

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-100863-98362

**KRESLENIE S PRIEMYSELNÝM ROBOTOM  
A GRAFITOVOU TUHOU**

**BAKALÁRSKA PRÁCA**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Evidenčné číslo: FEI-100863-98362

**KRESLENIE S PRIEMYSELNÝM ROBOTOM  
A GRAFITOVOU TUHOU**

**BAKALÁRSKA PRÁCA**

Študijný program:	Robotika a kybernetika
Študijný odbor:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce/školiť:	Ing. Michal Adamík

**Bratislava 2021**

**Alexander Polťák**



## ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Alexander Polťák**  
ID študenta: 98362  
Študijný program: robotika a kybernetika  
Študijný odbor: kybernetika  
Vedúci práce: Ing. Michal Adamík  
Miesto vypracovania: Ústav robotiky a kybernetiky

Názov práce: **Kreslenie s priemyselným robotom a grafitovou tuhou**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Vplyvom rýchlo sa rozvíjajúceho odvetvia priemyselnej robotiky, sa robotické manipulátory začínajú používať v rôznych oblastiach ako je architektúra umenie a dizajn. Cieľom práce je pripraviť generátor trajektórii pre 6-osi priemyselný robot, ktorý umožní kreslenie s grafitovou tuhou umiestnenou na pasívnom pružnom nástroji. Vstupom do generátora bude bitmapova grafika. Programovanie bude prebiehať v prostredí Grasshopper, ktoré umožňuje programovať interakciu objektov, umiestnených v priestore CAD programu Rhinoceros 3D. Grasshopper má veľmi prívetivý grafický programovací jazyk a komunitu, ktorá pridáva funkčné súčasti, medzi ktorými sú aj zásuvné moduly Kuka PRC, Hal Robotics alebo Taco ABB. Každý z týchto zásuvných modulov umožňuje prípravu kódu pre rôzne značky robotických ramien.

Úlohy:

1. Oboznámte sa s kinematickou štruktúrou priemyselného manipulátora.
2. Naštudujte si online/offline spôsoby programovania priemyselného manipulátora.
3. Vytvorte virtuálny model pracoviska v prostredí Rhinoceros 3d.
4. Naprogramujte generátor trajektórii.
5. Otestujte a vyhodnoťte dosiahnuté výsledky.

Riešenie zadania práce od: 21. 10. 2020

Dátum odovzdania práce: 04. 06. 2021

**Alexander Polťák**  
študent

**prof. Ing. Jarmila Pavlovičová, PhD.**  
vedúci pracoviska

**doc. Ing. Eva Miklovičová, PhD.**  
garantka študijného programu

## PodĎakovanie

Touto cestou ďakujem môjmu školiteľovi Ing. Michalovi Adámikovi, za jeho odborné vedenie, pomoc a cenné rady, ktoré mi počas vypracovania tejto práce poskytol.

## ANOTÁCIA BAKALÁRSKEJ PRÁCE

Slovenská technická univerzita v Bratislave  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný odbor: kybernetika

Študijný program: Robotika a kybernetika

Autor: Alexander Polt'ák

Bakalárska práca: Kreslenie s priemyselným robotom a grafitovou tuhou

Vedúci bakalárskej práce: Ing. Michal Adamík

Mesiac, rok odovzdania: Jún, 2021

Kľúčové slová: priemyselný robot, kreslenie, Rhinoceros, Grasshopper, generátor trajektórií

Cieľom práce bolo vytvorenie generátora trajektórií, ktorý umožňuje priemyselnému robotovi kresliť s grafitovou tuhou. V úvode práce sme nadobudli teoretické vedomosti ohľadom problematiky. Naučili sme sa pracovať s konkrétnym typom priemyselného robota ABB IRB120. Samotný generátor sme programovali offline spôsobom. Použili sme grafický programovací jazyk Grasshopper. K vytvoreniu virtuálneho modelu prostredia sme použili plugin Taco ABB a CAD modelovací program Rhinoceros. Pre ďalšie informácie ohľadom tlaku na papier a sýtosti vykreslených farieb sme vykonali rôzne merania. Na meranie tlaku sme použili snímač sily, krútiaceho momentu. Na základe zistených informácií sme usúdili jeho nevhodnosť. Na zistenie sýtosti farieb sme použili kameru. V jej užívateľskom programe sme si pripravili program na meranie. Vykonali sme merania pri rôznych tvrdostiach túh a z dát usúdili ich vhodnosť.

## **BACHELOR THESIS ABSTRACT**

Slovak University of Technology in Bratislava  
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION  
TECHNOLOGY

Branch of Study: Cybernetics

Study Programme: Robotics and Cybernetics

Author: Alexander Polt'ák

Bachelor Thesis: Drawing with industrial robot and graphite pencil

Supervisor: Ing. Michal Adamík

Year, Month: 2021, June

Keywords: industrial robot, drawing, Rhinoceros, Grasshopper,  
trajectories generator

The aim of the work was to create a trajectory generator that allows an industrial robot to draw with graphite. In the introduction we gained theoretical knowledge about the issue. We learned to work with a specific type of industrial robot ABB IRB120. We programmed the generator offline. We used the graphical programming language Grasshopper. The Taco ABB plugin and the CAD modeling program Rhinoceros are used to create a virtual environment model. To evaluate pressure on the paper and the saturation of the drawn colors, we performed various measurements. We used a force and torque sensor to measure the pressure. Based on the information obtained, we judged its unsuitability. We used a camera to determine the color saturation. In her user program, we have prepared a program for measurement. We performed measurements at different hardnesses of graphite and judged their suitability from the data.

# Obsah

<b>Zoznam použitých obrázkov</b>	<b>9</b>
<b>Zoznam použitých tabuliek</b>	<b>10</b>
<b>Zoznam použitých skratiek</b>	<b>11</b>
<b>Úvod</b>	<b>12</b>
<b>1 Priemyselný robot</b>	<b>13</b>
1.1 Aplikácie priemyselného robota.....	13
1.2 Zostava priemyselného robota .....	14
<b>2 Kinematika priemyselného robota</b>	<b>15</b>
2.1 Kinematická štruktúra na základe konštrukčného usporiadania .....	16
2.1.1 Kartézska (TTT) kinematická štruktúra .....	16
2.1.2 Cylindrická (RTT) kinematická štruktúra.....	17
2.1.3 Sféricá (RRT) kinematická štruktúra .....	17
2.1.4 SCARA (RRT) kinematická štruktúra .....	18
2.1.5 Angulárna (RRR) kinematická štruktúra .....	19
<b>3 Riadenie priemyselného robota</b>	<b>20</b>
3.1 Interpolácia.....	20
3.2 Typy pohybov .....	21
3.3 Aproximácia pohybov .....	22
<b>4 Programovanie priemyselného robota</b>	<b>24</b>
4.1 On-line programovanie .....	24
4.2 Off-line programovanie.....	25
4.3 Výber typu programovania.....	26
4.3.1 Pracovné prostredie.....	27
<b>5 Meranie prítlaku</b>	<b>28</b>
5.1 Zhodnotenie.....	29
<b>6 Vytvorenie generátora trajektórií</b>	<b>30</b>

6.1	Vytvorenie čiar .....	30
6.2	Posun v osi Z .....	32
6.3	Vytvorenie bodov .....	33
6.4	Kalibrácia nástroja.....	34
6.5	Definovanie TCP bodov .....	35
6.6	Spustenie programu.....	36
<b>7</b>	<b>Meranie sýtosti farieb kamerou</b>	<b>38</b>
7.1	Priebeh merania.....	39
7.1.1	Nastavenie programu na meranie .....	39
7.1.2	Výber túh.....	40
7.2	Zhodnotenie merania.....	41
	<b>Záver</b>	<b>43</b>
	<b>Literatúra</b>	<b>44</b>
	<b>Prílohy</b>	<b>46</b>
Príloha A:	GTAMP.rar – obsahuje generátor trajektórií a model prostredia ...	46
Príloha B:	Odfotené vzorky na meranie sýtosti farieb .....	47



# Zoznam použitých obrázkov

Obrázok 1: Zostava priemyselného robota[2] .....	14
Obrázok 2: Typy kinematických párov[3].....	15
Obrázok 3:Typy kinematických reťazcov[4].....	16
Obrázok 4:Kartézska (TTT) kinematická štruktúra[5] .....	17
Obrázok 5:Cylindrická (RTT) kinematická štruktúra[5] .....	17
Obrázok 6:Sférická (RRT) kinematická štruktúra[5] .....	18
Obrázok 7:SCARA (RRT) kinematická štruktúra[5] .....	18
Obrázok 8:Angulárna (RRR) kinematická štruktúra robota ABB IRB 120 [6][7].....	19
Obrázok 9:Súradnicové systémy robota .....	21
Obrázok 10: Hlavné typy pohybov robota[6] .....	22
Obrázok 11: Aproximácie troch hlavných pohybov robota[8] .....	23
Obrázok 12: Pendant (programovacia jednotka)[10].....	25
Obrázok 13: Virtuálny model robota ABB IRB120 .....	26
Obrázok 14: Reálne pracovné prostredie s jeho identickým virtuálnym modelom.....	27
Obrázok 15: Zapojený senzor s nástrojom na efektore robota[15].....	29
Obrázok 16: Rozdelenie testovacieho programu na úseky .....	30
Obrázok 17: Úsek A. Nastavovanie parametrov, zúženie pracovného priestoru a vytvorenie začiatkovej čiary .....	31
Obrázok 18: Úsek B. Výpočet počtu externých čiar a vytvorenie všetkých čiar .....	32
Obrázok 19: Úsek C. Posun externých čiar v osi Z a prípadná kalibrácia .....	33
Obrázok 20: Úsek D. Otočenie čiar a vytvorenie všetkých potrebných bodov .....	33
Obrázok 21: Úsek E. Rozdelenie bodov na samostatné celky.....	34
Obrázok 22: Úsek F. Ukážka kalibračného prípravku, nástroja s tuhou a kalibračného bloku v Grasshopperi .....	35
Obrázok 23: Aproximácia bodu, konkrétne s hodnotou z5 .....	36
Obrázok 24: Úsek G. Vytvorenie začiatkového, koncového a TCP bodov .....	36
Obrázok 25: Úsek H. Model nástroja, robota s riadiacou jednotkou a pridanie kalibračnej procedúry .....	37
Obrázok 26: Vytvorené meracie obdĺžniky v používateľskom programe CamTool.....	40
Obrázok 27: Typy túh[13] .....	40
Obrázok 28: Korekcia údajov čistého papiera .....	41
Obrázok 29: Priebehy všetkých vzoriek .....	42

# **Zoznam použitých tabuliek**

Tab. 1: Základné parametre robota ABB IRB120[7] .....	19
Tab. 2: Merania v osi Fz (citlivosť 4,24)[12] .....	29
Tab. 3: Nastavené hodnoty rýchlostí aaproximácie pre TCP body .....	35
Tab. 4: Základné parametre kamery xiQ MQ042CG-CM[16] .....	38
Tab. 5: Percentuálne zloženie vybraných túh[14] .....	40

# **Zoznam použitých skratiek**

CP – Continuous Path

FFT – Flat Field Correction

F/T – Force/Torque

I/O – Input/Output

IP – Internet Protocol

PC – Personal Computer

PTP – Point To Point

px – pixels

TCP – Tool Center Point

Tzv – takzvaný

# Úvod

Priemyselná robotika zažíva v súčasnosti obrovský úspech. Roboty vykonávajú ľudskú činnosť, ale omnoho precíznejšie a rýchlejšie. To umožňuje omnoho rozsiahlejšie aplikácie. Robotické manipulátory sú dnes takmer v každej väčšej spoločnosti. Vykonávajú prevažne náročné operácie ako manipulovanie s ťažkým materiálom, zváranie a podobne. Uľahčujú ľuďom prácu, čo im zároveň dovoľuje sa viac venovať rozvoju, ktorý posúva priemysel a iné odvetvia o krok vpred.

Robotické manipulátory sa taktiež používajú v samotnom rozvoji priemyslu. Vďaka ich presnosti sa začínajú používať napríklad aj v medicíne. V súčasnosti sa použitiu robotických manipulátorov medze nekladú a nájdeme ich využitie takmer v každej oblasti. Začínajú sa uplatňovať aj v odvetví ako architektúra, umenie a dizajn. Do tejto roviny môže byť zaradená aj táto práca. Jej hlavným cieľom je pripraviť generátor trajektórii pre šesť osí priemyselný robot. Ten zo vstupných parametrov vytvorí trajektóriu, ktorú bude robotický manipulátor s grafitovou tuhou umiestnenou na pasívnom pružnom nástroji opisovať. Preto je potrebné zohľadniť veľa aspektov kreslenia ľudskou rukou, od pritláčania tuhy, skrz kalibráciu tuhy, až po samotné kreslenie.

Výstupom bude robotom nakreslený obrazec na papieri. Snaha je, aby nakreslený obrazec vyzeral, ako nami pripravená bitmapa, ktorá bola pripravená na základe vstupných parametrov generátora. Tento obrazec bol nakreslený viac krát s tuhami s rôznou tvrdosťou. Cieľom bolo vybrať tu najvhodnejšiu tuhu pre aplikáciu robotického kreslenia.

# 1 Priemyselný robot

V súčasnosti môže za priemyselný pokrok predovšetkým globálne implementovanie automatizácie. Najpokrokovejšími prvkami automatizácie sú predovšetkým priemyselné roboty, ktoré vykonávajú väčšinu technologických procesov. Priemyselný robot odbremeňuje človeka od vykonávania nebezpečných a monotónnych prác. Umožňuje tak ľuďom využiť ich potenciál v prospech ďalšieho rozvoja.[1]

Definíciou priemyselného robota môžeme rozumieť univerzálne automatické zariadenie pripomínajúce ľudskú ruku a vykonávajúce určitú technickú činnosť.[1]

## 1.1 Aplikácie priemyselného robota

Hlavné uplatnenie priemyselných robotov je permanentne v najväčšom pomere vo výrobných halách. Súvisí to najmä s ich rýchlosťou, opakovateľnosťou a v niektorých odvetviach nedostatkom kvalifikovaných pracovníkov. Vykonávajú rôzne činnosti ako:

- nakladanie, vykladanie,
- obrábanie, rezanie, frézovanie,
- bodové zváranie, oblúkové zváranie,
- montáž, paletizovanie a balenie,
- leštenie, brúsenie,
- striekanie,
- kontrola kvality.

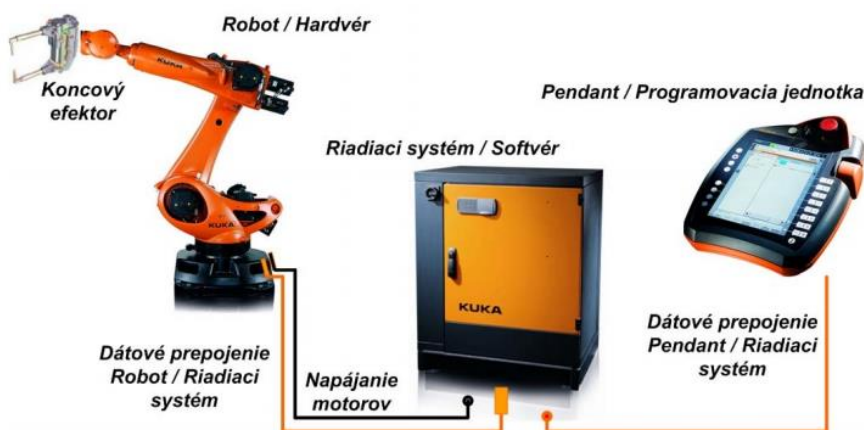
Oblasť využitia je ale podstatne viac. Vďaka svojej univerzálnosti nájdeme značne menej oblastí, kde ešte robotické manipulátory nie sú implementované.

Ich zavedenie môžeme vidieť aj pri zložitých úkonoch zásluhou sofistikovaného riadenia. Veľmi užitočné sú potom v zdravotníctve pri operáciách, kde je robot riadený samotným chirurgom. Ten mu dáva inštrukcie pomocou pedálov a ručných ovládačov. Ide tak o určitú spoluprácu medzi človekom a robotom. Takúto súčinnosť môžeme vidieť aj v iných oblastiach.

Realizácie sú zavedené aj pri vedecko-edukačných oblastiach, kde sa vykonávajú rôzne experimenty. Takto sa postupne vynachádzajú nové využitia priemyselných robotov a zdokonaľujú sa už poznané.

Robotické manipulátory majú taktiež čoraz väčšie uplatnenie pri vykonávaní určitej umeleckej činnosti ako vyrezávanie, maľovanie a napríklad aj kreslenie.

## 1.2 Zostava priemyselného robota



Obrázok 1: Zostava priemyselného robota[2]

Zostavu priemyselného robota môžeme rozdeliť na tri časti:

1. mechanická časť robota – skladá sa z:
  - základňa – časť, na ktorej je umiestnený celý robot a ktorá je pevne pripevnená k určitej horizontálnej alebo vertikálnej ploche,
  - kĺby (kĺb lakťa, kĺb ramena, kĺb zápästia, kĺb tela) – každý kĺb umožňuje pohyb v jednej osi otáčania, vytvára jeden stupeň voľnosti,
  - väzby – tvoria pevnú časť, ktorá spája kĺby,
  - ramená – sú prepojené kĺbmi a poskytujú v súlade s rozmermi a dovolenými pohybmi pracovný priestor okolo robota,
  - koncový efektor – koncová časť robota slúžiaca na vykonávanie činností, napríklad uchopenie telesa alebo inú technologickú činnosť ako zváranie a podobne. Často sa navrhujú a vyvíjajú len na konkrétnu vykonávanú činnosť. Pri tejto práci sa taktiež používa nami navrhnutý efektor, ktorý je modifikovaný na konkrétnu činnosť, konkrétne kreslenie,
2. riadiaci systém robota – slúži na riadenie pohybov robota v osiach otáčania a umožňuje komunikáciu s obsluhou na podstate dát z riadiaceho systému, snímačov,
3. pendant (programovacia jednotka) – slúži na obsluhu robota. Je možná automatická aj ručná činnosť robota s naprogramovaním jednotlivých úkonov.[2]

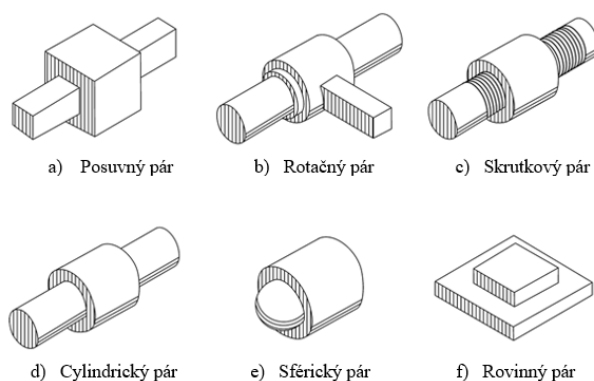
## 2 Kinematika priemyselného robota

Kinematika robota je ovplyvňovaná samotným konštrukčným usporiadaním a počtom kinematických párov, ktoré uskutočňujú jednotlivé pohyby priemyselného robota.[2]

Kinematické páry sú pohyblivé časti robota obojstranne spojené určitou väzbou. Jednotlivé časti páru sa vzájomne obmedzujú v pohybe, či už kvôli rozmerom alebo spôsobu spojenia.[2]

Rôzne typy kinematických párov:

- a) posuvný pár – osi obidvoch častí páru sú rovnobežné. Výslednou kinematikou je len posun pozdĺž osi. (jeden stupeň voľnosti),
- b) rotačný pár – osi obidvoch častí páru sú rovnobežné. Výslednou kinematikou je len rotácia okolo tejto osi. (jeden stupeň voľnosti),
- c) skrutkový pár – osi obidvoch častí páru sú rovnobežné. Výslednou kinematikou je kombinácia posunu a rotácie vzhľadom na os. (jeden stupeň voľnosti),
- d) cylindrický pár – osi obidvoch častí páru sú rovnobežné. Výslednou kinematikou je nezávislý posun a otáčanie častí vzhľadom na os. (dva stupne voľnosti),
- e) sférický pár – sférické stredy častí páru sú zarovnané. Výslednou kinematikou je rotácia okolo troch osí. (tri stupne voľnosti),
- f) rovinný pár – povrchy častí páru sú v kontakte. Výslednou kinematikou je posun v dvoch smeroch a rotácia okolo osi, ktorá je kolmá na tieto povrchy. (tri stupne voľnosti).[3]



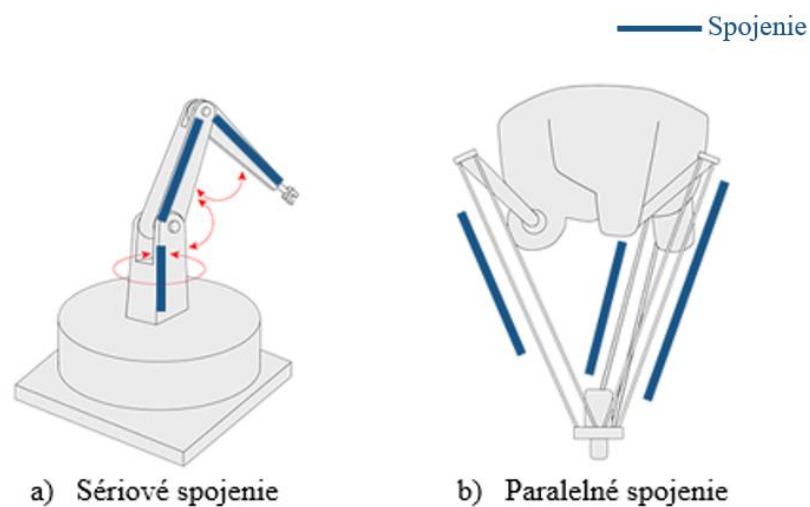
Obrázok 2: Typy kinematických párov[3]

Kinematický pár môže mať viacero stupňov voľnosti, ale pri konštrukcii priemyselného robota sú zvyčajne používané posuvné a rotačné páry s jedným stupňom voľnosti.

## 2.1 Kinematická štruktúra na základe konštrukčného usporiadania

Spájaním kinematických dvojíc vieme vytvárať kinematické reťazce s rôznym typom spojení:

- a) sériové spojenia,
- b) paralelné spojenia. [2]



Obrázok 3: Typy kinematických reťazcov [4]

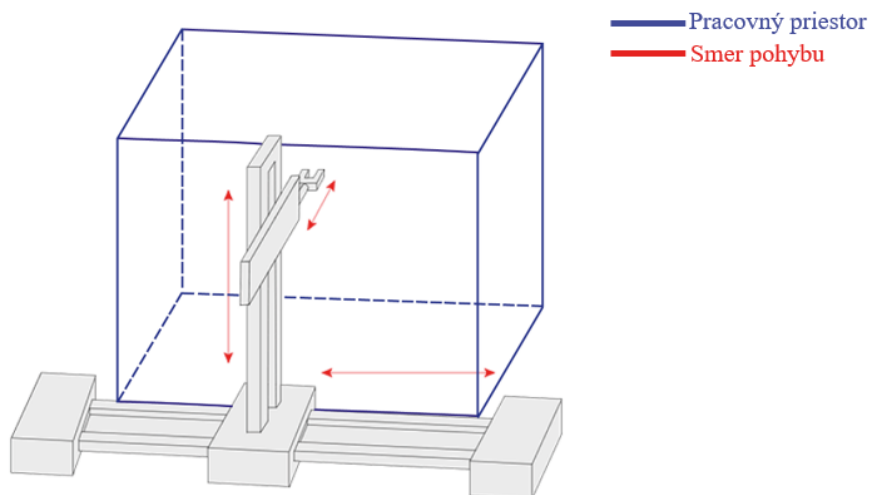
Konštrukciu priemyselných robotov potom určujeme podľa usporiadania kinematických párov v kinematickom reťazci. Najviac rozšírené sú reťazce s posuvnými-T a rotačnými-R párami. [2]

### 2.1.1 Kartézska (TTT) kinematická štruktúra

Štruktúra sa skladá z troch translačných (posuvných) párov, ktoré sú na seba navzájom kolmé. Pracuje v pravouhlom súradnicovom systéme a pracovný priestor má tvar kvádra alebo kocky vzhľadom na konštrukciu. [2]

Častou praktickou aplikáciou sú rôzne technologické procesy (nanášanie lepidla, rôzne druhy inšpekcií a podobne) a manipulovanie s bremenami (portálové žeriavy).



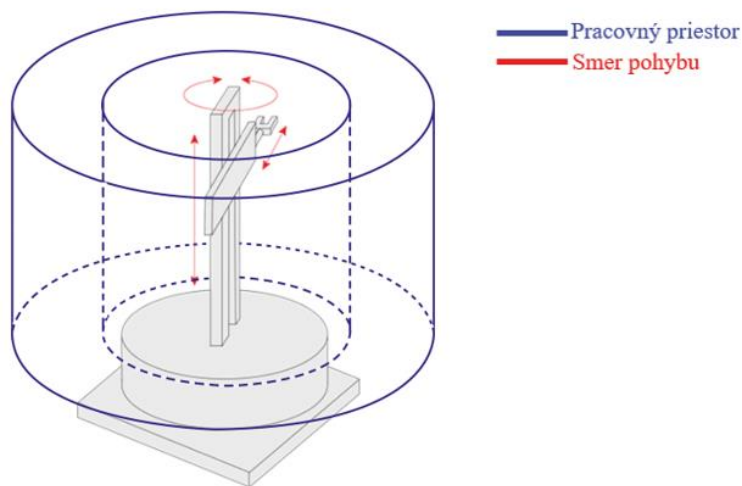


Obrázok 4:Kartézska (TTT) kinematická štruktúra[5]

### 2.1.2 Cylindrická (RTT) kinematická štruktúra

Štruktúra pozostáva z dvoch posuvných párov, ktoré sú na seba navzájom kolmé a z jedného rotačného kinematického páru.[2]

Najčastejšie aplikácie sú manipulačné práce, bodové zváranie a podobne. V súčasnosti nie sú často využívané.

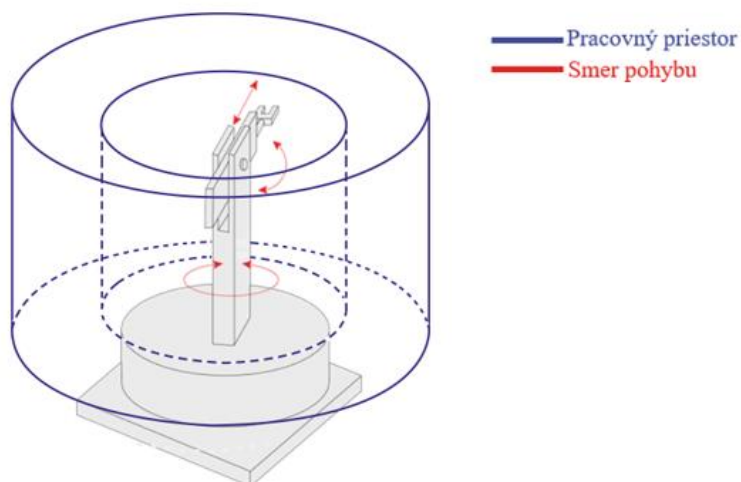


Obrázok 5:Cylindrická (RTT) kinematická štruktúra[5]

### 2.1.3 Sférická (RRT) kinematická štruktúra

Štruktúra je zložená z dvoch rotačných a jedného posuvného páru. Je jednou z prvých uskutočnených konfigurácií. Pracovná plocha je podľa konštrukcie valcového alebo guľovitého tvaru.[2]

Často sa využíva na operácie ako zváranie, brúsenie, striekanie a iné. V súčasnosti sa používa omnoho menej ako v minulosti kvôli nahradeniu angulárnymi robotmi.

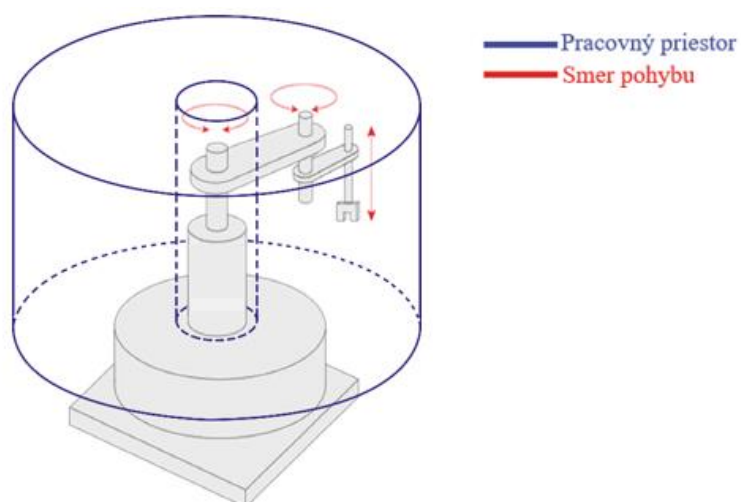


Obrázok 6:Sférická (RRT) kinematická štruktúra[5]

#### 2.1.4 SCARA (RRT) kinematická štruktúra

Štruktúra sa skladá z dvoch rotačných a jedného posuvného páru. Prednosťou je vyššia pohyblivosť a naopak nevýhodou je menší pracovný priestor a zložitejšie riadenie.[2]

Priemyselné roboty s týmto typom štruktúry sú najčastejšie využívané roboty pri manipulačných operáciách. To vďaka svojej rýchlosti a presnosti polohovania.

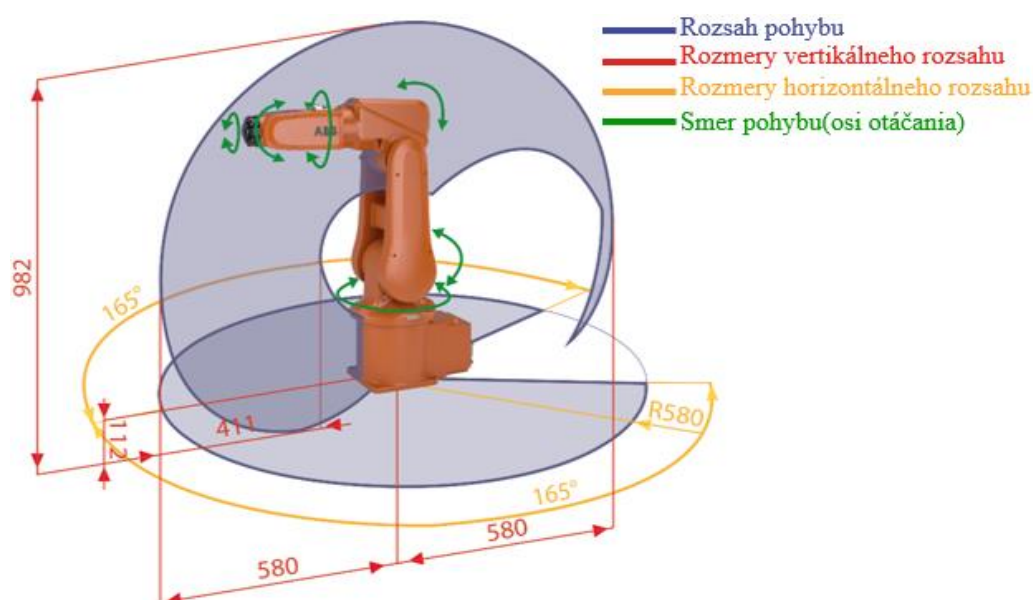


Obrázok 7:SCARA (RRT) kinematická štruktúra[5]

### 2.1.5 Angulárna (RRR) kinematická štruktúra

Štruktúra pozostáva z troch rotačných dvojíc. V súčasnosti je to najviac využívaný typ kinematickej štruktúry u robotov. To vďaka dobrým manipulačným schopnostiam alebo aj schopnosti vyhýbania sa prekážkam. Pracovný priestor má podobu multiuhlového telesa.[2]

Vyznačuje sa vysokou mierou voľnosti, ale riadenie robota je pomerne komplikované. Najpoužívanějšími robotmi s touto štruktúrou sú šesť osé roboty. Nájde ich vo väčšine fabrík, kde vykonávajú rôzne technologické procesy. Robota s touto kinematickou štruktúrou, konkrétne robota ABB IRB 120 používame v tejto práci. Jeho osi otáčania, rozsah pohybu vo vertikálnom aj horizontálnom smere, a rozmery jednotlivých rozsahov sú zobrazené na obrázku 8.



Obrázok 8: Angulárna (RRR) kinematická štruktúra robota ABB IRB 120 [6][7]

Tab. 1: Základné parametre robota ABB IRB120[7]

<b>Zaťaženie[kg]</b>	3
<b>Ochrana</b>	IP 30
<b>Opakovateľnosť polohy[mm]</b>	0.01
<b>Dosah[m]</b>	0.58
<b>Váha[kg]</b>	25

## 3 Riadenie priemyselného robota

Poznáme dva typy riadenia priemyselných robotov:

- PTP alebo bodové riadenie – má význam tam, kde je potrebné polohovanie robota do určitých pracovných bodov v priestore bez toho, aby jednotlivé body mali medzi sebou nejaký funkčný súvis. K danému riadeniu sa zaraďujú aj typy pracovných úkonov ako bodové zváranie, diskretná montáž, obsluha strojov a iné,
- CP alebo dráhové riadenie – využíva sa tam, kde má pohyb robota funkčný súvis s technologickým úkonom, ktorý vykonáva. Medzi takéto úkony môžeme zaradiť spojité zváranie, lakovanie a rôzne povrchové modifikácie, výrobné práce, zložitejšie montáže a iné.[8]

Bodové riadenie ma príbuznosť so stavaním súradníc pri číslicovo riadených výrobných strojoch a dráhové riadenie so súvislým číslicovým riadením. Stavanie súradníc je obyčajne použité pri číslicovo riadených strojoch, kde sa nastavuje nástroj v dvoch súradniciach. Takéto riadenie môžeme prirovnať k bodovému riadeniu, ale len pri malom počte bodov, desať až dvadsať. Pri väčšom počte rozlične orientovaných bodov, ktoré nie sú zorganizované v nejakej jednoduchšej geometrickej sústave, riadenie pripomína skôr stavenie súradníc s ohľadom na možné prekážky. U takéhoto riadenia je pre rôznorodosť jednotlivých bodov nereálne vypočítavať a nastavovať ich samostatné súradnice. Samotný priebeh sa skôr podobá na súvislé číslicové riadenie. Pri tomto riadení sú výsledkom krivky spojnic samostatných bodov.[8]

### 3.1 Interpolácia

Trajektória tvorená riadiacim systémom musí odpovedať dráhe pomedzi body, ktorú má robot opisovať. Takáto trajektória je priestorová a s časovým priebehom. Daná je rýchlosť a čas pohybu TCP v každom bode. Trajektóriu zariaďuje interpolačný blok riadenia v súlade so zadávanými kritériami a v zhode s vlastnosťami konštrukcie priemyselného robota. Pri PTP riadení interpolácia taktiež typu PTP zabezpečuje, aby sa rýchlostné priebehy pohonov samostatných ôs zosúladiť s najpomalším. To kvôli ustáleniu pohybu všetkých ôs v totožnej chvíli. Táto interpolácia pohybu je časovo kratšia ako CP dráhová interpolácia, pri ktorej je občas nutné aj ošetrenie singulárnych stavov. Samotné typy interpolácií sú dané požiadavkami na úlohy. Pri CP interpolácii môže byť úlohou napríklad oblúkové zváranie, kde je nutné pre dobré vytvorenie zvaru konštantná

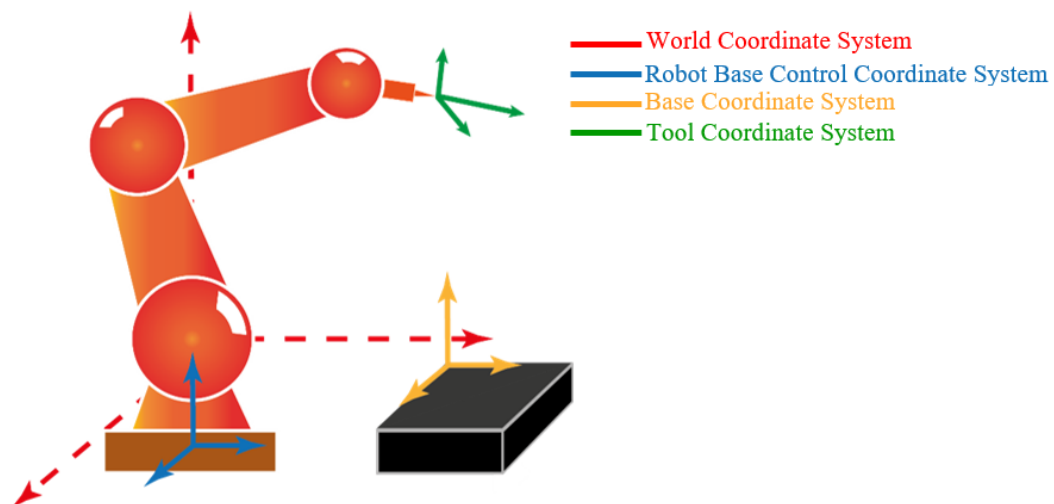
rýchlosť. Naopak napríklad pri manipulačných úlohách s PTP interpoláciou je dôležité minimalizovať takt. To je možné zmenami rýchlostí s určitou možnou mierou aproximácie.[8]

### 3.2 Typy pohybov

Pohyby sa vykonávajú v kartézskych súradnicových systémoch. V týchto systémoch vieme polohovať TCP v samostatných osiach (x, y, z) a taktiež rotovať okolo každej osi.[8]

Pri práci s robotom poznáme niekoľko súradnicových systémov, pomocou ktorých vieme robota polohovať:

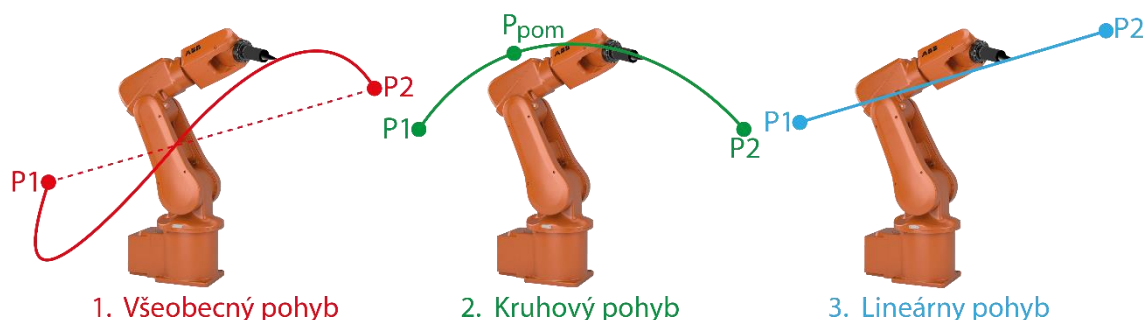
- World coordinate system (svetový súradnicový systém) – hlavný súradnicový systém. Spája sa s ním Robot Base Control Coordinate System u ABB robotov. Ten určuje pozíciu robota vzhľadom k svetovému súradnicovému systému a je umiestnený v základni robota,
- Base coordinate system (bázový súradnicový systém) – súradnicový systém určujúci pozíciu objektu so zreteľom na svetový súradnicový systém. Je zadávaný užívateľom,
- Tool coordinate system (súradnicový systém nástroja) – systém umiestnený v TCP. Je relatívny ku konkrétnej báze.[8]



Obrázok 9: Súradnicové systémy robota

Poznáme tri hlavné typy pohybov:

1. všeobecný pohyb – pri tomto pohybe sa TCP premiestňuje z bodu P1 do bodu P2 najrýchlejšie realizovateľným spôsobom. To ale neznamená, že to musí byť aj najkratšia možná dráha, ako je napríklad priamka. Angulárny robot sa pohybuje skrz rotačné pohyby v jednotlivých osiach a robot tak môže vykonávať rôzne zakrivené dráhy. V priestore je vykonanie týchto dráh robotom rýchlejšie ako vykonanie priamky. Tento typ pohybu je používaný tam, kde potrebujeme rýchlo polohovať v priestore a tam, kde nehrozia žiadne kolízie. Pri uložení súradníc samotných bodov s týmto pohybom vopred nepoznáme výslednú trasu,
2. kruhový pohyb – TCP robota sa premiestňuje z bodu P1 do P2 po kruhovej trajektórii. Pri tomto pohybe definujeme konštantnú rýchlosť. Kruhová trajektória je reprezentovaná tromi bodmi. Tými sú začiatočný, koncový a pomocný bod, ktorý zároveň vymedzuje výslednú dráhu medzi začiatočným a koncovým bodom,
3. lineárny pohyb – pohyb TCP robota vykonávaný priamo medzi bodmi P1 a P2. Tento pohyb je vykonávaný konštantou rýchlosťou.[8]



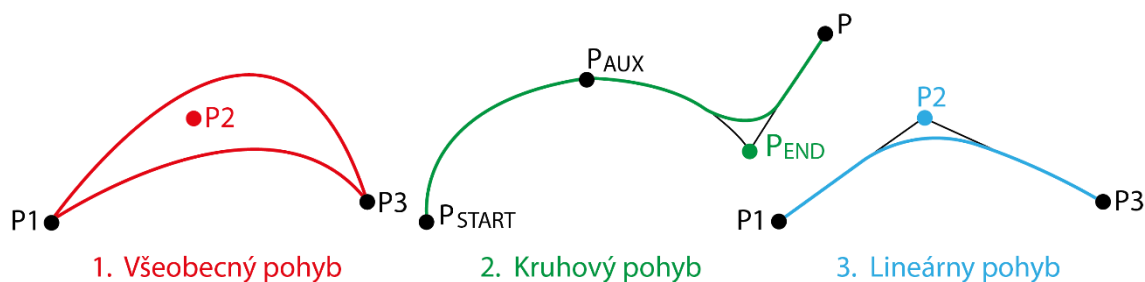
Obrázok 10: Hlavné typy pohybov robota[6]

### 3.3 Aproximácia pohybov

V praxi je využitie len základných pohybov nedostatočné. Pre všetky spomínané typy pohybov platí, že pri koncových bodoch sa robot začne pohybovať rýchlosťou blížiacou sa k nule. V týchto bodoch sa teda skoro zastaví a až potom začne vykonávať ďalší pohyb z inštrukcií. Preto, aby k tomu nedochádzalo a vykonávaná trasa bola plynulejšia a rýchlejšia sa využívajú aproximácie. Vďaka ich použitiu je na pohybové ústrojenstvo robota vyvíjaný menší nápor. Ďalšou z výhod aproximácií je aj zníženie pracovného cyklu.[8]

Ku každému z troch hlavných pohybov poznáme aproximáciu:

- aproximácia všeobecného pohybu – TCP by bez aproximácie viedol do koncového bodu P3 skrz bod P2. S jej použitím sa odlúči od trasy a po plynulejšej trajektórii sa dostane do koncového bodu. Aproximovaná dráha tu nie je vopred známa,
- aproximácia kruhového pohybu – TCP by bez aproximácie viedol do koncového bodu PEND a potom do bodu P. S jej použitím sa vychýli od trasy a po rýchlejšej trajektórii sa dostane do bodu P. Je tu možné určenie aproximačnej vzdialenosti. Aproximácia sa neaplikuje na pomocný bod,
- aproximácia lineárneho pohybu – TCP by bez tejto aproximácie viedol najprv od bodu P1 skrz bod P2 do bodu P3. S ňou sa nedostane do bodu P2, ale po rýchlejšej, plynulejšej trajektórii sa dostane do bodu P3. Definícia aproximačnej vzdialenosti je tu taktiež možná.[8]



Obrázok 11: Aproximácie troch hlavných pohybov robota[8]

## 4 Programovanie priemyselného robota

Priemyselný robot pracuje na základe inštrukcií, ktoré sú nahraté do riadiacej jednotky robota. V inštrukciách sú zapísané všetky pohyby so súradnicami, rýchlosťami a aproximáciami, ktoré má robot po zavedení do výroby vykonávať.

### 4.1 On-line programovanie

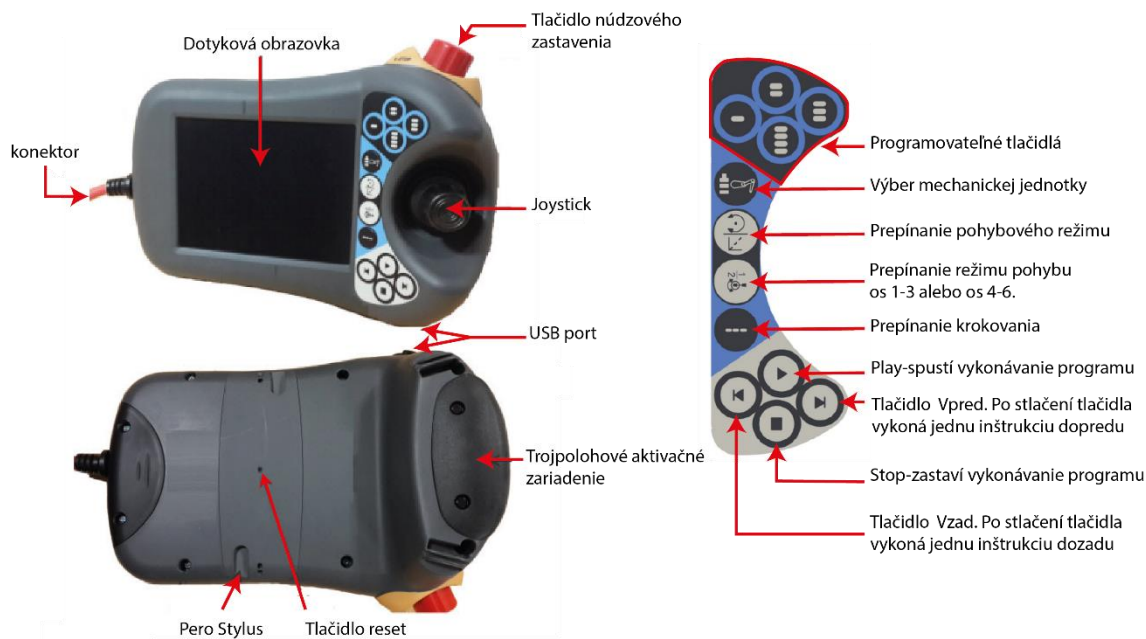
Programovanie je vykonávané priamo na pracovisku. Robot je pomocou programovacieho panelu vedený po bodoch. Tieto body si riadiaci systém ukladá do pamäte. Ďalej je potrebné naprogramovať logickú časť riadenia chápadla a aj periférne zariadenia. To znamená dlhý čas programovania a tým aj fyzickú náročnosť pre programátora. Tento typ programovania značne uľahčujú programovacie jednotky postavené na základe PC. V nich sa zobrazujú pomocou displeja rôzne vizualizácie a taktiež umožňujú priame programovanie cez I/O.[2]

Keďže sa programuje priamo na pracovisku, tak výroba v určitom úseku alebo dokonca celá výroba stojí. Naopak veľkou výhodou je práca v reálnom čase a prostredí, kde je zároveň možné vykonávať rôzne testy funkčnosti.[2]

Poznáme dve metódy on-line programovania:

1. play-back metóda – operátor vedie robota manuálne v celom technologickom pohybe. Každých pár milisekúnd je údaj o polohe a orientácii TCP uložený do riadiacej jednotky robota. Po zavedení robota do automatickej činnosti robot opakuje uložené pohyby. Pri týchto opakovaníach vznikajú nepresnosti kvôli zaťaženiu ramena v opačnom smere, ako pri vedení. Preto sa táto metóda využíva pri činnostiach nenáročných na presnosť,
2. teach-in metóda – programovanie prebieha prostredníctvom tlačidiel na pendante (programovacej jednotke). Programátor navádza rameno robota na samotné pozície, kde robot vykonáva určitú činnosť. Tieto polohy a orientácia nástroja sú následne uložené do pamäte. V automatickom režime potom robot pracuje pomocou údajov z pamäte. Podľa ručného nastavenia je automaticky zapísaná orientácia nástroja spolu so samostatnými významnými bodmi. Zvyšné funkcie robota sa programujú skrz PC alebo editor v programovacej jednotke.[2]





Obrázok 12: Pendant (programovacia jednotka)[10]

## 4.2 Off-line programovanie

Pri tomto type programovania sa kopíruje reálne prostredie robota do virtuálnej podoby. Programovanie je realizované pomocou 3D modelov s definovanou kinematikou. To umožňuje simulovať rôzne situácie, ktoré by mohli nastať v reálnom prevedení. Môže sa tak predísť rôznym kolíziám a zlepšiť nedostatky. Keďže sa simuluje vo virtuálnom prostredí, tak sa zvyšuje aj miera bezpečnosti. Kým v reálnom svete by mohlo dôjsť z dôvodu poruchy k zraneniu personálu, vo virtuálnom prostredí nie sú žiadne bezpečnostné hrozby. Program je nahratý až po vytvorení a otestovaní.[9]

Pri simulácii je najlepšie použiť zhodný 3D model robota, s ktorým pracujeme. Tieto modely majú navrhnutú geometriu a kinematiku čo najpodobnejšiu reálnemu typu robota. Tento model je možné samozrejme modifikovať alebo je možné vytvoriť aj vlastný model s vlastnou navrhnutou geometriou a kinematikou.[9]

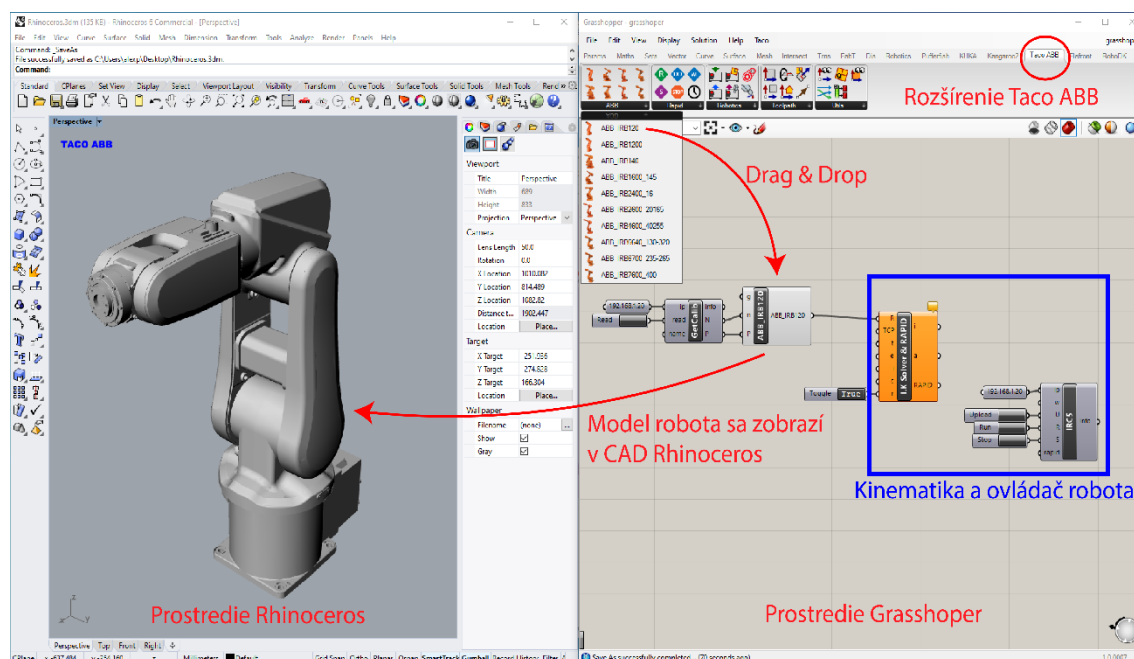
Nevýhodou u tohto typu programovania je korelácia medzi simuláciou a reálnym systémom. Dôležité sú súradnice predmetov vzhľadom k robotovi, ktoré musia byť ekvivalentné vo virtuálnom prostredí. Taktiež je dôležité voliť také body pohybu, pri ktorých kĺby robota nedosahujú krajné polohy. V prípade nepresného polohovania robota je to potrebné riešiť. Riešením môže byť aktualizácia pozícií, kontrola s akou presnosťou robot pracuje alebo aj kalibrácia robota či nástroja.[9]

### 4.3 Výber typu programovania

Robota, konkrétne ABB IRB120, ktorého sme používali pri tejto práci sme programovali offline spôsobom. Programovali sme konkrétne pomocou programovacieho jazyka Grasshopper.

Samotný Grasshopper je grafický generatívny programovací jazyk, ktorý je súčasťou CAD modelovacieho programu Rhinoceros. Využíva jeho štandardnú geometriu a má samostatné prostredie, v ktorom sa programuje. Keďže je to grafický programovací jazyk, tak sa v ňom jednotlivé operácie, algoritmy nezapisujú textom. Tie sú vkladané ako grafické objekty, ktoré sú navzájom prepojené. V samotnom Rhinocerose je možné program aj písať, ale vďaka Grasshopperu to nie je vôbec nutné. Za týmto účelom Grasshopper vlastne vznikol. Zjednodušuje, zefektívňuje modelovanie a programovanie. Skrz Grasshopper je možné vytvárať, modifikovať rôznu geometriu v prostredí Rhinoceros.

Pri tvorbe generátora trajektórií sme využili plugin TACO ABB, v ktorom je predpripravený virtuálny model robota, s ktorým sme pracovali. Programovali sme teda skrz Grasshopper a vizualizovali v prostredí Rhinoceros. Samotný model ma predpripravenú kinematickú štruktúru a kinematiku podobnú s reálnym robotom ABB IRB120. Skrz tento plugin sme boli taktiež schopní vytvorený program nahráť do reálnej riadiacej jednotky robota.



Obrázok 13: Virtuálny model robota ABB IRB120

### 4.3.1 Pracovné prostredie

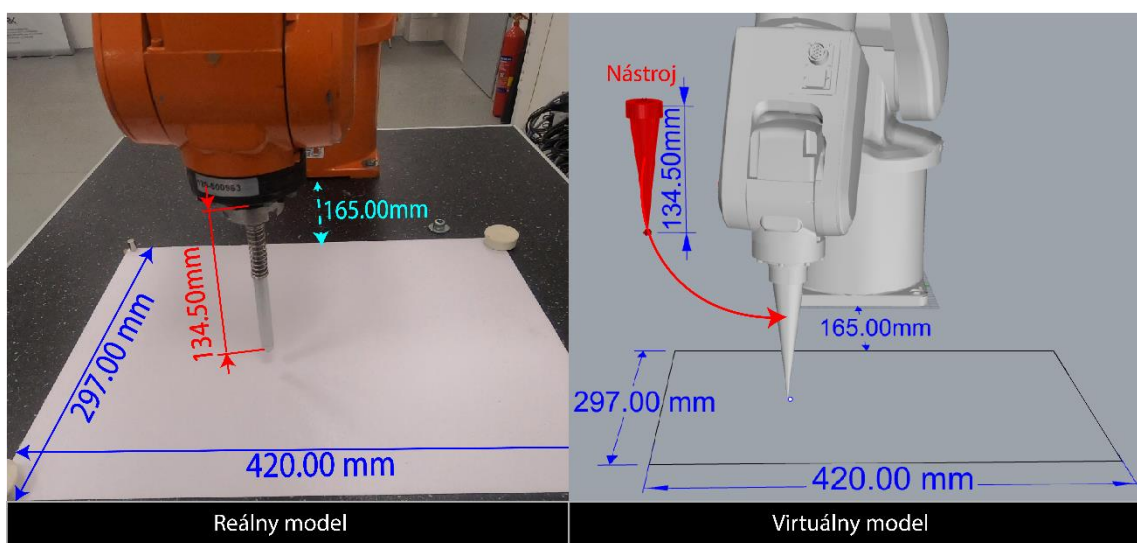
Ako pracovné prostredie sme si zvolili rovný pracovný stôl, kde sa zmestil robot ABB IRB120 a papier na kreslenie. Papier sme vybrali konkrétne s rozmermi A3. To preto, aby robot mohol kresliť na čo najväčšiu plochu bez vychýlenia sa z vlastného rozsahu pohybu.

Keďže sme programovali offline spôsobom, tak sme museli toto reálne prostredie premietnuť na identické virtuálne prostredie. To je dôležité hlavne pri nahrávaní kódu do riadiacej jednotky robota. Ak by sa model prostredia nezhodoval, tak by mohlo dôjsť k tomu, že robot bude v reáli prechádzať bodmi, do ktorých sme ho navigovať vôbec nechceli. Môže dôjsť k poškodeniu zariadení, napríklad pri náraze s nedefinovanými objektami a podobne.

V našom prípade sme do virtuálneho prostredia premietali len robota s nástrojom a pracovný priestor s rozmermi zvoleného papiera. Ten má rozmery 420mm po dĺžke a 297mm po šírke.

Pri nástroji sme museli zohľadniť najmä dĺžku. To preto, aby sme vedeli určiť v akej výške v osi Z sa má manipulátor s nástrojom v jednotlivých bodoch nachádzať. Naš nástroj, ktorý sme vytvorili má dĺžku 134.5 mm.

Samotnou polohou v osi Z sme definovali, ako robot tlačí s tuhou na papier a tým aj aká bola výsledná sýtosť farby nakreslenej línie. Na obrázku 14 je možné vidieť reálneho robota s nástrojom, pracovný priestor a taktiež ich zhodný virtuálny model v CAD modelovacom programe Rhinoceros.



Obrázok 14: Reálne pracovné prostredie s jeho identickým virtuálnym modelom

## 5 Meranie prítlaku

Pre zistenie akým tlakom pôsobil robot tuhou na papier sme použili snímač sily, krútiaceho momentu, tzv. F/T senzor.

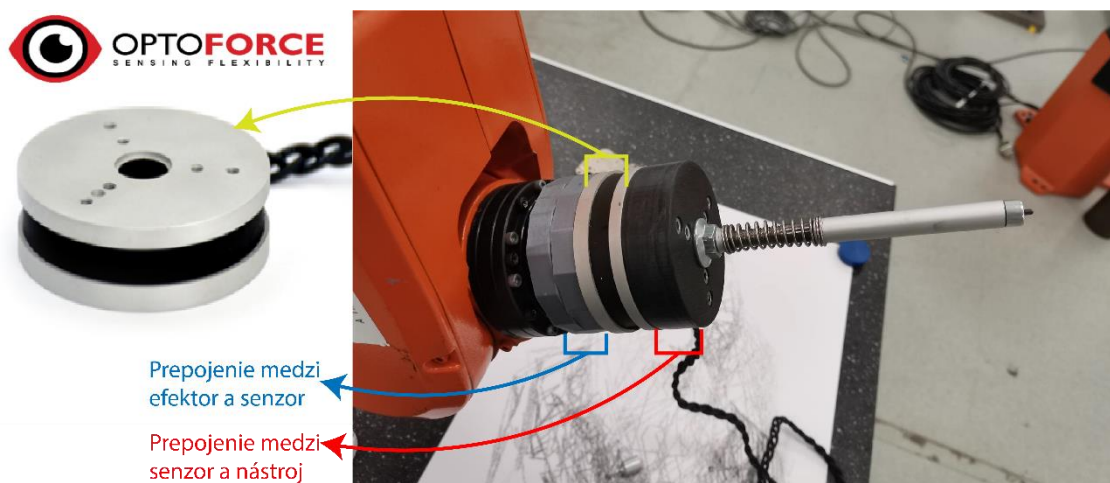
Takýto snímač je elektronický mechanizmus, ktorým je možné zaznamenávať, detegovať, regulovať lineárne aj rotačné sily naň pôsobiace. Je navrhnutý na interakciu s fyzickými objektmi v prostredí. Taktiež je konštruovaný tak, aby odolal rôznym vnútorným a vonkajším podmienkam. Najpoužívanejším typom je šesť osí snímač. Je schopný merať v šiestich rôznych smeroch a využíva technológiu tenzometra. To znamená, že ak na senzor pôsobí sila, tak vo vnútri meradla rastie alebo klesá odpor proporciálne k tejto prijímanej sile. Takéto snímače sa nachádzajú napríklad aj v robotických ramenách, konkrétne v kĺboch. Najväčšie využitie majú pri kolaboratívnych robotoch, ktoré pracujú priamo pri ľuďoch. Tieto roboty musia merať silu, ktorá je na nich vyvíjaná, aby pri kontakte s človekom dokázal robot včas spomaliť alebo zastaviť.[11]

V našom prípade sme použili šesť osí senzor Optoforce HEX-70-CE-2000N. Tento senzor je odolný voči prachu a vode (IP65). Je odolný aj proti nárazom a má vysoké rozlíšenie.

Jednotlivé osi merania majú rôzny rozsah meraných síl. Najväčšiu silu sme týmto senzorom schopný zmerať v osi  $F_z$ , ktorú výrobca udáva hodnotou 2000N. Aby sa zabránilo poškodeniu pri tlaku, tak výrobca vytvoril pre snímače prevenčnú vrstvu, silikón. Flexibilita silikónu umožňuje senzorom bez porušenia pretrvať v jednej osi najviac v dvesto percentnom preťažení.[12]

Pre tento senzor sme taktiež navrhli a vytlačili na 3D tlačiarňi prepojenie na koncový efektor robota. Zo strany, na ktorú máme pôsobiť silou sme upevnili pracovný nástroj.

Senzorom vieme v konečnom dôsledku zistiť, akým tlakom je potrebné tlačiť nástrojom na papier pre získanie požadovanej sýtosti farby. V prípade zmeny hrúbky tuhy tento tlak potom vieme skrz plochu tuhy prepočítať. Tým vieme zaručiť rôznou hrúbkou tuhy rovnakú výslednú farbu. Meranie prítlaku nám môže pomôcť taktiež odhaliť postupné opotrebovanie pružiny v našom nástroji. Taktiež vieme tieto údaje využiť napríklad pri kalibrácii nového nástroja, kde potrebujeme zhodný prítlak.



Obrázok 15: Zapojený senzor s nástrojom na efektore robota[15]

## 5.1 Zhodnotenie

Pri hľadaní informácií ohľadom samotného senzora OptoForce HEX-70-CE-2000N sme natrafili na diplomovú prácu zaoberajúcu sa meraním s týmto senzorom. V samotnej práci je meraná správnosť citlivosti, korekcia konštantnej odchýlky a zisťovanie možného driftu. Na meranie citlivosti pre os Fz boli uskutočnené dve merania. Meranie naprázdno a meranie s konštantnou záťažou 1kg. Výstup senzora bol nastavený:  $1 \text{ kg} = -9,806 \text{ N}$ . Výsledky sú v Tab. 2.

Tab. 2: Merania v osi Fz (citlivosť 4,24)[12]

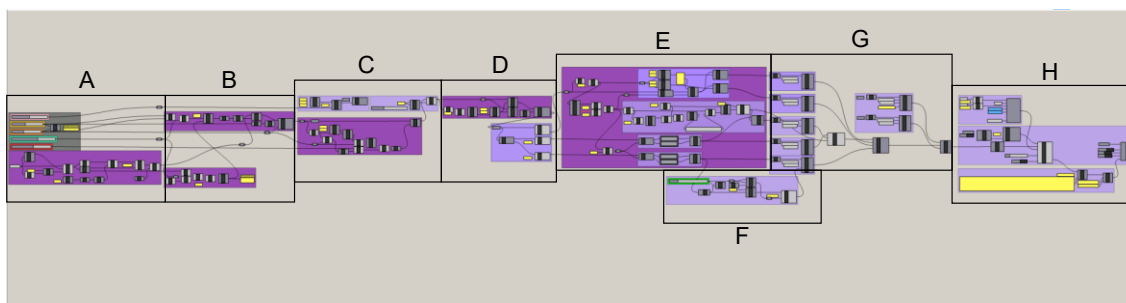
Počet vzoriek	Hmotnosť závažia [kg]	Teoretický výstup [N]	Maximálna hodnota [N]	Minimálna hodnota [N]	Priemerná nameraná hodnota [N]	Priemerná chyba merania[N]
6578	0	0	2,1226	0,7075N	1,4258	1,4258
7898	1	-9,806	-4,0094	-5,8962	-5,2917	3,0885

„Na základe meraní zisťujeme, že dáta zo senzora oscilujú a výstup je zaťažený chybami.“ (Gregor, 2018, s.42). Pri meraní bez závažia bola zaznamenaná priemerná odchýlka 1,4258N a so závažím 3,0885N. To znamená relatívnu chybu tridsať percent. Ďalšími meraniami sa zistilo, že má senzor v osi Fz taktiež aj drift a odchýlka narastá každou hodinou o 0,26924 N.[12]

Vďaka týmto údajom sme usúdili, že tento senzor nie je vhodný na našu problematiku. Pre naše účely by sme potrebovali senzor s čo najvyššou presnosťou a opakovateľnosťou.

## 6 Vytvorenie generátora trajektórií

Celý program sme pre viditeľnosť objektov a pre jednoduchšie opisovanie rozdelili na úseky, ktoré sú zobrazené na obrázku 16.



Obrázok 16: Rozdelenie testovacieho programu na úseky

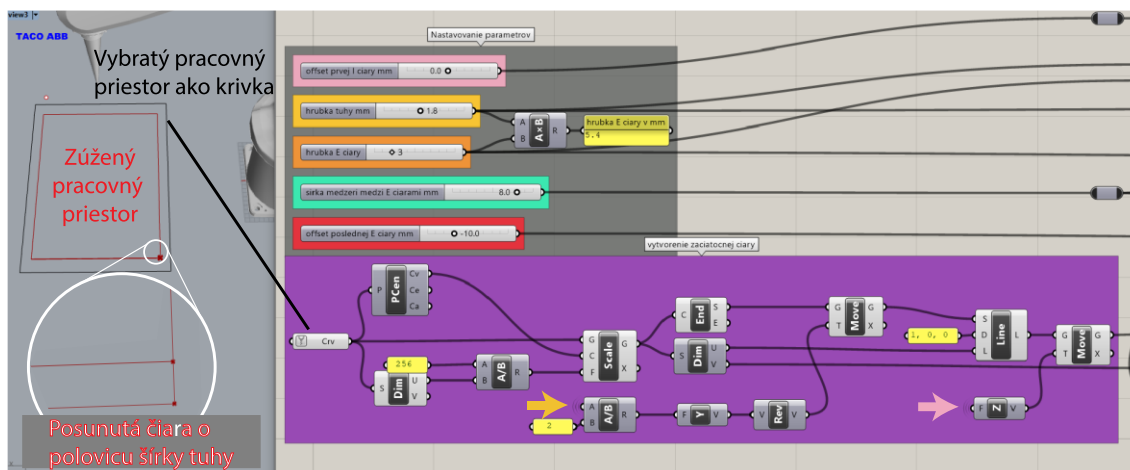
Pri vytváraní programu sme museli zohľadniť parametre, ktorými disponoval náš systém. Do programu tak vstupoval nami vytvorený virtuálny pracovný priestor. Tento priestor sme definovali v Rhinocerosi ako obdĺžnik s určitými rozmermi. Definovanie pracovného priestoru sme rozoberali v podkapitole 4.3.1 Pracovné prostredie.

V Grasshoperi sme potom pracovný priestor označili ako krivku a následne sme boli schopní s touto krivkou pracovať v rôznych operáciách. Najprv sme túto krivku zúžili a vytvorili tak zúžený pracovný priestor, ďalej nazývaný len pracovný priestor. Urobili sme tak pre jednoduchšiu manipuláciu s papierom. Tak sme ho nemuseli nastavovať úplne presne, ale mali sme určitú toleranciu.

### 6.1 Vytvorenie čiar

Pri vytváraní čiar na testovanie a meranie sme si najprv určili začiatočný bod. Umiestnili sme ho v polovici označenej zúženej krivky, pracovného priestoru (ľavý roh bližšie k robotovi). Z tohto bodu sme potom vytvorili čiaru dlhú po celej šírke pracovného priestoru. Tato čiara mala nulovú šírku. Keďže robot prechádza stredom nástroja po tejto čiare, museli sme v programe zohľadniť aj šírku nástroja, tuhy. Aby prvá, tuhou nakreslená čiara bola na okraji pracovného priestoru, museli sme ju v programe posunúť o polovicu hrúbky tuhy. V prípade zámény tuhy za širšiu alebo užšiu sme v programe dali posúvač na nastavenie rovnakej hrúbky. To spôsobilo pri zámene tuhy za inú, s inou hrúbkou potrebné posunutie čiary. Tak bola prvá tuhou nakreslená čiara vždy presne na okraji pracovného priestoru.



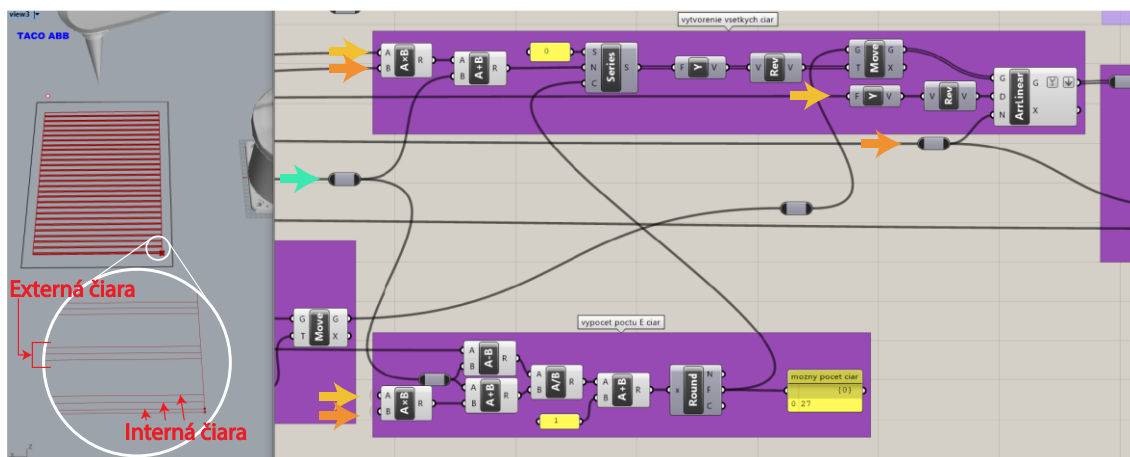


Obrázok 17: Úsek A. Nastavovanie parametrov, zúženie pracovného priestoru a vytvorenie začiatkovej čiary

Pre vykreslenie čiary hrubšej, ako je hrúbka tuhy musíme túto čiaru vytvoriť možným počtom čiar s hrúbkou veľkosti tuhy. Preto sme pre jednoduchosť zmeny šírky použili opäť číselný posúvač, ale vynásobený hrúbkou tuhy. To preto, aby výsledná čiara mala n-násobnú hrúbku tuhy, a aby sa čiary, ktoré ju tvoria neprekrývali.

V ďalšom kroku sme vypočítali koľko takýchto čiar vieme vtesnať na celý pracovný priestor. Najprv sme vypočítali aká bude šírka jednej externej čiary, a to jednoducho vynásobením šírky tuhy a nami zadaným počtom interných čiar pre šírku. K tomuto údaju sme pripočítali ešte nami danú šírku medzery medzi jednotlivými externými čiarami. Pre zmenu šírky tejto medzery sme opäť vytvorili posúvač. Vypočítaným údajom sme vydělili dĺžku pracovného prostredia skráteného o šírku jednej medzery. Tak sme dostali možný počet čiar. Posledná vykreslená čiara bola zároveň čo najbližšie k okraju pracovného priestoru.

V Grasshoperi sme potom len jednoducho prvú internú čiaru vytvorenú na samom začiatku kopírovali a posúvali možným počtom čiar, ktorý sme získali. Posun čiar sme robili o súčet šírky celej vytvorenej čiary a šírky medzery medzi externými čiarami. Takto nám vznikli prvé interné čiary zo všetkých širokých externých čiar. Tie sme potom všetky skopírovali a posunuli o šírku tuhy počtom, ktorý sme zadali pre celkovú šírku externej čiary. Tým sme vytvorili samotnú šírku všetkých externých čiar.



Obrázok 18: Úsek B. Výpočet počtu externých čiar a vytvorenie všetkých čiar

## 6.2 Posun v osi Z

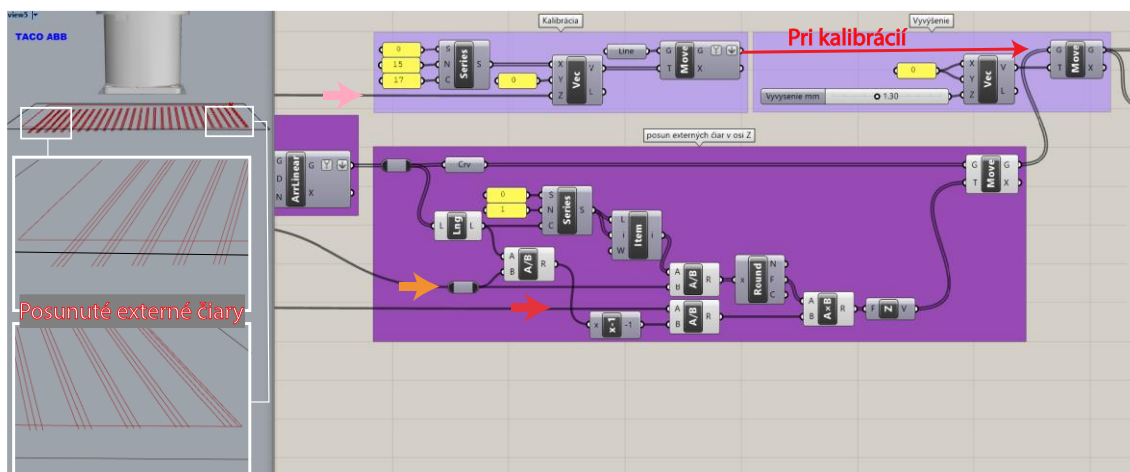
Samotné kreslenie čiar sme navrhli tak, že jednotlivé čiary sme posúvali po zápornej osi Z. Pri reálnom kreslení potom TCP bolo navigované do bodov umiestnených pod pracovným prostredím. Tým robot vyvíjal s nástrojom, tuhou tlak na papier. Čím sme čiary posunuli nižšie, tým robot tlačil viac a kreslil tmavšie čiary. Pri posunutí čiar vyššie zase naopak robot kreslil svetlejšie čiary.

V programe sme najprv určili výšku prvej a poslednej externej čiar skrz numerické posúvače, ktoré sme si na to vytvorili. Zmena výšky prvej internej čiar vytvorenej na začiatku mala za následok zmenu výšky pri všetkých čiarach. Následne sme si výpočtom zistili o koľko má byť každá ďalšia čiara posunutá nižšie alebo vyššie. Vypočítali sme to ako podiel nami danej výšky poslednej externej čiar počtom všetkých externých čiar. Následne sme o násobky tejto hodnoty v osi Z posúvali každú externú čiara. Takto sme dosiahli odstupňovanie robotom vykreslených čiar od svetlej po tmavú alebo naopak.

Pre čo najslabší dotyk nástroja o papier a tým vytvorenie čo najsvetlejšej farby sme vytvorili aj kalibračný blok. Na samom začiatku ešte pred vykresľovaním sme tento blok zapojili ako na obrázku 19. V samotnom bloku sme vytvorili niekoľko čiar po dĺžke pracovného priestoru. Tieto čiary sme potom skrz posúvač posúvali po osi Z, kým sme nedosiahli to, že po skončení programu boli tuhou nakreslené čiary čo najsvetlejšie. Takto sme si skalibrovali náš program a následne boli posuny čiar v programe rátané od tejto hodnoty. Po tejto kalibrácii bol manipulátor schopný tuhou vykresliť čiary od najsvetlejšej po čo najtmavšiu.

Z dôvodu plynulejšieho prechodu robota medzi čiarami sme ešte každú druhú čiara obrátili. Tým sme docielili, že sa robot presúval od čiary k čiare po najkratšej dráhe.

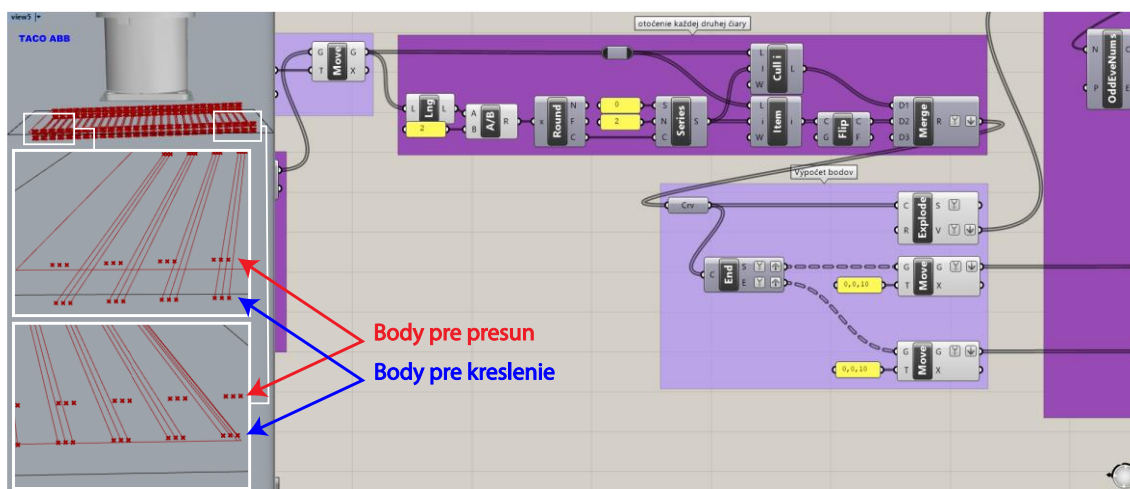




Obrázok 19: Úsek C. Posun externých čiar v osi Z a prípadná kalibrácia

### 6.3 Vytvorenie bodov

Aby sme vedeli robota navigovať do jednotlivých bodov, tak sme si zo všetkých čiar zistili začiatkové a koncové body. Také isté body sme ešte posunuli aj v kladnej osi Z o nami danú výšku. Tak sme vytvorili body na presun medzi čiarami a získali všetky body potrebné ku kresleniu.

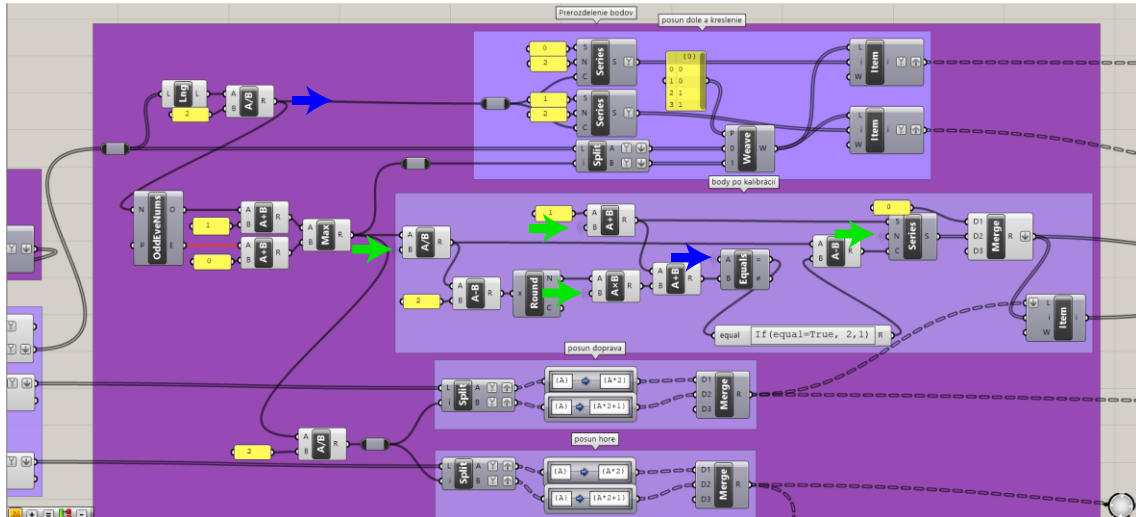


Obrázok 20: Úsek D. Otočenie čiar a vytvorenie všetkých potrebných bodov

Na vytvorené body sme potom boli schopní robota navigovať. Museli sme ešte zadať akou rýchlosťou a aproximáciou mal robot s nástrojom do jednotlivých bodov pristupovať. Rozdelili sme si preto všetky body na samostatné celky:

1. body pre posun smerom dole k pracovnému prostrediu,
2. body kreslenia,

3. body pre posun smerom hore od pracovného prostredia,
4. body pre presun medzi čiarami,
5. body po kalibrácii nástroja.

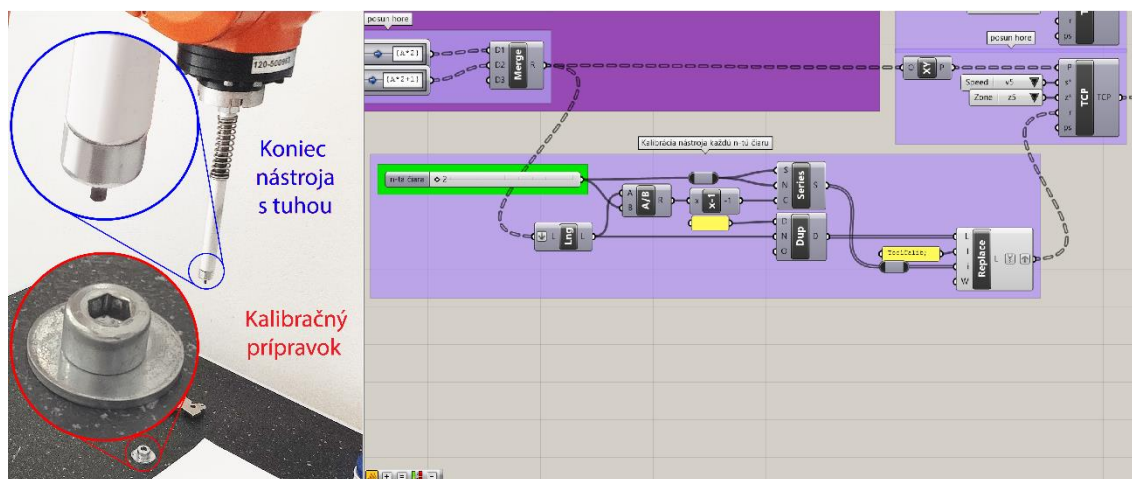


Obrázok 21: Úsek E. Rozdelenie bodov na samostatné celky

## 6.4 Kalibrácia nástroja

Kvôli úbytku tuhy pri kreslení sme museli vytvoriť kalibračný prípravok. Keď robot s nástrojom pritlačil na tento prípravok, tak sa tuha vysunula vždy na rovnakú veľkosť.

Pre tento účel sme si vytvorili v programe blok kalibrácie nástroja. V ňom sme skrz číselný posúvač nastavovali po akom počte vykreslených čiar sa bude nástroj pravidelne kalibrovať. Následne sme za každým týmto počtom vkladali k bodom pre presun od priestoru procedúru na kalibrovanie. Tú sme zadefinovali v programe neskôr. V nej sme najprv určili bod presne nad kalibračným prípravkom. Do tohto bodu sme robota naviedli a následne sme určili bod pre priamy presun dolu pod prípravok. Tak sme zabezpečili zatlačenie nástroja do prípravku a tým vysunutie tuhy. Následne sme určili bod pre presun od prípravku. Potom už mohol robot opäť pokračovať ďalej v kreslení.



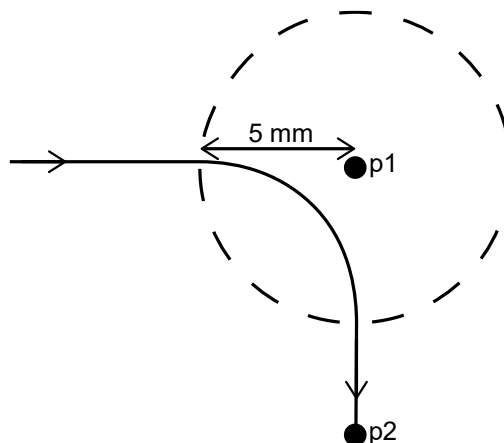
Obrázok 22: Úsek F. Ukážka kalibračného prípravku, nástroja s tuhou a kalibračného bloku v Grasshopperi

## 6.5 Definovanie TCP bodov

Zo všetkých bodov sme potom potrebovali vytvoriť TCP body, do ktorých bol robot navádzaný s nastavenou rýchlosťou a aproximáciou. Rýchlosť je definovaná ako „v100“, kde číslo udáva rýchlosť v mm/s. Aproximácia je stanovená ako „z5“, kde číslo definuje v mm polomer kruhu okolo bodu, do ktorého je robot navigovaný. Dráha robota sa potom v priesečníku tohto kruhu s pôvodnou dráhou začne aproximovať ku ďalšiemu bodu, ako na obrázku 23. Pre robota sú väčšie aproximácie vhodnejšie z dôvodu plynulosti. Motory a kĺby robota sú tak menej namáhané. V našom prípade sme v bodoch nad pracovným priestorom mohli zvoliť určitú aproximáciu. Pri bodoch v pracovnom priestore sme museli túto hodnotu nastaviť na nulu. Urobili sme tak kvôli rovnomernému prítlaku nástroja aj v koncových bodoch vykresľovaných čiar. Hodnoty rýchlosti a aproximácie sme nastavili ako v Tab. 3.

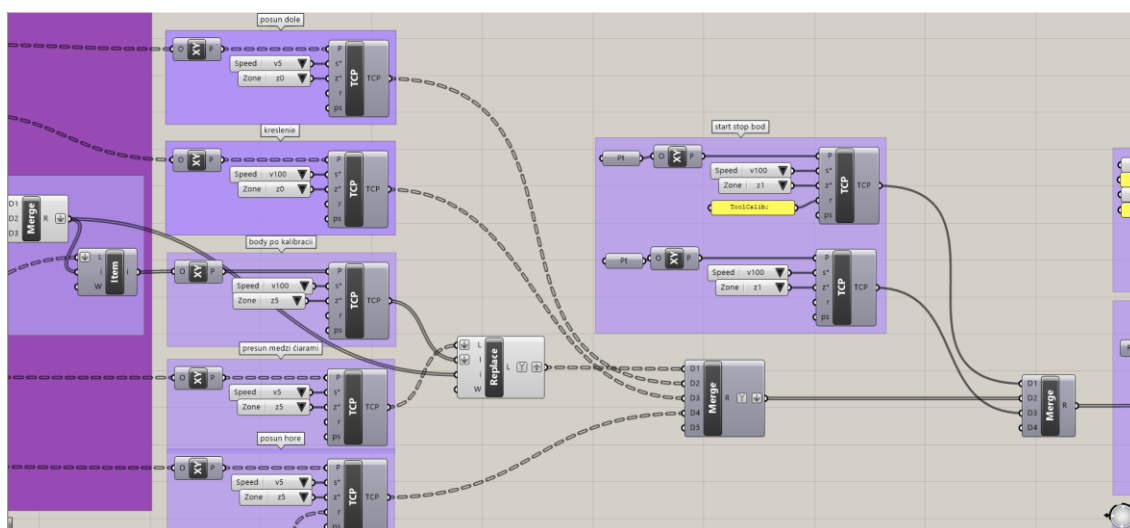
Tab. 3: Nastavené hodnoty rýchlostí a aproximácie pre TCP body

	Posun dole	Kreslenie	Presun medzi čiarami	Posun hore	Po kalibrácií
Rýchlosť[mm/s]	5	100	5	5	100
Aproximácia[mm]	0	0	5	5	5



Obrázok 23: Aproximácia bodu, konkrétne s hodnotou z5

Ku všetkým bodom sme ešte pridali začiatkový a konečný bod, kde robot začal a ukončil vykonávanie programu.



Obrázok 24: Úsek G. Vytvorenie začiatového, koncového a TCP bodov

## 6.6 Spustenie programu

Posledným potrebným krokom bolo pridanie bloku na riešenie inverznej kinematiky a bloku riadiacej jednotky k samotnému modelu robota. Do bloku inverznej kinematiky sme pripojili model robota, nástroja a všetky nami určené TCP body. Výstup z tohto bloku sme následne pripojili do bloku riadiacej jednotky. V ňom sme museli nastaviť IP adresu robota a pridať tlačidlá na nahratie, prípadne spustenie, zastavenie kódu. Následne sme pripojili k robotovi skrz ethernet kábel PC a nastavili ho na lokálnu adresu robota. Uviedli sme reálnu riadiacu jednotku do automatického módu a skrz Grasshopper sme do nej

[illegible]

37

## 7 Meranie sýtosti farieb kamerou

Aby sme dokázali vykresliť požadované farby, tak sme kreslenie porovnávali s už definovanými farbami. K týmto farbám sme sa snažili s našimi nakreslenými vzorkami čo najviac priblížiť, a tým dosiahnuť požadované farby. Týmto spôsobom sme kalibrovali náš kresiaci systém. Manipulátorom vykreslené sýtosti farieb by mali byť podľa úspešnosti kalibrácie čo najviac zhodné s preddefinovanými. Ako vzor na porovnanie sme si vytlačili gradient farieb, tvorený súborom čiar, z už kalibrovanej tlačiarne.

Porovnávanie vzoru a manipulátorom vykreslenej vzorky sme vykonávali pomocou kamery, ktorá dokáže merať jednotlivé farby z obrazu, ktorý sníma. Používali sme model xiQ MQ042CG-CM od spoločnosti XIMEA.

Tieto kamery sa používajú aj na inšpekciu, rýchle snímanie procesov, monitorovanie dopravy, strojové videnie, rozpoznávanie tváre, snímanie pohybu, nárazové skúšky automobilov, 3D skenovanie, bezpečnosť, v lekárstve a pod. Majú veľké využitie vďaka svojej kompaktnosti a mnohým ďalším vlastnostiam.[16]

Tab. 4: Základné parametre kamery xiQ MQ042CG-CM[16]

Názov parametra	Hodnota
Rozlíšenie[pixel]	$2048 \times 2048$
Veľkosť pixelu[ $\mu\text{m}$ ]	$5.5 \times 5.5$
Veľkosť aktívnej oblasti[mm]	$11.27 \times 11.27$
Dynamický rozsah[dB]	60
Citlivosť[V/(Lux s)]	4.64
Digitalizácia[bit]	10
Podporované bitové rozlíšenie[bit/pixel]	8, 10
Obnovovacia frekvencia[fps]	90
Typická spotreba energie[W]	1.5
Spotreba energie vrchol[W]/ [ $\mu\text{s}$ ]	6 / 20

Kvôli jednoduchosti a porovnávaniu vykreslenej vzorky s vytlačenou sme vytvorili gradient farieb ako súbor čiar. Pri meraní sme potom z konkrétnych čiar vedeli jednoducho zmerať sýtosť farby. Aby sa na papier zmestil najväčší možný rozsah farieb, tak sme orientovali čiary kolmo na dĺžku papiera. Farby jednotlivých čiar sme potom stupňovali od čo najsvetlejšej po čo najtmavšiu. Takto sme mohli zistiť aký rozsah farieb je manipulátor schopný vykresliť a taktiež pri akých odtieňoch mal problém. Pre účel vykresľovania čiar manipulátorom sme použili vytvorený generátor trajektórií.

## **7.1 Priebeh merania**

Pred samotným meraním sme museli vhodne umiestniť kameru a taktiež zvoliť vhodné osvetlenie. Kameru sme umiestnili pre potrebu manipulovania na vysúvací stojan. Tento stojan s kamerou sme umiestnili do miestnosti s minimálnym denným svetlom. Urobili sme tak preto, aby sa nám pri meraní nevyskytovali výkyvy v osvetlení. Ako osvetlenie sme teda použili LED panely, konkrétne model VT-6060. Pod kameru sme umiestnili pracovný stôl, na ktorý sme pokladali vzorky na meranie. Kameru sme postupne vyvyšovali, až kým sme nezamerali celú vykreslenú alebo vytlačenú vzorku.

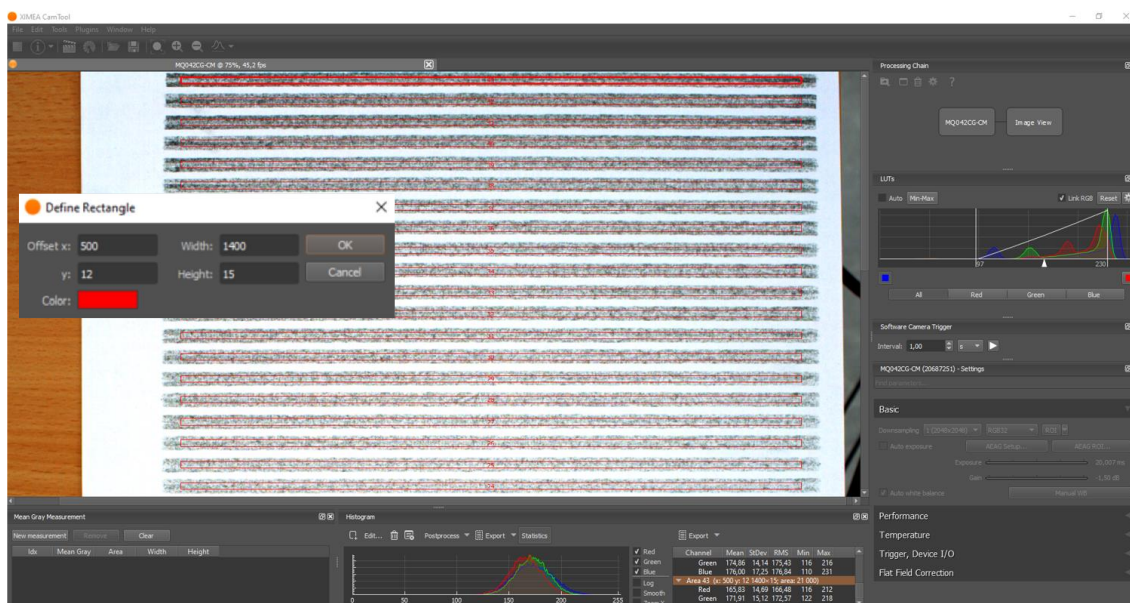
### **7.1.1 Nastavenie programu na meranie**

V používateľskom programe XIMEA CamTool pre zaznamenávanie z kamery sme boli schopní vytvoriť snímacie obdĺžniky. Do nich sme potom mohli zamerať nakreslenú aj vytlačenú vzorku a tiež ich potom na základe hodnôt z kamery porovnať.

Na začiatku sme určili dĺžku snímacích obdĺžnikov približne na tretinu šírky obrazu, ktorý dokáže kamera snímať. Šírku obdĺžnikov sme nastavili čo najužšiu, aby sme boli schopní zosnímať najväčšie množstvo čiar. Taktiež tak, aby sme boli schopní do týchto obdĺžnikov jednoducho zamerať vzorky. Pri meraní sme potom dĺžku a šírku ešte upravovali, aby nepresahovali mimo rozmery vykreslených čiar.

Vzniknutému počtu čiar sme potom prispôbili počet čiar na vykresľovanie manipulátorom a tiež aj na vytlačenie z tlačiarne. Vyskúšaním rôznych variantov sme určili počet čiar konkrétne na štyridsaťtri. Pri tomto počte sme boli schopní ešte pomerne jednoducho zameriavať jednotlivé vzorky a taktiež zamerať dostatočne veľký rozsah farieb. Šírka snímacích obdĺžnikov bola stanovená na 15px a dĺžka na 1400px. Medzera medzi obdĺžnikmi bola určená na 32px.

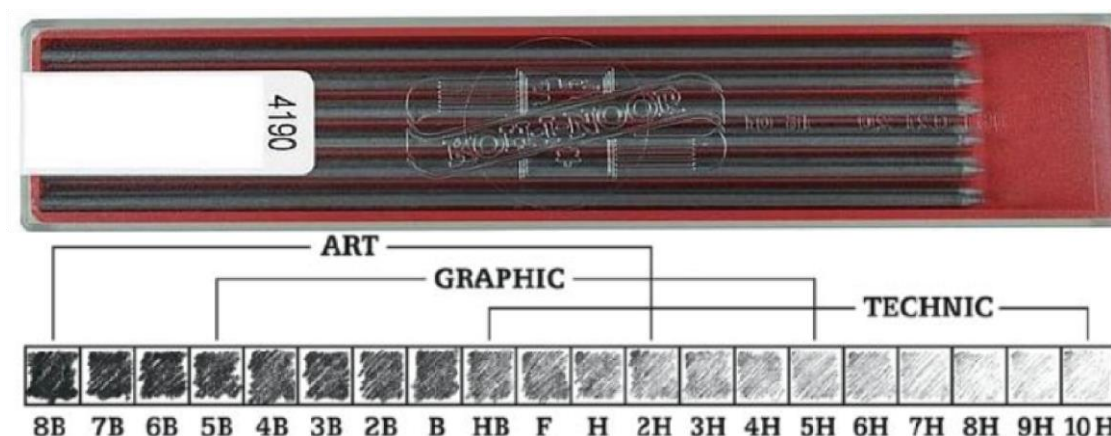




Obrázok 26: Vytvorené meracie obdĺžniky v používateľskom programe CamTool

## 7.1.2 Výber túh

Meraniami sme chceli zistiť sýtosť farieb jednotlivých čiar pri rôznych typoch túh. Vykreslili sme tak pomocou manipulátora osem vzoriek, každú s iným typom tuhy. Tuhy sme vybrali od výrobcu Koh-i-noor, konkrétne typy s tvrdosťou od 8B po B.



Obrázok 27: Typy túh[13]

Tab. 5: Percentuálne zloženie vybraných túh[14]

Označenie	Grafit	Hlina	Vosk
B	0.71	0.23	0.05
2B	0.74	0.20	0.05

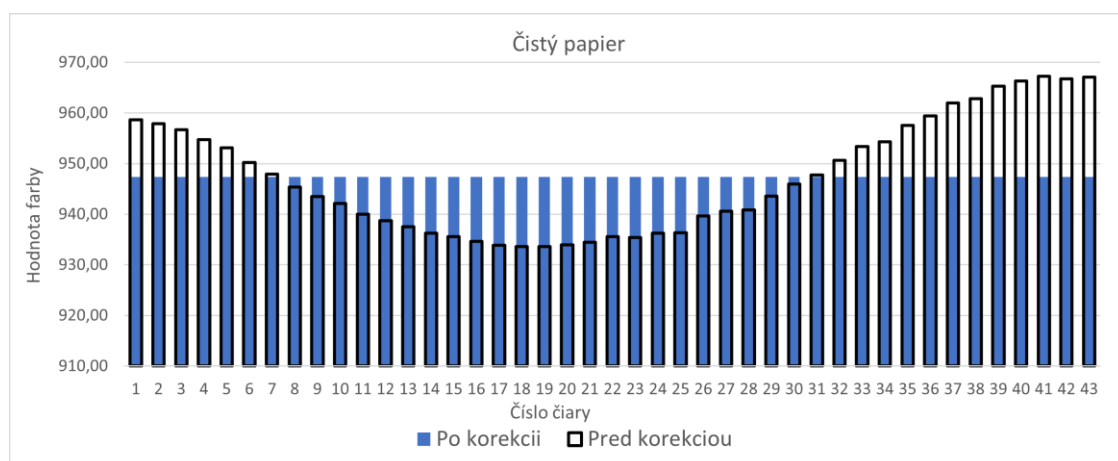


3B	0.76	0.18	0.05
4B	0.79	0.15	0.05
5B	0.82	0.12	0.05
6B	0.84	0.10	0.05
7B	0.87	0.07	0.05
8B	0.90	0.04	0.05

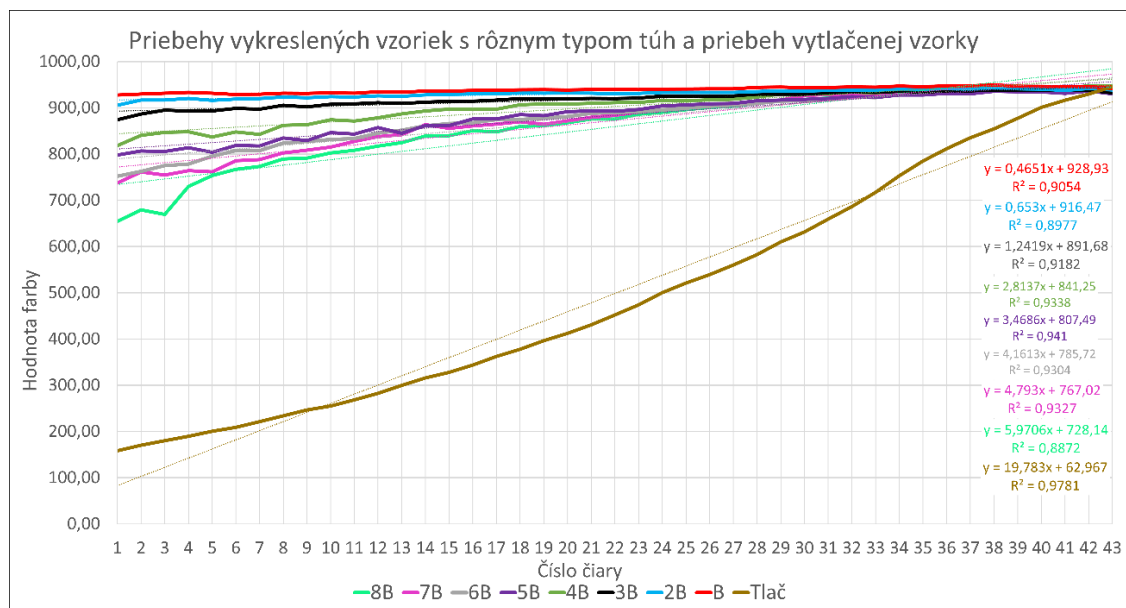
## 7.2 Zhodnotenie merania

Po všetkých nastaveniach sme postupne umiestňovali pod kameru jednotlivé vzorky. Tie sme umiestnili tak, aby jednotlivé meracie obdĺžniky snímali konkrétne čiary. Pre prípadnú korekciu údajov sme taktiež zmerali hodnoty pre čistý papier. Následne sme si vyexportovali hodnoty jednotlivých meraní.

Po vyexportovaní údajov z bieleho papiera sme zistili, že údaje neboli po celej dĺžke papiera lineárne. Bolo to kvôli nerovnomernému osvetleniu a optickým vlastnostiam kamery. Preto sme z údajov, z bieleho papiera zistili priemer. Následne sme odpočítali rozdiel medzi konkrétnym údajom z čiary a zisteným priemerom. Od všetkých ostatných nameraných vzoriek sme potom taktiež odpočítali tieto rozdiely zistené z bieleho papiera. Tak sme urobili celkovú korekciu dát alebo FFT.



Obrázok 28: Korekcia údajov čistého papiera



Obrázok 29: Priebehy všetkých vzoriek

Pri porovnaní vzoriek sme zistili, že čím bola tuha tvrdšia, tak bol rozsah vykreslených farieb nižší. Taktiež sme skonštatovali, že najlepšia tuha na našu problematiku je typ 7B. Tá mala po type 8b najväčší rozsah vykreslených farieb. Uprednostnili sme ju pred typom 8b z dôvodu vyššej linearít.

Pri vykresľovaní vzoriek sme podľa pozorovania usúdili, že je sýtosť farieb pri maximálnom prítlaču takmer nemenná. Z meraných dát sme zistili pravý opak. Vyšší prítlak sme ale kvôli možnosti poškodenia nástroja s robotom nevyvíjali. Ďalším typom merania by mohlo byť meranie viacerých prechodov po jednotlivých čiarach. Na toto meranie sme do budúcnosti už predpripravili program. Z tohto merania by sme pravdepodobne vedeli zistiť maximálny rozsah vykreslených farieb robotom.

# Záver

Cieľom práce bolo pripraviť generátor trajektórií pre šesť osí priemyselný robot, ktorý umožní kreslenie s grafitovou tuhou umiestnenou na pasívnom pružnom nástroji.

Pred samotnou prípravou sme sa oboznámili s rôznymi typmi priemyselných robotov. Získali sme si informácie o ich konštrukciách a programovaní. Naučili sme sa pracovať s konkrétnym typom priemyselného robota ABB IRB120.

Následne sme nadobudli vedomosti programovania v grafickom programovacom jazyku Grasshoper. Grasshoper je súčasťou CAD modelovacieho programu Rhinoceros, s ktorým sme sa teda taktiež oboznámili. S pomocou pluginu Taco ABB sme vytvorili virtuálny pracovný priestor zhodný s reálnym. So získanými vedomosťami sme potom v Grasshoperi vytvorili generátor trajektórií, ktorý zo vstupu parametrov, čiár vytvára bitmapový obrazec. Vďaka pluginu Taco ABB sme vedeli vygenerovať a nahráť vytvorený program do riadiacej jednotky robota.

Urobili sme taktiež rôzne merania a z dát zanalyzovali výsledky. Pre zistenie akým tlakom pôsobí nástroj na papier sme urobili meranie prítlaku. Zo získaných informácií sme zistili, že dáta zo senzora oscilujú a výstup je zaťažený chybami. Preto sme usúdili nevhodnosť konkrétneho snímača sily Optoforce HEX-70-CE-2000N.

Ďalším meraním bolo optické meranie s rôznymi typmi túh. Pre toto meranie sme vykreslili s robotom pomocou vytvoreného generátora vzorky s rozdielnymi odtieňami. Oboznámili sme sa s kamerou XIMEA xiQ MQ042CG-CM, s ktorou sme vykonávali merania a taktiež s jej užívateľským programom CamTool. Po oboznámení sa, sme vytvorili program na meranie, vytvorený zo snímacích obdĺžnikov. Pod kameru sme následne umiestňovali vzorky tak, aby snímacie obdĺžniky snímali konkrétne čiary. Na porovnanie sme použili vzorky, ktoré vykreslil robot pomocou generátora a vzorovú vzorku vytlačenú z kalibrovannej tlačiarne. Pri každej robotom vykreslenej vzorke bol použitý iný typ tuhy. Z vyexportovaných dát sme zanalyzovali, že najvhodnejším typom tuhy na kreslenie s robotom bol v našom prípade typ 7B. Usúdili sme tak vďaka najväčšiemu rozsahu a zároveň linearite priebehu vykreslených farieb pri tomto type.

Ďalším meraním do budúcnosti by mohlo byť meranie sýtosti farieb pri viacerých prechodoch po jednotlivých čiarach. Aj týmto meraním by sa získali dáta potrebné na vytvorenie matematického modelu zmiešavania odtieňov.

# Literatúra

- [1] CHVÁLA, B. – MATIČKA, R. – TALÁCKO, J. 1990. *Průmyslové roboty a manipulátory*. Praha : SNTL, 1990. 275 s. ISBN 80-03-00361-X.
- [2] 2011. Příručka Automatizační a robotická technika. In *VITRALAB* [online]. 2011 [cit. 2021-02-20]. Dostupné na internete: < [http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka\\_cz\\_final.pdf](http://www.sjf.tuke.sk/vitralab/files/vystupy/prirucka_cz_final.pdf) >.
- [3] LEE, J. 2019. Computer-Aided Architectural Design. "Hello, Culture". Singapore : Springer, 2019. ISBN 978-981-13-8409-7.
- [4] 2018. *How Are Industrial Robots Built? A Guide on the Components and the Movement of Robot Arms*. In *XYZ* [online]. 2018 [cit. 2009.06.10]. Dostupné na internete: < <https://robotics.kawasaki.com/ja1/xyz/en/1804-03/> >
- [5] 2018. *What Kinds of Industrial Robots Are There? A Guide on the Features of the Major 6 Types*. In *XYZ* [online]. 2018 [cit. 2021-02-20]. Dostupné na internete: < <https://robotics.kawasaki.com/ja1/xyz/en/1803-01/> >
- [6] 2017. *Industrial Robot ABB IRB 120 3D model*. In *cgtrader* [online]. 2017 [cit. 10-5-2020]. Dostupné na internete: < <https://www.cgtrader.com/3d-models/industrial/machine/industrial-robot-abb-irb-120> >
- [7] 2021. *Technical data for the IRB 120 industrial robot*. In *ABB* [online]. 2021 [cit. 10-5-2020] dostupné na internete: < <https://new.abb.com/products/robotics/industrial-robots/irb-120/irb-120-data> >
- [8] KOLÍBAL, Z. 2016. Roboty a robotizované výrobní technologie. Brno : Vysoké učení technické v Brně - nakladatelství VUT IUM, 2016. 787s. ISBN 978-80-214-4828-5.
- [9] TRNKA, K. – BOŽEK, P. Simulácia a off-line programovanie robotov. In *engineering* [online]. [cit. 2021-02-25]. Dostupné na internete: < <https://www.engineering.sk/clanky2/automatizacia-robotizacia/686-simulacia-a-off-line-programovanie-robotov> >.
- [10] 2020. Operating manual IRC5 with FlexPendant. In *ABB ROBOTICS* [online]. 2020 [cit. 2021-03-10]. Dostupné na internete: <

<https://abb.sluzba.cz/Pages/Public/IRC5RoboticsDocumentationRW6/Controllers/IRC5/FlexPendant/cz/3HAC050941-014.pdf>>.

- [11] JOST, D. 2020. What is a force torque sensor? In *FIERCE Electronics* [online]. 2020 [cit. 2021-03-15]. Dostupné na internete: <<https://www.fierceelectronics.com/sensors/what-a-force-torque-sensor>>.
- [12] GREGOR, R. 2018. Prepojenie silového senzora a robotického manipulátora v ROS : diplomová práca. Bratislava : STU FEI, 2018. 98 s.
- [13] 2017. Obyčajná ceruzka KOH-I-NOOR. In *Materiály pre knižných umelcov* [online]. 2017 [cit. 2021-04-20]. Dostupné na internete: <<http://hingan.ru/prostoy-karandash-koh-i-noor/>>.
- [14] SOUSA, M. – BUCHANAN, J. 2000. Observational Models of Graphite Pencil Materials. In *COMPUTER GRAPHICS forum* [online]. 2000, vol. 19, no. 1 [cit. 2021-04-20]. Dostupné na internete: <<https://www.visgraf.impa.br/Courses/npr07/materials/2%20lapis/Observational%20Models%20of%20Graphite%20Pencil%20Materials.pdf>>.
- [15] 2016. 6-Axis F/T Sensor HEX-70-CE-2000N. In *schlu.com* [online]. 2016 [cit. 2021-04-22]. Dostupné na internete: <[https://www.schlu.com/pdf/Optoforce\\_Sensore\\_di\\_forza\\_6D\\_HEX-70-CE-2000N\\_1.5\\_EN.pdf](https://www.schlu.com/pdf/Optoforce_Sensore_di_forza_6D_HEX-70-CE-2000N_1.5_EN.pdf)>.
- [16] 2019. xiQ USB 3.0 camera series Technical Manual. In *ximea.com* [online]. 2019 [cit. 2021-04-25]. Dostupné na internete: <[https://www.ximea.com/downloads/usb3/manuals/xiq\\_technical\\_manual.pdf](https://www.ximea.com/downloads/usb3/manuals/xiq_technical_manual.pdf)>.

# Prílohy

## **Príloha A: GTAMP.rar – obsahuje generátor trajektórií a model prostredia**

Na gitlab repozitári : <https://gitlab.com/xpoltak/generator-trajektorii>



# Príloha B: Odfotené vzorky na meranie sýtosti farieb

