů a převodových mechanismů	19		
3.2.1 DC motor	21		
3.2.2 BLDC motor	21		
3.2.3 Krokový motor	22		
3.2.4 Servomotory	23		
3.2.5 Volba motorů	24		
3.2.6 Převodové mechanismy	24		
3.2.7 Návrh konstrukce ramene	27		
3.2.8 Návrh prvního prototypu cykloidní převodovky	29		
3.3 Návrh prvního prototypu ramene	32		
3.4 Návrh vylepšeného prototypu	33		
3.5 Návrh dvojité cykloidní převodovky	35		
3.5.1 Testování nového prototypu ramene	36		
3.6 Testování prototypů	37		
3.6.1 Testování nových převodovek	37		
<b>4 Hardware</b>	<b>39</b>		
4.1 Volba mikrokontroleru	39		
4.1.1 Arduino nano	39		
4.1.2 STM32	40		
4.2 Volba driveru	42		
4.3 Koncové senzory	44		
4.4 Zdroj	44		
<b>5 Software</b>	<b>45</b>		
5.1 První řízení	45		
5.2 Řízení STM32	46		
5.2.1 STM32CubeIDE	46		
5.2.2 Psaní programu v C++ v prostředí STM32CubeIDE	46		
5.2.3 Návrhový class diagram	47		
5.2.4 Motor controller	48		
5.2.5 Plánování pohybu - Kinematika	51		
5.2.6 Komunikace	56		
5.3 ROS	57		
5.3.1 Framework ROS	57		
5.3.2 Struktura ROS	58		
5.3.3 Námi využívané programy a funkce	59		
5.3.4 ROS Implementace	61		
<b>6 Závěr</b>	<b>65</b>		
<b>A Literatura</b>	<b>67</b>		

## Obrázky

2.1 Robotický písář . . . . .	4	3.16 Řez dvojitou cykloidní převodovkou . . . . .	36
2.2 Průmyslový robot UNIMATE . . . . .	4	3.17 Boční pohled na řez dvojitou cykloidní převodovkou . . . . .	37
2.3 Chirurgický robot Da Vinci . . . . .	5	3.18 Prototyp ramene s dvojitou cykloidní převodovkou . . . . .	38
2.4 Stupně volnosti v prostoru . . . . .	7	4.1 Rozložení pinů na Arduino Nano	40
2.5 Letecký simulátor . . . . .	8	4.2 Rozložení pinů ST Morpho a Arduino . . . . .	41
2.6 Uzavřený kinematický řetězec . . . . .	8	4.3 DRV8825 . . . . .	42
2.7 Otevřený kinematický řetězec . . . . .	9	4.4 CNC shield V3.0 . . . . .	43
2.8 Sférické a cylindrické souřadnice vzhledem ke kartézským . . . . .	9	4.5 Napájecí deska XH-M229 . . . . .	44
2.9 Robotické rameno ABB - IRB 1100 a jeho pracovní oblast . . . . .	11	5.1 Návrhový class diagram . . . . .	48
2.10 Robotické rameno UR3e a jeho pracovní oblast . . . . .	12	5.2 Metoda pro výpočet rychlosti . . . . .	49
2.11 Robotické rameno Kuka - KR 4 a jeho pracovní oblast . . . . .	13	5.3 Metoda pro dobrzdění do pozice	50
2.12 Robotické rameno FANUC - CR-4iA a jeho pracovní oblast . . . . .	14	5.4 Metoda pro kontrolu sepnutí spínače . . . . .	50
2.13 Robotické rameno Arctosrobotics a jeho rozměrové schéma . . . . .	15	5.5 Metody pro řízení více motorů . . . . .	51
2.14 Robotické rameno KAUDA a jeho rozměrové schéma . . . . .	16	5.6 Systémy souřadnic F1 a F2 . . . . .	52
2.15 Robotické rameno BCN3D . . . . .	17	5.7 Definice souřadných systémů D-H úmlouvou . . . . .	55
3.1 Návrhové schéma ramene . . . . .	20	5.8 Node pro seriovou komunikaci s STM32 a topics (témata) se kterými pracuje . . . . .	58
3.2 Převod na lineární pohyb za pomoci rotačního servomotoru . . . . .	20	5.9 Rviz s modulem MoveIt . . . . .	59
3.3 Schéma DC motoru . . . . .	21	5.10 MoveIt Setup Assistant . . . . .	60
3.4 Schéma BLDC motoru . . . . .	22	5.11 Graf nodů a topiců pro simulaci v Gazebo . . . . .	62
3.5 Schéma krokového motoru . . . . .	23	5.12 Graf nodů a topiců při řízení reálného ramene . . . . .	63
3.6 Využití motory v rameni . . . . .	24		
3.7 Schéma planetového stupně . . . . .	25		
3.8 Princip postupné rotace harmonické převodovky . . . . .	26		
3.9 Základní části cykloidní převodovky . . . . .	27		
3.10 Návrhové schéma s posunutým kloubem 3 . . . . .	28		
3.11 Řez částí prvního prototypu cykloidní převodovky . . . . .	30		
3.12 Orientační parametry 3D tisku	31		
3.13 Řez prototypem cykloidní převodovky . . . . .	32		
3.14 Model a sestavený prototyp ramene . . . . .	33		
3.15 Řez cykloidní převodovkou s řemenicí (kloub 3) . . . . .	34		

## Tabulky

2.1 Typické kinematické dvojice . . . . .	7
3.1 Důležité parametry motorů . . . . .	25
5.1 Popis jednotlivých parametrů	
Denavit-Hartenbergovy úmluvy . . . . .	54
5.2 Naše D-H parametry . . . . .	56



# Kapitola 1

## Úvod

Robotika je v současné době často zmiňovaný termín díky nástupu technologií průmyslu 4.0, tedy automatizací výroby a dalších procesů. Robotická ramena jsou jednou z klíčových částí této automatizace, která umožňují například manipulaci s předměty, svařování a další kroky výroby, které by jinak musel dělat člověk.

Rozvoj robotiky a robotických ramen začíná převážně v druhé polovině 20. století, kdy byly vyvinuty první průmysloví roboti schopni vykonávat jednoduchou opakující se činnost. Od té doby došlo k velkému pokroku ve všech oblastech spojených s robotikou. Nové motory umožňují robotickým ramenům dosahovat vyšších přesností, výkonnější mikrokontrolery vylepšují možnosti řízení a technologie spojené s umělou inteligencí vylepšují roboty na novou úroveň. [1]

S rozvojem těchto technologií stoupá i mezi „obyčejnými“ lidmi zájem o tyto témata a možnosti jejich využití. Pořízení takového robotického ramene pro vlastní testování je ovšem velmi nákladná záležitost a tato překážka zastaví mnoho lidí, aby se o tuto problematiku zajímala dále. Tak jak se postupně rozvíjí technologie 3D tisku, objevila se nová možnost, jak si poměrně levně vyrobit robotické rameno. Díky této aditivní výrobě jsme nejen schopni vytvářet jednoduché části robota, ale vzhledem ke způsobu výroby jsme schopni navrhovat i složité geometrické tvary, které by byly jinak velmi nákladné na výrobu. S využitím těchto možností jsme schopni vytvořit robotické rameno uzpůsobené přesně našim požadavkům.

Cílem této diplomové práce je navrhnout koncept robotického ramene, které bude dále možné používat pro různé úlohy dle potřeby a bude snadno vyrobitelné. Aby bylo možné rameno lehce modifikovat a vylepšovat, bude kladen důraz právě na využití technologie 3D tisku. U návrhu ramene není kladen důraz na vysokou přesnost, ale na využití dostupných komponent a technologií, tak aby bylo možné výsledné rameno jednoduše duplikovat.

Jelikož návrh celého ramene je komplexní záležitost, budou během vývoje vytvořeny jednotlivé prototypy robota, které budou po následném testování vylepšeny. I když je samotný mechanický návrh velmi důležitý, bez řídicího systému by se jednalo pouze o nefunkční celek. Proto se práce dále zabývá návrhem elektroniky a softwaru pro řízení. Vzhledem k nspecifikovanému účelu robota by mělo rameno být schopné se i po softwarové stránce přizpůsobit případným požadovaným změnám.

Práce je strukturována do třech částí:

- řešení problému návrhu robotického ramene a následný jeho návrh
- hardware a software pro řízení námi vytvořeného ramene
- Robot Operating System (ROS)



# Kapitola 2

## Rešerše

### 2.1 Historie

Robotika je obor, který se zabývá designem robotů, jejich výrobou a řízením. Přestože o opravdové průmyslové robotice můžeme mluvit až od druhé poloviny 20. století, zmínky o umělých pracujících dělnících najdeme i ve více vzdálené historii.

Jednu z prvních zmínek najdeme již v řecké mytologii, kde si podle legend bůh kovářství Hefaistos vyráběl mechanické sluhy, dělníky nebo například robota Talose, který měl dohlížet na Krétu. Pokud se zaměříme na první návrhy opravdu fungujících robotů, můžeme zmínit robotického ptáka poháněného párou, kterého sestavil řecký matematik Archytas z Tarentu přibližně ve 4. st. př. n. l. [2]

Pokud se přesuneme až do 15. století vynálezcem byl Leonardo da Vinci, který kolem roku 1495 navrhl robota proporcčně podobného člověku. Mělo se jednat o jakéhosi rytíře, který by zvládal hýbat pažemi, otáčet hlavou a vykonávat podobné jednoduché úkony. Toto zatím byly pouze určité návrhy nebo velmi jednoduché konstrukce. Ovšem v 18. století švýcarští mistři Pierre a Henry Droz vyrobili prvního androida, tedy robota připomínajícího člověka. Tento robot představoval písaře, který pohyboval psacím perem, viz obrázek 2.1 [3, 4, 5]

Zatím tyto stroje neměly žádné specifické pojmenování. S průlomem a novým pojmem, který všechna tyto zařízení sjednotil, přišel v roce 1920 spisovatel Karel Čapek. Ve své divadelní hře R.U.R. (Rossumovi univerzální roboti) použil slovo robot pro označení umělých bytostí, které zastávaly práci lidí a nakonec i lidstvo přemohly. Samotné vymyšlení slova je připisováno Josefu Čápkovi, bratrovi Karla Čapka. Původně zvažovali označení „laboři“, což je odvozeninou od latinského slova "lagor magnus" pro těžkou práci. Nakonec zvolili známé označení robot, které je odvozené ze staroslovanského jazyka. [1, 2, 6]

Od této doby se používá slovo robot pro popis automatizovaných zařízení. Ing. Ivan M. Havel, CSc. definoval robota takto: „Robot je automatický nebo počítačem řízený integrovaný systém, schopný autonomní cílově orientované interakce s přirozeným prostředím, podle instrukcí od člověka. Tato interakce spočívá ve vnímání a rozpoznávání tohoto prostředí a v manipulování



**Obrázek 2.1:** Robotický písář [4]



**Obrázek 2.2:** Průmyslový robot UNIMATE [7]

s předměty, popř. v pohybování se v tomto prostředí.“ [6]

První opravdu použitelné aplikace průmyslových strojů, o kterých se dá říci, že spadají do oblasti robotiky, se začínají objevovat až v polovině 20. století. Jednalo se například o teleoperátory sloužící k manipulaci s radioaktivními nebo jinými nebezpečnými materiály. Velký průlom v této oblasti nastal v roce 1961, kdy byl uveden do provozu první průmyslový robot UNIMATE vytvořený firmou General Motors. Tohoto robota můžeme vidět na obrázku 2.2. Od této doby se robotika rychle rozvíjí a stává se z ní běžný prostředek využívaný pro automatizaci výrobních procesů. Díky velkým pokrokům v technologiích, umělé inteligenci a pokročilým sensorům vznikají sofistikovanější a specializovanější roboti pro náročnější operace, než je pouhá manipulace. V současnosti se roboti a robotická ramena používají pro svařování, nanášení barev, ale mimo to i v jiných oborech než jen v průmyslu, například ve zdravotnictví, výzkumu a v armádě. [7, 1, 2]

Ve zdravotnictví pomáhají chirurgičtí roboti s náročnějšími operacemi, což umožňuje přesné a minimálně invazivní zákroky. Tedy snižují nejen riziko případných komplikací, ale také zkracují čas nutný pro následnou rekonvalescenci pacientů. Jedním z takovýchto robotů je systém Da Vinci, který můžeme vidět na obrázku 2.2. Jedná se o víceramenný laparoskopický



**Obrázek 2.3:** Chirurgický robot Da Vinci [1]

operační systém, který je ovládán lékařem sedícím u řídicí konzole. Tento lékař má k dispozici joysticky, kterými je schopný přesně ovládat jednotlivá ramena robota. Dalšími příklady ze zdravotnictví jsou například robotické protézy.

S postupujícím vývojem umělé inteligence můžeme očekávat i velký paralelní vývoj v oblasti robotiky. Roboti budou více schopni se sami učit, adaptovat a lépe interagovat s lidmi a se svým okolím. S touto rostoucí sofistikovaností a větší mírou autonomie budeme v budoucnu nejspíše svědky ještě většího uplatnění a integrace robotů nejen v průmyslu, ale i v běžném životě.

## 2.2 Dělení průmyslových robotů a manipulátorů

Rozdělení robotických ramen je možné na základě několika kritérií, která zohledňují například konstrukční provedení, aplikační možnosti, míru autonomie, kinematické struktury, použité pohony a další faktory. Jedna z prvních používaných klasifikací se zaměřovala na rozdíly mezi manipulátory a roboty z hlediska jejich řízení a programování. Byly definovány pojmy [8]:

- Manipulátor (často nazýván jednoúčelový manipulátor, manipulátor s pevným programem)
- Synchronní manipulátor (člověk v ovládací smyčce, master - slave manipulátor)
- Robot (manipulátor s programem schopným flexibilních zásahů a změn)
- Adaptivní robot (robot schopný reagovat na změny ve svém pracovním prostředí)
- Kognitivní robot (robot s určitou, i když ne zcela definovanou, mírou autonomie)

### 2.2.1 Dělení dle autonomie

Průmyslové roboty můžeme dle jejich stupně autonomie dělit na pět generací [9]:

- Nultá generace – roboti bez zpětné vazby. V případě jakýchkoli odlišností od očekávaného stavu se robot zastaví.
- První generace – mají naučené postupy pro několik možných scénářů.
- Druhá generace – roboti mají schopnost optimalizovat, tedy vybírat nejvhodnější program pro řešení aktuálního problému.
- Třetí generace – roboti jsou schopni si sami upravovat svůj řídicí program, ovládají schopnost učit se. Mají jasně daný cíl a snaží se ho dosáhnout.
- Čtvrtá generace – autonomní roboti, chovají se podobně jako lidé, dokáží si volit aktuálně potřebné cíle.

### ■ 2.2.2 Dělení dle kinematiky

Dle kinematiky můžeme zavést několik různých způsobů dělení.

#### ■ Dělení dle počtu stupňů volnosti

Robotická ramena se skládají z jednotlivých kinematických dvojic. Každá tato dvojice je charakterizována počtem stupňů volnosti, které nabízí. Tedy počtem nezávislých posuvů a rotací, které mohou členy této dvojice konat. V tabulce 2.1 vidíme typické kinematické dvojice a jejich stupně volnosti.

Výsledný počet stupňů volnosti prostorového kinematického řetězce (robotického ramene) spočítáme rovnicí [10]:

$$i \geq 6 \cdot (n - 1) - \sum_{j=1}^5 j \cdot d_j \quad (2.1)$$

- $i$  – počet stupňů volnosti kinematického řetězce
- $n$  – počet členů ramene včetně rámu
- $j$  – třída kinematické dvojice (počet stupňů volnosti, které daná dvojice odebírá)
- $d_j$  – počet kinematických dvojic dané třídy

Počet stupňů volnosti robotického ramene ovlivňuje schopnost pohybu ramene v prostoru a shoduje se v případě robotického ramene typu RRR (viz kapitola 2.2.2) s počtem os, dle kterých se může rameno otáčet. Často se počet os, tedy stupňů volnosti, shoduje s počtem motorů, ale není to pravidlem. Některé konstrukce mají pro posílení dva motory ovládající stejnou osu. Do stupňů volnosti se nepočítá případný úchopový pohyb koncového zařízení na robotickém rameni.

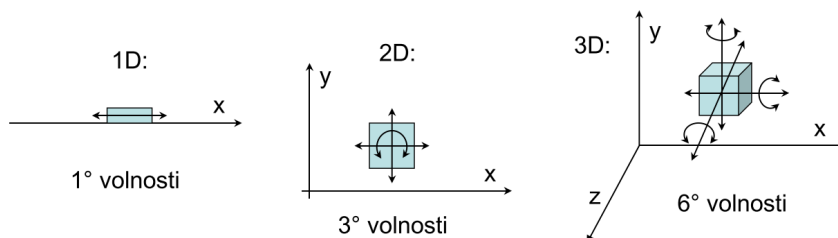
S počtem stupňů volnosti rostou i manipulační schopnosti robota, respektive jeho koncové části. Dle toho, jaký pohyb by mělo rameno vykonávat, tolik potřebuje stupňů volnosti, viz obrázek 2.4. Tedy obecně pro manipulátor

Název kinematické dvojice	Schéma	Počet stupňů volnosti
Rotační		1
Posuvná		1
Šroubová		1
Valivá		1
Válcová		2
Sférická		3
Rovinná		3
Válec na rovině		4
Obecná		5

Tabulka 2.1: Typické kinematické dvojice [10]

platí, že pro pohyb v 1D prostoru (1 stupeň volnosti tělesa) bude zapotřebí minimálně jeden motor. Pro pohyb v 2D prostoru (3 stupně volnosti) jsou zapotřebí již motory 3. Rameno navrhované v této práci bude určeno pro plnou schopnost manipulace s předmětem v 3D prostoru (6 stupňů volnosti – 3 translační pohyby a 3 rotační). Tedy musí mít těchto 6 stupňů volnosti, nebo více.

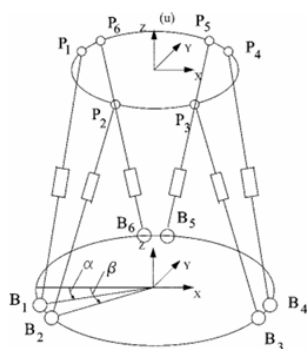
Obecně lze říci, že s rostoucím počtem stupňů volnosti rostou manipulační schopnosti robota, ale také se zvyšuje složitost řídicího systému. Pokud má robot větší počet stupňů volnosti než nutně potřebuje, je tzv. redundantní. [11, 10]



Obrázek 2.4: Stupně volnosti v prostoru [11]



**Obrázek 2.5:** Letecký simulátor [12]



**Obrázek 2.6:** Uzavřený kinematický řetězec [13]

### ■ Dělení dle kinematického řetězce

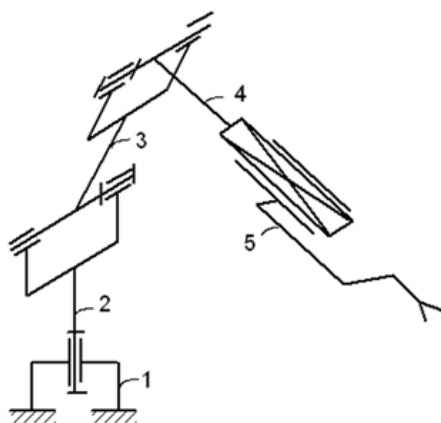
Dle kinematického řetězce, na jehož principu roboti pracují, je možné zavést dělení do 3 skupin.

- Sérioví roboti – otevřený kinematický řetězec - případ našeho robotického ramene. Viz obrázek 2.7.
- Paralelní roboti – uzavřený kinematický řetězec. Viz obrázek 2.6 (využití v praxi můžeme vidět na obrázku 2.5).
- Hybridní roboti – kombinují oba typy řetězců.

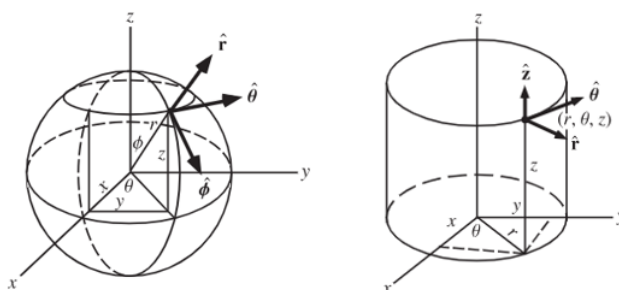
### ■ Dělení dle použitých kinematických dvojic

Robotické manipulátory můžeme podle uspořádání prvních třech kinematických dvojic rozlišovat na [10, 11]:

- TTT – tři translační dvojice – Jedná se o konstrukci s kartézským souřadnicovým systémem, která se často využívá v oblasti 3D tisku. V této konstrukci stačí zajistit lineární pohyb koncového nástroje ve třech osách. V případě 3D tiskáren je k vidění varianta, kdy ve směru svislé osy se nepohybuje koncový nástroj (extruder), ale výrobek samotný.



Obrázek 2.7: Otevřený kinematický řetězec [10]



Obrázek 2.8: Sférické a cylindrické souřadnice vzhledem ke kartézským [15]

- RTT – rotační a dvě translační dvojice – pohybuje se v cylindrických souřadnicích
- RRT – dvě rotační a jedna translační dvojice - Scara konstrukce – pohybuje se v polárních souřadnicích
- RRR – tři rotační dvojice - Kloubové rameno, nejčastější pro manipulátory s 6 stupni volnosti

### ■ Dělení dle pracovních oblastí a převody mezi nimi

Podle kinematiky daného robota se mění i jeho pracovní prostor (souřadnicový systém). Můžeme se setkat se souřadnicovým systémem kartézským, cylindrickým nebo sférickým (obrázek 2.8), který je v případě robotických ramen používán nejčastěji. [14]

Jelikož je pro nás nejběžnější využívat kartézské souřadnice, je nutné znát převody mezi těmito systémy.

Kartézské souřadnice → Cylindrické souřadnice

$$r = \sqrt{x^2 + y^2} \quad (2.2)$$

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (2.3)$$

$$z = z \quad (2.4)$$

Kartézské souřadnice → Sférické souřadnice

$$r = \sqrt{x^2 + y^2 + z^2} \quad (2.5)$$

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (2.6)$$

$$\phi = \arccos\left(\frac{z}{r}\right) \quad (2.7)$$

Cylindrické souřadnice → Kartézské souřadnice

$$x = r \cdot \cos(\theta) \quad (2.8)$$

$$y = r \cdot \sin(\theta) \quad (2.9)$$

$$z = z \quad (2.10)$$

Sférické souřadnice → Kartézské souřadnice

$$x = r \cdot \cos(\theta) \cdot \sin(\phi) \quad (2.11)$$

$$y = r \cdot \sin(\theta) \cdot \sin(\phi) \quad (2.12)$$

$$z = r \cdot \cos(\phi) \quad (2.13)$$

## 2.3 Existující aplikace průmyslových robotických ramen

Pro roboty a robotické manipulátory najdeme v průmyslovém odvětví mnoho možností využití. Pokud budeme uvažovat pouze robotická ramena, kterými se převážně zabýváme v této práci, můžeme zmínit několik typických příkladů využití.[16]

### Paletizační robotické rameno

Jelikož proces paletování některých výrobků je velmi stereotypní, přímo se vybízí využívat robotické manipulátory, nebo pro tvarově složitější produkty přímo robotická ramena.

### Podavače materiálu, polotovarů

Robotická ramena mohou být součástí výrobní linky, kde například mohou zakládat ocelové tyče do obráběcích strojů a následně odebírat hotový výrobek.

### Svářecí roboti

Tyto roboti mají velké uplatnění například v automobilovém průmyslu. Díky velké přesnosti ramen není poté výroba pouze rychlejší, ale také kvalitnější a bezpečnější.



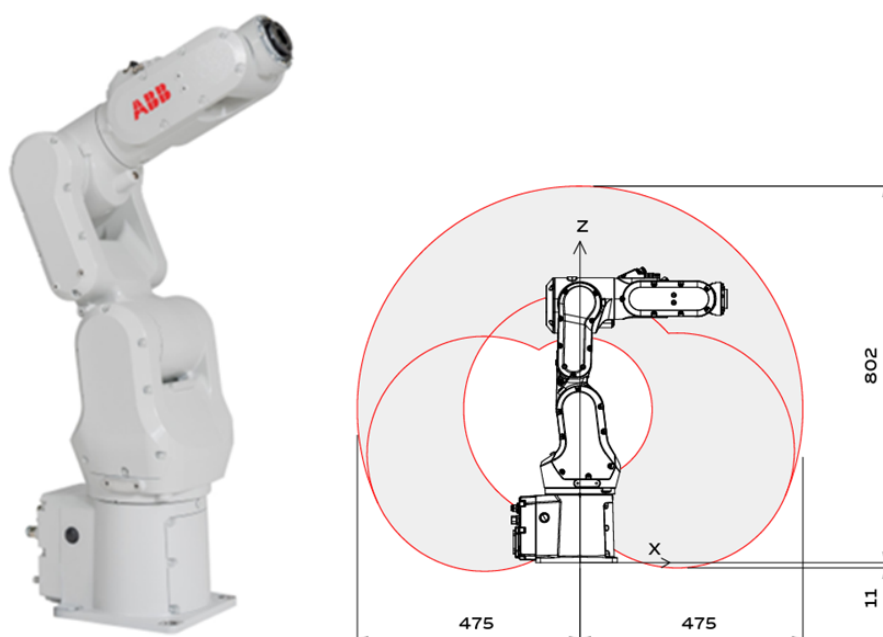
## Kontrola kvality

Jako koncový efektor může být použita například dotyková sonda, která nám v kombinaci s kamerou dokáže s využitím umělé inteligence zajistit dostatečnou kontrolu kvality.

## Jednotlivé příklady průmyslových robotů

S rostoucí poptávkou po robotických ramenech roste i počet jejich výrobců. Jak již bylo zmíněno, existuje spousta druhů robotických ramen, proto zde v příkladech reálných průmyslových ramen bude uvedeno jen několik zástupců od nejvýznamnějších výrobců. Vzhledem k povaze této práce budou vybírána ramena s minimálně 6 stupni volnosti a s maximálním poloměrem pracovního prostoru 0,6 m. Pro možnost porovnání jednotlivých ramen zde budou také uvedeny hodnoty opakovatelnosti pozice, celková hmotnost robota a nosnost robota. Nosnost při plném vyložení ramene je typicky menší a je možné ji dohledat v dokumentaci jednotlivých ramen. Pro přehlednost zde uvádíme pouze charakteristickou nosnost ramene, která nesmí být přesažena pro zachování přesnosti.

### ABB - IRB 1100 [17]

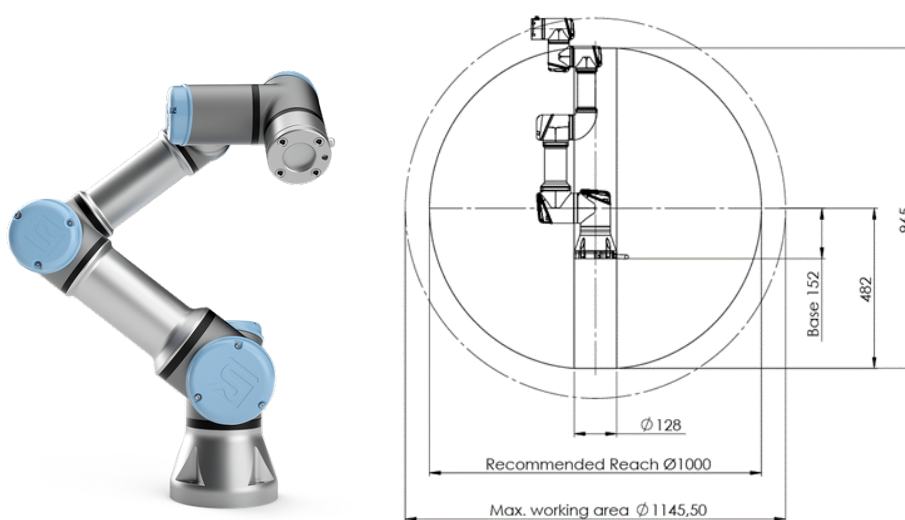


**Obrázek 2.9:** Robotické rameno ABB - IRB 1100 a jeho pracovní oblast [17]

- Dosah ramene: 0,475 m
- Hmotnost: 21 kg

- Charakteristická nosnost: 4 kg
- Počet os: 6
- Opakovatelnost pozice: 0,01 mm

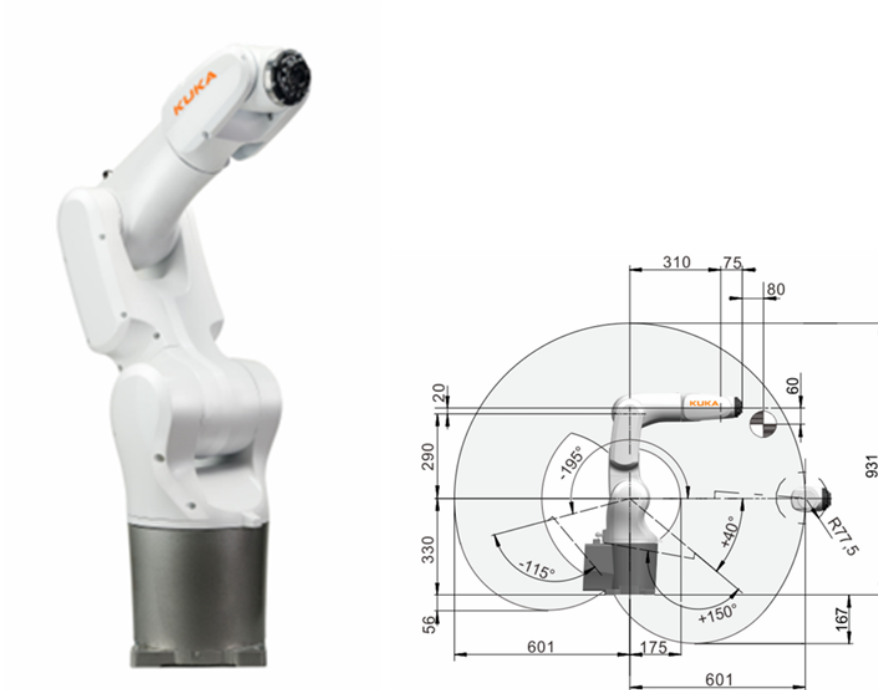
■ **Universal Robots UR3e [18]**



**Obrázek 2.10:** Robotické rameno UR3e a jeho pracovní oblast [18]

- Dosah ramene: 0,5 m
- Hmotnost: 11,2 kg
- Charakteristická nosnost: 3 kg
- Počet os: 6
- Opakovatelnost pozice: 0,03 mm

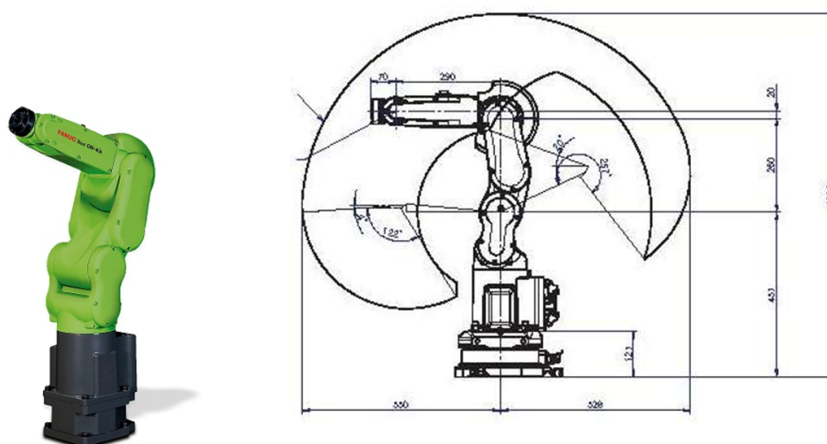
■ Kuka Robotics - KR 4 R600 [19]



Obrázek 2.11: Robotické rameno Kuka - KR 4 a jeho pracovní oblast [19]

- Dosah ramene: 0,6 m
- Hmotnost: 27 kg
- Charakteristická nosnost: 3 kg
- Počet os: 6
- Opakovatelnost pozice: 0,015 mm

## FANUC Robotika - CR-4iA [20]



**Obrázek 2.12:** Robotické rameno FANUC - CR-4iA a jeho pracovní oblast [20]

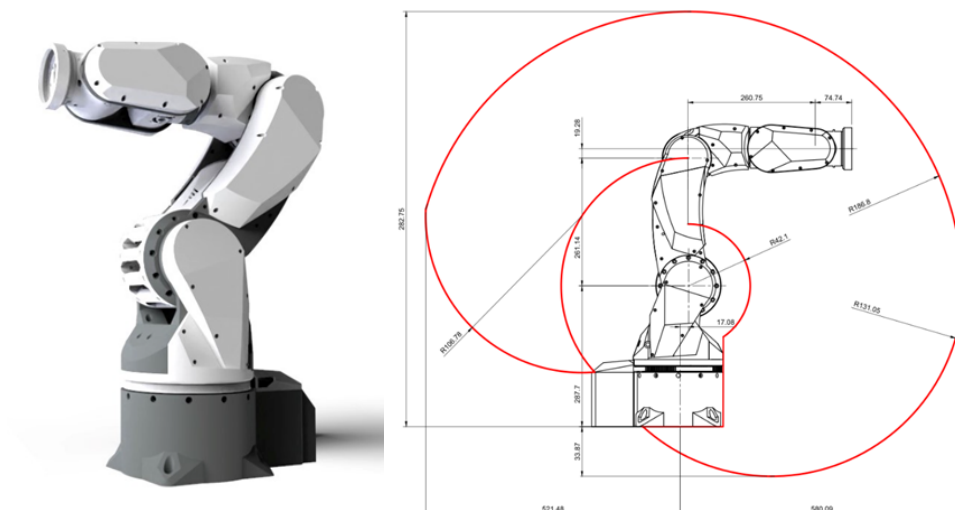
- Dosah ramene: 0,55 m
- Hmotnost: 48 kg
- Charakteristická nosnost: 4 kg
- Počet os: 6
- Opakovatelnost pozice: 0,01 mm

Zmínění zástupci robotických ramen od jednotlivých firem jsou svými parametry vcelku podobní. Pokud se ale podíváme na schémata pracovních oblastí jednotlivých typů, jsou zde již patrné rozdíly. Robotická ramena ABB, Kuka a FANUC mají pracovní oblast velmi podobnou. Tento fakt je dán konstrukčním řešením těchto ramen, které je z pohledu maximálních úhlů natočení jednotlivých os velmi podobné. Tyto ramena mají první a druhou část ramene umístěnou přímo v zákrytu, což má za následek omezení maximálních úhlů natočení jednotlivých os. U ramene FANUC je mírně omezenější pohyb v zadní části, ale vzhledem k možnosti otáčení kolem svislé osy, nemusí být tento fakt nikterak omezující. Naproti tomu rameno UR3e má oproti ostatním odlišnou konstrukci. Druhá část ramene je oproti první odsazená, díky tomu jsou maximální možné natočení jednotlivých os mnohem větší než u ostatních ramen.

## 2.4 Jednotlivé příklady 3D tištěných robotů

Existuje již mnoho projektů, které se zabývají právě návrhem 3D tištěného robotického ramene.

## ■ Arctosrobotics

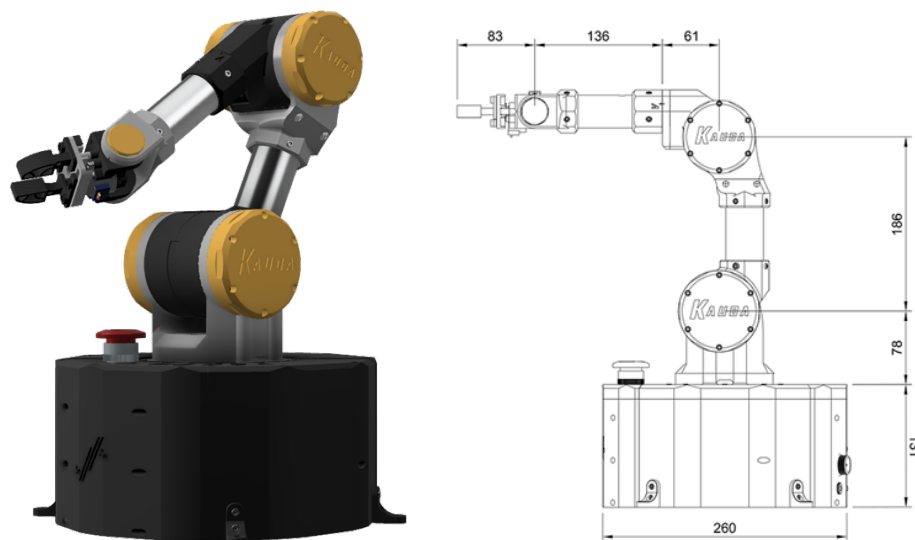


**Obrázek 2.13:** Robotické rameno Arctosrobotics a jeho rozměrové schéma [21]

- Dosah ramene: 0,6 m
- Charakteristická nosnost: 0,5 kg
- Počet os: 6
- Cena za CAD soubory: 40 Eur

Toto rameno využívá jako aktuátory krokové motory: 2x Nema23, 4x Nema17. Pro řízení je využita vývojová deska Arduino Mega, která je osazena modulem CNC shield V3. Pro dosažení dostatečných momentů v osách 2 a 3 jsou použity cykloidní převodovky s předřazeným řemenovým převodem. Zajímavě řešené je i umístění motoru pro osu 4, který je umístěn vedle motoru pro osu 3, čímž vzniká problém s přenosem momentu přes osu 3. To je vyřešeno pomocí prostorově vedeného řemenového převodu. Takováto konstrukce zřetelně vylepší umístění těžiště a tím i maximální únosnost. Obdobným způsobem je vyřešen přenos momentu pro osu 6. [21]

## ■ KAUDA PRO VERSION



**Obrázek 2.14:** Robotické rameno KAUDA a jeho rozměrové schéma [22]

- Dosah ramene: 0,4 m
- Charakteristická nosnost: 0,3 kg
- Počet os: 5
- Cena za CAD soubory: open source

V tomto rameni jsou použity pouze 3 krokové motory Nema17 a 2 servomotory. Pro osy 2 a 3 jsou tentokrát využity planetové převodovky. Pro řízení je použita stejná kombinace vývojové desky a modulu jako u předešlého typu, tedy Arduino Mega + CNC shield. Rozdílem je přítomnost raspbberyy Pi 3, které se dále může starat o plánování tras, zatímco arduino se stará pouze o řízení motorů a vyhodnocování enkodérů. [22]

## ■ BCN3D MOVEO



**Obrázek 2.15:** Robotické rameno BCN3D [23]

- Dosah ramene: 0,4 m
- Charakteristická nosnost: 0,7 kg
- Počet os: 5(6)
- Cena za CAD soubory: open source

I v tomto rameni je pro řízení použito Arduino Mega, které se stará o ovládní 6 krokových motorů. Použity jsou krokové motory Nema23 a Nema17. Veškeré převody jsou řešeny pouze řemenovými převody, což z ramene dělá konstrukčně méně složitý celek. Originální projekt navrhuje 5osé rameno, ale existuje mnoho variant úprav, které zvyšují momenty jednotlivých kloubů nebo přidávají 6 osu. [23]





## Kapitola 3

### Návrh ramene

#### 3.1 Specifikace požadavků na navrhované robotické rameno

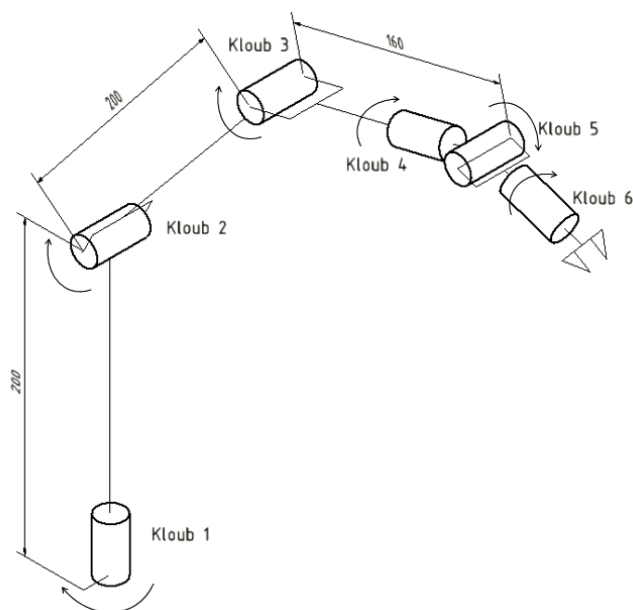
Tato část práce bude zaměřena přímo na samotný návrh robotického ramene. Pro další případné aplikace jsou specifikovány některé důležité požadavky na výsledné rameno, další parametry si volíme. Požadované specifikace charakterizují kinematickou strukturu jako 6osé rameno, které bude mít operační prostor v okolí 0,4 m od základny ramene.

Rameno by mělo být schopné po osazení správným úchopovým mechanismem unést lehké předměty a pokud to bude možné, mělo by zvládat unést kameru pro prostorové vidění. Jelikož tato kamera nebyla blíže specifikována, bude při návrhu počítáno s hmotností kamery Realsense D455, která váží 100 g. Tato kamera by vyžadovala i držák a kabely vedoucí ke kameře, které by též přidaly hmotnost, proto uvažujeme přidanou zátěž kamerou 200 g. Při uvažování hmotnosti neseného předmětu 200 g získáváme únosnost 400 g, na kterou bude nutné rameno minimálně konstruovat. Dále se umístěním kamery a její aplikací v návrhu konstrukce ramene zabývat nebudeme.

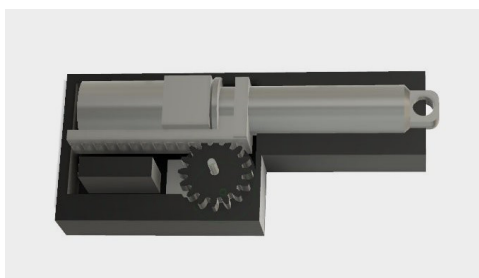
Jelikož není striktně definováno, pro co konkrétně by mělo rameno být využito, budeme se v průběhu návrhu snažit řešit konstrukci posledních členů tak, aby bylo možné je jednoduše nahradit jinými členy uzpůsobených konkrétní aplikaci. Návrh a výběr jednotlivých dílů také uzpůsobíme snaze dosáhnout cenově dostupného celku. Prvotní návrh možné rozměrové struktury ramene je zobrazen na obrázku 3.1.

#### 3.2 Volba motorů a převodových mechanismů

Při výběru motorů pro robotické aplikace je nutné zvážit několik faktorů, které pro naši konkrétní aplikaci budou důležité. Základním specifikačním parametrem je druh kinematické vazby, pro kterou bude motor určen. Pokud by se jednalo o lineární vazbu, bude nutné využít úplně jiný typ pohonu než pro vazbu rotační či kloubovou. Pokud ovšem pohonem uvažujeme spojení motoru s převodovkou, je možné, že samotný motor bude shodný, ale konstrukce převodovky bude uzpůsobena chtěnému pohybu, viz obrázek 3.2.



**Obrázek 3.1:** Návrhové schéma ramene



**Obrázek 3.2:** Převod na lineární pohyb za pomoci rotačního servomotoru [24]

Další parametr, který je nutné zvážit, je hluk, jaký jsme ochotni tolerovat při chodu robota. Jelikož navrhujeme pouze malé rameno, neočekáváme komplikace s tímto parametrem. Pokud bychom ale navrhovali velké průmyslové rameno, je již nutné uvažovat i tento faktor při výběru motoru.

Pokud bychom chtěli, aby bylo rameno schopné pouze jednoduchých operací, jako je přenášení objektů, stejně je nutné dosáhnout určité požadované přesnosti. Tento parametr je v robotice jedním z nejdůležitějších, proto je nutné ho vždy zohlednit i při volbě motoru.

S velmi přesným ramenem, které by ovšem nezvládlo unést námi specifikovanou zátěž, bychom nesplnili zadání. Je proto nutné sledovat i možný kroutící moment, který zvládne motor vyvinout. Kroutící moment je možné zvýšit volbou správné převodovky, poté se ale musí kontrolovat, zda již neklesly otáčky pod námi akceptovatelnou hranici.

V našem případě budeme volit mezi motory:

- DC motor
- BLDC motor

- Krokový motor
- Servomotor

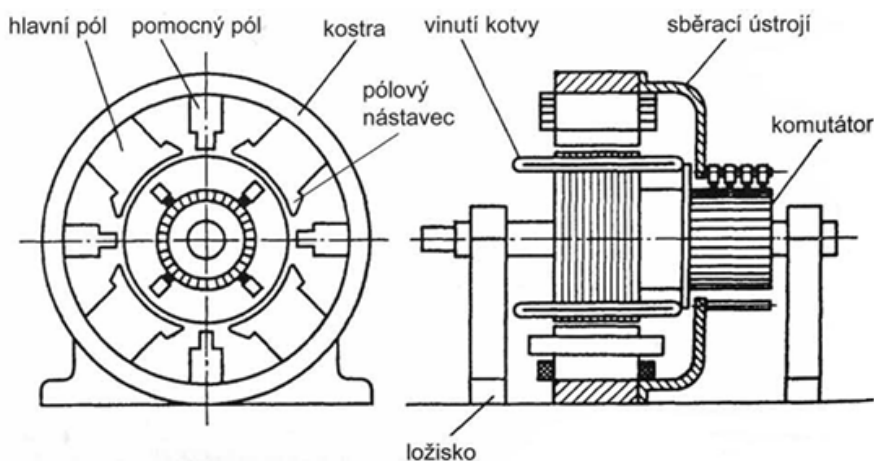
### ■ 3.2.1 DC motor

Stejnoseměrný motor je elektrický motor, který využívá ke svému fungování stejnosměrného proudu. Jedná se o nejstarší elektrický motor, který byl vyroben. Tento motor se skládá ze tří částí: statoru, rotoru a komutátoru. Stator slouží k vytvoření magnetického pole, čehož je docíleno pomocí permanentního magnetu nebo statorového vinutí. Schéma konstrukce je vidět na obrázku 3.3.

Rotor je i s komutátorem umístěn na stejné hřídeli. Vinutí rotoru tvoří cívky, které jsou uloženy v drážkách rotoru. Jednotlivé vývody cívek rotoru jsou vyvedeny na dotykové plochy komutátoru, který otáčením přepíná převod vnějšího proudu do různých cívek rotoru. Vlivem magnetického pole od statoru poté dochází k rotaci.

Toto konstrukční uspořádání má své výhody a nevýhody. Mezi výhody můžeme uvést jednoduchou regulaci a řízení tohoto motoru ve srovnání s ostatními typy motorů. Výstupní hodnoty otáček a výkonu můžeme snadno měnit změnou vstupního stejnosměrného napětí.

Mezi nevýhodou můžeme označit přítomnost komutátoru. Při jeho rotaci dochází k tření mezi ním a kartáči, což z něj dělá součástku, která se časem opotřebovává. Tato nevýhoda dělá motor náročnějším na údržbu například oproti motorům typu BLDC. Další nevýhoda je také spojena s přenosem energie na komutátoru. Mezi kartáči a komutátorem dochází ke vzniku odporu, což způsobuje ztráty a úbytek výkonu. [25, 26]



Obrázek 3.3: Schéma DC motoru [25]

### ■ 3.2.2 BLDC motor

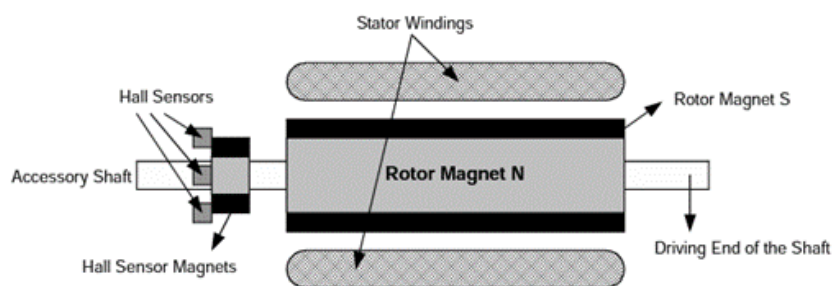
Bezkartáčové stejnosměrné motory, známé jako BLDC (Brushless Direct Current) jsou moderní motory, které mají oproti staršímu typu stejnosměrných

motorů mnoho výhod.

Konstrukčně se BLDC motory skládají z rotoru a statoru, jak je vidět na obrázku 3.4. Stator obsahuje po obvodu vinutí, které tvoří sudé počty pólů. Existuje více možných variant statorového vinutí, jako například trapezoidální a sinusoidální, které se od sebe odlišují tvarem zpětné motorické síly a průběhem fázového proudu.

Tyto motory mají také velmi dobré charakteristiky točivého momentu a rychlosti. Mohou být provozovány v rychlostech až do 150 % jejich nominální rychlosti, přičemž točivý moment zůstává relativně stabilní. Oproti tomu, vzhledem k vyšší setrvačnosti rotoru, je při brždění a rozběhu motoru potřeba vyšší kroutící moment než je jeho nominální hodnota.

Mezi další výhody BLDC motorů patří jejich dlouhá životnost, efektivita a nízká hlučnost. Pro svoji životnost jsou používány například v pevných discích v počítačích. Pokud bychom tyto motory porovnávali s klasickými DC motory s kartáči, jsou BLDC méně náročné na údržbu a mají i vyšší výkon. V současné době je BLDC vhodnou alternativou v mnoha oblastech, kde se dříve využívaly jiné motory, jako příklad můžeme uvést ruční vrtačky, brusky a podobné ruční nářadí.[27]

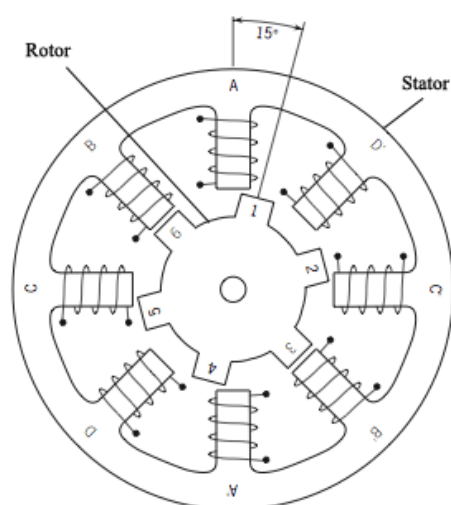


Obrázek 3.4: Schéma BLDC motoru [27]

### 3.2.3 Krokový motor

Krokový motor je speciální druh bezkartáčového stejnosměrného motoru. Skládá se z rotoru a statoru, což je vidět na obrázku 3.5. Stator je tvořen několika cívkami, které se starají o vytváření magnetického pole, kterým přitahují zuby na rotoru. Při přepínání jednotlivých cívek dosáhneme postupné rotace vždy o jeden krok. Velikost tohoto kroku je závislá na konkrétní konstrukci motoru. Toto diskrétní rozdělení otáček motoru generuje typický zvuk a vibrace, které jsou s krokovými motory spojené. Tyto motory můžeme rozdělit na unipolární a bipolární. Zatímco unipolární motor pracuje pouze s jednou polaritou proudu, bipolární motor má směr průtoku proudu cívkou střídavý. Díky této konstrukci má bipolární motor větší kroutící moment, ale je složitější pro řízení.

Mezi hlavní výhody krokových motorů můžeme uvést jejich přesnost, dobrou říditelnost a velký rozběhový moment. Nevýhodou oproti tomu je možnost



**Obrázek 3.5:** Schéma krokového motoru [28]

ztráty kroku, čímž dojde ke změně natočení motoru oproti očekávanému. Tento problém jde vyřešit přidáním senzoru pro snímání natočení motoru. Také jako nevýhodu můžeme vnímat větší energetickou náročnost.

Pro řízení krokových motorů se využívají různé drivery a při výběru tohoto typu motoru je nutné vybrat i správný driver pro jeho řízení. Tyto drivery většinou umožňují i možnosti mikrostepu, díky čemuž jeden krok můžeme rozdělit na několik dílčích a tím zjemnit chod samotného motoru. [28, 29]

### ■ 3.2.4 Servomotory

Tyto elektrické pohony se vyznačují vysokou přesností a možností regulace polohy a rychlosti. Skládají se z elektromotoru, řídicího obvodu a případného napájecího měniče. Pokud budeme porovnávat servomotory s krokovými motory, zjistíme, že servomotory jsou vždy zapojeny v uzavřené regulační smyčce se zpětnou vazbou. Tedy jedním z parametrů, dle kterých se při výběru servomotorů můžeme řídit, jsou i parametry samotného řídicího obvodu, jako je přesnost a rychlost regulační smyčky. V závislosti na použité konstrukci servomotorů existuje více typů. [30]

#### ■ Stejnoseměrné servomotory

Tyto motory jsou ideální pro svou snadnou regulaci pomocí změny napájecího napětí. Používají se obvykle ve spojení s kartáčovým stejnosměrným motorem. Tento typ motoru je blíže popsán v kapitole 3.2.1.

#### ■ Komutovaný servomotor

Tento typ využívá stejnosměrný motor bez komutátoru. O těchto motorech se zmiňuje kapitola 3.2.2.

### ■ Asynchronní servomotor

Asynchronní motory jsou hojně využívané pohony v technice. S rozvojem řídicí techniky se začaly používat i v servosystémech. Tyto motory jsou většinou napájeny třífázovým proudem a vynikají svou robustností a spolehlivostí. V našem případě těžko využitelné, jednak z důvodu napájení a také, protože se nevyrábí v tak malých výkonech.

### ■ Synchronní servomotor

Synchronní servomotory mají rotor otáčející se synchronně s magnetickým polem statoru. Jsou menší a lehčí než asynchronní motory. Používají permanentní magnety na rotoru a stator je napájen třífázovým sinusovým proudem.

### ■ 3.2.5 Volba motorů

Jelikož navrhujeme pouze malé robotické rameno a není zadán požadavek na jeho přesnost, jeví se jako nejlepší volba využít kombinaci krokových a servomotorů. Vzhledem ke snaze snížit náklady na výrobu, využijeme dostupné krokové motory ze staré 3D tiskárny a k nim následně navrhne vhodné převodovky. Pro pohon osy 1 a 2 jsou zvoleny silnější krokové motory 42BYGHW811, viz obrázek 3.6a, pro osu 3 zůstává menší z motorů. O pohon os 4,5 a 6 se budou starat modelářské servomotory HK 15288A, viz obrázek 3.6b. Parametry těchto motorů jsou napsány v tabulce 3.1.



(a) : Krokový motor 42BYGHW811

(b) : Servo motor HK 15288A

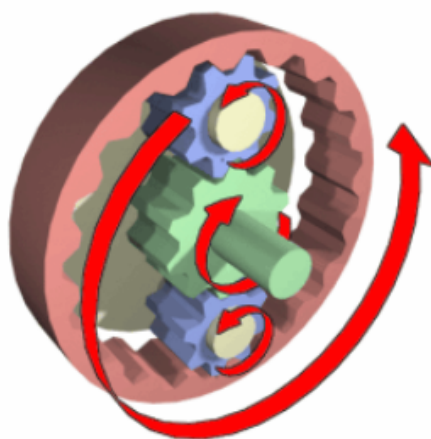
**Obrázek 3.6:** Využité motory v rameni

### ■ 3.2.6 Převodové mechanismy

K motorům bude nutné navrhnout převodové mechanismy, které by měly mít dostatečný převodový poměr, tak aby docílili potřebného momentu v jednotlivých osách ramene. Převodových mechanismů existuje více druhů, ale

Motor	Model	Počet	Úhel kroku	I	Přidrzný moment	Váha	Odpor fáze
Nema17	42BYGHW811	2	1,8°	2,5 A	0,48 Nm	0,34 kg	1,25 Ω
Nema17	42BYGHW609	1	1,8°	1,7 A	0,4 Nm	0,26 kg	1,5 Ω
Motor	Model	Počet	V	Kroucí moment	Váha	Rozměry	
Servo	HK 15288A	3	4,8 - 6 V	0,8 Nm	0,05 kg	40x20x39 mm	

**Tabulka 3.1:** Důležité parametry motorů [31]



**Obrázek 3.7:** Schéma planetového stupně [32]

pokud potřebujeme docílit velkého převodového poměru s malým zastavěným rozměrem, bude nutné využít některé méně běžné řešení. Tato část je zaměřena na rešerši možných reduktorů využívaných v robotice. Následně v části zabývající se konstrukcí budeme z těchto poznatků čerpat.

### ■ Planetová převodovka

Vícestupňová planetová převodovka je tvořena jednotlivými planetovými převody, které jsou spolu provázány. Samotný planetový převod je složen z centrálního kola, unašeče satelitů a korunového kola. Centrální kolo, korunové kolo a unašeč satelitů sdílejí společnou osu, zatímco satelity jsou uloženy mimo centrální osu na unašeči a zasahují jak do centrálního, tak do korunového kola.

Princip činnosti planetové převodovky spočívá v tom, že jedna z tří částí rotujících kolem společné osy je zastavena, zatímco zbývající dvě slouží jako vstup a výstup převodovky (viz obrázek 3.7). [32]

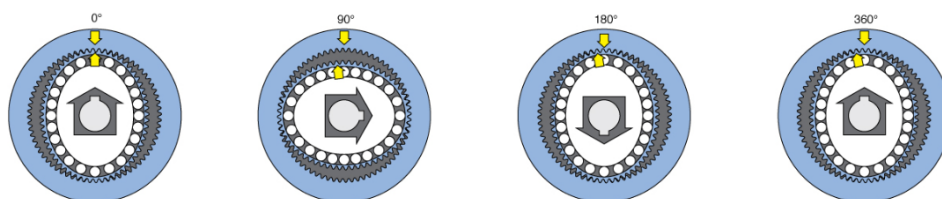
Dle konstrukce je možné měnit převodový poměr změnou statické části převodovky. Toho je dosaženo zabrzděním nebo odbrzděním jednotlivých

částí převodovky. Pro brzdění se obvykle používají lamelové spojky.

### ■ Harmonická převodovka

Princip fungování harmonické převodovky je založen na elasticitě materiálů. Celá převodovka se skládá ze tří hlavních částí, kterými jsou generátor vln (wave generator), pružné ozubené kolo (flexspline) a pevné ozubené kolo (circular spline). Princip fungování této převodovky vidíme na obrázku 3.8.

Pružné ozubené kolo je v průměru o něco menší a má o dva zuby méně než pevné ozubené kolo. Generátor vln eliptického tvaru způsobuje, že zuby pružného kola zapadají do pevného ozubeného kola na protilehlých stranách. Vzhledem k rozdílnému počtu zubů, dojde při otočení generátoru vln o  $180^\circ$  k posunutí pružného kola o jeden zub vůči pevnému kolu. Tedy při plném otočení dojde k posunu o dva zuby. Díky tomu, že jsou zuby převodovky vždy plně do sebe zapojeny, mají harmonické převodovky nulovou zpětnou vůli. Tato konstrukce umožňuje řádově větší převodové poměry než planetová převodovka, není tolik složitá na konstrukci a má teoreticky nulovou vůli. Jejím problémem je právě pružné ozubené kolo, které musí být vytvořeno z takového materiálu, aby vydrželo cyklicky probíhající deformace.[33]



**Obrázek 3.8:** Princip postupné rotace harmonické převodovky [33]

### ■ Cykloidní převodovka

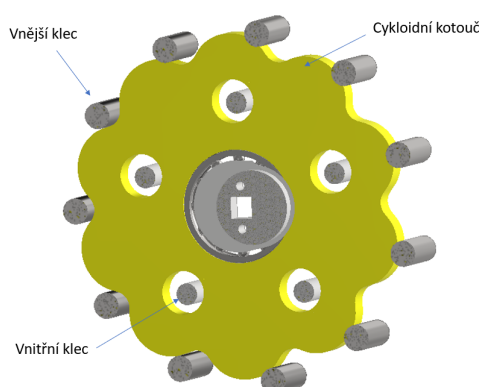
Cykloidní převodovka se skládá ze čtyř částí. Tyto části jsou excentrická vstupní hřídel, dva vačkové kotouče, korunové kolo a výstupní hřídel, jak můžeme vidět na obrázku 3.9. Tato převodovka může být také konstruována jako dvoustupňová, kde je zde navíc za vstupní hřídel předřazen planetový převod a excentrické hřídele jsou poté 2 až 3. Excentrická hřídel pohání vačkové kotouče, které jsou vůči sobě otočený o  $180^\circ$ . Tyto kotouče se odvalují po vnějším korunovém kole, které má o jeden profil více než je na cykloidních vačkách. Díky tomu dochází k pomalému otáčení celé vačky vůči pevnému kolu. Tento pohyb je poté přenášen na výstupní hřídel. [34, 35]

Převodový poměr se určí na základě počtu válečků korunového kola a počtu cykloid na vačkovém kotouči. Pro výpočet platí tato rovnice:

$$i = \frac{n}{N - n} \quad (3.1)$$

- $n$  - počet zubů kotouče





**Obrázek 3.9:** Základní části cykloidní převodovky

- $N$  - počet válečků korunového kola
- $i$  - převodový poměr

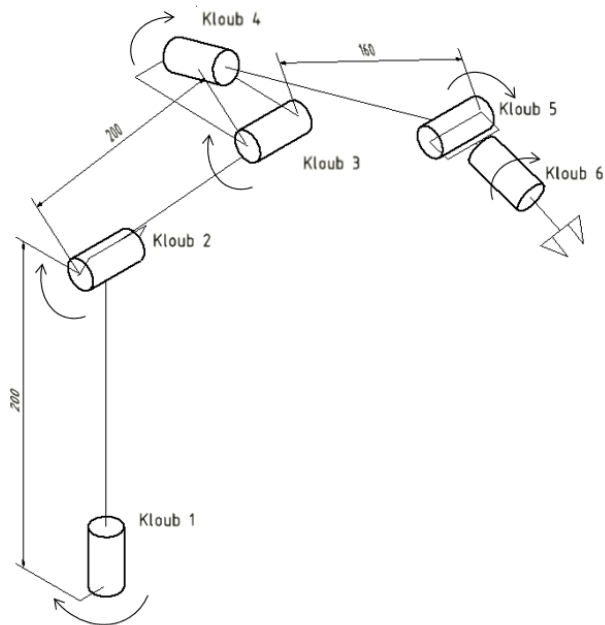
Cykloidní převod je odolný proti nárazovému zatížení, poskytuje vysokou přesnost a vysokou tuhost. Často se využívá v robotice v kombinaci se servopohony, zvláště tam, kde je nutné přenést velké kroučící momenty. Oproti planetovým převodovkám má toto řešení nižší hmotnost, jelikož konstrukce nevyžaduje víceúrovňový převod, ale je možné navrhnout cykloidní převodovku s velkým převodovým poměrem. Pro menší převodové poměry se spíše využívají planetové převodovky, pro jejich snazší výrobu a větší účinnost.

### ■ 3.2.7 Návrh konstrukce ramene

Při zahájení návrhu konstrukce jsme již znali návrhové rozměry ramene, viz obrázek 3.1. Na základě předpokladu využití dostupných krokových motorů je  $i$  bez výpočtu jasné, že bude nutné minimálně pro krokové motory navrhnout převodové mechanismy. Tímto se zbývá první část návrhu, kde navrhne některá možná řešení a následně je otestujeme.

### ■ Návrh převodovek

Pro robotické rameno jsme zvolili kombinaci 3 servomotorů a 3 krokových motorů. Servomotory mají pracovní rozsah  $180^\circ$  a mají dostatečnou sílu, aby mohly zprostředkovávat pohyb na posledních třech osách ramene, aniž by bylo nutné pro ně navrhovat reduktor. Pro přesunutí těžiště blíže k podstavě ramene je možné tyto serva umístit dále od os, které ovládají, a pohyb k nim přenést pomocí ozubených řemenových převodů nebo například pomocí bowdenů. Ozubené řemeny jsou vyrobeny z elastického materiálu s výřezy ve tvaru zubů, které zapadají do řemenic umístěných na hnací a hnané hřídeli. Při využití řemenů je nutné řešit jejich napínání. Doporučuje se využívat ozubených řemenů do převodového poměru 6. Bowdenův mechanismus oproti tomu dokáže díky stacionárnímu a posuvnému prvku zajistit přenos na delší vzdálenosti. Tento mechanismus se často využívá například pro ovládání brzd



**Obrázek 3.10:** Návrhové schéma s posunutým kloubem 3

jízdního kola. Jako pohyblivý prvek se využívá lanko, které je pružné, což v případě pohybu robotického ramene není příliš vhodné.

Jelikož váha těchto motorů (50 g) je v porovnání s uvažovanou únosností nízká, zvolíme umístění servomotorů přímo k osám, kterými pohybují. Tím získáme v porovnání s ostatními možnostmi horší rozložení váhy, ale odstraníme další mechanismy, které by potenciálně mohly být náchylné na poruchu. Přestože nevyužijeme možnosti vzdáleného přenosu kroutícího momentu pro jednotlivé osy, můžeme upravit konstrukční návrh tak, abychom aspoň částečně vylepšili polohu těžiště. Kloub 4 přesuneme až ke kloubu 3 tak, aby se servomotor pro 4 osu nacházel při vzpřímené poloze ramene v horizontálním směru za pozicí osy 3. Tato úprava je vidět na schématu 3.10. Touto úpravou přesuneme pozici těžiště blíže ke středu ramene a díky principu dvouramenné páky dokonce mírně pomůžeme ose 3, která nebude potřebovat takový kroutící moment pro zvedání ramene.

Pro první tři klouby jsme pro pohon zvolili krokové motory. I v tomto případě je možné umístit motory například přímo na podstavu robota a přenos kroutícího momentu řešit pomocí zmíněného řemenového převodu nebo bowdenového mechanismu. V případě prvního kloubu je ale velmi jednoduché motor umístit přímo k ose tak, aby byl zároveň umístěn na podstavě ramene. Tedy případné posunutí motorů vůči osám kloubů zbývá vyřešit v případě druhé a třetí osy. Jelikož nepředpokládáme příliš velké otáčky ramene, můžeme i druhý motor umístit přímo do osy ramene. Tím zvýšíme moment setrvačnosti, se kterým se potýká první kloub, ale vzhledem k očekávaným nízkým otáčkám v tomto není nutné vidět problém. Dále v návrhu konstrukce ramene bude dokonce i motor pro první osu přesunut

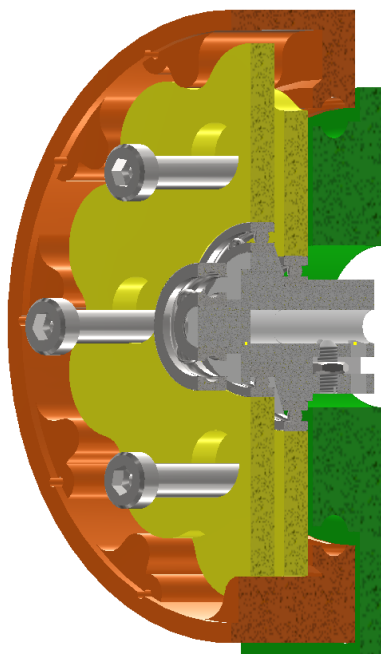
z podstavy. Nevyřešený tedy zůstává pouze krokový motor pro pohon třetí osy. Jelikož tento motor oproti servomotorům je již celkem těžký, budeme předpokládat využití řemenového převodu tak, aby mohl samotný motor být umístěn blíže k druhé ose.

Jelikož již známe předběžný návrh rozložení motorů, můžeme začít s návrhem samotných převodových mechanismů pro krokové motory. Před samotnou volbou těchto mechanismů si vzhledem k našim požadavkům na celé rameno řekneme, zda není možné navrhnout tyto převodové mechanismy stejné. Díky tomu by celá konstrukce ramene byla z méně různých částí, čímž by došlo ke zjednodušení prototypování, výroby a následných úprav. Pokud bychom předpokládali návrh rozložení motorů popsany výše, bylo by nutné převodový mechanismus dimenzovat podle druhého kloubu, ale poté by první a třetí kloub, který má ještě předřazený řemenový převod, byly zbytečně předdimenzovány. Došlo by tedy ke zbytečnému zvýšení hmotnosti a neoptimálnímu zpřevodování. Pokud bychom navrhovali shodný převodový mechanismus pouze pro první a třetí kloub, je možné dimenzovat tuto převodovku dle prvního kloubu, jelikož požadovaný moment na třetím kloubu získáme správně zvoleným převodovým poměrem na předřazeném řemenovém převodu. Aby bylo možné stejné úpravy využít v případě druhého kloubu, předřadíme také řemenový převod. Tedy řemenový převod u druhého kloubu není umístěn primárně z důvodu lepšího rozložení hmotnosti, ale pro snazší optimalizaci návrhu jednotlivých převodových mechanismů.

### 3.2.8 Návrh prvního prototypu cykloidní převodovky

Jelikož navrhujeme převodový mechanismus pro všechny tři konstrukce a předpokládáme, že vlivem nepřesností 3D tisku bude docházet k velkým ztrátám, potřebný převodový poměr reduktoru se nebudeme snažit přesně vypočítat, ale spíše odhadnout. Vyjdeme ze znalosti hmotností motorů, doporučeného maximálního převodového poměru pro řemenový převod a minimálního zatížení, které by rameno mělo unést. Jelikož předpokládáme vývoj více prototypů, nebude problém další prototypy upravit dle zjištěných kroutících momentů na výstupu navržené převodovky. Převodový poměr tedy volíme pro první prototyp 12.

Z kapitoly 3.2.6 známe možné konstrukce, které bychom pro převodový mechanismus mohli využít. Jako první a nejlepší možnost se jeví využít harmonickou převodovku, která poskytuje dostatečný převodový poměr a zároveň při správné konstrukci by měla téměř nulové vůle. Ovšem po průzkumu debat a videí dostupných na internetu a zároveň na základě zkušeností konstruktérů, kteří se již pokoušeli sestojit harmonickou převodovku tištěnou na 3D tiskárně, docházíme k závěru, že tato kombinace není příliš vhodná. Problematický se po čase provozu stává flexibilní prvek, který je složité vytisknout v dostatečné kvalitě a přesnosti, aby vše správně dosedalo. Další a také jednodušší se jeví variantou je využití planetové převodovky. Tato převodovka by již šla nejspíše úspěšně využít, ale při zkrácení prvních návrhů od této možnosti také upouštíme. Důvodem se stává převážně rozměr této převodovky. Na základě předešlých zkušeností nenavrhujeme modul ozubených



**Obrázek 3.11:** Řez částí prvního prototypu cykloidní převodovky

kol příliš malý, tak aby nedocházelo k problémům způsobeným technologií 3D tisku. Díky tomuto omezení vychází planetová převodovka s převodovým poměrem 12 příliš velká a využití dvoustupňové planetové převodovky by mohlo představovat problém v přesnosti. Tedy pro první detailní návrh volíme cykloidní převodovku.

Zvolení cykloidní převodovky bylo z velké části i ovlivněno osobní snahou pochopit tento mechanismus, jelikož s planetovými převodovými mechanismy se můžeme relativně často setkat i v běžném životě.

Průmyslové cykloidní převodovky většinou najdeme navrhované jako více stupňové, kdy mají 2 a více excentrických hřídelí. Výstupem se poté stává středová excentrická hřídel. Jelikož nepředpokládáme absolutní tuhost a přesnost, budeme využívat středovou excentrickou hřídel jako vstup a výstupem se stane vnitřní klec. Tato vnitřní klec by mohla být tvořena excentrickými hřídeli, ale jelikož výstupem je rotace celé klece kolem osy převodovky a ne rotace jednotlivých částí vnitřní klece, budou využity pouze kruhové tyče. Tato vnitřní klec bude spojena s výstupem převodovky, v našem případě přímo s ramenem. Na obrázku 3.9 vidíme schéma rozložení jednotlivých prvků. V cykloidní převodovce se pro zajištění správného chodu musí nacházet minimálně dva vačkové disky, které jsou vůči ose převodovky otočeny o  $180^\circ$ . Pro plynulý chod je nutné mezi excentrickou hřídel a vačkové disky vložit ložiska. Ty budou do disků pouze zalisovány. Jelikož jsou disky vůči sobě excentrické, jsou excentrické vůči sobě na hřídeli i ložiska. Aby byla umožněna smontovatelnost, je nutné excentrickou hřídel navrhnout z více částí, které jsou spolu sešroubovány. Korunové kolo neboli vnější klec v prvním prototypu vytiskneme přímo s konstrukcí stacionární části. Pro zajištění správné polohy

Tiskárna	Prusa i3 mk3s					
Materiál	Teplota první vrstvy	Teplota tisku	Teplota podložky	Výška vrstvy (detaily)	Výplň	Součásti
PLA	215 °C	210 °C	70 °C	0,2 mm (0,07 mm)	Dle součástí 20 %-80 %	Téměř vše
PETG	235 °C	230 °C	70 °C	0,2 mm (0,07 mm)	80 %	Více namáhané (hřídele, řemenice)

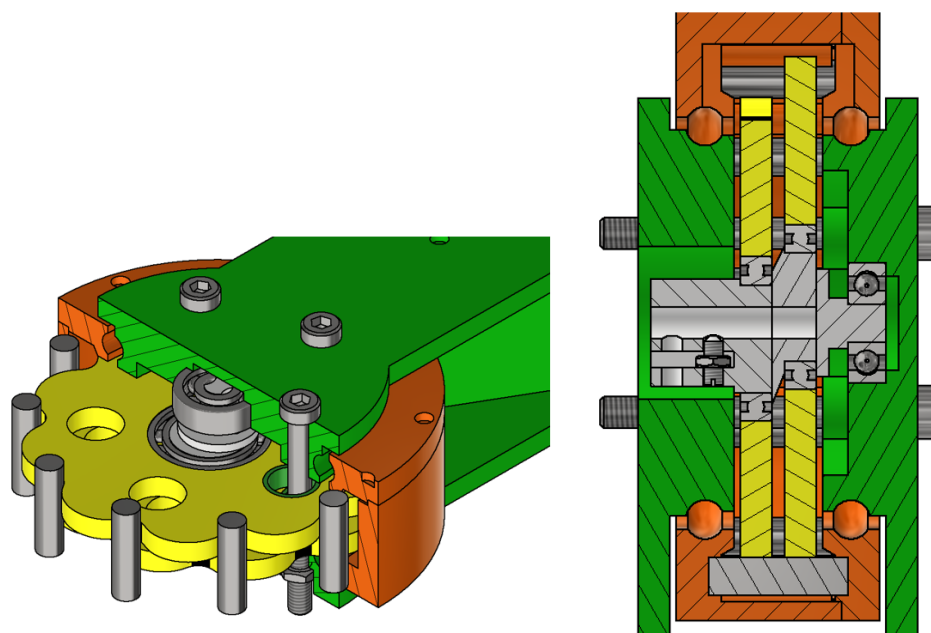
Obrázek 3.12: Orientační parametry 3D tisku

stacionární části vůči výstupu, tedy ramenu s vnitřní klecí, je nutné využít ložiska. Z konstrukčních důvodů musí tyto ložiska mít relativně velký průměr, proto se v tomto prototypu pokusíme tyto ložiska částečně vytisknout. Na obrázku 3.11 můžeme vidět řez modelem tohoto prototypu převodovky.

Tento prototyp byl jako první i reálně vytisknut. Požitá tiskárna a materiály jsou vidět v tabulce 3.12. Po začátek výtisku byly již při kompletaci tohoto prototypu zřejmé problémy s velkým třením mezi vnější klecí a vačkovými kotouči. Díky nepřesnosti tisku docházelo k nesprávnému odvalování a ztráty byly natolik patrné, že jsme ani nepostoupili k testování s osazeným motorem a pokračovali jsme návrhem upraveného prototypu.

Úprava spočívala převážně v nahrazení vnější klece ocelovými tyčemi. Ideálním řešením by bylo umístit každou z těchto tyčí do ložisek, ale pro úsporu váhy budou tyto tyče pevně umístěny v konstrukci ramene. Aby se tyto tyče bez větších komplikací vešly do navržené konstrukce, změnili jsme i převodový poměr na 10. Dále pro zlepšení přesnosti a zamezení rychlému opotřebování se nebude vnitřní klec přímo dotýkat tištěného vačkového disku, nýbrž vnitřní klec bude vložena do pouzder, které jsou nalisovány do vačkového disku. Tuto upravenou konstrukci můžeme vidět na obrázku 3.13.

Po vytisknutí a začátek byl tento prototyp cykloidní převodovky složen. Jelikož se již zdál dostatečně vhodný, nastal čas testu s osazeným motorem. Byl využit motor Nema17 42byghw811, viz tabulka 3.1. Při tomto testování jsme se zaměřili převážně na měření kroutícího momentu na výstupu převodovky. Tento kroutící moment jsme orientačně zjišťovali pomocí 50 cm hliníkového profilu, který se dá připevnit na konstrukci převodovky. Poté přidáváním a posouváním závaží jsme postupně zjistili potřebné hodnoty. Maximální moment na výstupu byl přibližně 1,5 Nm, ale při tomto zatížení již občas docházelo ke ztrátě kroků na krokovém motoru. Bezpečný provozovatelný moment byl po opakovaném testování určen jako 1,2 Nm. Teď již máme představu o kroutícím momentu, který zvládne naše převodovka vyvinout, dalším pro nás důležitým parametrem je její hmotnost. Jelikož před některé převodovky bude umístěn i řemenový převod, zajímá nás hmotnost bez samotného motoru. Váha tohoto prototypu převodovky je 400 g a vnější rozměr válcové části převodovky je 95 mm.



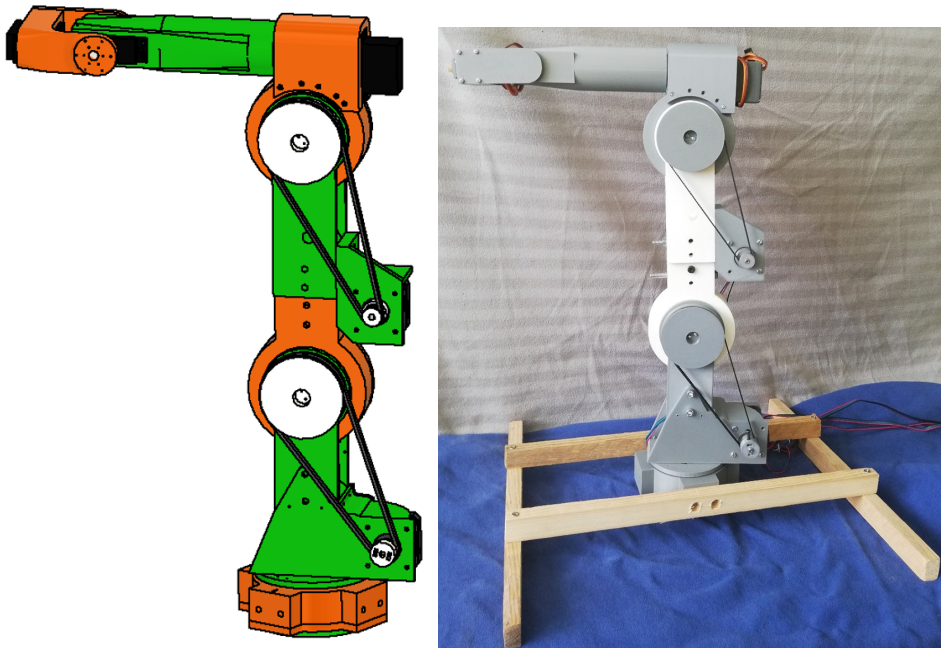
Obrázek 3.13: Řez prototypem cykloidní převodovky

### 3.3 Návrh prvního prototypu ramene

Po drobných úpravách cykloidní převodovky, spočívajících převážně ve zlepšení konstrukce vzhledem k jednodušší smontovatelnosti, můžeme začít s návrhem prvního celého prototypu ramene. Tento prototyp využívá tři cykloidní převodovky, které jsou vnitřním uspořádáním a díly totožné. Liší se pouze ve vnějších částech. Pro pohon druhého kloubu je mezi motor a cykloidní převodovku vložen řemenový převod s převodovým poměrem 5. Maximální teoreticky použitelný kroutící moment ve druhé ose je 6 Nm. Pro třetí kloub jsme využili slabší krokový motor a řemenový převod s poměrem 3. Napínání řemenů je vyřešeno posuvným uložením motorů, čímž můžeme nastavit napnutí řemenů. Koncová část ramene obsahující 4., 5. a 6. osu je navržena tak, aby ji bylo možné jednoduše od těla robota odpojit a nahradit upravenou částí v závislosti na konkrétní aplikaci.

Po vytištění následuje část testování. Sestavený prototyp můžeme vidět na obrázku 3.14. Při testování bylo k řízení využito Arduino Nano, viz 4.1.1. Nejprve testujeme samotnou funkčnost jednotlivých převodových mechanismů. Při prvních pohybech a zatěžování se rozhodneme pro zvýšení převodového poměru řemenového převodu na 3. ose, kdy z převodového poměru 3 přecházíme na 5. Bez zatížení se jeví rameno vcelku stabilní a přesné, ale po větším zatížení při plném vyložení ramene dochází k velkému vychýlení v prvním kloubu. Toto je způsobeno velkým momentem, který působí na ložiska v cykloidní převodovce. Jelikož jsou tyto ložiska částečně tištěna, projeví se nepřesnosti 3D tisku až při větším zatížení. Tedy opakovatelnost a i přesnost ramene je velmi špatná. Pokud se ale zaměříme na maximální nosnost ramene,





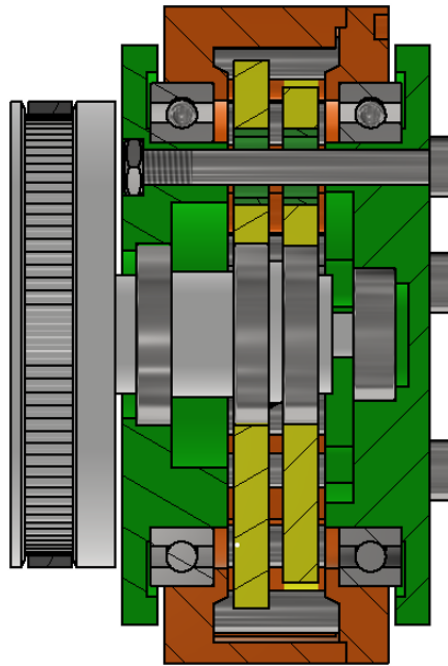
**Obrázek 3.14:** Model a sestavený prototyp ramene

můžeme říci, že jsou požadavky splněny, jelikož rameno zvládá při plném vyložení zvednout závaží o maximální hmotnosti 1,5 kg. Ovšem s tímto zatížením při větších zrychleních dochází ke ztrátě kroků na motorech. Bezpečně zvládá opakovaně zvednout závaží o hmotnosti 1 kg.

Při dlouhodobém testování se začínají více projevovat problémy s tištěnými ložisky, kde je nutné často převodovky rozebírat a zmíněná ložiska očistit a promazat. Tento problém by mohl být vyřešen využitím jiného materiálu, který by se tolik neodíral.

### 3.4 Návrh vylepšeného prototypu

První prototyp byl navrhován jako testovací, od druhého prototypu již očekáváme funkčnost. Proto je nutné konstrukci dle získaných poznatků přepracovat a doplnit o některé části. Pro správné řízení krokových motorů potřebujeme znát jejich absolutní polohu. Tu můžeme zjistit pomocí počáteční inicializace, kdy si řídicí elektronika dále sama pamatuje z počtu kroků, jak je rameno natočené. Jinou možností je osazení některého ze snímačů absolutní polohy. Tato varianta je vhodná především k možné kontrole ztráty kroků na krokových motorech. Protože při testování nedochází ke ztrátě kroků se zatížením, které stále splňuje naše požadavky, rozhodneme se další prototyp navrhnout s možností osazení koncových snímačů. Tyto koncové senzory jsou levnější variantou ve srovnání s absolutními snímači polohy. V průběhu testování předešlého prototypu jsme narazili na dlouhodobou nestálost navržených vnějších ložisek, které byly částečně tištěné. Pro docílení větší robustnosti ramene bude nutné osadit převodovky klasickými ocelovými kuličkovými



**Obrázek 3.15:** Řez cykloidní převodovkou s řemenicí (kloub 3)

ložisky. Z důvodů funkčnosti převodovky je nutné využít ložiska poměrně velká vzhledem k samotné konstrukci převodovky. Po drobných úpravách navrhne využití ložisek 61811 dle normy CSN 02 4630. Konstrukční řešení této úpravy je vidět na obrázku 3.15. Problém, který vzniká s využitím těchto ložisek, je nárůst váhy převodovky. Jelikož váha každého ze 2 ložisek je 90 g a váha původní převodovky byla 400 g, při využití těchto ložisek nám vzroste váha téměř o polovinu. Z tohoto důvodu bude nutné přepracovat převodové poměry převodových mechanismů jednotlivých os ramene. V případě první osy nám vystačí současný převodový poměr. Pro 3. osu taktéž není nutné nic zásadně měnit, jelikož hmotnost, kterou zvedá, se vlivem změny ložisek nezměnila. Problém vzniká v případě 2. osy, kde již nejsme schopni zvýšit kroutící moment zvětšením převodového poměru předřazeného řemenového převodu, jelikož doporučený maximální převodový poměr pro řemenový převod je 5.

V praxi se pro velké převodové poměry, v případě využití planetových převodovek, často využívá víceúrovňových konstrukcí. My se pokusíme podobného efektu docílit s námi navrženou cykloidní převodovkou. V případě pouhého umístění 2 cykloidních převodovek za sebe do série dojde k velkému nárůstu rozměru celé konstrukce převodovky. Ve snaze zmenšit tento vnější rozměr se pokusíme navrhnout kompaktní dvojitou cykloidní převodovku.



### 3.5 Návrh dvojité cykloidní převodovky

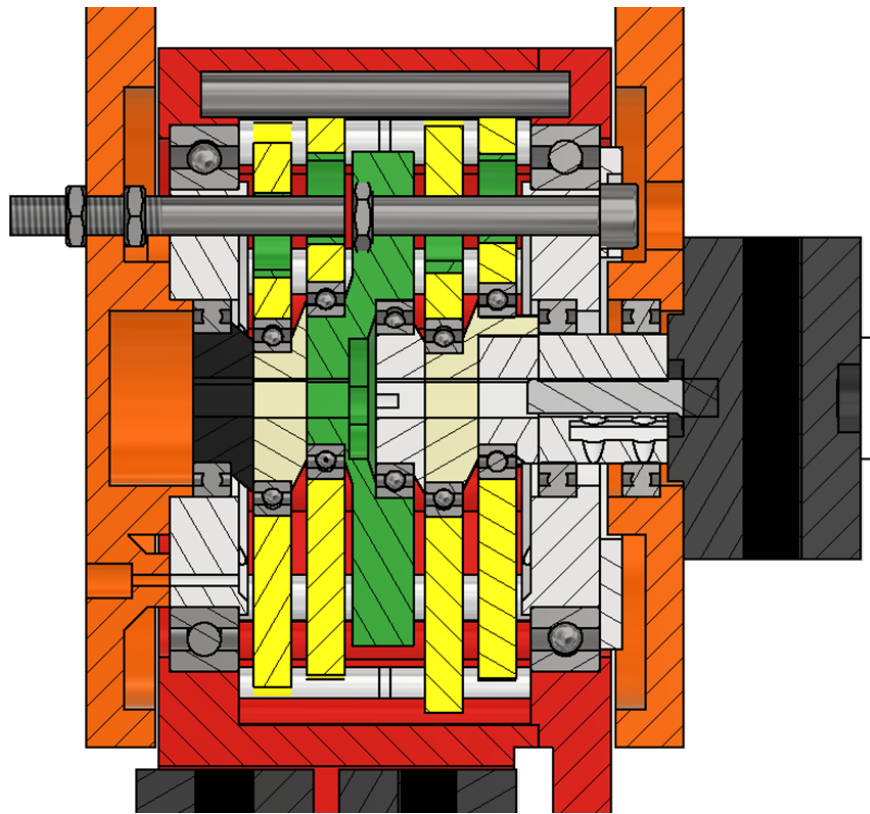
Abychom dodrželi jednoduchost celé konstrukce, budeme klást důraz na využití co nejvíce stejných komponentů jako v případě obyčejných cykloidních převodovek, které již v rameni používáme. Stejně jako v případě planetových převodovek je i u cykloidních převodovek možné využít několik možných kombinací vstupů a výstupů. Například pokud jako vstup uvažujeme excentrickou hřídel umístěnou v ose převodovky, výstupem převodovky může být buď vnější nebo vnitřní klec. Po zkusnění možných kombinací nám jako nejjednodušší varianta vychází zkonstruovat dvojitou cykloidní převodovkou s vnější klecí shodnou pro oba stupně převodu. V tomto případě je převod krouticího momentu z prvního stupně na druhý možný dvěma způsoby. Složitější způsob spočívá v nahrazení vnitřní klece excentrickými hřídeli, které jsou prodlouženy jako mimoosé excentrické hřídele pro 2. stupeň. Tato konstrukce je obdobná průmyslovým řešením, ve kterých je předřazený planetový převod a ne cykloidní. Problémem tohoto řešení je nutnost umístění velkého množství ložisek, která se při našich malých rozměrech do daného prostoru těžko vejdu. Konstrukčně jednodušší řešení je využití vnitřní klece prvního stupně, jako vstupu na excentrickou hřídel, uloženou v ose převodovky druhého stupně. Na obrázku 3.16 a 3.17 můžeme vidět řez finální konstrukcí této dvojité cykloidní převodovky. Z tohoto řezu je patrné, že pro tuto převodovku je motor uložen v její ose.

Pro zvýšení tuhosti osových excentrických hřídelí je do jejich středu vlisována čtyřhranná tyč. Dále si můžeme na tomto řezu všimnout, že vnitřní klec 2. stupně je uložena pouze na vnější straně. Jelikož je toto vnější uložení dostatečně široké, není nutné do vnitřku převodovky umísťovat disk, který by zajišťoval i vnitřní uložení tyčí vnitřní klece. Nejen díky této úpravě má finální návrh samotné převodovky v axiálním směru rozměr pouze 8 cm (bez motoru). Převodový poměr této dvojité cykloidní převodovky vypočítáme dle vzorce 3.4. Při výpočtu uvažujeme vnější klec jako stacionární část. Tedy motor se otáčí současně s výstupem převodovky. Z tohoto důvodu není převodový poměr 100 ale pouze 99. Značnou komplikací při návrhu této převodovky bylo množství komponent a složitost jejich zkompletování. Z tohoto důvodu bylo nutné některé komponenty rozdělit na více částí, což má ve finální konstrukci špatný vliv na rozměrovou přesnost, jelikož dochází ke sčítání nepřesností způsobených technologií 3D tisku.

$$i_r = i_1 \cdot i_2 = \left(\frac{n}{N-n}\right)^2 = \left(\frac{10}{11-10}\right)^2 = 100 \quad (3.2)$$

$$i_r = \frac{\omega_m + \omega_{out}}{\omega_{out}} = \frac{\omega_m}{\omega_{out}} + 1 \quad (3.3)$$

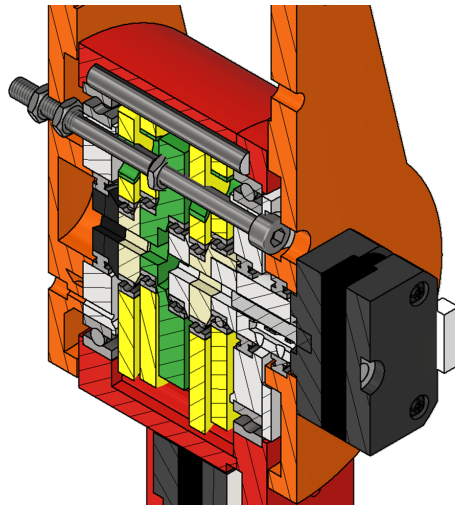
$$i_m = \frac{\omega_m}{\omega_{out}} = i_r - 1 = 100 - 1 = 99 \quad (3.4)$$



Obrázek 3.16: Řez dvojitou cykloidní převodovkou

### 3.5.1 Testování nového prototypu ramene

Již při kompletaci převodovky byly zřejmé problémy s přesností 3D tisku, kde některé komponenty bylo nutné vytisknout znovu s menšími rozměry tak, aby bylo vůbec možné převodovku zkompletovat. Těmto problémům by bylo možné se částečně vyhnout nastavením lepších parametrů při 3D tisku. Po celkovém složení této převodovky i s motorem se při prvních testech zdála být plně funkční. Ovšem při zatížení byly velmi patrné ztráty v převodovce. Pro možný odhad účinnosti této převodovky, byla vyrobena a zkompletována obyčejná cykloidní převodovka s osazenými ocelovými kuličkovými ložisky. Při testování tato převodovka dosahovala přibližné účinnosti 80 %. Původní konstrukce bez ocelových ložisek dosahovala účinnosti maximálně 75 %. Pokud bychom uvažovali pro oba stupně dvojitě cykloidní převodovky shodnou účinnost, tedy 80 %, celková účinnost obou stupňů by měla být 64 %. Tedy na výstupu převodovky by kroutící moment měl dosahovat přibližně 63násobek kroutícího momentu motoru. Pro řemenový převod předpokládáme nejmenší předpokládanou účinnost 90 %.



Obrázek 3.17: Boční pohled na řez dvojitou cykloidní převodovkou

## ■ 3.6 Testování prototypů

### ■ 3.6.1 Testování nových převodovek

Po prvotních testech dvojitě cykloidní převodovky a obyčejné cykloidní převodovky s osazenými ložisky došlo k vytištění ostatních komponent a sestavení nového prototypu ramene. Sestavené rameno je spolu s modelem vidět na obrázku 3.18. Při testování tohoto prototypu bylo vylepšení velmi patrné hlavně v 1. ose ramene, kde oproti 1. prototypu nedochází při zatížení k velkým výkyvům v ložiscích. Z důvodů zvýšení váhy cykloidních převodovek a nízké účinnosti dvojitě cykloidní převodovky je rameno při plném vyložení schopné bezpečně unést pouze 0,8 kg. Po dlouhodobém testování dvojitě cykloidní převodovky nedošlo k nijak výraznému snížení účinnosti, ovšem velmi razantně se začaly projevovat vůle v převodovce. Oproti tomu obyčejné cykloidní převodovky si svoji přesnost zachovaly i po dlouhodobém testování. Pro vylepšení přesnosti dvojitě cykloidní převodovky by bylo vhodné tyče vnější klece umístit do ložisek a zvětšení celkových rozměrů excentrických hřídelí tak, aby nedocházelo k jejich zkrutu. Jinou možnou variantou je navrhnout cykloidní převodovky s větším převodovým poměrem a předsazením řemenového převodu. Ovšem těmito dalšími vylepšeními se již tato práce dále nezabývá.



**Obrázek 3.18:** Prototyp ramene s dvojitou cykloidní převodovkou

## Kapitola 4

### Hardware

Jelikož nemáme specifikovanou konkrétní aplikaci, ve které by mělo být rameno využito, musíme i při návrhu hardwaru a hlavně softwaru, klást důraz na jednoduché přizpůsobení různým požadavkům. Vzhledem k předpokladu možného napojení na ROS (viz 5.3) je nutné, aby námi navržený hardware a software toto propojení umožňoval. Požadavky na hardware tedy můžeme rozdělit na 2 základní a to schopnost zvládat komunikaci s řízením, na kterém běží ROS, a schopnost bezpečně ovládat celé rameno jako celek. Od softwaru očekáváme flexibilitu, díky které nebude problém upravit software pro různé aplikace. Dále by měl zajistit základní bezpečnost a schopnost ovládat rameno i bez podpůrného softwaru pro plánování pohybu jako je ROS.

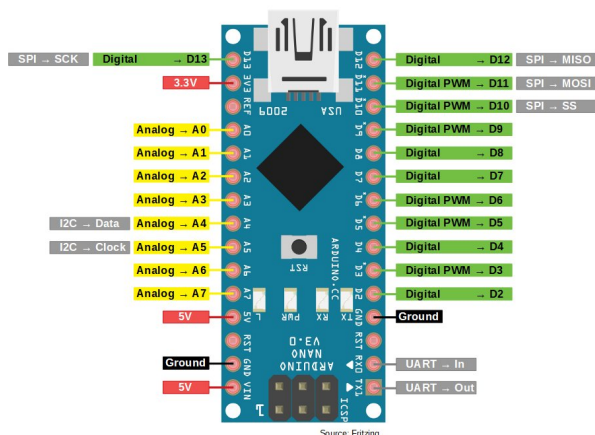
### 4.1 Volba mikrokontroleru

#### 4.1.1 Arduino nano

Jelikož se snažíme cílit na dostupné a cenově nenákladné komponenty, v prvních prototypch pro řízení volíme mikrokontroler Arduino nano, viz obrázek 4.1. Tento mikrokontroler je open source deska založená na chipu ATmega 328P, kterou vyvinula společnost Arduino. Deska má 22 IOPinů, z toho 6 lze využít pro generování PWM signálu, který potřebujeme pro řízení servomotorů.

Technické specifikace pro Arduino nano: [36]

- Chip: ATmega328
- Operační napětí: 5 V
- Paměť flash: 32 kB
- Frekvence: 16 MHz
- Analogové vstupní piny: 8
- Maximální proud na pinu: 40 mA (I/O Pins)
- Vstupní napětí: 7-12 V



Obrázek 4.1: Rozložení pinů na Arduino Nano [37]

- Digitalní piny: 22 (6 of which are PWM)
- PWM výstup: 6
- Spotřeba: 19 mA
- PCB velikost: 18 x 45 mm

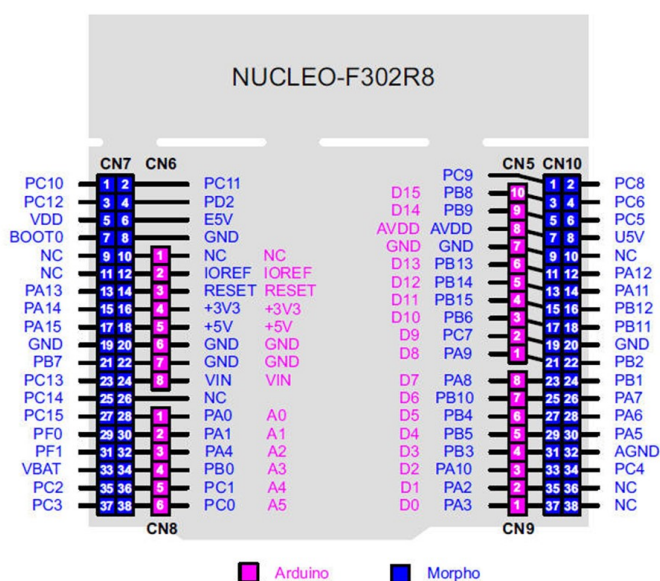
Velkou výhodou tohoto mikrokontroleru je možnost připojení pomocí USB, které je následně možné použít pro sériovou komunikaci například se zařízením, na kterém je spuštěn ROS. Programován může být pomocí vývojového prostředí Arduino Software IDE.

#### ■ 4.1.2 STM32

Po prvotních testech s mikrokontrolerem Arduino nano jsme začali narážet na jeho limity. Proto jsme pro možné zlepšení řízení a zajištění dostatečného výkonu pro možné aplikace, nahradili v dalších prototypch Arduino vývojovou deskou STM32 F302R8 NUCLEO-64. Tento mikrokontroler obsahuje výkonné 32-bitové jádro RISC ARM Corex-M4, které je schopné pracovat až s frekvencí 72 MHz. Pro nás důležitou součástí je množství časovačů, kterými tato deska disponuje. Tři univerzální 16bitové časovače a 12 PWM časovačů nám zajistí pohodlné řízení všech motorů. Pro komunikaci můžeme využít I2C, SPI, USART a CAN. Výhodou vývojové desky je i integrovaný programátor, respektive debugger ST-link V2. Pro nahrávání programu tedy nepotřebujeme další hardware, jak by tomu bylo při využití samotného čipu. V případě potřeby je možné tento programátor od desky oddělit jednoduchým odlomením. Poté je možné ho využívat externě. [38]

Základní parametry pro desku STM32 F302R8 NUCLEO-64:

- Chip: 32-bitové jádro RISC ARM Corex-M4

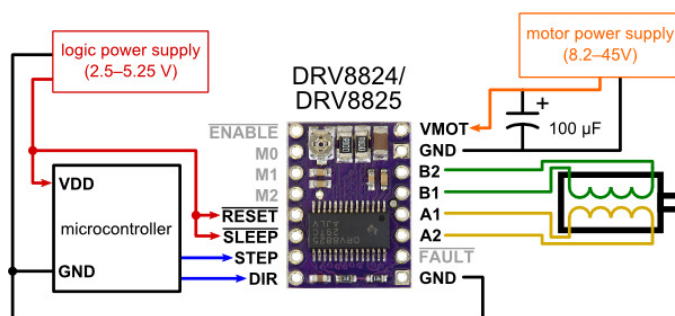


Obrázek 4.2: Rozložení pinů ST Morpho a Arduino [39]

- Operační napětí: 3,3 V
- Paměť flash: 64 kB
- Frekvence: 72 MHz
- Převodníky ADC: 1x12bit 0-3,6 V 15 kanálů
- Převodníky DAC: 1x12bit 2,4-3,6 V 1 kanál
- Možné napájení: Podle užitého způsobu 3,3 V, 5 V, 7-12 V
- IOPinů: 64 Morpho(část sdílána s Arduino konektory)
- PWM výstupů: 12
- Komunikace: 3xI2C,3xUSART,2xSPI,1xCAN,1xUSB2.0
- PCB velikost: 82,5 x 70 mm

Pro nás velkou výhodou je osazení nejen pinovými lištami ST Morpho, ale také konektory shodnými pro Arduino Uno platformu, jak můžeme vidět na obrázku 4.2. Díky tomu je možné snadno využít rozšiřující moduly navržené pro Arduino. Tato vývojová deska může být programována pomocí vývojového prostředí STM32CubeIDE. Toto prostředí nám ulehčí nastavení parametrů časovačů, pinů i komunikace. Také je zde mnoho různých ukázek a návodů, které často mohou pomoci při řešení některých problémů.





Obrázek 4.3: DRV8825 [40]

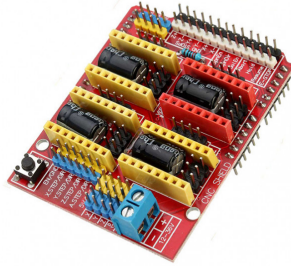
## 4.2 Volba driveru

Servomotory jsme schopni ovládat přímo pomocí PWM signálu z mikrokontroleru, ale pro řízení krokových motorů je vhodnější využít některý z dostupných driverů. Pro řízení tohoto ramene jsme volili mezi drivery A4988 a DRV8825. DRV8825 (viz obrázek 4.3) je driver pro řízení bipolárních krokových motorů. Skládá se ze dvou H-můstek, které jsou tvořeny tranzistory typu MOSFET. Jeho výhodou je podpora mikrokrokování s rozlišeními: plný krok, 1/2, 1/4, 1/8, 1/16 a 1/32. Díky tomu nám umožňuje velmi precizní řízení motorů s možností volby vhodného rozlišení v závislosti na požadované přesnosti kroku. Tento driver zvládá pracovat s napětím od 8 V do 40 V. Maximální možný proud, který je bezpečný pro chod, je 1,5 A. V případě zajištění dodatečného chlazení je možné pracovat s proudem až do 2,2 A. Pro ovládání je nutné použít napětí v rozmezí od 2,5 V do 5,25 V. S tímto driverem se můžeme často setkat v 3D tiskárnách, domácích CNC strojích a obecně v malých robotech, proto je pro nás ideálním řešením. [40]

Pokud se dále zaměříme na parametry A4988, zjistíme, že oproti DRV8825 podporuje mikrokrokování pouze do 1/16 kroku. Maximální doporučené napětí je pouze 30 V a maximální proud při chlazení je taktéž menší a to 2 A. Tedy tento driver dosahuje menších výkonů. Z testů také vyplývá jeho větší zahřívání. Počet pinů mají oba drivery stejný, viz obrázek 4.3, ale v určení některých pinů se mírně liší. Příkladem může být pin pro přivedení napětí pro logický obvod v případě A4988.

Pomocí pinu STEP ovládáme impulzy krok motoru. Pin DIR slouží pro nastavení směru rotace motoru. Pokud bychom využívali pouze rotaci v jednom směru, je možné tento pin pouze propojit na VCC nebo GND. Tyto drivery mají několik ochranných prvků. Mezi ty nejdůležitější patří tepelná a proudová ochrana. [41]





**Obrázek 4.4:** CNC shield V3.0

Pro bezpečný chod je nutné na obou zmíněných driverech nastavit proud dodávaný do jednotlivých motorů. Toto nastavení se provádí nastavením napětí pomocí potenciometru umístěném na každém driveru. Pro případ driveru A4988 platí vztah:

$$I_{MAX} = \frac{V_{REF}}{8R_{CS}} \quad (4.1)$$

Kde  $V_{REF}$  je námi nastavované napětí,  $I_{MAX}$  představuje maximální hodnotu jmenovitého proudu krokového motoru a  $R_{CS}$  je odpor driveru. Pro nastavení  $V_{REF}$  v případě DRV8825 platí tento vztah:

$$I_{MAX} = 2V_{REF} \quad (4.2)$$

V případě řízení pomocí Arduino nano bylo využito nepájivé pole, na kterém byly propojeny potřebné piny driverů s mikrokontrolerem. Po té co jsme změnili řídicí jednotku na Nucelo-64, bylo již možné využít jednodušší řešení. Jelikož tato deska sdílí rozložení pinů s Arduino standardem, můžeme využít rozšiřující modul určený přímo pro osazení driverů pro krokové motory. Zvolili jsme velmi rozšířený modul Arduino CNC Shield V3.00 vyvinutý společností Protoneer. Tento modul můžeme vidět na obrázku 4.4. [42]

Výhodou tohoto modulu je kompaktní řešení celého zapojení driverů. Na modulu je možné pomocí propojení pinů nastavit mikrokrokování jednotlivých driverů, deska obsahuje vlastní svorkovnici pro přivedení napájení motorů a obsahuje přípravu pro připojení například koncových snímačů. Tyto piny pro připojení spínačů a senzorů jsou pouze vyvedené piny z pinů Arduino, do kterých je deska osazena. Tedy oproti lepší přehlednosti nepřináší žádnou výhodu. Tento modul slouží pro řízení až 4 motorů. Zajímavou možností je, možnost určit si, zda 4. motor bude kopírovat pohyby některého z prvních třech motorů nebo zda bude řízen zvlášť. V našem případě bude rameno osazeno pouze třemi motory, tedy tuto funkci nebudeme potřebovat. Modul obsahuje i tlačítko pro resetování Arduino. Díky shodným pinům je tato funkce tlačítka zachována i v případě využití na Nucelo-64. Tato maličkost se ukázala v průběhu testování velmi užitečná, jelikož tlačítko na vývojové desce pro resetování mikrokontroleru je umístěno pod samotným CNC modulem, tedy bez rozpojení je nedostupné.



**Obrázek 4.5:** Napájecí deska XH-M229 [43]

### 4.3 Koncové senzory

Jako koncové senzory využijeme mikrospínač připravený pro třívodičové zapojení. Tedy má jak spínací, tak rozpínací kontakt. Pro větší bezpečnost by bylo vhodné využívat oba kontakty a porovnávat jejich hodnotu. Jelikož v našem případě budou koncové snímače sloužit hlavně pro inicializaci polohy motorů, předpokládáme v tento okamžik přítomnost obsluhy, která by případný problém zjistila a rameno by zastavila. Proto zapojíme koncové senzory pouze ve spínacím režimu, tedy při nesepnutém spínači budou kontakty rozepnuty.

### 4.4 Zdroj

Pro napájení celého ramene je nutné využít vhodný zdroj. Pro testování byl vhodný laboratorní zdroj, ale pro samotné využití ramene v konkrétních aplikacích je nutné využít některý z dostupných zdrojů, který nám poskytne požadované napětí pro napájení celého systému. Pro provoz servomotoru budeme hledat zdroj o napětí 5 V a pro krokové motory si vystačíme i s 12 V. Řídící mikrokontroler můžeme napájet pomocí USB, jelikož předpokládáme stálé připojení k počítači. Vzhledem k těmto požadavkům a ke skutečnosti, že celý projekt se snažíme navrhovat jako velmi cenově dostupný, rozhodneme se pro napájení využít starý ATX zdroj. Tyto zdroje poskytují 3,3 V, 5 V a 12 V. Pro propojení využijeme molex konektor, díky kterému je možné zdroj v budoucnu jednoduše nahradit. Pokud využíváme neupravený počítačový ATX zdroj, je nutné pomocí dalšího hardwaru zajistit jeho spínání. K tomuto účelu můžeme použít například napájecí desku XH-M229, viz obrázek 4.5. Z této konkrétní desky je možné přímo ze svorek odvádět požadované napětí, ale jak jsme již zmínili, my využijeme pro propojení přímo molex konektor.

## Kapitola 5

### Software

#### 5.1 První řízení

Po vytvoření prvních prototypů převodových mechanismů bylo nutné zajistit ovládání motorů pro jejich správné otestování. Jak již bylo zmíněno v kapitole 4.1.1, nejprve bylo pro řízení zvoleno Arduino. Pro psaní testovacích programů bylo zvoleno vývojového prostředí Arduino Software IDE. U servomotorů jsme od prvních návrhů nepředpokládali žádné námi dodělané převodovky, protože v počátcích testování pro nás nebyla schopnost jejich řízení tolik důležitá. Oproti tomu pro otestování funkčnosti navržených převodových mechanismů jsme nutně potřebovali mít schopnost řídit krokové motory.

Právě k tomuto účelu slouží například knihovna AccelStepper. Tato knihovna je psána přímo pro Arduino, je proto velmi snadné ji přímo z vývojového prostředí stáhnout a implementovat. Tato knihovna nám poskytuje mnoho funkcí, mezi velké výhody patří podpora akcelerace. Díky tomu jsme schopni plynule zvyšovat otáčky a sledovat případné problémy v prototypch. Další pro nás zajímavou možností je podpora řízení více motorů současně, bez čehož se v našem případě neobejdeme. Tuto možnost nakonec nevyužijeme, jelikož tuto platformu využijeme pouze pro testování prvních prototypů samotných převodovek, ale pro řízení celého ramene již budeme využívat STM32. Autoři této knihovny si pro další uživatele připravili několik ukázkových kódů pro snazší pochopení celé knihovny. Pro řízení krokových motorů, s pomocí driveru DRV8825, můžeme přímo využít právě jeden z těchto ukázkových kódů.

V průběhu testování si k vývojové desce připojujeme další pomocné spínače a LED kontrolky pro možnost zastavování motorů a podobně. V tomto okamžiku začínáme přemýšlet nad volbou jiné vývojové desky, jelikož v případě rozšíření ramene o další funkčnosti by nemusela deska obsahovat dostatek pinů pro připojení všech komponent. Později jsme zjistili, že nejsme schopni s tímto typem Arduina generovat signál s dostatečně velkou frekvencí. Pro PWM signál se v případě Arduino Nano uvádí maximální možná frekvence 62 kHz. Pokud bychom nechtěli využívat tuto desku pro žádný jiný účel, je možné dosáhnout i na vyšší frekvence. Ale jelikož naším požadavkem na mikrokontroler je řídit minimálně 6 motorů a k tomu zajišťovat komunikaci, je nutné přejít k výkonnější verzi Arduina nebo zvolit jiný typ mikrokontroleru.

Z kapitoly 4.1.2 již víme, že jsme zvolili vývojovou desku STM32 F302R8 NUCLEO-64.

## ■ 5.2 Řízení STM32

Tato platforma pro nás představuje výkonově velký skok oproti dosavadnímu řešení. Pro programování využijeme vývojové prostředí STM32CubeIDE, jelikož z předešlých projektů s ním již máme zkušenosti.

### ■ 5.2.1 STM32CubeIDE

STM32CubeIDE je výkonný multiplatformní nástroj, který je součástí softwarového ekosystému STM32Cube. Toto prostředí neslouží pouze pro psaní a vývoj kódu v jazyce C/C++, ale umožňuje nám také konfiguraci periférií, generování samotného kódu, kompilaci a ladění mikrokontrolerů a procesorů STM32. STmicroelectronic založila toto prostředí na Eclipse, což je univerzální platforma určená obecně pro programování. Díky tomu je možné využít spoustu doplnků usnadňujících práci, které existují právě pro Eclipse. STM32CubeIDE můžeme rozdělit na tři základní části podle jejich funkčnosti. Mezi jednotlivými částmi jde v prostředí přepínat. Dle zvoleného prostředí se mění celé zobrazení programu.

První část, která nahrazuje práci s STM32CubeMX, slouží pro generování projektu a knihoven náležitých právě k našemu mikrokontroleru. V této části lze jednoduše nastavovat funkce jednotlivých pinů, konfigurovat komunikaci nebo například nastavovat časovače. Po ukončení konfigurace dojde k vygenerování zdrojových souborů. Pokud se v průběhu práce na projektu rozhodneme změnit některou část konfigurace, lze se znovu přepnout do tohoto konfiguračního zobrazení, provést úpravy a nechat znovu vygenerovat zdrojové soubory. Do projektu se nám při konfiguraci přidají mimo jiné i HAL knihovny. Ty slouží pro jednodušší práci s hardwarem. Pokud bychom chtěli ovládat samotný hardware sami, je možné využít LL knihovny, které slouží k přístupu k jednotlivým registrům.

Další částí je samotné prostředí pro vývoj a psaní kódu. Zde najdeme vše, co bychom od takového vývojového prostředí očekávali. Poslední část je určená k ladění programu, kde můžeme číst aktuální hodnoty dat přímo z programu běžícím na mikrokontroleru. Tento nástroj nám velmi ulehčí práci nejen při hledání chyb.

### ■ 5.2.2 Psaní programu v C++ v prostředí STM32CubeIDE

Na první pohled je vývojové prostředí plně připraveno pro podporu jazyka C++, ale několik drobností dokáže navodit pocit u začínajících uživatelů, že je možné kód psát pouze v C. Hned při první konfiguraci je možné nastavit využívání C++ v projektu. Bez tohoto nebudeme schopni vůbec C++ kód kompilovat. Pokud máme projekt vytvořený jako C projekt, je možné i

následně provést překonfigurování a to buď automaticky, nebo ručním přenastavením parametrů kompilátoru. Automatické překonfigurování nemusí vždy správně fungovat, a je proto nutné se přesvědčit, že jsou správně zadané cesty ke zdrojovým souborům.

V obou těchto případech se nám po konfiguraci vygenerují zdrojové soubory spolu se souborem „main.c“. Je patrné, že tento soubor nemá koncovku .cpp, tedy je psaný v C. Nabízí se více možností řešení. První možností je pouze přepsat koncovku. Projekt je na C++ připraven, tedy neměl by nastat žádný problém. Tato možnost zdárně funguje do okamžiku, kdy potřebujeme učinit změnu v konfiguraci vývojové desky. Po uložení změny dojde k přepsání zdrojových souborů. V případě souboru „main.c“ dojde k novému vygenerování celého souboru až na části, které jsou označeny pomocí poznámek `/* USER CODE BEGIN */` a `/* USER CODE END */`. Tedy o kód, který napíšeme mimo tyto poznámky, můžeme velmi snadno přijít. K tomuto chytrému přepsání ovšem nedojde v případě, kdy máme u souboru změněnou koncovku na .cpp. V takovémto případě se vygeneruje nový soubor a je nutné veškeré části kódu přepsat zpět. Tento problém jde obejít přepsáním koncovky zpět na .c vždy, když budeme provádět změnu v konfiguraci. Tato varianta je vhodná, pokud často nedochází ke změnám, které by nás nutily stále přepisovat soubor.

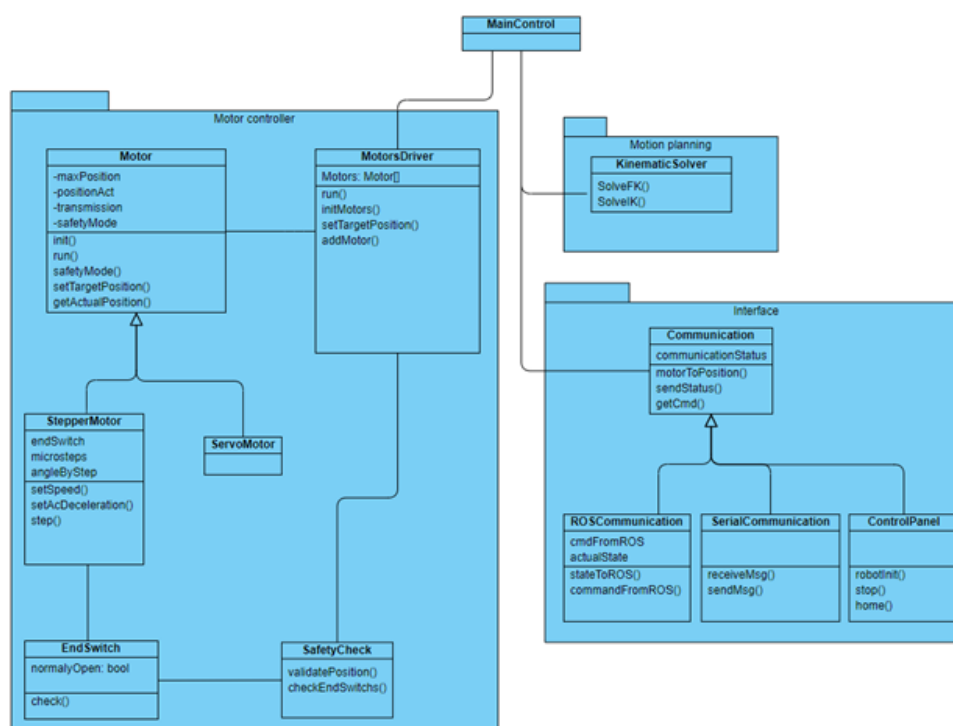
Druhou a snazší možností je vytvoření dalšího souboru například „mainpp.cpp“, ve kterém budeme mít funkci `cppMain`. V původním hlavičkovém souboru stačí pouze přidat tuto novou funkci a v našem novém souboru načíst právě tento hlavičkový soubor, jelikož hlavičkový soubor může obsahovat deklarace funkcí, které jsou definovány v rozdílných souborech. Poté v původní funkci `main` zavoláme naši novou funkci připravenou na využití C++. Díky tomuto obejití problému získáme v nové funkci `CppMain` přístup například ke všem strukturám, které nakonfiguroval `STM32CubeMX`. Tato část je zde zmíněna hlavně jako návod pro začátečníky s tímto prostředím, kteří by mohli narazit na tento problém a toto triviální řešení by jim mohlo ušetřit spoustu práce.

### 5.2.3 Návrhový class diagram

Vzhledem k očekávané složitější koncepci celého programu je dobré si před samotným psaním kódu rozvrhnout strukturu a promyslet principiální fungování. Jako první je potřeba zmínit, že řízení budeme psát v C++, tedy můžeme využívat principů objektově orientovaného programování, dále OOP, což by bylo například při využití Arduina mírně komplikované. Díky myšlence OOP si můžeme práci vcelku snadno rozdělit na několik tříd. Tyto třídy si seskupíme do tří základních skupin. Tyto skupiny se zaměřují na:

- řízení motorů,
- komunikaci,
- plánování pohybu.

Jednotlivé návrhy tříd pro každou skupinu můžeme vidět na obrázku 5.1. Tento class diagram nepopisuje finální schéma programu, ale představuje pro



Obrázek 5.1: Návrhový class diagram

nás určitou osnovu, dle které jsme začali tvořit program. Postupným vývojem tedy přidáváme další třídy, metody a proměnné. Dále se budeme zabývat postupně jednotlivými skupinami, tak jak byly vytvářeny.

#### 5.2.4 Motor controller

Pro řízení krokových motorů pomocí STM32 existuje částečně přepsaná knihovna AccelStepper, kterou jsme využívali v Arduino. Bohužel, jak již název napovídá, tato knihovna je určena hlavně pro řízení krokových motorů a jejich skupin. V tomto okamžiku si pokládáme otázku, zda neexistuje univerzální knihovna pro řízení více druhů motorů. Bez takovéto knihovny bychom ovládání krokových motorů řešili pomocí jedné knihovny a řízení servomotorů jiným způsobem. A jelikož nám při využití přepsané knihovny AccelStepper dochází k problémům, rozhodneme se vytvořit si vlastní třídy pro řízení, které se budeme snažit navrhovat co nejuniverzálněji. Této univerzálnosti docílíme například využitím dědičnosti tříd. Třída Motor nám bude představovat motor obecně s tím, že obsahuje funkce, které bychom obecně od motoru mohli očekávat. Jsou to například inicializační funkce, funkce pro natočení do určité polohy nebo nastavení určité rychlosti. Většinu z těchto funkcí budou dědit jednotlivé třídy potomků a následně si je upravovat pro své potřeby. Jak je vidět na obrázku 5.1, díky tomuto rozdělení můžeme pro řízení většího počtu motorů vytvořit zvláštní třídu. Této třídě již nebude záležet na konkrétním typu motoru, protože pro všechny motory bude volat stejné metody a až

```

198 void MotorStepper::speedCalculate(){
199     if(!_enableAcceleration&&_deceleration<abs(_speed-_targetSpeed)){
200         if(_speed==0||(_speed>0&&_speed<_targetSpeed)||(_speed<0&&_speed>_targetSpeed)){ //acceleration/deceleration
201             if(cycleDifference(&_cycleNumber,&_lastCycleSpeedCalc)/_accelerationStep>=1){
202
203                 _direction=(_targetSpeed>0);
204                 float y = converseRLTimeAccelerationToProgramCycle((converseAngleToStep(_acceleration)));
205                 _speed=_speed+round(cycleDifference(&_cycleNumber,&_lastCycleSpeedCalc)*y)*(2*_direction-1);
206                 _lastCycleSpeedCalc=_cycleNumber;
207             }
208         }else{//deceleration
209             if(cycleDifference(&_cycleNumber,&_lastCycleSpeedCalc)/_decelerationStep>=1){
210                 _direction=(_speed>0);
211                 float y = converseRLTimeAccelerationToProgramCycle((converseAngleToStep(_deceleration)));
212                 _speed=_speed-round(cycleDifference(&_cycleNumber,&_lastCycleSpeedCalc)*y)*(2*_direction-1);
213                 _lastCycleSpeedCalc=_cycleNumber;
214                 _direction=(_speed>0);
215             }
216         }
217     }else{
218         _speed=_targetSpeed;
219     }
220 }
221 _speedStep=speedToStep(_speed);
222 checkDirection();
223 }

```

Obrázek 5.2: Metoda pro výpočet rychlosti

samotná instance daného motoru bude vědět, jak přepsanou metodu má právě pro své řízení. Pro účely robotického ramene je toto řešení nejspíše zbytečně příliš složité. Zde v rozhodování o návrhu hraje velkou roli osobní snaha zpracovat tuto část i pro možné další využití v jiných projektech. Přidání další třídy například pro DC motor již není složité a takováto knihovna by v budoucnu mohla ulehčit mnoho času při prototypování. Z tohoto důvodu obsahují třídy i funkce, které naše rameno nevyužije, například plynulé roztočení motoru na určité otáčky.

## ■ Třída StepperMotor

Pro tyto motory je typická znalost jejich relativní polohy. Abychom se mohli zorientovat v prostoru, potřebujeme zjistit aktuální polohu tohoto motoru. K tomu potřebujeme znát koncový senzor, který při inicializaci motoru předáváme jako ukazatel na instanci jeho třídy. Poté po zavolání metody „init()“, dojde k přepnutí motoru do pomalého režimu a postupně se začne otáčet směrem ke koncovému senzoru. Mezi tím běží celý program cyklicky dále a testuje, zda již došlo k dosažení koncového senzoru. Po dosažení senzoru dojde k nastavení aktuální polohy a přesunutí do domovské polohy pro daný kloub. Po dosažení této polohy dojde k přepnutí stavu motoru a ukončení inicializace polohy. Při samotné inicializaci motoru dále požadujeme piny, kterými motor budeme řídit, časovač dle kterého si bude motor řídit otáčky, převodový stupeň a nastavení mikrostepů. Metoda „run()“ se spouští cyklicky při každém průběhu programu. Tato metoda kontroluje na základě časovače, zda má dojít k vyslání signálu pro vytvoření kroku na motoru. Rychlost, tedy časy, kdy má dojít k sepnutí, se vypočítávají na základě maximální rychlosti, dovoleného zrychlení a brždění, viz kód 5.2. Při rozběhu se začne motor zrychlovat s maximálním dovoleným zrychlením a kontroluje, zda již musí začít brzdit, aby byl schopný s jeho dovoleným zpomalením dobrzdit včas, viz kód 5.3. Tento způsob řízení je vcelku primitivní, ale pro pozicování motorů je dostatečný. V případě nutnosti přesného řízení zrychlení a pro plynulé rozběhy by bylo nutné využít jiný způsob.



```

162 void MotorStepper::accelerationCalcToPosition(){
164     int s = abs(_positionTarget-_positionAct);
169     int s = abs(_positionTarget-_positionAct);
173     int s = abs(_positionTarget-_positionAct);
174     float v = abs(_speed);
175     float s2=(v*v)/(2*(_angleByStep/_microStep)*(_deceleration));
176
177     if(s==0){
178         setSpeed(0);
179     }else if(s>s2){
180         if(_positionTarget-_positionAct>0){
181             setSpeed(_maxSpeed);
182         }else{
183             setSpeed(-_maxSpeed);
184         }
185     }else{
186         if(_positionTarget-_positionAct>0){
187             setSpeed(-sqrt(2*_deceleration*s));
188         }else{
189             setSpeed(sqrt(2*_deceleration*s));
190         }
191     }
192 }

```

Obrázek 5.3: Metoda pro dobrzdění do pozice

### ■ Třída ServoMotor

Tato třída obsahuje pouze metody, které dědí a přepisuje od nadřazené třídy. Pro řízení servomotorů je nutné generovat PWM signál. Tento signál musí mít určitou frekvenci pro daný typ motoru. Poté délkou pulzu ovládáme natočení motoru. STM32 obsahuje velké množství časovačů, které jsme schopni využít generováním pulzů nezávisle na hlavním programu. Abychom zbytečně nezatěžovali mikrokontroler, využijeme tyto časovače právě pro řízení servomotorů. Tedy stačí nám v metodě „run()“ upravovat nastavení PWM signálu. Toto nastavování by teoreticky mohlo probíhat vždy pouze při změně této hodnoty, ale pro kontrolu změny bychom museli přidat podmínku, která by výkon zase snížila. Při inicializaci motoru požaduje konstruktor ukazatel na časovač a hodnoty PWM signálu.

### ■ Třída EndSwitch

Slouží převážně pro lepší organizaci a snadné předávání koncových senzorů mezi metodami. Obsahuje pouze jednu veřejnou metodu pro kontrolu sepnutí, viz kód 5.4. Zde se zohledňuje možnost spínače s rozpínacím nebo spínacím kontaktem.

```

11 EndSwitch::EndSwitch(GPIO_TypeDef* GPIOendSwitchPort, uint16_t GPIOendSwitchPin, bool normallyOpen) {
12     _GPIOendSwitchPort=GPIOendSwitchPort;
13     _GPIOendSwitchPin=GPIOendSwitchPin;
14     _normallyOpen=normallyOpen;
15 }
16 EndSwitch::~EndSwitch() {
17 }
18
19 bool EndSwitch::check(){
20     return HAL_GPIO_ReadPin(_GPIOendSwitchPort,_GPIOendSwitchPin)^_normallyOpen; //platí pro pullUp
21 }
22 {

```

Obrázek 5.4: Metoda pro kontrolu sepnutí spínače



```

68 void MotorsDriver::run(){
69     (*this.*_runFunction)();
70 }
71 void MotorsDriver::initMotors(){
72     _runFunction=&MotorsDriver::_initMotors;
73 }
74 void MotorsDriver::_initMotors() {
75     if (_initialization == 0) {
76
77         for (int i = 0; i < _numberOfMotors; i++) {
78             _motors[i]->setPositionDegree(_motors[i]->getActualPositionDegree());
79         }
80         _motorIdInit = 0;
81         if (_motorIdInit > _numberOfMotors - 1) {
82             //ErrorHandle - žádný motor
83         }
84         _motors[_motorIdInit]->init();
85         _initializationComplete=0;
86         _initialization = 1;
87     }
88     if(_motors[_motorIdInit]->initCheck()){
89         _motorIdInit++;
90         if (_motorIdInit > _numberOfMotors - 1) {
91             _initialization=0;
92             _initializationComplete=1;
93             _runFunction=&MotorsDriver::checkRunToPos;
94         }
95         _motors[_motorIdInit]->init();
96     }
97     checkRunToPos();
98 }
99

```

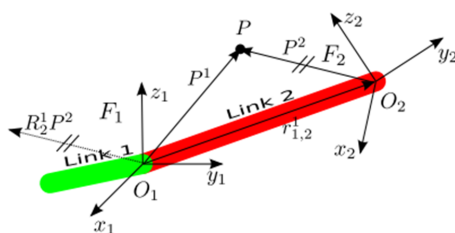
Obrázek 5.5: Metody pro řízení více motorů

## ■ Třída MotorsDriver

Tato třída se stará o správný chod všech motorů, kontroluje jejich pozici a sjednocuje všechny motory, tak abychom se nemuseli o každý motor starat zvlášť. Pro správný chod je nutné nechat běžet program ve smyčce a pravidelně nechat tuto třídu obstarávat motory. Jelikož na krokových motorech si generování signálu zprostředkováváme sami, bez využití externího PWM generování, musí být cyklus programu dostatečně rychlý. V této třídě najdeme metody pro přidávání dalších motorů do správy, metody pro nastavování pozic či získání aktuální pozice motorů. Po spuštění programu je nutné provést inicializaci všech motorů. K tomu účelu slouží funkce „\_initMotors()“, ta postupně spouští motory a čeká, až dojde k jejich plné inicializaci, poté spouští inicializaci pro další motor. Abychom v každém běhu programu nemuseli kontrolovat, zda má dojít pouze k vykonání pohybu u všech motorů nebo i ke kontrole inicializace, voláme z hlavního programu cyklicky funkci „run()“, viz kód 5.5. Ta obsahuje ukazatel na vlastní funkci, díky čemuž je možné zavoláním příkazu pro inicializaci přepsat funkci, která se bude cyklicky spouštět. Po vykonání inicializace se zase předá ukazateli funkce pro obvyčejný běh motorů.

### ■ 5.2.5 Plánování pohybu - Kinematika

Kinematika je obor zabývající se pohybem těles, ale nestará se o příčiny tohoto pohybu. Pokud budeme chtít mít možnost řídit i polohu ramene a ne pouze úhly natočení jednotlivých kloubů, bude nutné zabývat se kinematikou



**Obrázek 5.6:** Systémy souřadnic F1 a F2

tohoto ramene. Pro řízení budeme potřebovat vyřešit dva základní typy úloh. První z těchto úloh se bude zabývat dopřednou kinematikou. Znalosti této kinematiky využijeme v případě, kdy známe natočení jednotlivých kloubů ramene a potřebujeme znát pozici a natočení některého z bodů na rameni. Typicky vyšetřujeme pozici koncového mechanismu, ale lze obdobným způsobem kontrolovat, zda se některá část robota nedostala do zakázané oblasti. Druhá a složitější úloha se zabývá kinematikou inverzní. Jak název napovídá, jedná se o opačný problém. Známe polohu, natočení koncového bodu a potřebujeme zjistit natočení jednotlivých os. Oproti přímé kinematice, v tomto případě může existovat více řešení.

#### ■ Reprézentace souřadnic [44]

Pro obě kinematiky potřebujeme mít definované souřadnicové systémy a převody mezi nimi. Pro popis polohy se v robotice často využívá homogenních transformačních matic. Pomocí této matice můžeme zapsat polohu souřadného systému F2 vzhledem k systému souřadnic F1 jako:

$$T_2^1 = \left[ \begin{array}{ccc|c} & R_2^1 & & r_{1,2}^1 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right] \quad (5.1)$$

$R$  zde představuje matici rotací o velikosti  $3 \times 3$ , která charakterizuje natočení souřadných systémů vůči sobě. Matice  $r$  oproti tomu představuje translační vektor  $s$  velikostí  $1 \times 3$ . Tento vektor nám charakterizuje, jak jsou vůči sobě dané souřadné systémy posunuty.

Rotací matici  $R$  můžeme rozdělit na součin matic:

$${}^A_B R = R_x \cdot R_y \cdot R_z \quad (5.2)$$

Zde každá z matic  $R_x$ ,  $R_y$ ,  $R_z$  charakterizuje natočení souřadného systému pouze kolem své jedné osy  $x$ ,  $y$ , nebo  $z$ .

■ Rotace kolem osy  $x$ :

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (5.3)$$

- Rotace kolem osy y:

$$R_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (5.4)$$

- Rotace kolem osy z:

$$R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

Jelikož naše rameno se skládá z více na sobě nezávisle se pohybujících částech, nebudeme vždy přímo znát potřebné parametry pro přímý převod souřadnicových systémů. Vždy ale známe vzájemnou polohu jednotlivých vedlejších částí ramene. Potřebujeme tedy znát způsob, jak díky znalosti vzájemné polohy souřadných systémů F1—F2 a F2—F3 zjistit transformační matici pro souřadné systémy F1—F3. Tuto matici zjistíme pouhým vynásobením transformačních matic předešlých systémů:

$$\begin{aligned} T_3^1 &= \left[ \begin{array}{ccc|c} & R_3^1 & & r_{1,3}^1 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right] = T_2^1 \cdot T_3^2 \\ &= \left[ \begin{array}{ccc|c} & R_2^1 & & r_{1,2}^1 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right] \cdot \left[ \begin{array}{ccc|c} & R_3^2 & & r_{2,3}^2 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right] \\ &= \left[ \begin{array}{ccc|c} & R_2^1 \cdot R_3^2 & & R_2^1 \cdot r_{2,3}^2 + r_{1,2}^1 \\ - & - & - & - \\ 0 & 0 & 0 & 1 \end{array} \right] \end{aligned} \quad (5.6)$$

$$R_3^1 = R_2^1 \cdot R_3^2 \quad (5.7)$$

$$r_{1,3}^1 = R_2^1 \cdot r_{2,3}^2 + r_{1,2}^1 \quad (5.8)$$

Zde je dobré si všimnout, jak se změní matice rotací a matice translací, kde výslednou rotační matici získáme pouhým vynásobením předešlých rotačních matic, viz rovnice 5.7. Pro výpočet translace je nutné uvažovat nejen vektory posunutí souřadných systémů, ale i jejich rotaci vůči sobě, viz rovnice 5.8. [44, 45]

Parametr	Popis
$\theta$	rotace kolem osy z
d	translace v ose z
a	translace v ose x
$\alpha$	rotace kolem osy x

**Tabulka 5.1:** Popis jednotlivých parametrů Denavit-Hartenbergovy úmluvy

### ■ Dopředná kinematika s pomocí Denavit-Hartenbergovy úmluvy

Pro jednotlivé části ramene bychom teoreticky mohli souřadné systémy zavést náhodně, ale poté by takový popis robotického ramene byl chaotický a nepřehledný. Proto bylo zavedeno několik metod, které se snaží jednoduchou a systematickou cestou rekurzivně definovat souřadné systémy pro jednotlivé části robotů. Mezi nejvyužívanější patří Khalil-Kleinfingerova úmluva a Denavit-Hartenbergova (D-H), kterou využijeme k popisu i my. Tato úmluva je vhodná hlavně pro sériové kinematiky, což je i náš případ. Při využití D-H úmluvy nám stačí pouze 4 parametry k převodu souřadnicového systému jednoho ramene na druhé. Tyto parametry můžeme vidět v tabulce 5.1.[46]

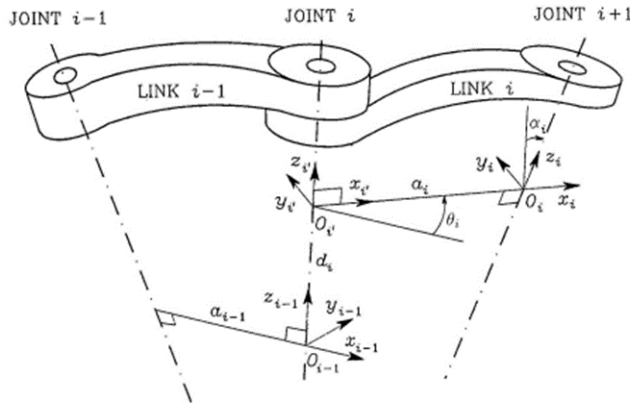
Pomocí těchto parametrů vytvoříme následující matice translací a rotací:

$$T_{Rz}(\theta_n) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

$$T_z(d_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.10)$$

$$T_x(a_n) = \begin{bmatrix} 1 & 0 & 0 & a_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.11)$$

$$T_{Rx}(\alpha_n) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_n & -\sin \alpha_n & 0 \\ 0 & \sin \alpha_n & \cos \alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.12)$$



**Obrázek 5.7:** Definice souřadných systémů DH úmluvou [44]

Výslednou transformační matici získáme vynásobením těchto matic:

$$\begin{aligned}
 T_{n-1}^n &= T_{Rz}(\theta_n) \cdot T_z(d_n) \cdot T_x(a_n) \cdot T_{Rx}(\alpha_n) \\
 &= \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cdot \cos \alpha_n & \sin \theta_n \cdot \sin \alpha_n & a_n \cdot \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cdot \cos \alpha_n & -\cos \theta_n \cdot \sin \alpha_n & a_n \cdot \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.13)
 \end{aligned}$$

Pro převod v celém kinematickém řetězci stačí jednotlivé transformační matice vynásobit, viz rovnice 5.6. [44, 45]

Samotná definice souřadných systémů je pak dle D-H úmluvy stanovena takto (za předpokladu znalosti souřadného systému  $F_{i-1}$  viz obrázek 5.7) [44]:

- Osu  $z_i$  volíme podél osy rotace/translace kloubu Joint  $i+1$  a osu  $z'_i$  podél osy rotace/translace kloubu Joint  $i$ .
- Počátek  $O_i$  souřadného systému  $F_i$  umístíme do průsečíku osy  $z_i$  a normály os  $z_{i-1}$  a  $z_i$ . Počátek  $O'_i$  souřadného systému  $F'_i$  umístíme do průsečíku osy  $z_{i-1}$  a již zmíněné normály.
- Osy  $x_i$  a  $x'_i$  volíme podél normály ve směru od kloubu Joint  $i$  do kloubu Joint  $i+1$ .
- Osy  $y$  volíme tak, aby výsledné souřadné systémy byly pravotočivé.

Pro snadné určení pravotočivého systému si můžeme pomoci pravidlem pravé ruky. Pokud palec ukazuje ve směru osy  $z$ , ukazováček ve směru  $x$ , tak poté prostředníček nám určuje směr osy  $y$ . Tvrzení je pravdivé za předpokladu, že nám prsty mezi sebou svírají  $90^\circ$ .

Dle tohoto návodu k D-H úmluvě jsme stanovili parametry pro naše rameno, viz tabulka 5.2.

Díky znalosti všech transformačních matic již není problém pouhým násobením těchto matic určit pozici koncového bodu v závislosti na zadaných úhlech natočení kloubů, tedy máme matematicky popsanou dopřednou kinematiku.

Přechod mezi částmi	$\theta$	d	a	$\alpha$
0-1	0	213 mm	0	-90°
1-2	-90°	0	210 mm	0
2-3	90°	65 mm	0	90°
3-4	0	162 mm	0	-90°
4-5	0	0	0	90°
5-6	0	78 mm	0	0

**Tabulka 5.2:** Naše D-H parametry

### ■ Inverzní kinematika

Pokud se podíváme na transformační matice, zjistíme, že jednotlivé prvky obsahují goniometrické funkce. Inverzní kinematika se tedy potýká s problémem řešení nelineárních rovnic. Takových řešení je možných více a existuje více způsobů, jak řešení zjistit. Jedním z možných způsobů je využití Newton-Raphsonovy metody, což je iterační metoda na řešení odhadu pro řešení soustavy nelineárních rovnic. Princip využití této metody pro inverzní kinematiku je sepsán v knize "Theory of applied robotics"[45].

### ■ Implementace do STM32

V našem případě využijeme pro řešení inverzní i dopředné kinematiky již napsanou knihovnu 6-DOF Robot Arm Kinematics [47]. Tato knihovna je psána v jazyce C++, tedy ji můžeme jednoduše implementovat do našeho kódu. Tato knihovna je schopná na základě D-H matice vypočítat řešení dopředné kinematiky. Pro případ inverzní kinematiky nám vrátí jako výstup několik možných řešení a proměnnou oznamující, zda bylo řešení vůbec nalezeno. Pro správné fungování s naším ramenem bylo nutné upravit parametry D-H matice a upravit způsob výpočtu dopředné kinematiky.

Funkce pro výpočet inverzní a dopředné kinematiky jsme otestovali a odladili. Dále je ale již nevyužíváme a jsou v programu určeny pro případ využití ramene bez řídicího systému ROS, kterým se zabýváme v kapitole 5.3. Dále by tato část mohla být rozšířena o algoritmy sloužící pro plánování pohybu po určité trajektorii. Takováto trajektorie by mohla být například charakterizována Bézierovou křivkou.

### ■ 5.2.6 Komunikace

#### ■ Třída ControlPanel

Pro ovládání ramene jsme zvolili 3 různé způsoby. Prvním důležitým způsobem, jak dodat určité povely ramenu, je využití ovládacího panelu, který je

součástí krytu elektroniky. Zde se nachází 3 LED indikátory a 3 spínače. LED indikátory nám signalizují, zda probíhá komunikace, zda je robot správně inicializován a nebo jestli nedošlo k některému z problémů, jako může být požadování neplatného úhlu na některém z ramen. Tlačítka slouží pro nouzové zastavení, zapnutí inicializačního cyklu a poslední lze využít pro spouštění testovacích pohybů.

### ■ Třída `SerialCommunication`

V případě nevyužití ROS, je nutné zajistit jiný způsob komunikace přes sériovou linku. O tuto komunikaci se stará třída `SerialCommunication`. Díky ní jsme schopni odesílat do terminálu a číst z terminálu příkazy.

### ■ Třída `ROSCommunication`

Aby bylo možné využít rameno v kombinaci s ROS, je nutné zajistit komunikaci s tímto systémem. O tuto komunikaci se stará tato třída, která využívá knihovny `ROSSerial` viz kapitola 5.3.3. Komunikace probíhá přes USB pomocí sériového rozhraní `USART`. Pro zrychlení komunikace a snížení zátěže pro mikrokontroler využijeme možnost aktivovat `DMA` (`Direct Memory Access`). Díky tomu jsou hodnoty pro komunikaci zapisovány přímo do paměti a čteme je až ve chvíli, kdy je skutečně potřebujeme. [48]

## ■ 5.3 ROS

Vzhledem k předpokladu, že rameno bychom měli být schopni řídit s pomocí ROS, je nutné se s tímto frameworkem seznámit a vytvořit rozhraní pro komunikaci s našim ramenem.

### ■ 5.3.1 Framework ROS

Robot Operating System (ROS) je open-source framework, který slouží k prototypování robotů. Zajišťuje simulaci hardwaru, management softwarových knihoven, řízení hardwaru, poskytuje nástroje pro často využívané funkcionality, jako je například plánování pohybu. ROS zajišťuje tzv. plumbing, tedy pro více na sobě nezávisle běžících programů zajišťuje vzájemnou komunikaci. Vzhledem k tomu, že tyto jednotlivé části mohou být psány v různých jazycích a díky faktu, že je open-source, má ROS velkou komunitu, která vytváří další balíčky, vytváří k nim návody a je ochotná poradit ostatním s řešením problémů. Vzhledem k velkému množství knihoven je pro běžné aplikace téměř nepotřebné vytvářet nové rozsáhlé aplikace, pouze si vystačíme s úpravami a nastavováním balíčků, které jsou již otestovány a využívány komunitou. [49, 50]

Samotný ROS nemusí běžet pouze na jednom zařízení. Vzhledem k přístupu komunikace peer to peer mohou jednotlivé části běžet na více zařízeních a komunikovat spolu přes lokální síť.

První verze ROS Mango Tango byla vydána již v roce 2009. Od té doby vyšlo mnoho distribucí, které obsahují nové aktualizace jednotlivých balíčků. V roce 2017 byla vydána nová verze ROS2, ovšem v současnosti stále zůstává podpora pro některé distribuce verze ROS1. V této práci jsme zvolili distribuci ROS Noetic. Tato distribuce má plánovanou podporu do roku 2025 a vzhledem k velkému množství návodů, které pracují právě s touto distribucí, byla pro nás nejlepší volbou. [51, 52]

## 5.3.2 Struktura ROS

### ROS Master

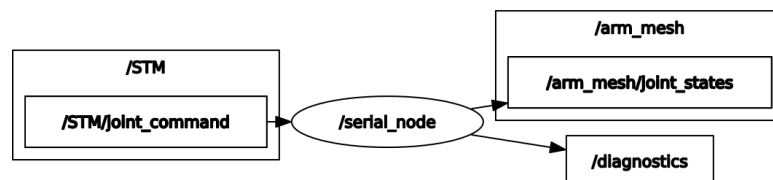
Základní strukturu tvoří ROS Master a ROS Nodes. Master nezajišťuje žádné funkce pro robota, pouze se stará o správné registrování uzlů (nodes) a následně obstarává komunikaci mezi nimi. Zpravidla na jednom zařízení běží pouze jeden Master.

### ROS Nodes

Nodes, neboli uzly, představují softwarové procesy, které mohou být samostatně kompilovány, spouštěny a spravovány. V ideálním případě by každý node měl sloužit pouze k jednomu účelu, tedy například získání teploty ze senzoru nebo ovládání motoru. Každý z těchto uzlů se při svém spuštění musí registrovat u Mastera, aby byl propojen na síť ROSu a mohl komunikovat s dalšími uzly. Tato komunikace je zprostředkována přes ROS Topics.

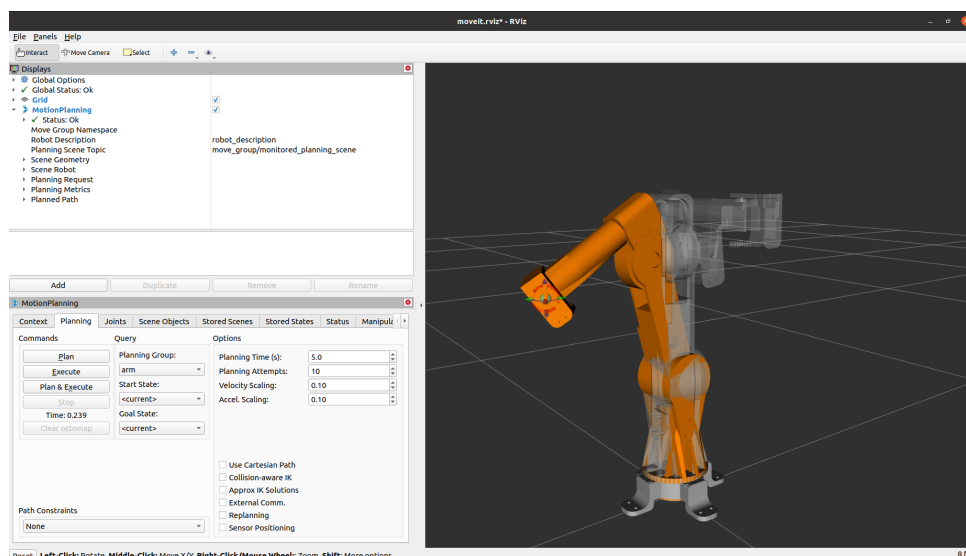
### ROS Topics

Topics, neboli téma, slouží jako komunikační prvek mezi uzly. Každý topic má k sobě napojených několik Nodes, kdy tyto uzly mohou být buď publisher, nebo subscriber. Publisher do topicu nahrává nová data, oproti tomu subscriber tyto data čte a dále využívá. Jeden nodes může přijímat a odesílat data do více Topiců, tím vzniká rozsáhlá struktura komunikace, která je pro ROS typická. Na obrázku 5.8 můžeme vidět jednoduché schéma komunikace získané programem Rqt\_graph, který prochází registrované nodes a topics a na jejich základě vytváří grafickou mapu.



**Obrázek 5.8:** Node pro seriovou komunikaci s STM32 a topics (témata) se kterými pracuje





Obrázek 5.9: Rviz s modulem MoveIt

## ROS Services

Services slouží ke stejnému účelu jako topics, s tím rozdílem, že nepředávají data kontinuálně, ale pouze pokud si o ně nodes přímo zažádá.

### 5.3.3 Námí využívané programy a funkce

#### Rviz

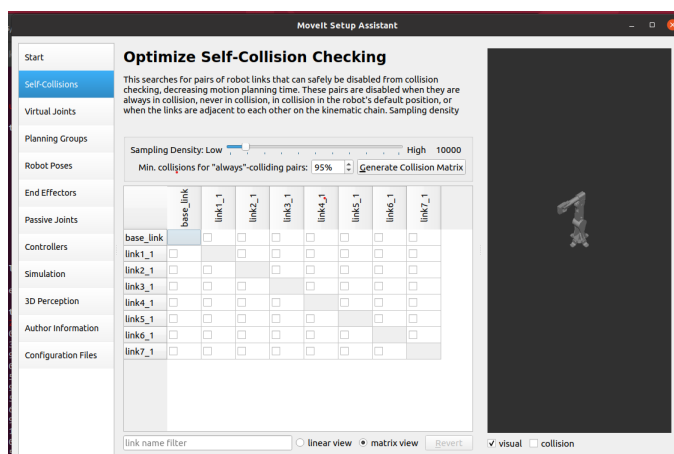
ROS visualization, neboli Rviz, je nástroj pro 3D vizualizaci robotů, senzorů, prostředí. Nestará se o simulaci robota, nýbrž zobrazuje jeho aktuální stav viz obrázek 5.9. Je tedy vhodný pro kontrolu správnosti dat, se kterými pracuje robot, nebo například v kombinaci s modulem MoveIt lze využít pro přípravu, plánování a vizualizaci pohybu.

#### Gazebo

Gazebo oproti aplikaci Rviz slouží jako 3D grafický simulátor. Výhodou je přítomnost fyzikálního enginu, díky kterému umožňuje realistické testování robotů před samotnou implementací na fyzickém robotu.

#### MoveIt

Balíček knihoven MoveIt v ROS slouží k plánování pohybu robota na základě našich požadavků. Tento nástroj je schopný generovat trajektorie pohybu i pro roboty s velkým stupněm volnosti. S jeho pomocí je možné plánovat trajektorie s vyhnutím se kolizí s okolními předměty. Aby byl schopný takto plánovat pohyb, musí ovládat výpočet dopředné a inverzní kinematiky. Pokud budeme navrhovat jakéhokoliv robota, který má být schopný pohybu a bude



Obrázek 5.10: MoveIt Setup Assistant

řízení pomocí ROS, můžeme téměř určitě MoveIt využít, což platí i pro náš případ. [53, 54]

Pro řešení trajektorií využívá tato knihovna jako základní plánovací algoritmus framework OMPL (Open Motion Planning Library). Jde o soubor algoritmů, které zvládají numericky řešit inverzní kinematiku a díky tomu plánovat trajektorii pohybu. [55]

Jednou ze součástí MoveIt balíčku je asistent pro konfiguraci robota, viz obrázek 5.10. Tento program nám pomůže s konfigurací ramene tak, aby měl MoveIt dostatek potřebných informací. Tato konfigurace probíhá na základě URDF modelu robota, který potřebujeme i pro případné vizualizace. Pokud máme vytvořený model ramene například v programu Fusion360, jsme schopni vygenerovat URDF model přímo z této sestavy ramene. [54]

Pro snadné ovládání MoveIt funkcí jej můžeme spustit jako plugin pro Rviz, viz obrázek 5.9.

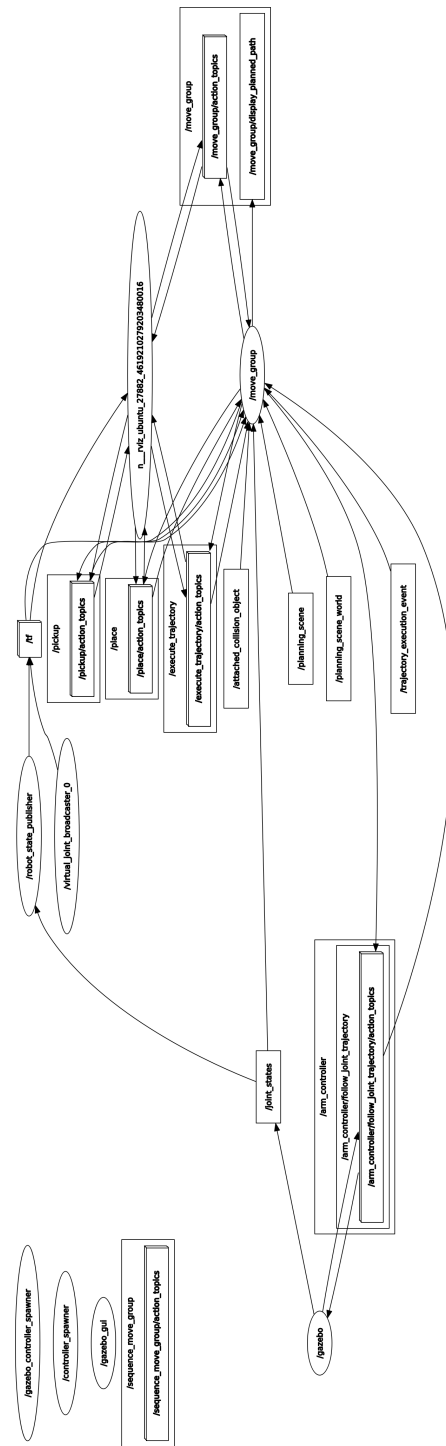
## ROS Serial

Abychom mohli ROS využívat, musíme zajistit komunikaci mezi našim mikrokontrolerem a samotným frameworkem ROS. Pokud bychom využívali platformu ROS2, mohli bychom zvolit možnost micro-ROS, což je knihovna obsahující upravený samotný ROS tak, aby mohl běžet i na zařízeních jako je STM32. Jelikož ale samotné řízení motorů ramene řešíme nezávisle na ROS, zvolili jsme pro komunikaci knihovnu ROS Serial. [56]

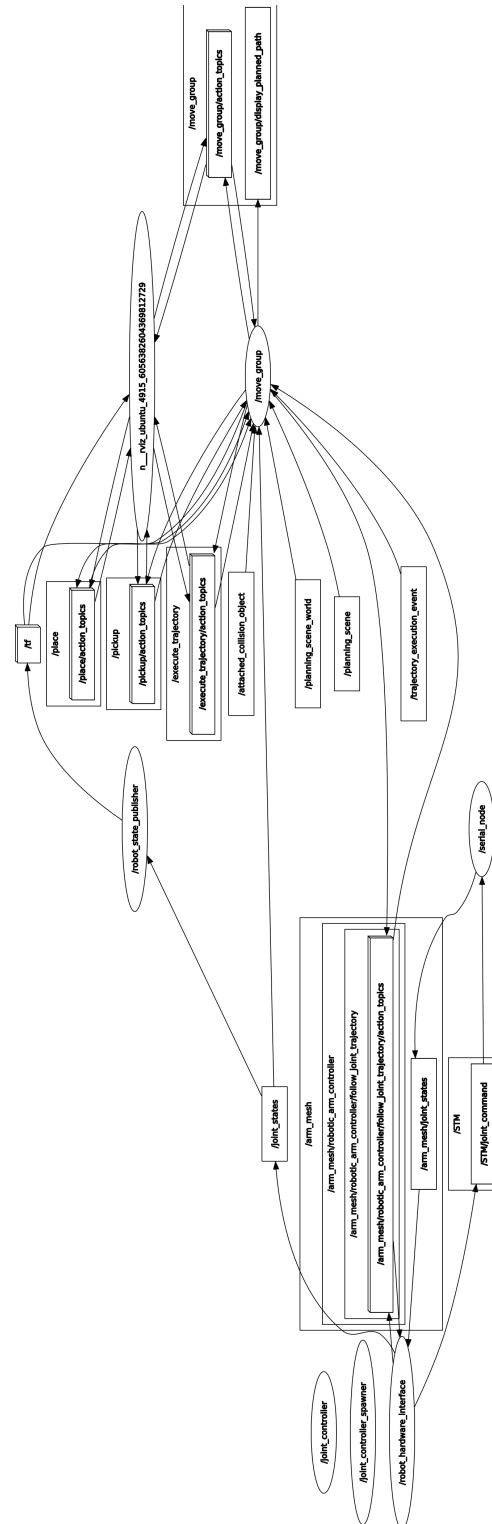
Jde o soubor knihoven zajišťujících protokol pro komunikaci mezi systémy s ROS a jinými systémy přes sériové porty. Díky tomu jsme například schopni odesílat zprávy z ROS přes sériové rozhraní (jako je v našem případě UART) do embedded systému. ROS Serial obsahuje knihovny jak pro stranu systému s ROS, tak pro zařízení jako jsou Arduino, STM32 a další. Tyto knihovny jsou psány v C++, což nám umožňuje snadnou implementaci přímo do našeho kódu. [57]

### ■ 5.3.4 ROS Implementace

Vzhledem k tomu, že jsme se s frameworkem ROS začínali teprve seznamovat, tato část práce pro nás byla náročnější. Ovšem velkou pomocí nám byla komunita kolem ROS, která vytvořila velké množství návodů a videí popisujících téměř celý postup konfigurace a nastavování podobných ramen. Naším cílem v této části je ověřit a zprovoznit řízení ramene pomocí MoveIt. Jak zmiňujeme v kapitole 5.3.3, MoveIt obsahuje konfiguračního asistenta (viz 5.10), ve kterém postupně nastavíme rozmezí pohybů kloubů, řešič kolizí a další parametry. Po dokončení konfigurace nám tento asistent vygeneruje soubory obsahující konfiguraci MoveIt, ale také soubory pro spuštění simulace v Gazebo s řízením z Rviz, pomocí modulu pro MoveIt. V této části jsme narazili na problémy s URDF modelem, který jsme generovali přímo z CAD modelu. Tento URDF model jsme museli upravit, jelikož z důvodu špatně nastavených vazeb v sestavě ramene docházelo ke špatnému generování parametrů kloubů. Po vyřešení těchto problémů jsme již měli možnost spouštět funkční simulaci v Gazebo. Schéma topiců a nodů, vygenerované pomocí Rqt\_graph je vidět na obrázku 5.11. Pro řízení reálného ramene jsme museli vytvořit node pro hardware interface, tak aby ROS věděl jakým způsobem lze s tímto ramenem komunikovat. Poté bylo již nutné pouze poupravit spouštěcí soubory, abychom využívali náš interface místo Gazebo simulace. Celé schéma topiců a nodů pro řízení reálného ramene můžeme vidět na obrázku 5.12. V tento okamžik jsme již schopni plánovat pohyb za pomoci MoveIt a nechávat tento pohyb vykonávat na reálném ramenu.



Obrázek 5.11: Graf nodů a tópiců pro simulaci v Gazebo



Obrázek 5.12: Graf nodů a topiců při řízení reálného ramene



## Kapitola 6

### Závěr

Cílem této práce bylo navrhnout a vyrobit robotické rameno, které bude vyrobeno převážně pomocí technologie 3D tisku. Po navržení a vyrobení mechanické části jsme se zabývali i návrhem hardwaru a softwaru pro řízení. Aby bylo možné v budoucnu jednoduše využívat vyvinuté rameno, zabývali jsme se i propojením ramene se systémem ROS, který značně ulehčí plánování pohybu ramene.

Během rešerše jsme prozkoumali historii průmyslových robotů, vytvořili si přehled o robotických ramenech využívaných v průmyslu a poté jsme prozkoumali existující řešení, které využívají 3D tisk jako hlavní technologii výroby. Tyto poznatky nám dále ulehčily práci při samotném návrhu jednotlivých částí.

V rámci návrhu jsme specifikovali požadavky na rameno a provedli volbu vhodných motorů a převodových mechanismů. Konkrétně jsme zvolili modelářské servomotory a krokové motory v kombinaci s cykloidními převodovkami. Pro tyto převodovky jsme navrhli několik prototypů, které jsme postupně otestovali a vylepšovali. Speciálním případem se stala námi navržená dvojitá cykloidní převodovka, která zajišťuje velký převodový poměr. Během testování jsme však zjistili, že námi navržená dvojitá cykloidní převodovka nedosahuje tak dobrých výsledků, v porovnání s ostatními převodovkami. Díky tomu je značně snížena přesnost celého ramene. Pro využití plného potenciálu ramene by tedy bylo nutné přehodnotit a předělat konstrukci této části mechanismu.

V hardwarové části jsme se zabývali volbou mikrokontroleru, driveru pro naše pohony a koncovými senzory pro detekci polohy krokových motorů. Pro finální řešení jsme zvolili prototypovou desku Nucleo-64 STM32F302. Tato deska se ukázala být dostatečně výkonná a bylo by možné využít i některou ze slabších variant. Pro řízení krokových motorů jsme využili drivery DRV8825 v kombinaci s CNC modulem pro jejich snadné připojení.

Softwarové řešení jsme rozdělili na dvě části. V první části jsme se zabývali programováním řídicího systému, který běží na samotném mikrokontroleru STM32. Zde jsme se zaměřili na univerzálnost námi psaného kódu za využití principů objektově orientovaného programování. Jedná se hlavně o knihovnu zabývající se řízením samotných motorů, kterou jsme koncepčně řešili tak, aby bylo možné ji využít i v jiných aplikacích. Součástí našeho kódu jsou jak funkce pro řízení samotných motorů, tak i funkce pro výpočet inverzní a

dopředné kinematiky. Druhá část je věnována implementaci řídicího systému ROS a jeho následné využívání pro řízení našeho ramene. Toto propojení jsme odzkoušeli a můžeme říci, že díky množství knihoven připravených pro ROS je využití takového systému velké ulehčení pro řízení jakéhokoliv robota, nejen robotických ramen.

Robotické rameno splňuje zadané požadavky. Při dalším rozvoji by bylo možné dosáhnout vyšší přesnosti ramene například úpravou dvojité cykloidní převodovky. Věříme, že díky zvoleným technologiím není problém takovéto rameno různě modifikovat a upravovat pro námi požadovanou konkrétní aplikaci.



# Příloha A

## Literatura

- [1] N. G. Hockstein, C. G. Gourin, R. A. Faust, and D. J. Terris, “A history of robots: From science fiction to surgical robotics,” *Journal of Robotic Surgery*, vol. 1, pp. 113–118, 3 2007. [Online]. Available: <https://link.springer.com/article/10.1007/s11701-007-0021-2>
- [2] “Historie robotů? sahá až do řecké mytologie!” 2014. [Online]. Available: <https://factoryautomation.cz/historie-robotu-saha-az-do-recke-mytologie/>
- [3] F. Šolc and L. Žalud, *Základy robotiky*, 11 2002.
- [4] C. Jobson, “This programmable 6,000-part drawing boy automata is arguably the first computer and it was built 240 years ago — colossal,” 2013. [Online]. Available: <https://www.thisiscolossal.com/2013/11/the-writer-automata/>
- [5] M. E. Moran, “Evolution of robotic arms,” *Journal of Robotic Surgery*, vol. 1, pp. 103–111, 5 2007. [Online]. Available: <https://link.springer.com/article/10.1007/s11701-006-0002-x>
- [6] M. I. Havel, *Úvod do teorie kognitivních robotů*. SNTL, 1990.
- [7] A. Marsh, “Past forward - in 1961, the first robot arm punched in,” *IEEE Spectrum*, vol. 59, 9 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9881265/>
- [8] S. Jiří, *Průmyslové roboty a manipulátory*. Vysoká škola báňská-Technická univerzita Ostrava, 2007. [Online]. Available: [http://www.elearn.vsb.cz/archivcd/FS/PRM/Text/Skripta\\_PRaM.pdf](http://www.elearn.vsb.cz/archivcd/FS/PRM/Text/Skripta_PRaM.pdf)
- [9] R. Pavel, *AUTOMATIZACE (roboty a manipulátory)*. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STROJNÍHO INŽENÝRSTVÍ, 2003. [Online]. Available: [https://ust.fme.vutbr.cz/tvareni/img/opory/emm\\_mechanizace\\_a\\_automatizace\\_roboty\\_rumisek.pd](https://ust.fme.vutbr.cz/tvareni/img/opory/emm_mechanizace_a_automatizace_roboty_rumisek.pd)
- [10] H. Vlastimil, *Průmyslové roboty a manipulátory*. Technická Univerzita v Liberci, 2019. [Online]. Available: <https://elearning.tul.cz/mod/resource/view.php?id=198854>

- [11] “Kinematika robotů.” [Online]. Available: <http://skola.hellebrand.cz/text1011/mch/robotika-kinematika.pdf>
- [12] “Full flight simulator - Wikipedia.” [Online]. Available: [https://en.wikipedia.org/wiki/Full\\_flight\\_simulator](https://en.wikipedia.org/wiki/Full_flight_simulator)
- [13] A. Bicchi and D. Prattichizzo, “Manipulability of cooperating robots with unactuated joints and closed-chain mechanisms,” *IEEE Transactions on Robotics and Automation*, vol. 16, pp. 336–345, 8 2000.
- [14] E. Rosales, Q. Gan, and J. Gan, “Forward and inverse kinematics models for a 5-dof pioneer 2 robot arm,” 01 2002.
- [15] C. Stover and E. W. Weisstein, “Coordinate geometry – from wolfram mathworld.” [Online]. Available: <https://mathworld.wolfram.com/CoordinateGeometry.html>
- [16] RAYMING, “The different applications of the factory robot arm.” [Online]. Available: <https://www.raypcb.com/factory-robot-arm/>
- [17] ABB Robotics, “Irb 1100.” [Online]. Available: <https://new.abb.com/products/robotics/robots/articulated-robots/irb-1100>
- [18] UNIVERSAL ROBOTS, “Ur3e ultra-lightweight, compact cobot.” [Online]. Available: <https://www.universal-robots.com/products/ur3-robot/>
- [19] KUKA, “Kr 4 r600.” [Online]. Available: [https://my.kuka.com/s/product/detail/01t1i000001Vn3HAAS?language=en\\_US](https://my.kuka.com/s/product/detail/01t1i000001Vn3HAAS?language=en_US)
- [20] FANUC, “Collaborative industrial robot fanuc cr-4ia.” [Online]. Available: <https://www.fanuc.eu/pl/en/robots/robot-filter-page/collaborative-robots/collaborative-cr4ia?returnurl=https%3A%2F%2Fwww.fanuc.eu%2Fpl%2Fen%2Frobots%2Frobot-filter-page%23%3Ft%3Dced8e7c08dc14f4b8bface83fc85337d%2C1fc42cb423f2445bb0da1ea4a885389a>
- [21] ARCTOS, “Diy 3d printed robotic arm.” [Online]. Available: <https://arctosrobotics.com/#Assembly>
- [22] DIY Tech, “Kaudapro.” [Online]. Available: <https://www.diy-tech.it/030-kaudapro>
- [23] “Bcn3d moveo: A fully open source 3d printed robot arm,” 2016. [Online]. Available: <https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm/>
- [24] D. Papp, “Servo becomes mini linear actuator,” 2018. [Online]. Available: <https://hackaday.com/2018/06/02/servo-becomes-mini-linear-actuator/>
- [25] K. Maléř, “Měření a simulace na dc motoru,” 2016. [Online]. Available: <https://dspace.cvut.cz/handle/10467/64859>

- [26] “Different types of motors used in robotics.” [Online]. Available: <https://robocraze.com/blogs/post/types-of-motors-used-in-robotics>
- [27] P. Yedamale and M. T. Inc., “Brushless dc (bldc) motor fundamentals,” 2003. [Online]. Available: <https://ww1.microchip.com/downloads/en/AppNotes/00885a.pdf>
- [28] A. Raj, “What is stepper motor and how it works,” 2018. [Online]. Available: <https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works>
- [29] “Krokový motor – druhy a příklady aplikací krokových motorů,” 2020. [Online]. Available: <https://www.tme.eu/cz/news/library-articles/page/41861/krokovy-motor-druhy-a-priklady-aplikaci-krokovych-motoru/>
- [30] J. Skalický, *Elektrické servopohony*, vyd. 2 ed. VUT, 2001.
- [31] Open Impulse, “42byghw811 stepper motor datasheet.” [Online]. Available: <https://www.openimpulse.com/blog/products-page/product-category/42byghw811-stepper-motor-2-5-4-8-kg%E2%8B%85cm/>
- [32] oneindustry, “Planetová převodovka - oneindustry – průmyslový portál, výroba a technologie,” 2019. [Online]. Available: <https://www.oneindustry.cz/lexikon/planetova-prevodovka/>
- [33] Harmonic Drive, “Harmonic drive® strain wave gear - zero backlash.” [Online]. Available: <https://www.harmonicdrive.net/technology>
- [34] tec-science, “How does a cycloidal drive work? -,” 2019. [Online]. Available: <https://www.tec-science.com/mechanical-power-transmission/planetary-gear/how-does-a-cycloidal-gear-drive-work/>
- [35] C. Gorla, P. Davoli, F. Rosa, C. Longoni, F. Chiozzi, and A. Samarani, “Theoretical and experimental analysis of a cycloidal speed reducer,” *Journal of Mechanical Design, Transactions of the ASME*, vol. 130, 2008.
- [36] Arduino Official, “Arduino nano.” [Online]. Available: <https://store.arduino.cc/products/arduino-nano>
- [37] PROKYBER s.r.o., “Arduino nano - atmega328 (klon).” [Online]. Available: <https://e-shop.prokyber.cz/arduino/arduino-nano/>
- [38] STMicroelectronics, “Nucleo-f401re - stm32 nucleo-64 development board with stm32f401re mcu.” [Online]. Available: <https://www.st.com/en/evaluation-tools/nucleo-f401re.html>
- [39] LaskaKit, “Arm stm32 nucleo-f302r8 nucleo-64.” [Online]. Available: <https://www.laskakit.cz/arm-stm32-nucleo-f302r8-nucleo-64-vyvojova-deska/>

- [40] Pololu, “Drv8825 stepper motor driver carrier, high current (header pins soldered).” [Online]. Available: <https://www.pololu.com/product/2982/specs>
- [41] William, “A4988 vs drv8825: What are the differences between them?” 2022. [Online]. Available: <https://www.utmel.com/components/a4988-vs-drv8825-what-are-the-differences-between-them?id=2003>
- [42] Protoneer.co.nz and B. Kruger, “Arduino cnc shield,” 2015. [Online]. Available: <https://blog.protoneer.co.nz/arduino-cnc-shield/>
- [43] Pájeníčko s.r.o., “Napájecí deska xh-m229 pro zdroje typu atx 24pin (3.3v, 5v a 12v).” [Online]. Available: <https://pajenicko.cz/napajeci-deska-xh-m229-pro-zdroje-typu-atx-24pin-3v-5v-12v>
- [44] I. M. Švejda, “Optimization of robot architectures,” 2016.
- [45] R. N. Jazar, “Theory of applied robotics: Kinematics, dynamics, and control (2nd edition),” *Theory of Applied Robotics: Kinematics, Dynamics, and Control (2nd Edition)*, pp. 1–883, 2010.
- [46] H. Zhuang, Z. S. Roth, and F. Hamano, “A complete and parametrically continuous kinematic model for robot manipulators,” *IEEE Transactions on Robotics and Automation*, vol. 8, 1992.
- [47] jacobsayono, “Github - jacobsayono/6dof-kinematic: 6 dof robotic arm (c++).” [Online]. Available: <https://github.com/jacobsayono/6dof-kinematic>
- [48] R. Nimkar, “Working with usart and dma in stm32 (arm cortex m3),” 2022. [Online]. Available: <https://medium.com/@csrohit/working-with-usart-and-dma-in-stm32-arm-cortex-m3-6a0e8b3174f3>
- [49] Open Robotics, “Ros: The ros ecosystem.” [Online]. Available: <https://www.ros.org/blog/ecosystem/>
- [50] —, “Ros: Concepts - ros wiki.” [Online]. Available: <http://wiki.ros.org/ROS/Concepts>
- [51] R. Tellez and The Construct, “A history of ros (robot operating system) - the construct,” 2019. [Online]. Available: <https://www.theconstructsim.com/history-ros/>
- [52] Open Robotics, “Distributions — ros 2 documentation.” [Online]. Available: <https://docs.ros.org/en/rolling/Releases.html>
- [53] PICKNIK, “Moveit motion planning framework.” [Online]. Available: <https://moveit.ros.org/>
- [54] R. Tellez and The Construct, “What is moveit ros? all about movelt! ros,” 2018. [Online]. Available: <https://www.theconstructsim.com/ros-movelt/>

