

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STROJNÍCKA FAKULTA**

**VYUŽITIE POČÍTAČOVÉHO VIDENIA NA  
ROBOTICKÝCH PRACOVISKÁCH**

**Diplomová práca**

**SjF-5226-87762**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
STROJNÍCKA FAKULTA**

**VYUŽITIE POČÍTAČOVÉHO VIDENIA NA  
ROBOTICKÝCH PRACOVISKÁCH**

**Diplomová práca**

SjF-5226-87762

Študijný program:           Automatizácia a informatizácia strojov a procesov  
Študijný odbor:            Kybernetika  
Školiace pracovisko:      Ústav automatizácie, merania a aplikovanej informatiky  
Vedúci záverečnej práce: doc. Ing. Ján Vachálek, PhD.  
Konzultant:                Ing. Ivan Fiřka

**Bratislava, 2021**

**Bc. Marek Sulír**



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Marek Sulír**  
ID študenta: 87762  
Študijný program: automatizácia a informatizácia strojov a procesov  
Študijný odbor: kybernetika  
Vedúci práce: doc. Ing. Ján Vachálek, PhD.  
Konzultant: Ing. Ivan Fiťka  
Miesto vypracovania: Ústav automatizácie, merania a aplikovanej informatiky

Názov práce: **Využitie počítačového videnia na robotických pracoviskách.**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

1. Teoreticky naštudujte problematiku počítačového videnia. V oblastiach sa sústreďte na technológie súčasne dostupné na našom ústave v rámci 2D a 3D priestorov.
2. Na základe naštudovanej teórie navrhnete využitie počítačového videnia v rámci priemyselného využitia vkladania a odoberania súčiastok do robotického pracoviska.
3. Navrhnuté robotické pracovisko realizujte za pomoci off-line návrhu a preukážte jeho funkčnosť.

Rozsah práce: 60 – 80 strán

Riešenie zadania práce od: 15. 02. 2021

Dátum odovzdania práce: 28. 05. 2021

**Bc. Marek Sulír**  
študent

**prof. Ing. Cyril Belavý, CSc.**  
vedúci pracoviska

**prof. Ing. Cyril Belavý, CSc.**  
garant študijného programu

## Čestné prehlásenie

Vyhlasujem, že som záverečnú prácu vypracoval samostatne s použitím uvedenej literatúry.

Bratislava, 28. mája 2021

.....  
Vlastnoručný podpis



Ďakujem vedúcemu diplomovej práce, doc. Ing. Jánovi Vachálkovi, PhD., za odbornú pomoc pri vypracovaní diplomovej práce. Chcem poďakovať aj konzultantovi diplomovej práce, Ing. Ivanovi Fitkovi, za pomoc a pripomienky pri návrhu a realizácii úlohy na robotizovanom pracovisku.

Bratislava, 28. mája 2021

Bc. Marek Sulír

**Názov práce:** Využitie počítačového videnia na robotických pracoviskách

**Kľúčové slová:** počítačové videnie, manipulačná úloha, robotizované pracovisko, 3D skener Photoneo, priemyselný robot KUKA

**Abstrakt:** Táto diplomová práca popisuje riešenie manipulačnej úlohy s použitím počítačového videnia na robotizovanom pracovisku, ktoré sa nachádza v laboratóriu ústavu. Práca je rozdelená do postupnosti bodov, ktoré zahŕňajú teoretické informácie, návrh a následnú realizáciu použitia počítačového videnia na robotickom pracovisku. V prvom bode sa nachádzajú teoretické informácie určené na pochopenie základnej logiky systémov počítačového videnia. Ďalší bod zohľadňuje nadobudnuté vedomosti z tejto oblasti s následným návrhom a implementáciou aplikácie počítačového videnia do robotizovaného pracoviska. Posledný bod obsahuje realizáciu pracoviska a výpis programovej úlohy s vysvetlením jednotlivých bodov.

**Title:** Use of the vision system at robotized workplaces

**Keywords:** vision system, manipulating task, robotized workplace, 3D scanner Photoneo, industrial KUKA robot

**Abstract:** This diploma thesis describes the solution of manipulating task with using the vision system at robotized workplace located in laboratory of the institute. Thesis is divided into sequence of points which include theoretical information, plan and consecutive realisation of use of the vision system at robotical workplace. There are mentioned theoretical information in the first point intended for understanding the basic logic of systems of the vision system. Next point focuses on acquired knowledge in this sphere with consecutive plan and application implementation of the vision system in the robotized workplace. The last point includes realisation of workplace and the listing of the program's task with explanation of individual points.

# Obsah

Úvod	5
<b>1 Systémy strojového videnia</b>	<b>6</b>
1.1 2D systémy	6
1.1.1 Využitie	6
1.1.2 Obmedzenia	7
1.1.3 Technológia	8
1.2 3D systémy	8
1.2.1 Využitie	9
1.2.2 Príklad použitia	10
1.2.3 Technológia	11
1.3 Porovnanie 2D a 3D systémov	12
1.4 Výrobcovia	13
1.4.1 Cognex Corporation	13
1.4.2 SICK AG	14
1.4.3 Photoneo s.r.o.	15
<b>2 Návrh aplikácie a implementácie 3D skenera Photoneo PhoXi do robotizovaného pracoviska</b>	<b>17</b>
2.1 Popis zariadení na pracovisku	17
2.1.1 Photoneo 3D skener PhoXi	17
2.1.2 Riadiaci systém Vision	20
2.1.3 Robotický systém KUKA KR 3 R540	22
2.2 Popis navrhutej úlohy	27
2.3 Topológia použitých zariadení	28
2.4 Návrh koncového efektora manipulátora	28
2.5 Rozmiestnenie zariadení na robotizovanom pracovisku	29
<b>3 Popis realizácie robotizovaného pracoviska a programovej úlohy</b>	<b>31</b>
3.1 Nastavenie robotického systému na Photoneo Bin Picking Studio	31
3.1.1 Nastavenie ovládača KUKA a konfigurácia siete	31
3.1.2 Nastavenie modulu konfigurácie Photoneo KUKA	32
3.1.3 Programové volania KRL	38
3.2 Konfigurácia Bin Picking Studia	40
3.2.1 Konfigurácia riešenia	40
3.2.2 Metóda uchopenia	43

3.2.3	Systémové videnie . . . . .	47
3.2.4	Konfigurácia siete . . . . .	48
3.2.5	Kalibrácia . . . . .	50
3.2.6	Tvorba prostredia . . . . .	52
3.3	Realizácia . . . . .	53
3.3.1	Spustenie Bin Picking Studia . . . . .	53
3.3.2	Spustenie programu KUKA . . . . .	54
3.3.3	Popis realizácie programovej úlohy . . . . .	55
3.3.4	Výpis programu manipulačnej úlohy . . . . .	60
<b>4</b>	<b>Záver</b>	<b>66</b>

# Zoznam obrázkov

1.1	Princíp 2D systému [1]	6
1.2	Area Scanning [3]	8
1.3	Line Scan [3]	9
1.4	Princíp 3D systému [3]	9
1.5	Príklad použitia 3D systémov [10]	10
1.6	Laser Triangulation [3]	11
1.7	Time of Flight [3]	12
1.8	Stereo [3]	12
1.9	Vision systém Cognex [10]	14
1.10	Vision systém SICK [7]	15
1.11	Zorné pole v mm (inch) TriSpectorP1000 [7]	15
1.12	Vision systém Photoneo [8]	16
2.1	3D skener PhoXi L [10]	18
2.2	Rozsah skenovania PhoXi L [10]	18
2.3	Príklad použitia v potravinárskom priemysle [10]	19
2.4	Príklad použitia kontroly lesklých povrchov [10]	19
2.5	Technológia kazety riadiaceho systému Vision [13]	20
2.6	Zadná strana Vision riadiaceho systému [13]	21
2.7	Predná strana Vision riadiaceho systému [13]	21
2.8	Robotický systém KUKA KR 3 R540 [14]	22
2.9	Manipulátor KUKA KR 3 R540	23
2.10	Pracovný priestor manipulátora v mm [19]	24
2.11	Riadiaci systém KUKA KR C4 Compact	25
2.12	Predná strana KUKA SmartPad-u [15]	26
2.13	Zadná strana KUKA SmartPad-u [15]	27
2.14	Zariadenia a súčasti navrhutej úlohy [15, 21]	27
2.15	Topológia použitých zariadení [14, 15, 21]	28
2.16	3D model koncového efektoru manipulátora	29
2.17	Rozmiestnenie zariadení na robotizovanom pracovisku	30
3.1	Sieťová konfigurácia riadiaceho systému manipulátora	32
3.2	Reštart riadiaceho systému manipulátora po nastavení sieťovej konfigurácie	33
3.3	Nahratie priečinku EthernetKRL Config	34
3.4	Nastavenie IP adresy Vision riadiaceho systému	34
3.5	Nastavenie IP adresy riadiaceho systému manipulátora	35

3.6	Odvolanie Submit Interpretera . . . . .	36
3.7	Výber \$ NULLFRAME pre rámce výber nástroja a výber základu . . . . .	36
3.8	Zvolenie programu main_ application.src . . . . .	37
3.9	Potvrdenie zmeny pozície manipulátora . . . . .	37
3.10	Konfigurácia riešenia . . . . .	41
3.11	Konfigurácia manipulátora . . . . .	42
3.12	Konfigurácia nástroja . . . . .	42
3.13	Konfigurácia objektu . . . . .	43
3.14	Definovanie uchopovacieho bodu nástroja . . . . .	43
3.15	Výber odoberanej súčiastky . . . . .	44
3.16	Definovanie uchopovacieho bodu súčiastky . . . . .	44
3.17	Definovanie fázy štart . . . . .	45
3.18	Definovanie fázy priblíženia . . . . .	46
3.19	Definovanie fázy uchopenia . . . . .	46
3.20	Definovanie fázy vzdialenia . . . . .	47
3.21	Definovanie fázy koniec . . . . .	48
3.22	Nastavenie systémového videnia . . . . .	49
3.23	Komunikačné rozhranie manipulátora . . . . .	49
3.24	Komunikačné rozhranie skenera . . . . .	50
3.25	Komunikačné rozhranie riadiaceho systému manipulátora . . . . .	50
3.26	Prvý identifikovaný bod kalibrácie . . . . .	51
3.27	Celková presnosť kalibrácie . . . . .	51
3.28	Tvorba scény prostredia . . . . .	52
3.29	Nastavenie manipulátora v prostredí . . . . .	53
3.30	Aktualizácia nastaveného prostredia . . . . .	53
3.31	Užívateľská obrazovka Bin Picking Štúdia . . . . .	54
3.32	Spustenie programu KUKA . . . . .	55
3.33	Skenovanie pracovného priestoru . . . . .	56
3.34	Nájdanie objektov skenerom . . . . .	56
3.35	Simulačný pohľad na manipulátor pri odoberaní objektu . . . . .	57
3.36	Fáza štart . . . . .	57
3.37	Fáza priblíženia . . . . .	57
3.38	Fáza uchopenia . . . . .	58
3.39	Fáza vzdialenia . . . . .	58
3.40	Fáza koniec . . . . .	59
3.41	Pozícia odkladania súčiastky . . . . .	59
3.42	Pozícia uloženia súčiastky . . . . .	59
3.43	Vzdialenie od uloženej súčiastky . . . . .	60
3.44	Grafické zobrazenie behu programu . . . . .	61

# Zoznam tabuliek

2.1	Parametre PhoXi L [10] . . . . .	20
2.2	Technické údaje manipulátora KUKA KR 3 R540 [19] . . . . .	23
2.3	Parametre osí manipulátora KUKA KR 3 R540 [19] . . . . .	24

# Zoznam skratiek a jednotiek

atď. - a tak ďalej

CAD - computer-aided design, počítačové projektovanie

CAM - Computer Aided Manufacturing, počítačom riadená výroba

GB - gigabyte, jednotka kapacity pamäťových médií

GPU - graphic processing unit, grafický procesor

HDMI - High-Definition Multimedia Interface, digitálne rozhranie pre multimediálne zariadenia

kg - kilogram, fyzikálna jednotka hmotnosti

KRL - KUKA Robot Language, programovací jazyk KUKA

KRC - Knowledge, Responsibility & Control, ovládacia časť robota KUKA

mm - milimeter, fyzikálna jednotka dĺžky

PTP - point-to-point, pohyb robota pod po bode

RGB - Red - Green - Blue, farebný model

TCP - Tool Center Point, centrovací bod

TCP/IP - Transmission Control Protocol/Internet Protocol, prenosový protokol sieťovej vrstvy

USB - Universal Serial Bus, univerzálna sériová zbernica

W - watt, fyzikálna jednotka výkonu

° - stupeň

% - percento



# Úvod

V súčasnosti sa čoraz viac uplatňuje automatizácia a robotizácia s použitým strojovým videním, a to v manipulačných alebo technologických úlohách rôznych odvetví. Uvedenie do prevádzky a následná obsluha robotov, ktoré opticky vnímajú hmotné veci v ich zornom poli vyžaduje vedomosti a skúsenosti vo viacerých oblastiach ako len teoretické znalosti programovania. Je potrebné nadobudnúť vedomosti a s nimi spojené skúsenosti so všeobecnou znalosťou elektroniky, chápania logiky programových úloh, programovania atď.

Cieľom diplomovej práce je získať vedomosti v problematike programovania priemyselných robotov s využitím strojového videnia a následne tieto vedomosti použiť v návrhu a realizácii takéhoto pracoviska. Mojim zámerom v tejto práci bude nadviazať na moju bakalársku prácu s názvom Programovanie robotizovaných pracovísk s priemyselným robotom KUKA KR3R540. Po úspešnej realizácii zadania mojej diplomovej práce sa robotizované pracovisko na našom ústave stane omnoho komplexnejším a vytvorí tak lepší praktický príklad pre študentov, ktorí sa budú zaoberať témou robotiky, strojového videnia s najmodernejšími technológiami alebo budú mať túto problematiku v rámci študijného plánu.

Práca je rozdelená do teoretickej a praktickej časti spojenej s návrhom manipulačnej úlohy. Teoretická časť obsahuje popis, vysvetlenie a základnú logiku problematiky 2D a 3D systémov strojového videnia s ich následným porovnaním a spomenutím popredných výrobcov týchto systémov.

V praktickej časti návrhu využitia počítačového videnia v rámci priemyselného využitia vkladania a odoberania súčiastok je popísaný návrh aplikácie a implementácie počítačového videnia v robotizovanom pracovisku. V tomto návrhu sú popísané použité zariadenia, ich vzájomná komunikácia a návrh s následným rozmiestnením zariadení na robotizovanom pracovisku.

Posledným bodom práce je realizácia robotizovaného pracoviska. V tomto bode je popísaná konfigurácia manipulátora na použité softvérové prostredie s následnou konfiguráciou tohto softvérového prostredia. Na záver tohto bodu je popísané spustenie programov potrebných na realizáciu robotizovaného pracoviska s využitím strojového videnia a popis realizácie programovej úlohy.

# 1 Systémy strojového videnia

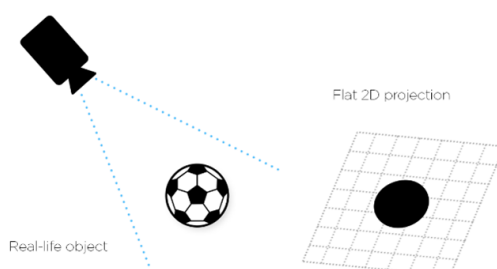
2D systémy strojového videnia sú nasadené už niekoľko desaťročí, čo je dôkazom ich účinnosti. Robia svoju prácu dobre v širokej škále aplikácií, čiže vývin 3D systémov im neublížil. Existuje však veľa druhov výrobných liniek, kde praktické obmedzenia 2D znamenajú, že sa jednoducho nedajú použiť, a tu už potrebujeme nové systémy strojového videnia [1, 2].

## 1.1 2D systémy

V prípade 2D systému strojového videnia je obraz cieľového objektu získaný na spracovanie v skutočnosti plochým dvojrozmerným pôdorysom. 2D obraz neposkytuje vôbec žiadne informácie o výške - sú tu údaje X a Y, ale žiadne údaje o hĺbke ostrosti osi Z. [1, 2].

### 1.1.1 Využitie

To, čo vidíme, sú skutočné kontúry 3D objektu pri pohľade z určitého uhla. Rôzne uhly pohľadu a rôzne objekty vytvárajú úplne odlišné kontúry, vďaka čomu je 2D strojové videnie obmedzené na použitie v aplikáciách, kde sú informácie o tvare pre vykonanie úlohy rozhodujúce [1, 2].



Obr. 1.1: Princíp 2D systému [1]

Nedostatok výšky a skutočných informácií o tvare však nemá pre mnoho primerane jednoduchých aplikácií žiadny dopad. Výsledkom je, že systémy strojového videnia 2D (buď plošného, alebo líniového skenovania) sa široko používajú v celom priemysle pri širokej škále úloh. [1, 3]. Tie obsahujú:

- overenie vlastností a polohy,
- kontrola rozmerov,
- čítanie čiarových kódov,
- overenie štítku,
- kontrola kvality,
- dozor a sledovanie objektov,
- detekcia prítomnosti.

Vo všetkých takýchto aplikáciách je presný 2D obraz odvodený od variácií intenzity odrazeného svetla na povrchu cieľového objektu a od kľúčových vlastností. Závisí to od kontrastov identifikovaných buď v odtieňoch šedej, alebo na farebnom obrázku objektu. A tu leží prvá zo série praktických výziev pre systémy 2D strojového videnia [1, 3].

### 1.1.2 Obmedzenia

Pretože obraz cieľového objektu je tvorený zo svetla, ktoré sa od neho odráža, môžu mať zmeny osvetlenia v zornom poli v dôsledku zmien okolitých podmienok alebo umelého osvetlenia nepriaznivý vplyv na presnosť. Príliš veľa svetla, príliš málo svetla alebo tieňovanie v prostredí továrne môže nepriaznivo ovplyvniť čistotu hrán a prvkov objavujúcich sa v 2D rovine. Citlivosť na osvetlenie je teda problém [1, 3].

Navyše, keďže sú to práve tie jasné kontrasty (a tým jasné hrany a prvky) na povrchu cieľového objektu, od ktorých závisí 2D strojové videnie, bude sa tiež snažiť zvládnuť veľmi tmavé aj veľmi lesklé cieľové objekty. Aj keď existuje veľa rôznych techník na rozsvietenie objektu, jeho hrany a vlastnosti nemožno len tak zistiť. Nedostatok kontrastu predstavuje problém [1, 4].

Taktiež, keď pri 2D strojovom videní nespracovávame žiadne informácie o výške, predstavujú chyby v dôsledku pohybu cieľového objektu v rovine Z ďalšie obmedzenie. Keby objekt vždy sedel na dokonale rovnom povrchu v presnej ohniskovej vzdialenosti od obrazového snímača, potom by to nemal byť problém s presnosťou zobrazenia [1, 3].

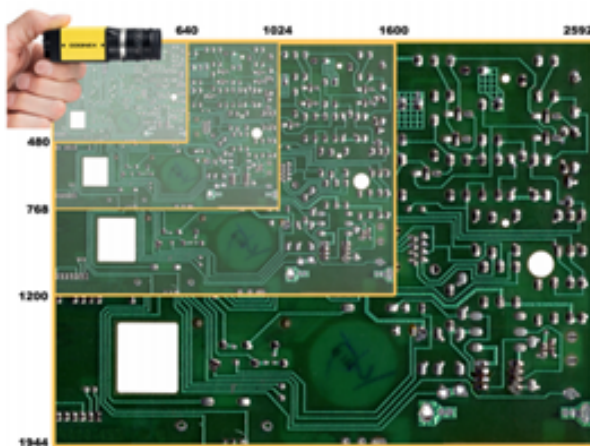
Ako už bolo povedané, systémy 2D strojového videnia nemôžu zvládnuť zložitosti trojrozmerného tvaru alebo formy. Najmä pre zložité diely a zostavy, kde je potrebné merať rozmery inak, ako v rovine X alebo Y, kde je potrebné určiť objem časti, alebo keď je potrebné časť vybrať a umiestniť presne. V takto podmienených úlohách potom 2D strojové videnie nezodpovedá danej úlohe, teda nerozpozna tvar [1, 2].

### 1.1.3 Technológia

V oblasti 2D vision systémov poznáme dve základné technológie tvorenia obrazu. Obraz je vyhotovený buď ako skenovanie určitej plochy (Area Scanning), alebo vytváraním obrazu na základe skenovania jednotlivých línií a ich následným skladaním do obrazu (Line by Line scanning)[1, 2].

- Area Scanning

Väčšina bežných inšpekčných kamier vykonáva skenovanie oblasti. Pri statických foteniach, kde má súčiastka pevnú statickú polohu, je možné využitie najčastejšie využíwanej metódy 2D kamerových systémov, a to Area Scanning, ktoré zahŕňa snímanie 2D snímok v rôznych rozlíšeniach podľa typu kamery. Znáznornenie tejto technológie vid' obrázok 1.2 [1, 2].



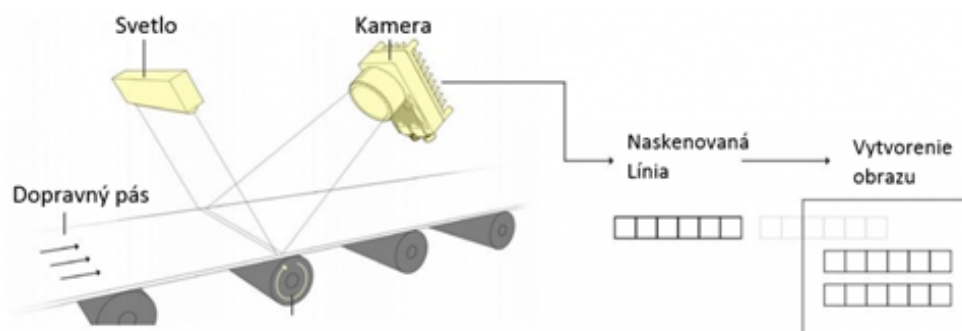
Obr. 1.2: Area Scanning [3]

- Line by Line scanning (Line Scan)

Ďalším typom 2D strojového videnia je Line Scan, kde sa vytvára 2D obraz po riadku. Line Scan prevažuje predovšetkým pri inšpekcii guľatých, oválnych súčiastok vyžadujúcich si viac skenovaní na pokrytie celého povrchu. V takomto prípade je častým využívaním rotovanie skenovaného objektu cez snímanú líniu 2D kamery a následného skladania obrazu. V podstate môžeme povedať, že Line Scan je výhodnejší tam, kde je skenovaná súčiastka vedená kontinuálnym pohybom. Znáznornenie tejto technológie vid' obrázok 1.3 [1, 2].

## 1.2 3D systémy

V systéme 3D strojového videnia už nie je obraz cieľového objektu iba plochým obrazom. Teraz je to trojrozmerné mračno bodov s presnými súradnicami, kde je známa poloha



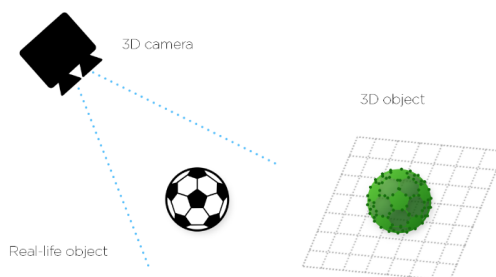
Obr. 1.3: Line Scan [3]

každého pixelu vo vesmíre. Súčasne poskytuje tiež údaje o rovine X, Y a Z spolu s príslušnými informáciami o rotácii (okolo každej z osí) [1, 2].

### 1.2.1 Využitie

V porovnaní s dvojrozmerným spracovaním obrazu je práca v 3D časovo náročnejšia na procesor a softvér, avšak rýchly pokrok vo viacjadrových procesoroch, 3D algoritmoch a softvérových nástrojoch znamená, že systémy 3D strojového videnia sú dnes viac než schopné udržať s požiadavkami na priechodnosť výrobnéj linky [2, 3].

Vďaka schopnosti spoľahlivého zachytenia údajov tretej dimenzie sú systémy 3D strojového videnia imúnne voči faktorom prostredia, ktoré nepriaznivo ovplyvňujú systémy 2D - aspekty osvetlenia, kontrastu a vzdialenosti od objektu už nie sú problémom [2, 3].



Obr. 1.4: Princíp 3D systému [3]

Pretože pracujeme s vysoko presným trojrozmerným digitalizovaným modelom cieľového objektu, môžu naše stroje teraz pohodlne zvládať tvar aj polohu. Poznajú presné umiestnenie cieľového objektu v priestore, jeho presný objem, povrchové uhly, stupne rovinnosti a vlastnosti - bez ohľadu na podmienky výrobnéj linky a to, či je objekt lesklý alebo svetlo absorbujúci čierny. To výrazne zjednodušuje upevnenie cieľového objektu a celkový dizajn systému [1, 2].

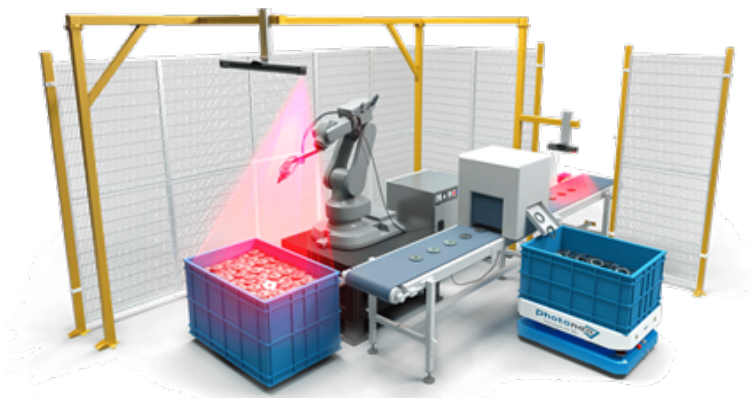
V dôsledku tejto oveľa rozšírenej schopnosti sa 3D strojové videnie aplikuje na široké spektrum úloh, kde 2D schopnosti zaostávajú, okrem iných aj [2, 3]:

- meranie hrúbky, výšky a objemu,
- dimenzovanie a správa priestoru,
- meranie tvarov, otvorov, uhlov a kriviek,
- zisťovanie povrchových alebo montážnych chýb,
- kontrola kvality a overovanie pomocou 3D CAD modelov,
- vedenie robota a sledovanie povrchu (napr. na zváranie, lepenie, odhrotovanie a ďalšie),
- vychystávanie koša na umiestnenie, zabalenie alebo montáž,
- skenovanie a digitalizácia objektov [2, 3].

### 1.2.2 Príklad použitia

Ako jeden z príkladov na pochopenie komplexnosti riešenia si môžeme uviesť jeden príklad, a to koš plný polojasných, kužeľovitých oceľových dielov. Robot musí nielen vybrať jednotlivý diel z náhodnej hromady a zistiť, ako ho najlepšie uchopiť. Musí tiež brať do úvahy veľkosť nástroja, bočné strany koša a susedné časti, aby sa zabránilo kolíziám [3, 4].

Aby to mohol urobiť, musí robot vidieť diely, a to veľmi zreteľne. Musí zistiť, ako všetky ležia, aký majú trojrozmerný prístup, či sú úplne alebo čiastočne nad sebou. A ak sú tmavé alebo svetlo absorbujúce, lesklé a reflexné, musí zvládnuť dynamiku meniacej sa scény, a to odrazy, chýbajúce údaje a vlastný šum. Tento proces musí prebiehať rýchlo a presne [3, 4].



Obr. 1.5: Príklad použitia 3D systémov [10]

Takéto zložité výzvy s vysokou presnosťou v reálnom čase sa však úspešne riešia použitím správnej kombinácie technológií 3D strojového videnia. Poskytuje efektívne a nákladovo efektívne riešenia tam, kde 2D strojové videnie nemôže. Príklad použitia vid' obrázok 1.5 [3, 4].

### 1.2.3 Technológia

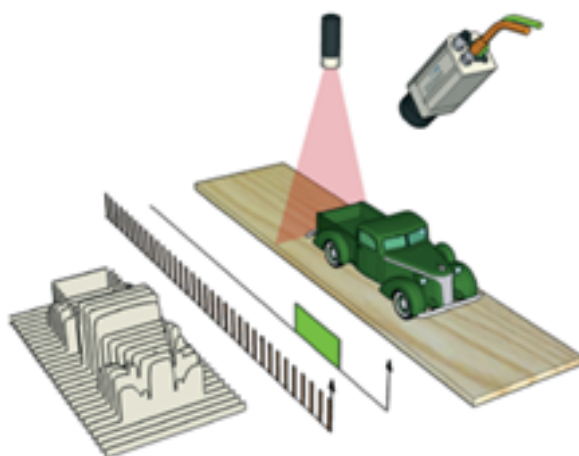
Systémy 3D strojového videnia typicky zahŕňajú viac kamier alebo jeden a viac laserových senzorov posunu. Vo všeobecnosti poznáme dve základné metódy tvorenia obrazu, a to technológia skenovania a technológia „Snapshot“ (zachytenie obrazu) [2, 3].

Pri technológii skenovania je 3D obraz vytvorený skenovaním profilu skenovaného objektu počas jeho pohybu, alebo počas pohybu kamery ponad skenovaný objekt. Pre korektné vytvorenie obrazu musí byť zabezpečená konštantná rýchlosť pohybu, alebo musí byť táto rýchlosť známa použitím enkodérov. 3D obraz vytvorený uvedeným spôsobom je zvyčajne veľmi presný [1, 2].

Pri technológii „Snapshot“ je vytvorenie 3D obrazu realizované fotením objektu rovnakým spôsobom, ako je realizované foteenie bežným osobným fotoaparátom, avšak obraz je vytvorený v 3D. Oproti technológii skenovania tu nie je potrebný pohyb kamery ani objektu. Pre lepšie pochopenie si uvedieme základné typy 3D [1, 2].

- Laser Triangulation (laserová triangulácia)

Tento systém zahŕňa laser a kameru, ktorá zachytáva odraz, na základe ktorého vytvára profil skenovaného objektu. Keďže ide o technológiu skenovania, profil objektu je vytváraný počas pohybu objektu (napríklad po dopravnom páse). Znázornenie tejto technológie vid' obrázok 1.6 [2, 3].



Obr. 1.6: Laser Triangulation [3]

- Time of Flight (TOF)

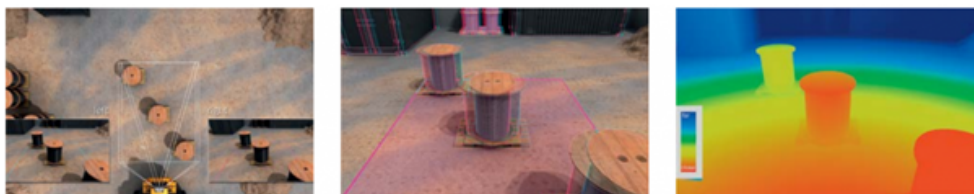
Time of Flight 3D kamery vytvárajú obraz technológiou „Snapshot“. Nie je teda potrebný žiadny pohyb kamery ani objektu. 3D senzory času letu (TOF) merajú, koľko času trvá cesta svetla na scénu a späť k senzorum v poli. Fázový posun vo vyžarovanom svetle oproti prijatému svetlu poskytuje dostatok informácií na výpočet časového rozdielu. Priestorová poloha sa potom vypočíta uplatnením tejto informácie na hodnotu rýchlosti svetla. TOF je vhodný pre použitie predovšetkým v aplikáciách s veľkým zorným poľom a veľkou pracovnou vzdialenosťou. Znázornenie tejto technológie vid' obrázok 1.7 je rozdelené do troch krokov, a to vyslanie svetelného lúča kamerou, meranie času doletu a vytvorenie hĺbkovej mapy [2, 3, 4].



Obr. 1.7: Time of Flight [3]

- Stereo

3D informácie o objekte sa získavajú pozorovaním spoločného znaku z dvoch rôznych hľadísk. Vzdialenosť vypočítaná pre každé hľadisko vráti hodnoty XYZ. Ak sa na jednej časti nachádza viac objektov, je možné vypočítať 3D orientáciu. Na rozdiel od TOF nie je nutné vysielanie svetelného lúča. Akokoľvek, pre vytvorenie 3D obrazu, je nutné vytvorenie rôznych snímok s dostatočnou kvalitou, ktoré sa následne „preložia“. Využitie uvedenej technológie je výhodné predovšetkým pre aplikácie s potrebou veľkého snímaného zorného poľa a taktiež na vonkajšie použitie. Znázornenie tejto technológie vid' obrázok 1.8 je rozdelené do troch krokov, a to vytvorenie dvoch snímok, preloženie obrazu a vytvorenie hĺbkovej mapy [1, 2, 3].



Obr. 1.8: Stereo [3]

### 1.3 Porovnanie 2D a 3D systémov

Hlavný rozdiel 2D a 3D systémov je hlavne v cene a ďalej podľa ceny je rozdiel v rýchlosti spracovania údajov, ktoré potrebujeme. Keďže medzi týmito systémami je veľký cenový



rozdiel, druh strojového videnia vyberáme podľa požiadaviek na konkrétnu aplikáciu [1, 3].

2D strojové videnie sa používa hlavne na riešenie štyroch úloh videnia, vrátane úloh polohovania, kontroly, merania a čítania. Táto technológia je obzvlášť užitočná pre aplikácie s vysokým kontrastom, alebo vtedy, keď je kľúčom k riešeniu textúra alebo farba objektu [1, 3].

3D strojové videnie je možné bez problémov vložiť do akejkoľvek aplikácie priemyselnej automatizácie, ako je polohovanie súčiastok, kontrola, meranie a čítanie. Môžeme získať spoľahlivú kontrolu 3D videnia, aj keď sa farba, poloha a výška súčiastky líšia. Tieto systémy poskytujú 3D laserovú trianguláciu, ktorá môže pomôcť prekonať mnohé výzvy spojené s nízkym kontrastom a slabým osvetlením. 3D videnie je obzvlášť užitočné pre vysoko presné inšpekčné a meracie aplikácie [1, 2, 3].

## 1.4 Výrobcovia

Roboty hrajú v Priemysle 4.0 čoraz dôležitejšiu úlohu. Adaptívne priemyselné roboty a výrobné linky potrebujú víziu podobnú človeku a pomocou ocenenej 3D kamery na extrahovanie detailov mikrometrov a farebných informácií v reálnom čase môžu roboty konečne ako človek sledovať, pohybovať sa a konať v zložitých scénach a cieľových objektoch. Prídanie strojového videnia do robotických systémov zvyšuje flexibilitu a možnosti riešenia problémov v reálnom svete [1, 3].

V dnešnej dobe existuje mnoho výrobcov vision systémov, každá firma má svoje plusy aj mínusy. Pri výbere strojového videnia nás ovplyvňuje veľa faktorov, ako napríklad aká je veľkosť skenovaného objektu, použitý materiál, požiadavky na rýchlosť, kvalita zobrazenia, plnosť farieb a mnoho ďalšieho [1, 2].

### 1.4.1 Cognex Corporation

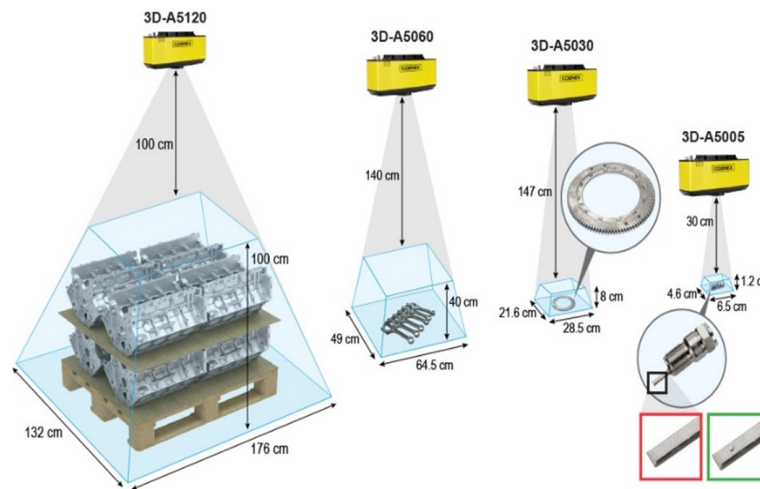
Séria Cognex 3D-A5000 je 3D kamera s plošným skenovaním určená na snímanie 3D obrázkov s vysokým rozlíšením. Je vybavený technológiou 3D LightBurst, ktorá rýchlo získava obrázky a maximalizuje tak priepustnosť. 3D obrázky vo vysokom rozlíšení v kombinácii so špičkovými 3D vizuálnymi nástrojmi Cognex umožňujú spoľahlivé a presné riešenia aplikácií, ako je overenie zostavy, priame meranie a robotické vedenie [5, 6].

Model 3D-A5000 ponúka rôzne zorné polia a rozsahy meraní pre použitie v automobilovom, spotrebnom a logistickom priemysle. Medzi ďalšie funkcie a výhody patrí [5, 6]:

- patentovaná technológia 3D LightBurst umožňuje rýchle získanie obrazu,
- viac ako 1,5 milióna 3D dátových bodov detekuje jemné prvky na súčiastkach,
- režim vysokého dynamického rozsahu (HDR) sníma scény s vysokým kontrastom,

- výkonné 3D nástroje vo VisionPro riešia náročné 3D aplikácie,
- továrenská kalibrácia poskytuje výsledky v skutočných jednotkách,
- robustné puzdro s krytím IP65 poskytuje ochranu v drsnom prostredí továrne [5, 6].

Model 3D-A5000 má robustné puzdro z hliníkového tlakového odliatku s krytím IP65, ktoré je vyrobené tak, aby vydržalo na podlahe továrne. Priemyselný dizajn zaisťuje konzistentné výsledky merania aj pri rôznych teplotách [5, 6].



Obr. 1.9: Vision systém Cognex [10]

#### 1.4.2 SICK AG

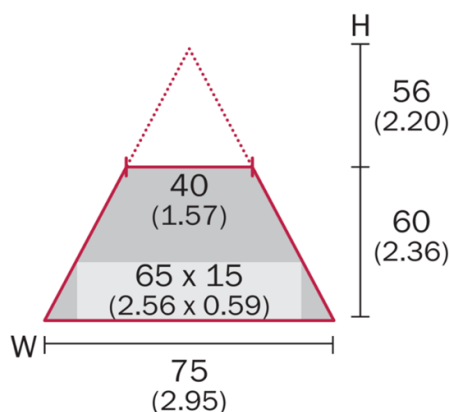
TriSpectorP1000 je programovateľná 3D kamera pripravená pre priemysel 4.0. Súčasťou ekosystému SICK AppSpace sú 3D kamera a softvérové nástroje, ktoré sú súčasťou balenia a otvárajú možnosti riešení šitých na mieru - s ľahkým uvedením do prevádzky a obsluhou v oblastiach, ako je kontrola kvality, manipulácia s robotom a overovanie profilov. TriSpectorP1000 pracuje samostatne a obsahuje zobrazovanie, osvetlenie a analýzu v jednom robustnom kryte. Technológia laserovej triangulácie poskytuje údaje nezávislé od farieb, kontrastu a skutočného tvaru objektu v milimetroch. TriSpectorP1000 je perfektnou voľbou pre plne flexibilné, nákladovo efektívne a in-line priemyselné automatizačné riešenia [7].

- plne flexibilná automatizácia vďaka údajom skutočného tvaru v mm, ktoré sú nezávislé od kontrastu,
- nákladovo efektívne riešenia so samostatnou 3D kamerou pripravené na priemysel 4.0,
- zvýšená kvalita a menej odpadu s in-line kontrolami všetkých častí v troch rozmeroch,



Obr. 1.10: Vision systém SICK [7]

- ľahká obsluha s prispôsobeným webovým rozhraním,
- jednoduché uvedenie do prevádzky a výmena zariadenia so zaručeným zorným poľom,
- prevádzka v drsnom prostredí s krytím IP67 [7].



Obr. 1.11: Zorné pole v mm (inch) TriSpectorP1000 [7]

### 1.4.3 Photoneo s.r.o.

3D skener PhoXi bol ocenený programom Vision Systems Design Innovators Awards Program za bezkonkurenčný výkon a najlepší pomer rozlíšenie/rýchlosť vo svojej triede. Každý skener z rodiny PhoXi (modely XS-XL) je vhodný pre rôzne aplikácie, v závislosti od veľkosti skenovaného objektu, objemu skenovania a ďalších aspektov [8].

Objem skenovania rodiny 3D skenerov PhoXi sa pohybuje od 15 cm do 4 metrov, čo nám umožní skenovať veľké aj veľmi malé objekty [8].

Vďaka nízkej hmotnosti menšej ako 1,5 kg je možné 3D skener PhoXi namontovať priamo na robotické rameno, čo môže byť veľmi praktické, ak potrebujeme skenovať veľký



Obr. 1.12: Vision systém Photoneo [8]

kôš pomocou menšieho skenera z rôznych uhlov pohľadu, alebo získať podrobný pohľad v rohoch koša [8].

Vďaka triede ochrany IP65 je 3D skener PhoXi prachotesný a chránený proti nízko-tlakovým prúdom vody prichádzajúcim z ľubovoľného smeru a z každého uhla. 3D skener PhoXi ponúka najlepšiu tepelnú stabilitu vo svojej triede vďaka hardvérovým vylepšeniam spolu s modelom tepelnej kalibrácie [8].

## 2 Návrh aplikácie a implementácie 3D skenera Photoneo PhoXi do robotizovaného pracoviska

Návrh a implementáciu robotizovaného pracoviska s využitím 3D strojového videnia pomocou skenera PhoXi som opísal v tejto kapitole v jednotlivých bodoch. V prvom bode sme potrebovali zdefinovať zariadenia, ktoré budú využívané na pracovisku. Po návrhu zariadení sme zrealizovali návrh aplikácie a následne sme vyriešili komunikáciu zariadení, navrhli sme koncový efektor manipulátora a logicky sme rozmiestnili zariadenia na robotizovanom pracovisku.

### 2.1 Popis zariadení na pracovisku

Na našom pracovisku používame zariadenia, ktoré sa nachádzajú v laboratóriu pokročilej robotiky na ústave automatizácie a merania strojníckej fakulty v Bratislave. Aby sme dokázali splniť zadanie diplomovej práce a vytvoriť pracovisko s využitím 3D strojového videnia, potrebujeme 3D snímacie zariadenie, ktoré je v našom prípade Photoneo 3D skener PhoXi L. Skener je pripojený k riadiacemu systému Vision, ktorý je prepojovacím uzlom medzi ostatnými zariadeniami, a to užívateľskými zariadeniami ako obrazovka, klávesnica a myš. Vision riadiaci systém je ďalej prepojený s riadiacim systémom manipulátora KUKA KR C4 Compact a ten je prepojený so samotným manipulátorom KUKA KR3R540. Detailnejší popis jednotlivých zariadení je v nasledujúcich podkapitolách.

#### 2.1.1 Photoneo 3D skener PhoXi

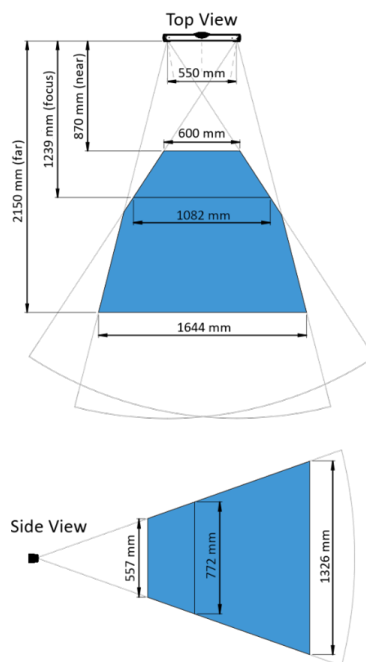
3D skener PhoXi od spoločnosti Photoneo je skenerom, ktorý vyhľadáva statické scény s vysokým rozlíšením a vysokou presnosťou. Jeho prístup štruktúrovanej svetelnej projekcie poskytuje výstup vo forme presných mračien bodov pre rýchlu lokalizáciu ktorejkoľvek požadovanej časti. Robustnosť skenera zaisťuje vynikajúcu kvalitu skenov, čo z neho robí najlepší skener aj v tých najnáročnejších prostrediach [8].



Obr. 2.1: 3D skener PhoXi L [10]

### Využitie, typy aplikácií

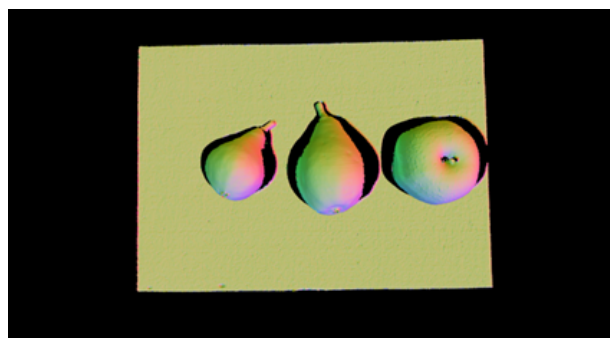
Skener s veľkým dosahom, obzvlášť užitočný na skenovanie predmetov a košov s veľkou europalety. Svoje uplatnenie našiel v projektoch robotiky s riadeným videním, najmä v automobilovom priemysle. Vďaka svojej univerzálnosti sa však používal v rôznych priemyselných odvetviach, od lekárskeho, logistického, poľnohospodárskeho alebo ťažkého priemyslu [10].



Obr. 2.2: Rozsah skenovania PhoXi L [10]

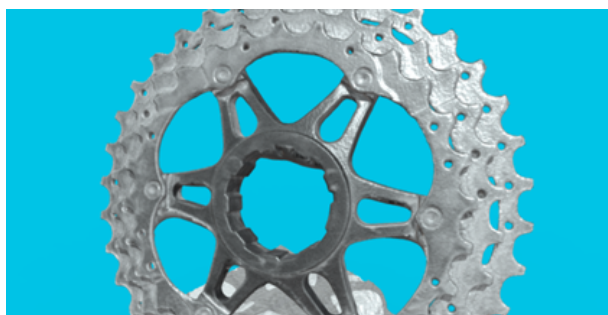
PhoXi 3D Scanner je možné použiť v potravinárskom priemysle na skenovanie „živých“ predmetov, ako je ovocie, zelenina alebo ryby vid' obrázok 2.3. Jeho vynikajúci výkon umožňuje vyberať objekty s deformovateľnými povrchmi, rôznymi tvarmi a rôznymi veľkosťami [10].

Vďaka pokročilým algoritmom filtrovania šumu nemá PhoXi 3D Scanner žiadne ťaž-



Obr. 2.3: Príklad použitia v potravinárskom priemysle [10]

kosti s lesklými alebo odrážajúcimi materiálmi. Môže byť použitý na skenovanie a kontrolu kovových predmetov, napríklad ozubených kolies vid' obrázok 2.4. Dokáže naskenovať aj ľudské telo alebo jeho časti. Medzi jeho široké spektrum aplikácií patrí lekársky priemysel [10].



Obr. 2.4: Príklad použitia kontroly lesklých povrchov [10]

### **Technológia použitia, parametre**

3D skener Photoneo PhoXi je zariadenie, ktoré využíva štruktúrovanú svetelnú projekciu na rekonštrukciu geometrie 3D povrchu kontrolovaného objektu. Počas skenovania sa na scénu premieta jeden alebo viac štruktúrovaných vzorov, ktorý umožňuje výpočet hĺbky zo snímok scény. Geometria povrchu je užívateľovi poskytnutá v podobe hĺbkovej mapy alebo vo forme mračna bodov [10].

Proces skenovania pozostáva z troch fáz: snímania (alebo akvizície), spracovania (výpočtu) a prevodu. V prvej fáze premieta 3D skener Photoneo PhoXi na scénu sériu svetelných vzorov. Po získaní údajov skener vypočíta všetky viditeľné body na povrchu a tieto údaje odošle prostredníctvom ethernetového pripojenia. Komunikácia so skenerom by mala byť minimálne 1 Gb/s, aby sa zabránilo prekážkam počas procesu prenosu údajov [10].

Predmety vhodné na skenovanie 3D skenerom PhoXi sa používajú na získavanie 3D údajov štruktúrovaného vzoru svetla. Scéna musí byť počas skenovania úplne nehybná,

bez dymu a častíc rozptýlených vo vzduchu. Parametre skenera môžeme vidieť v tabuľke 2.1 [10].

Parameter	Hodnota
Rozlíšenie (3D body)	Až 3,2 milióna
Rozsah skenovania	870 - 2150 mm
Optimálna skenovacia vzdialenosť	1239 mm
Skenovacia oblasť na sladkom mieste	1082 x 772 mm
Vzdialenosť medzi bodmi	0,524 mm
Čas skenovania	250 - 2750 ms
Rozmery	77 x 68 x 616 mm
Východisková hodnota	550 mm
Váha	1100 g
Priepustnosť 3D bodov	16 miliónov bodov za sekundu

Tabuľka 2.1: Parametre PhoXi L [10]

### 2.1.2 Riadiaci systém Vision

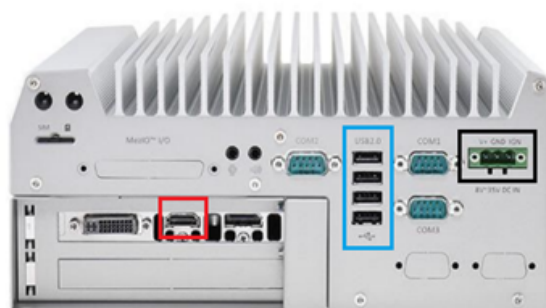
Riadiaci systém Vision pozostáva z priemyselného počítača Nuvo-5095GC s výkonnou výpočtovou platformou GPU. Vďaka patentovanej technológii kazety, vid' obrázok 2.5, a dizajnu pre lepší rozptyl tepla dokáže efektívne odvádzať teplo generované GPU a zabezpečiť pracovnú teplotu 60°C pri 100% načítavaní. Nuvo-5095GC je založený na platforme Intel Skylake a podporuje 35W/65W 6. generácie procesorov Core a až 32 GB pamäte DDR4. Na tomto priemyselnom počítači je nainštalovaný bezplatný otvorený počítačový operačný systém Linux. Pre rýchlo rastúce počítačové aplikácie GPU predstavuje Nuvo-5095GC prvú priemyselnú, kompaktnú a odolnú platformu obsahujúcu CPU a GPU s výkonom ďaleko presahujúcim tradičné priemyselné počítače. Všetky tieto vlastnosti sú integrované do veľmi kompaktného pôdorysu 240 x 225 x 110 mm [13].



Obr. 2.5: Technológia kazety riadiaceho systému Vision [13]



Riadiaci systém Vision vytvára rozhranie pre ovládanie všetkých zariadení spomenutých v kapitole 2.1. Jeho hlavnou úlohou je beh softvérového rozhrania Bin Picking Studio, nevyhnutného na používanie skenera, a je taktiež hlavným uzlom medzi všetkými zariadeniami. Ako môžeme vidieť na obrázku 2.6 zadnej strany riadiaceho systému, slúži na pripojenie operátorskej obrazovky cez HDMI port, vid' červený obdĺžnik, pripojenie myši, klávesnice a USB zariadení na nahrávanie modelov cez USB port, vid' modrý obdĺžnik, a v neposlednom rade portu napájacieho kábla tohto riadiaceho systému, vid' čierny obdĺžnik [13].



Obr. 2.6: Zadná strana Vision riadiaceho systému [13]

Predná strana Vision riadiaceho systému, vid' obrázok 2.7, slúži na pripojenie zariadení potrebných na konfiguráciu výrobných nastavení a prídavných modulov pre náročnejšie požiadavky výpočtového výkonu, vid' červené obdĺžniky. Pripojenie samotného skenera je zabezpečené cez ethernetový kábel, vid' modrý obdĺžnik, ethernetový port v oranžovom obdĺžniku slúži na pripojenie do internetovej siete a ethernetový vstup v zelenom obdĺžniku slúži na pripojenie riadiaceho systému manipulátora [13].



Obr. 2.7: Predná strana Vision riadiaceho systému [13]

### 2.1.3 Robotický systém KUKA KR 3 R540

Robotický systém KUKA KR 3 R540 pozostáva z troch samostatných zariadení na umožnenie jeho prevádzky, a to riadiaceho systému KUKA KR C4 Compact, manipulátora KUKA KR KR3R540 a SmartPADu, viď obrázok 2.8. Tieto zariadenia sú medzi sebou prepojené, navzájom komunikujú a vytvárajú tak užívateľské rozhranie. Manipulátor je pripojený k riadiacemu systému pomocou dvoch pripojovacích káblov, a to dátového káblu a káblu na ovládanie motorov. SmartPad je pripojený k riadiacemu systému pomocou komunikačného káblu. Riadiaci systém je napájaný priamo z 230V siete. Detailnejší popis a funkcionality robotického systému KUKA KR 3 R540 je popísaný v nasledujúcich podkapitolách [17].



Obr. 2.8: Robotický systém KUKA KR 3 R540 [14]

#### Manipulátor KUKA KR 3 R540

Manipulátor KUKA KR KR3R540, viď obrázok 2.9, je 6-osový priemyselný robot vhodný na použitie najmä vtedy, keď sa musia malé komponenty a výrobky vyrábať na minimálnom priestore [17].

Nadväzujúc na Vachálka (2014, s. 28), pojem priemyselný robot môžeme chápať ako autonómne fungujúci stroj—automat, ktorý je určený k reprodukcii niektorých pohybových a imitácii duševných funkcií človeka pri vykonávaní pomocných a základných výrobných operácií bez bezprostrednej účasti človeka a ktorý je k tomuto účelu vybavený niektorými jeho schopnosťami (sluchom, zrakom, hmatom, pamäťou a podobne), schopnosťou samovýuky, samoorganizácie a adaptácie, t.j. schopnosť prispôbeniu sa danému prostrediu [18].

Manipulátor KUKA KR KR3R540 je možné použiť všade tam, kde sú potrebné krátke výrobné cykly s najvyššou úrovňou výkonu. Vďaka svojim kompaktným rozmerom ponúka maximálnu produktivitu na meter štvorcový [17].



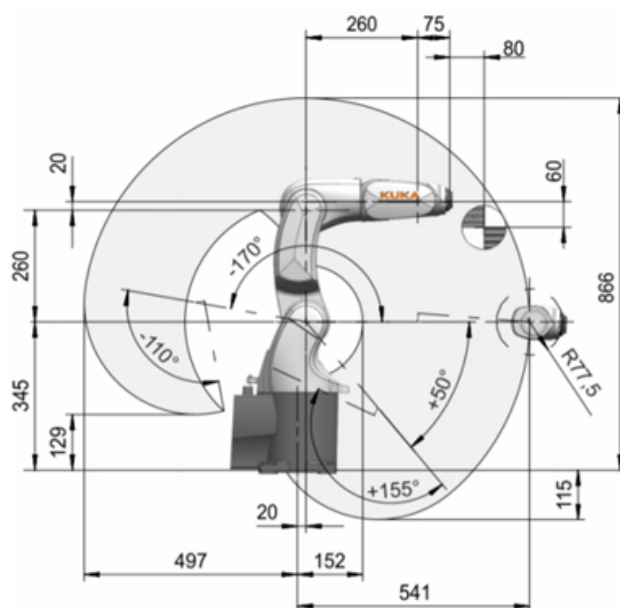
Obr. 2.9: Manipulátor KUKA KR 3 R540

Je to priemyselný robot pre vysoko presnú prevádzku bez ohľadu na to, aká je poloha, časový rámec a montážna poloha. KR3R540 pripravuje pôdu pre flexibilnú a efektívnu automatizáciu na všetky druhy koncepcií buniek [17].

Manipulátor je navrhnutý pre menovité užitočné zaťaženie 2 kg, aby sa optimalizoval dynamický výkon manipulátora. Pri zmenených vzdialenostiach stredu nákladu sa môžu použiť vyššie zaťaženia až do maximálneho pracovného zaťaženia, ktoré je 3 kg [17, 19]. Základné technické parametre manipulátora sú zobrazené na obrázku 2.10 a v tabuľkách 2.2 a 2.3.

Parameter	Hodnota
Maximálny dosah	541 mm
Maximálna nosnosť	3 kg
Presnosť opakovania polohy	+ - 0,02 mm mm
Počet osí	6
Montážna poloha	Podlaha, strop, stena
Inštaláčna plocha	179 mm x 179 mm
Hmotnosť	26,5 kg

Tabuľka 2.2: Technické údaje manipulátora KUKA KR 3 R540 [19]



Obr. 2.10: Pracovný priestor manipulátora v mm [19]

Os	Rozsah v °
A1	+ -170 °
A2	-170 ° / 50 °
A3	-110 ° / 155 °
A4	+ -175 °
A5	+ -120 °
A6	+ -350 °

Tabuľka 2.3: Parametre osí manipulátora KUKA KR 3 R540 [19]

### Riadiaci systém KUKA KR C4 Compact

Riadiaci systém KUKA KR C4 Compact zobrazený na obrázku 2.11 je flexibilný riadiaci systém, ktorý zabezpečuje nízke náklady pri integrácii, údržbe a servise. Softvérová architektúra integruje ovládače Robot Control, PLC Control, Motion Control a Safety Control [17].

Vďaka ovládačom Robot Control a Motion Control sú bezproblémovo a interaktívne integrované do riadiacich procesov pre PLC, CNC a Safety. Vďaka jednoduchému a flexibilnému programovaniu manipulátora prostredníctvom vložených formulárov a novému programovaniu pohybu Spline majú automatizačné riešenia založené na KR C4 inteligentný, flexibilný a škálovateľný aplikačný potenciál [17].

Ovládač PLC Control umožňuje plný prístup k celému vstupno-výstupnému systému riadiaceho systému a má vysoký výkon za behu. Umožňuje vstupno-výstupnú manipuláciu s manipulátorom, kompletnou robotickou bunkou alebo radom manipulátorov. Ďalej je

možné pomocou funkčných blokov čítať a spracovávať premenné, ako sú polohy osí alebo rýchlosti. Vďaka zvyšujúcemu sa počtu funkcií špecifických pre manipulátor v predchádzajúcich systémoch CAD / CAM môže manipulátor hrať priamu úlohu v procesoch obrábania [17].

Medzi hlavné komponenty, z ktorých sa skladá Riadiaci systém KUKA KR C4 Compact, patria ventilátor, pevný disk, nízkonapäťový napájací zdroj, pamäťová karta, riadiaca jednotka, porty na pripojenie periférií, hlavný vypínač, základná doska a batérie [17].

Bezpečnostné funkcie a komunikácia zameraná na bezpečnosť sa realizujú prostredníctvom protokolov založených na protokole Ethernet. Bezpečnostná koncepcia využíva viacjadrovú technológiu a umožňuje dvojkanálovú prevádzku potrebnú pre bezpečnostné aplikácie [17].



Obr. 2.11: Riadiaci systém KUKA KR C4 Compact

### **KUKA smartPAD**

SmartPAD je ovládač na ovládanie priemyselného robota. Má všetky ovládacie a zobrazovacie funkcie potrebné na obsluhu a programovanie. Obsahuje taktiež dotykovú obrazovku smartHMI na jednoduchú manipuláciu. Predná a zadná strana smartPAD-u sú zobrazené na obrázkoch 2.12 a 2.13 [15].

Popis obrázku 2.12 :

1. Tlačidlo na odpojenie KUKA smartPADu,
2. klávesový prepínač používateľa,
3. tlačidlo núdzového zastavenia,
4. 6D polohová myš pre ručné ovládanie robota,



Obr. 2.12: Predná strana KUKA SmartPad-u [15]

- 5. tlačidlá na ovládanie jednotlivých kĺbov,
- 6. tlačidlo na určenie rýchlosti robota pri behu programu,
- 7. tlačidlo na určenie rýchlosti pohybu pri manuálnom ovládaní,
- 8. tlačidlo návratu do hlavného menu,
- 9. tlačidlá pre zobrazenie statusu nastavených technologických parametrov,
- 10. tlačidlo štart,
- 11. tlačidlo späť,
- 12. tlačidlo stop,
- 13. tlačidlo zobrazenia klávesnice. [15]

Popis obrázku 2.13 :

- 1. Tlačidlo povolenia pohybov,
- 2. tlačidlo štart,
- 3. tlačidlo povolenia pohybov,
- 4. USB port,
- 5. identifikačný štítok. [15]



Obr. 2.13: Zadná strana KUKA SmartPad-u [15]

## 2.2 Popis navrhnutej úlohy

Úlohou mojej diplomovej práce bolo navrhnúť a spojzdníť robotizované pracovisko v laboratóriu pokročilej robotiky na ústave automatizácie a merania strojníckej fakulty STU v Bratislave za pomoci manipulátora KUKA KR3R540 umiestneného na stole, ktorý má odoberať a vkladať ozubené kolesá do bloku motora s využitím počítačového videnia pomocou 3D skenera PhoXi. Táto úloha bola podmienená tým, že na stole sa nachádzajú rôzne priestorovo orientované súčiastky a robot má za pomoci skenera nájsť, uchopiť a uložiť len konkrétne súčiastky, v našom prípade 15-zubové ozubené kolesá. Blok motora, do ktorého sa ukladajú ozubené kolesá, je uložený na lineárnom posuve. Zariadenia a súčasti navrhnutej manipulačnej úlohy sú zobrazené na obrázku 2.14.

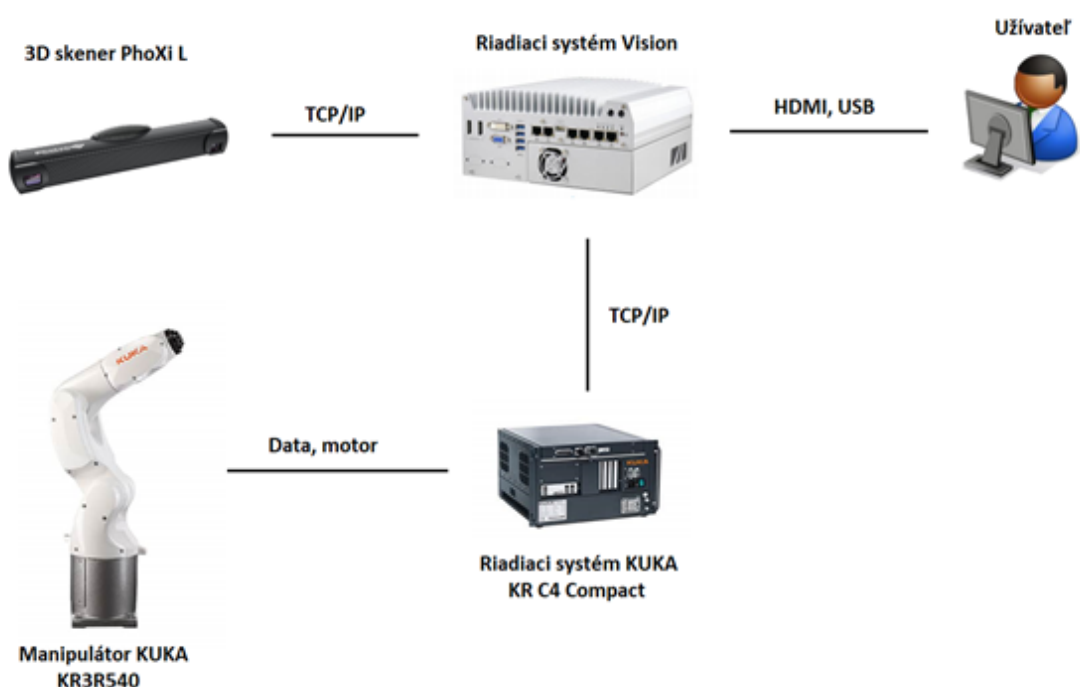


Obr. 2.14: Zariadenia a súčasti navrhnutej úlohy [15, 21]



## 2.3 Topológia použitých zariadení

Zariadenia použité na pracovisku musia medzi sebou komunikovať, preto bolo potrebné navrhnuť a spojazdniť komunikáciu. Topológia použitých zariadení je zobrazená na obrázku 2.15. Všetky pripojenia medzi 3D skenerom, riadiacim systémom Vision a riadiacim systémom manipulátora sú realizované pomocou Ethernet TCP/IP komunikácie, teda sú prepojené ethernetovými káblami. Užívateľ je prepojený so zariadeniami pomocou obrazovky spojenej s riadiacim systémom Vision cez HDMI kábel a klávesnicou s myšou, ktoré sú spojené s riadiacim systémom Vision pomocou USB káblov [21].



Obr. 2.15: Topológia použitých zariadení [14, 15, 21]

## 2.4 Návrh koncového efektora manipulátora

Na základe návrhu manipulačnej úlohy vysvetlenej v kapitole 2.2. bolo potrebné navrhnuť a skonštruovať koncový efektor manipulátora na uchopenie súčiastok. Na tento návrh sme použili pneumatický uchopovač s lineárnym vedením MHF2-12D, hliníkový profil, 3D tlačené komponenty a spojovací materiál. 3D tlačené komponenty sme navrhli konkrétne na uchopenie ozubených kolies s vnútorným priemerom 30 mm. Koncový efektor sa skladá z viacerých 3D tlačných komponentov, a to príruby, medzikusu uchytenia profilu a uchopovača a prstov uchopovača, viď obrázok 2.16. Prsty uchopovača sú navrhnuté tak, aby uchopovali súčiastky pomocou vnútorného priemeru, takže vedú uchopovať akékoľvek ozubenie s vnútorným priemerom o veľkosti 25 až 35 mm [16].





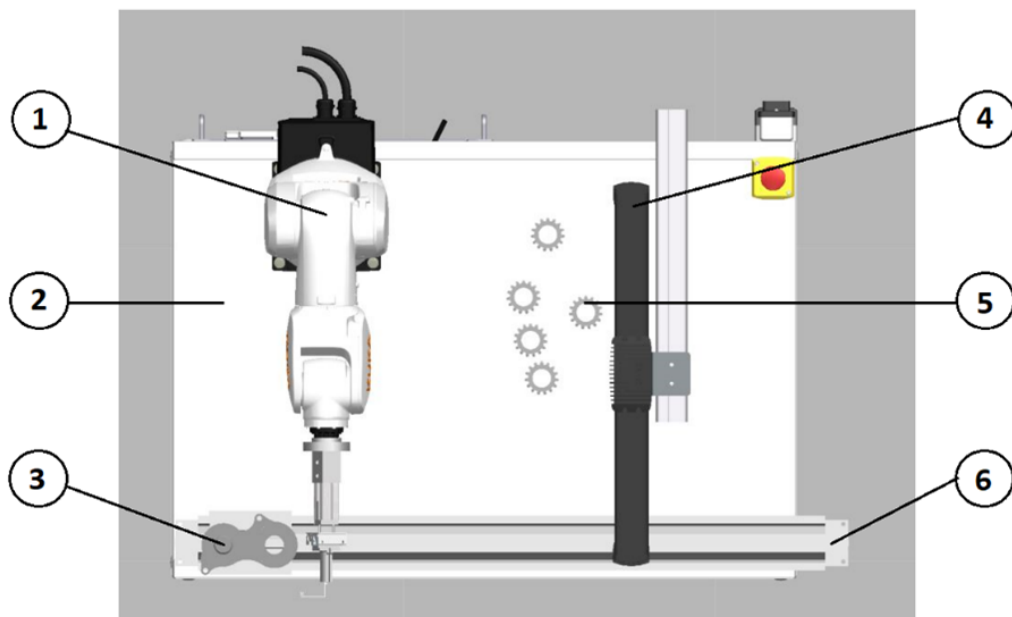
Obr. 2.16: 3D model koncového efektoru manipulátora

## 2.5 Rozmiestnenie zariadení na robotizovanom pracovisku

Zariadenia použité v našej aplikácii sme umiestnili na stôl, na ktorom bol upnutý manipulátor KUKA KR3R540. 3D skener sme umiestnili nad pracovisko tak, aby sme vedeli zoskenovať všetky potrebné komponenty v ideálnej vzdialenosti. Vzdialenosť skenera od stola bola navrhnutá podľa odporúčaní od výrobcu, uvedené v kapitole 2.1.1. Odoberané súčiastky, v našom prípade ozubené kolesá, sme umiestnili priamo pod skener aj vo vzdialenostiach, ktoré manipulátor nie je schopný dosiahnuť, aby sme vedeli následne overiť funkčnosť skenera. Lineárny posuv sme umiestnili tak, aby mohol manipulátor uchopenú súčiastku bez kolízií umiestniť do bloku motora, ktorý je umiestnený na posuvnej časti lineárneho posuvu. Rozmiestnené zariadenia a súčasti využívané v manipulačnej úlohe sú zobrazené na obrázku 2.17 a jeho popis sa nachádza hneď pod ním.

Popis obrázku 2.17 :

1. Manipulátor KUKA KR3R540,
2. stôl,
3. blok motora,
4. 3D skener PhoXi L,
5. ozubené kolesá,
6. lineárny posuv.



Obr. 2.17: Rozmiestnenie zariadení na robotizovanom pracovisku

## 3 Popis realizácie robotizovaného pracoviska a programovej úlohy

V tejto kapitole si priblížime realizáciu robotizovaného pracoviska a programovej úlohy. V prvej podkapitole sa budeme venovať nastaveniu robotického systému KUKA KR3R540 a jeho komunikácii s 3D skenerom a softvérovým prostredím Bin Picking Studio od spoločnosti Photoneo. V ďalšej kapitole si vysvetlíme používanie a konfiguráciu tohto softvérového prostredia aj s jeho následným príkladom využitia v našej aplikácii. V poslednej podkapitole si zhrnieme realizáciu a spustenie našej aplikácie s následným grafickým zobrazovaním a popisom programovej úlohy.

### 3.1 Nastavenie robotického systému na Photoneo Bin Picking Studio

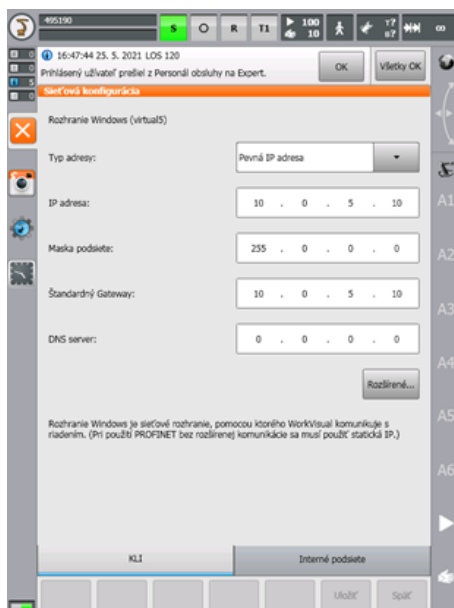
Predpoklady na použitie KUKA ovládačov a modulov od spoločnosti Photoneo sú, aby riadiaci systém manipulátora bol vo verzii KRC v.8.3. alebo vyšší, čo sme v našom prípade museli riešiť updatom softvéru riadiaceho systému manipulátora, keďže verzia nebola dostatočná. Ďalším predpokladom je nainštalované komunikačné rozhranie Ethernet KRL Interface v.2.2.8 a vyššia. Táto verzia bola v radiacom systéme manipulátora už predinštalovaná, a tak sme mohli začať s inštalovaním ovládačov a modulov do riadiaceho systému manipulátora, tento postup je vysvetlený na nasledujúcich stranách tejto podkapitoly [21, 22].

Jednotlivé obrázky v tejto podkapitole sme vytvorili snímkami obrazovky smartHMI na smartPAD ovládači pomocou nami nainštalovaného programu Screenshot od spoločnosti OrangeApps. Tento program sme nainštalovali ako prídavný podprogram do riadiaceho systému manipulátora a jeho ikonu môžeme vidieť na ľavej strane obrazovky smartHMI.

#### 3.1.1 Nastavenie ovládača KUKA a konfigurácia siete

Ako prvý krok sme museli nakonfigurovať sieť riadiaceho systému manipulátora tak, aby sa spolu s Vision riadiacim systémom vedeli spojiť v spoločnom sieťovom rozhraní. Modul Photoneo KUKA využíva komunikáciu TCP/IP na prenos údajov medzi riadiacim sys-

témom manipulátora a Bin Picking Studia. Ako prvý krok pri uvedení do prevádzky sa musíme uistiť, že adresa IP riadiaceho systému manipulátora vyhovuje našim sieťovým konfiguračným požiadavkám. Prepne sa teda do expertného režimu, otvoríme obrazovku hlavného menu a otvoríme priečinkovú prvotnú spustenie, následne priečinkovú sieťovej konfigurácie a nastavíme sieťové rozhranie tak, aby vyhovovala našim požiadavkám, viď obrázok 3.1 [22].



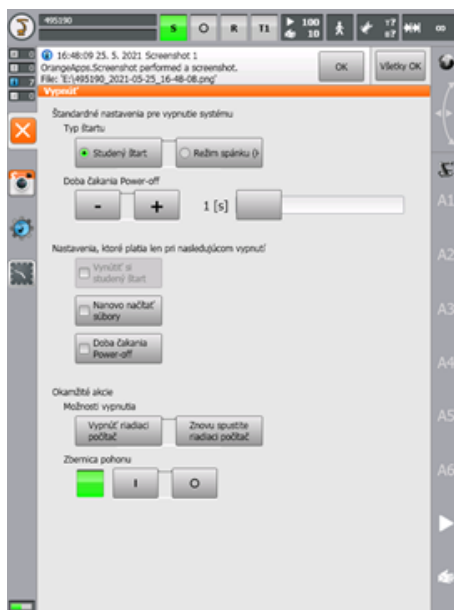
Obr. 3.1: Sieťová konfigurácia riadiaceho systému manipulátora

Aby bolo možné použiť novú konfiguráciu siete, musíme riadiaci systém manipulátora reštartovať, otvoríme obrazovku hlavného menu, zvolíme priečinkovú s názvom Vypnúť a následne v možnostiach vypnutia zvolíme znovu spustenie riadiaceho počítača, čím sa systém reštartuje, viď obrázok 3.2.

V priečinku vypnúť sa ďalej nachádzajú štandardné nastavenia pre vypnutie systému s voľbou studeného štartu alebo režimu spánku. Keďže v našom prípade je robotický systém málo vyťažovaný, pretože slúži len na výuku, máme systém nastavený na studený štart, režim spánku sa využíva hlavne ak je robot v takmer nepretržitej prevádzke, teda je neefektívne ho neustále vypínať a zapínať studeným štartom. V tomto priečinku sa dá ďalej nastaviť doba čakania vypnutia systému, toto nastavenie času sa využíva hlavne vtedy, ak robotický systém napája alebo ovláda ďalšie zariadenia, ktoré potrebujú čas na dosiahnutie koncových polôh pri ich vypnutí.

### 3.1.2 Nastavenie modulu konfigurácie Photoneo KUKA

Pred začatím konfigurácie modulov musíme nahráť Photoneo KUKA moduly do riadiaceho systému manipulátora pomocou USB zariadenia, ktoré zapojíme do riadiaceho systému manipulátora. Tieto moduly musíme vložiť do separátnych priečinkov, do priečinku



Obr. 3.2: Reštart riadiaceho systému manipulátora po nastavení sieťovej konfigurácie

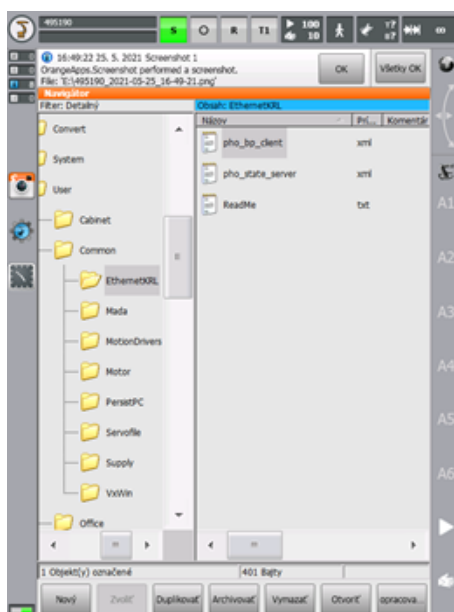
EthernetKRL Config vložíme súbory `pho_bp_client.xml` a `pho_state_server.xml`. Do priečinku Photoneo vložíme súbory `example_programs`, `customer_definitions.src`, `pho_common.src`, `pho_common.dat`, `pho_motion.src`, `pho_motion.dat`, `pho_state_server.src`. V priečinku `example_programs` sa nachádzajú tri programové príklady použitia skenera, a to `main_application`, `multi_vision_systems` a `change_solution` [22].

## Konfigurácia Ethernet KRL

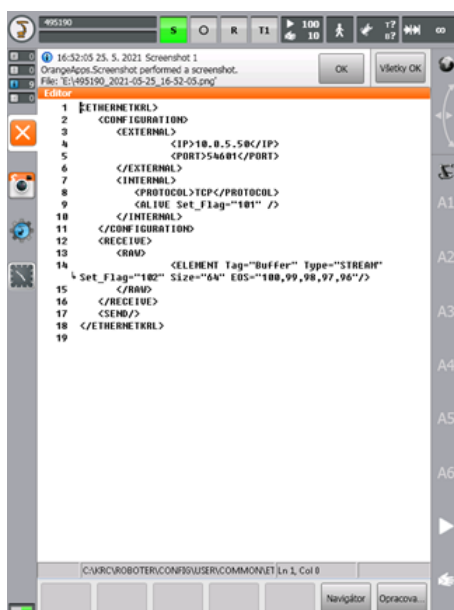
Priečinok EthernetKRL Config obsahuje dva súbory XML - `pho_bp_client.xml` a `pho_state_server.xml`. Tieto súbory je potrebné nahráť prostredníctvom USB kľúča do riadiaceho systému manipulátora, do priečinka C: System KRC Roboter Config User Common EthernetKRL tak, ako sú zobrazené na obrázku 3.3. Je potrebné dodržať toto konkrétne umiestnenie, pretože je kľúčové pre správnu konfiguráciu a následnú funkčnosť modulu EthernetKRL Config [22].

Súbory s príponou xml obsahujú konfiguráciu komunikačného rozhrania. Jediné položky, ktoré musia byť zmenené, sú značky IP adres. Adresu IP Vision riadiaceho systému zadáme na externú značku IP do súboru `pho_bp_client.xml`, ako je uvedené na obrázku 3.4. Túto IP adresu musíme zadať podľa toho, aké máme sieťové rozhranie a akú IP adresu máme nastavenú na riadiacom systéme Vision. V tejto konfigurácii môžeme vidieť, aký port je prednastavený na túto nami zadanú IP adresu. Vidíme, že táto konfigurácia je rozdelená na externú, teda riadiaci systém Vision, a internú, teda riadiaci systém manipulátora, kde sa pri správnej konfigurácii aktivuje značka s číslom 101 [22].

IP adresu riadiacej jednotky manipulátora zadáme do internej značky IP v súbore `pho_state_server.xml`, ako je uvedené na obrázku 3.5. IP adresy musíme zadať podľa



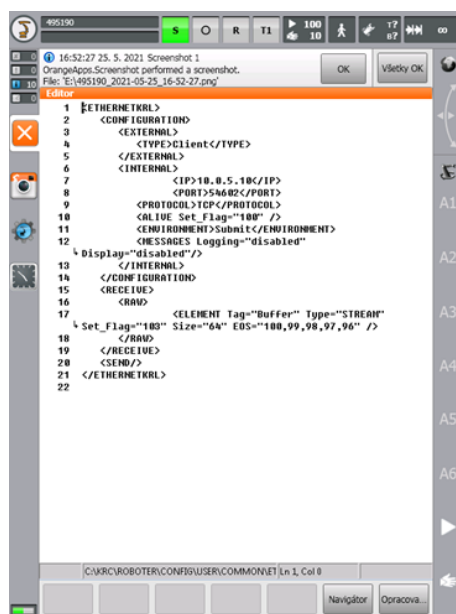
Obr. 3.3: Nahranie priečinku EthernetKRL Config



Obr. 3.4: Nastavenie IP adresy Vision riadiaceho systému

toho, aké máme sieťové rozhranie a akú IP adresu máme nastavenú v riadiacom systéme manipulátora. V tejto konfigurácii môžeme vidieť taktiež to, aký port je prednastavený na túto nami zadanú IP adresu. Vidíme, že táto konfigurácia je rozdelená na externú, teda užívateľskú, a internú, teda riadiaci systém manipulátora, kde sa pri správnej konfigurácii aktivuje značka s číslom 100 [22].

Po tomto nastavení je opäť nutné reštartovať riadiaci systém manipulátora rovnako ako pri sieťovej konfigurácii riadiaceho systému, teda otvoríme obrazovku hlavného menu, zvolíme priečinkok s názvom Vypnúť a následne v možnostiach vypnutia zvolíme znovu



Obr. 3.5: Nastavenie IP adresy riadiaceho systému manipulátora

spustenie riadiaceho počítača, čím sa systém reštartuje.

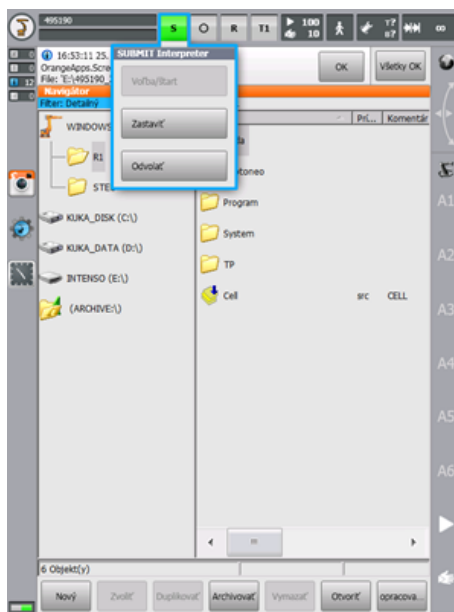
## Konfigurácia serveru

Aby bol server funkčný, je potrebné upraviť Submit Interpreter jeho odvolaním. Pred vykonaním akýchkoľvek zmien sa prepne do expertného režimu a odvoláme Submit Interpreter, viď obrázok 3.6. Do expertného režimu sa prihlásime kliknutím na obrazovku hlavného menu a otvoríme priečinok prihlásenie užívateľa, následne klikneme na užívateľský režim expert a zadáme heslo na prihlásenie. V tomto režime nám je umožnené konfigurovať všetky funkcie v riadiacom systéme manipulátora [22].

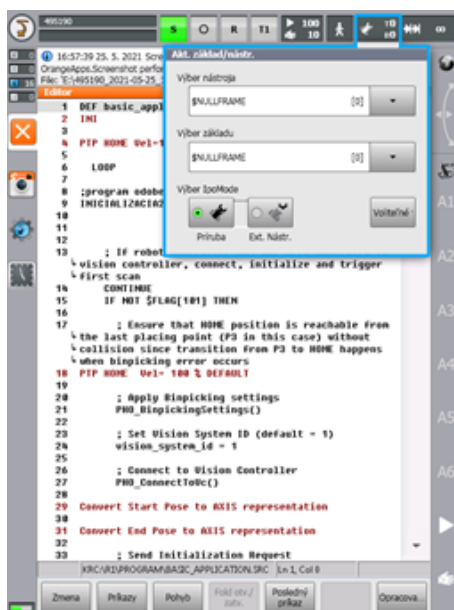
State Server je potrebný na účely kalibrácie a vizualizácie. Hneď ako reštartujeme Submit Interpreter, mal by čakať na pripojenie z Bin Picking Studia. Pred začatím procesu kalibrácie sa uistíme, či sú vybraté nástroj a základ. Ak nástroj a základ nie sú vybraté, server nebude schopný nahlásiť správny TCP do Bin Picking Studia a bude hlásiť poruchu nástroja a základne. Následne vyberieme \$ NULLFRAME pre rámce výber nástroja a výber základu, viď obrázok 3.7, aby sme umožnili správnu funkčnosť stavu servera. Toto nastavenie musíme vykonať, pretože Bin Picking Studio si prepočítava súradnice nástroja a základu na základe modelov nahraných v Bin Picking Studiu. Ďalšia potrebná konfigurácia je nastavenie IpoMode na prírubu, pretože v našom prípade sa nástroj nachádza na prírubu manipulátora [22].

## Nastavenie Bin Picking užívateľa

Bin Picking užívateľa je možné spustiť pomocou hlavného programu (napríklad basic\_application.src). Hlavný program obsahuje definície pevných pozícií manipulátora, inicia-



Obr. 3.6: Odvolanie Submit Interpretera

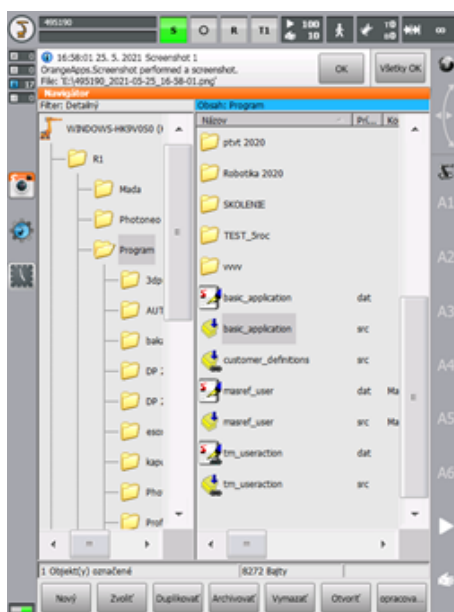


Obr. 3.7: Výber \$ NULLFRAME pre rámce výber nástroja a výber základu

lizáciu výberu z koša a hlavnú slučku vyberania z koša. Od používateľa sa očakáva, že pred výberom tohto súboru na spustenie zadefinuje domovskú pozíciu manipulátora, fázy začiatok a koniec a pozíciu umiestnenia odoberanej súčiastky. Prepne sa do expertného režimu a zvolíme program `basic_application.src`. Zvolenie programu vykonáme kliknutím na daný program a následným kliknutím na tlačidlo `Zvoliť` na spodnej lište obrazovky smartHMI, viď obrázok 3.8 [22].

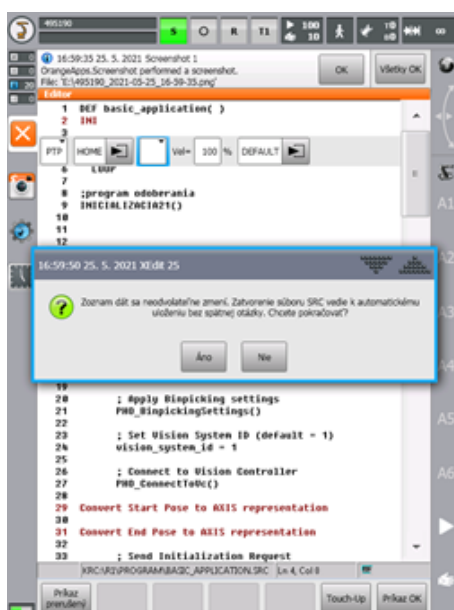
Pred samotným spustením programu je nutné predefinovať jednotlivé pozície podľa nášho konkrétneho riešenia. Vyberieme programový riadok s pozíciou, ktorú chceme zme-





Obr. 3.8: Zvolenie programu main\_ application.src

niť, v našom príklade je to domovská pozícia manipulátora. Stlačením tlačidla výber bloku sa presunie ukazovateľ na aktuálny riadok. Teraz môžeme prejsť manipulátorom pomocou smartPADu na novú pozíciu a zadefinovať ju stlačením potvrdzujúcich tlačidiel, viď obrázok 3.9. Pomocou formulárov používaných v jazyku KRL môžeme zmeniť nielen pozíciu, ale aj druh pohybu manipulátora a rýchlosť pohybu. Rovnakým postupom zmeníme pozície začiatok, koniec a pozíciu umiestnenia odoberanej súčiastky.



Obr. 3.9: Potvrdenie zmeny pozície manipulátora

### 3.1.3 Programové volania KRL

Rozhranie Photoneo KUKA bolo navrhnuté na ľahkú integráciu do existujúcich aplikácií napísaných v KRL kóde. KRL API a KRL Config by sa mali použiť na prispôsobenie predvolenej šablóny na vyberanie z koša požiadavkám systému konkrétnej aplikácie [22].

KRL API – je sada KRL postupov používaných na riadenie sekvencie odoberania; nachádza sa tu inicializácia odoberania, hlavná slučka a implementované spracovanie chýb [22]

KRL CONFIG – `customer_definitions.src` umožňuje používateľom konfigurovať špecifické nastavenia aplikácií, ako napr. príkazy chápadla, rýchlosť, zrýchlenie a približné vzdialenosti [22]

#### KRL API, riadenie sekvencie odoberania

Táto časť popisuje funkcie dostupných volaní API poskytovaných rozhraním Photoneo KUKA. Tieto procedúry sú určené na vysokoúrovňové riadenie rutiny vyberania z koša a zvyčajne sa volajú v hlavnom programe. Jednotlivé volania sú v podstate podprogramy obsiahnuté v moduloch Photoneo KUKA [22].

`PHO_ConnectToVC ()` – funkcia na vytvorenie pripojenia k riadiacemu systému Vision; blokujúca funkcia pozastaví program, kým sa nenadviaže správne pripojenie

`PHO_RequestInit (start_pose: IN, end_pose: IN)` – požiadavka na inicializáciu aplikácie bin picking na strane Vision riadiaceho systému; štart a koncové pózy definované operátorom sa prenesú do Vision riadiaceho systému a používajú sa v plánovaní trajektórie ako štartovacie a koncové body

`PHO_RequestScan ()` – požiadavka na spustenie skenovania s následnou lokalizáciou s plánovaním cesty

`PHO_WaitForScan ()` – funkcia čakania na ukončenie skenovania; musí byť volaný `PHO_RequestScan`

`PHO_RequestTrajectory ()` – žiadosť na trajektóriu z bin pickingu na odoslanie do riadiaceho systému manipulátora

`PHO_ReceiveTrajectory ()` – funkcia na príjem požadovaných trajektórií bin pickingu (pozostávajúcich zo sekvencie dráh a operácií chápadiel); musí sa volať po volaní `PHO_RequestTrajectory`

`PHO_RequestPickFailed ()` – požiadavka použitá, keď sa nepodarí vybrať objekt počas vykonávania bin pickingu (napr. uviaznutý predmet) a uchopovač (alebo iný systém) je vybavený snímačom, ktorý je schopný vyhodnotiť chybu; program Bin Picking Studio

znižuje preferenciu objektu a nebude zvolený na odobranie v nasledujúcom cykle

PHO\_ RequestChangeSol (solution\_ id: IN) – žiadosť o zmenu zadaného riešenia. Vstupný parameter solution\_ id definuje ID riešenia, ktoré sa má zadať.

PHO\_ PickPart () – vykoná štandardnú sekvenciu vyberania z koša, ktorá pozostáva z rôzneho počtu operácií, ktoré sa majú vykonať. Je to taktiež blokovacia funkcia, program nebude pokračovať, kým sa nedokončí celá sekvencia. Pred spustením tejto funkcie sa vždy ubezpečíme, že je manipulátor v pozícii Start Pose.

## **KRL CONFIG, konfigurácia**

Táto časť popisuje dostupné volania CONFIG poskytované rozhraním Photoneo KUKA. Tieto procedúry sa nachádzajú v programe customer\_ definitions.src a sú určené na konfiguráciu niekoľkých nastavení vychytávania z koša vrátane chápadla, rýchlosti trajektórie a úpravy aproximácie. Jednotlivé volania sú v podstate podprogramy obsiahnuté v moduloch Photoneo KUKA [22].

PHO\_ GripperAttach () - tu implementujeme príkaz na pripojenie odoberaného objektu k chápadlu

PHO\_ GripperDetach () - tu implementujeme príkaz na odpojenie odoberaného objektu od chápadla

PHO\_ GripperUser\_ 1 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 2 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 3 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 4 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 5 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 6 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 7 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ GripperUser\_ 8 () - implementujeme vlastný príkaz uchopovača používateľa

PHO\_ BinpickingSettings () - špecifikácia rýchlosti, zrýchlenia a približnej vzdialenosti trajektórie pre vyberanie z koša (segmenty SPLINE)

## KRL ERROR, chybové hlásenia

Rozhranie Photoneo KUKA poskytuje základné spracovanie chýb. Ak sa vyskytne chyba počas operácií vyberania z koša, `error_code` `global_variable` obsahuje kód konkrétnej chyby, vďaka ktorému vieme identifikovať zdroj problému [22].

PHO\_OK: = 0 - odozva služby od riadiaceho systému Vision je platná

PHO\_SERVICE\_ERR: = 1 - odozva služby od riadiaceho systému Vision je neplatná

PHO\_UNKNOWN\_REQ: = 2 - riadiaci systém Vision dostal neznámu požiadavku

PHO\_COM\_FAILURE: = 3 - zlyhanie komunikácie z dôvodu chybného pripojenia zásuvky

PHO\_BAD\_DATA: = 4 - kontrola overenia údajov zlyhala

PHO\_TIMEOUT: = 5 - zlyhanie komunikácie z dôvodu časového limitu zásuvky

PHO\_LONG\_TRAJ: = 6 - dĺžka jedného zo segmentov trajektórie bola vyššia ako maximálna povolená veľkosť trajektórie

PHO\_UNKNOWN\_ERR: = 99 - nešpecifikovaná interná chyba

PHO\_PLANNING\_FAILED: = 201 - plánovanie trajektórie zlyhalo

PHO\_NO\_PART\_FOUND: = 202 - žiadna súčiastka nebola lokalizovaná

PHO\_NOT\_INITIALIZED: = 203 - aplikácia na vyberanie z koša nebola vo Vision riadiacom systéme správne inicializovaná

PHO\_PART\_LOST: = 204 - súčiastka sa stratila počas vykonávania pohybu

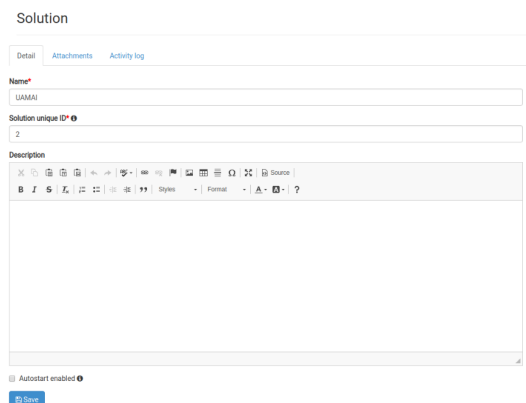
PHO\_WRONG\_BP\_CONFIG: = 255 - nesprávna konfigurácia bin pickingu

## 3.2 Konfigurácia Bin Picking Studia

### 3.2.1 Konfigurácia riešenia

Ako prvý krok sme si vytvorili nové riešenie s názvom a identifikačným číslom (ID) vid' obrázok 3.10. ID sme zvolili 2, pretože ID1 je už výrobcom použité pre výrobné nastavenia softvéru. V tomto kroku môžeme do podrobností vytvoriť popis nastavenia nášho riešenia. Tento popis sa vytvára najmä vtedy, ak je vytvorených viacero riešení s viacerými zariadeniami na rôznych pracoviskách pri použití jednej riadiacej jednotky Vision. Konfigurácia riešenia nám ďalej poskytuje vkladanie rôznych príloh, ktoré by sme mohli potrebovať

pri danom riešení. Pre automatické spustenie nášho riešenia hneď po spustení riadiaceho systému Vision môžeme zakliknúť povolenie automatického štartu, toto zaškrtvacie pole sa nachádza nad tlačidlom uložiť na konci konfigurácie riešenia. Po nastavení konfigurácie riešenia naše riešenie uložíme kliknutím na tlačidlo uložiť [21].



Obr. 3.10: Konfigurácia riešenia

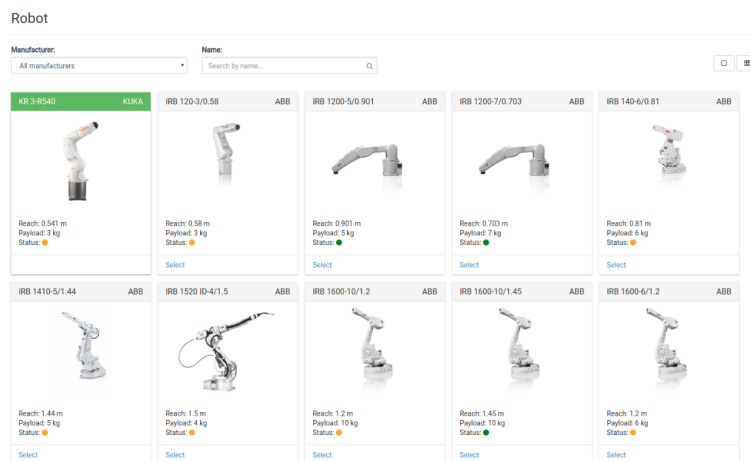
## Konfigurácia manipulátora

Po úspešnom uložení konfigurácie riešenia prejdeme na konfiguráciu manipulátora. Táto konfigurácia nám poskytuje výber z manipulátorov rôznych druhov. Z tohto zoznamu manipulátorov sme zvolili manipulátor použitý v našej úlohe, teda manipulátor KUKA KR KR3R540. Pri výbere manipulátora sa zobrazujú statusy konkrétneho manipulátora. Tieto statusy sú indikované farebným rozlíšením, zelená indikácia znamená, že manipulátor je validovaný na skutočnej inštalácii s fyzickým robotom, oranžová farba indikuje, že daný model je vytvorený a otestovaný v simulácii. Posledným statusom je červený indikátor, ktorý signalizuje, že aktuálny robot je zatiaľ vo vývoji, a tak ho zatiaľ nie je možné použiť. Konfigurácia manipulátora je zobrazená na obrázku 3.11 [21].

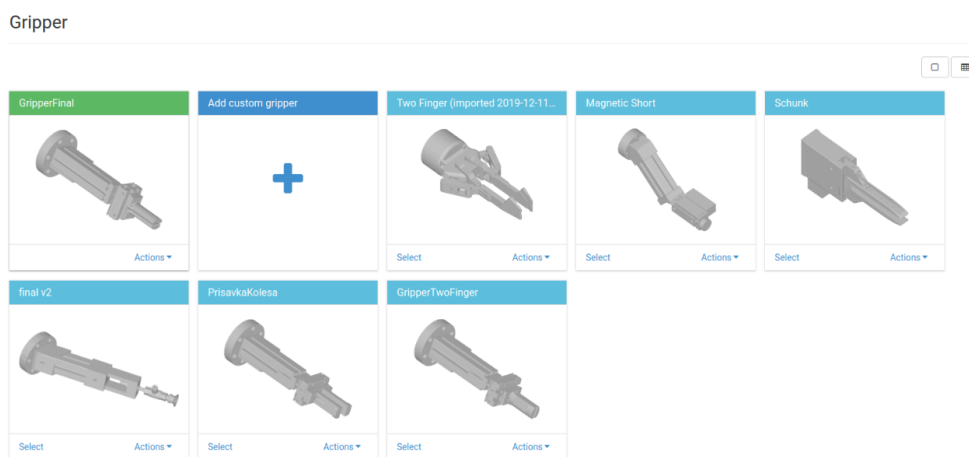
## Konfigurácia nástroja

Pri konfigurácii nástroja sme ako prvý krok pridali náš nástroj do databázy nástrojov ako 3D model v STL formáte. Nástroj musel byť namodelovaný tak, že jeho centrálny bod, teda bod  $[0,0,0]$ , bol umiestnený v strede päty nástroja a v kartézskom súradnicovom systéme smerovala os  $Z$  z tohto stredu v pozitívnom smere k špičke nástroja. Toto nastavenie bolo pri modelovaní nástroja potrebné dodržať, pretože systém Bin pickingu pripája centrálny bod modelu ku koncovému bodu ramena manipulátora [21].

Správne nastavenie nášho modelu nástroja si môžeme overiť pri nahrávaní modelu do databázy modelov nástrojov. Ak máme dodržané spomínané nastavenia, zvolíme si náš model kliknutím na rolovacie menu akcie a zvolíme ho ako aktívny. Aktívny model nástroja je od ostatných nástrojov rozlíšený zeleným orámovaním, vid' obrázok 3.12.



Obr. 3.11: Konfigurácia manipulátora



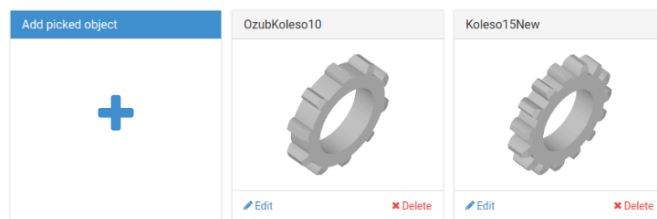
Obr. 3.12: Konfigurácia nástroja

## Konfigurácia objektu

Pri konfigurácii objektu, v tomto prípade odoberaného ozubeného kolesa, sme ako prvý krok pridali náš objekt do databázy odoberaných objektov ako 3D model v STL formáte. Pri nahraní objektu do databázy mu nastavíme meno a prípadne poznámku k aktuálnemu modelu objektu. Do tejto databázy je možné vložiť maximálne 4 rôzne objekty. Odoberaný objekt musel byť namodelovaný tak, aby sa súradnicový systém nástroja a objektu zhodoval [21].

Objekt sa používa pri nastavení uchopovacích bodov vo vizualizácii, určení lokalizačných modulov a na kontrolu kolízií. Volumetrický model sa generuje automaticky po nahraní 3D modelu do databázy. Nahrané objekty v databáze sú zobrazené na obrázku 3.13.

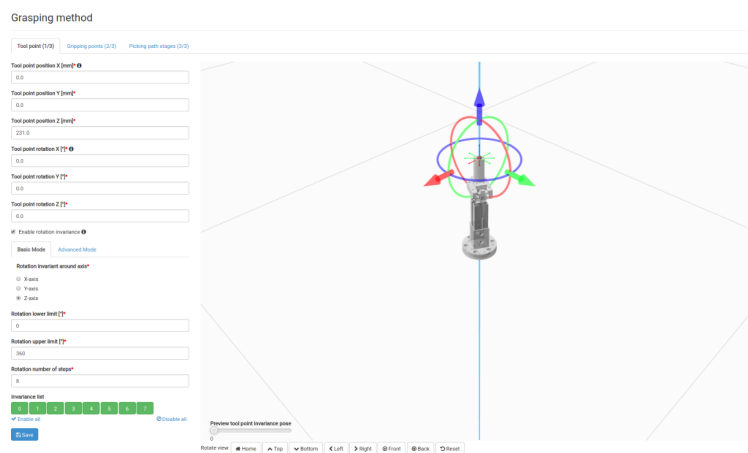
### Picked object



Obr. 3.13: Konfigurácia objektu

## 3.2.2 Metóda uchopenia

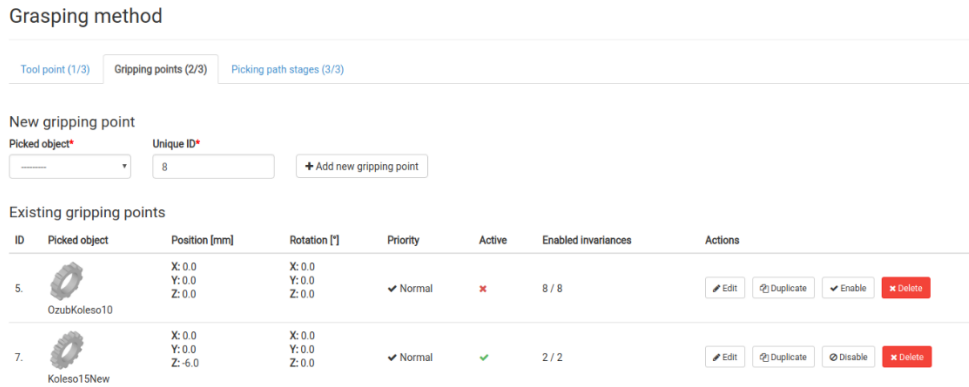
Metódu uchopenia môžeme charakterizovať ako prekrytie súradnicového systému nástroja so súradnicovým systémom objektu. V prvom kroku musíme zdefinovať bod na nástroji, kde bude uchopovač nadväzovať kontakt s odoberaným objektom. Tento bod sme posunuli z päty nástroja v osi Z, ďalej sme nastavili limit rotácie nástroja pri uchopovaní. Limit sme nastavili od 0 do 360 stupňov rotácie okolo osi Z, zvolili sme 8 bodov rotácie, čo znamená, že nástroj môže odoberať súčiastku v ôsmich natočeniach v rozmedzí našich limitov. V danej metóde uchopenia sa používa konvencia RGB (XYZ), kde šípky smerujú pozitívnym smerom v osi. Definovanie uchopovacieho bodu nástroja môžeme vidieť na obrázku 3.14 [21].



Obr. 3.14: Definovanie uchopovacieho bodu nástroja

Ďalším bodom v metóde uchopenia je definovanie uchopovacieho bodu odoberaného objektu. Najprv si zvolíme, aký objekt budeme odoberať, pokiaľ by sa nám náš objekt nezobrazoval medzi existujúcimi odoberanými objektmi, pridáme tento objekt z možností v rolovacom menu odoberaného objektu. Po následnom vybraní daného objektu mu zvolíme identifikačné číslo ID a stlačením tlačidla pridať nový uchopovací bod pridáme náš objekt do databázy existujúcich uchopovacích objektov. Pokiaľ sa nám náš objekt už zobrazuje, aktivujeme ho jednoduchým kliknutím na tlačidlo povoliť a v riadku nášho objektu v stĺpci aktívne sa nám zobrazí zelený indikátor. Výber uchopovacieho objektu v metóde

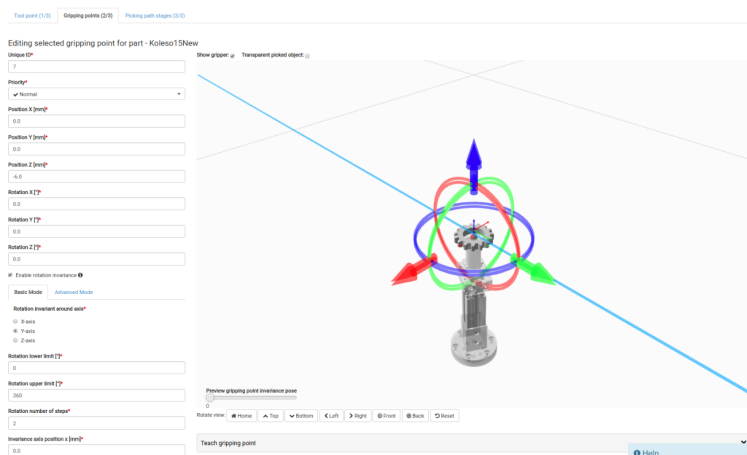
uchopenia je zobrazené na obrázku 3.15.



Obr. 3.15: Výber odoberanej súčiastky

Keď máme súčiastku aktívnu, môžeme prejsť na nastavenie jej uchopovacieho bodu. Tento bod sme umiestnili do stredu odoberanej súčiastky, pretože ju budeme uchopovať za jej vnútorný priemer. Tento bod sme následne posunuli o 6 mm v smere osi Z, aby sme pri odoberaní súčiastky predišli novej kolízii, keďže nám stačí uchopiť danú súčiastku v polovici jej výšky.

Rovnako ako pri nastavovaní bodu pri nástroji zvolíme limit rotácie od 0 do 360 stupňov, ale v tomto prípade bude rotácia okolo osi Y s dvoma bodmi rotácie, čo znamená, že súčiastku bude môcť manipulátor odoberať z jej hornej aj spodnej strany, vid' obrázok 3.16. Nastavenie limitu rotácie je dôležité hlavne z toho dôvodu, že sú na sebe rôzne orientované objekty. V takom prípade by mohol riadiaci systém Vision vyhodnotiť danú pozíciu súčiastky ako nedosiahnuteľnú.



Obr. 3.16: Definovanie uchopovacieho bodu súčiastky

Pokiaľ máme definované uchopovacie body nástroja a odoberanej súčiastky, môžeme prejsť na posledný bod metódy uchopovania, a to na nastavenie jednotlivých fáz odobera-



nia súčiastky. Sú to fázy skladajúce sa z piatich povinných bodov, a to štart, priblíženie, uchopenie, vzdialenie a koniec. Body priblíženia a vzdialenia sú preddefinované lineárne pohyby.

Prvou fázou je štart, ktorá definuje východiskovú polohu manipulátora pri odoberaní súčiastky. Pri tejto fáze nastavujeme len meno procedúry, v našom prípade oddelenie, čo znamená, že nástroj bude v tomto bode uzavretý, viď obrázok 3.17. Nastavenie tejto procedúry predchádza možnej kolízii pri odoberaní súčiastky zapríčinením chybového programu zadaného užívateľom. Vďaka tejto procedúre vieme, že pokiaľ sa manipulátor dostane do štartovacej pozície odoberania, určite bude mať nástroj, teda koncový efektor manipulátora pripravený v stave odoberania, čiže v našom prípade bude uzavretý. Po nastavení tejto fázy uložíme danú konfiguráciu stlačením tlačidla uložiť.

Grasping method

Tool point (1/3) Gripping points (2/3) Picking path stages (3/3)

#	Name	Actions	Trajectory Type	Position Tolerance
1	Start	<a href="#">Edit</a>	N/A	N/A
2	Approach	<a href="#">Edit</a>	Joint	Approx.
3	Grasp	<a href="#">Edit</a>	Linear	Precise
4	Deapproach	<a href="#">Edit</a>	Linear	Approx.
5	End	<a href="#">Edit</a>	Joint	Approx.

[+ Add a new picking path stage](#)

Edit picking path stage

Type\*

Start

Procedure name

Detach

[Save](#)

Obr. 3.17: Definovanie fázy štart

Následne definujeme fázu priblíženia, v tejto fáze sa manipulátor s uchopovačom presunie do polohy definovanej relatívnym vyrovnaním vzhľadom na umiestnený uchopovací bod súčiastky. Relatívny posun je nastavený na 100 mm v smere od osi Z uchopovacieho bodu, viď obrázok 3.18. Týmto nastavením predchádzame prípadnej kolízii pri odoberaní súčiastky, keďže vieme, že robot sa v ďalšom kroku pohne už len v osi odoberania preddefinovaným lineárnym pohybom.

Pri nastavení fázy priblíženia môžeme definovať taktiež relatívny posun v každej z osí. Môžeme zakliknúť zaškrtávacie tlačidlo plánovania lineárnej trajektórie, čo je v našom prípade nepotrebné, keďže nemáme v okolí odoberaných súčiastok kolízne objekty. Po nastavení tejto fázy uložíme danú konfiguráciu stlačením tlačidla uložiť.

Po priblížení nasleduje fáza uchopenie, ide o výrobcom prednastavený lineárny pohyb definovaný ako posunutie zo stavu priblíženia k uchopeniu. Pri tejto fáze nastavujeme len meno procedúry, v našom prípade spojenie, čo znamená, že nástroj bude v dosiahnutí tohto bodu otvorený, teda uchopí odoberanú súčiastku, viď obrázok 3.19. Nastavenie

Tool point (1/3) Gripping points (2/3) Picking path stages (3/3)

#	Name	Actions	Trajectory Type	Position Tolerance
1	Start	<a href="#">Edit</a>	N/A	N/A
2	Approach	<a href="#">Edit</a>	Joint	Approx.
3	Grasp	<a href="#">Edit</a>	Linear	Precise
4	Deapproach	<a href="#">Edit</a>	Linear	Approx.
5	End	<a href="#">Edit</a>	Joint	Approx.

[+ Add a new picking path stage](#)

Edit picking path stage

Type\*  
Approach

Origin of relative offset\*  
Found gripping point

Relative offset X [mm]  
0.0

Relative offset Y [mm]  
0.0

Relative offset Z [mm]  
-100.0

Plan linear trajectory

Position tolerance\*  
Approximative

Procedure name  
\_\_\_\_\_

[Save](#)

Obr. 3.18: Definovanie fázy priblíženia

tejto procedúry predchádza možnej kolízii pri uchopení súčiastky zapríčinením chybového programu zadaného užívateľom. Vďaka tejto procedúre vieme, že pokiaľ sa manipulátor dostane do pozície definovanej pre uchopenie, určite bude mať nástroj, teda koncový efektor manipulátora v stave uchopenia, čiže v našom prípade bude otvorený. Toto nastavenie nám zaručí uchopenie súčiastky v dosiahnutí pozície definovanej pre uchopenie. Po nastavení tejto fázy uložíme danú konfiguráciu stlačením tlačidla uložiť.

Grasping method

Tool point (1/3) Gripping points (2/3) Picking path stages (3/3)

#	Name	Actions	Trajectory Type	Position Tolerance
1	Start	<a href="#">Edit</a>	N/A	N/A
2	Approach	<a href="#">Edit</a>	Joint	Approx.
3	Grasp	<a href="#">Edit</a>	Linear	Precise
4	Deapproach	<a href="#">Edit</a>	Linear	Approx.
5	End	<a href="#">Edit</a>	Joint	Approx.

[+ Add a new picking path stage](#)

Edit picking path stage

Type\*  
Grasp

Plan linear trajectory

Position tolerance\*  
Precise

Procedure name  
Attach

[Save](#)

Obr. 3.19: Definovanie fázy uchopenia

Ak máme uchopenú súčiastku, nakonfigurujeme fázu vzdialenia už z uchopenou súčiastkou. V našom prípade zvolíme relatívny posun s ohľadom na základňu manipulátora v smere osi Z o 100 mm, viď obrázok 3.20. Týmto nastavením predchádzame prípadnej

kolízii pri odoberaní súčiastky, keďže vieme, že sa manipulátor pohne už len v osi odoberania preddefinovaným lineárnym pohybom.

Pri nastavení fázy vzdialenia môžeme definovať taktiež relatívny posun v každej z osí. Pri konfigurácii vzdialenia môžeme zakliknúť zaškrtačacie tlačidlo plánovania lineárnej trajektórie, čo je v našom prípade nepotrebné, keďže nemáme v okolí odoberaných súčiastok kolízne objekty. Po nastavení tejto fázy uložíme danú konfiguráciu stlačením tlačidla uložiť.

#	Name	Actions	Trajectory Type	Position Tolerance
1	Start	<a href="#">Edit</a>	N/A	N/A
2	Approach	<a href="#">Edit</a>	Joint	Approx.
3	Grasp	<a href="#">Edit</a>	Linear	Precise
4	Deapproach	<a href="#">Edit</a>	Linear	Approx.
5	End	<a href="#">Edit</a>	Joint	Approx.

Edit picking path stage

Type\*  
 Deapproach

Origin of relative offset\*  
 Robot

Relative offset X [mm]  
 0.0

Relative offset Y [mm]  
 0.0

Relative offset Z [mm]  
 100.0

Plan linear trajectory

Position tolerance\*  
 Approximative

Procedure name\*  
 \_\_\_\_\_

Obr. 3.20: Definovanie fázy vzdialenia

Ako poslednú fázu v metóde uchopenia zdefinujeme fázu koniec, kde sa manipulátor aj s uchopenou súčiastkou presunie na pozíciu definovanú v riadiacom systéme manipulátora. Pri tejto fáze nastavujeme meno procedúry na oddelenie, čo znamená, že nástroj bude v dosiahnutí tohto bodu zatvorený, teda pustí zo zovretia odoberanú súčiastku. Nastavenie poslednej fázy uchopovacej metódy je zobrazené na obrázku 3.21.

Nastavenie tejto procedúry predchádza možnej kolízii pri ukladaní súčiastky zapríčinením chybového programu zadaného užívateľom. Vďaka tejto procedúre vieme, že pokiaľ sa manipulátor dostane do pozície definovanej pre koniec, určite bude mať nástroj, teda koncový efektor manipulátora v stave pustenia, čiže v našom prípade bude uzavretý. Toto nastavenie nám zaručí uloženie súčiastky v dosiahnutí pozície definovanej pre koniec. Po nastavení tejto fázy uložíme danú konfiguráciu stlačením tlačidla uložiť.

### 3.2.3 Systémové videnie

Nastavenie systémového videnia sa skladá z nastavenia mena, jedinečného ID, ID skenera, modelu skenera, jeho umiestnenia, odoberanej súčiastky, lokalizačného profilu a kalibrač-

Grasping method

Tool point (1/3) Gripping points (2/3) Picking path stages (3/3)

#	Name	Actions	Trajectory Type	Position Tolerance
1	Start	<a href="#">Edit</a>	N/A	N/A
2	Approach	<a href="#">Edit</a>	Joint	Approx.
3	Grasp	<a href="#">Edit</a>	Linear	Precise
4	Deapproach	<a href="#">Edit</a>	Linear	Approx.
5	End	<a href="#">Edit</a>	Joint	Approx.

+ Add a new picking path stage

Edit picking path stage

Type\*

End

Plan linear trajectory

Position tolerance\*

Approximative

Procedure name

Detach

Save

Obr. 3.21: Definovanie fázy koniec

nej matice, vid' obrázok 3.22. Všetky tieto hodnoty sa preberajú už z predošlých nastavení, avšak je potrebné ich skontrolovať a uložiť, aby sme mohli pokračovať ďalej v nastavovaní Bin Picking Studia. Kalibračná matica sa vytvára automaticky po ukončení kalibrácie, viac podrobností o kalibrácii je uvedených v kapitole 3.2.5.

Kalibračná matica je v podstate transformačná matica medzi kamerou a súradnicovým priestorom robota. Vypočítaná matica transformuje body zo súradnicového systému kamery na súradnicový systém robota. Pri vytváraní nového riešenia a nastavenia nového systémového videnia je nutné vykonať kalibráciu vždy nanovo kvôli vzájomnému nastaveniu medzi manipulátorom a skenerom. Kalibračnú maticu môžeme vložiť aj ručne, pokiaľ ju vieme určiť na dané systémové videnie. Máme ju možnosť aj prípadne upraviť, ak by sa nezhodovala či už s kalibráciou, alebo s kalibračnou maticou vloženou ručne [23].

### 3.2.4 Konfigurácia siete

Pre správne fungovanie komunikácie medzi riadiacim systémom manipulátora, skenerom a riadiacim systémom Vision je potrebné nakonfigurovať sieť pre všetky spomenuté zariadenia. Ako prvé nastavíme komunikačné rozhranie manipulátora. Musíme dodržať rovnakú podsieť s riadiacim systémom Vision. Po zadaní IPv4 adresy a masky podsiete klikneme na tlačidlo uložiť, následne sa naša konfigurácia uloží a môžeme pokračovať na sieťovú konfiguráciu ďalších zariadení. Komunikačné rozhranie manipulátora je zobrazené na obrázku 3.23.

Ako ďalší krok v konfigurácii siete nastavíme komunikačné rozhranie skenera. Po zadaní IPv4 adresy a masky podsiete nastavíme skeneru statickú IP adresu. Následne klikneme na tlačidlo uložiť, naša konfigurácia sa uloží a môžeme pokračovať na sieťovú konfiguráciu posledného zariadenia. Komunikačné rozhranie skenera je zobrazené na obrázku 3.24.

## Vision system

**Name\***

**Unique ID\***

**Scanner ID\***  
 Available scanners ▾

After a change of vision system scanner ID, make sure that calibration is still valid for the new scanner.

**Scanner model\***

**Scanner mount position\***

**Description**

**Picked object\***

**Localization profile\* ⓘ**  
Currently: [RobLabUAMAI\\_generated\\_MWB4x.plcf](#)

+ Create new 📁 Upload other 📄 Edit current

**Calibration matrix**

-0.986222	0.028219	0.163002	-77.088323
0.028544	0.999592	-0.000346	440.975597
-0.162945	0.004312	-0.986626	1226.799075
0.000000	0.000000	0.000000	1.000000

[🔗 Edit matrix](#) [📄 Copy to clipboard](#)

💾 Save

Obr. 3.22: Nastavenie systémového videnia

Robot interface Scanner interface Robotic controller

**Robot interface**

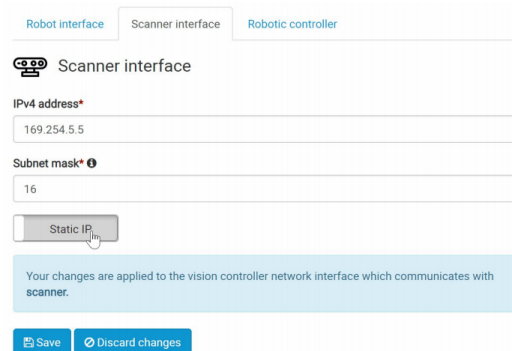
**IPv4 address\***

**Subnet mask\* ⓘ**

Your changes are applied to the vision controller network interface which communicates with robot.

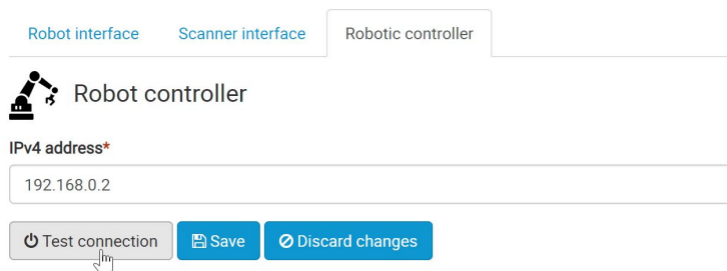
💾 Save 🗑 Discard changes

Obr. 3.23: Komunikačné rozhranie manipulátora



Obr. 3.24: Komunikačné rozhranie skenera

V poslednom kroku konfigurácie siete nastavíme komunikačné rozhranie riadiaceho systému manipulátora. Po zadaní IPv4 adresy otestujeme našu sieťovú konfiguráciu stlačením tlačidla Testovanie prepojenia. Po úspešnom teste klikneme na tlačidlo uložiť, naša konfigurácia sa následne uloží. Komunikačné rozhranie riadiaceho systému manipulátora je zobrazené na obrázku 3.25.



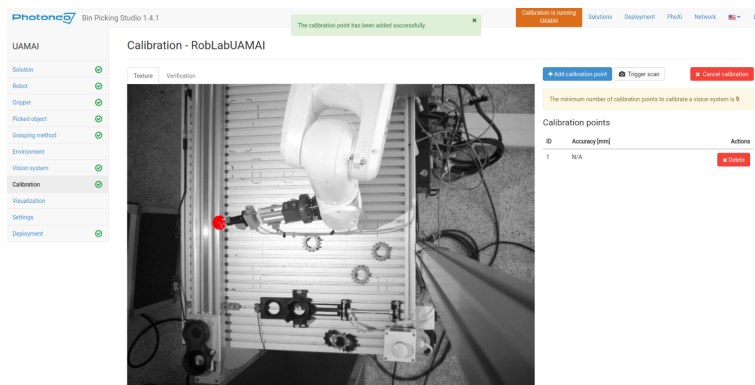
Obr. 3.25: Komunikačné rozhranie riadiaceho systému manipulátora

### 3.2.5 Kalibrácia

Kalibrácia pozostáva z kalibrácie medzi manipulátorom a skenerom. Kalibrácia kamery robota podporuje kalibráciu pre dva typy uchytenia skenera, a to pre typ, keď je skener umiestnený v statickej polohe a pre typ, keď je skener namontovaný na ramene manipulátora. Pokiaľ je skener umiestnený v statickej polohe (zvyčajne nad robotom), kalibrácia sa vykonáva použitím klasickej pingpongovej loptičky s priemerom 38mm pripevnenej ku špičke koncového efektora manipulátora. Táto kalibrácia sa bude označovať ako sférická kalibrácia. Pokiaľ je skener namontovaný na ramene manipulátora a pohybuje sa spolu s manipulátorom, kalibrácia sa vykonáva pomocou špeciálnej značkovej platne poskytnutej spoločnosťou Photoneo. Táto kalibrácia sa bude označovať ako kalibrácia štítkovej tabuľky [23].

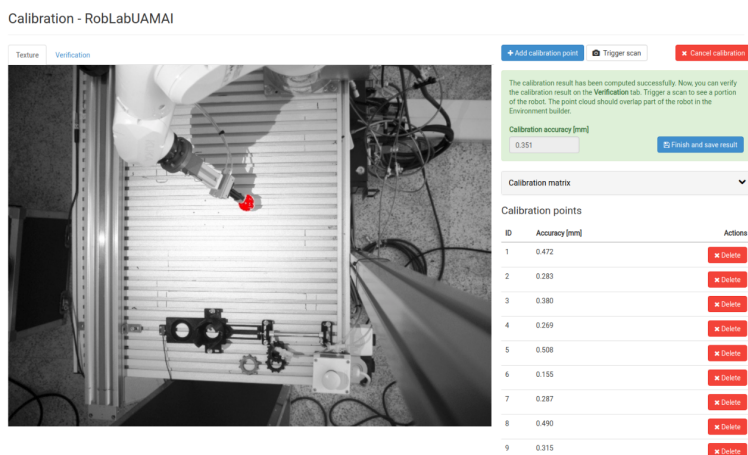
V našom prípade ide o typ kalibrácie, kde je skener umiestnený v statickej polohe nad manipulátorom. Manipulátor sa kalibruje vzhľadom na svoju základňu a skener vzhľadom

k základni v centre čipu snímača. Po kalibrácii sa vypočítajú výsledky lokalizácie priamo v základných súradniciach manipulátora. Na kalibráciu je potrebné vykonať minimálne 9 kalibračných bodov, v ktorých musí byť manipulátor natočený v každej osi rôzne a skener musí identifikovať pingpongovú loptičku, vid' obrázok 3.26. Pokiaľ skener identifikuje loptičku správne, zobrazí sa na obrazovke červenou farbou [23].



Obr. 3.26: Prvý identifikovaný bod kalibrácie

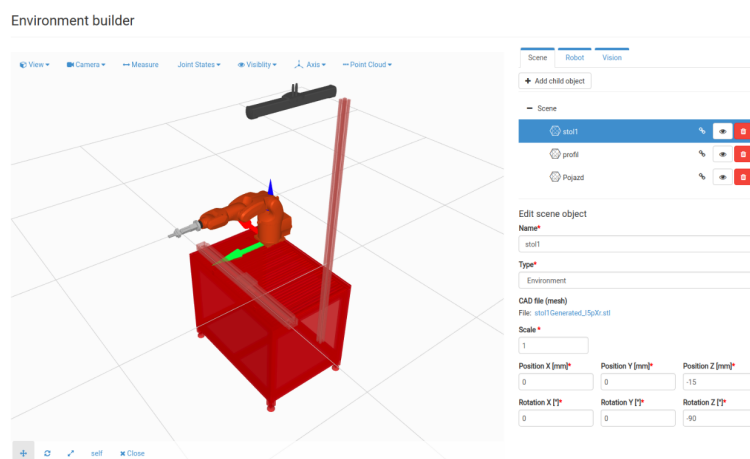
Ak skener pri jednotlivých bodoch identifikoval loptičku a má dostatočné rozdiely v natočení jednotlivých kĺbov, systém zapíše tento kalibračný bod do zoznamu kalibračných bodov a vypíše nám systémovú správu, že bol tento bod úspešne pridaný. Po dosiahnutí 9 bodov kalibrácie sa nám zobrazí celková presnosť kalibrácie vypočítaná na základe kalibračných bodov. Po dosiahnutí celkovej presnosti kalibrácie a jej následným uložením sa vytvorí kalibračná matica potrebná na nastavenie systémového videnia. Kalibračná matica je transformačná matica medzi kamerou a súradnicovým priestorom robota. Vypočítaná matica transformuje body zo súradnicového systému kamery na súradnicový systém robota. V našom prípade sa nám podarilo nakalibrovať manipulátor a skener s celkovou presnosťou 0,351 mm, vid' obrázok 3.27.



Obr. 3.27: Celková presnosť kalibrácie

### 3.2.6 Tvorba prostredia

Pri tvorbe prostredia nastavíme ako prvú scénu prostredia. Scéna prostredia sa skladá z manipulátora a všetkých kolíznych objektov na našom pracovisku. Jednotlivé objekty nahráme do tohto prostredia ako STL modely. V našom prípade ide o kolízne objekty stôl, na ktorom je uchytený manipulátor, lineárny pojazd, na ktorom je umiestnený blok motora na uloženie odoberaného objektu a hliníkový profil, na ktorom je uchytený skener. Zadanie kolíznych objektov je dôležité na predídenie kolízie manipulátora s možnými objektmi. Manipulátor sa zobrazí v scéne prostredia automaticky, tento model je vytiahnutý z databázy manipulátorov, okrem manipulátora sa automaticky zobrazí aj jeho koncový efektor. Toto zobrazenie, teda manipulátor aj so svojim koncovým efektorom, sa musí úplne zhodovať s reálnym systémom. Ostatné kolízne objekty vkladáme do nastavenia scény prostredia v posuvoch a rotáciách osí X, Y a Z vzhľadom na základňu manipulátora, vid' obrázok 3.28 [23].

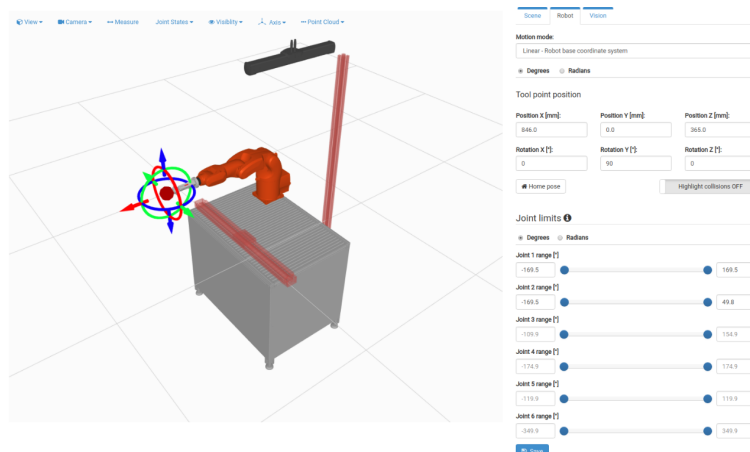


Obr. 3.28: Tvorba scény prostredia

V ďalšom kroku nastavíme manipulátor v prostredí, skontrolujeme nastavenie pozície bodu nástroja a následne nastavíme limity manipulátora v jednotlivých jeho osiach rotácie, vid' obrázok 3.29. Pri nastavení bodu nástroja je dôležité, aby sa zhodoval s bodom uchopovania, teda musí byť v našom prípade na špičke nástroja. V tomto nastavení môžeme bod posúvať v súradniciach X, Y a Z a taktiež určiť rotáciu okolo týchto osí. Nastavenie limitov manipulátora sa používa hlavne vtedy, ak vieme, že mimo scény prostredia sú kolízne objekty, s ktorými by mohol manipulátor prísť do kolízie. Pri tejto konfigurácii je dôležité dané nastavenia uložiť kliknutím na tlačidlo uložiť.

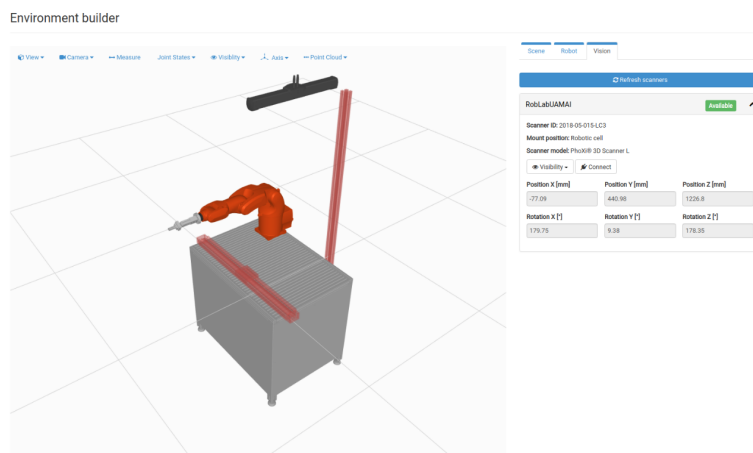
V poslednom kroku tvorby prostredia aktualizujeme zoskenovanie nastaveného prostredia. Pokiaľ sa zhoduje tvorba nášho prostredia s obrazom vytvoreným skenerom, rozsvieti sa nám zelené tlačidlo. Ďalej sa zobrazí ID skenera, montážna pozícia a jeho umiestnenie voči manipulátorovi, vid' obrázok 3.30. V tomto okne môžeme menením parametrov zobrazenia skrývať a odkrývať zariadenia a kolízne objekty použité v nastavení tohto prostredia. Táto zmena zobrazenia sa využíva hlavne vtedy, ak je v prostredí použitých mnoho zariadení a chceme skontrolovať konkrétne pozície jednotlivých zariadení





Obr. 3.29: Nastavenie manipulátora v prostredí

a kolíznych objektov. Pokiaľ máme všetky parametre v tomto okne skontrolované, môžeme pokračovať kliknutím na tlačidlo pripojiť. Po úspešnom pripojení skenera s riadiacim systémom manipulátora a riadiacim systémom Vision máme ukončenú konfiguráciu Bin Picking Studia a môžeme toto prostredie používať na potrebu skenovania.



Obr. 3.30: Aktualizácia nastaveného prostredia

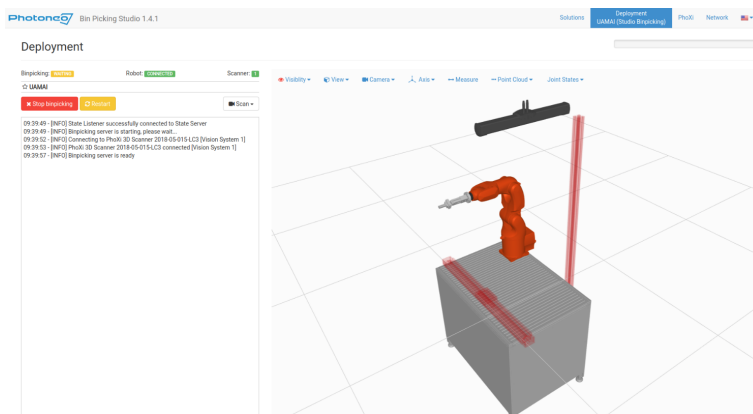
## 3.3 Realizácia

### 3.3.1 Spustenie Bin Picking Studia

Pokiaľ na poslednom tlačidle hlavného panela konfigurácie Bin Picking Studia svieti zelené pole, ktoré indikuje správnosť nastavení a kompatibilitu so skenerom, kliknutím na toto tlačidlo spustíme proces skenovania. Zobrazí sa nám užívateľská obrazovka, na ktorej môžeme vidieť signalizačný riadok binpickingu, manipulátora a skenera. Ako vidíme na obrázku 3.31, manipulátor aj skener sú pripravené na proces skenovania a binpicking

je pripravený v stave čakania. Binpicking čaká na signál povolenia pohybov manipulátora z riadiaceho systému manipulátora. Podrobné informácie o procese skenovania sa nám zobrazujú v stavovom riadku, ktorý je umiestnený hneď pod signalizačným riadkom. Na tejto užívateľskej obrazovke si môžeme nastaviť, čo chceme, aby systém zobrazoval. Môžeme napríklad vypnúť zobrazenie jednotlivých kolíznych objektov alebo používaných zaradení, pokiaľ by nám to prekážalo pri manipulácii v tomto prostredí. Taktiež môžeme použiť funkciu merania, pomocou ktorej si vieme skontrolovať rozmery v našom prostredí a porovnať ich s rozmermi reálnych objektov.

Nad stavovým riadkom sa taktiež nachádza tlačidlo na zastavenie Bin Picking Studia a tlačidlo na reštartovanie tohto softvérového prostredia. Tieto tlačidlá sú dôležité hlavne pri prvotnom spustení Bin Picking Studia, kedy ešte môžeme ladiť niektoré parametre na strane riadiaceho systému manipulátora.



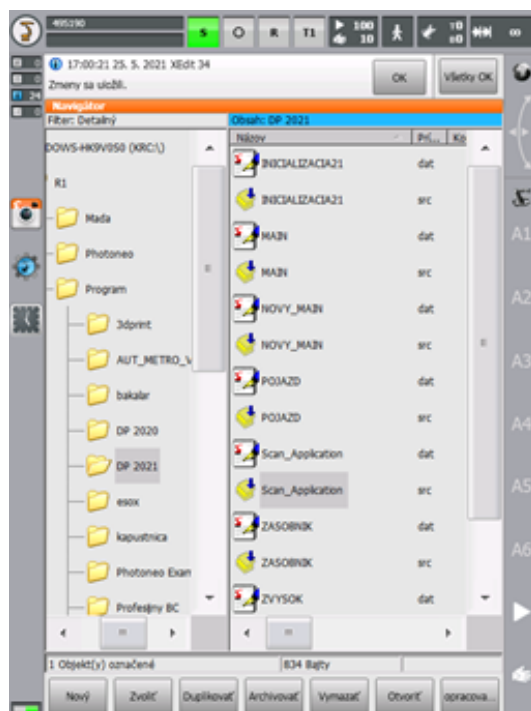
Obr. 3.31: Užívateľská obrazovka Bin Picking Štúdia

### 3.3.2 Spustenie programu KUKA

Program našej manipulačnej úlohy je vytvorený v jazyku KRL a je uložený v pamäti riadiaceho systému manipulátora. Tento program sme uložili do pamäte C nášho riadiaceho systému manipulátora do priečinkov R1 Program DP 2021. Náš program skenovania sme nazvali Scan\_Application, viď obrázok 3.32. K tomuto programu sa automaticky vytvoril aj dátový blok s rovnakým názvom, do ktorého sú automaticky vkladané všetky nastavenia nášho programu.

Ako príklad si môžeme uviesť nastavenie pozície nejakého bodu používaného v našom programe, kým v našom programe zadáme len druh pohybu, názov bodu a rýchlosť pohybu, ktoré následne v programe aj vidíme. Do dátového bloku sa prenesú kompletne informácie o tomto bode, ako sú presné súradnice v osiach robota a natočenia v jednotlivých kľboch manipulátora aj s nastavením rýchlosti a pomenovaním tohto bodu. Tento program zvolíme pomocou dotykovej obrazovky SmartHMI na SmartPADe a následne povolíme pohyby manipulátora pomocou jedného z tlačidiel na povolenie pohybov. Pokiaľ je program napísaný správne a riadiaci systém manipulátora nehlási žiadne chyby, môžeme

tlačidlom štart na SmartPADe spustí program našej manipulačnej úlohy.

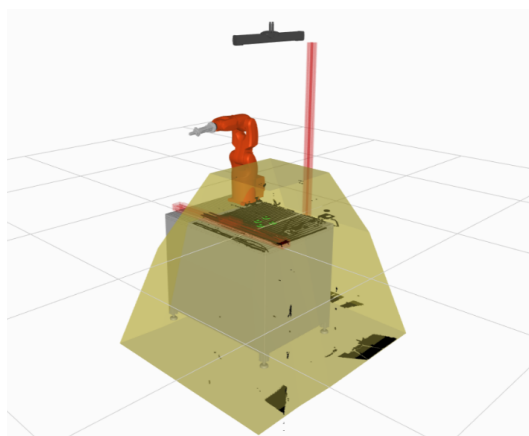


Obr. 3.32: Spustenie programu KUKA

### 3.3.3 Popis realizácie programovej úlohy

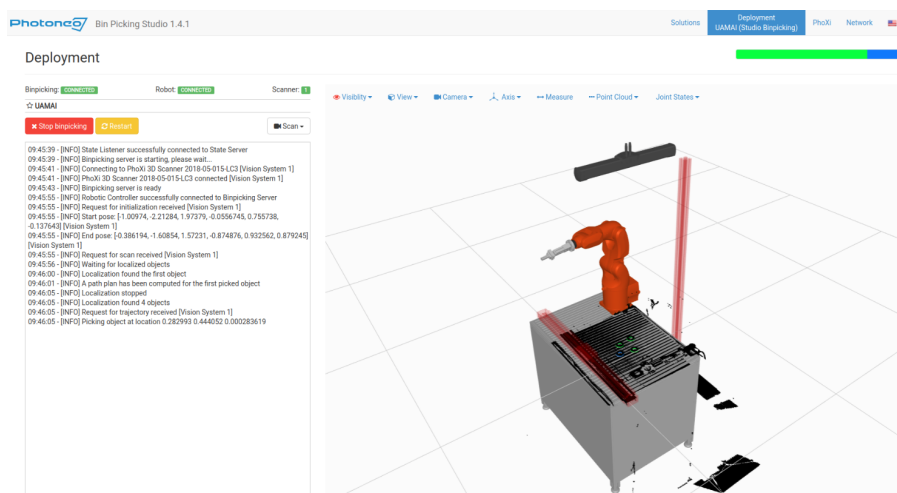
Pokiaľ máme pripravené užívateľské rozhranie Bin Picking Studia a zvolený program našej manipulačnej úlohy, spustíme program. Spustíme ho povolením pohybov jedným z tlačidiel na to určených a stlačením tlačidla štart, oba tieto tlačidlá sa nachádzajú na SmartPADe robotického systému KUKA KR3R540. Ako prvý krok sa spustí skener a zoskenuje všetky objekty nachádzajúce sa v jeho pracovnom priestore, viď obrázok 3.33. Na tomto obrázku sme si zvolili viditeľnosť pracovného priestoru robota, aby sme s určitosťou vedeli, že skener skenuje priestor, v ktorom sú umiestnené naše odoberané súčiastky. V ďalších krokoch toto zobrazenie pracovného priestoru už potrebovať nebudeme a tak jeho zobrazenie vypneme.

V stavovom riadku si môžeme všimnúť podrobnú komunikáciu jednotlivých zariadení aj s ich následnou reakciou, viď obrázok 3.34. Vidíme, že riadiaci systém robota odoslal informáciu riadiacemu systému Vision o štartovacej a koncovej pozícii manipulátora. Ďalej vidíme vyžiadanie skenu a proces čakania na lokalizovanie odoberaných objektov. Následne vidíme správu o nájdení prvého objektu, proces plánovania posunutí manipulátora pre potrebu odoberania objektu, zastavenie lokalizácie aj s následným počtom lokalizovaných súčiastok. V našom prípade ide o 4 úspešne lokalizované objekty. Po lokalizácii je zo strany Bin Picking Studia vyžiadaný výpočet trajektórie na odoberanie nájdeného prvého objektu s následnou odpoveďou riadiaceho systému Vision s danými súradnicami. Náj-



Obr. 3.33: Skenovanie pracovného priestoru

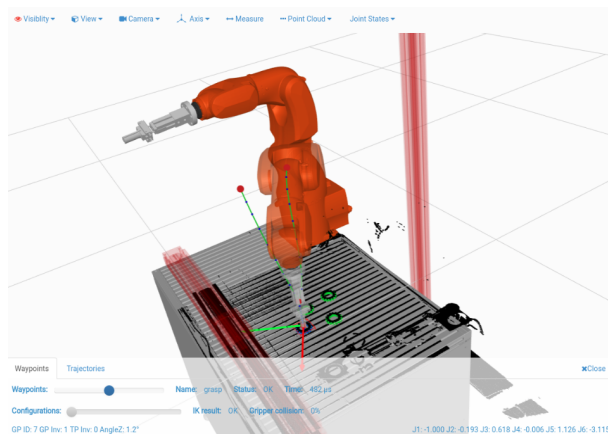
dené objekty sa zvýraznia zelenou farbou a následne sa jeden z nich zvýrazní na modro, čo znamená, že tento objekt bude odoberaný ako prvý v poradí.



Obr. 3.34: Nájdenie objektov skenerom

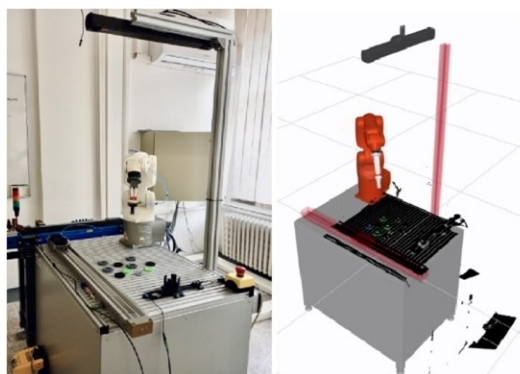
Ak sa chceme uistiť, že manipulátor dosiahne pozíciu, ktorú vypočítal Vision riadiaci systém, môžeme si ju overiť kliknutím na objekt a Bin Picking Studio nám poskytne simulčný náhľad na manipulátor v tejto pozícii, vid' obrázok 3.35. V tomto náhľade sa nám zobrazia aj medzibody odoberania, v ktorých si rovnako vieme simulačne overiť pozície. Ďalej tu môžeme vidieť meno procedúry, jej status a čas výpočtu, výsledok procedúry a kolíziu koncového efektora manipulátora vyjadrený v percentách.

Hneď po skenovaní a úspešnom lokalizovaní odoberanej súčiastky sa manipulátor začne hýbať podľa prednastavených fáz. Najprv sa manipulátor pohne do fázy štart, táto fáza je zobrazená na obrázku 3.36, ktorý sa skladá z dvoch samostatných obrázkov, a to snímky obrazovky zo softvérového prostredia Bin Picking Studia a reálnej fotografie manipulátora v rovnakej fáze. Ďalšie fázy budú zobrazované rovnako v jednom obrázku z



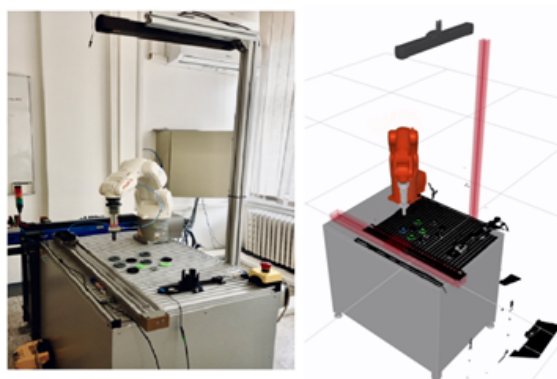
Obr. 3.35: Simulačný pohľad na manipulátor pri odoberaní objektu

dvoch pohľadov, reálneho a simulačného.



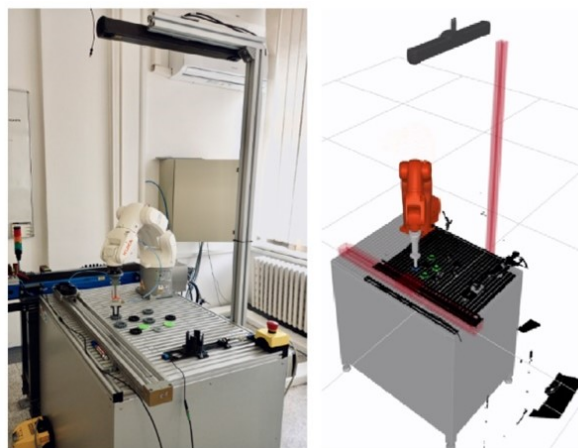
Obr. 3.36: Fáza štart

Po tejto fáze nasleduje fáza priblíženia, manipulátor sa priblíži nad odoberaný objekt s nami definovaným posuvom v osi, viď obrázok 3.37.



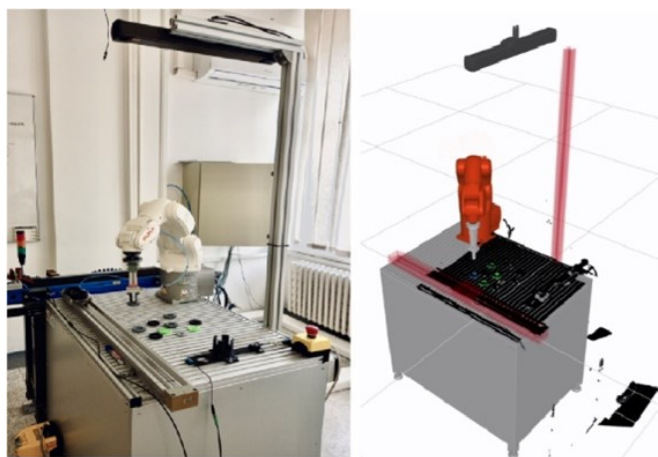
Obr. 3.37: Fáza priblíženia

Ak je manipulátor nad odoberanou súčiastkou, nasleduje fáza uchopenia. Manipulátor sa posunie lineárnym pohybom do pozície definovanej fázou uchopenia a uchopí objekt. V našom prípade sa roztvoria prsty koncového efektora manipulátora, vid' obrázok 3.38.



Obr. 3.38: Fáza uchopenia

Keď manipulátor uchopí súčiastku, môže sa s ňou lineárne vzdialiť, a teda dosiahnuť pozíciu definovanú fázou vzdialenia. Táto fáza je zobrazená na obrázku 3.39.



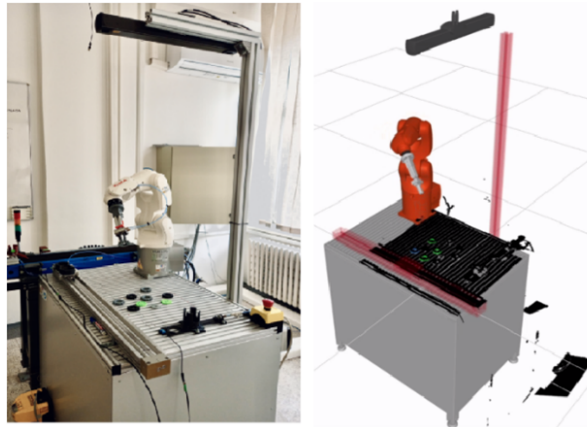
Obr. 3.39: Fáza vzdialenia

Po fáze vzdialenia prejde manipulátor do preddefinovanej pozície a dostane sa do fázy koniec, vid' obrázok 3.40.

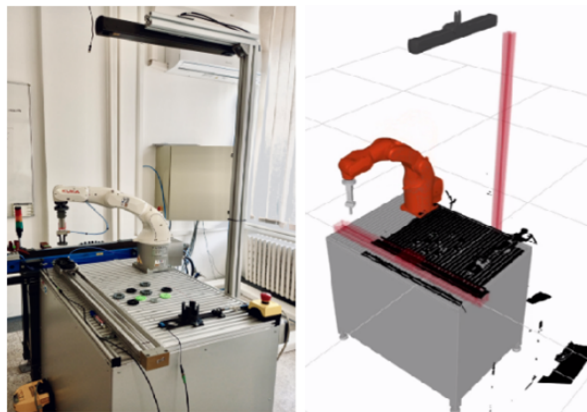
Predtým, ako sa bude opätovne odoberať súčiastka, treba ju uložiť do bloku motora. Manipulátor sa teda pohne do pozície na odkladanie súčiastky. Ako prvá pozícia je zvolená pozícia nad blokom motora, vid' obrázok 3.41.

V ďalšom bode sa manipulátor pohne lineárnym pohybom tesne nad blok motora a



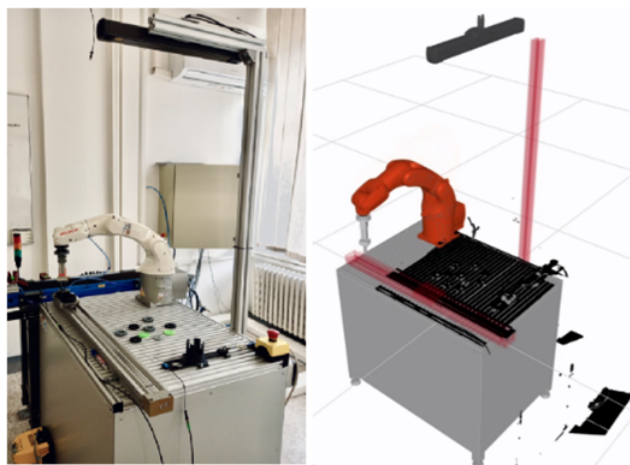


Obr. 3.40: Fáza koniec



Obr. 3.41: Pozícia odkladania súčiastky

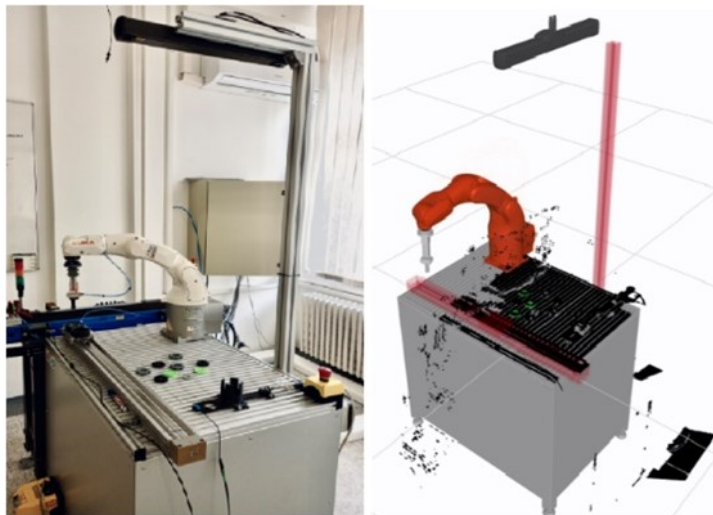
uloží súčiastku do bloku motora, vid' obrázok 3.42.



Obr. 3.42: Pozícia uloženia súčiastky

V poslednom bode sa manipulátor vzdiali lineárnym pohybom od uloženej súčiastky,

viď obrázok 3.43. Po tomto bode nasleduje opätovné skenovanie a čakanie na moment, kedy sa manipulátor opäť dostane do fázy štart.



Obr. 3.43: Vzdialenie od uloženej súčiastky

### 3.3.4 Výpis programu manipulačnej úlohy

Logika programu manipulačnej úlohy je pre jednoduchšie pochopenie graficky zobrazená na obrázku 3.44. Bod 1 znázorňuje začiatok manipulačnej úlohy, kedy sa robot nachádza vo svojej domácej pozícii.

Bod 2 znázorňuje skenovanie pracovného priestoru skenera. Pokiaľ skener nelokalizuje žiadny objekt na odoberanie, program sa vráti slučkou opäť na začiatok tohto bodu a nasleduje ďalší sken. Toto skenovanie v slučke pokračuje až pokým skener úspešne lokalizuje objekt na odoberanie a presunie sa na ďalší bod.

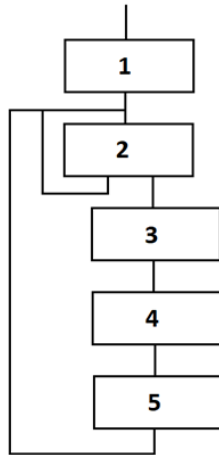
Bod 3 znázorňuje fázy riadené Bin Picking Studiom, teda fázy štart, priblíženie, uchopenie, vzdialenie a koniec. Po dosiahnutí pozície definovanej pre fázu koniec program pokračuje do ďalšieho bodu.

Bod 4 znázorňuje pohyb manipulátora definovaný ako odkladanie a uloženie súčiastky. Po úspešnom uložení súčiastky program pokračuje do bodu 5, ktorý znázorňuje vzdialenia manipulátora od uloženej súčiastky a vráti sa slučkou do bodu 2.

Program bude bežať v týchto slučkách až pokým operátor nezastaví program našej manipulačnej úlohy.

Program manipulačnej úlohy je napísaný v programovacom jazyku KRL. Je zložený z hlavného programu s názvom Scan\_ Application a podprogramu s názvom customer\_ definitions. Hlavný program obsahuje všetky funkcie skenovania a pohybov odoberania a uloženia odoberanej súčiastky. Podprogram customer\_ definitions obsahuje podprogramy pre zopínanie a rozopínanie koncového efektrota manipulátora. Program je podrobne vysvetlený v komentároch, väčšina použitých funkcií je detailnejšie rozpísaná v kapitole 3.1.3.





Obr. 3.44: Grafické zobrazenie behu programu

### Výpis hlavného programu Scan\_ Application:

```

DEF Scan_ Application( ) ; definovanie programu Scan_ Application

INI ; inicializácia premenných, začiatok manipulačnej úlohy

BOOL trajectory_ ok, scan_ ok
E6AXIS temp
AXIS start_ joint_ pos, end_ joint_ pos
INT conversion_ status, scan_ status, trajectory_ status

    PTP HOME Vel=100% ; PTP pohyb do domovskej pozície manipulátora

    LOOP ; začiatok slučky skenovania

        ; pokračovanie do podmienky IF, ak riadiaci systém manipulátora je pripojeny k riadiacemu systému Vision, spustí sa prvý sken
        CONTINUE
        IF $FLAG[101] THEN

            PTP HOME Vel=100% ; PTP pohyb do domovskej pozície manipulátora

            PHO_ BinpickingSettings() ; systémová konfigurácia nastavení Binpickingu

            PHO_ ConnectToVc() ; funkcia na vytvorenie pripojenia k riadiacemu systému Vision

            Convert Start Pose to AXIS representation ; konvertovanie štartovacej pozície, nastavenie hodnôt manipulátora do riadiaceho systému Vision
            temp = A1 0, A2 0, A3 0, A4 0, A5 0, A6 0
            conversion_ status = 0
            IF FSTART.TOOL_ NO == 0 THEN
  
```

```

$TOOL = $NULLFRAME
ELSE
$TOOL = TOOL_ DATA[FSTART.TOOL_ NO]

    IF FSTART.BASE_ NO == 0 THEN
$BASE = $NULLFRAME
ELSE
$BASE = BASE_ DATA[FSTART.BASE_ NO]
ENDIF
start_ joint_ pos = INVERSE(XSTART, temp, conversion_ status)

    Convert End Pose to AXIS representation
; konvertovanie koncovej pozície, nastavenie hodnôt manipulátora do riadiaceho systému
Vision
temp = A1 0, A2 0, A3 0, A4 0, A5 0, A6 0
conversion_ status = 0
IF FEND.TOOL_ NO == 0 THEN
$TOOL = $NULLFRAME
ELSE
$TOOL = TOOL_ DATA[FEND.TOOL_ NO]
ENDIF
IF FEND.BASE_ NO == 0 THEN
$BASE = $NULLFRAME
ELSE
$BASE = BASE_ DATA[FEND.BASE_ NO]
ENDIF
end_ joint_ pos = INVERSE(XEND, temp, conversion_ status)

    PHO_ RequestInit(start_ joint_ pos, end_ joint_ pos) ; požiadavka na inicializáciu
aplikácie bin picking na strane Vision riadiaceho systému

    PHO_ RequestScan() ; požiadavka na spustenie skenovania

    WAIT SEC 10 ; 10-sekundové čakanie na inicializáciu skenera a manipulátora

    ENDIF ; ukončenie podmienky IF

    trajectory_ ok = FALSE ; resetovanie príznaku trajectory_ ok

    ; Skenovanie a plánovanie trajektórie
    WHILE trajectory_ ok == FALSE ; začiatok podmienky WHILE, pokiaľ trajectory_ ok
    bude v stave FALSE pokračuj ďalej

    scan_ status = PHO_ WaitForScan() ; nastavenie statusu skenovania na funkciu ča-
kania na ukončenie skenovania

```

IF (scan\_status == PHO\_OK) THEN ; začiatok podmienky IF, ak je status skenovania v stave OK tak potom pokračuj ďalej

PHO\_RequestTrajectory() ; žiadosť na trajektóriu z bin pickingu na odoslanie do riadiaceho systému manipulátora

trajectory\_status = PHO\_ReceiveTrajectory() ; nastavenie statusu trajektórie na funkciu na príjem požadovaných trajektórií bin pickingu

SWITCH trajectory\_status ; začiatok podmienky SWITCH pre chybové hlásenia statusu trajektórie

CASE PHO\_NOT\_INITIALIZED ; ak je status v stave nesprávnej inicializácie, tak potom

PHO\_RequestInit(start\_joint\_pos, end\_joint\_pos) ; poslanie požiadavky na inicializáciu riadiaceho systému Vision

PHO\_RequestScan() ; poslanie požiadavky na skenovanie

WAIT SEC 10 ; 10-sekundové čakanie na inicializáciu skenera a manipulátora

CASE PHO\_SERVICE\_ERR ; ak je status v stave neplatnej odozvy riadiaceho systému Vision, tak potom

PHO\_RequestInit(start\_joint\_pos, end\_joint\_pos) ; poslanie požiadavky na inicializáciu riadiaceho systému Vision

PHO\_RequestScan() ; poslanie požiadavky na skenovanie

WAIT SEC 10 ; 10-sekundové čakanie na inicializáciu skenera a manipulátora

CASE PHO\_BAD\_DATA ; ak je status v stave zlyhania kontroly overenia údajov, tak potom

PHO\_RequestInit(start\_joint\_pos, end\_joint\_pos) ; poslanie požiadavky na inicializáciu riadiaceho systému Vision

PHO\_RequestScan() ; poslanie požiadavky na skenovanie

WAIT SEC 10 ; 10-sekundové čakanie na inicializáciu skenera a manipulátora

CASE PHO\_PLANNING\_FAILED ; ak je status v stave zlyhania plánovania trajektórie, tak potom

PHO\_ RequestScan() ; poslanie požiadavky na skenovanie

CASE PHO\_ NO\_ PART\_ FOUND ; ak je status v stave nelokalizovaného objektu, tak potom

PHO\_ RequestScan() ; poslanie požiadavky na skenovanie

CASE PHO\_ OK ; ak je status v stave OK, tak potom

trajectory\_ ok = TRUE ; zapnutie príznaku trajectory\_ ok do statusu TRUE

DEFAULT ; predvolené nastavenie slučky pri nešpecifikovanej chybe zlyhania trajektórie

LOOP

msgNotify(UNKNOWN TRAJECTORY ERROR: %1", "BP\_ CLIENT", trajectory\_ status)

ENDLOOP

ENDSWITCH ; koniec podmienky SWITCH pre status trajektórie

ELSE ; pokračovanie podmienky IF

PHO\_ RequestInit(start\_ joint\_ pos, end\_ joint\_ pos) ; poslanie požiadavky na inicializáciu riadiaceho systému Vision

PHO\_ RequestScan() ; poslanie požiadavky na skenovanie

WAIT SEC 10 ; 10-sekundové čakanie na inicializáciu skenera a manipulátora

ENDIF ; koniec podmienky IF

ENDWHILE ; koniec podmienky WHILE

PTP START CONT Vel=30 % PDAT12 Tool[1] Base[0]

; pohyb manipulátora do štartovacej pozície

PHO\_ PickPart ( ) ; funkcia na vykonanie štandardnej sekvencie vyberania z koša

PTP END CONT Vel=30 % PDAT9 Tool[1] Base[0] ; pohyb manipulátora do koncovej pozície

TRIGGER WHEN DISTANCE = 0 DELAY = 0 DO PHO\_ RequestScan() PRIO = 99 ; preddefinované spustenie ďalšieho skenovania

; ukladanie súčiastky

PTP P1 CONT Vel=40 % PDAT1 Tool[1] Base[0] ; pohyb robota do pozície odkladania súčiastky

PTP P2 CONT Vel=40 % PDAT3 Tool[1] Base[0] ; pohyb robota do pozície uloženia súčiastky

PHO\_ GripperUser\_ 1 ( ) ; funkcia rozopnutia koncového efektora manipulátora

WAIT Time=1 ; 1-sekundové čakanie na vykonanie funkcie rozopnutia koncového efektora manipulátora

PTP P3 CONT Vel=40 % PDAT2 Tool[1] Base[0] ; pohyb robota do pozície vzdialenia od uloženej súčiastky

ENDLOOP ; ukončenie slučky

END ; koniec programu Scan\_ Application

### **Výpis podprogramu customer\_ definitions:**

DEF customer\_ definitions( ) ; definovanie programu customer\_ definitions

INI ; inicializácia programu

GLOBAL DEF PHO\_ GripperAttach() ; definovanie podprogramu zopnutia koncového efektora manipulátora

\$OUT[30] = TRUE  
END ; koniec podprogramu

GLOBAL DEF PHO\_ GripperUser\_ 1() definovanie podprogramu rozopnutia koncového efektora manipulátora

\$OUT[30] = FALSE  
END ; koniec podprogramu

## 4 Záver

Cieľom diplomovej práce bolo získať a aplikovať vedomosti v problematike programovania priemyselných robotov s využitím strojového videnia. Mojou motiváciou pre vytvorenie tejto práce bolo nadviazať na moju bakalársku prácu s názvom Programovanie robotizovaných pracovísk s priemyselným robotom KUKA KR3R540. Po vytvorení mojej diplomovej práce sa robotizované pracovisko na našom ústave stalo omnoho komplexnejším a vytvára tak lepší praktický príklad pre študentov, ktorí sa budú zaoberať témou robotiky, strojového videnia s najmodernejšími technológiami alebo budú mať túto problematiku v rámci študijného plánu.

V úvode práce som teoreticky opísal technológie a aplikácie použitia 2D a 3D systémov strojového videnia s ich následným porovnaním.

V praktickej časti návrhu využitia počítačového videnia v rámci priemyselného využitia vkladania a odoberania súčiastok som popísal návrh aplikácie a implementácie počítačového videnia v robotizovanom pracovisku. V tomto návrhu sú popísané použité zariadenia, ich vzájomná komunikácia a návrh rozmiestnenia týchto zariadení na robotizovanom pracovisku.

Posledným bodom práce som opísal realizáciu počítačového videnia na robotizovanom pracovisku. V tomto bode som popísal konfiguráciu manipulátora na použité softvérové prostredie a taktiež konfiguráciu tohto softvérového prostredia. Na záver tohto bodu je popísané spustenie programov potrebných na realizáciu robotizovaného pracoviska s využitím strojového videnia a popis programovej úlohy.

Po získaní potrebných vedomostí a informácií som v ďalšom bode navrhol program pre manipulačnú úlohu s použitím dostupných periférií. Pri tvorbe vizualizácie robotizovaného pracoviska som vychádzal z reálnych periférií, ktoré sú následne ovládané pomocou robota. Popis a vysvetlenie ovládania periférií je popísaný v jednotlivých bodoch podľa priebehu programu.

# Literatúra

- [1] Zivid, Why 3D machine vision? What's wrong with 2D machine vision? 2020. [Online]. Dostupné: <https://blog.zivid.com/why-3d-machine-vision-whats-wrong-with-2d-machine-vision>. [Cit. 28.05.2021]
- [2] DATALOGIC, Machine Vision REFERENCE GUIDE. 2021. [Online]. Dostupné: [https://www.datalogic.com/upload/Flipbook/RG\\_Machine|20Vision\\_en/files/assets/common/downloads/RG-MachineVision-EN.pdf](https://www.datalogic.com/upload/Flipbook/RG_Machine|20Vision_en/files/assets/common/downloads/RG-MachineVision-EN.pdf). [Cit.28.05.2021]
- [3] DAILY AUTOMATION, Kamerové systémy. 2021. [Online]. Dostupné: [https://www.dailyautomation.sk/tvoje\\_vedomosti/senzory/kamerove-systemy/](https://www.dailyautomation.sk/tvoje_vedomosti/senzory/kamerove-systemy/). [Cit. 28.05.2021]
- [4] VISION ONLINE, Emerging 3D Vision Technologies for Robot Vision & Machine Vision. 2021. [Online]. Dostupné:<https://www.visiononline.org/blog-article.cfm/Emerging-3D-Vision-Technologies-for-Robot-Vision-Machine-Vision/194>. [Cit. 28.05.2021]
- [5] Cognex Corporation, 3D VISION SYSTEMS. 2021. [Online]. Dostupné: <https://www.cognex.com/what-is/machine-vision/system-types/3d-systems>. [Cit. 28.05.2021]
- [6] Cognex Corporation, DataMan 150/260 Series Features. 2021. [Online]. Dostupné: <https://www.cognex.com/products/barcode-readers/fixed-mount-barcode-readers/dataman-150-260-series>. [Cit. 28.05.2021]
- [7] SICK AG, 3D machine vision TriSpectorP1000 / TriSpectorP1008. 2021. [Online]. Dostupné: <https://www.sick.com/>. [Cit. 28.05.2021]
- [8] Photoneo s.r.o., PhoXi 3D Scanner. 2021. [Online]. Dostupné: <https://www.photoneo.com/phoxi-3d-scanner/>. [Cit. 28.05.2021]
- [9] Photoneo s.r.o., PhoXi 3D Scanner L. 2021. [Online]. Dostupné: <https://www.photoneo.com/products/phoxi-scan-l/>. [Cit. 28.05.2021]
- [10] Cognex Corporation, 3D-A5000 Series Area Scan 3D Camera. 2021. [Online]. Dostupné: <https://www.cognex.com/products/machine-vision/3d-machine-vision-systems/3d-a5000-series-area-scan>. [Cit. 28.05.2021]
- [11] SICK AG, Bezpečnostní laserové skenery. 2016. [Online]. Dostupné: [https://cdn.sick.com/media/docs/6/56/356/Operating\\_instructions\\_S3000\\_Safety\\_Laser\\_scanner\\_cs\\_IM0067356.PDF](https://cdn.sick.com/media/docs/6/56/356/Operating_instructions_S3000_Safety_Laser_scanner_cs_IM0067356.PDF). [Cit. 28.05.2021]

- [12] KUKA AG. Kr 3 agilus. 2021. [Online]. Dostupné: <https://www.kuka.com/en-gb/products/robotics-systems/industrial-robots/kr-3-agilus>. [Cit. 28.05.2021]
- [13] Neosys Technology Inc. 2017. [Online]. Dostupné: [https://omtec.de/media/pdf/62/06/74/User-Manual\\_Neosys\\_Nuvo-5095GC-series.pdf](https://omtec.de/media/pdf/62/06/74/User-Manual_Neosys_Nuvo-5095GC-series.pdf). [Cit. 28.05.2021]
- [14] Digi-Key Electronics. 2021. [Online]. Dostupné: <https://www.digikey.at/product-detail/en/kuka-robotics-corporation/KR-3-R540/2122-KR3R540-ND/10239959> [Cit. 28.05.2021]
- [15] ROBOTER, K. U. K. A. KUKA System Software 8.3, Operating and Programming Instructions for System Integrators. 2015. [Cit.28.05.2021]
- [16] Low Profile Air Gripper. Series MHF2. 2021 [Online]. Dostupné: <https://www.smc pneumatics.com/pdfs/MHF.pdf> [Cit.28.05.2021]
- [17] ROBOTER, K. U. K. A. KUKA System Software KR C..., Expert Programming. 2004. [Online]. Dostupné: <http://img53.hbzhan.com/2/20171205/636480698963759998207.pdf>. [Cit. 28.05.2021]
- [18] VACHÁLEK, Ján; TAKÁCS, Gergely. 2014. Robotika. Bratislava : STU, 2014. 166 s. ISBN 978-80-227-4163-7 [Cit. 28.05.2021]
- [19] KUKA. KR 3 R540. 2021. [Online]. Dostupné: [https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000270971\\_en.pdf?rev=bc7b2ed653874310b20f46a357ce7730&hash=FD87C1547FE6EFCAC7811C2D849C2C35](https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000270971_en.pdf?rev=bc7b2ed653874310b20f46a357ce7730&hash=FD87C1547FE6EFCAC7811C2D849C2C35) [Cit. 28.05.2021]
- [20] KUKA AG. KR 3 AGILUS. 2021. [Online]. Dostupné: <https://www.kuka.com/cs-cz/produkty>. [Cit.28.05.2021]
- [21] Bin Picking Studio. Introduction and Training version 1.4. 2021. [Online]. Dostupné: <https://www.photoneo.com/>. [Cit.28.05.2021]
- [22] Bin Picking Tutorial. Setting up Photoneo Bin Picking Studio with KUKA robots. 2021. [Online]. Dostupné: <https://www.photoneo.com/>. [Cit.28.05.2021]
- [23] Robot Camera Calibration. User Manual and Installation Instructions. 2018. [Online]. Dostupné: [https://photoneo.com/files/manuals/Robot\\_Camera\\_Calibration\\_User\\_Manual\\_EN\\_v2.pdf](https://photoneo.com/files/manuals/Robot_Camera_Calibration_User_Manual_EN_v2.pdf). [Cit.28.05.2021]