

České vysoké učení technické v Praze  
Fakulta strojní  
Ústav přístrojové a řídicí techniky



# Robotický manipulátor

Diplomová práce

*Bc. Jan Mohelník*

Magisterský program: Automatizační a přístrojová technika

Magisterský obor: Automatizace a průmyslová informatika

Vedoucí práce: Ing. Pavel Trnka, Ph.D.

Praha, červen 2023

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mohelník** Jméno: **Jan** Osobní číslo: **475430**  
Fakulta/ústav: **Fakulta strojní**  
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:

**Robotický manipulátor**

Název diplomové práce anglicky:

**Robotic manipulator**

Pokyny pro vypracování:

1. Proveďte rešerši používaných konstrukcí robotických manipulátorů se zaměřením na aditivní technologie.
2. Proveďte rešerši dostupných open-source SW nástrojů pro řízení robotických manipulátorů a 3D tiskáren.
3. Navrhněte a realizujte konstrukční řešení robotického manipulátoru s možností výměny pracovních nástrojů.
4. Navrhněte způsob řízení robotického manipulátoru a implementujte řízení v rámci zvoleného existujícího SW nástroje.
5. Navrhněte a realizujte uživatelské rozhraní pro ovládání manipulátoru.

Seznam doporučené literatury:

- [1] Craig, J.J.: Introduction to Robotics : Mechanics and Control. Fourth edition. Global ed. Harlow United Kingdom: Pearson Education Limited; 2022. <https://worldcat.org/cs/title/1259046068>
- [2] Han, X.S., Qiang, T.: Kinematics Analysis of Palletizing Robot. Advanced Materials Research 2014 pp. 477–481. <https://doi.org/10.4028/www.scientific.net/AMR.915-916.477>
- [3] Jazar, R.N.: Theory of Applied Robotics: Kinematics, Dynamics, and Control. 2010, ISBN-13: 978-1441917492

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Pavel Trnka, Ph.D. U12110.3**

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

Datum zadání diplomové práce: **28.04.2023**

Termín odevzdání diplomové práce: **08.06.2023**

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

Ing. Pavel Trnka, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

doc. Ing. Miroslav Španiel, CSc.  
podpis diktora(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

**Vedoucí práce:**

Ing. Pavel Trnka, Ph.D.  
Ústav přístrojové a řídicí techniky  
Fakulta strojní  
České vysoké učení technické v Praze  
Technická 4  
160 00 Praha 6  
Česká republika

## **Prohlášení**

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s tím, že její výsledky mohou být dále použity podle uvážení vedoucího diplomové práce jako jejího spoluautora. Souhlasím také s případnou publikací výsledků diplomové práce, nebo její podstatné části, pokud budu uveden jako její spoluautor.

V Praze 2023

.....  
Jan Mohelník

## **Poděkování**

Rád bych touto cestou vyjádřil poděkování Ing. Pavlu Trnkovi Ph.D. za jeho cenné rady, doporučení a trpělivost při vedení mé diplomové práce. Rád bych také poděkoval své rodině, přítelkyni a všem přátelům, kteří mě při vytváření této práce podpořili, a bez jejichž pomoci by nebylo možné práci dokončit.

## Abstrakt

Tato práce se zabývá sestavením robotického manipulátoru, který má možnosti pohybu se svým koncovým nástrojem ve třech stupních volnosti, což jsou typicky směry označované souřadnicemi X, Y, Z. Tento manipulátor je vybaven výměnnou hlavicí pro práci s vícero nástroji, jako je například 3D tisk nebo plotter. Pro potřeby 3D tisku je hlava vybavena ohřívanou tryskou a extruderem, čímž nanáší vrstvy termoplastu na pracovní plochu. V případě plotteru je hlava vybavena tužkou a umožňuje tak kreslit 2D obrazce. Velkou výzvou zde nebyl jen samotný návrh konstrukce robota, který je inspirován skutečně používaným typem v průmyslu, ale také samotné řízení. Byly zde vyzkoušeny různé přístupy k získání dopředné a inverzní kinematiky. Kód s kinematikou byl poté implementován do veřejně používaného open-source firmwaru pro 3D tiskárny Marlin a k němu bylo vytvořeno grafické ovládání pro řízení robota z počítače.

Klíčová slova: Manipulátor, Paletizační robot, Plotter, 3D Tisk, Inverzní kinematika, Dopředná kinematika, Marlin

## Abstract

This work deals with the construction of a robotic manipulator that has the ability to move with its end tool in three degrees of freedom, which are typically the directions indicated by coordinates X, Y, Z. This manipulator is equipped with an exchangeable head for working with multiple tools, such as 3D printing, or plotter. For 3D printing, the head is equipped with a heated nozzle and an extruder, which applies layers of thermoplastic to the work surface. In the case of a plotter, the head is equipped with a pencil and thus allows drawing 2D shapes. The big challenge here was not only the design of the robot construction itself, which was inspired by a model actually used in the industry, but also the control itself. Various approaches to obtain forward and inverse kinematics have been tested here. The kinematics code was then implemented into the public open-source firmware for Marlin 3D printers, and a graphical control was created for controlling the robot from a computer.

Keywords: Manipulator, Palletizing Robot, Plotter, 3D Printing, Inverse Kinematics, Forward Kinematics, Marlin

# Seznam zkratek

- SLS** Selective Laser Sintering – aditivní prášková technologie
- SLM** Selective Laser Melting – aditivní prášková technologie
- FDM** Fused Deposition Modeling – aditivní výtlačná technologie
- AKF** Arburg Kunststoff Freiformen – aditivní výtlačná technologie
- MJ** Material Jetting – aditivní technologie s fotopolymerem
- SLA** Stereo Lithography – Proces tvrzení pryskyřice pomocí laseru
- DLP** Direct Light Processing – Proces tvrzení pryskyřice pomocí projektoru
- SCARA** Selective Compliance Articulated Robot Arm – typ robotického manipulátoru
- CAD** Computer Aided Design – software pro tvorbu 3D modelů
- CAM** Computer Aided Manufacturing – software pro počítačové výroby modelu
- CNC** Computer Numerical Control – numericky řízený stroj
- GPL** General Public License – typ license
- STL** Standart Triangle Language – formát souboru pro CAD
- REPRAP** Replicating Rapid Prototyper – software pro 3D tiskárny
- PLA** Polylactic acid – kyselina polyléčná – materiál pro 3D tisk
- NEMA** National Electrical Manufacturers Association – značení velikosti motorů
- SPI** Serial Peripheral Interface – sériová komunikace
- UART** Universal Asynchronous receiver-transmitter – počítačová sběrnice
- USB** Universal Serial Bus – univerzální sériová sběrnice
- PCB** Printed Circuit Board – tištěný spoj
- CoreXY** – Typ konstrukce 3D tiskárny
- Cpp** – přípona souboru formátu C++
- IDE** Integrated Development Environment – vývojové prostředí
- C++** – Programovací jazyk
- BL-Touch** – Senzor pro automatickou nivelaci podložky tiskárny
- GUI** Graphical user interface – uživatelské rozhraní

# OBSAH

<b>1 ÚVOD .....</b>	<b>- 10 -</b>
1.1 CÍLE PRÁCE .....	- 11 -
<b>2 TEORETICKÁ ČÁST.....</b>	<b>- 12 -</b>
2.1 DRUHY KONSTRUKCÍ PRO ADITIVNÍ FDM TECHNOLOGIE .....	- 13 -
2.1.1 Pracovní oblasti robotů.....	- 14 -
2.2 PRŮMYSLOVÉ ROBOTICKÉ MANIPULÁTORY V ADITIVNÍ TECHNOLOGIE .....	- 15 -
2.2.1 Paralelogram .....	- 16 -
2.3 OPEN-SOURCE SOFTWAREY PRO ŘÍZENÍ MANIPULÁTORŮ A 3D TISKÁREN .....	- 18 -
2.3.1 RepRap .....	- 19 -
2.3.2 Marlin.....	- 19 -
2.3.3 Klipper.....	- 19 -
2.3.4 GRBL.....	- 20 -
2.3.5 LinuxCNC.....	- 20 -
2.3.6 Shrnutí.....	- 21 -
2.4 G-CODE.....	- 22 -
2.4.1 Výběr některých používaných příkazů .....	- 22 -
<b>3. NÁVRH A KONSTRUKCE ROBOTICKÉHO MANIPULÁTORU.....</b>	<b>- 23 -</b>
3.1 VOLBA PŘEVODOVKY A SPOJKY .....	- 25 -
3.1.1 Zvolená planetová převodovka.....	- 26 -
3.1.2 Zvolená spojka.....	- 27 -
3.2 ZAKÁZKOVĚ VYROBENÉ DÍLY .....	- 28 -
3.3 NÁVRH A KONSTRUKCE ROTAČNÍHO STOLU .....	- 29 -
3.3.1 Volba ložiska a jeho uložení.....	- 30 -
3.4 NÁVRH A KONSTRUKCE HORNÍ ČÁSTI ROBOTA .....	- 31 -
3.4.1 Volba ložisek a uložení.....	- 32 -
3.5 NÁVRH A KONSTRUKCE VÝMĚNNÉHO NÁSTROJE.....	- 34 -
3.5.1 Maxwellova kinematická vazba.....	- 34 -
<b>4 ELEKTRICKÁ KONSTRUKČNÍ ČÁST .....</b>	<b>- 36 -</b>
4.1 SCHÉMA PROPOJENÍ ELEKTRONIKY V ZÁKLADNĚ .....	- 36 -
4.2 ŘÍDÍCÍ DESKA .....	- 37 -
4.3 MOTORY .....	- 38 -
4.3.1 Servo pro zámek nástroje .....	- 39 -
4.3.2 Zvolený motor pro rotační stůl.....	- 39 -
4.3.3 Zvolený motor pro ramena .....	- 40 -
4.3.4 Motor pro extruder.....	- 40 -



4.4 DRIVE CL86T A TMC2209.....	- 41 -
4.4.1 Mikrostepping.....	- 42 -
4.4.2 Komunikace s drivery.....	- 44 -
4.5 VOLBA ZDROJE.....	- 45 -
4.6 PŘEDNÍ PANEL.....	- 46 -
4.6.1 KONCOVÁ ČIDLA A E-STOP .....	- 47 -
<b>5 NÁVRH A IMPLEMENTACE ŘÍZENÍ MANIPULÁTORU .....</b>	<b>- 49 -</b>
5.1 PŘÍSTUPY PRO ZJIŠTĚNÍ INVERZNÍ A DOPŘEDNÉ KINEMATIKY MANIPULÁTORU .....	- 49 -
5.1.1 Dopředná a inverzní kinematika analytickou metodou .....	- 49 -
5.1.2 Dopředná kinematika s homogenními transformačními maticemi .....	- 53 -
5.1.3 Inverzní kinematika numerickou metodou.....	- 59 -
5.2 ŘÍDÍCÍ FIRMWARE .....	- 62 -
5.2.1 Generování firmwaru s Visual Studio Code + Platformio .....	- 62 -
5.2.2 Úpravy Marlinu.....	- 63 -
5.3 TVORBA ŘÍDÍCÍHO GUI .....	- 65 -
5.3.1 Tvorba samotných obrazovek skrze QT Designer .....	- 66 -
5.3.2 Hlavní lišta a funkcionality grafického ovládání.....	- 67 -
5.3.3 Tlačítka v menu s obrazovkami .....	- 68 -
5.3.4 Komunikace s Marlinem skrze click eventy.....	- 70 -
<b>6. TESTOVÁNÍ MANIPULÁTORU.....</b>	<b>- 73 -</b>
6.1 VOLBA SOFTWARE PRO TESTOVÁNÍ KINEMATIKY .....	- 73 -
6.2 PAPIROVÝ MODEL .....	- 74 -
6.3 ZKOUŠENÍ KRESLENÍ A 3D TISKU .....	- 74 -
6.4 TESTOVÁNÍ PŘESNOSTI A OPAKOVATELNOSTI .....	- 76 -
<b>7. ZÁVĚR .....</b>	<b>- 77 -</b>
7.1 NÁVRHY NA ZLEPŠENÍ.....	- 78 -
<b>SEZNAM OBRÁZKŮ .....</b>	<b>- 79 -</b>
<b>SEZNAM TABULEK.....</b>	<b>- 83 -</b>
<b>SEZNAM KÓDŮ .....</b>	<b>- 84 -</b>
<b>BIBLIOGRAFIE.....</b>	<b>- 85 -</b>
<b>PŘÍLOHY .....</b>	<b>- 89 -</b>
<i>Příloha na CD.....</i>	- 89 -

# 1 Úvod

Robotický manipulátor je zařízení, se kterým se lze setkat nejen v oblastech průmyslového použití, kde se často vyskytuje v aplikacích jako jsou svařování, lakování nebo pro pouhou manipulaci s objekty. Uplatnění najde i ve věcech, které lze využít v domácích dílnách, jako je například 3D tisk, obrábění, kreslení či gravírování. Tvorba takového robotického manipulátoru se tak stává disciplínou nejen konstruktérskou, ale stojí také na důkladném rozboru kinematiky, volbě elektroniky a celkovém návrhu řízení robota.

Motivací pro tuto práci se tak stala snaha o vytvoření zmenšené verze takového průmyslového robota a aspirace dostat jej do domácích dílen, kde by bylo možno využít jeho benefitů stejně, jako se to děje v průmyslové praxi. Samotných konstrukcí manipulátoru je hned několik a odvíjejí se převážně od aplikací, na které jsou používány. Inspirací pro stavbu se stalo průmyslové odvětví, a to specificky v oblasti, kde se používá robotické rameno k aditivní technologii, jelikož 3D tisk je jedním z cílů pro aplikaci tohoto manipulátoru. Další cílem je pak například kreslení.

Co se týče řízení robota, tak vývojem celého programu pro ovládání se tato práce nezabývá. Jde zde o to ukázat možnost aplikace vlastní konstrukce a řízení manipulátoru se všemi prvky, v běžně dostupném open-source firmwaru, na kterém fungují dnešní 3D tiskárny. To se jeví jako zajímavá volba, jelikož v současnosti nenajdeme příliš velkou podporu pro tento typ manipulátoru s principy robotického ramene v dostupných firmwarech, které se v 3D tiskárnách používají. A tak bylo nutno najít takový firmware, který by šel legálně přeprogramovat (tedy spadal pod licenci, která dovoluje úpravy kódu) a to tak, aby ovládal robota potřebným způsobem.

Důležitým měřítkem zde byla také snaha o dosažení co největší přesnosti pohybů a kvality konstrukce. Z tohoto důvodu nebyly při stavbě konstrukce robota použity tištěné díly z termoplastu, ale z duralu či oceli. Ty byly vyřezány na laseru či vodním paprskem. Pracovní oblast, kam může robot dosáhnout, byla navržena tak, aby byla vzdálenost nástroje od základny přibližně jeden metr. Byl to tedy často velký kompromis mezi cenou a kvalitou, jelikož s rozměry jednotlivých ramen manipulátoru už bylo nutno využít v průmyslu používaných součástek, kde jsou zařízení typu převodovky a ložiska poměrně drahé.

Při tvorbě grafického prostředí pro ovládání a zadávání úkolů manipulátoru, byl kladen důraz na uživatelsky přívětivé a jednoduché rozhraní, se kterým se půjde po sériové komunikaci propojit z počítače do řídicí desky robota.

## 1.1 Cíle práce

Tato práce při vytváření robotického manipulátoru „robota“ zahrnuje několik okruhů, počínaje volbou typu konstrukce robotického manipulátoru, od kterého se pak odvíjí velikost pracovní plochy a složitost kinematiky. Neméně důležitou volbou je řízení a návrh grafického ovládání. Jsou zde tedy stanoveny cíle:

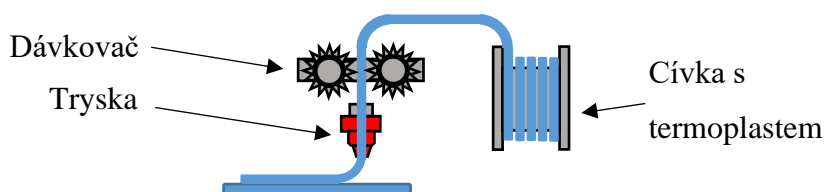
- Seznámit se s používanými konstrukcemi a principy robotických manipulátorů pro aditivní technologii.
- Manipulátor by měl mít možnost výměnného nástroje pro vícero aplikací.
- Seznámit se s dostupnými open-source firmwary, kde půjde implementovat vlastní kinematika a ovládání (např. firmware od 3D tiskáren).
- Tento robotický manipulátor zkonstruovat a spolu s upraveným firmwarem rozpohybovat.
- Vytvořit grafické řízení pro daný manipulátor.

## 2 Teoretická část

Pojem robotický manipulátor v aditivní technologii není nic nového. S rostoucím zájmem o tuto technologii přibývá řada nových způsobů a experimentů s konstrukcemi, jak zde co nejvíce zefektivnit tento technologický proces. Aditivní technologie je proces, kde vytváříme finální produkt na základě 3D modelu postupným přidáváním materiálu. Nevzniká tak takové množství odpadu jak při metodách obráběcích. [1]

Existuje vícero technologických postupů pro aditivní technologie. V dnešní době používáme převážně tyto čtyři přístupy. Prášková metoda, kde pomocí laseru vytvrzujeme vrstvu kovového prášku na podložce (SLS „Selective Laser Sintering“, SLM „Selective laser melting“). Metoda vytlačování, kdy je daný materiál tlačen přes vyhřívanou trysku na podložku (FDM „Fused depositon modeling“, AKF „Arburg Kunststoff Freiformen“). Metoda tryskání, kde pomocí kapek nanášíme fotopolymer na pracovní plochu a vytvrzujeme pomocí UV světla (MJ „Material Jetting“), nebo fotopolymerizace, kde vytvrzujeme tekutý fotopolymer pomocí laseru, či projektoru (SLA „Stereo Lithography“, DLP „Direct Light Processing“). [2]

My se budeme zabývat technologií FDM pro 3D tisk termoplastů. Ten je založen na principu nanášení vrstev na tiskovou podložku. Tento termoplast je zde dodáván ve formě vlákna, které se taví v zahřívané trysce, do níž je zaváděn pomocí extruderu (dávkovače). Výsledný výrobek je tak nanášen vrstvu po vrstvě, přičemž se chladnutím velice rychle vytvrzuje.



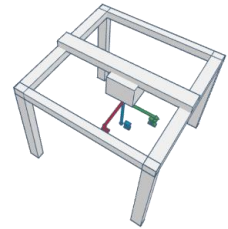
Obr. 1: Technologie FDM

Co se týče konstrukce takového zařízení, tak se odvíjí od daného technologického procesu. Pro náš případ FDM si ukážeme v následující kapitole několik z nich. Nejprve je však potřeba upozornit, že když se řekne robotický manipulátor, tak se ne vždy jedná o typické robotické rameno, ale do této kategorie spadají i například roboti s kartézskou kinematikou, delta a mnoho dalších. My se budeme specificky zabývat těmi, jež se používají pro aditivní technologie, a to jak v průmyslu, tak i v hobby odvětví. Další aplikace, která nás zajímá je plotter (souřadnicový zapisovač). Zde robot stejně jako při 3D tisku pohybuje hlavou ve směru programu, ale namísto nanášení vrstev plastu, nanáší vrstvu tuhy z tužky, či inkoustu z pera na papír a kreslí tak výsledný obraz.

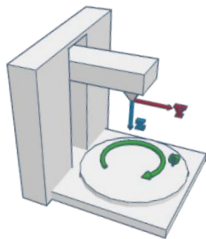
## 2.1 Druhy konstrukcí pro aditivní FDM technologie

V praxi se používá několik typů konstrukcí a jejich volba se často odvíjí od požadavků na přesnost, rychlost a velikost pracovní oblasti.

Kartézská konstrukce je v dnešní době nejpoužívanějším typem, co se týče aditivní FDM technologie. Důvodem je její jednoduchost konstrukce a řízení, jelikož používá tři lineární osy X, Y, Z k pohybu nástrojem. Konstrukce pak vypadá jako kvádr, tudíž zařízení bohužel zabírá velký prostor, ale je zde vysoká tuhost a přesnost. [3] [4]



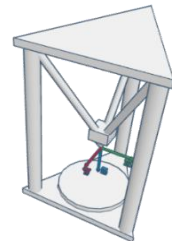
Obr. 2: Kartézská konstrukce



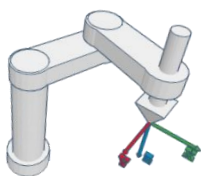
Obr. 3: Polární konstrukce

Dalším typem je konstrukce založená na polární kinematice. Zde jsou obvyklé dva typy uspořádání tohoto robota. Často je pohyb tvořen otočným stolem (jež také funguje jako pracovní plocha) a výsuvným ramenem, které se zároveň posouvá i nahoru a dolů. Nebo je část s otočným stolem nahrazena tím, že se otáčí celý robot sám. Toto uskupení je výhodné v tom, že pracuje právě v polárních souřadnicích, a tak se může hodit pro tvorbu kruhových výrobků.

Delta kinematika je založena na principu tří vertikálních ramen, které pohybují zavěšenou hlavou v centru sférickým pohybem. Výhoda tohoto uskupení je velká rychlost a přesnost pozicování. Často se s tímto robotem můžeme setkat v aplikacích typu „Vezmi a polož“ na dopravnících. Nemá však schopnost manipulovat s velkou zátěží, ale zato se pohybuje velice rychle. [3] [4]



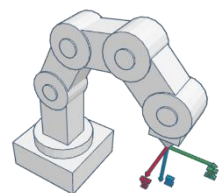
Obr. 4: Delta konstrukce



Obr. 5: Scara konstrukce

Scara „Selective Compliance Articulated Robot Arm“ robot, zde k pohybu nástroje používá dvě rotační a jednu lineární osu. Lineární pohyb reprezentuje osu Z. Výhody konstrukce jsou rychlost pohybů a díky cylindrické pracovní ploše, je také vhodný pro tvorbu kruhových výrobků. V praxi se často používá, stejně jako delta, pro aplikace „Vezmi a polož“ na dopravnících. Oproti deltě má ale výhodu, že snese větší zátěž, bohužel kvůli typu konstrukce jsou u něj pohyby pomalejší.

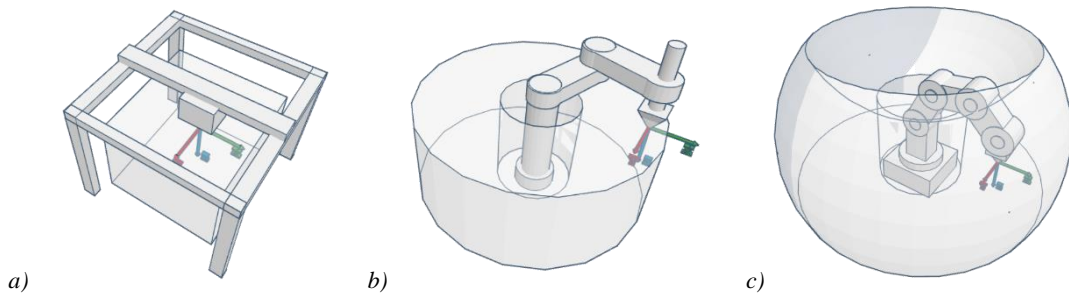
Kloubové rameno je nejsložitější z výše zmíněných, jelikož je schopno zvládat „dle konstrukce“ pohyby i v šesti stupních volnosti, tudíž zvládne dosáhnout velmi složitých pozic. Tomuto typu se budeme věnovat více, jelikož jej používáme pro robotický manipulátor. [3] [4]



Obr. 6: Kloubové rameno

### 2.1.1 Pracovní oblasti robotů

Robotické manipulátory se také často dělí tvarem své pracovní oblasti, kde se pohybuje jejich koncový bod nástroje „P“. Nejtypičtější tvar je kartézský, kde pracovní oblastí je kvádr. Cylindrický, kde pracovní oblastí je válec a sférický, kde pracovní oblast bývá koule, či elipsa. Zde na obrázku (Obr. 7) můžeme vidět pracovní oblasti některých výše popsaných mechanismů: [5]



Obr. 7: Pracovní oblast: a) Kvádr „Kartézská“, b) Cylindrická „SCARA“, c) Sférická „Rameno“

Převody mezi jednotlivými souřadnými systémy [6]:

Cylindrický  $P(r, \theta, z) \rightarrow$  Kartézský  $P(x, y, z)$

$$x = r \cdot \cos(\theta) \quad (1)$$

$$y = r \cdot \sin(\theta) \quad (2)$$

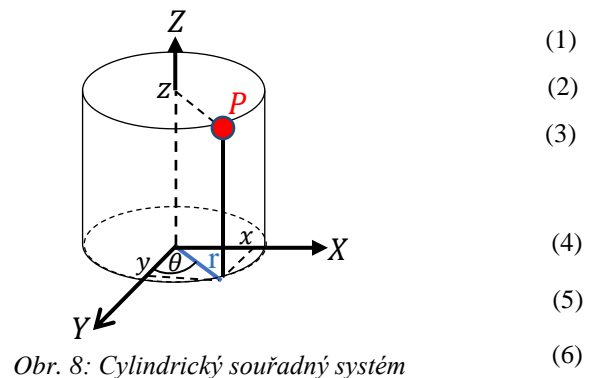
$$z = z \quad (3)$$

Kartézský  $P(x, y, z) \rightarrow$  Cylindrický  $(r, \theta, z)$

$$r = \sqrt{x^2 + y^2} \quad (4)$$

$$\theta = \operatorname{atan}\left(\frac{y}{x}\right) \quad (5)$$

$$z = z \quad (6)$$



Sférický  $P(k, \theta, \gamma) \rightarrow$  Kartézský  $P(x, y, z)$

$$x = k \cdot \cos(\theta) \cdot \sin(\gamma) \quad (7)$$

$$y = k \cdot \sin(\theta) \cdot \sin(\gamma) \quad (8)$$

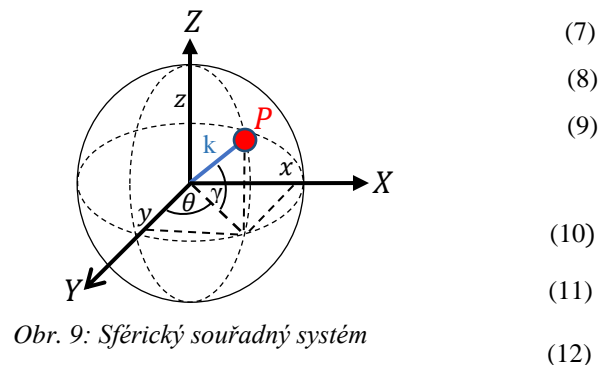
$$z = k \cdot \cos(\gamma) \quad (9)$$

Kartézský  $P(x, y, z) \rightarrow$  Sférický  $P(k, \theta, \gamma)$

$$k = \sqrt{x^2 + y^2 + z^2} \quad (10)$$

$$\theta = \operatorname{atan}\left(\frac{y}{x}\right) \quad (11)$$

$$\gamma = \operatorname{acos}\left(\frac{z}{k}\right) \quad (12)$$



Každá z těchto oblastí má své výhody a nevýhody, velice často se tak můžeme setkat i s jejich kombinacemi, kde například v průmyslu se robotické rameno pohybuje do stran pomocí vozíku na kolejničích, ke kterému je připevněno (Kartézský + Sférický).

## 2.2 Průmyslové robotické manipulátory v aditivní technologii

Zde bývá častější možnost se setkat s kloubovými roboty, kteří se běžně v hobby oblasti kvůli ceně nepoužívají. Hlavním rozdílem od konvenčních konstrukcí FDM technologie je, že robot bývá složen z několika kloubů, s kterými se můžeme volně pohybovat ve více stupních volnosti, což nám dovoluje dosahovat velké pracovní plochy ve všech možných směrech a úhlech. Často je zde tak i jednoduchá možnost výměny nástroje. Nevýhodou však zůstává více komplikovaná a dražší konstrukce s řízením. Ukázkou použití FDM technologie v praxi je například architektura a stavařství [7]. Kde se roboti používají při stavbě domů (Obr. 10), či tvorbě uměleckých děl [1].



Obr. 10: 3D tisk betonu (polární konstrukce) [1]

V automobilovém průmyslu se takto roboti používají pro nanášení vrstvy lepidla, či svařování. Co se týče materiálu, tak existuje velké množství druhů, se kterými roboti pracují, například i tisknutí „navarování“ kovů (Obr. 11). [8]



Obr. 12: 3D tisk termoplastu (kartézská konstrukce) [9]      Obr. 11: 3D tisk kovů (kloubové rameno) [8]

Pro použití robota jako 3D tiskárny se využívá jak běžných průmyslových robotů, tak pro menší operace pak roboti kolaborativní. Vždy se to tedy odvíjí od potřebné velikosti pracovní plochy, a hlavně možnosti prostoru, kam robota usadit. V průmyslu se však také používají velké kartézské tiskárny (Obr. 12) [9] a další typy konstrukcí, které byly popsány v kapitole 2.1.

Ideálním kandidátem na se tak pro nás stalo průmyslové řešení robota sloužícího pro přenos těžkých břemen. V praxi se tento typ nazývá „Paletizační robot“. [10]

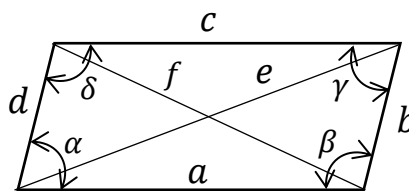


Obr. 13: Paletizační manipulátor [10]

Z obrázku (Obr. 13) jsou hned patrné hlavní výhody tohoto mechanismu. První důležitý fakt je, že jsou motory připevněny k základně robota, čímž se přesune celkové těžiště a můžeme použít větší, silnější motory. Rovnoběžníky nám zde vytvářejí uzavřené kinematické smyčky, které přinášejí výhodou, že nástroj je pokaždé rovnoběžný s pracovní plochou. Pro pohybování v prostoru tedy stačí pouhé tři motory.

### 2.2.1 Paralelogram

V českém překladu „Rovnoběžník“. Jedná se o čtyřúhelník, který má vždy dva protilehlé páry stran se stejnou délkou rovnoběžné a také protější úhly rovnoběžníku jsou shodné. [11]



Obr. 14: Geometrie rovnoběžníku při úhlech  $\alpha = \gamma$  a  $\beta = \delta$  a stranách  $a=c$  a  $b=d$

Úhlopříčky lze vypočítat z:

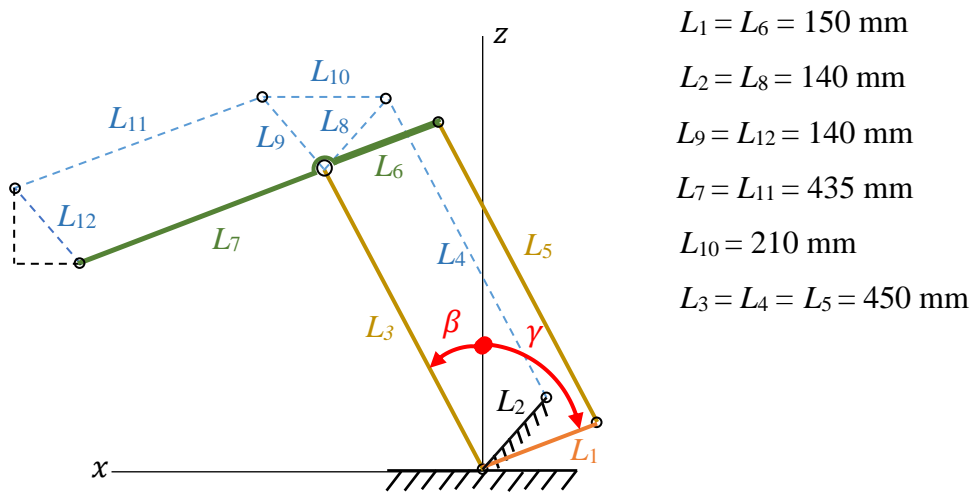
$$e = \sqrt{a^2 + d^2 + 2 \cdot a \cdot d \cdot \cos(\alpha)} \quad (13)$$

$$f = \sqrt{a^2 + d^2 - 2 \cdot a \cdot d \cdot \cos(\alpha)} \quad (14)$$

Platí zde, že úhly  $\alpha = \gamma$  a  $\beta = \delta$  a strany  $a=c$  a  $b=d$ . Úhlopříčka pak rozděluje rovnoběžník na dva shodné trojúhelníky. [11]



Toto tedy bylo aplikováno na návrh konstrukce robota, a byly zvoleny délky ramen:



Obr. 15: Návrh délek ramen paralelogramu

Úhly, s kterými se pomocí motorů pohybuje jsou  $\alpha$ ,  $\beta$ ,  $\gamma$ . Úhel alfa zde není znázorněn, jelikož se jedná o rotaci stolu manipulátoru kolem osy Z. Jak byly jednotlivé díly navrženy lze vidět v (Obr. 16), nebo v další kapitole 3.2, zabývající se konstrukcí. Vzhled ramen už je pak zvolen dle osobního vkusu a se snahou co nejvíce ušetřit na hmotnosti a tloušťce daných částí. [12]



Obr. 16: Konstrukční návrh

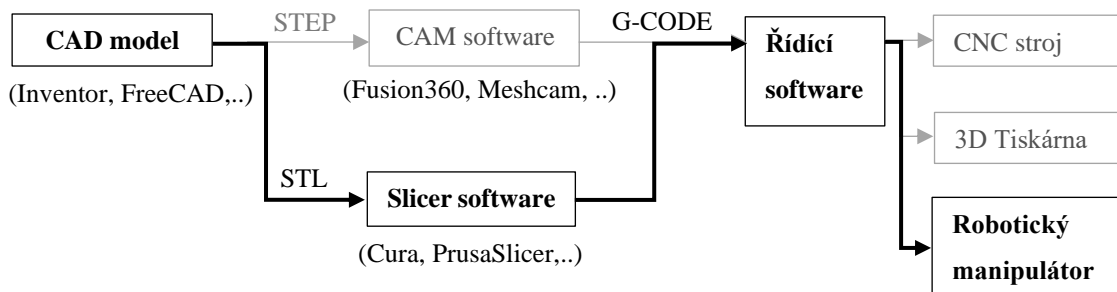
Délky ramen byly navrženy tak, aby přibližně dosahovaly pracovní oblasti jednoho metru. V některých dílech byly vyřezány odlehčující otvory pro snížení váhy celkové konstrukce, ale v takové míře, aby to neohrozilo tuhost a celistvost rámu.

## 2.3 Open-source softwary pro řízení manipulátorů a 3D tiskáren

V dnešní době existuje spousta možností „open source“ firmwaru na trhu, jež lze aplikovat pro řízení různorodých zařízení. Zde se specificky zaměřujeme na takové, které lze použít pro 3D tiskárny, jelikož přesně splňují požadavky pro využití robotického manipulátoru. Důležité tady také bylo, aby daný firmware spadl pod licenci, která dovoluje jeho modifikaci tak, aby zvládal pracovat s vlastní navrženou kinematikou.

Byla by tu samozřejmě možnost napsat celý kód sám, v této práci ale šlo více o to se seznámit a ukázat možnosti použití dostupných řešení, která jsou již odzkoušena dlouhodobým vývojem a používáním v praxi. Toto řízení pak stačí modifikovat pro náš případ kinematiky robotického manipulátoru. [2]

Na obrázku lze vidět postupný proces zpracování povelu pro manipulátor. Postup je takový, že výrobek (3D tisk) / obrázek (plotter) namodelujeme v dostupném CAD „Computer-aided design“ softwaru, převedeme ho do formy G-Kódu (pomocí veřejně dostupného softwaru) a ten pošleme do řídicí desky, která dle instrukcí pohybuje s akčními členy stroje. [2]



Obr. 17: Postup při tvoření požadavku pro manipulátor

V našem případě používáme pro modelování Autodesk Inventor (v kterém je z modelu vygenerován STL „Standart Triangle Language“ soubor) a pro převod do G-Kódu je využít PrusaSlicer, který daný model rozseká na jednotlivé vrstvy. Existuje však vícero softwaru, ať už placených, nebo volně dostupných, které lze použít.

Důležitou volbou pak už zůstává řídicí software. Představme si tedy několik v dnešní době používaných řešení, která všechna spadají pod licenci GPL „General Public License“, tedy zdrojový kód je volně dostupný a editovatelný. Jedinou podmínkou této licence je, že musí být upravený kód volně dostupný ke stažení.

### 2.3.1 RepRap

Jedná se o projekt z roku 2004, celým názvem „Replicating rapid prototyper“, s cílem vytvořit „open source“ tiskárnu, která dokáže sama sebe zreplikovat, tedy vyrábět si vlastní náhradní díly. Kód je psán v C++ jazyce a program je z vnějšku řízen pomocí G-kódu, jež spadá pod normu NIST RS274NGC. Původně byl tento software navržen pro 8bitové mikroprocesory, ale postupem času a vývoje, se RepRap přesunul na 32bitové mikroprocesory osazené na vývojových deskách s názvem „Duet“. Jedná se o desky kompatibilní s Arduino Due. Tohle bylo kladeno často jako velká nevýhoda pro použití RepRapu, jelikož pro běh firmwaru byla právě potřeba tato výkonná řídicí jednotka. Dnes už však nabízí i podporu pro vícero vývojových desek. Velkou výhodou, proč se v dnešní době lidé obracejí na tento firmware je jeho jednoduchá konfigurace a nahrání do řídicí desky, jelikož je vytvořen webový portál, kde se pouze zadají parametry a on na základě těchto dat vygeneruje firmware pro tiskárnu. [13]



Obr. 18: RepRap logo [13]

### 2.3.2 Marlin

Byl vytvořen v roce 2011 jako další z možností řídicího softwaru pro 3D tiskárny. Vyvinul se z GRBL a Sprinter firmwaru. V dnešní době je hojně využíván i světovými společnostmi jako je Průša Research, Creality3D nebo Ultimaker. Tento firmware také kromě 3D tisku dokáže řídit CNC, laser, či plotter. Původně byl postaven na 8bitových mikrokontrolérech Atmel AVR, jež se používají ve známé vývojové desce Arduino, s postupem času a vývojem se však přesunul na 32bitové desky a obrovsky tak rozšířil možnosti použití. Kód je psán v C++. Oproti RepRapu jde tedy použít i na méně výkonné desky, ale jeho často zmiňovanou nevýhodou je o něco komplikovanější konfigurace, tedy úprava kódu a následné nahrání na desku. [14]



Obr. 19: Marlin logo [14]

### 2.3.3 Klipper

Objevil se roku 2016 a stal se unikátním v tom, že k němu bylo zapotřebí oproti ostatním firmwarům dvě desky. Byl navržen tak, že bude používat běžnou vývojovou desku od 3D tiskárny, na které budou připojeny všechny periferie a veškeré výpočty budou prováděny na RaspberryPi, což je malý jednodeskový počítač. V době, kdy autor



Obr. 20: Klipper logo [15]

firmwaru přišel s tímto nápadem, nebylo typické řídit 3D tiskárnu na 32bitových deskách (kromě Due pro RepRap), a proto se pyšnil velkým výpočetním výkonem, kde klasické mikrokontroléry, jako je například Arduino, nemohly konkurovat výkonu RaspberryPi. Velký rozdíl je také, že velká část kódu je napsána v Pythonu, což se může jevit uživatelsky přívětivější než C++. Pokud chceme například změnit konfiguraci tiskárny, nemusíme používat tak složitý proces jako je bootování u Marlinu, ale stačí pouze upravit textový soubor skrze RaspberryPi. Stejným způsobem také lze přidávat vlastní makra, „G-kódy“, bez zásahu do zdrojového kódu. Jeho nevýhodou se tedy stává primárně pořizovací cena, jelikož je nutno ještě navíc koupit k řídicí desce i RaspberryPi, na kterém pak probíhají výpočty a máme zde ovládání ve formě webového rozhraní Mainsail nebo Fluidd (podobné, jako je Octoprint, dnes asi nejoblíbenější open source aplikace nabízející webové rozhraní pro vzdálené ovládání 3D tiskárny). [15]

### 2.3.4 GRBL

Byl vyvinut v roce 2009 jako řídicí firmware pro řízení pohybu na mikrokontrolérech ATmega328, což jsou běžná Arduina. To z něj dělá velmi levnou a dostupnou volbu. Lze ho použít pro ovládání CNC „Computed-Aided Manufacturing“, frézek, laserů, či 3D tiskáren. Umožňuje také řízení z počítače přes paralelní port. Je napsán v C++. Podporuje velkou škálu G-kódů a ovládání v reálném čase. Tento firmware se stal velmi oblíbený pro tvorbu vlastních domácích CNC. Má podporu ovládání tří os, avšak práce s rotačními osami zde zatím není oficiálně nabízena. Plně podporuje lineární i kruhovou interpolaci. S příchodem výkonnějších desek a dosažením limitů 8bitové verze na Arduinu vznikla verze GRBL pro ESP32 či STM32, aby tak mohl konkurovat dnešním 32bitovým deskám. [16]



Obr. 21: GRBL logo [16]

### 2.3.5 LinuxCNC

Jedná se o open-source software běžící na operačním systému Linux. Původně se skrýval pod názvem EMC (Enhanced Machine Controller) a byl vyvinut v roce 1993 pro využití na frézovacích strojích. V roce 2003 se však část vývojářů odtrhla. Pod novým názvem EMC2 (později LinuxCNC) pak začala svůj firmware šířit pod volně dostupnou licenci. V průběhu času se rozrostl až do možnosti



Obr. 22: LinuxCNC logo [17]

kontinuálního řízení 9 os s podporou pro lasery, roboty, 3D tisk a další. Jedná se o velice propracovaný řídicí systém, který však běží pouze na Linuxu. To se může jevit jako nevýhoda, jelikož je tak potřeba počítač s operačním systémem. LinuxCNC také přijímá G-kód příkazy na vstupu, má obrovskou podporu vnějších hardwarových zařízení. Prakticky se dá říct, že může řídit běžné víceosé CNC zařízení (obsahuje funkce jako např. řezání závitů). Dokáže tedy kontrolovat nejen krokové motory, ale i zpětnovazebně řízená serva. Pro propojení s elektronikou se zde používá vývojová deska s paralelním portem. Kód je psán v C++, C a Pythonu. [17]

### 2.3.6 Shrnutí

Dá se říct, že výše uvedené firmwary se dají použít jak pro 3D tisk, plotter, laser, nebo CNC. Všechny tyto zmíněné „open source“ systémy se z vnějšku ovládají obdobným způsobem, a to skrze G-Kód příkazy. Hlavní rozdíl mezi nimi je tedy především použitý hardware a náročnost, co se týče upravování kódu, či jen konfigurace pro jiný typ zařízení. O něco menší rozdíl je pak ve funkcionalitě, jelikož všechny dnes nabízí širokou škálu funkcí. Jde pak tedy o to se rozhodnout, co uživatel se zařízením zamýšlí. Pro pouhé řízení 3D tiskárny, tak volba GRBL, či LinuxCNC není úplně ideální, i když tuto možnost má v nabídce. Oba tyto firmwary byly totiž primárně vyvíjeny pro CNC a v tomto směru jsou také především komunitou využívány. Pokud však chceme stavět CNC, tak zase nastává otázka: bude nám stačit řízení ve třech osách? Pokud ano, GRBL je jasná volba, pokud ne, budeme muset více zainvestovat a použít LinuxCNC.

Co se týče porovnání jednotlivých firmwarů pro 3D tiskárny (RepRap, Marlin, Klipper), tak zde byla snaha v krátkých popisech vystihnout možné rozdíly, ale dá se říct, že v dnešních verzích těchto firmwaru už běžný uživatel nepozná rozdíl. Každý z nich má totiž své speciální funkcionality, které uživatel ani nepoužívá a s postupným vývojem se spolu vyrovnávají. Zásadním rozdílem tak zůstává použitý hardware. Například takový Marlin, pro běžnou práci s ním stačí jedna řídicí deska, ale pokud chceme tisknout pomocí webového rozhraní online (Octoprint), tak je nutno pořídit RaspberryPi a tu se dostáváme k stejné pořizovací hodnotě, jako kdybychom chtěli používat Klipper. Jaký firmware nejlépe použít se tedy nedá úplně dobře říct, každý z nich má své benefity a slabých stránek už tolik v dnešní době není. Pokud je tedy člověk začátečník, tak je dobré volit takový firmware, který má za sebou velkou komunitu, aby v případě problému našel snadné řešení. Já jsem se rozhodl pro svůj robotický manipulátor použít Marlin.

## 2.4 G-CODE

Jedná se o programovací jazyk používaný pro řízení strojů a zařízení. Tento jazyk se skládá z jednotlivých instrukcí, které popisují stroji, co má přesně dělat, ať už se jedná o polohovací povely, vypnutí, nebo třeba změnu rychlosti. [14]

### 2.4.1 Výběr některých používaných příkazů

- G0, G1 – Tento příkaz slouží k rychlému, či lineárnímu pohybu [14]

```
G0/G1 [X<pos>] [Y<pos>] [Z<pos>] [E<pos>] [F<rate>] [S<power>]
G1 X12; Lineární pohyb po ose X na pozici 12mm
G0 X5 Y12 F1500; Rychlý pohyb na pozici X5,Y12 s rychlostí 1500mm/min
G1 X90.6 Y13.8 E22.4; Lineární pohyb a vytlačení extrud. o 22.4mm
```

*Kód 1: Příkaz pro pozicování [14]*

- G92 – Příkaz pro přepsání aktuálních hodnot pozic

```
G92 [X<pos>] [Y<pos>] [Z<pos>] [E<pos>]
G92 X0 Y0 Z0 E0; Vynulování pozic motorů robota
```

*Kód 2: Příkaz pro přepsání pozice [14]*

- G90, G91 – Absolutní a relativní pozicování [14]

```
G90; Absolutní - posun si pamatuje předchozí pozici a navazuje
G91; Relativní - posun si nepamatuje předchozí pozici a jede od 0
```

*Kód 3: Příkaz pro nastavení pozicování [14]*

- M114 – Příkaz pro report aktuální pozice robota [14]

```
M114; Vypíše do sériové komunikace pozici nástroje
```

*Kód 4: Příkaz pro vypsání pozice [14]*

- M104 – Příkaz pro nastavení teploty [14]

```
M104 [B<temp>] [F<flag>] [I<index>] [S<temp>] [T<index>]
M104 S210; Nastaví teplotu hotendu na 210°C
```

*Kód 5: Příkaz pro nastavení teploty [14]*

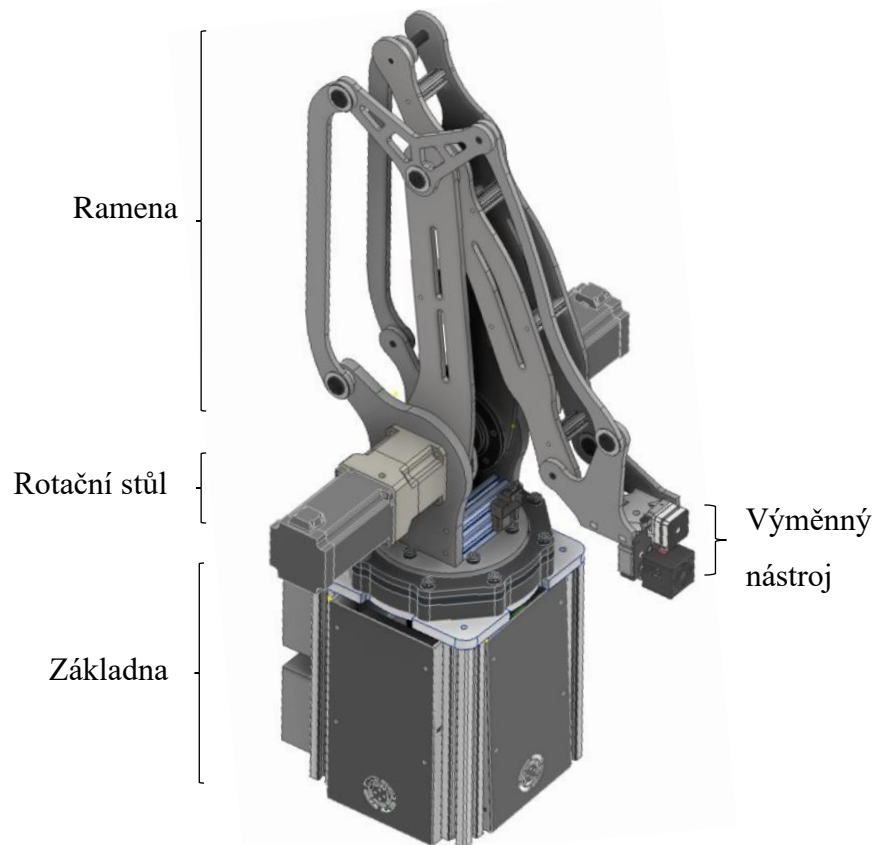
- M280 – Nastavení pozice serva [14]

```
M280 [P<index>] [S<pos>]
M280 P0 S180; Natočí servo s indexem 0 na pozici 180°
```

*Kód 6: Příkaz pro servo [14]*

### 3. Návrh a konstrukce robotického manipulátoru

Samotná konstrukce robota se dá rozložit do čtyř částí: Základna, Rotační stůl, Ramena a Výměnný nástroj.



Obr. 23. Robotický manipulátor

Při návrhu konstrukce byl jako první načrtnut koncept celého manipulátoru a poté v počítači prověřena funkčnost geometrie stroje. Jelikož se jedná o paralelogram, tak bylo nutno dodržet určité proporce délek ramen. Ty byly zvoleny tak, aby pracovní plocha robota dosahovala od základny k nástroji až do přibližné vzdálenosti jednoho metru. Rozhodlo se tedy o zvolených délkách jednotlivých dílů robota viz **kapitola 2.2.1**.

Další volbou byl materiál, z kterého bude robot postaven. Bylo potřeba něco pevného a lehkého, aby nebyl robot zbytečně těžký, což by způsobilo nutnost pořídit silnější motory či převodovky. Volbou se tedy stal dural.

Tloušťka jednotlivých dílů byla volena kritériem, kdy každý díl musel splňovat vysokou pevnost a tuhost, ale nesměl být zbytečně tlustý. Tloušťka materiálu se totiž projevuje nejen na nadbytečné váze robota, ale i cena materiálu roste s každým milimetrem tloušťky navíc, a to v hodnotách tisíce korun.

Tloušťky jednotlivých dílů byly navrženy a počítány jako běžné nosníky, kde ze znalosti délky ramene, modulu pružnosti duralu v tahu ( $E = 72 \cdot 10^9 Pa$ ) a kvadratického průřezu dílu, byl počítán maximální průhyb profilu při očekávaném zatížení. [18]

Po provedených výpočtech se pro jednodušší a levnější náklady na výrobu rozhodlo sjednotit jednotlivé díly a nechat je vyrobit z dvou typů tlouštěk plechu. Méně namáhané díly se vyrobily z tloušťky 8 mm a více zatížené pak z 15 mm plechu. Celý model robota se pak v Inventoru podrobil zátěžové analýze, kde program vykreslil možná problematická místa konstrukce.

Jednotlivá ramena, jsou pak navzájem propojena konstrukčními hliníkovými profily, aby tak utvořila celkový tuhý rám robota.

Co se týče volby ložisek, tak zde volba padla převážně na použití samomazných kluzných pouzder, jelikož tento typ při dobrém uložení vykazuje malou vůli s vysokou tuhostí spoje. V oblasti kloubů ramen také dochází hlavně k radiálnímu zatížení, tudíž se nemuselo tolik řešit axiální zatížení a pod hlavu šroubu je jen vložena axiální kluzná podložka, kterou se zajistí vůle po dotažení šroubu viz **kapitola 3.4**.

Pro rotační stůl však bylo použito křížové ložisko, jelikož zde docházelo jak k radiálnímu, tak axiálnímu zatížení. Převodovka, která je zde použita, je schopna sama o sobě unést určité zatížení, ale z důvodu její životnosti ji není příliš dobré zatěžovat. Volba převodovky a spojky je pak rozepsána v **kapitole 3.1** níže.

Základna je složena z duralové desky o tloušťce 15 mm, která je zespolu podepřena čtyřmi hliníkovými konstrukčními profily 40x40. V této základně je pak umístěna veškerá elektronika, drivery a motor s převodovkou pro rotační stůl. Díly pro uchycení elektroniky a jednotlivé kryty jsou vytištěny z plastu na 3D tiskárně.

Rotační stůl je celý vyroben z ocelových dílů, které spolu vytvářejí „domek“ pro křížové ložisko, jež ukotvuje celého robota a umožňuje mu se otáčet. Tento domek je připevněn k duralové desce základny.

Ramena jsou vyřezána z duralu o tloušťce 8 mm na laseru a vzájemně v kloubech propojená kluznými pouzdry s vyrobenými lícovanými šrouby na míru.

Výměnné nástroje s hlavicí a mechanismus pro jejich uchycení byly vymodelovány a postupným prototypováním na 3D tiskárně uvedeny ve funkční verzi.



### 3.1 Volba převodovky a spojky

Při stavbě robotického manipulátoru, byla snaha se zaměřit více na typy převodovek používaných právě v robotických aplikacích v praxi, a to při udržení nízkých nákladů za koupi. Bylo také nutno posoudit jednotlivé výhody a nevýhody každého typu. Možnosti pak byly takové:

Řemenové převody mají velkou výhodu, že pro přenos síly je použit pružný element „řemen“ propojující dvě řemenice, což nám dává možnost tichého a poměrně přesného pozicování (při správném napnutí řemenů). Největší výhodou tohoto provedení je jednoduchá údržba a cena. Nevýhodou je zde, že se dá dobře využít jen pro nízké převodové poměry, tzn. do 5:1 maximálně. Do této kategorie bych zařadil i převody s řetězem, které oproti řemenovému dosahují možnosti větší zátěže. Na průmyslových robotech se tyto typy často vyskytují a jsou právě skryty uvnitř krytů ramen. [18] [19]

Šnekové převody fungují na principu šnekového soukolí, které slouží k přenosu síly mezi dvěma kolmými hřídelemi. Hlavní výhodou šneku je samosvornost a velký točivý moment. Avšak rychlost a převážně přesnost pozicování už zde nejsou tak dobré. Jejich cenová dostupnost je však velmi dobrá. Důvodem, proč v tomto případě nebyl šnek použit, je velká vůle hřídele a fakt, že prostorově svým uložením zabírá více místa než ostatní zde zmíněné. [18] [19]

Cykloidní převodovka je vysoce přesná a umožňuje přenos velkých záběrných momentů. Je založena na cykloidním pohybu a dobře zvládá i rázové změny zatížení s vibracemi, což jsou v robotice časté jevy. Její obrovskou nevýhodou je však náročná konstrukce, a tudíž i její pořizovací cena. Tato možnost se zdála jako perfektní, avšak cenová dostupnost rozhodla v neprospěch této volby. [18] [19]


Planetová převodovka, známá jako epicykloidní, je založena na principu několika ozubených kol obíhajících kolem centrálního kola. Planetová kola jsou připevněna k nosiči a zabírají jak s centrálním kolem, tak vnějším věncem. Jedná se o velice účinnou konstrukci, která dosahuje vysoké přesnosti. Přináší nám také možnost vysokých převodových poměrů v malé velikosti. Tato převodovka se již dá sehnat za dobré peníze, a to i při dodržení požadavku na přenášený moment, přesnost a kompaktnost konstrukce.

Volba spojky pro robota se pak odvíjela od zvolené převodovky. Zde byla snaha o malou prostorovou náročnost a možnost jednoduchého přichycení ke konstrukci. [18]

### 3.1.1 Zvolená planetová převodovka

Byla zvolena čelní planetová převodovka s převodem 15:1 pro rotační stůl a 10:1 pro ramena. Tato převodovka splňuje dobře kritérium „cena/výkon“, jelikož umožňuje poměrně velkou zátěž a má nízkou vůli volného pohybu, tedy i vysokou přesnost při pozicování. Možností přenosu momentu z toho typu převodovky bylo víc, ale z důvodu pořizovací ceny bylo rozhodnuto pro typ zakončený hřídelí s perem. Tyto převodovky se vyrábějí i ve variantě se zakončením s přírubou, která by šla rovnou napojit na rameno robota a nebylo by tak třeba řešit spojku, avšak cena této převodovky by vyšla až třikrát draž než typ s perem. Také šla koupit převodovka v provedení se zatočením o 90°, takže by robot nezabíral spolu s motory tak velkou šířku, avšak za cenu větší vůle a pořizovací částky. Tento typ převodovky se vyrábí ve dvou velikostech, a to s délkou 154,5 mm do poměru 10:1 a při použití vyššího převodového poměru pak délka roste na 189 mm, což se v domečku rotačního stolu s elektronikou krásně schová. Hřídel motoru je v převodovce uchycen svěrným pouzdrem a dotažen šroubem (Obr. 24). [20]

Tab. 1: Volba převodovky (hodnoty převzaty od prodejce [20])

Model:	PLG90-L-S2-P0	
Typ:	Planetová převodovka	
Převodový poměr	10:1	15:1
Vůle „backlash“	≤ 3 arcmin	≤ 5 arcmin
Max. výstupní moment	190 Nm	280 Nm
Instantní brzdící moment	285 Nm	420 Nm
Maximální radiální / axiální zatížení	1780 N /1500 N	
Torzí tuhost	12 Nm/arcmin	
Účinnost při plné zátěži	≥ 97 %	≥ 94 %
Váha	3,7 kg	4,1 kg
Průměr hřídele motoru „vstup“	14 mm	
Průměr hřídele převodovky „výstup“	22 mm	
Obrázek:		

Obr. 24: Planetová převodovka [20]

### 3.1.2 Zvolená spojka

Jako nejlepší možná volba, co se týče ceny, parametrů a rozměrů, se jevílo upínací kuželové pouzdro s adaptérem. Možností pro adaptér zde bylo více: mezikroužek, navařovací náboj či ve formě příruby s otvory. Výběh padl na přírubu s šesti otvory, kterými se šlo připevnit k jednotlivým ramenům robota. Díry bylo sice nutno lehce rozšířit, jelikož byl adaptér spojen s ramenem šrouby M8, ale to byl minimální problém.

Tab. 2: Volba spojky pro manipulátor (hodnoty převzaty od prodejce [21])

<b>Model:</b>	<b>Kuželové upínací pouzdro TB (TaperBush)</b>	<b>Našroubovácí adaptér s nábojem pro pouzdro</b>
<b>Typ:</b>	TB 1210-22	SM 1210-120
<b>Max Krouticí moment</b>	Bez pera 105 Nm S perem 407 Nm	Materiál: GG (blíže specifikováno u výrobce není)
<b>Svěrná síla</b>	9800 N	
<b>Max. výstupní moment</b>	190 Nm	
<b>Instantní brzdící moment</b>	285 Nm	
<b>Hmotnost</b>	0.28 kg	1,6 kg
<b>Obrázek:</b>	 <p>Obr. 25: Kuželové upínací pouzdro [21]</p>	 <p>Obr. 26: Adaptér pro pouzdro [21]</p>

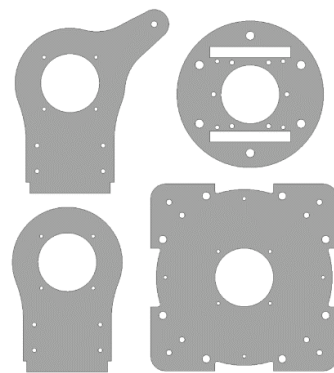
Lze zde využít jak velké svěrné síly, tak přesné umístění ramene robota pomocí upnutí přes dotažení kuželového tvaru spojky a také využití pera z hřídele převodovky. Spoj sice není nikterak pružný (není tedy dělaný pro dynamické změny momentu), ale uzpůsobením rozběhových a doběhových charakteristik robota se z větší části dokážeme vyhnout těmto nežádoucím dynamickým zátěžovým jevům [18] [21].

### 3.2 Zakázkově vyrobené díly

Horní část robota se skládá z 15 částí, které jsou všechny vyrobeny z duralu. Máme zde dvě skupiny, jedna má tloušťku 8 mm a druhá 15 mm. Tento materiál byl jasným kompromisem mezi pevností a lehkostí konstrukce. Zásadní zde byla celková tuhost konstrukce (pro přesné pohyby s nástrojem v hlavě robota) a lehkost (abychom nemuseli zbytečně předdimenzovat motory a převodovku). Díly z duralu byly vypáleny na laseru.

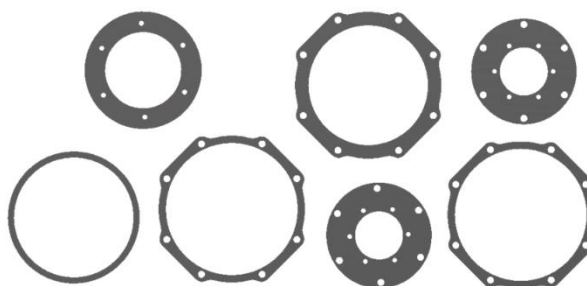


Obr. 28: Díly z duralu 8 mm



Obr. 27: Díly z duralu 15 mm

Další skupina dílů je zhotovena z oceli různých tlouštěk. Tyto tloušťky byly zvoleny tak, aby spolu tvořily domek pro křížové ložisko rotačního stolu, vše bylo řezáno vodním paprskem. Bylo zde nutno jednotlivé otvory pro šrouby rozšířit, jelikož průměr dřívku lícovaných šroubů byl o trochu větší než otvory vyřezané vodním paprskem a chtěl jsem zde docílit přesného uložení. Poté se celý domek složil a vcelku přivrtal k duralové desce základny.



Obr. 29: Díly z oceli

Jako další zakoupené díly byly hliníkové profily velikosti 40x40 pro nohy stolu, nebo profily 20x20 a 30x30 sloužící jako součást rámu robota. Samozřejmostí je pak velké množství šroubů, podložek a ložisek. V tenkých částech duralu o 8 mm tloušťce jsem volil závit s jemnou roztečí, pro lepší uchycení šroubu v rameni. Nechal jsem si také vyrobit některé lícované šrouby atypických délek.

### 3.3 Návrh a konstrukce rotačního stolu

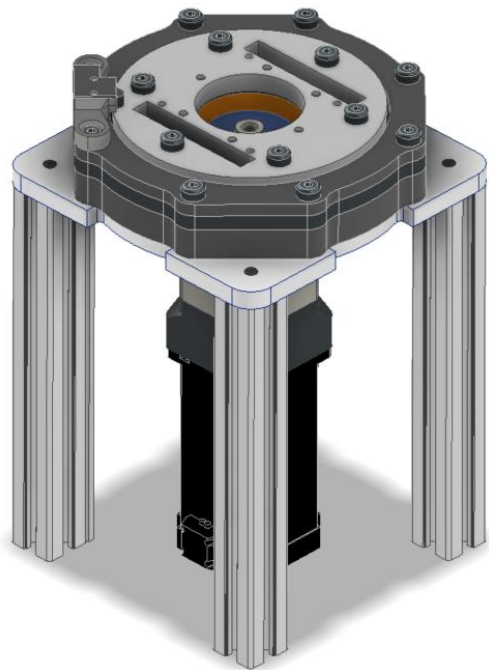
Při návrhu rotačního stolu byly dány požadavky:

- Stůl musí snést rotační pohyb robota, a tedy zvládat zatížení jak axiální, tak radiální.
- Volba velikosti, aby zabíral co nejméně prostoru.
- Aby šla do rotačního stolu dobře zakombinovat zvolená spojka s převodovkou.

Jako první pokus bylo robota umístit na velké kuličkové axiální ložisko. Toto řešení sice bylo z části funkční, ale znamenalo, že veškerá radiální síla byla opřena do hřídele převodovky. Což způsobovalo velké riziko, které by mohlo vést k poškození převodovky.

Poté byl vyzkoušen návrh kombinace různých radiálních a axiálních ložisek separátně, to už ale čistě ve formě modelu v Inventuru. Jako axiální se jevila nejlepší volba jehlová ložiska a jako radiální pak klasická kuličková. Jen bohužel kombinace těchto ložisek v jednom domečku a při použití adaptéru spojky netvořily dobrý mechanismus a zabíralo to obrovský prostor. Zda by to pak dobře fungovalo i v realitě, to si můžeme jen představit.

Jako poslední návrh, který byl uskutečněn i fyzicky, byla inspirace v jeřábech a jejich uchycení kabin. Po důkladné rešerši v katalozích s křížovými ložisky [22] bylo zvoleno takové, které svými vlastnostmi a rozměry pasuje dobře pro použití na našem manipulátoru. Byla zde možnost koupě tohoto ložiska už v předem připraveném domku s otvory na šrouby, ale to úplně dobře nepasovalo na celkovou konstrukci a cena této možnosti byla také o dost vyšší. Domek pro ložisko tedy bylo nutno navrhnout jako samostatnou část. Tento domek pak byl připevněn na duralovou desku o tloušťce 15 mm postavené na čtyřech nohách z hliníkových konstrukčních profilů 40x40, k této desce je také připevněn ze spodní strany motor s převodovkou.




Obr. 30: Rotační stůl

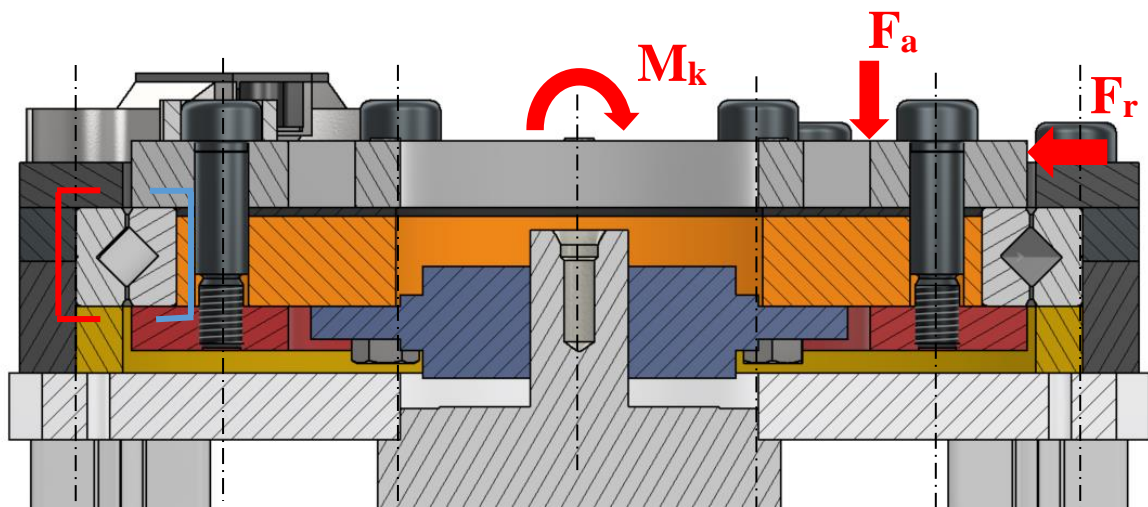
### 3.3.1 Volba ložiska a jeho uložení

Po procházení katalogů a konzultací s firmou, která tato ložiska nabízí, jsem zvolil:

Tab. 3: Volba ložiska pro rotační stůl (hodnoty převzaty od prodejce [22])

<b>Model:</b>	<b>SX011836-A [22]</b>
<b>Typ:</b>	Křížové válečkové ložisko
<b>Základní dynamická únostnost, axiální <math>C_a</math></b>	98 000 N
<b>Základní statická únostnost, axiální <math>C_{0a}</math></b>	360 000 N
<b>Základní dynamická únostnost, radiální <math>C_r</math></b>	70 000 N
<b>Základní statická únostnost, radiální <math>C_{0r}</math></b>	177 000 N
<b>Hmotnost</b>	2.2 kg
<b>Obrázek:</b>	 <p>Obr. 31: Křížové ložisko [22]</p>

Zde je zobrazen rotační stůl v řezu s ukázkou namáhání při otáčení robota:



Obr. 32: Řez rotačním stolem ( $F_r$  – radiální síla,  $F_a$  – axiální síla,  $M_k$  – kroučící moment)

Jelikož zhotovit takový domek z celistvých kusů by bylo více finančně náročné, z důvodu vytížení CNC stroje a obrábění materiálu, tak byl domek vyřezán vodním paprskem z jednotlivých plátů a ty poté spolu sešroubovány. Červená čára v obrázku řezu ukazuje na vnější uchycení ložiska skrze 4 díly, jež svírají vnější kruh ložiska a pevně jej přichytávají ke stolu. Modrá čára pak ukazuje, jak je přichycena vnitřní část ložiska, tedy i celý robot ke křížovému ložisku skrze další čtyři díly, jež svírají vnitřní kruh ložiska. Oranžový díl zde byl schválně zvolen o něco nižší než je výška ložiska, která činí 22 mm.

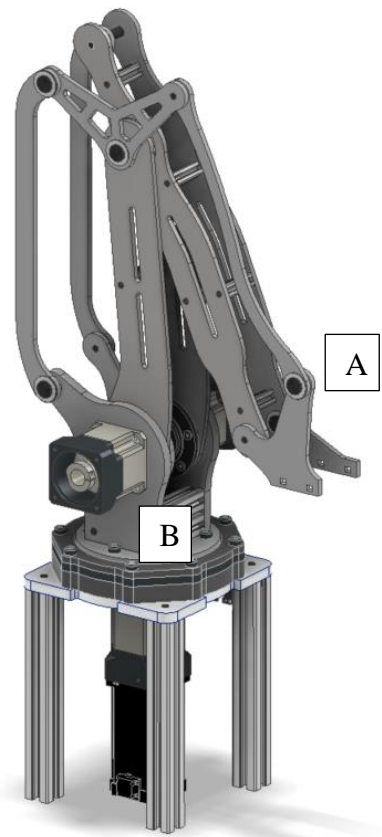
Dostupné plechy pro výrobu byly totiž pouze s tloušťkou 20 mm a 25 mm. Rozhodl jsem se tak ušetřit na obrábění tlustšího plechu a zvolil plech 20 mm. Vzniklou mezeru jsem vyplnil gumovou vložkou o stejném průměru jako oranžový díl s tloušťkou 2,5 mm. Tímto jsem vyplnil vzniklou vůli a zajistil dobré složení všech dílů dohromady.

### 3.4 Návrh a konstrukce horní části robota

Při návrhu horní části robota byly dány požadavky:

- Spodní část konstrukce musí unést zátěž připevněných motorů s převodovkami.
- Snaha o co největší tuhost celé konstrukce, aby tak byla možnost docílit přesných pohybů s koncovým nástrojem.

Vrchní část robota se celá skládá z duralu, abychom zachovali pevnost a lehkost konstrukce. Jednotlivé délky a rozměry byly navrženy tak, aby byla splněna potřebná geometrie paralelogramu i s poměrně velkou pracovní plochou. Spodní díly, na které jsou připevněny převodovky, jsou zvoleny z tloušťky 15 mm, jelikož se do nich opírá z jedné strany motor s převodovkou při společné váze cca 8 kg a délce 230 mm. Důležitým prvkem celé konstrukce při pohybu je její tuhost, ta byla zlepšena vyztužením celé konstrukce hliníkovými profily, samozřejmě by byla lepší volba místo těchto dutých profilů zvolit plné, ale zase by nám tak přibyla hmotnost celé konstrukce. Při výrobě byl kladen důraz na přesnost otvorů, pro co nejpřesnější uložení ložisek, tedy kvalitě kloubových spojů.



Jakým způsobem zvolit kloubové spoje jednotlivých ramen, aby správně fungovala geometrie a byla zde minimální vůle, to se ze začátku jevílo jako problém. První nápad byla klasická kombinace kuličkových ložisek axiálních s přírubou a malých radiálních, které by spolu se šroubem spojovala obě ramena. Nakonec se však od tohoto řešení upustilo, jelikož tato kombinace zabírala zbytečně velký prostor a oproti kluznému ložisku nepřinášela žádné benefity, naopak



kluzné zde nabízelo větší tuhost spoje, a to ještě navíc při lepší pořizovací ceně. Byla tak zvolena kombinace kluzného pouzdra s lícovaným šroubem. Axiální vůle pod hlavou šroubu pak byla vyplněna kluznou podložkou (Obr. 36). Těchto lícovaných šroubů se závitem M10x1 bylo vyrobeno celkem deset, z toho 1x pro délku 76 mm, 1x 42,5 mm, 1x 33,5 mm a 7x 20,5 mm. Z toho délka závitu vždy činila 8 mm.



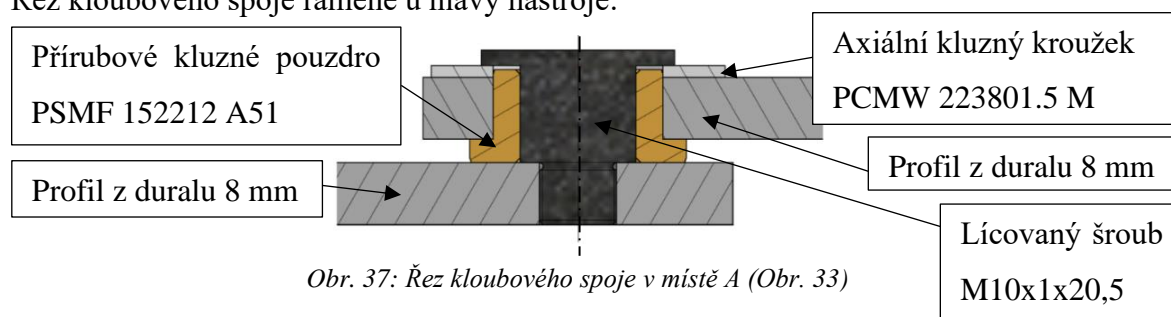
Obr. 34: Lícovaný M10x1x42,5

### 3.4.1 Volba ložisek a uložení

Tab. 4: Volba ložisek pro spoje (hodnoty převzaty od prodejce [23])

Model:	PSMF 152212 A51 [23]	PCMW 223801.5 M [23]
Typ:	Přírubové kluzné pouzdro	Axiální kluzný kroužek
Materiál:	Slinutý bronz, samomazné	Kompozit
Součinitel měrného dynam. zatížení K	10 N/mm <sup>2</sup>	120 N/mm <sup>2</sup>
Součinitel měrného static. zatížení K <sub>0</sub>	20 N/mm <sup>2</sup>	250 N/mm <sup>2</sup>
Přípustná kluzná rychlost	0.25 až 5 m/s	0 až 2.5 m/s
Součinitel tření μ	0.05 až 0.1	0.02 až 0.2
Obrázek:	 Obr. 35: Malé kluzné pouzdro [23]	 Obr. 36: Malý kluzný kroužek [23]

Řez kloubového spoje ramene u hlavy nástroje:



Další důležitý spoj, jenž bylo nutno řešit, bylo ukotvení velkého ramene. Zde se v praxi často sahá po křížovém ložisku. To do konstrukce přináší jak kvalitní tuhost spoje, tak zajištění pohybů v axiálním i radiálním směru, cena toho ložiska je však vysoká.

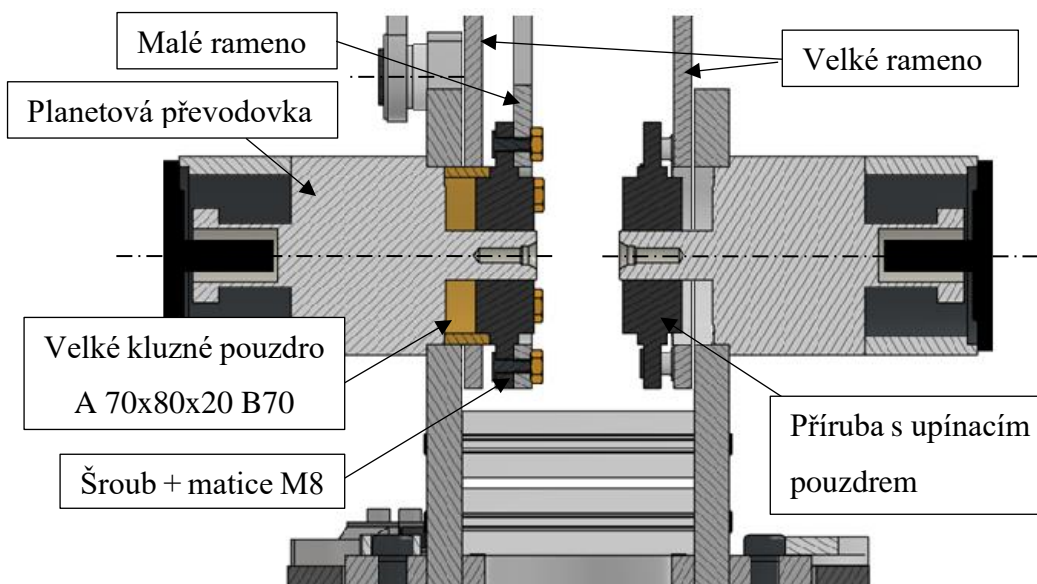


Byl zde tak proveden experiment s kluzným pouzdem. To vyšlo cenově mnohem lépe, jen bylo nutno jeho vnější průměr lehce osoustružit pro dosažení optimální vůle, aby se po něm mohlo rameno volně pohybovat. Radiální zátěž ramen se tak přenáší na kluzné pouzdro a díky pevnému spojení s převodovkou se nemusí v tomto místě (Obr. 39) řešit axiální posuvy (ty zajišťuje převodovka se spojkou).

Tab. 5: Volba ložiska pro velké rameno (hodnoty převzaty od prodejce [23])

<b>Model:</b>	<b>A 70x80x20 B70 [23]</b>
<b>Typ:</b>	Kluzné pouzdro
<b>Materiál:</b>	Slinutý bronz, samomazné
<b>Obrázek:</b>	 <p>Obr. 38: Velké kluzné pouzdro [23]</p>

V místě rotace je tedy umístěno kluzné pouzdro, o které se opírá jedna část velkého ramene. Druhá část ramene je spojena s převodovkou přes přírubu, čímž je do konstrukce přiváděn moment pro potřebné natočení ramene z motoru. Obě tato velká ramena jsou spolu spojena hliníkovými profily.



Obr. 39: Řez manipulátorem v místě B (Obr. 33)

Pro pohyb malého ramene pak skrze kluzné pouzdro prochází hřídel převodovky, na němž je připojena příruba s upínacím pouzdem. K přírubě je pak rameno připevněno.

### 3.5 Návrh a konstrukce výměnného nástroje

Jedním z úkolů bylo navrhnout hlavu manipulátoru tak, aby zde byla možnost výměny nástroje. Bylo tedy zapotřebí vymyslet způsob, kterým bude hlavice s nástrojem vzájemně dosedat k hlavě robota a to tak, aby pokaždé dosedla nejlépe do stejné polohy, tedy zajistit dobrou opakovatelnost spojení. Pro tento účel se dobře hodila Maxwelllova kinematická vazba [24]. Dalším důležitým prvkem bylo navrhnout, jak uzamknout nástroj po dosednutí v hlavě robota, aby tento spoj byl dostatečně pevný.

#### 3.5.1 Maxwelllova kinematická vazba

Princip je založen na třech drážkách tvaru V, které jsou vzájemně orientovány do středu nástroje. Protikus pak má tři kulové plochy, které zapadají do těchto V-drážek. Každá drážka tak poskytuje dva kontaktní body, tedy celkem šest. [24]



Obr. 40: Výměnná hlavice pro různé typy nástrojů    Obr. 41: Nástrojová část se zamykacím mechanismem

Místo V-drážek byla zvolena dvojice vedle sebe ležících válečků z ložisek, rozmístěných pod úhlem  $120^\circ$  od sebe a jako protikus kuličky o průměru 9,5 mm, jež do těchto drážek dosedají. Pro přenos elektrické energie a signálů z řídicí desky do nástroje byly použity magnetické kontakty, které se při přiblížení k sobě pevně spojí.

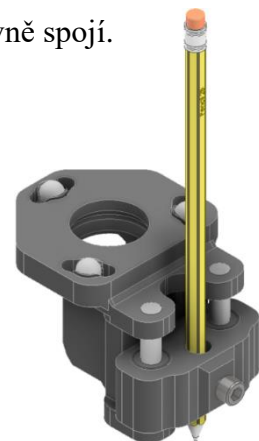
Byly vytvořeny celkem tři výměnné hlavice



Obr. 42: Nástroj pro měření s úchylkoměrem

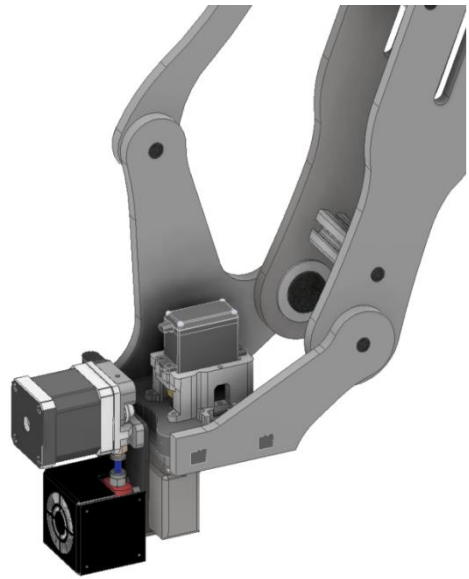


Obr. 43: Nástroj pro 3D tisk

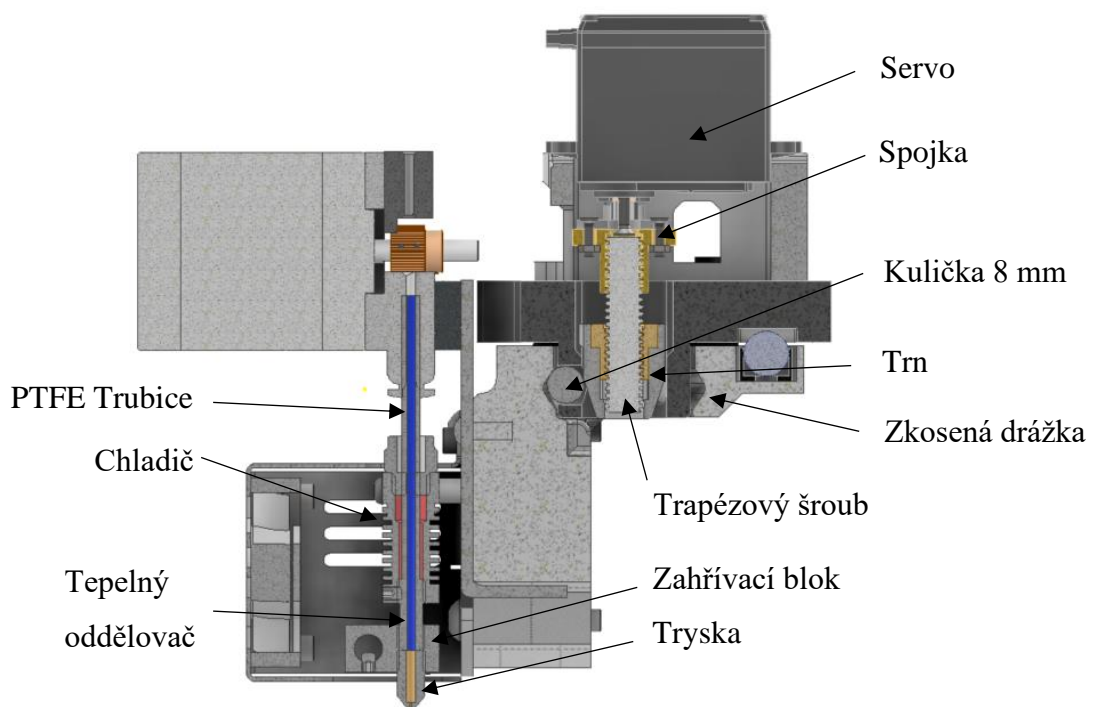


Obr. 44: Nástroj pro plotter

Zamykací mechanismus pak funguje následovně: V nástrojové části, jež je pevně přichycena k robotu, se nachází válcový výstupek (Obr. 41) s pěti ocelovými kuličkami o průměru 8 mm (byly zde použity náhradní kuličky z ložisek). Středem tohoto výstupku pak prochází trapézový šroub s trnem. Tento trn se posouvá směrem nahoru a dolů v závislosti na natočení serva ( $0^\circ$  až  $180^\circ$ ), jež je přes spojku spojeno se šroubem. Pokud se trn pohybuje po šroubovici dolů, tak se kuličky roztahují do stran a začínají tlačít na zkosené stěny výměnné hlavy, čímž ji lehce přizvednou a pevně zafixují. Při pohybu trnu zpátky nahoru se pak tento tlak kuliček uvolní a nástroj lze odejmout. Tyto úhly v drážce výměnné hlavy, do které se vtlačují kuličky, byly zvoleny za pomoci prototypování spolu s testováním (tedy náhodně). Z obrázku se může zdát, že bez přítomnosti nástroje, kuličky mohou vypadnout, v tom jim ale brání lehké zkosení do vnitřní strany otvoru, kde se nacházejí. Co se týče 3D tisku, tak byl zakoupen Creality Direct Extruder Kit (Motor, Chladič, Tryska, ...), které byly na nástroj namontovány [25].



Obr. 45: Nástroj upevněn v hlavě manipulátoru



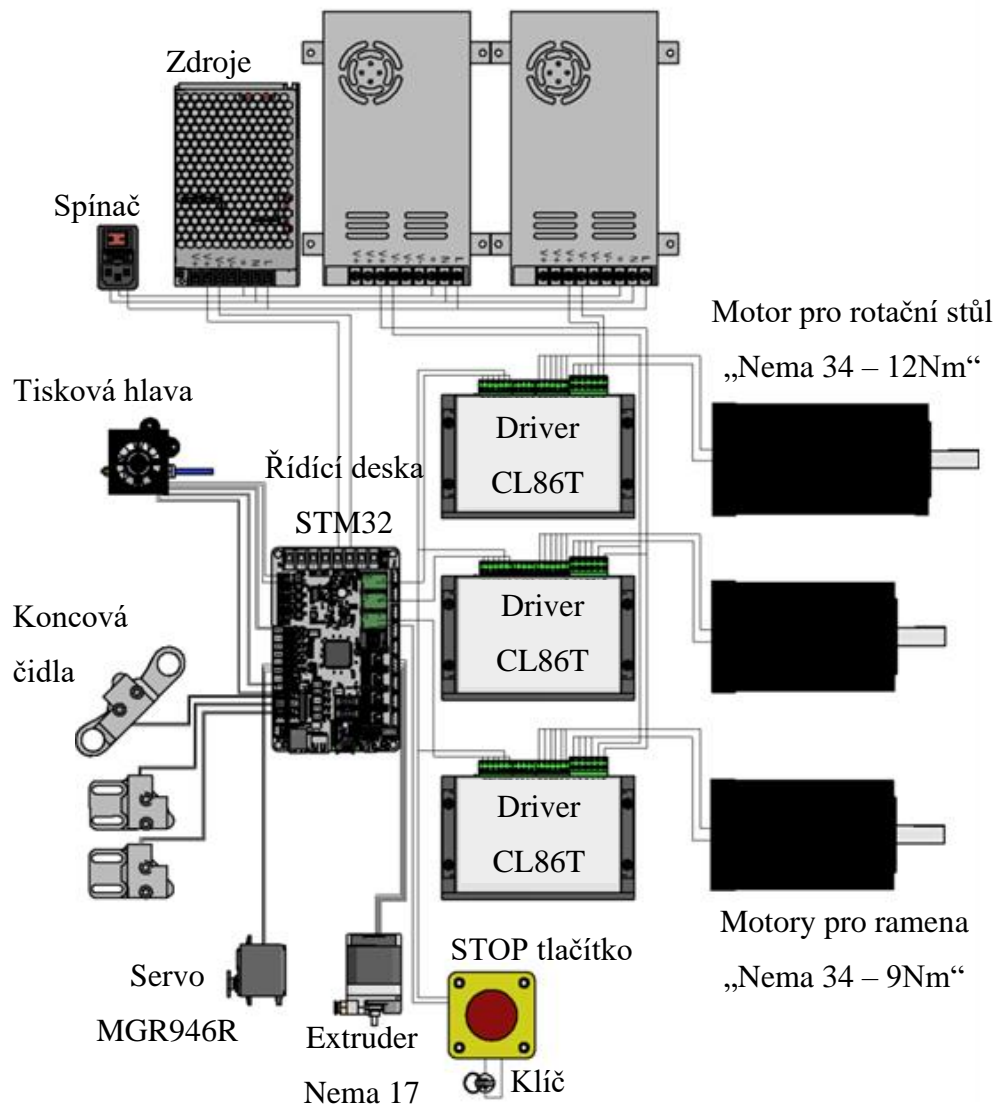
Obr. 46: Řez nástrojovou hlavou pro 3D tisk v manipulátoru

## 4 Elektrická konstrukční část

Zde bylo nutno rozhodnout, které součásti budou pro náš manipulátor použity. První volbou byla řídicí deska pro celkové ovládání. Poté se volily akční členy, kterými bude robot moci pohybovat a k těmto akčním členům bylo i zapotřebí pořídit patřičné drivery a napájení. Jako další pak byla volba pro vedlejší prvky, jako jsou spínače, čidla a možné součástky pro nástrojové hlavy robota.

### 4.1 Schéma propojení elektroniky v základně

Následující obrázek (Obr. 47) ukazuje náznak schématického zapojení jednotlivých částí. Akční členy jsou pak fyzicky z vnějšku spolu se STOP tlačítkem, připojeny přes kabely s průmyslovými konektory (Obr. 65) k řídicí desce (příloha [1]).

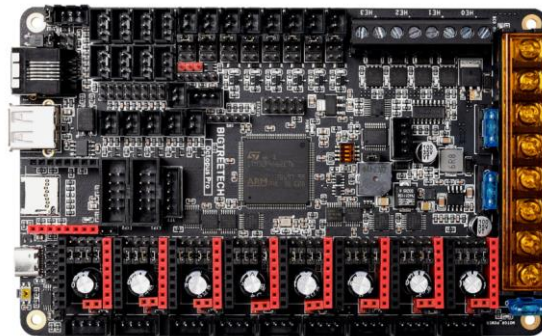


Obr. 47: Schéma zapojení

## 4.2 Řídící deska

Jelikož bylo rozhodnuto řídit manipulátor přes open-source software, který je využíván i pro 3D tisk, tak bylo potřeba zvolit takovou desku, která bude i pro tento typ firmwaru podporována. To však není nevýhoda, jelikož tyto desky pro řízení tiskáren mají právě velmi často dobře uzpůsobenou konstrukci a veškeré piny z procesoru jsou vyvedené ven na konektory, kde už dopředu počítají, že uživatel bude používat všemožné akční členy a senzory. Volba padla na desku BTT Octopus PRO V1.0 od firmy Bigtreetech.

Jejími kladem bylo velké množství připojitelných periférií, a také výkonný mikrokontrolér umístěný na desce ARM Cortex-M4-STM32F429ZGT6 (Obr. 48), vybavený 144 piny a frekvencí 168 Mhz s 1 MB flash pamětí. Napájecí napětí pro desku je 15-28 V, kde v tomto případě je připojen 24 VDC zdroj. Komunikace s deskou je umožněna přes USB-A, USB-C, CAN BUS s RJ11 konektorem nebo přímo přes UART piny na desce. [26]



Obr. 48: Octopus Pro STM32F429ZGT6 [26]

Deska také nabízí možnost připojení až čtyř vyhřívaných podložek, šesti PWM říditelných ventilátorů, slot pro ESP Wi-Fi modul, možnost řízení až osmi driverů motorů se signály o napěťové úrovni 5 V, sloty až pro pět teplotních senzorů a pro každý driver dva limitní koncové senzory. Tato deska má obrovskou výhodu také v tom, že může používat více úrovní napětí, jelikož napájení vyhřívací podložky, motorů a samotné desky je odděleno (jednotlivé okruhy jsou zde také chráněny pojistkou). U 3D tiskáren se často lze setkat se zařízeními, jež fungují na 12 V, či 24 V. Deska zde umožňuje napájení motorů až 60 V a způsob řízení motorů jako klasický STEP/DIR, UART, nebo SPI viz **kapitola 4.4.2**. Deska je také vybavena dalšími porty pro připojení vyrovnávacích senzorů (BL-Touch), dotykových obrazovek či čipem MAX31865 pro připojení termistorů. [26]

### 4.3 Motory

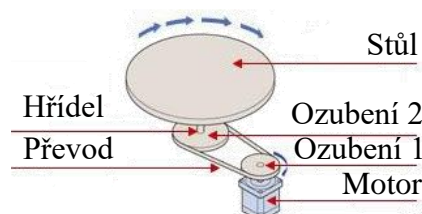
Při rozhodování, jaký motor nejlépe zvolit, bylo vybíráno ze dvou typů, a to buďto krokový motor, nebo střídavý servo pohon.

Výhodou krokového motoru je poměrně jednoduché řízení, má vysoký moment v nízkých otáčkách, plynulý chod při dobře zvoleném krokování, nepotřebuje zpětnou vazbu a pořizovací cena je nízká. Jeho velkou výhodou je však také, že je schopen držet plný moment při nulových otáčkách. Jeho nevýhodou je pokles momentu při větších otáčkách, konstantní odběr proudu a neumí dobře zvládat nárazové změny zátěže. [27]

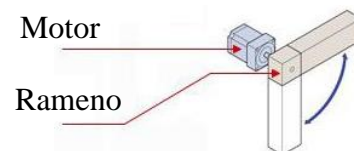
Výhodou střídavého serva je vysoký moment ve vysokých otáčkách a lepší volba pro různé typy zátěží, nevýhodou je zde potřeba zpětné vazby a složitější řízení, z toho je pak patrná vyšší cena. Ta se zde taky hodně odvíjí od složitosti driveru, kde cena pro AC servo dosahuje při obdobných parametrech až čtyřnásobku, a to nesmíme zapomenout na nutnost koupě brzdného odporu. [27]

Vítěznou volbou se tak staly krokové motory se zpětnou vazbou, jelikož nám nabízejí vysoký výkon a spolehlivost při poměrně nízké ceně. V našem případě se jedná o dvoufázové krokové motory s interním enkodérem. Tato kombinace nabízí obdobné vlastnosti jako servo systémy, ale za zlomek ceny. Díky zpětnovazební smyčce se vyhneme ztrátě kroků a díky způsobu ovládání skrze driver motoru jsme schopni dosáhnout vysokého momentu a otáček s nízkým ztrátovým teplem a hlukem. To, že motor dokáže mít maximální moment při nulových otáčkách, je pro robotické aplikace taky velkým bonusem.

Pro návrh, jak silný motor zvolit, byly použity online nástroje od různých firem [28] [29], kde byl zadáván jeden požadavek na motor pro rotační stůl a druhý pro pohyblivé rameno. Zde jsou vidět ukázkové obrázky (Obr.49,Obr.50) z online editoru od firmy Orientalmotor:



Obr.50: Schéma rotačního stolu [29]



Obr.49: Schéma ramene [29]

Zde byly zadány veškeré parametry, které editor umožnil a to od předpokládaných vlastností zátěže (délka, šířka, tloušťka, váha), tak i parametry jako je způsob uložení, nebo


zda je za motorem nějaký převod. Šlo zde pak i zadat požadavky na rychlosti a přesnosti při zastavování + bezpečnostní faktor.

Poté byl vybrán z dostupné cenové kategorie potřebný motor pro danou aplikaci. Zohledněno bylo i, aby motor a převodovka na sebe dobře pasovaly. Byl tedy vybrán jeden krokový motor Nema 34 – 12 Nm pro rotační stůl a dva Nema 34 – 9 Nm pro ovládání ramen. Spolu s těmito motory se také pořídily příslušné drivery a zdroje.

### 4.3.1 Servo pro zámek nástroje

Jelikož je potřeba vyvinout dostatečnou sílu na uzamknutí nástroje v hlavě manipulátoru, tak bylo použito modelářské servo s rozsahem pohybu  $0^\circ - 180^\circ$ . Návrh mechanismu je uveden v kapitole 3.5.

Tab. 6: Volba motoru pro zámek (hodnoty převzaty od prodejce [27])

Typ	Bipolární krok. motor	Obrázek
MG946-R [27]	$I = 1.2 \text{ A}$ , $U = 5 \text{ VDC}$ , Úhel = $0^\circ - 180^\circ$ Přidržený moment = 1.3 Nm (při 6V)	

Obr. 51: Servo MG946-R [27]

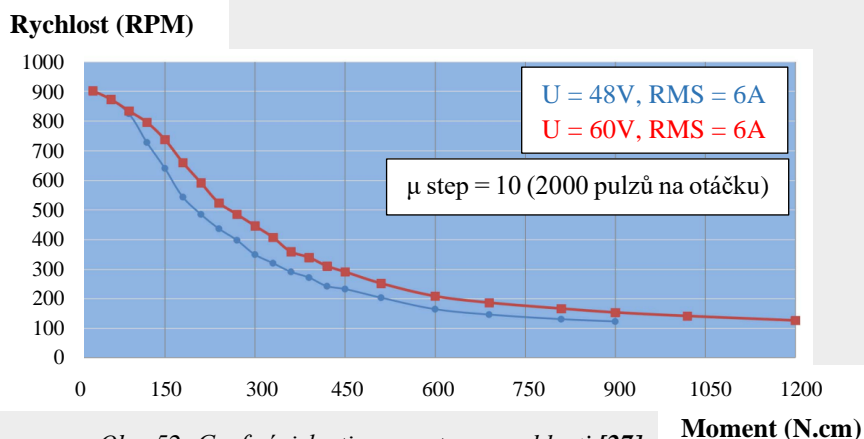
### 4.3.2 Zvolený motor pro rotační stůl

Tab. 7: Volba motoru pro rotační stůl (hodnoty převzaty od prodejce [27])

Typ	Bipolární krok. motor	Enkodér
34HS59-6004D-E1000 [27]	$I = 6 \text{ A}$ , $U = 60 \text{ VDC}$ , Úhel kroku = $1,8^\circ \pm 5\%$ Odpor fáze = $0,7 \pm 10\% \text{ Ohm}$ Induktance fáze = $7,9 \pm 20\% \text{ mH}$ Přidržený moment = 12 Nm	Optický inkrementální  Rozlišení enkodéru: 1000 PPR (4000 CPR)

#### Graf závislosti momentu na frekvenci a otáčkách

#### Obrázek



Obr. 52: Graf závislosti momentu na rychlosti [27]

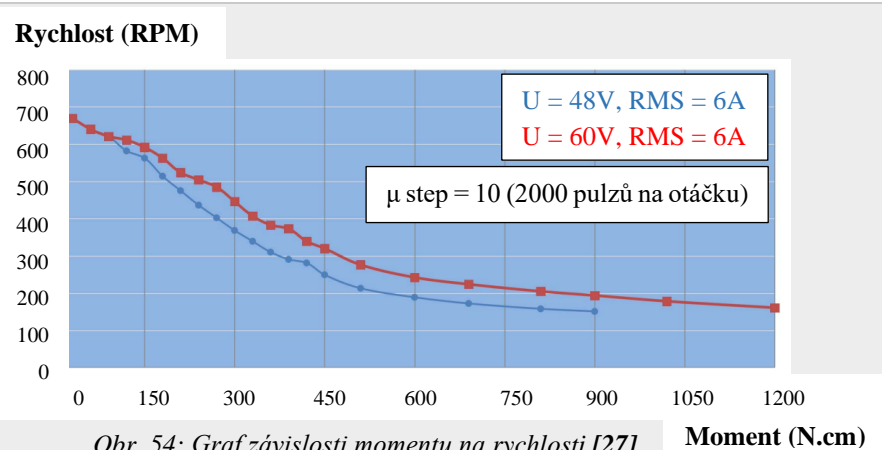



Obr. 53: Nema 34 – 12Nm [27]

Váha 5,5 kg

### 4.3.3 Zvolený motor pro ramena


Tab. 8: Volba motoru pro ramena (hodnoty převzaty od prodejce [27])

Typ	Bipolární krok. motor	Enkodér
<b>34HS46-6004D-E1000 [27]</b>	$I = 6 \text{ A}$ , $U = 60 \text{ VDC}$ , Úhel kroku = $1,8^\circ \pm 5\%$ Odpor fáze = $0,62 \pm 10\% \text{ Ohm}$ Induktance fáze = $6,3 \pm 20\% \text{ mH}$ Přídržný moment = $9 \text{ Nm}$	Optický inkrementální  Rozlišení enkodéru: $1000 \text{ PPR (4000 CPR)}$
<b>Graf závislosti momentu na frekvenci a otáčkách</b>		<b>Obrázek</b>
 <p>Obr. 54: Graf závislosti momentu na rychlosti [27]</p>		 <p>Obr. 55: Nema 34 - 9Nm [27]</p> <p><b>Váha 4,15 kg</b></p>

### 4.3.4 Motor pro extruder

Jelikož je jeden z nástrojů manipulátoru hlava pro 3D tisk, tak bylo nutno navrhnout součástky i pro tuto část. Extruder je v 3D tisku velmi podstatná část, jelikož s jeho pomocí dodáváme materiál do zahřátého elementu „trysky“, kde se pak daný materiál taví. Pro tyto účely tedy potřebujeme mechanismus, poháněný motorem, jenž bude tlačít tento materiál do trysky. V koupené sadě [25] se nacházel malý motorek NEMA17 a ten zde byl také použit. Ukázka mechanismu „dávkovače“ v řezu je uvedena v kapitole 3.5.1 (Obr. 46) o konstrukci nástrojové hlavy.

Tab. 9: Volba motoru pro extruder (hodnoty převzaty od prodejce [25])

Typ	Bipolární krok. motor	Obrázek
<b>Nema 17-42-40 [25]</b>	$I = 1.5 \text{ A}$ , $U = 24 \text{ VDC}$ , Úhel kroku = $1,8^\circ \pm 5\%$ Odpor fáze = $3,0 \pm 10\% \text{ Ohm}$ Induktance fáze = $3,0 \pm 20\% \text{ mH}$ Přídržný moment = $0,4 \text{ Nm}$	 <p>Obr. 56: Nema 17 [25]</p>



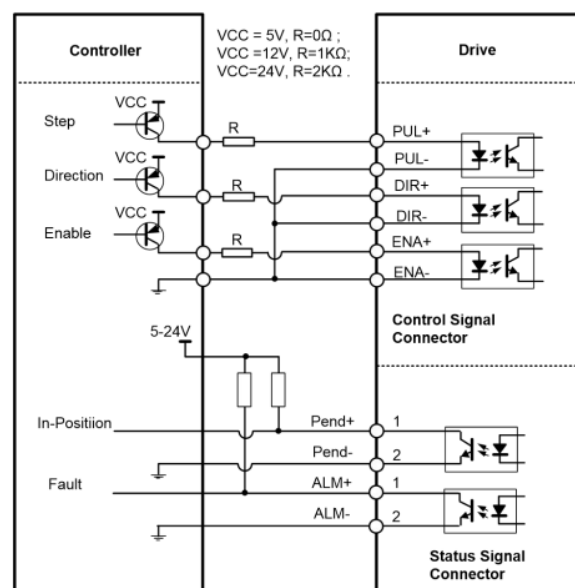
## 4.4 Drive CL86T a TMC2209

Pro krokové motory Nema34 byly zakoupeny drivery CL86T. Ty jsou napájeny na spínaných zdrojích o úrovni napětí 60 VDC skrze piny +Vdc a GND. Jelikož jsou všechny krokové motory bipolární, tak každý driver má vstup pro dvě cívky A a B, jejichž začátky a konce jsou připojeny k driveru. Poté je ještě nutno připojit enkodér motoru a lze začít ovládat motor [30].



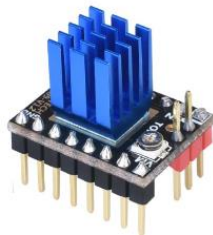
Obr. 57: CL86T - Driver pro krokové motory Nema 34 [30]

Ovládání je zde prováděno přes tzv. STEP/DIR způsob (**kapitola 4.4.2**). Tento driver má sám vlastní diagnostiku a software pro ovládání, či nastavení parametrů. Jelikož podporuje výstupní proud až 8 A, tak je zabudován v kovové krabici s velkým žebrováním pro pasivní chlazení. Na boční straně driveru pak lze manuálně nastavit skrze dip-switch volbu mikrosteppingu (**kapitola 4.4.1**).



Obr. 58: Schéma připojení driveru k řídicí desce [30]

Pro krokový motor použitý na extruderu jsem zvolil driver TMC2209, s kterým komunikuji skrze UART (**kapitola 4.4.2**) [31].



Obr. 59: TMC2209 – Driver pro krokové motory Nema 17 [31]

Z obrázku (Obr. 59) lze vidět, že se jedná o mnohem menší modul na menší výkony, ale i tak dle datasheetu je uvedeno, že snese proud až 2.8 A.

Tento driver nabízí funkce, jako je „StealthChop“, což zajišťuje tichý a velmi přesný chod motoru, nebo například „StallGuard“ pro bezsenzorové houmování (návrat motorů ve všech osách do výchozí polohy). Co se týče mikrosteppingu, tak má možnost dělit krok 2, 4, 8, 16, 32, 64, 128, 256. Výhoda tohoto driveru je ale hlavně v tom, že jde fyzicky zapojit přímo do řídicí desky, kde máme celkem osm slotů pro tyto typy driverů na motory. Pro CL86T to tak nešlo, zde bylo nutno jednotlivé piny z řídicí desky dovést až k driverům. Veškeré zapojení motorů je vypsáno v tabulce v příloze [1].

#### 4.4.1 Mikrostepping

Jedná se o dělení kroků na otáčku. Kdyby tato funkce nebyla použita, tak pro náš motor Nema 34 by jeden pulz znamenal otočení hřídele o 1,8°. To znamená, že motor se otáčí celý dokola 360° po 200 pulzech. Díky mikrosteppingu však dokážeme jednotlivé kroky „rozdělit“ na menší a zvýšit tak přesnost pohybu. Každý výrobce většinou uvádí tabulku, z které si pak lze vybrat požadované dělení. V našem případě u driveru CL86T se jedná o tabulku (Tab. 10: CL86T mikrostepping (úplnou tabulku lze vidět v manuálu driveru [30])):

Tab. 10: CL86T mikrostepping [30]

Mikro step	Steps/Revolution	SW1	SW2	SW3	SW4
1	Default	on	on	on	on
4	800	off	on	on	on
8	1600	on	off	on	on
16	3200	off	off	on	on
↓	↓	↓	↓	↓	↓
200	40000	off	off	off	off

Kde nastavení volby kroku volíme pomocí DIP-Switche na straně driveru. Co se týče driveru TMC2209, tak tam se mikrostepping volí pomocí softwaru. Další výhodou tohoto dělení kroků je redukce hluku běhu motoru a samotné otáčení motoru se pak chová hladčeji i při nízkých rychlostech. Zdá se to jako jasná volba, že čím větší mikrostepping, tím vyšší přesnost, tak proč tedy nedát tu nejvyšší možnou hodnotu? V tom je právě chyták, jelikož s rostoucí hodnotou mikrosteppingu nám drasticky klesá záběrový moment motoru viz grafy (Obr. 52, Obr. 54) v kapitole 4.3.2, 4.3.3. Je tak třeba najít takovou hodnotu, abychom měli hladké a přesné otáčky s dostatečným momentem pro náš stroj. Důležité je také nezapomenout, že pokud používáme převodovku, tak ta nám rozlišení otáček a moment kroku ovlivňuje také. [30] [31]

Zde je ukázáno, jaká hodnota dělení kroků byla zvolena v našem případě:

**Nema 34 pro ramena:** Zvolen 16 mikrostepping, převodovka 10:1

$$\text{Počet kroků na plnou otáčku: } 360^\circ = 200 \cdot 16 \cdot 10 = \mathbf{32000 \text{ kroků}} \quad (15)$$

$$\text{Počet kroků na jeden stupeň: } 1^\circ = 32000 : 360 = \mathbf{88.8888 \text{ kroků}} \quad (16)$$

$$\text{Přesnost jednoho kroku: } \mathbf{1 \text{ krok}} = 1 : 88.8888 = \mathbf{0.0125^\circ} \quad (17)$$

**Nema 34 pro rotační stůl:** Zvolen 16 mikrostepping, převodovka 15:1

$$\text{Počet kroků na plnou otáčku: } 360^\circ = 200 \cdot 16 \cdot 15 = \mathbf{48000 \text{ kroků}} \quad (18)$$

$$\text{Počet kroků na jeden stupeň: } 1^\circ = 48000 : 360 = \mathbf{133.3333 \text{ kroků}} \quad (19)$$

$$\text{Přesnost jednoho kroku: } \mathbf{1 \text{ krok}} = 1 : 133.3333 = \mathbf{0.0075^\circ} \quad (20)$$

**Nema 17 pro extruder:** Zvolen 16 mikrostepping

$$\text{Počet kroků na plnou otáčku: } 360^\circ = \mathbf{1.8^\circ} \cdot 200 \rightarrow 200 \cdot 16 = \mathbf{3200 \text{ kroků}} \quad (21)$$

$$\text{Počet kroků na jeden stupeň: } 1^\circ = 3200 : 360 = \mathbf{8.8888 \text{ kroků}} \quad (22)$$

$$\text{Přesnost jednoho kroku: } \mathbf{1 \text{ krok}} = 1 : 8.8888 = \mathbf{0.1125^\circ} \quad (23)$$

Pro extruder nás však nezajímá, kolik kroků je třeba na stupeň otáčky, ale záleží zde na posuvu v mm. Do extruderu byl tedy vložen 120 mm dlouhý kus filamentu a motor dostal úkol odjet 100 mm jeho délky, po tomto úkonu by nám mělo zůstat 20 mm. Při nastavené hodnotě kroku 93 step/mm však zbývalo cca 25 mm. Zde jsme pomocí vzorce

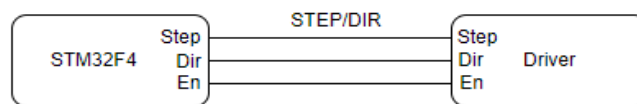
$$\text{NovéKroky} = \frac{\text{PuvodníKroky} \cdot \text{ZadanáDélka}}{\text{CelkováDélka} - \text{Zbytek}} = \frac{93 \cdot 100}{120 - 25} = \mathbf{97.89} \quad (24)$$

dopočetli přesnou hodnotu pro nastavení kroků extruderu. Tuto hodnotu pak vložíme do Marlinu: `STEPS_PER_UNIT {133.33333333, 88.88888889, 88.88888889, 97.89}`

## 4.4.2 Komunikace s drivery

Máme hned několik způsobů, jakými se drivery u krokových motorů dají ovládat a komunikují s řídicí deskou. Zde se specificky zaměříme na ty metody, jež se používají u 3D tiskáren a jdou tedy použít i pro naši řídicí desku. Jako první je tzv. Step/Direction mode. Jedná se o prosté posílání napěťových pulzů na vstupní piny driveru a ten pak podle toho rozhodne, co se stane.

Tyto vstupní piny jsou většinou nazvány Ena (povolení napájení motoru), Dir (směr otáčení motoru) a Step „Pul“, který na každou náběžnou / sestupnou hranu pootočí motor o patřičný úhel. Nevýhoda tohoto způsobu je v tom, že driver nevrací žádné informace do řídicí desky. [31]



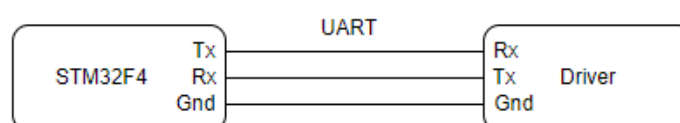
Obr. 60: Schéma ovládání přes STEP/DIR

Další běžně používaný je SPI mode (Serial peripheral interface). Tato konfigurace už zahrnuje více pinů, ale nabízí nám už informace o poloze motoru či odebíraném proudu. Pro komunikaci potřebuje minimálně čtyři piny „CLK, MISO, MOSI, CS1“. Výhoda SPI je rychlý přenos dat. [31]



Obr. 61: Schéma komunikace SPI

Dalším způsobem je UART (universal asynchronous receiver-transmitter). Je to tzv. hardwarový komunikační protokol. Komunikace je zde prováděna čistě mezi jedním čipem driveru a řídicí deskou. To je odlišné od SPI, kde komunikace může probíhat s více moduly naráz. Pro komunikaci UART potřebujeme tři piny „RX, TX a GND“. Výhodou UART je možnost komunikace na delší vzdálenosti (jednotky metrů). Jak SPI, tak UART dovolují komunikaci v obou směrech. [26] [31] [32]





Obr. 62: Schéma komunikace UART

## 4.5 Volba zdroje

Napájení bylo zvoleno dle doporučení prodejce krokových motorů, a tedy i přímo od něj tyto zdroje zakoupeny. Používáme zde dva 60 VDC spínané zdroje, kde je jeden připojen na motor rotačního stolu a druhý zdroj je společný pro oba motory na ramenech. [33]

Poté používáme ještě jeden zdroj, jenž je poloviční velikosti a napájí veškerou zbylou elektroniku, což je například řídicí deska, motor extruderu, ventilátory a v případě použití i hotbed (vyhřívaná podložka pro 3D tisk) s hotendem (vyhřívaný blok, ve kterém se nachází tryska pro 3D tisk).

Tab. 11: Volba zdroje (hodnoty převzaty od prodejce [33])

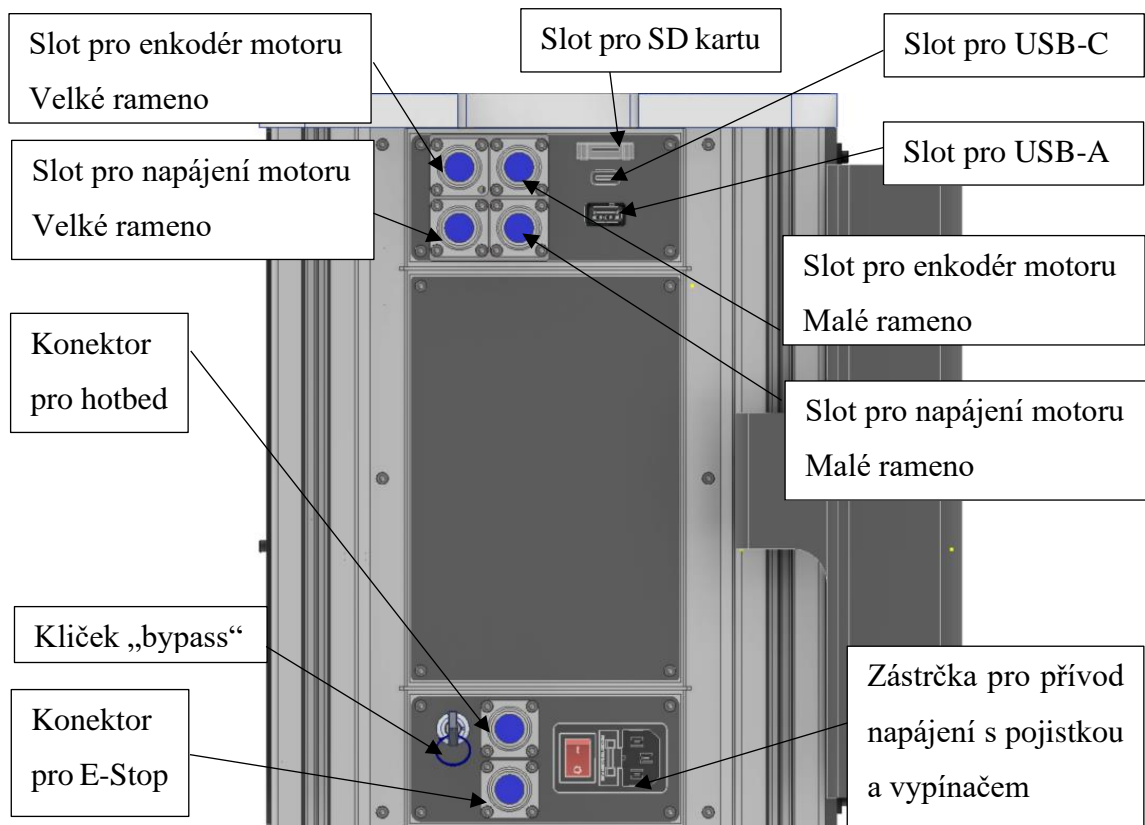
Typ	Parametry	
<b>S-350-60</b> [33]	Vstupní hodnoty 230 VAC, 4A Výstupní hodnoty 60 VDC, 5.9A Účinnost 86% Zabudovaný ventilátor pro chlazení Zabudovaná ochrana: tepelná, nadproudová, přepět'ová	 Obr. 63: Spínaný zdroj pro motory [33]
<b>RS-100-24</b> [33]	Vstupní hodnoty 230VAC, 2.5A Výstupní hodnoty 24VDC, 4.5A Účinnost 84% Zabudovaná ochrana: tepelná, nadproudová, přepět'ová	 Obr. 64: Spínaný zdroj pro desku a další prvky [33]

## 4.6 Přední panel

Důležitou součástí je také způsob, kterým se připojují veškeré venkovní periférie k řídicí desce. Pro tento účel byl vytvořen přední panel na základně manipulátoru, kam jsou všechny tyto potřebné konektory z řídicí desky přivedeny.

Máme zde kontakty pro připojení obou motorů ramen manipulátoru, a to jak pro napájení cívek, tak pro enkodér. Konektory jsou odlišeny tak, že pro napájení cívek byl zvolen konektor se čtyřmi piny a pro enkodér s piny devíti. Tímto provedením byla snaha zamezit špatnému připojení konektorů. Zde nastávají dvě možné varianty poruchy, pokud jsou prohozeny například enkodéry motorů, tak to driver pozná a ohlásí chybu, druhý scénář je, že se zapojí oba kabely špatně a poté to poznáme na tom, že robot dělá to, co nemá. Nedojde však nikdy ke zničení zařízení. Pak se zde nacházejí konektory pro komunikaci s řídicí deskou, k tomu používáme dva USB porty a SD kartu.

Ve spodní části panelu se pak nachází konektor pro připojení hotbedu (ten je napájen z vnitřního zdroje 24 VDC) a konektor pro připojení E-stopky, která jde překlenout klíčkem viz **kapitola 4.6.1** o e-stopu. Jako poslední je zde zásuvka pro přívod elektrické energie, která má v sobě zabudovaný vypínač a 10 A trubičkovou pojistku.

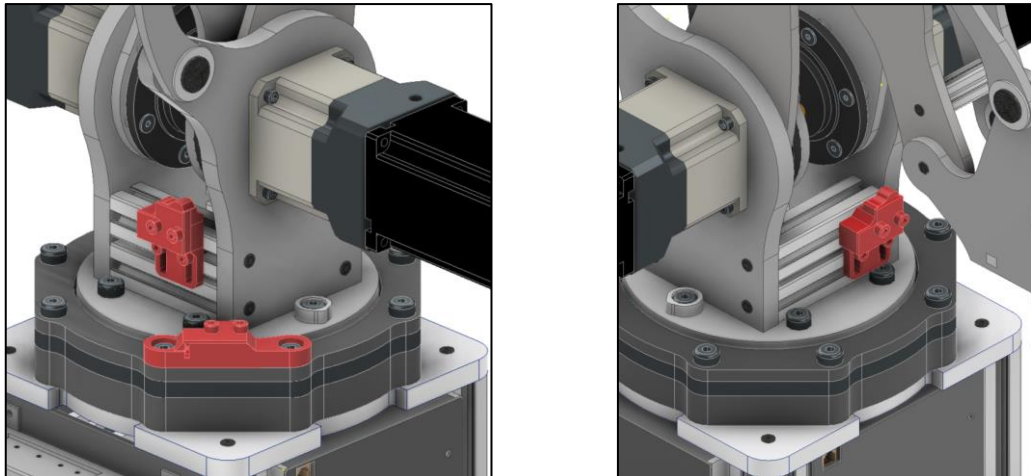


Obr. 65: Přední panel s konektory

U manipulátoru je většina rizik pokryta v softwaru. Tedy veškeré polohy manipulátoru, v kterých hrozí problém, nebo možná přetížení v motorech se ve většině případů řeší v programu a způsobí zastavení robota v jeho činnosti. Poté ale nastávají chvíle, kdy my jako uživatelé víme, že se děje něco špatného a je třeba robota zastavit. Nebo jako uživatel chceme s robotem zajet do nějaké pozice, o které přesně víme, jaké nastavení úhlů zde je. Z tohoto důvodu jsou zde přidána koncová čidla a e-stop.

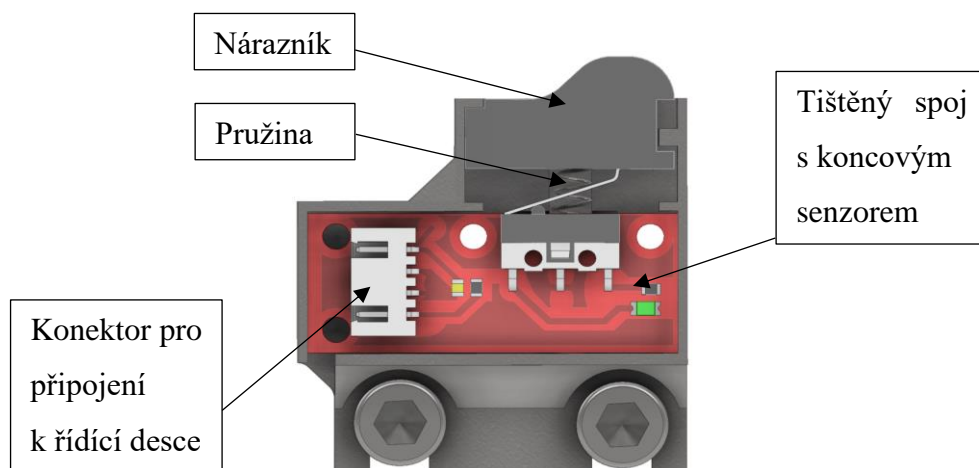
#### 4.6.1 Koncová čidla a E-stop

Červenou barvou je (Obr. 66) zvýrazněno umístění jednotlivých koncových senzorů pro rotační stůl a ramena na manipulátoru. Ty slouží jak už z názvu vyplývá, k detekci koncové pozice jednotlivých ramen.



Obr. 66: Umístění koncových spínačů pro tři motory

Uvnitř těchto senzorů se pak skrývá jednoduchý mechanismus složený z plastové krabičky, pružiny z propisky, nárazníku a PCB desky s koncovým senzorem.



Obr. 67: Konstrukce koncového spínače

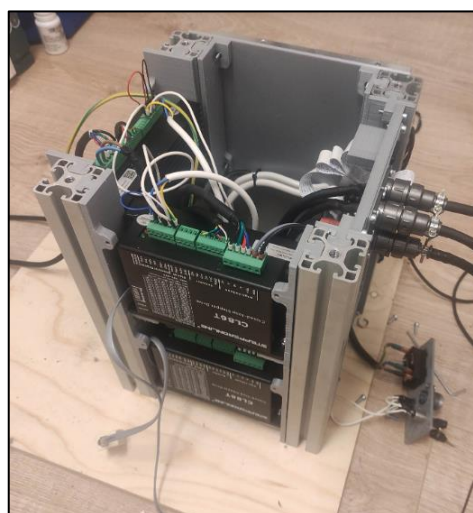
Pružina je zde použita z důvodu větší síly pro návrat nárazníku do původní polohy po kontaktu s konstrukcí. Jinak princip je snadný, po stisku se vyšle signál do řídicí desky a ta rozpozná, že v daném místě bylo dosaženo fyzického limitu a motor se zastaví.

I kdyby došlo k případu, že by čidlo nezafungovalo, tak každý driver sleduje odběr proudu motoru a pomocí enkodéru zase pozná, když kroky nejdou správně za sebou. Při kolizi tak dojde k vyhlášení chyby na daném driveru.

Jako další nutný prvek bylo vytvořeno tlačítko pro nouzové zastavení „E-Stop“. To je připojeno na povolovací „Enable“ pin na řídicí desce pro pohyb motorů a odtud vede k jednotlivým driverům. Toto nouzové tlačítko funguje jako NC „normally close“ kontakt, tedy pokud není tlačítko stisknuto (a je připojeno), tak může robot fungovat. Pokud však tlačítko stiskneme, obvod se přeruší a motorům se odpojí napájení.



Obr. 68: E-Stop tlačítko



Obr. 69: Zapojování a pájení v základně robota

Stejný efekt nastane, pokud někdo odpojí kabel s „E-stopem“ z konektoru, jelikož pro řídicí desku je to to samé, jako by bylo tlačítko stisknuto. Pro tyto případy je zde přidán klíček, který slouží jako „bypass“ této funkce, takže přemostíme rozpínací kontakty a tlačítko lze odpojit.

Pro připojení jednotlivých venkovních periférií (Obr. 69) jsou použity průmyslové konektory řady DS1110, které splňují požadavky na potřebné napětí a proud. Také jsou velmi robustní a mají možnost zajištění pomocí závitů. V případě, že není konektor k ničemu připojen, tak se na něj nasadí plastová krytka a nehrozí tak žádný úraz el. proudem pro osoby manipulující s robotem. Jak jsou tyto konektory připojeny k řídicí desce, je popsáno v příloze [2].



## 5 Návrh a implementace řízení manipulátoru

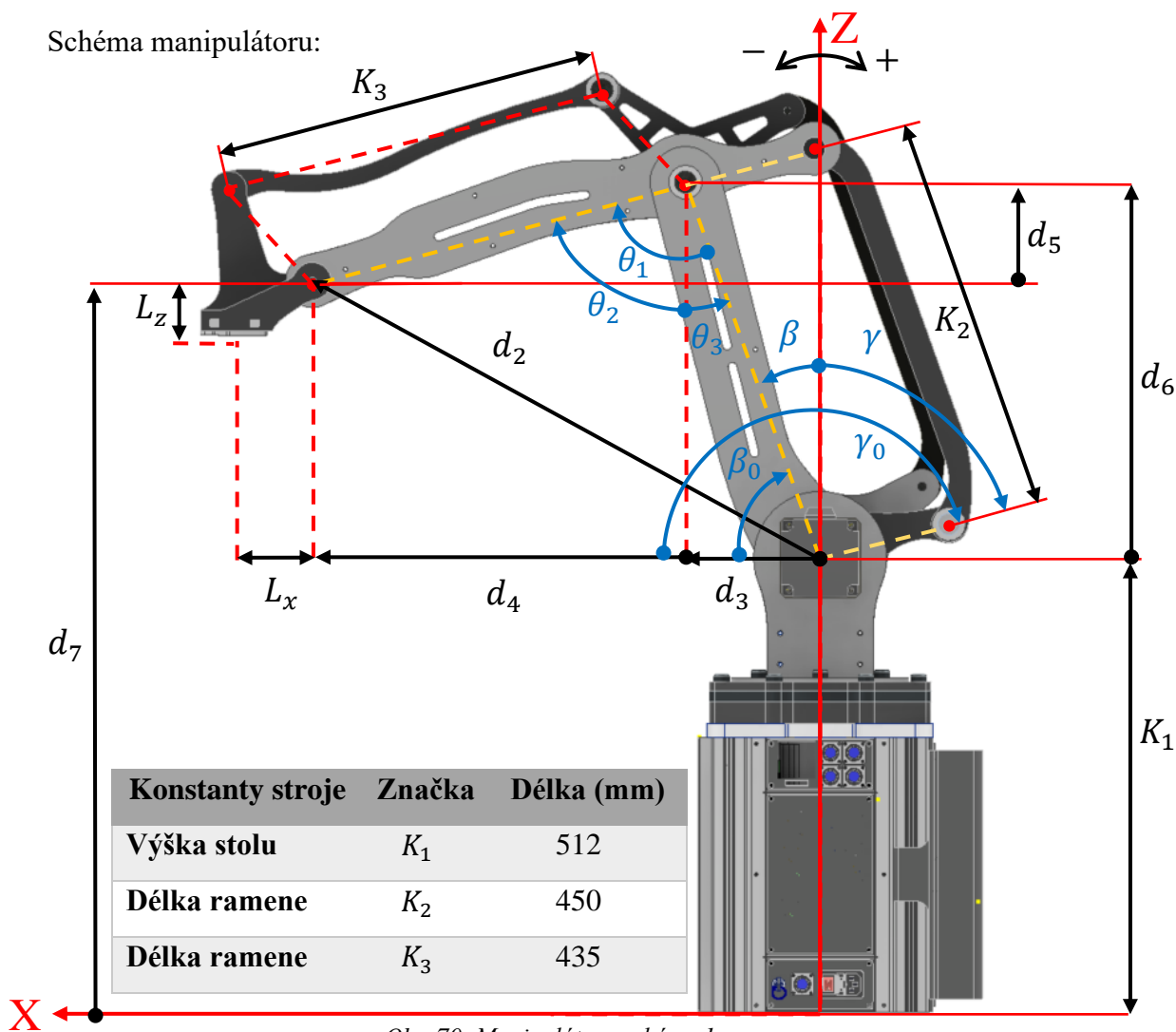
Jako první bylo nutno navrhnout kinematiku pro řízení robota, kterou pak implementujeme do zvoleného open-source firmwaru v řídicí desce. Tuto desku budeme poté ovládat přes grafické rozhraní v počítači

### 5.1 Přístupy pro zjištění inverzní a dopředné kinematiky manipulátoru

Dopředná a inverzní kinematika jsou základním kamenem pro řízení v robotice. Jedná se o matematický popis pohybů robota. Je zde však více způsobů, jak k jednotlivým metodám přistupovat a získat jejich výsledky. Dopředná kinematika se zabývá zjištěním pozice a orientace koncového nástroje robota (v našem případě hrot tužky, nebo trysky) na základě zadaných úhlů natočení jednotlivých ramen. To lze vypočítat například pomocí transformačních matic, nebo v případě, že konstrukce není příliš složitá, tak analytickou metodou. Inverzní kinematika je metoda pro zjištění natočení úhlu ramen, aby tak bylo dosaženo požadované pozice koncovým nástrojem. My si ukážeme řešení analytické, tedy ze znalosti konstrukce robota jednotlivé úhly dopočteme ručně. Druhé bude řešení numerickou metodou, která je náročnější na výpočetní čas a výkon procesoru, ale u komplexnější konstrukcí robota, se tomuto řešení nevyhneme. [3] [34]

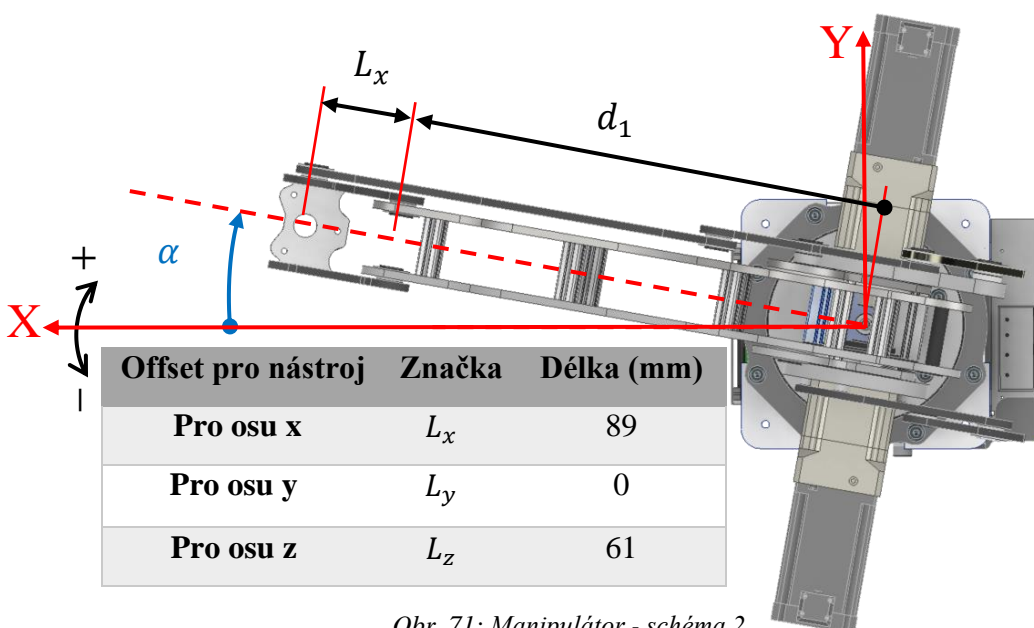
#### 5.1.1 Dopředná a inverzní kinematika analytickou metodou

Pro analytické řešení obou kinematik je nutná dobrá znalost konstrukce robota. Jako první tedy bylo nutno vytvořit schéma (Obr. 70, Obr. 71) a označit jednotlivé parametry, které budou používány při výpočtech. Zde je nutno si stanovit pravidla. Globální souřadný systém má počátek v základně a kladné směry os X, Y, Z jsou vyznačeny na obrázcích, stejně tak jsou vyznačeny kladné a záporné úhly. Máme zde mediánní rovinu (podélný řez robotem) v níž leží krátké i dlouhé rameno, tato rovina rotuje kolem osy Z. V této rovině pak leží všechny veličiny K, d, L a úhly vztahující se k těmto ramenům. Výjimkou je úhel  $\alpha$  (jenž leží v rovině XY) a vzdálenost  $L_y$ , která je kolmá na mediánní rovinu. Hodnoty označené písmenem „K“ jsou konstantní hodnoty stroje, které jsou dány jednotlivými rozměry konstrukce. Označené malým písmenem „d“ jsou pak vypočtené vzdálenosti z kinematiky. Písmeno „L“ označuje offset pro případný nástroj v hlavě robota. Tato hodnota je odlišná pro každý typ nástroje a změnu tohoto parametru pak řešíme v softwaru. Pro ukázkou zde byly zvoleny hodnoty pro prázdnou hlavu robota.



Obr. 70: Manipulátor - schéma 1

Hodnoty délek  $L_x$ ,  $d_4$ ,  $d_3$  jsou rovnoběžné s rovinou XY (tedy kolmé na osu Z).



Obr. 71: Manipulátor - schéma 2

**Dopředná kinematika** – Zde počítáme konečnou pozici robota v prostoru dle zadaných úhlů natočení hřídelí motorů  $\alpha, \beta, \gamma$  v radiánech. Po zadání úhlů nám tak vyjdou souřadnice nástroje robota  $x, y, z$  v mm. Důležité je také připočíst offset nástroje  $L_x, L_y, L_z$  v hlavě. Rozsahy ve stupních jsou:  $\alpha(-175^\circ; 175^\circ)$ ,  $\beta(-80^\circ; 60^\circ)$ ,  $\gamma(0; 115)$ . V našem případě počítáme pro  $L_y = 0$ , tedy jej nezahrnujeme do výpočtu.

$$\beta_0 = \frac{\pi}{2} + \beta, \quad (25)$$

$$\gamma_0 = \frac{\pi}{2} + \gamma, \quad (26)$$

kde úhly  $\gamma_0$  i  $\beta_0$  jsou natočení motorů vůči základně.

### Pozice X:

Z Obr. 71 vidíme, že pozice nástroje v ose X se získá jako:

$$X = \cos(\alpha) \cdot (d_1 + L_x), \quad (27)$$

kde potřebnou délku  $d_1$  získáme z Obr. 70 jako:

$$d_1 = d_3 + d_4, \quad (28)$$

$$d_3 = \cos(\beta_0) \cdot K_2, \quad (29)$$

$$d_4 = \sin(\theta_2) \cdot K_3, \quad (30)$$

kde  $K_2$  a  $K_3$  jsou hodnoty stroje a úhel  $\theta_2$  vezmeme ze vztahů:

$$\theta_2 = \theta_1 - \theta_3, \quad (31)$$

$$\theta_1 = \gamma_0 - \beta_0, \quad (32)$$

$$\theta_3 = \frac{\pi}{2} - \beta_0. \quad (33)$$

### Pozice Y:

Z Obr. 71 vidíme, že pozice bodu v ose Y se získá jako:

$$Y = \sin(\alpha) \cdot (d_1 + L_x), \quad (34)$$

kde  $d_1$  získáme stejným způsobem jako u výpočtu pozice X.

### Pozice Z:

Z Obr. 70 vidíme, že pozice bodu v ose Z se získá jako:

$$Z = d_7 - L_z, \quad (35)$$

$$d_7 = K_1 + d_6 - d_5, \quad (36)$$

kde  $K_1$  je pevná hodnota stroje,  $d_6$  a  $d_5$  dopočítáme jako:

$$d_6 = \sin(\beta_0) \cdot K_2, \quad (37)$$

$$d_5 = \cos(\theta_2) \cdot K_3, \quad (38)$$

kde  $K_2, K_3$  jsou pevné délky ramen a úhel  $\theta_2$  získáme ze závislostí výše.

$$X = \cos(\alpha) \cdot \left( \cos\left(\frac{\pi}{2} + \beta\right) \cdot K_2 + \sin(\gamma) \cdot K_3 + L_x \right), \quad (39)$$

$$Y = \sin(\alpha) \cdot \left( \cos\left(\frac{\pi}{2} + \beta\right) \cdot K_2 + \sin(\gamma) \cdot K_3 + L_x \right), \quad (40)$$

$$Z = K_1 + \sin\left(\frac{\pi}{2} + \beta\right) \cdot K_2 - \cos(\gamma) \cdot K_3 - L_z. \quad (41)$$

**Inverzní kinematika** – Zde zadáváme souřadnice  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  konečné polohy bodu nástroje robota a dopočítáváme úhly  $\alpha$ ,  $\beta$ ,  $\gamma$  pro natočení hřídelí motorů ve stupních. K zadaným souřadnicím nástroje robota  $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$  v mm je také nutno připočíst/odečíst offset nástroje  $L_x$ ,  $L_y$ ,  $L_z$ , jež má robot v hlav, kde  $L_y = 0$ .

**Úhel  $\alpha$  :**

$$\alpha = \text{atan2} \left( \frac{y}{x} \right), \quad (42)$$

kde používáme funkci atan2 pro případ kdy  $x = 0$  nebo  $x < 0$

**Úhel  $\beta$  :**

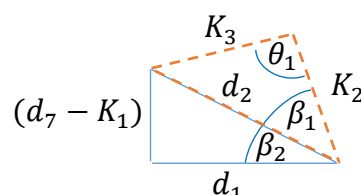
Tento úhel skládáme z dvou trojúhelníků (Obr. 72), kde si nejprve spočteme přeponu mezi nimi

$$d_2 = \sqrt{(d_1)^2 + (d_7 - K_1)^2}, \quad (43)$$

kde  $d_1$  a  $d_7$  získáme jako

$$d_1 = \frac{y}{\sin(\alpha)} - L_x, \quad (44)$$

$$d_7 = z + L_z. \quad (45)$$



Obr. 72: Pomocný obrázek

Poté z trojúhelníku mezi hlavními rameny můžeme přes kosinovou větu vyjádřit úhel

$$\beta_1 = \text{acos} \left( \frac{K_2^2 + d_2^2 - K_3^2}{2 \cdot K_2 \cdot d_2} \right) \quad (46)$$

a z druhého dopočteme

$$\beta_2 = \text{asin} \left( \frac{d_7 - K_1}{d_2} \right). \quad (47)$$

Výsledný úhel je pak sumou těchto dvou

$$\beta = \beta_1 + \beta_2 - \frac{\pi}{2}. \quad (48)$$

**Úhel  $\gamma$  :**

Když použijeme znovu kosinovou větu pro horní trojúhelník (Obr. 72), tak dostaneme

$$\theta_1 = \text{acos} \left( \frac{K_2^2 + K_3^2 - d_2^2}{2 \cdot K_2 \cdot K_3} \right). \quad (49)$$

Výsledný úhel je pak součtem

$$\gamma = \beta_1 + \beta_2 + \theta_1 - \frac{\pi}{2}. \quad (50)$$

**Úhel  $\alpha$**  tedy používáme pro motor rotačního stolu, **úhel  $\beta$**  pro natočení většího ramene, **úhel  $\gamma$**  pro natočení menšího ramene.

### 5.1.2 Dopředná kinematika s homogenními transformačními maticemi

Homogenní transformační matice jsou nástroj používaný v robotice k popisu polohy a orientace koncového nástroje vzhledem k jeho základnímu rámu. Obecný tvar homogenní transformační matice 4x4 je složen ze čtyř matic. [34] [35]

$${}^A_B T = \begin{bmatrix} {}^A_B R & {}^A_B P \\ \mu^T & \sigma \end{bmatrix}, \quad (51)$$

kde R reprezentuje rotační matici (3x3) a P translační vektor (3x1),  $\mu$  perspektivní vektor (1x3), který je v kinematice nulový a měřítko  $\sigma$  (1x1), které obvykle nabývá hodnoty 1. Písmeno A nám říká, odkud jdeme, tedy „počátek“ a písmeno B označuje, kam jdeme, „cíl“. Tvary jednotlivých matic potom vypadají: [35]

$${}^A_B R = \begin{bmatrix} {}^A_B X & {}^A_B Y & {}^A_B Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (52)$$

$${}^A_B P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \quad (53)$$

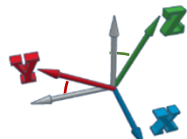
$$\mu^T = [0 \ 0 \ 0], \quad (54)$$

$$\sigma = [1], \quad (55)$$

Rotační matici jinak můžeme také zapsat jako:

$${}^A_B R = R_x \cdot R_y \cdot R_z, \quad (56)$$

kde  $R_x$  symbolizuje rotaci kolem osy X,  $R_y$  rotaci kolem osy Y a  $R_z$  rotaci kolem osy Z, translační vektor P pak symbolizuje posuv v osách X, Y, Z [3].



Obr. 73: Rotace kolem osy X

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \quad (57)$$



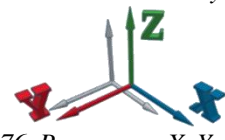
Obr. 74: Rotace kolem osy Y

$$R_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \quad (58)$$



Obr. 75: Rotace kolem osy Z

$$R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (59)$$



Obr. 76: Posuv v ose X, Y, Z

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}, \quad (60)$$

Výsledná matice má tedy tvar:

$${}^A_B T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (61)$$

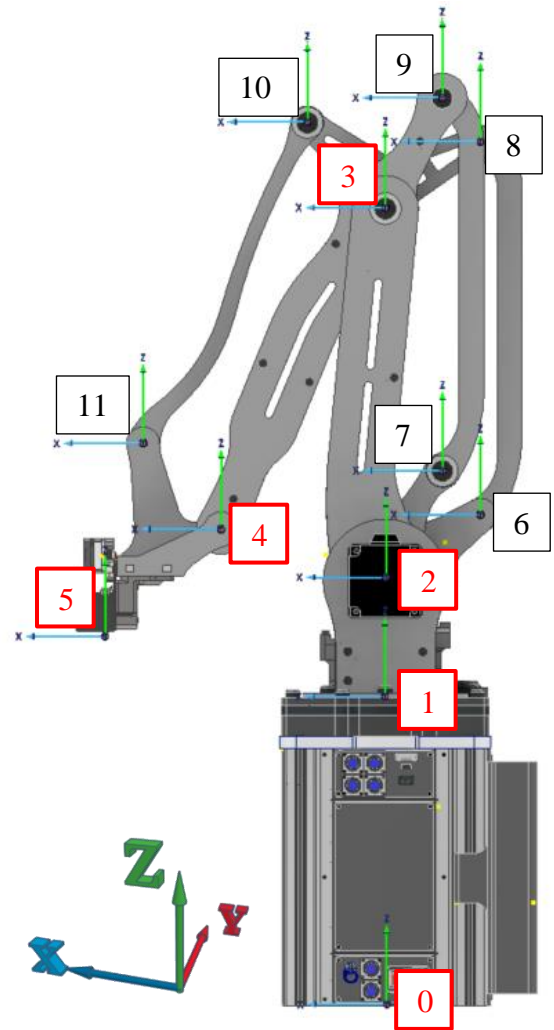
Aby šla použít tato matice pro dopřednou kinematiku, tak si musíme definovat souřadnicový systém pro každý kloub v ramenech robota. Jakmile tak učiníme, tak můžeme postupně začít vypisovat transformační matice pro každý spoj. [36]

Popisujeme tak vždy vztah mezi kloubem aktuálním a jeho předchůdcem. Poté výsledným roznásobením těchto matic dostaneme polohu a orientaci k základnímu rámu, „počátku“ robota. Jedná se o velice mocný a univerzální nástroj pro popis kinematiky (pohybu) robota přehledným způsobem. Navíc lze z těchto matic posléze vypočíst úhly natočení jednotlivých kloubů pomocí inverzní kinematiky. [36]

Byla tedy zvolena jednotlivá místa „kloubů“, kam se umístí souřadný systém a tyto místa očíslováme (Obr. 77). Hlavní klouby, které nás nejvíce pro řízení zajímají jsou: 0, 1, 2, 3, 4, 5. Vedlejší klouby, skrze něž se vytvářejí uzavřené kinematické smyčky značíme: 6, 7, 8, 9, 10, 11. Pro hlavní výpočty pohybu robota by nám tedy stačila pouze znalost dopředné kinematiky od počátku až k pozici nástroje:

$${}^0_5 T = {}^0_1 T \cdot {}^1_2 T \cdot {}^2_3 T \cdot {}^3_4 T \cdot {}^4_5 T, \quad (62)$$

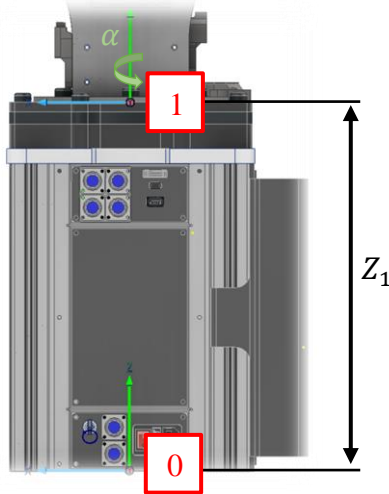
Ale pro vizualizaci celé konstrukce si vypočteme polohu všech kloubů. Tyto výpočty by se pak daly například použít pro zabránění kolize s okolím, kdy díky znalosti polohy všech hlavních částí robota, jsme schopni softwarově zamezit pohybu robota v určitých oblastech.



Obr. 77: Klouby manipulátoru

## Transformační matice pro hlavní klouby

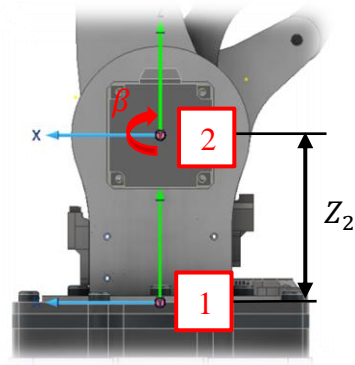
Vycházíme z počátku do výšky  $Z_1 = 372$  mm a rotujeme podél osy Z úhlem  $\alpha$ . Tento úhel se může měnit přibližně v rozsahu  $-175^\circ$  až  $175^\circ$ .



Obr. 78: Přejchod z 0 do 1

$${}^0_1T = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & Z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (63)$$

Posunujeme se do výšky hřídele motoru  $Z_2 = 140$  mm a rotujeme kolem osy Y úhlem natočení velkého ramena  $\beta$ . Tento úhel se může otáčet přibližně v rozsahu  $-80^\circ$  až  $60^\circ$  (šipka na obrázku ukazuje kladný směr otáčení)

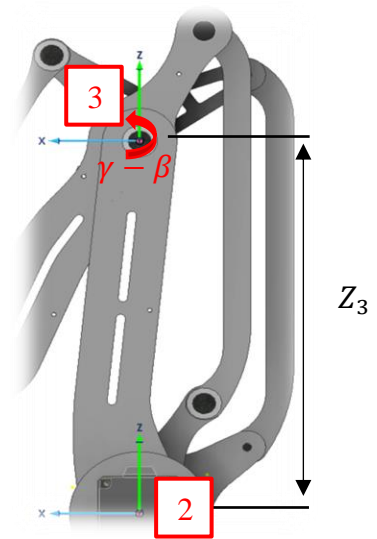


Obr. 79: Přejchod z 1 do 2

$${}^1_2T = \begin{bmatrix} \cos(-\beta) & 0 & \sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\beta) & 0 & \cos(-\beta) & Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (64)$$

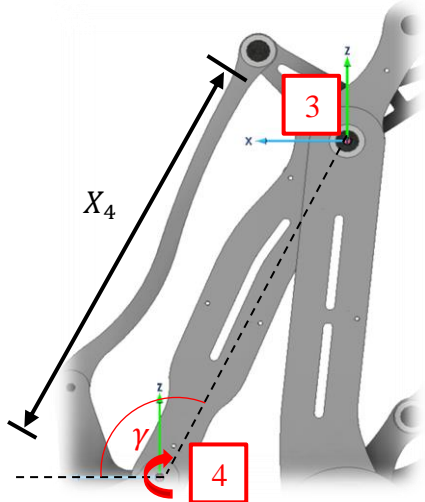
Posunujeme se do výšky kloubu o  $Z_3 = 450$  mm a rotujeme kolem osy Y úhlem  $\gamma - \beta$ . Proč zrovna tento rozdíl úhlů, to můžete vidět v **kapitole 5.1.1**, kde je tak vypočten úhel  $\theta_1$ . Jedná se totiž o rozdíl natočení obou ramen zároveň. Úhel malého ramene  $\gamma$  se může otáčet přibližně v rozsahu  $0^\circ$  až  $115^\circ$ .

$${}^2_3T = \begin{bmatrix} \cos\left(\frac{\pi}{2} - (\gamma - \beta)\right) & 0 & \sin\left(\frac{\pi}{2} - (\gamma - \beta)\right) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\left(\frac{\pi}{2} - (\gamma - \beta)\right) & 0 & \cos\left(\frac{\pi}{2} - (\gamma - \beta)\right) & Z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (65)$$



Obr. 80: Přejchod z 2 do 3

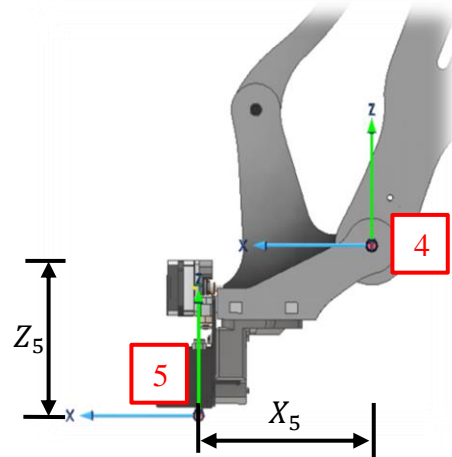
Posunujeme se v ose X o  $X_4 = 435$  mm , kde rotujeme kolem osy Y o úhel  $(\gamma - \frac{\pi}{2})$ . Jde zde o návrat k rovnoběžné poloze vůči podlaze.



Obr. 81: Přechod z 3 do 4

$${}^3_4T = \begin{bmatrix} \cos(\gamma - \frac{\pi}{2}) & 0 & \sin(\gamma - \frac{\pi}{2}) & X_4 \\ 0 & 1 & 0 & 0 \\ -\sin(\gamma - \frac{\pi}{2}) & 0 & \cos(\gamma - \frac{\pi}{2}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, (66)$$

Jako poslední je třeba započíst offset pro nástroj v hlavě manipulátoru. Pro 3D tisk, nebo plotter jsou offsety  $X_5 = 139.5$  mm a  $Z_5 = -129.43$  mm

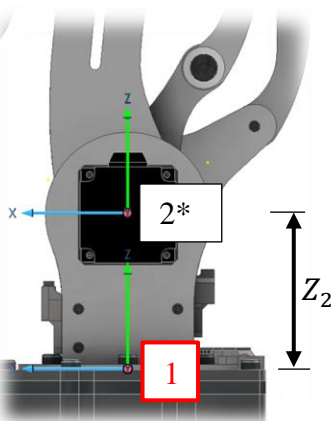


Obr. 82: Přechod z 4 do 5

$${}^4_5T = \begin{bmatrix} 1 & 0 & 0 & X_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}, (67)$$

### Transformační matice pro vedlejší klouby

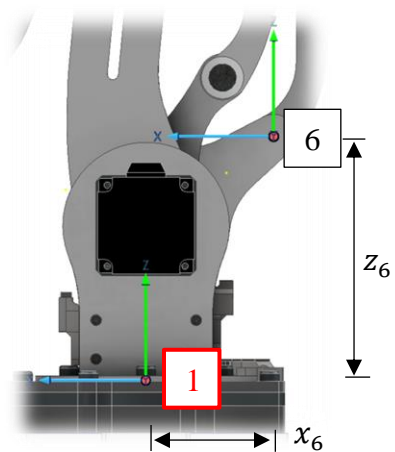
Posunujeme se o  $Z_2 = 140$ mm



Obr. 83: Přechod z 1 do 2\*

$${}^1_{2^*}T = \begin{bmatrix} \cos(-\gamma) & 0 & \sin(-\gamma) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\gamma) & 0 & \cos(-\gamma) & Z_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, (68)$$

Posunujeme se o  $x_6 = -114.6$  mm,  $z_6 = 220.3$  mm

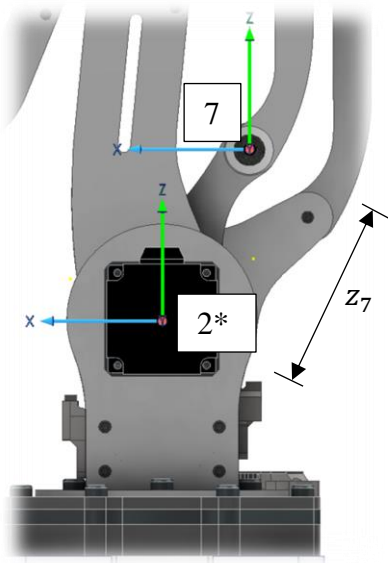


Obr. 84: Přechod z 1 do 6

$${}^1_6T = \begin{bmatrix} \cos(-\beta) & 0 & \sin(-\beta) & x_6 \\ 0 & 1 & 0 & 0 \\ -\sin(-\beta) & 0 & \cos(-\beta) & z_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}, (69)$$



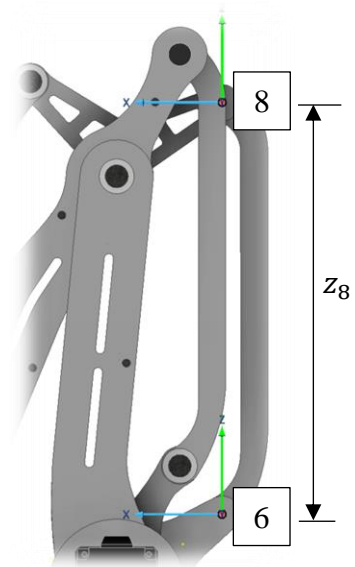
Posunujeme se o  $z_7 = 150$  mm



Obr. 85: Přechod z 2\* do 7

$${}_{7}^{2*}T = \begin{bmatrix} \cos(\gamma - \beta) & 0 & \sin(\gamma - \beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\gamma - \beta) & 0 & \cos(\gamma - \beta) & z_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (70)$$

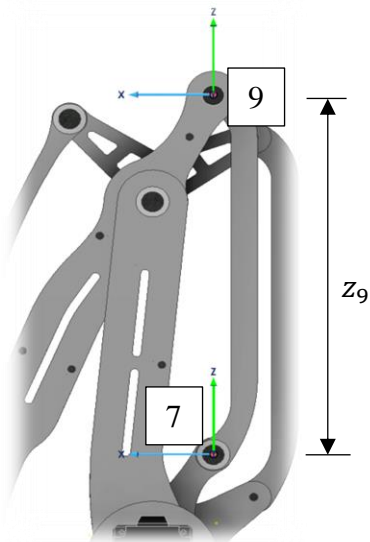
Posunujeme se o  $z_8 = 450$  mm



Obr. 86: Přechod z 6 do 8

$${}_{8}^{6}T = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & z_8 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (71)$$

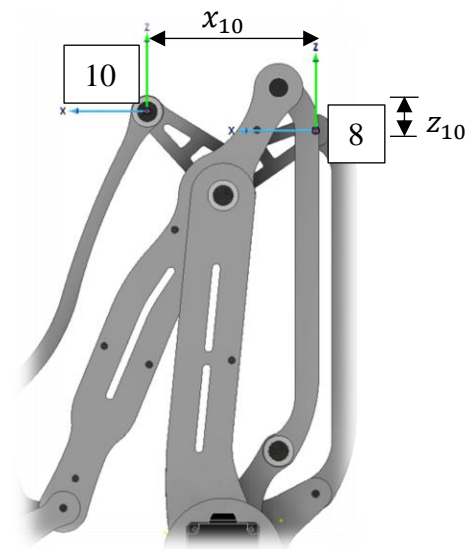
Posunujeme se o  $z_9 = 450$  mm



Obr. 87: Přechod z 7 do 9

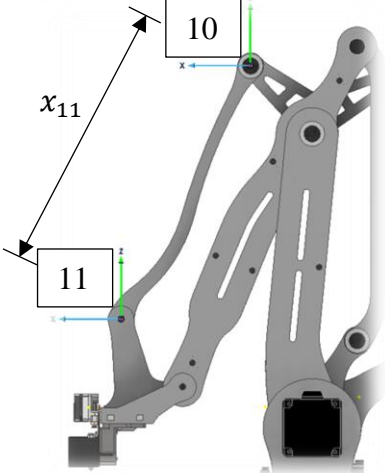
$${}_{9}^{7}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z_9 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (72)$$

Posunujeme se o  $x_{10} = 208.7$  mm,  $z_{10} = 23.3$  mm



Obr. 88: Přechod z 8 do 10

$${}_{10}^{8}T = \begin{bmatrix} \cos\left(\frac{\pi}{2} - \gamma\right) & 0 & \sin\left(\frac{\pi}{2} - \gamma\right) & x_{10} \\ 0 & 1 & 0 & 0 \\ -\sin\left(\frac{\pi}{2} - \gamma\right) & 0 & \cos\left(\frac{\pi}{2} - \gamma\right) & z_{10} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (73)$$

<p>Posunujeme se o <math>x_{11} = 435</math> mm</p>  <p>Obr. 89: Přechod z 10 do 11</p> ${}^{10}_{11}T = \begin{bmatrix} 1 & 0 & 0 & x_{11} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (74)$	<p>Souhrn vypočtených souřadnic kloubů:</p> <p>Hlavní klouby</p> <p>Kloub 1: <math>{}^0_1T = {}^0_1T,</math> (75)</p> <p>Kloub 2: <math>{}^0_2T = {}^0_1T \cdot \frac{1}{2}T,</math> (76)</p> <p>Kloub 3: <math>{}^0_3T = {}^0_1T \cdot \frac{1}{2}T \cdot \frac{2}{3}T,</math> (77)</p> <p>Kloub 4: <math>{}^0_4T = {}^0_1T \cdot \frac{1}{2}T \cdot \frac{2}{3}T \cdot \frac{3}{4}T,</math> (78)</p> <p>Nástroj: <math>{}^0_5T = {}^0_1T \cdot \frac{1}{2}T \cdot \frac{2}{3}T \cdot \frac{3}{4}T \cdot \frac{4}{5}T,</math> (79)</p> <p>Vedlejší klouby</p> <p>Kloub 6: <math>{}^0_6T = {}^0_1T \cdot \frac{1}{6}T,</math> (80)</p> <p>Kloub 7: <math>{}^0_7T = {}^0_1T \cdot \frac{1}{2^*}T \cdot \frac{2^*}{7}T,</math> (81)</p> <p>Kloub 8: <math>{}^0_8T = {}^0_1T \cdot \frac{1}{6}T \cdot \frac{6}{8}T,</math> (82)</p> <p>Kloub 9: <math>{}^0_9T = {}^0_1T \cdot \frac{1}{2^*}T \cdot \frac{2^*}{7}T \cdot \frac{7}{9}T,</math> (83)</p> <p>Kloub 10: <math>{}^0_{10}T = {}^0_1T \cdot \frac{1}{6}T \cdot \frac{6}{8}T \cdot \frac{8}{10}T,</math> (84)</p> <p>Kloub 11: <math>{}^0_{11}T = {}^0_1T \cdot \frac{1}{6}T \cdot \frac{6}{8}T \cdot \frac{8}{10}T \cdot \frac{10}{11}T,</math> (85)</p>
--	---

Veškeré výsledky byly zkontrolovány a přepočítány v Matlabu. Pro porovnání správnosti výpočtů byl použit 3D model v Inventoru, kde se dala změřit přesná vzdálenost. Tyto matice s reálnými hodnotami a výsledné pozice kloubů se pak zobrazují na obrazovce viz **kapitola 5.3.2**.

Při zadávání úhlů pro jednotlivé motory, je třeba dbát na jejich maximální rozsahy.

### 5.1.3 Inverzní kinematika numerickou metodou

Zde se zabýváme implementací numerické metody pro získání inverzní kinematiky při použití Newton-Raphsonovy metody [3]. Jedná se o iterační metodu založenou na řešení odhadu pro řešení sady nelineárních rovnic, což můžeme použít i pro vyřešení problému s inverzní kinematikou. [3] [4]

Pro jednoduchost a přehlednost v programu zde bude použit jazyk Python. Jako první bylo nutno určit startovní odhad pozice nástroje a požadavek na pozici.

```
theta_initial = np.array([0.0, 0.0, 27.3]) * (np.pi / 180)
#Home pozice:  $\alpha = 0^\circ$ ,  $\beta = 0^\circ$   $\gamma = 27.3^\circ$ ,
                x = 339.0126, y=0.0, z= 446.0215
PolohaNastroje = np.array([599.01, 350.0, 566.02])
```

*Kód 7: Inicializace numerické metody*

Poté potřebujeme funkci, která nám vypočte dopřednou kinematiku, odchylku chyby a Jakobián. Dopřednou kinematiku vezmeme z maticové metody v **kapitole 5.1.2**:

```
def forward_kinematics(theta):
# viz článek 5.1.2
return T05
```

*Kód 8: Funkce pro dopřednou kinematiku*

Výpočet chyby zvolíme jako:

```
def error(theta): # Regulacni odchylka
    VypPozice_sliced = forward_kinematics(theta)[:3, 3]
    error = cart2cyl(PolohaNastroje) - cart2cyl(VypPozice_sliced)
    return error
```

*Kód 9: Výpočet odchylky*

Jakobián z dopředné kinematiky vypočteme jako:

```
def jacobian(theta):
    PrubeznaPozice_sliced = forward_kinematics(theta)[:3, 3]
    J = np.zeros((3, 3))
    step = 1e-6 # "krok" diferenci
    for i in range(3):
        theta[i] += step # pridam krok k uhlu
        VypPozice_sliced = forward_kinematics(theta)[:3, 3]
        theta[i] -= step # odeberu krok
        deltaPoz = cart2cyl(VypPoz_sliced) - cart2cyl(PrubPoz_sliced)
        J[:, i] = deltaPozice / step
    return J
```

*Kód 10: Výpočet Jakobiánu*

Zde jako první vypočteme pozici z počátečního odhadu a poté tento výpočet zopakujeme, ale přitom přidáme malý „krok“ k uhlu. Hodnotu poté znovu odečteme, abychom se vrátili k originální hodnotě. Hodnotu rozdílu vydělíme krokem a výsledky pak uložíme do matice J. Tato funkce tedy používá metodu konečných rozdílů k aproximaci derivací k výpočtu Jakobiánu: [3] [4]

$$f' \approx \frac{f(x_0+h)-f(x_0)}{h} \approx \frac{\text{VypPozsliced}-\text{PrubPozsliced}}{\text{step}}, \quad (86)$$

Všechny části kódu jsou pak spolu volány ve funkci „inverse\_kinematics“, kde je před spuštěním ještě nutno zvolit počet iterací a velikost tolerance pro správný výsledek:

```
def inverse_kinematics(theta_initial):
    max_iterations = 1000 # Počet iterací
    tolerance = 0.01 # Tolerance požadované od aktuální iterace
    for i in range(max_iterations):
        start_time = time.time()
        J = jacobian(theta_initial)
        e = error(theta_initial)
        if np.linalg.norm(cyl2cart(e)) < tolerance:
            end_time = time.time()
            elapsed_time = end_time - start_time
            sprint("Uběhnutý čas:", elapsed_time, "s")
            print("Počet iterací", i)
            return theta_initial
            #Pokud je vypočtena v toleranci, tak hotovo
        try:
            delta_theta = np.dot(np.linalg.inv(J), e)
            #Ax=b => x=A-1·b ≈ delta_theta=J-1·e
        except np.linalg.LinAlgError:
            print("Singular matice error")
            #Pokud jsme zadali hloupost, tak error
            return None
        theta_initial += delta_theta
    print("Uběhly všechny iterace a nenašel výsledek")
    return None
```

Kód 11: Výpočet inverzní funkce

Pokud je hodnota vypočtené chyby menší než tolerance, tak je hotovo. Pokud ne, tak se připočítává k poslední použité hodnotě úhlu hodnota zohledňující chybu. Následný kód je pak otestován a výsledné úhly pro zadanou pozici vytištěny do konzole.

```
if __name__ == "__main__":
    theta_solution = inverse_kinematics(theta_initial)
    if theta_solution is not None:
        print("Uhly kloubu:", theta_solution*(180/np.pi))
        print("Pozice nastroje:", forward_kinematics(theta_solution)[:3, 3])
```

Kód 12: Main()

Vše bylo testováno v Pythonu, kde se navíc přidalo i měření času a počet iterací k dosažení požadované hodnoty. Kód je pak převeden do C++, v němž je psán firmware. Zde je zobrazen výsledek po spuštění skriptu v Pythonu:

```

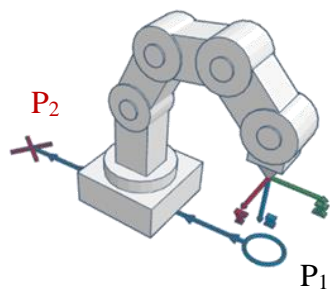
Run: InverzniKinematikaDH x
▶ Uběhnutý čas: 0.0009980201721191406 s
⚙ Počet iterací 4
☐ Uhly kloubu: [ 30.29763396 -24.06099338  58.47473865]
☐ Pozice nástroje: [599.00999968 349.99999981 566.01999922]
☐ Process finished with exit code 0
  
```

Obr. 90: Terminál: Výsledek skriptu pro inverzní kinematiku

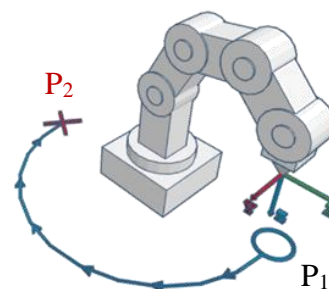
Je také důležité si všimnout, že v kódu jsou převáděny kartézské souřadnice na cylindrické. To bylo provedeno z důvodu, že pro určité případy tato metoda dobře nefungovala. Nejlépe to půjde vysvětlit na příkladu, kde:

Máme startovní bod  $P_1 [100; 0; 0]$  a cílový bod  $P_2 [-100; 0; 0]$ . Jedná se o posun nástroje v ose X o vzdálenost 200. Pokud jsou iterace prováděny v kartézských souřadnicích, tak metoda vyhodnotí jako nejkratší cestu možnost A (Obr. 92). To je však špatně, jelikož robot vůbec nebere v úvahu rotační stůl (úhel  $\alpha$ ) a rozhodne se pro dosažení cíle pouze změnit natočení ramen, čímž narazí do vlastní konstrukce.

Proto je nutné počítat v cylindrických souřadnicích, možnost B (Obr. 91), díky kterým metoda bude pracovat s rotací kolem osy Z a nabídne nám tak správné řešení. Nutno však podotknout, že i tato metoda má svá úskalí, a proto z důvodu použité funkce atan2 musíme pracovat se zadanými úhly v rozmezí  $-180^\circ$  až  $180^\circ$ . Také je zde potřeba dávat pozor na singulární body, ty je nutno ošetřit v softwaru.



Obr. 92: A) Numerické řešení s kartézskými souřadnicemi



Obr. 91: B) Numerické řešení s cylindrickými souřadnicemi

## 5.2 Řídící firmware

Jak už bylo psáno v **kapitole 2.3.6** na konci, tak volba pro robotický manipulátor byla Marlin psaný v jazyce C++. Postup byl takový, že jako první se vytvořil vlastní kód pro řízení motorů a komunikaci ve směru řídící deska ↔ počítač. Zde se testovaly po sériové lince zprávy ve formě G-kódu (kombinace znaků a čísel), podle kterých deska dostávala příkazy a vykonávala tak svůj kód. Tímto způsobem byla otestována nejen funkčnost komunikace, ale i správnost funkčnosti ovládní motorů a kinematika. Taky se tímto způsobem dal odchytnout správný formát, v jakém se mají data z grafického rozhraní posílat. Poté bylo zapotřebí implementovat vlastní kód do Marlinu.

### 5.2.1 Generování firmwaru s Visual Studio Code + Platformio

Pro úpravu Marlinu byla použita kombinaci softwaru Visual Studio Code ve spojení s pluginem Platformio IDE „Integrated Development Environment“.

Jedná se o oblíbenou kombinaci pro programování vestavěných systémů. Visual studio je klasické vývojové prostředí pro psaní v C++, ale pomocí Platformia, je zde možnost kód integrovat do různých typů řídicích desek (Arduino, Esp32, Stm32, ...).



Obr. 93: Visual Studio Code + Platformio IDE logo [37]

Z webu Marlinu byla použita tehdy poslední dostupná verze Marlinu, která v tu dobu byla „2.1.2“. Tato verze pak byla naimportována do složky, kde se vše otevřelo ve Visual Studio Code - Platformio IDE [37].

Zde byl jako první nadefinován konfigurační soubor „platformio.ini“, kde se zvolila použitá deska „BIGTREE\_OCTOPUS\_PRO\_V1\_F429“ a další potřebné parametry. Stejná deska se pak musela nadefinovat i v konfiguračním souboru pro Marlin „Configuration.h“. V tuto chvíli byl kód připraven pro generování.

Pokud se tedy povedlo zkompilevat celý kód bez chyby, stačilo už jen stisknout tlačítko Platformio IO Build a vygeneroval se nám „firmware.bin“ soubor, který se nahrál na SD kartu, vložil do STM desky a nabootoval ji. Tento proces bylo nutno dělat pokaždé, když došlo ke změnám ve firmwaru. Existuje i jiný způsob bootování desky, a to přes USB kabel, ale pro použitý typ desky nebyla tato možnost povolena.

```

Advanced Memory Usage is URL via "PlatformIO Home > Project Inspect"
RAM:  [=          ]  7.4% (used 9732 bytes from 131072 bytes)
Flash: [=         ] 11.7% (used 123188 bytes from 1048576 bytes)
Building .pio\build\BIGTREE_OCTOPUS_PRO_V1_F429\firmware.bin
===== [SUCCESS] Took 292.00 seconds =====

Environment          Status      Duration
-----
BIGTREE_OCTOPUS_PRO_V1_F429  SUCCESS    00:04:52.000
===== 1 succeeded in 00:04:52.000 =====

```

Obr. 94: Terminal: Report z konzole po úspěšném buildu firmwaru

Takto pak vypadá (Obr. 94), když se povede úspěšně vygenerovat kód pro řídicí desku. Platformio má přitom v sobě uložena data o různých deskách a díky informacím, které při změně konfiguračního souboru uděláme, je nám schopen říct, kolik například zabíráme paměti.

## 5.2.2 Úpravy Marlinu

Marlin se s desetiletým vývojem obrovsky rozrostl a v dnešní době obsahuje plno funkcionalit. Má spousty typů kinematik (Kartézské, Delta, Scara, CoreXY), ale některé z nich ještě nejsou oficiálně vydány a jsou pouze ve formě testování. Toto se týká také kinematiky pro robotický manipulátor, tu bylo nutno napsat a implementovat ji do firmwaru. První základní úpravou, kterou musí udělat každý uživatel Marlinu, je editace dvou konfiguračních souborů Configuration.h a Configuration\_adv.h. Jedná se o soubory, kde se definují požadavky na vlastnosti používané tiskárny. [14]

### Configuration.h

- Volba desky
- Sériový port, baudrate
- Kinematika, drivery
- Počet motorů, extruderů
- Endstopy, LCD, teploty

### Configuration\_adv.h

- Definice použitých senzorů a driverů
- Min a Max teploty
- PID
- Homing, BI-Touch senzor
- Input shaping

Důležité je však upozornit, že jsou zde pouze definována povolení pro dané funkce, tudíž pouhá volba, zda je chceme použít a jak, nebo při jakých rozsazích mají pracovat. Samotná funkcionalita je pak psána v dalších samostatných cpp souborech. Hlavní loop (programová smyčka) pak probíhá v MarlinCore.cpp, který vykonává všechny použité podprogramy. Jeho hlavním úkolem je také neustále sledovat nově

příchozí G-Kód požadavky a postupně je vykonávat. Ty jsou všechny vypsány v gcode.cpp souboru a formou „case“ selekce vybírány.

V našem případě byl tedy prvně nakonfigurován firmware tak, aby splňoval všechny potřebné funkce. Zde bylo nutno vytvořit cpp soubor pro výpočet kinematiky. Poté bylo nutno provést změny v různých částech Marlinu, které se týkaly řízení pohybu. Při programování kinematiky zde bylo použito už funkčních datových struktur, které se zde nacházely: cartes, delta. Kromě zásahů do kinematiky a celkově ovládání pohybů, byly upraveny některé části G-kódů a testovány vlastní příkazy.

Ukázka funkce dopředné a inverzní kinematiky napsané v C++ pro Marlin:

```
void forward_kinematic(const_float_t a, const_float_t b, const_float_t c)
{
    const double alfa = RADIANS(a),
                beta  = M_PI/2 + RADIANS(b),
                gama  = M_PI/2 + RADIANS(c);
    cartes.x = cos(alfa)*(cos(beta)*k2 + sin((gama - beta) - (M_PI/2 -
                beta))*k3+lx);
    cartes.y = sin(alfa)*(cos(beta)*k2 + sin((gama - beta) - (M_PI/2 -
                beta))*k3+lx);
    cartes.z = k1 + sin(beta)*k2 - cos((gama - beta) - (M_PI/2 -
                beta))*k3 -(lz);
}
```

*Kód 13: Dopředná kinematika - analyticky pro C++*

```
void inverse_kinematic(const xyz_pos_t &raw)
{
    double alfa, alfares, beta, gamma, received_x, received_y, received_z,
           d1, d2, beta1, beta2, o1;
    const xyz_pos_t spos = raw;
    received_x = spos.x;
    received_y = spos.y + 0.0001;
    received_z = spos.z + lz;
    alfares = atan2(received_y, received_x);
    d1 = (received_y/sin(alfares)) -lx;
    d2 = sqrt(d1*d1 + (received_z-k1)*(received_z-k1));
    beta1 = (acos((k2*k2+d2*d2-k3*k3)/(2*k2*d2)))*(180/M_PI);
    beta2 = (asin((received_z - k1)/(d2)))*(180/M_PI);
    o1= (acos((k2*k2-d2*d2+k3*k3)/(2*k2*k3)))*(180/M_PI);
    alfa = alfares *(180/M_PI);
    beta = beta1 + beta2 - 90.0;
    gamma = beta1 + beta2 + o1 - 90.0;
    delta.set(alfa, beta, gamma);}
}
```

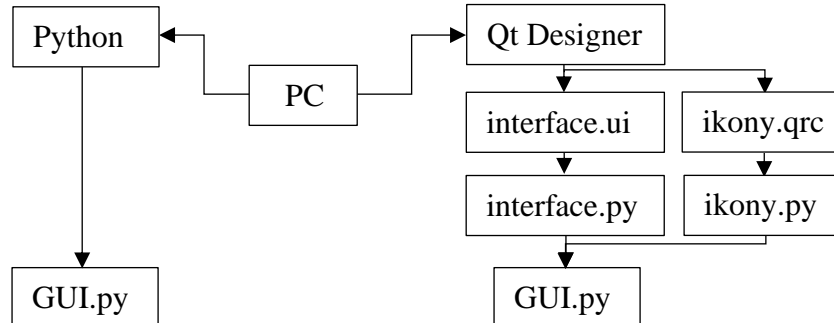
*Kód 14: Inverzní kinematika - analyticky pro C++*



## 5.3 Tvorba řídicího GUI

Při tvorbě GUI „Graphic User Interface“ bylo zapotřebí něčeho, kde půjde lehce přidávat různé typy prvků (jako jsou tlačítka, posuvníky...) a půjde odtud posílat příkazy do řídicí desky. Možností na trhu je opravdu mnoho a primárně se odvíjí od toho, v jakém jazyce chce člověk dané GUI programovat. V našem případě padla volba na Python, tedy přesněji knihovnu Pyside2. Jedním z důvodů této volby byla možnost použití jejich IDE prostředí s názvem Qt Designer při tvorbě grafického rozhraní. [38]

Existují dvě cesty, jak vytvořit GUI skrze knihovnu PySide2. První variantou je, že pokud tvoříme jednoduché a malé grafické ovládání, tak si můžeme vše naprogramovat rovnou v Pythonu. Druhá varianta je, že chceme dělat komplexnější ovládání, a tak navrhne grafiku v prostředí Qt Designer, ten vytvoří soubor „ui“ (user interface), který obsahuje jednotlivé prvky obrazovky a ten poté převedeme do Pythonu. V obou variantách musíme jednotlivé prvky a jakékoliv akce s nimi doprogramovat následně v kódu, ale díky Qt Designeru, jsme schopni ušetřit čas na tvorbě vzhledu. Velké plus je také, že jsme tak schopni kdykoliv jednoduše celkový vzhled upravit, stačí pak pouze přegenerovat „ui“ soubor a náš vlastní kód nebude nijak narušen. [38]



Obr. 95: Posloupnost při tvorbě GUI

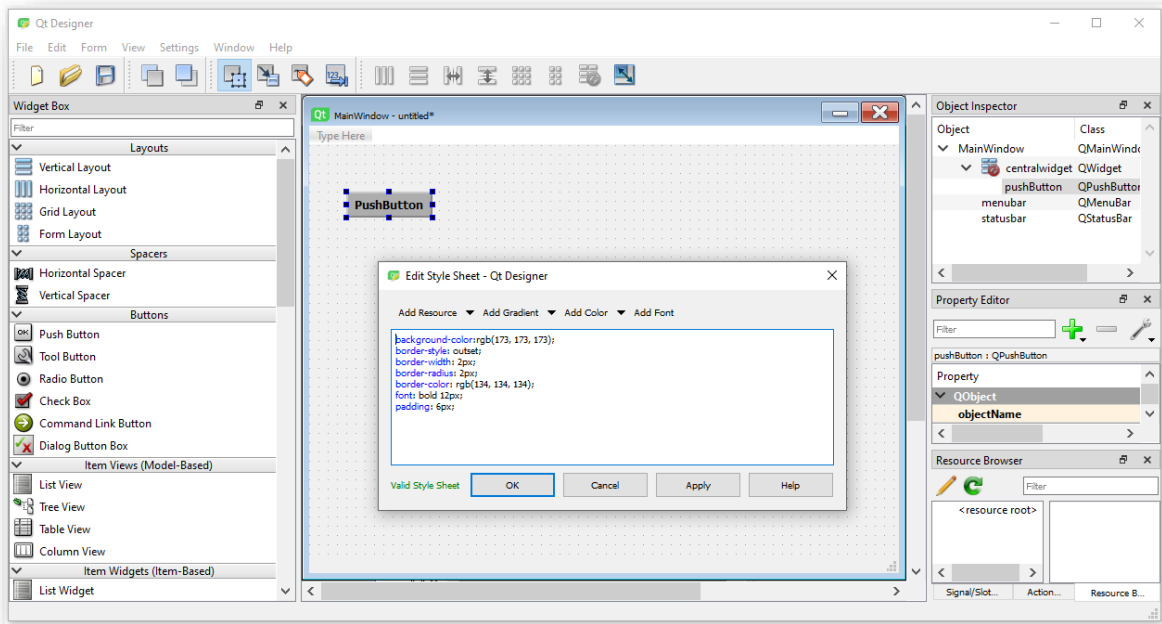
Postup je tedy následující:

- Po dokončení práce v designeru uložíme projekt ve formátu „interface.ui“. Tento formát s koncovou „ui“ v sobě ukládá konfiguraci obrazovek v XML formátu.
- Tento formát pomocí souboru „pyrcc5.exe“ převedeme do Pythonu, čímž se nám vytvoří soubor „interface.py“. Stejný postup pak musíme provést i pokud používáme v obrazovkách vložené obrázky, či ikonky. Jediný rozdíl je, že ikonky se ukládají ve formátu ikony.qrc a pro převedení používáme „pyuic5.exe“
- Poté už jen do našeho kódu naimportujeme soubory skrze příkazy:

```
from interface import *
import ikony
```

### 5.3.1 Tvorba samotných obrazovek skrze QT Designer

Jedná se o nástroj pro vývoj grafického uživatelského prostředí „GUI“. Umožňuje uživateli pracovat s designem přístupem „přetáhni a polož“, kde tedy rozmístíme po obrazovce objekty, které chceme používat. Je zde podpora všemožných prvků, jako jsou tlačítka, popisky, tabulky a mnoho dalšího. Pro nastavování vzhledu jednotlivých prvků, jako je například barva, písmo, se používá CSS kód, který lze psát přímo v tomto editoru.



Obr. 96: QT Designer editor

Jedná se však pouze o editor, jenž nám pomáhá s tvorbou vizuálního vzhledu obrazovky. Toto grafické rozhraní pak musíme převést do námi používaného (aplikací podporovaného) kódu (C++, nebo Python) a zde s ním dále pracovat.

Tento postup pak přináší obrovskou výhodu pro rychlou tvorbu, „prototypování“, obrazovek. Jelikož je hned přímo vidět vzhled jednotlivých prvků, tak jde na nich dělat okamžité úpravy, potom stačí přegenerovat „ui“ soubor. Náš kód s jednotlivými prvky přitom zůstává nezměněn. Chce to sice určitou praxi a trpělivost, naučit se s tímto designerem pracovat, ale velmi dobře urychluje práci.

Další věcí je pak programování fungování jednotlivých prvků, jelikož po vygenerování z editoru je vše ve stavu statických objektů, které nic nedělají a neumí. K tomu je zapotřebí nainportovat do Python kódu knihovnu PySide2. Ta umí přistupovat k jednotlivým objektům (jejich vlastnostem), které jsou v daném .ui souboru a dle typu s objektem pracovat. [38]

Ukázka nastavení tlačítka pro spuštění tisku:

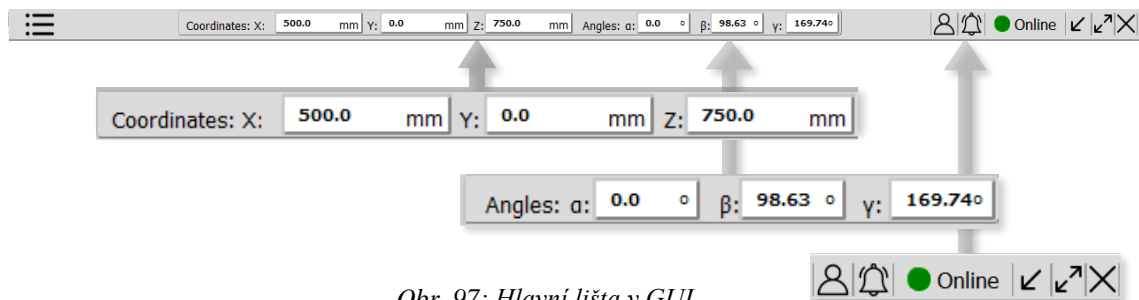
```
self.ui.StartPrint.clicked.connect(lambda: self.startprint())
self.ui.StartPrint.pressed.connect(lambda: self.ui.StartPrint.setStyleSheet("background-color:rgb(123, 123, 123)"))
self.ui.StartPrint.released.connect(lambda: self.ui.StartPrint.setStyleSheet("background-color:rgb(173, 173, 173)"))
self.ui.StartPrint.setCursor(QCursor(QtCore.Qt.PointingHandCursor))
```

*Kód 15: Click event tlačítka*

Funkcionalita je zde snadná, na kliknutí tlačítka se spustí skript, který pošle příkaz do řídicí desky (startprint()). Při pohybu myši nad tlačítkem se ukáže jiný kurzor a při stisku dochází k změně barvy. Název objektu tlačítka je „StartPrint“ a „ui“ je odkaz na celou třídu (okno obrazovky). Je zde tedy možnost přistupovat ke všem objektům, které spadají pod tuto třídu a parametrizovat je. Pokud je tedy zapotřebí změna barvy či názvu, vše lze v programu předělat, ale pro odlehčení kódu je lepší dělat grafické úpravy vzhledu v designeru a v hlavním kódu se soustředit pouze na skripty.

### 5.3.2 Hlavní lišta a funkcionality grafického ovládání

Základem je záhlaví, kde je vypsána aktuální poloha nástroje manipulátoru a úhly natočení jednotlivých motorů (jsou zde zobrazeny úhly  $\alpha$ ,  $\beta$ ,  $\gamma$  viz kapitola 5.1.1). Také je zde status, zda je počítač připojen k řídicí desce, či nikoliv. Rozkliknutí ikonky menu na levé straně záhlaví pak vyjede popis jednotlivých tlačítek v menu



*Obr. 97: Hlavní lišta v GUI*

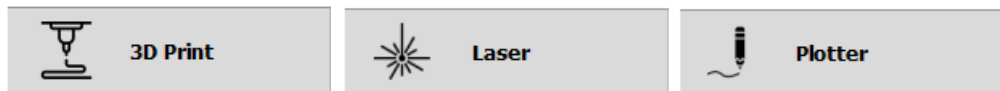
GUI manipulátoru také obsahují plno funkcionalit:

- Možnosti základní práce s oknem (zavření, rozšíření, schování, možnost přesouvat okno po obrazovce).
- Menu s vyjízďjícími názvy, a tedy několik obrazovek pro jednotlivé sekce.
- Sériová komunikace s řídicí deskou (vyhledávání partnera).
- Vykreslování grafů (model manipulátoru, sledování nástroje).
- Ovládání jednotlivých pohonů a jejich parametrizace.
- Pozicování rychlé i lineární.

### 5.3.3 Tlačítka v menu s obrazovkami

Takto pak vypadají jednotlivé položky v menu i s názvy. Menu se dělí na čtyři kategorie

- Pracovní obrazovky.



Obr. 98: Tlačítka v MENU pro práci

Ty jsou používány pro určitou automatickou činnost manipulátoru, kde stačí pouze vybrat patřičný model a spustit proces.

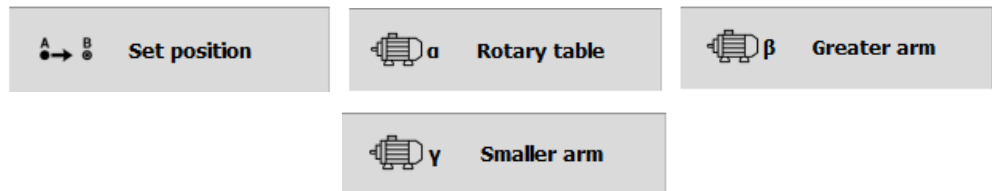
- Simulační obrazovky.



Obr. 99: Tlačítka v MENU pro simulaci

Ty slouží jen jako ukázka naprogramované kinematiky a geometrie stroje v souřadnicích. Nachází se zde simulovaný 3D model manipulátoru.

- Servisní obrazovky.



Obr. 100: Tlačítka v MENU pro servis

Jsou k ovládání manipulátoru, a to jak pro přesné pozicování, tak i pouhé natočení jednotlivých úhlů motorů.

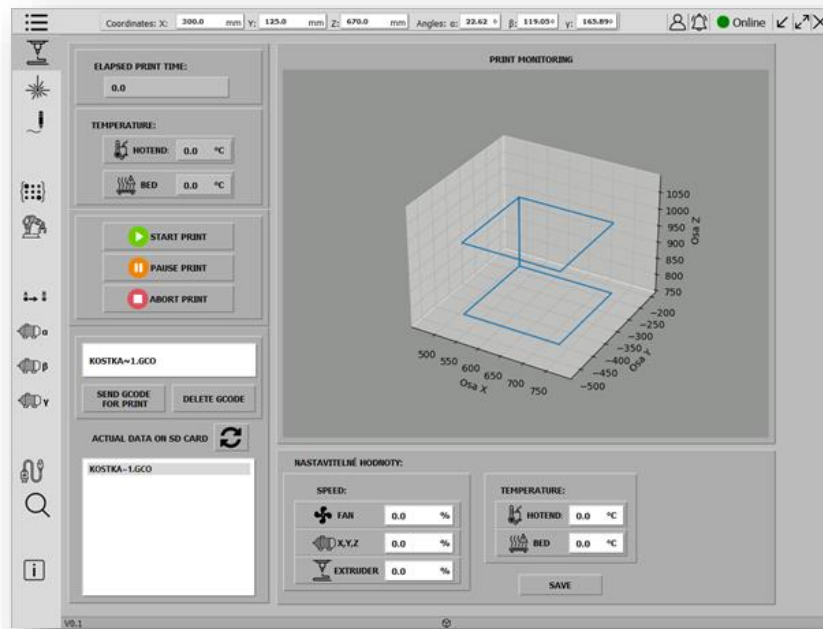
- Komunikační obrazovky.



Obr. 101: Tlačítka v MENU pro komunikaci

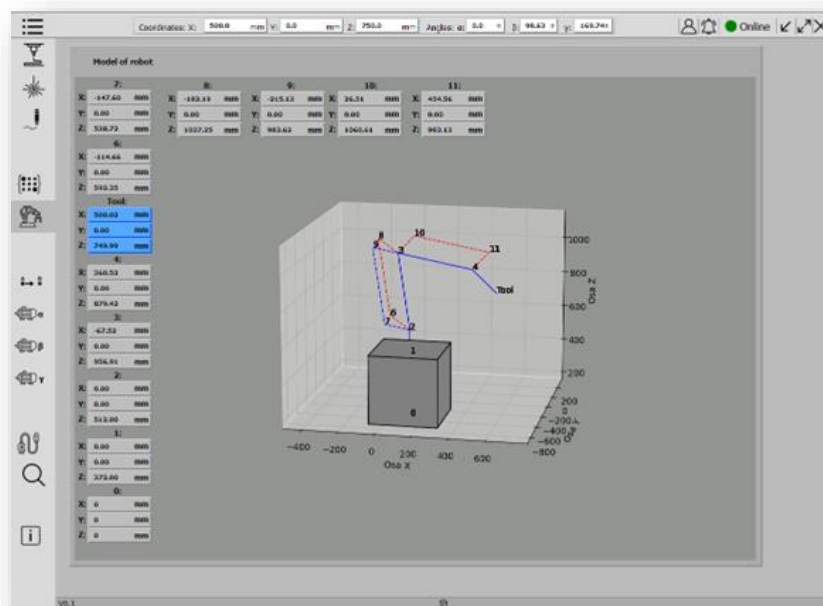
Tyto obrazovky jsou určeny k připojení řídicí desky s GUI přes sériovou linku a monitorování sériové linky.

Obrazovka pro ovládání 3D tisku obsahuje možnost číst z SD karty vložené do robota, načtež z ní můžeme zvolit model, který chceme tisknout a začít tisk. Tisk lze sledovat v 3D grafu, kde vykresluje pohyby nástroje a také můžeme ovlivnit nastavení rychlosti akčních členů či teplot.



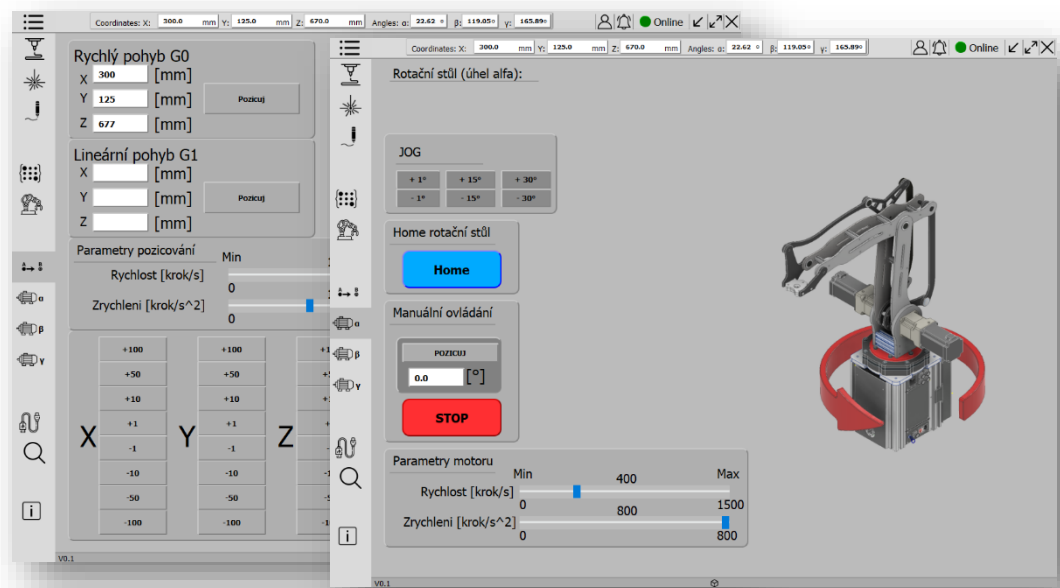
Obr. 102: Obrazovka pro ovládání 3D tisku

Další důležitou obrazovkou je zobrazení jednotlivých pozic kloubů manipulátoru z vypočtených matic (obrazovka se obnovuje co jednu sekundu, dle aktuálních hodnot). Z důvodu ladění a diagnostiky jsou zde zobrazeny i jednotlivé transformační matice, z kterých byly souřadnice robota vypočteny.



Obr. 103: Obrazovka s 3D modelem manipulátoru

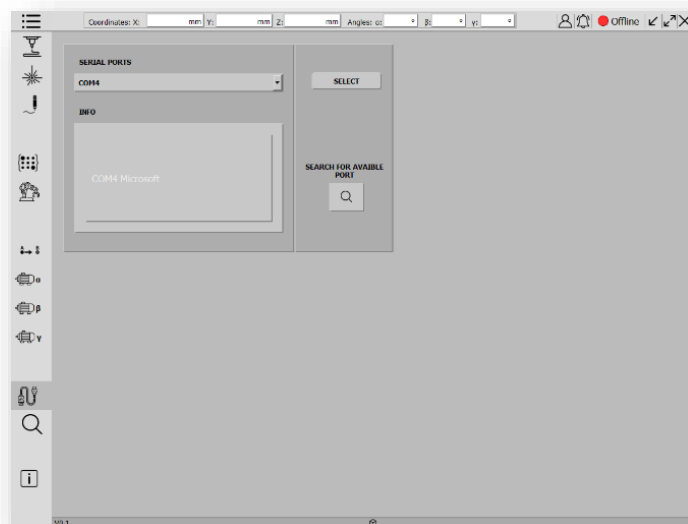
V neposlední řadě jsou zde obrazovky pro ovládání pohybů, a to jak jednotlivých motorů zvlášť, tak i společným pozicováním. Pozicování se pak dělí na rychlé, tedy příkazem G0, anebo linearizované, s příkazem G1.



Obr. 104: Obrazovky pro řízení pohybu

### 5.3.4 Komunikace s Marlinem skrze click eventy

Jelikož je veškerá komunikace GUI s řídicí deskou manipulátoru provozována přes sériovou linku, tak je vytvořena obrazovka pro navázání této komunikace, kde po stisku „lupy“ volá skript, který hledá možné partnery pro komunikaci na pevně nastavené rychlosti 115200 Bd. Pokud skript někoho takového najde, tak vhodné kandidáty vypíše do boxu „Serial ports“ a do info vypíše parametry. Po kliknutí na SELECT pak naváže komunikaci s řídicí deskou. Pokud je komunikace navázána, tak v horní liště zezelená puntík a status „Offline“ se přepíše na „Online“.



Obr. 105: Obrazovka pro navázání komunikace

Jak už bylo zmíněno v **kapitole 5.3.1**, tak jednotlivé objekty, „tlačítka“, mají k sobě přiřazeny `clicked.events()`, kterými spouští skripty. Pro náš případ komunikace s řídicí deskou existují dva takové typy skriptů. První typ běží v časové smyčce s určitým vzorkováním. Ten slouží k vyčítání dat z desky. Jsou to například polohy jednotlivých motorů manipulátoru, teploty, rychlosti atd. Druhý typ skriptu slouží pro zápis dat do desky a ten je prováděn právě přes `clicked.events()` jednotlivých tlačítek. Veškerá čtení či zapisování dat jsou prováděna skrze G-CODE příkazy.

Zde je ukázka skriptů použitých při stisku tlačítka pro zápis a čtení:

```
# Čtení - Soubory na SD kartě – příkazu desce vyčíst data z vložené SD karty a uložit si prvních 8
hodnot, co mi pošle, filtruji, aby zpráva obsahovala správnou koncovku; pokud ano, tak zapisuji;
nakonec vypisuji v seznamu GUI.

def showlist(self):
    self.ser.write(('M21\n').encode())
    self.ser.write(('M20\n').encode())

    gcode = []
    for i in range(8):
        gcode.append(self.ser.readline().decode())

    filename = []
    substring = ".GCO"
    gcode_vars = gcode

    for gcode in gcode_vars:
        if substring in gcode:
            parts = gcode.rsplit(' ', 1)
            filename.insert(1, parts[0])

    self.ui.ListOfGcodes.clear()
    self.ui.ListOfGcodes.addItem(filename)
    self.ui.ListOfGcodes.itemClicked.connect(self.on_itemClicked)
```

*Kód 16: Skript pro čtení*

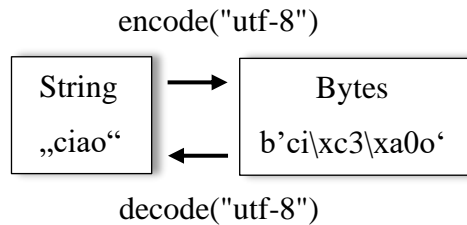
```
# Zápis - Pozicování manipulátoru – přečtu hodnoty zadané v GUI a pošlu po sériové komunikaci ve
tvaru, kterému bude deska rozumět

def MotorPositionXYZ(self):
    PosX = self.ui.PoziceX.text()
    PosY = self.ui.PoziceY.text()
    PosZ = self.ui.PoziceZ.text()
    Pozspeed = self.ui.RychlostPozicovani.value()
    Pozaccel = self.ui.ZrychleniPozicovani.value()

    self.ser.write(('M204 P{} T{}\n').format(Pozaccel, Pozaccel).encode())
    self.ser.write(('G0 X{} Y{} Z{} E0 F{}\n').format(PosX, PosY, PosZ, Pozspeed).encode())
```

*Kód 17: Skript pro zápis*

V jednotlivých řádcích skriptů lze vidět, jak se pokaždé odkazují na „ui“, to je pro připomenutí hlavní okno, na kterém se jednotlivé prvky nacházejí. Před odesláním textového řetězce („string“ zprávy) po sériové komunikaci musíme zprávu „zakódovat“ pomocí příkazu `encode()`. Tento příkaz převádí string hodnoty do balíku bajtů. Pokud nebyla vybrána metoda převodu `encode()`, tak se defaultně volí UTF-8. [38]



*Obr. 106: Ukázka kódování zprávy přes utf-8*



## 6. Testování manipulátoru

Samotné testování se provádělo po částech. Jako úplně první se provedla zkouška řízení pohonů přes drivery a testování různých rychlostí a zrychlení.

Jakmile už byla větší jistota, že se nic nezničí, tak byly motory připojeny k převodovkám na manipulátoru a začaly se zkoušet pohyby s robotem (v tuto chvíli ještě s papírovým modelem, viz kapitola 6.2). Celý kód, který byl v tuto chvíli pro řízení používán, byl zatím mimo Marlin, ale principiálně fungoval obdobně. Do manipulátoru byly po sériové lince posílány jednotlivé příkazy a robot se dle toho pohyboval.

Po otestování kinematiky a ujištění se, že robot do ničeho nenabourá, byly nahrazeny kartónové díly těmi z duralu a začalo se s přesnějším pohybováním. Byly vytvořeny fixační prvky, kterými po jejich instalaci na rám robota byla určena přesná poloha nástroje robota. Poté byl implementován kód do Marlinu a začalo se debuggovat. Zprvu pozicování probíhalo s prázdnou nástrojovou hlavou. Poté byl zprovozněn mechanismus výměnné hlavy a už byl manipulátor schopen držet tužku a vykreslovat čáry. Ty zpočátku rovné nebyly, a tak bylo nutné optimalizovat linearizaci pohybu. Poté se testoval 3D tisk a zkoušely se další testy při různých změnách rychlostí a parametrů manipulátoru.

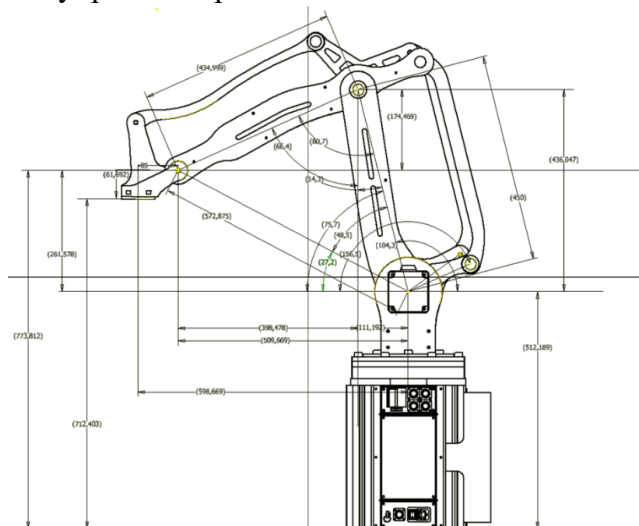
### 6.1 Volba softwaru pro testování kinematiky

Pro návrh a výpočet kinematiky manipulátoru byl převážně používán Matlab a Inventor.

V Inventoru vytvořené výkresy sloužily jako vzor správných výsledků pro různá náhodná nastavení úhlů, a to se pak porovnávalo s výpočty dopředné, nebo inverzní kinematiky z Matlabu.

Stejným způsobem se také postupovalo při určování homogenních matic pro dopřednou kinematiku, kde se krok po kroku kontrolovalo, zda nedošlo k chybě.

Možnost této zpětné kontroly vůči 3D modelu tak značně urychlila proces kontroly správnosti výpočtů kinematiky.



Obr. 107: Výkres v Inventoru

## 6.2 Papírový model

Z tvrdého kartónu byly vyřezány v měřítku 1:1 jednotlivé díly horní části robota, abychom tak při testování neriskovali poškození stroje nebo někoho nezranili. Nejednalo se o testování přesnosti pohybů či pozicování, ale dalo se zde dobře zjistit, zda je v kódu někde chyba a co je třeba dodělat.

Samozřejmě to také byla první ostrá kontrola, zda byla celková konstrukce dobře navržena, jelikož do té doby se s manipulátorem hýbalo pouze jako s modelem v počítači.

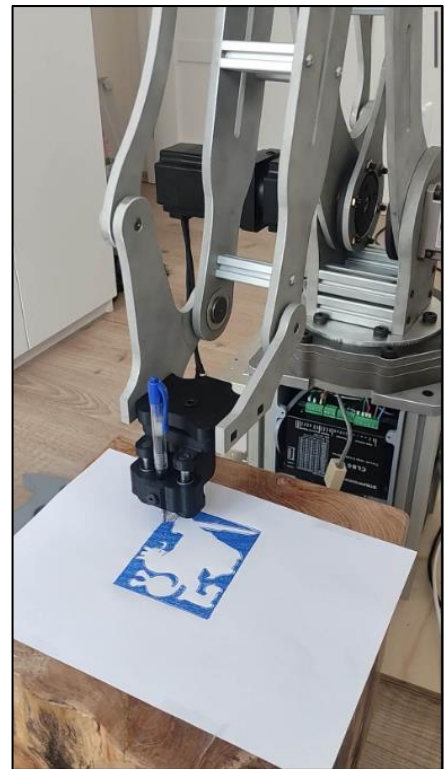


Obr. 108. Kartónový model manipulátoru

## 6.3 Zkoušení kreslení a 3D tisku

Jako první ostrý test robota s nástrojem bylo použito kreslení koleček a čtverců. Na tomto úkolu se daly dobře ukázat lineární a nelineární pohyby robota v kombinaci s různými rychlostmi. Také zde šly dobře odpozorovat různé neduhy a možnosti pro zlepšení tuhosti celkové konstrukce. Tou důležitější částí testování pak už bylo vyzkoušet robota na komplexnějším úkolu. V tomto případě to bylo nakreslit znak ČVUT v malých rozměrech, a to 10 x 8 cm.

Zde se ukázala důležitost volby povrchu, jelikož dřevěný kvádr neměl rovný povrch a volba papíru taky nebyla úplně ideální. Z důvodu tenkosti papíru se brzo napil inkoustu z propisky a začal se vlnit. To způsobilo, že než byla vychytána dobrá vzdálenost propisky od papíru, tak se povedlo roztrhnout tři papíry. Další výkres už pak byl prováděn na stěně od akvária, což se ukázalo jako mnohem lepší pracovní plocha.



Obr. 109. Testování plotteru

Poté bylo třeba s manipulátorem otestovat další nástroj, a to 3D tisk. Při tomto úkolu vzniklo hned několik výzev. Jako první bylo nutno dobře zvolit rychlosti jednotlivých motorů a extruderu. Pak bylo nutné zajistit co nejvíce rovnou podložku, na



*Obr. 110. Testování 3D tisku*

Výtisk se prováděl s červeným PLA „Polylactic Acid“ a tryskou 0,4 mm a výškou vrstvy 0,2 mm. Celkové nastavení a rychlosti byly nechány takové, jaké byly ve sliceru pro klasickou kartézskou tiskárnu. Tato lodička o rozměrech 6 cm x 3 cm x 4 cm se tiskla přibližně hodinu a půl. G-code pro modely, jak pro malování, tak pro 3D tisk byl vždy generován v PrusaSliceru, kde se dá vytvořit vlastní plocha sliceru kruhového tvaru, která tak odpovídá pracovní oblasti robota.

kteřé by šlo tisknout. První plán byl, že se bude tisknout na vyhřívané podložce od klasické 3D tiskárny, ta však byla vytížena tiskem dalších potřebných dílů, a tak se zkušební tisk prováděl na skleněném akváriu.

Sklo má díky způsobu výroby tažením velice rovnou plochu, ale z důvodu větších rozměrů akvária docházelo v jeho středu k určitému propadu. Sklo bylo také potřeba natřít obyčejným lepidlem z tuby z papírnickví pro lepší adhezi.

Při výtisku bylo důležité vytisknout plochou přídržnou vrstvu (tzv. „brim“), která držela výtisk u skla, jelikož bez vyhřívané podložky má plast při chladnutí tendenci se kroutit a odlepovat od podložky.

## 6.4 Testování přesnosti a opakovatelnosti

Za pomoci třetího nástroje, číselníkového úchylkoměru, byla měřena opakovatelnost pozice, když robot jel z pozice A do pozice B a zpátky. Bohužel nebylo k dispozici žádné digitální zařízení pro přesné měření pozice, a tak bylo vše odečítáno z analogového kruhového číselníku. Tento test se prováděl nejprve pro ověření přesnosti při výměně nástrojů a potom pro ověření stability výšky Z při maximálním natažení ramene.

Výsledek byl takový, že uchycení nástroje má velice dobrou opakovatelnost, jelikož ručka budíku se při opakovaném pozicování téměř nehýbe. Možný problém však nastává, když pohybujeme s ramenem v plném natažení. Zde dochází k poklesu výšky v ose Z přibližně v řádech desetin milimetru při plném vyložení ramene. Příliš přesný údaj se však nepodařilo získat, jelikož samotné akvárium, na kterém byl tento test prováděn, nemělo příliš rovnou plochu.



*Obr. 111. Měření přesnosti a odchylek*

## 7. Závěr

Jako první jsem provedl rešerši dostupných a používaných konstrukcí v aditivních technologiích a robotice všeobecně, abych našel ideálního kandidáta pro svůj manipulátor. Zvolil jsem si tak robotické rameno s principy paralelogramu, které splňovalo mé požadavky na velkou pracovní plochu a možnost výměnného nástroje.

V druhé části jsem si prošel dostupné open-source nabídky softwaru pro řízení manipulátorů a 3D tiskáren. Bylo nutné najít takový, který by se dal ideálně napasovat na můj stroj. Zde jsem si zvolil Marlin, jelikož se jednalo o v dnešní době nejpoužívanější firmware pro 3D tiskárny a díky dostupnosti zdrojového kódu a volné licenci, jsem ho mohl upravit tak, aby fungoval i s mým robotem.

V třetí části, kterou bych nazval konstrukční, jsem navrhl 3D model robotického manipulátoru s výměnnou hlavou a nástroji. Jednotlivé díly pro robota jsem si pak nechal vyrobit a robota zkonstruoval. Součástí toho byla i volba správných ložisek, převodovky a spojky. Vše se mi povedlo obstarat a manipulátor celý funkčně složit.

Čtvrtá část se zabývala návrhem elektrické části robota, kde jsem zvolil veškerou potřebnou elektroniku a pohony. Bylo tak nutno nejen zvolit správné součástky, ale také je spolu propojit a uspořádat v základně robota. Po nákupu součástek jsem všechny otestoval a úspěšně integroval do robotického manipulátoru.

Pátá část práce se zabývala implementací zvoleného řídicího softwaru na rozhýbání postaveného robota. Bylo tedy nutné vymyslet a implementovat vlastní navrženou kinematiku tak, aby se robot dal řídit pomocí přichozího G-kódu. Navíc musel být robot schopen vyměňovat nástrojovou hlavu pro různé typy úloh (3D tisk, plotter). Navrhl jsem dva způsoby pro získání dopředné a inverzní kinematiky. Tuto kinematiku jsem poté převedl do C++ a úspěšně aplikoval v Marlinu.

Jako další bylo třeba napsat program pro ovládání robotického manipulátoru v grafickém prostředí, aby tak bylo robota možno snadno ovládat z jakéhokoliv běžného počítače. Pro tento úkol jsem zvolil knihovnu Pyside2 v jazyce Python, kde jsem vytvořil několik obrazovek, jak pro ovládání robota, tak i například jeho 3D model.

Co se týče výsledků 3D tisku, tak bych práci označil za úspěšnou. S robotem se mi povedlo vyrobit model loďky z termoplastu PLA při použití základní trysky pro stolní tiskárny o průměru otvoru 0,4 mm. Kvalita sice nebyla úplně dokonalá, ale zde velmi záleželo na nastavení jednotlivých rychlostí tisku a velikosti tištěného modelu. Je tedy

zapotřebí provést více testů, a to na větších modelech při různých rychlostech a při použití vyhřívané podložky

U aplikace kreslení jsem byl spokojen více, protože zde tolik nehrála roli rychlost, rovnost podložky a chybějící vyhřívání pracovní plochy. Manipulátor krásně vykresloval zadané křivky a linie.

Co se týče GUI, tak se povedlo úspěšně naprogramovat grafické prostředí, ze kterého se daří sledovat a ovládat manipulátor.

## 7.1 Návrhy na zlepšení

Jelikož se jedná o můj první prototyp robotického manipulátoru, tak návrhů na zlepšení je hned několik.

V mechanické části je prostor pro zlepšení přesnosti uložení a samotného smontování jednotlivých dílů. Tento nedostatek byl poznat na nedokonalé kolmosti nástroje vůči podlaze (testováno přes digitální úhloměr vůči podlaze). Je zde náklon asi  $0,2^\circ$ . Tato nedokonalost se nejvíce ukazuje na poklesu výšky Z osy nástroje při delším natažení ramene.

Další prostor pro zlepšení je v celkové tuhosti robota. Ta při maximálním natažení ramene není úplně dobrá a při špatně zvolených rychlostech se občas nástroj robota lehce rozkmitá.

Co se týče programu pro řízení, tak Marlin mi přijde jako dostatečně výkonný firmware pro použití na tuto aplikaci. Jeho nevýhodou je možná složitost úpravy, jelikož obsahuje obrovské množství různých funkcí, které nejsou potřeba. Co se týče pohybů, tak pro řízení stačí pouze tři stupně volnosti. Pokud bych tedy chtěl řídit něco složitějšího, je nutné se podívat po něčem jiném, například LinuxCNC.

U použití 3D tisku bych určitě doporučoval tisknout na vyhřívané podložce, jelikož se pak lze více soustředit na vylepšení samotného robota a jeho parametrů, které se dají ovlivnit a nemusí se řešit odlepování a kroucení výtisku.

Grafické ovládání by se dalo převést do formátu aplikace a stát se tak univerzálnějším, jelikož v tuto chvíli jde spustit pouze přes nainstalovanou PyCharm aplikaci.

## Seznam obrázků

Obr. 1: Technologie FDM .....	- 12 -
Obr. 2: Kartézská konstrukce.....	- 13 -
Obr. 3: Polární konstrukce .....	- 13 -
Obr. 4: Delta konstrukce .....	- 13 -
Obr. 5: Scara konstrukce.....	- 13 -
Obr. 6: Kloubové rameno .....	- 13 -
Obr. 7: Pracovní oblast: a) Kvádr, b) Cylindrická, c) Sférická .....	- 14 -
Obr. 8: Cylindrický souřadný systém .....	- 14 -
Obr. 9: Sférický souřadný systém.....	- 14 -
Obr. 10: 3D tisk betonu (polární konstrukce) [1] .....	- 15 -
Obr. 11:3D tisk kovů (kloubové rameno) [8] .....	- 15 -
Obr. 12: 3D tisk termoplastu (kartézská konstrukce) [9].....	- 15 -
Obr. 13: Paletizační manipulátor [10].....	- 16 -
Obr. 14: Geometrie rovnoběžníku .....	- 16 -
Obr. 15: Návrh délek ramen paralelogramu .....	- 17 -
Obr. 16: Konstrukční návrh .....	- 17 -
Obr. 17: Postup při tvoření požadavku pro manipulátor .....	- 18 -
Obr. 18: RepRap logo [13] .....	- 19 -
Obr. 19: Marlin logo [14] .....	- 19 -
Obr. 20: Klipper logo [15] .....	- 19 -
Obr. 21: GRBL logo [16].....	- 20 -
Obr. 22: LinuxCNC logo [17].....	- 20 -
Obr. 23. Robotický manipulátor .....	- 23 -
Obr. 24: Planetová převodovka [20].....	- 26 -
Obr. 25: Kuželové upínací pouzdro [21] .....	- 27 -
Obr. 26. Adaptér pro pouzdro [21] .....	- 27 -
Obr. 27: Díly z duralu 8 mm.....	- 28 -
Obr. 28: Díly z duralu 15 mm.....	- 28 -
Obr. 29: Díly z oceli .....	- 28 -
Obr. 30: Rotační stůl.....	- 29 -
Obr. 31: Křížové ložisko [22] .....	- 30 -
Obr. 32: Řez rotačním stolem .....	- 30 -

Obr. 33: Konstrukce manipulátoru .....	- 31 -
Obr. 34: Lícovaný M10x1x42,5 .....	- 32 -
Obr. 35: Malé kluzné pouzdro [23] [23] .....	- 32 -
Obr. 36: Malý kluzný kroužek [23] .....	- 32 -
Obr. 37: Řez kloubového spoje v místě A (Obr. 33) .....	- 32 -
Obr. 38: Velké kluzné pouzdro [23] .....	- 33 -
Obr. 39: Řez manipulátorem v místě B (Obr. 33) .....	- 33 -
Obr. 40: Výměnná hlavice pro různé typy nástrojů .....	- 34 -
Obr. 41: Nástrojová část se zamykacím mechanismem .....	- 34 -
Obr. 42: Nástroj pro měření s úchytkoměrem .....	- 34 -
Obr. 43: Nástroj pro 3D tisk .....	- 34 -
Obr. 44: Nástroj pro plotter .....	- 34 -
Obr. 45: Nástroj upevněn v hlavě manipulátoru .....	- 35 -
Obr. 46: Řez nástrojovou hlavou pro 3D tisk v manipulátoru .....	- 35 -
Obr. 47: Schéma zapojení .....	- 36 -
Obr. 48: Octopus Pro STM32F429ZGT6 [26] .....	- 37 -
Obr. 49: Schéma ramene [29] .....	- 38 -
Obr. 50: Schéma rotačního stolu [29] .....	- 38 -
Obr. 51: Servo MG946-R [27] .....	- 39 -
Obr. 52: Graf závislosti momentu na rychlosti [27] .....	- 39 -
Obr. 53: Nema 34 – 12Nm [27] .....	- 39 -
Obr. 54: Graf závislosti momentu na rychlosti [27] .....	- 40 -
Obr. 55: Nema 34 - 9Nm [27] .....	- 40 -
Obr. 56: Nema 17 [25] .....	- 40 -
Obr. 57: CL86T - Driver pro krokové motory Nema 34 [30] .....	- 41 -
Obr. 58: Schéma připojení driveru k řídicí desce [30] .....	- 41 -
Obr. 59: TMC2209 – Driver pro krokové motory Nema 17 [31] .....	- 42 -
Obr. 60: Schéma ovládání přes STEP/DIR .....	- 44 -
Obr. 61: Schéma komunikace SPI .....	- 44 -
Obr. 62: Schéma komunikace UART .....	- 44 -
Obr. 63: Spínaný zdroj pro motory [33] .....	- 45 -
Obr. 64: Spínaný zdroj pro desku a další prvky [33] .....	- 45 -
Obr. 65: Přední panel s konektory .....	- 46 -



Obr. 66: Umístění koncových spínačů pro tři motory .....	- 47 -
Obr. 67: Konstrukce koncového spínače .....	- 47 -
Obr. 68: E-Stop tlačítko .....	- 48 -
Obr. 69: Zapojování a pájení v základně robota .....	- 48 -
Obr. 70: Manipulátor - schéma 1 .....	- 50 -
Obr. 71: Manipulátor - schéma 2 .....	- 50 -
Obr. 72: Pomocný obrázek .....	- 52 -
Obr. 73: Rotace kolem osy X.....	- 53 -
Obr. 74: Rotace kolem osy Y.....	- 53 -
Obr. 75: Rotace kolem osy Z.....	- 53 -
Obr. 76: Posuv v ose X, Y, Z.....	- 53 -
Obr. 77: Klouby manipulátoru.....	- 54 -
Obr. 78: Přejchod z 0 do 1 .....	- 55 -
Obr. 79: Přejchod z 1 do 2 .....	- 55 -
Obr. 80: Přejchod z 2 do 3 .....	- 55 -
Obr. 81: Přejchod z 3 do 4 .....	- 56 -
Obr. 82: Přejchod z 4 do 5 .....	- 56 -
Obr. 83: Přejchod z 1 do 2* .....	- 56 -
Obr. 84: Přejchod z 1 do 6 .....	- 56 -
Obr. 85: Přejchod z 2* do 7 .....	- 57 -
Obr. 86: Přejchod z 6 do 8 .....	- 57 -
Obr. 87: Přejchod z 7 do 9 .....	- 57 -
Obr. 88: Přejchod z 8 do 10 .....	- 57 -
Obr. 89: Přejchod z 10 do 11 .....	- 58 -
Obr. 90: Terminál: Výsledek skriptu pro inverzní kinematiku.....	- 61 -
Obr. 91: B) Numerické řešení s cylindrickými souřadnicemi .....	- 61 -
Obr. 92: A) Numerické řešení s kartézskými souřadnicemi .....	- 61 -
Obr. 93: Visual Studio Code + Platformio IDE logo [37].....	- 62 -
Obr. 94: Terminal: Report z konzole po úspěšném buildu firmwaru .....	- 63 -
Obr. 95: Posloupnost při tvorbě GUI.....	- 65 -
Obr. 96: QT Designer editor .....	- 66 -
Obr. 97: Hlavní lišta v GUI .....	- 67 -
Obr. 98: Tlačítka v MENU pro práci .....	- 68 -

Obr. 99: Tlačítka v MENU pro simulaci .....	- 68 -
Obr. 100: Tlačítka v MENU pro servis.....	- 68 -
Obr. 101: Tlačítka v MENU pro komunikaci .....	- 68 -
Obr. 102: Obrazovka pro ovládání 3D tisku.....	- 69 -
Obr. 103: Obrazovka s 3D modelem manipulátoru .....	- 69 -
Obr. 104: Obrazovky pro řízení pohybu .....	- 70 -
Obr. 105: Obrazovka pro navázání komunikace .....	- 70 -
Obr. 106: Ukázka kódování zprávy přes utf-8 [38] .....	- 72 -
Obr. 107: Výkres v Inventoru .....	- 73 -
Obr. 108. Kartónový model manipulátoru.....	- 74 -
Obr. 109. Testování plotteru .....	- 74 -
Obr. 110. Testování 3D tisku.....	- 75 -
Obr. 111. Měření přesnosti a odchylek.....	- 76 -

## Seznam tabulek

Tab. 1: Volba převodovky (hodnoty převzaty od prodejce [20]).	- 26 -
Tab. 2: Volba spojky pro manipulátor (hodnoty převzaty od prodejce [21]).	- 27 -
Tab. 3: Volba ložiska pro rotační stůl (hodnoty převzaty od prodejce [22]).	- 30 -
Tab. 4: Volba ložisek pro spoje (hodnoty převzaty od prodejce [23]).	- 32 -
Tab. 5: Volba ložiska pro velké rameno (hodnoty převzaty od prodejce [23]).	- 33 -
Tab. 6: Volba motoru pro zámek (hodnoty převzaty od prodejce [27]).	- 39 -
Tab. 7: Volba motoru pro rotační stůl (hodnoty převzaty od prodejce [27]).	- 39 -
Tab. 8: Volba motoru pro ramena (hodnoty převzaty od prodejce [27]).	- 40 -
Tab. 9: Volba motoru pro extruder (hodnoty převzaty od prodejce [25]).	- 40 -
Tab. 10: CL86T mikrostepping [30].	- 42 -
Tab. 11: Volba zdroje (hodnoty převzaty od prodejce [33]).	- 45 -

## Seznam kódů

Kód 1: Příkaz pro pozicování [14].....	- 22 -
Kód 2: Příkaz pro přepsání pozice [14] .....	- 22 -
Kód 3: Příkaz pro nastavení pozicování [14].....	- 22 -
Kód 4: Příkaz pro vypsání pozice [14] .....	- 22 -
Kód 5: Příkaz pro nastavení teploty [14].....	- 22 -
Kód 6: Příkaz pro servo [14].....	- 22 -
Kód 7: Inicializace numerické metody .....	- 59 -
Kód 8: Funkce pro dopřednou kinematiku .....	- 59 -
Kód 9: Výpočet odchylky .....	- 59 -
Kód 10: Výpočet Jakobiánu.....	- 59 -
Kód 11: Výpočet inverzní funkce .....	- 60 -
Kód 12: Main() .....	- 60 -
Kód 13: Dopředná kinematika - analyticky pro C++ .....	- 64 -
Kód 14: Inverzní kinematika - analyticky pro C++.....	- 64 -
Kód 15: Click event tlačítka .....	- 67 -
Kód 16: Skript pro čtení.....	- 71 -
Kód 17: Skript pro zápis .....	- 71 -

# Bibliografie

- [1] Evjemo, L. D., Moe, S., Gravdahl, J. T., Roulet-Dubonnet, O., Gellein, L. T., „Additive manufacturing by robot manipulator: An overview of the state-of-the-art and proof-of-concept results.“ In: *22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Limassol, Cyprus, 2017, pp. 1-8, DOI: 10.1109/ETFA.2017.8247617.
- [2] Gibson, I., Rosen, D., Stucker, B., Khorasani, M., „Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing.“, second edition, Georgia, Springer, 2015, ISBN: 978-3030561291.
- [3] Jazar, N. R., „Theory of Applied Robotics: Kinematics, Dynamics, and Control.“, Springer, 2010, ISBN: 978-1441917492.
- [4] Lynch, K. M., Park, F. C., „Modern Robotics: Mechanics, Planning, and Control.“, Cambridge University Press, 2017, ISBN: 978-1107156302.
- [5] Rosales, E. M., Gan, Q., „Forward and Inverse Kinematics Models for a 5-dof Pioneer 2 Robot Arm.“, University of Essex, 2002.
- [6] Neustupa, J., „Matematika II. Skriptum Strojní fakulty.“, Praha: ČVUT, 2015, ISBN: 978-80-01-05684-4.
- [7] Reinhardt, D., Saunders, R., Burry, J., „Robotic Fabrication in Architecture, Art and Design.“, Sydney, Australia: Springer, 2016, ISBN: 978-3-319-26378-6.
- [8] Hussey, M., „3D-printing robot by Joris Laarman draws freeform metal lines.“ In: *dezeen.com* [Online]. Hussey, M., 21. 02. 2014 [Cit. 12. 04. 2023]. URL: <https://www.dezeen.com/2014/02/21/3d-printing-robot-by-joris-laarman-draws-freeform-metal-lines/>.
- [9] Thermwood corporation, „Thermwood - Large Scale Additive Manufacturing.“ In: *thermwood.com* [Online]. Thermwood, 2018 [Cit. 16. 05. 2023]. URL: [https://www.thermwood.com/lam\\_home.htm](https://www.thermwood.com/lam_home.htm).
- [10] JH Robotics Inc. company, „JH robotics - manipulators.“ In: *jhrobotics.com* [Online]. JH robotics, Inc. 2023 [Cit. 22. 04. 2023]. URL: <https://www.jhrobotics.com/palletizing-depalletizing-pg.html>.

- [11] Harris, W. J., Stöcker, H., „Parallelogram.“ In: Handbook of Mathematics and Computational Science §3.6.3, New York:: Springer-Verlag, 1998, ISBN: 978-0387947464.
- [12] Kong, X. M., You, W., Du, J. Z., Sun, N. L., „Optimal design for a 2-DOF high dynamic manipulator based on parallelogram mechanism.“ In: *Advanced Intelligent Mechatronics*, Montréal, Canada, 2010, DOI: 10.1109/AIM.2010.5695735
- [13] RepRap developers, „RepRap - Open-source firmware.“ In: repprap.org [Online]. RepRap, 16. 09. 2022 [Cit. 01. 05. 2023]. URL: <https://repprap.org/wiki/RepRap>.
- [14] Marlin developers, „Open-source firmware.“ In: marlinfw.org [Online]. Marlin, 2023 [Cit. 05. 05. 2023]. URL: <https://marlinfw.org/>.
- [15] Klipper developers, „Klipper - Open-source firmware.“ In: klipper3d.org [Online]. Klipper, 2023 [Cit. 01. 05. 2023]. URL: <https://www.klipper3d.org/>.
- [16] GRBL, „GRBL - Open-source firmware.“ In: github.com/grbl [Online]. GRBL, 17. 03. 2016 [Cit. 01. 05. 2023]. URL: <https://github.com/grbl/grbl>.
- [17] LinuxCNC developers, „LinuxCNC - Open source controller.“ In: linuxcnc.org [Online]. LinuxCNC, 20 09. 2022 [Cit. 28. 04. 2023]. URL: <http://linuxcnc.org/>.
- [18] Jančík, L., Zýma, J., „Části a mechanismy strojů.“, Praha: ČVUT, 2004, ISBN: 8001028917.
- [19] García, L. P., Crispel, S., Saerens, E., Verstraten, T., Lefeber, D., „Compact Gearboxes for Modern Robotics: A Review.“ In: *Frontiers in Robotics and AI*, Belgium , 2020, Volum 7, DOI: 10.3389/frobt.2020.00103.
- [20] FTDCNTOP, „CN TOP - Shop with gearboxes.“ In: ebay.com [Online]. CN TOP Limited, 2023 [Cit. 05. 02. 2023]. URL: <https://www.ebay.com/str/cntop201301>.
- [21] Pikron, „Pikron - Upínací prvky, Řetězy, Spojky.“ In: pikron.cz [Online]. Pikron, 2021 [Cit. 01. 03. 2023]. URL: <https://www.pikron.cz/>.
- [22] Schaeffler „Křížová ložiska - Kombinovaná axiální a radiální.“ In: medias.schaeffler.us [Online]. Schaeffler, 2023 [Cit. 04. 04. 2023]. URL: <https://medias.schaeffler.us/en/product/rotary/rolling-and-plainbearings/combined-axial-%2c-radial-bearings/crossed-roller-bearing/sx011836-a/p/397888>.

- [23] SKF company, „Kluzná ložiska a podložky.“ In: skf.com [Online]. SKF, 2023 [Cit. 05. 03. 2023]. URL: <https://www.skf.com/cz/productinfo/productid-PSMF%20152212%20A51>.
- [24] Weigert, F., Hebenstreit, R., Füßl, R., Theska, R., „Experimental Setup for the Investigation of Reproducibility of Novel Tool Changing Systems in Nanofabrication Machines.“ In: *Nanomanufacturing and Metrology*, 2021, pp.181-189, DOI: <https://doi.org/10.1007/s41871-021-00103-9>.
- [25] Creality3D, „Ender 3 - Direct Extruding Kit.“ In: [creality3dofficial.eu](http://creality3dofficial.eu) [Online]. China: Creality3D, 2023 [Cit. 10. 05. 2023]. URL: <https://www.creality3dofficial.eu/products/direct-extruding-extruder-nozzle-kit-for-ender-3>.
- [26] BIGTREETECH BQ, „BQ-BIQU-BTT Octopus Pro V1.0.“ In: [biqu.equipment.com](http://biqu.equipment.com) [Online]. BIQU, 2023 [Cit. 17. 04. 2023]. URL: <https://biqu.equipment/collections/control-board/products/bigtreetech-octopus-pro-v1-0-chip-f446?variant=40144816767074>.
- [27] Stepper online, „Omc-Stepperonline - Closed loop steppers.“ In: [omc-stepperonline.com](http://omc-stepperonline.com) [Online]. Stepperonline, 2023 [Cit. 20. 05. 2023]. URL: <https://www.omc-stepperonline.com/nema-34-closed-loop-stepper-motor>.
- [28] Electromate, „Smart cheat list for motor design.“ In: [electromate.com](http://electromate.com) [Online]. Canada: Electromate, 2023 [Cit. 10. 05. 2023]. URL: <https://www.electromate.com/resources/smart-motion-cheat-sheet/>.
- [29] Corp. Oriental Motor USA, „Motor sizing guide.“ In: [orientalmotor.com](http://orientalmotor.com) [Online]. USA: Oriental Motor, 2022 [Cit. 04. 04. 2023]. URL: <https://www.orientalmotor.com/motor-sizing/index.html>.
- [30] Stepper online, „Omc-Stepperonline - Stepper motor driver and controller.“ In: [omc-stepperonline.com](http://omc-stepperonline.com) [Online]. Stepperonline, 2023 [Cit. 20. 05. 2023]. URL: <https://www.omc-stepperonline.com/stepper-motor-driver>.
- [31] 3Dwork, „Introduction of TMC Drivers.“ In: [3dwork.io](http://3dwork.io) [Online]. Peter, 09. 05. 2022 [Cit. 17. 04. 2023]. URL: <https://3dwork.io/en/tmc-drivers/>.
- [32] Noviello, C., „Mastering STM32.“ In: [sciarium.com](http://sciarium.com) [Online]. YuriDp, 24. 11. 2016 [Cit. 10. 04. 2023]. URL: <https://sciarium.com/file/237582/>.

- [33] Stepper online, „Omc-Stepperonline - Power supply.“ In: omc-stepperonline.com [Online]. Stepperonline, 2023 [Cit. 20. 05. 2023]. URL: <https://www.omc-stepperonline.com/power-supply>.
- [34] Nicolescu, F. A., Ilie, M. F., Alexandru, G. T. , „Forward and inverse kinematics study of industrial robots taking into account constructive and functional parameters modeling.“ In: *Proceedings in Manufacturing Systems*, Bucharest, Romania, 2015, Volume 10, Issue 4, p.157-164, ISSN: 2067-9238.
- [35] Cristoiu, C., Nicolescu, A., „New approach for forward kinematic modeling of industrial robots.“ In: *Research and Science Today Supplement No. 2*, Gorj, Romania, 2017, ISSN: 2344-0007.
- [36] Han, X., Tian, Q., „Kinematics Analysis of Palletizing Robot.“ In: *Advanced Materials Research*, Switzerland, Trans Tech Publications, 2014, pp. 915-916, DOI: 10.4028/www.scientific.net/AMR.915-916.477.
- [37] Johnson, B., „Visual Studio Code.“, Wiley, 2019, ISBN: 978-1119588184.
- [38] Fitzpatrick, M., „Create GUI Applications with Python and Qt5.“, Netherlands, 2022, ISBN: 979-8585904158.



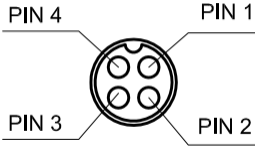

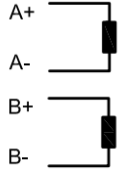
## **Přílohy**

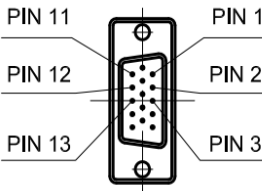

- [1]. Konektory použité pro motory
- [2]. Konektory použité pro panel
- [3]. CD disk

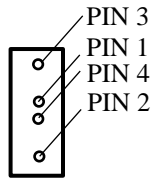

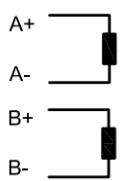
### **Příloha na CD**

- [4]. Inventor 2021 – Model manipulátoru
- [5]. Python – kód pro grafické ovládání
- [6]. C++ - Celý upravený kód Marlinu
- [7]. Matlab – skripty pro kinematiku

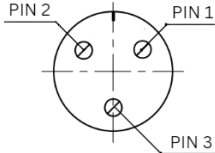

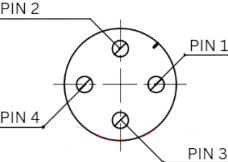

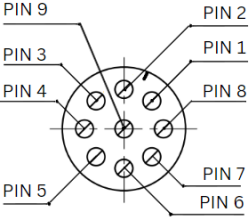

## Příloha: Konektory pro motory

Konektor pro napájení vinutí Nema34						Motor
Popis	PIN	Barva vodiče	Popis	Schéma	Obrázek	Vinutí
Vyvedený kabel z motoru od výrobce	1	černá	A+			
	2	zelená	A-			
	3	červená	B+			
	4	modrá	B-			

Konektor pro enkodér Nema34					
Popis	PIN	Barva vodiče	Popis	Schéma	Obrázek
Vyvedený kabel z motoru od výrobce	1	černá	EA+		
	2	červená	VCC		
	3	bílá	GND		
	11	žlutá	EB+		
	12	zelená	EB-		
	13	modrá	EA-		

Konektor pro napájení vinutí Nema17						Motor
Popis	PIN	Barva vodiče	Popis	Obrázek	Vinutí	
Vyvedený vodiče z motoru od výrobce	1	černá	A+			
	2	zelená	A-			
	3	červená	B+			
	4	modrá	B-			

**Příloha: Konektory použité pro panel**

Konektory pro připojení periferii k řídicí desce a driverům						
Popis	Typ	PIN	Barva vodiče	Popis	Schéma	Obrázek
Pro stopku	DS1110-06-03LYP	1	-	-		
		2	-	-		
		3	-	-		
Pro připojení napájení cívek motoru k driveru a hotbedu	DS1110-06-04LYP	1	černá	A+		
		2	zelená	A-		
		3	červená	B+		
		4	modrá	B-		
Pro připojení enkodéru motoru k driveru	DS1110-06-09LYP	1	černá	EA+		
		2	červená	VCC		
		3	bílá	GND		
		4	žlutá	EB+		
		5	zelená	EB-		
		6	modrá	EA-		
		7	-	-		
		8	-	-		
		9	-	-		