

Teemu Pesonen

GoFa™-yhteistyörobotin ohjelmointi

Insinööri (AMK)

Konetekniikka

Kevät 2024



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä: Pesonen Teemu

Työn nimi: GoFa™-yhteistyörobotin ohjelmointi

Tutkintonimike: Insinööri (AMK), konetekniikka

Asiasanat: Robotiikka, ohjelmointi, yhteistyörobotti, cobotti, integrointi, konenäkö

Opinnäytetyössä keskityttiin ABB:n GoFa-yhteistyörobotin ohjelmointiin opetuskäyttöön soveltuvassa ympäristössä. Pää tavoitteena ja työn ideana oli luoda Kajaanin ammattikorkeakouluun hankitulle yhteistyörobotille valmiiksi ohjelmoitu pohja kappaleenkäsittelyä ja -tunnistusta varten.

Robottisolun suunnittelu sai alkunsa ammattikorkeakoululla robotiikan suuntaavien opintojen aikana. Kesällä suoritetun työharjoittelujakson aikana robottisolun 3D-mallia ja simulaatiota kokoonpanosta täydennettiin, joiden perusteella myöhemmin vastaanotettu fyysinen robottisolun koottiin. Harjoittelun loppuvaiheessa työskentely jatkui keskeneräisen robottisolun parissa opinnäytetyön merkeissä.

Robottisolun käyttöönoton seuraava vaihe oli ohjelmointi, josta opinnäytetyön aihe sai alkunsa. Tämän lisäksi työ mahdollisti tutustumisen graafiseen Easy Wizard -ohjelmointityökaluun ja Skill Creator -lisäosaan. Työssä verrattiin niiden ominaisuuksia perinteiseen tietokoneella suoritettavaan RAPID-ohjelmointimenetelmään sekä yhteenvedossa tuotiin esiin molempien ohjelmointikeinojen edut ja haitat.

Ohjelmassa hyödynnettiin konetekniikan opintojen aikana opittuja menetelmiä robottien ohjelmoinnin sekä konenäön kursseilta. Ohjelmaan sisältyi perinteistä kappaleen poimintaa koordinaattipisteitä hyödyntäen sekä kappaleen tunnistusta konenäön avulla. Työssä huomioitiin kohderyhmä ja käyttötarkoitus, joka työssä tarkoittaa ammattikorkeakoulun opetuskäyttöön soveltuvuutta. Pohjaa ja koordinaatistoa hyödyntämällä on mahdollista luoda konetekniikan opiskelijaharjoitukseen soveltuva kappaleenkäsittelytyö, jossa opiskelijat pääsevät tutustumaan modernin yhteistyörobotin toimintaan ja käyttöön.

Robottisolulle jäi ohjelmoinnin jälkeen mahdollisuuksia jatkokehitykselle ohjelmoinnin ja konenäön parissa. Soluun lisättäviä komponentteja olisi muun muassa turvaskanneri ja sen määrittäminen sekä HMI-paneelin asennus. Ohjelmaan olisi mahdollista ohjelmoida lisäksi robotin työkalun vaihto ja konenäön opettaminen tunnistamaan erisävyisiä kappaleita tarkemmin, jotka molemmat sopisivat projektiaiheiksi robotiikan suuntaaviin opintoihin.

Abstract

Author: Pesonen Teemu

Title of the Publication: Programming of the GoFa™ Collaborative Robot

Degree Title: Bachelor of Engineering, Mechanical Engineering

Keywords: Robotics, programming, collaborative robot, cobot, integration, machine vision

The aim of the thesis is to study the programming of ABB's GoFa collaborative robot in an educational environment and use. The primary goal and the idea were to create a pre-programmed foundation for workpiece handling and recognition for the collaborative robot acquired by Kajaani University of Applied Sciences.

The design of the robot cell began at the Kajaani UAS during the advanced studies in robotics. During the summer internship at the Kajaani UAS, an unfinished 3D model and simulation of the robot cell was completed by the thesis author. Receiving and assembling the robot cell was partaken. Towards the end of the internship, working on the incomplete robot cell continued.

The next step in setting up the robot cell was programming, from which the topic of the thesis originated from. Additionally, the Easy Wizard graphical programming tool was studied as well as the Skill Creator add-on. In the thesis, their features were compared to the more traditional RAPID programming method performed on a computer, and the advantages and disadvantages of both programming methods were highlighted in the summary.

The program utilized methods learned during the mechanical engineering studies from robot programming and machine vision courses. The program included traditional workpiece handling using coordinate points and object recognition through machine vision. The thesis took into account the target audience and the intended use, which meant its suitability for educational use. By utilizing the base and the coordinate system, a workpiece handling program suitable for mechanical engineering studies can be created, allowing students to learn and familiarize themselves with the operation and use of a modern collaborative robot.

After the programming, there were opportunities for further robot cell development in both programming and machine vision. Additional components to be added to the cell include a safety scanner and its configuration, as well as the installation of an HMI panel. Furthermore, it would be possible to include a tool change feature within the program and to teach the machine vision to recognize workpieces of different shades, both of which would be suitable as project topics for the advanced studies in robotics.

Sisällys

1	Johdanto	1
2	Yhteistyörobotiikka ja cobotit	2
3	Ohjelmistot	4
3.1	RobotStudio.....	4
3.2	Wizard Easy Programming & Skill Creator	6
3.3	Zimmer HMI.....	7
3.4	Comfort App	9
3.5	SolidWorks.....	9
4	GoFa™	10
5	Suunnittelu ja kokoonpano	11
5.1	Solun komponentit.....	11
5.2	3D-malli	15
5.3	Simulaatio.....	16
5.4	Konenäkö.....	17
6	Ohjelmointi	19
6.1	Koordinaatistot.....	19
6.2	Työkalumäärittäminen	21
6.3	RAPID.....	22
6.3.1	Laakeri ja akseli	22
6.3.2	Kannen kappaleentunnistus	26
6.4	Easy Wizard	27
7	Yhteenveto	28
	Lähteet	29

Termit ja lyhenteet

SCM = Smart Communication Module, viestintämoduuli

HMI = Human Machine Interface, ihmisen ja laitteen välinen käyttöliittymä

TCP = Tool Center Point, työkalun keskipiste

ASI = Arm-side-interface, käsivarren käyttöliittymä

FlexPendant = Robotin käsiohjain

FlexFeeder = Kappaleensyöttölaite

1 Johdanto

Opinnäytetyöni aiheena on ohjelmoida ABB:n GoFa-yhteistyörobotille kappaleenkäsittelyohjelma, jota voidaan hyödyntää tulevaisuudessa robotiikan opinnoissa ja harjoituksissa. Ehdotuksia opinnäytetyön aiheeseen kävin läpi jo työharjoittelujakson aikana, jonka suoritin Kajaanin ammattikorkeakoululla kesällä 2023. Olin tietoinen koululle hankitusta GoFa-yhteistyörobotista jo alkuvuodesta 2023, jolloin suunnittelimme GoFan solua integroinnin kurssilla osana robotiikan syventäviä opintoja.

Minulle ohjelmointipohjainen aihe oli tärkeä robotiikan kannalta, sillä toivoin jatkavani opintoja ja työskentelyä robotiikkaan liittyen syventävien opintojen jälkeenkin. Ohjelmointi toi myös itselleni realistiset tavoitteet ja koin, että kaipasin enemmän käytännön kokemusta siihen liittyen. Graafisen Wizard-ohjelmointinäkökulman ja perinteisen RAPID-ohjelmointimenetelmän yhdistäminen sekä niiden vertaaminen keskenään oli sopivan haasteellista ja itselleni mielekästä. Vertailulla pystyin selvittämään, onko graafinen työkalu tehnyt ohjelmoinnista käyttäjäystävällisempää ja helpompaa.

2 Yhteistyörobotiikka ja cobotit

Yhteistyörobotiikka, jota voidaan kutsua myös kollaboratiiviseksi robotiikaksi, viittaa robotiikan alueeseen, jossa robotit työskentelevät ihmisten kanssa tai heidän läheisyydessänsä tietyissä ympäristöissä. [1.] Yhteistyörobotit näin ollen eroavat perinteisemmistä teollisuusroboteista, jotka usein toimivat ilman suoraa vuorovaikutusta ihmisten kanssa.

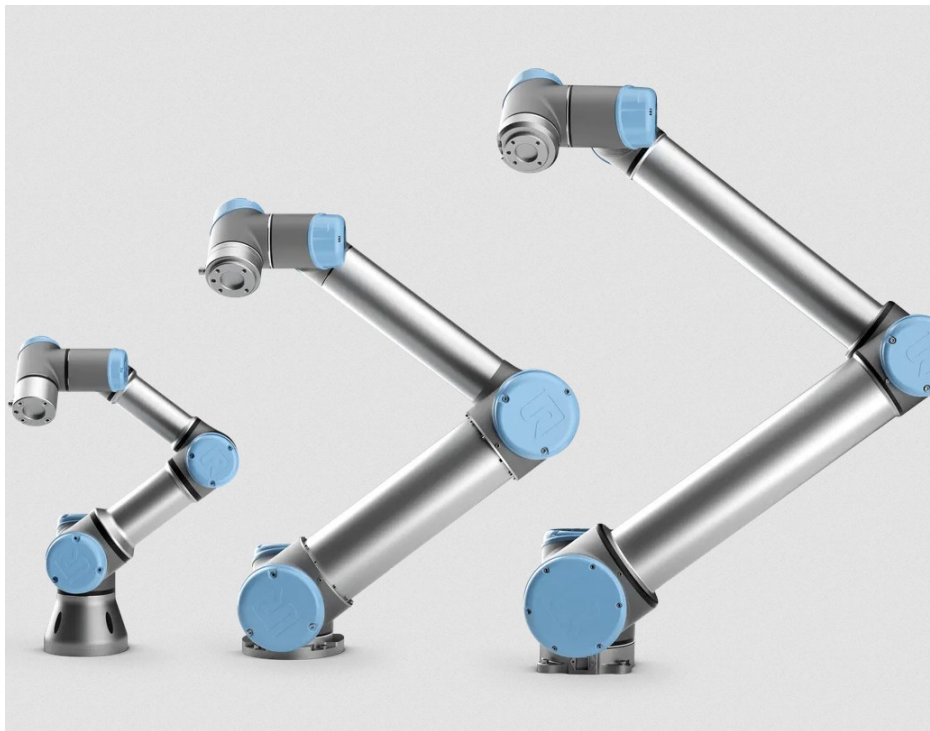
Yhteistyörobotiikan tavoitteena on luoda turvallinen ja tehokas yhteistyö ihmisten ja robottien välille eri sovelluksissa, kuten esimerkiksi terveydenhuollossa, logistiikassa ja eri palvelualoissa. Yleisesti tarkoituksena on parantaa työturvallisuutta, tuottavuutta ja joustavuutta monenlaisissa eri tehtävissä. Tyypillisesti yhteistyörobotit suunnitellaan niin, että ne pystyvät toimimaan käyttäjän läheisyydessä ilman, että ne aiheuttaisivat vaaraa ihmiselle. Yhteistyöroboteissa, eli coboteissa voi olla muun muassa pyöristetyt kulmat, pehmeämmät pinnat ja törmäyksen tunnistavat anturit estämässä vahinkotilanteita.

Ensimmäiset cobotit keksittiin vuonna 1996, jolloin cobottia kutsuttiin ”laitteeksi ja menetelmäksi suoraa fyysistä vuorovaikutusta varten ihmisen ja tietokoneohjatun manipulaattorin välillä.” [2.] Teollisuusrobottien ja yhteistyörobottien eroavaisuudet näkyvät erityisesti turvaominaisuuksissa, joustavuudessa ja mukautuvuudessa, käyttäjäystävällisyydessä ja ohjelmoinnissa, sekä kustannuksissa. Cobotit ovat nykyisin tasaantuneet keskisuurille ja pienille valmistajille, jotka hyötyvät yhteistyöautomaation tuomasta tekniikasta pienivolyymillisissa tuotannoissa, kun taas perinteiset teollisuusrobotit ovat usein kalliita ja monimutkaisia laitteita suuria ja muuttumattomia tuotantoprosesseja varten.

Monille roboteille on varta vasten suunniteltu turvahäkki tai muulla tavalla estetty työympäristö. Tällä tavoin robottien nopeus, teho tai koko eivät aiheuta sen käyttäjälle riskitilanteita. Siksi teollisuudessa käytettyjen robottien pääpaino on niiden tehokkuudessa ja tarkkuudessa tietyissä tehtävissä, kun taas pienempikokoisten ja kevyempien cobottien siirto ja integrointi eri työtehtäviin on helpompaa ja käyttäjäystävällisempää.

International Federation of Robotics, IFR, vastaa robotiikkatilastoista, joiden mukaan maailmanlaajuisesti uusia teollisuusrobotteja otettiin käyttöön 553 052 kappaletta vuonna 2022. [3.] Suomessa teollisuusrobotteja otettiin käyttöön 631 kappaletta. Vaikka cobottien käyttöönottoja ei erikseen tilastoida, niiden osuus kokonaisinvestoinneista on noin 10 %, mikä tarkoittaisi noin 60 cobotin käyttöönottoa Suomessa.

Robottivalmistajista suurimpia cobottien valmistajia ovat Universal Robots, FANUC, ABB ja KUKA. Kolmen suurimman valmistajan markkinaosuus yhteistyöroboteista on noin 74 %. [4.] Valmistajista suurin tanskalainen UR on julkaissut kaksi eri yhteistyörobottisarjaa, joissa on yhteensä 9 erilaista cobottia. UR-yhteistyörobottien nimeäminen perustuu niiden kantokykyyn. Vuonna 2015 julkaistu CB3 -sarja sisältää cobotit UR3, UR5 ja UR10 (kuva 1).



Kuva 1. Universal Robots CB3 -sarja. [5.]

Uudempi vuonna 2018 julkaistu E-sarja sisältää cobotit UR3e, UR5e, UR10e, UR16e, UR20 ja UR30. [6.] E-sarja eroaa vanhemmasta CB3-sarjasta sisältämällä päivitetyn käyttöliittymän, sekä sisäänrakennetun voima-momenttianturin.

3 Ohjelmistot

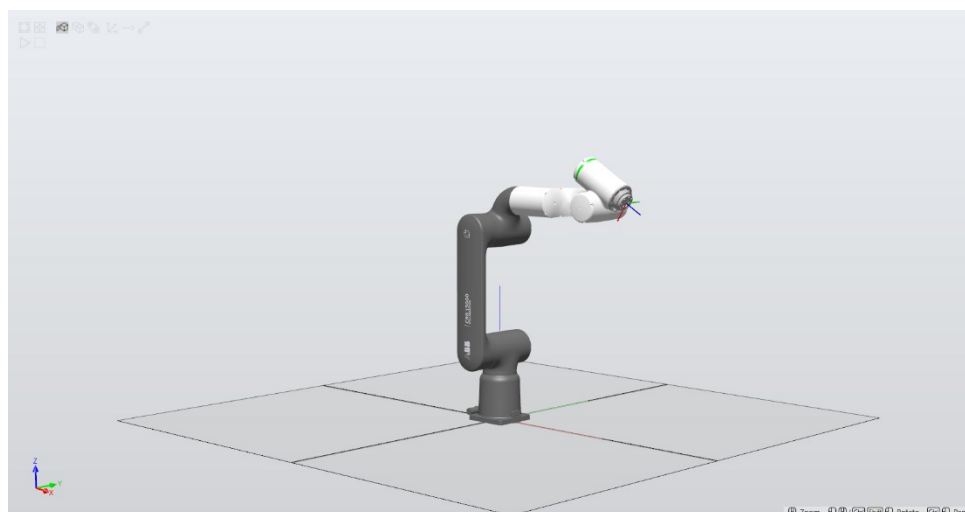
Robottiikassa simulointiohjelma tarkoittaa tietokoneohjelmaa, joka mahdollistaa robotiikkajärjestelmien ja kokoonpanojen simuloinnin digitaalisessa ympäristössä. Ohjelmat mahdollistavat robottien mallintamisen ja testauksen ennen niiden toteuttamista tai käyttöönottoa fyysisessä ympäristössä. [7.]

Simulointiohjelmillä hallitaan robotin liikkeiden simulointia, toimintoja ja vuorovaikutusta ympäristönsä kanssa. Simulointi auttaa myös havainnollistamaan, miten robotin ohjausjärjestelmä reagoi erilaisiin skenaarioihin ja mahdollistaa esimerkiksi virheiden korjaamisen tai työympäristön tilan havainnollistamisen.

Robottia voidaan ohjelmoida johdattamalla, jolloin ihmisvoimin tehtyä ja tallennettua liikerataa toistetaan ohjelmassa. Robotille voidaan myös opettaa paikoituspisteitä, joiden kautta se muodostaa ohjelmaan liikeradan.

3.1 RobotStudio

RobotStudio on ABB:n kehittämä suunnittelu-, ohjelmointi- ja simulointiohjelmisto. RobotStudion avulla käyttäjät voivat virtuaalisesti suunnitella ja simuloida ABB:n robotteja ennen niiden käyttöönottoa fyysisessä tuotantoympäristössä (kuva 2).



Kuva 2. CRB 15 000 RobotStudiassa.

ABB:n ohjelmointikielenä toimii RAPID, jolla korvattiin aikaisempi ARLA-ohjelmointikieli vuonna 1994. RAPID-kieli on hyvin C-ohjelmointikielen tyylinen, jossa on useita upotettuja robotin toimintoja. [8.] Toiminnot perustuvat pää- ja alirutiineihin. Alirutiinit voidaan jakaa useampiin eri moduuleihin, joita on helppo suorittaa kutsumalla esimerkiksi päärutiinin kautta.

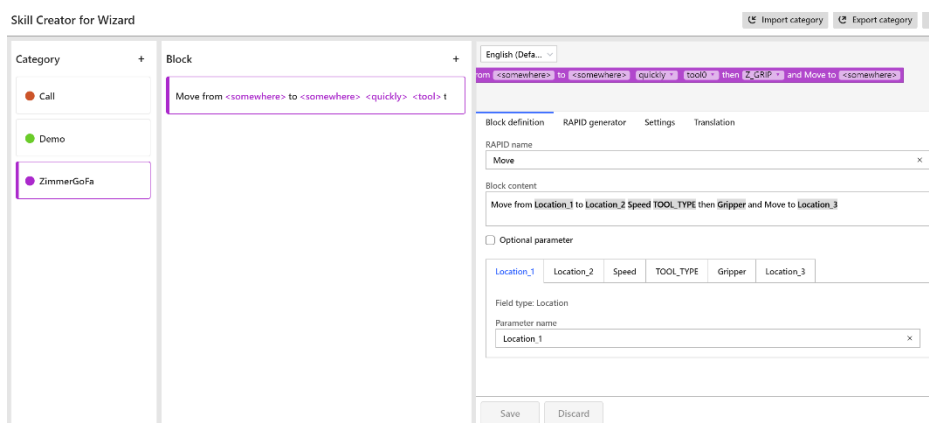
RobotStudio mahdollistaa myös I/O-kytkentöjen luomisen ja turvarajojen määrittämisen robottisoluun. RobotStudion lisäosana toimiva SafeMove mahdollistaa robotin turvallisen käytön ohjelman aikana hallitsemalla robotin nopeutta, pysähtymistä sekä rajaamalla sallittua työtilaa.

SafeMoven määrittäminen voidaan suorittaa RobotStudion tai käsiohjaimen kautta graafisen näkymän avulla, jossa robotin työalueeksi asetetaan nopeusrajoitettu vyöhyke. Tämän lisäksi robotin törmäysriskiä voidaan vähentää kapseloimalla robotin osat.

3.2 Wizard Easy Programming & Skill Creator

Wizard Easy Programming on ABB:n luoma ohjelmointityökalu yhteistyö- ja teollisuusrobotteja varten, jonka toiminta perustuu valmiiksi luotuihin RAPID-komentoihin. Työkalun tavoitteena on helpottaa ohjelmointia käyttäen graafisia ja helposti ymmärrettäviä lohkoja, jotka sisältävät erilaisia komentoja robotin ohjaamista ja signaalien määrittystä varten. Lohkot voidaan valita ja liittää valintapuusta ja siirtää ohjelmanäkymään kasaamalla ne järjestyksessä päällekkäin. Komentoja voidaan tarkastella tarvittaessa myös RobotStudiassa, jossa valmiit lohkot näkyvät komentolauseina.

Mikäli ohjelmassa vaaditaan mukautettuja ohjelmointikomentoja, voidaan Skill Creator -lisäosalla luoda tarvittaessa omia komentolohkoja (kuva 3). Uuden mukautetun lohkon luominen on viisivaiheinen prosessi, joka alkaa valitsemalla lohkolle kategoria. Kategoriassa määritetään, onko komento esimerkiksi robottia vai robotin työkalua varten.



Kuva 3. Skill Creator -lisäosan näkymä.

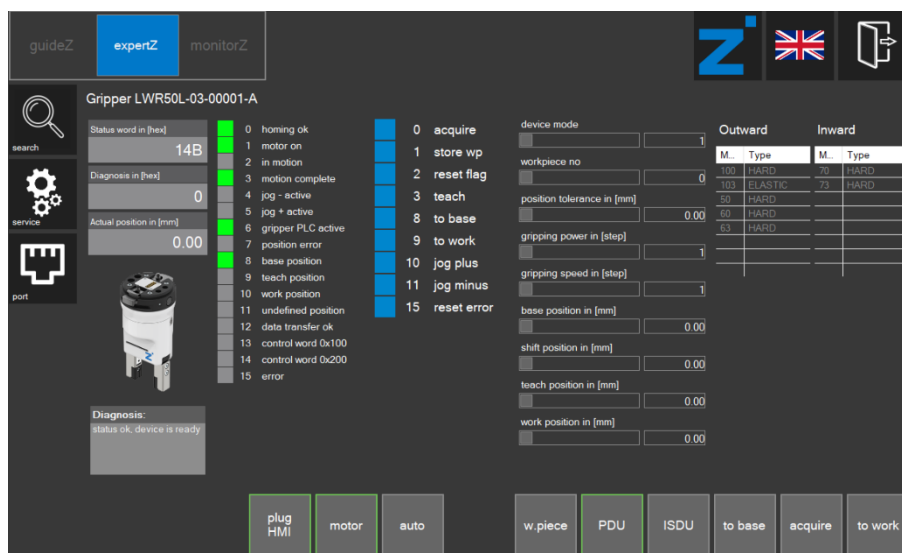
Seuraavassa vaiheessa luodaan lohkon määrittelyt, joissa asetetaan käskyttävä työkalu, liikenoisuus ja pisteet robotin koordinaatistossa. Määrittämätön liikekäsky näyttää esimerkiksi tältä:

```
Move <tool> <quickly> through <somewhere> and <somewhere> circularly
```

Kolmannessa vaiheessa määritetään lohkon graafiset ominaisuudet, komennon ohjetiedot ja pikakuvakekohde FlexPendant-sovellusten avaamista varten. Tämän jälkeen komentolohko tallennetaan ja lopuksi vietään robotin ohjaimen verkkoyhteyden tai USB-muistilaitteen avulla.

3.3 Zimmer HMI

HMI, eli Human Machine Interface on Zimmer Groupin luoma tietokoneohjelmisto robotin tarraimen määrittystä ja konfigurointia varten (kuva 4). Ohjelmassa luodut parametrit tallentuvat robotin työkaluun SCM:n ja tietokoneen välillä RJ45-yhteyden avulla.

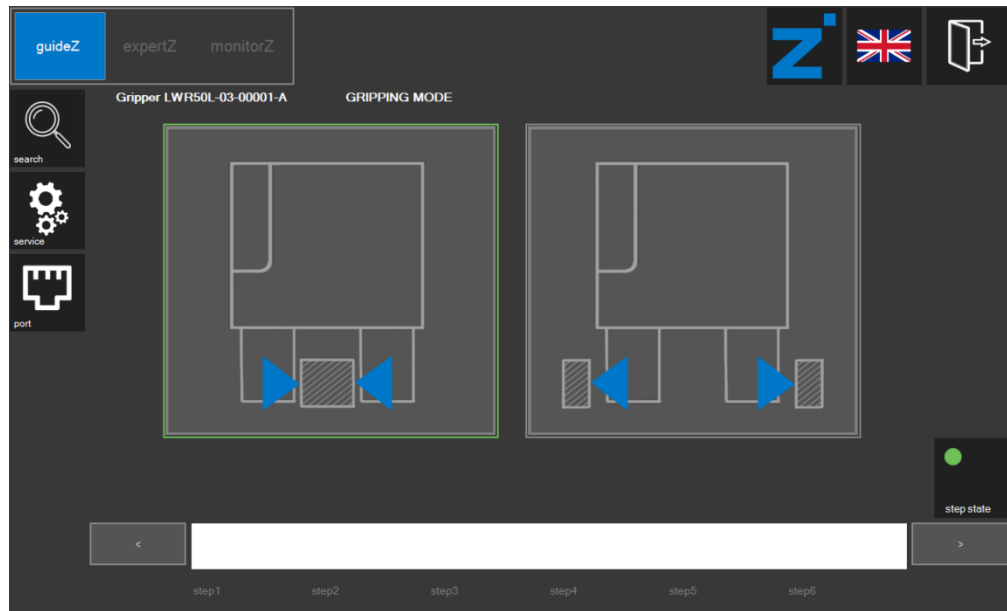


Kuva 4. Tarraimen konfigurointi.

HMI sisältää kolme eri ohjausikkunaa, joissa robotin työkalun tunnistus ja käyttöönotto suoritetaan:

- guideZ on konfiguraationäkymä robotin työkalun käyttöönottoa varten. Näkymässä voidaan opettaa tarraimelle tarvittavat mitat kappaleentunnistusta ja -käsittelyä varten.
- monitorZ on työkalun valvontaa varten, jossa voidaan tarkistaa robotin työkalun asennot ja käyttötilat ohjelman aikana.
- expertZ on diagnostiikkanäkymä, jossa ovat kaikki robotin työkalun tiedot. Näkymässä optimoidaan robotin työkalun parametrit eri käyttösovelluksia varten.

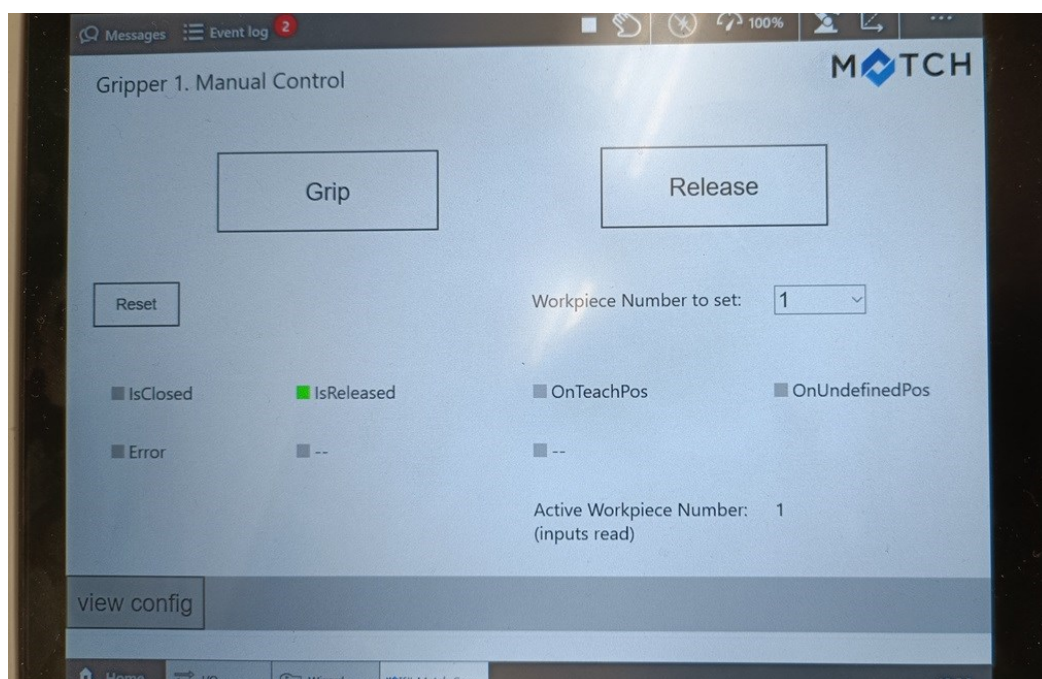
Robotin tarraimelle opetetaan kokoonpanon kolmelle osalle kullekin oma workpiece. Opettaminen on kuusivaiheinen prosessi, jossa määritetään tarrainsuunta, kappaleen asento, toleranssi, tartuntavoima, kappaleen mitat sekä tartuntanopeus (kuva 5).



Kuva 5. guideZ-näkymä.

3.4 Comfort App

Match Comfort App on robotin käsiohjaimen ladattava sovellus, jossa voidaan seurata ja ohjata robotin työkalua manuaalisesti käyttäen aikaisemmin määritettyjä parametrejä (kuva 6). Työkalun ohjaamisen lisäksi sovelluksessa pystytään tarkkailemaan työkalun lähtösignaaleja ohjelman aikana tai mahdollisesti virhetilanteiden diagnosoinnissa sekä valitsemaan tarraimelle eri työkappaleen parametrit.



Kuva 6. Comfort App -näkyvä käsiohjaimella.

3.5 SolidWorks

SolidWorks on CAD-ohjelmisto, joka tarjoaa laajan valikoiman työkaluja ja toimintoja 3D-suunnittelua ja -simulointia varten. Ohjelman avulla käyttäjät pystyvät simuloimaan luomiaan komponentteja tai kokoonpanoja virtuaalisesti ennen niiden valmistusta.

Työssä luotu visio valmiista kokoonpanosta sai alkunsa vuoden 2023 robotiikan integroinnin kursilla. Kappaleiden suunnittelu ja kokoonpanon rakentaminen suoritettiin ryhmissä ja kurssin päätyttyä vastuu mallintamisesta päättyi minulle työharjoittelun ajaksi.

4 GoFa™

GoFa™, eli CRB 15 000 on ABB:n vuonna 2021 valmistama monipuolinen ja intuitiivinen yhteistyörobotti. GoFa kuuluu ABB:n luomaan cobottiperheeseen, johon kuuluu myös GoFaa kompaktimmat SWIFTI- sekä YuMi-cobotit. [9.]

GoFan ja muiden nykyisten cobottien tavoitteena on automatisoinnin lisääminen erityisesti työympäristöissä, joissa ei välttämättä ole aikaisempaa ohjelmointiosaamista. Niiden helppokäyttöisyys ja nopea asennettavuus tulevat helpottamaan cobottien käyttöönottamista automaation lisääntyessä yrityksissä. [10.]

GoFasta on olemassa kooltaan ja hyötykuormaltaan kolme eri versiota. Opinnäytetyö on suoritettu käyttäen pienikokoisinta 5 kg:n versiota, mutta ABB tarjoaa myös 10 kg:n ja 12 kg:n hyötykuorman versioita robotista (kuva 7).



Kuva 7. CRB 15000 -tuotesarja. [10.]

5 Suunnittelu ja kokoonpano

Suunnittelu sai virallisesti alkunsa työharjoittelun aikana, jolloin suoritin muutoksia ja parannuksia robotiikan integroinnin kurssin aikana tehtyyn kokoonpanomalliin. Keskenpäiseksi jäänyt malli sisälsi myös puutteellisia komponentteja tai mallinnusvirheitä.

Ennen robotin saapumista koululle, tein kokoonpanosta 3D-mallin lisäksi animoidun simulaation käyttäen Visual Components -ohjelmistoa sekä RobotStudio virtuaalinäkymää. Simulaatioiden avulla pystyin selvittämään robotin liikeradat, asennot ja nivelrajat ennen varsinaista kokoamista ja ohjelmointia.

Ohjelman tarkoitus on suorittaa yksinkertainen kokoonpano, jossa robotti kasaa järjestyksessä laakerin, akselin sekä kannen kokoonpanopöydälle (kuva 8). Työssä verrataan kahta eri ohjelmointitapaa ja selvitetään, pystytäänkö graafisessa ohjelmointinäkyvässä suorittamaan kappaleenkäsittely yhtä tehokkaasti.



Kuva 8. Kokoonpanon osat.

5.1 Solun komponentit

Robottiin oli hankittu Zimmerin MATCH -paketti, johon kuului myös ECBPMi-paineilmatarra- (kuva 9), RMQC 50 -pikavaihtomoduli (kuva 10), LWR50L-tarttuja (kuva 10), RECBI-ejektorimoduuli ja kolme työkalupidikettä. Kappaleenkäsittely suoritetaan käyttäen LWR50L-tarttujaa ja pikavaihtomodulia. Paineilmatarra- oli työkaluista soveltuvin koordinaatistojen määrittelyyn liisäämällä työkalun kärkeen metallinen piikki.

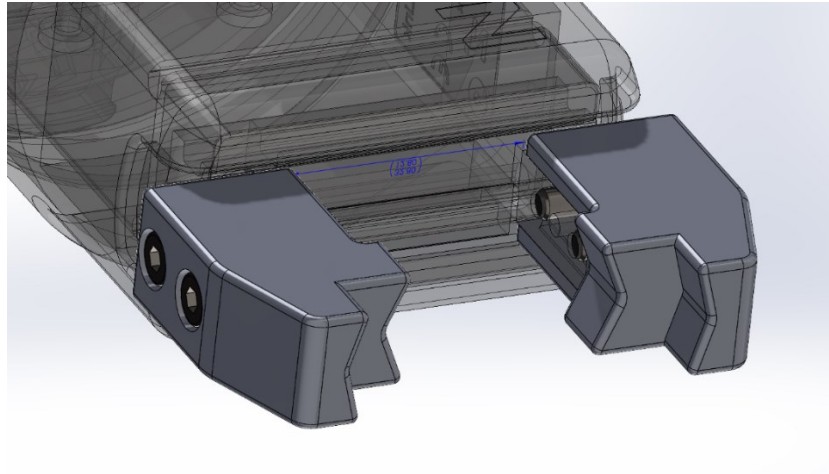


Kuva 9. Paineilmatarrain ja kalibrointiipikki.



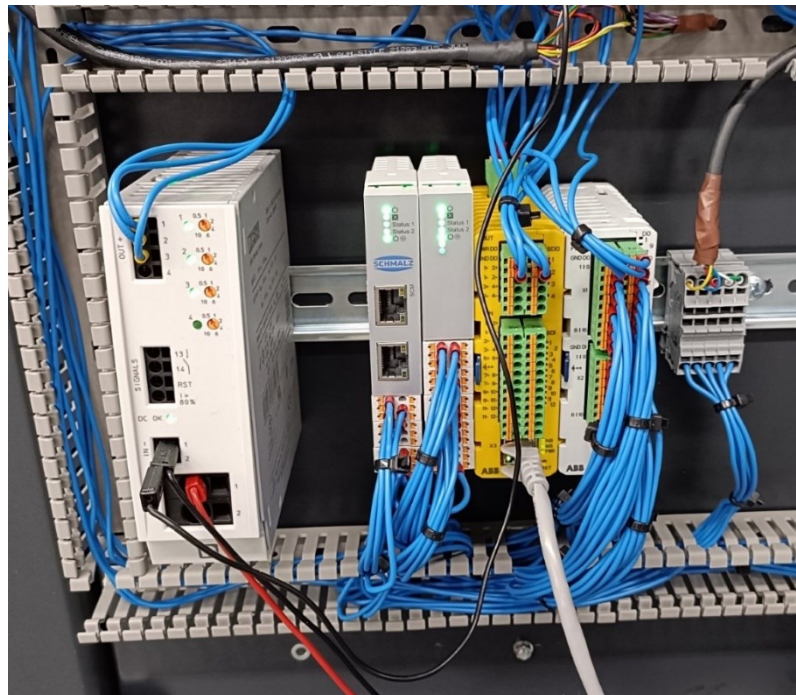
Kuva 10. MATCH-startipaketin tarrain ja pikavaihtomoduli. [11.]

LWR50L-tarraimen oletuksena olleet mekaaniset sormet eivät soveltuneet laakerin ja akselin käsitteelyyn, joten suunnittelimme CAD-mallit ja 3D-tulostimme tarraimeen soveltuvammat sormet (kuva 11).



Kuva 11. CAD-malli tarraimen sormista.

Pöytäsolun ulkopuolelle asennettiin Phoenixin elektroninen katkaisija, Schmalzin SCM -viestintämoduuli sekä ABB:n skaalautuvat I/O-laitteet (kuva 12). Solussa on käytössä DSQ1042-turvayksikkö ja DSQ1031-lisälaite, jonka kautta solun tulo- ja lähtösignaalit määritetään kahteen eri kategoriaan.



Kuva 12. Katkaisija, viestintämoduuli, turvayksikkö, sekä lisälaite.

Solussa on 6 kappaletta induktiivisia antureita, joista kolme on kiinnitetty työkalupidikkeisiin, kaksi luisuihin sekä yksi kokoonpanopöytään antamaan tulosignaalityiedot robotille (kuva 13).

Name	Assigned to Device	Type of Signal	Device Mapping	Signal Identification Label	Category	Access Level	Default Value	Filter Time Passive (ms)	Filter Time Active (ms)	Invert Physical Value
DI9_ToolStandSensor1	ABB_Scalable_IO1	Digital Input	8	DSQC1031	Sensors	Default	0	0	0	No
DI10_ToolStandSensor2	ABB_Scalable_IO1	Digital Input	9	DSQC1031	Sensors	Default	0	0	0	No
DI11_ToolStandSensor3	ABB_Scalable_IO1	Digital Input	10	DSQC1031	Sensors	Default	0	0	0	No
DI12_TraySensor	ABB_Scalable_IO1	Digital Input	11	DSQC1031	Sensors	Default	0	0	0	No
DI13_AxisSensor	ABB_Scalable_IO1	Digital Input	12	DSQC1031	Sensors	Default	0	0	0	No
DI14_BearingSensor	ABB_Scalable_IO1	Digital Input	13	DSQC1031	Sensors	Default	0	0	0	No

Kuva 13. Antureiden I/O-määritykset.

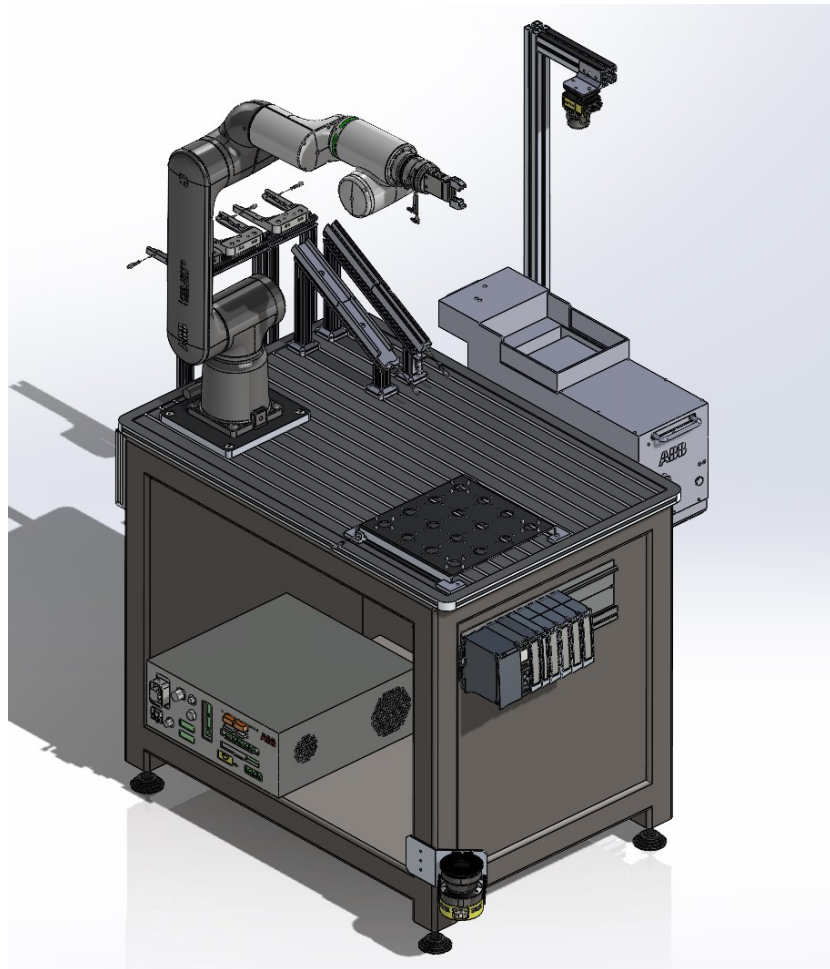
Tarrainta varten on DSQC1031-laitteeseen määritetty 7 lähtösignaalia ohjausta varten sekä 8 tulosignaalia (kuva 14). Signaalit on nimetty teknisen ohjekirjan mukaan. Työssä käytetään lähtösignaaleja ZG_DO0 ja ZG_DO1 ohjaamaan työkalun tartuntaa ja vapautusta.

Name	Assigned to Device	Type of Signal	Device Mapping	Signal Identification Label	Category	Access Level	Default Value	Filter Time Passive (ms)	Filter Time Active (ms)	Invert Physical Value	Safe Level
ZG_DO0	ABB_Scalable_IO1	Digital Input	0	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO1	ABB_Scalable_IO1	Digital Input	1	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO2	ABB_Scalable_IO1	Digital Input	2	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO3	ABB_Scalable_IO1	Digital Input	3	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO4	ABB_Scalable_IO1	Digital Input	4	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO5	ABB_Scalable_IO1	Digital Input	5	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO6	ABB_Scalable_IO1	Digital Input	6	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO7	ABB_Scalable_IO1	Digital Input	7	DSQC1031	Gripper	Default	0	0	0	No	
ZG_DO8	ABB_Scalable_IO1	Digital Output	0	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO1	ABB_Scalable_IO1	Digital Output	1	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO2	ABB_Scalable_IO1	Digital Output	2	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO3	ABB_Scalable_IO1	Digital Output	3	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO4	ABB_Scalable_IO1	Digital Output	4	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO5	ABB_Scalable_IO1	Digital Output	5	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel
ZG_DO6	ABB_Scalable_IO1	Digital Output	6	DSQC1031	Gripper	Default	0			No	DefaultSafeLevel

Kuva 14. Tarraimen lähtö- ja tulosignaali.

5.2 3D-malli

Kappaleiden ja kokoonpanon mallintamisen suoritin SolidWorksin avulla, jossa lopullinen kokoonpanotiedosto sisälsi yhteensä yli 300 komponenttia (kuva 15). Osa laitteista oli valmistajien luomia CAD-tiedostoja, jotka ovat liitetty kokoonpanoon käyttäen itse mallinnettuja adaptoreita ja kiinnikkeitä. Tiedoston suuren koon ja komponenttien laajan määrän vuoksi on kiinnitysruuvit piilotettu valmiista kokoonpanosta.

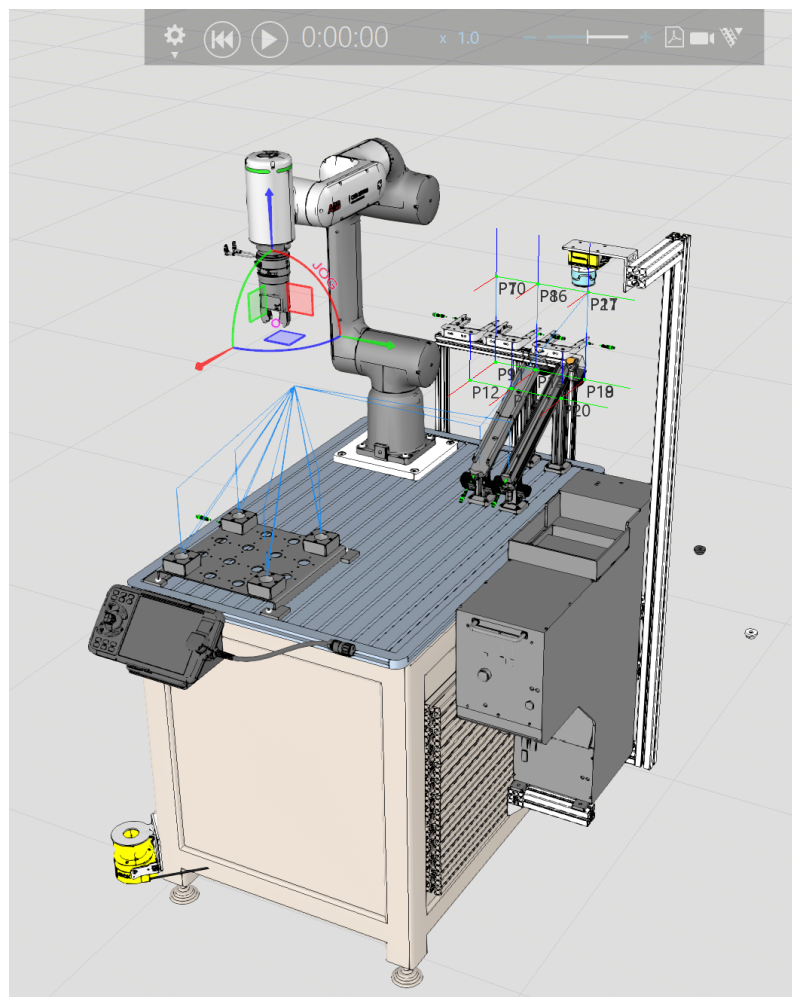


Kuva 15. CAD-kokoonpano robottisolusta.

5.3 Simulaatio

Ensimmäiset versiot simuloidusta kappaleenkäsittelyohjelmasta loin Python-ohjelmointiin pohjautuvassa Visual Components -ohjelmistossa (kuva 16). Simulaatiossa käytin suurimmalta osin valmista kokoonpanoa ilman robottia sekä akseleiden ja laakereiden syöttöluisuja. Kyseiset komponentit oli tuotava simulaatioon erillisinä tiedostoina, jotta niiden ominaisuudet olisivat muokattavissa.

Kappaleensyöttöä varten on luisujen ominaisuudet vaihdettava tavallisesta lohkotiedostosta liukuhihnaksi. Muutoksen avulla luisujen yläpäähän on mahdollista sijoittaa tuotteiden syöttölaite, johon määritetään syötettäväksi kappaleeksi 3D-malli laakerista tai akselistä. Luisujen alapäähän sijoitetaan anturi, joka kommunikoi robotin ja syöttölaitteen kanssa. Tällöin signaalin saadessaan robotti pystyy hakemaan luisun alapäästä halutun kappaleen ja siirtämään sen kokoonpanopöydälle.



Kuva 16. Visual Components -malli robottisolusta.

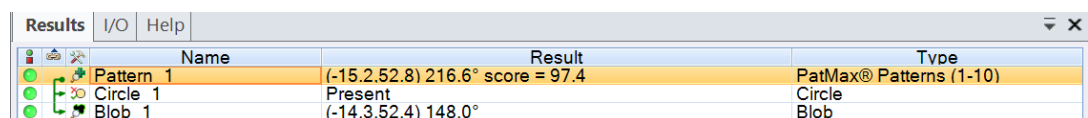
5.4 Konenäkö

Kansikomponentteja varten luotiin konenäköä hyödyntävä osio, jossa kuvattava kohde sijaitsee FlexFeeder-syöttölaitteen valopöydällä. Ideaaltilanteessa FlexFeeder keskustelee robotin kanssa ProfiNET-yhteyden avulla, joka on teollisuuskäyttöön tarkoitettu tavallista Ethernet-yhteyttä tehokkaampi viestintäväylä. [12.] Syöttölaite antaa signaalikäskyn saadessaan kannen kuvattavaksi valopöydälle. Yhteensopivuusongelmien vuoksi emme kuitenkaan saaneet kommunikointilaitteena toimivaa ADAM 6150PN-moduulia toimimaan halutulla tavalla, jonka vuoksi kappaleet kuvataan FlexFeederin valopöydältä ilman erillistä syöttöä.

Kuvaus suoritetaan Cognexin In-Sight 7600 -sarjan kameralla, joka on Cognexin 7000-sarjan perusmalli. Kamera toimii mustavalkoversiona, jonka resoluutio on 800 x 600 pikseliä. Kamera on kiinnitetty alumiiniprofiileilla FlexFeederin yläpuolelle ja itse pöytäsoluun.

Kuvaus suoritetaan RobotStudio Integrated Vision -konenäköjärjestelmän kautta, joka on RobotStudio sisäänrakennettu versio Cognexin erillisestä In-Sight Explorer -ohjelmistosta. Visiojärjestelmä mahdollistaa kuvattavan alueen ja kohteen paikallistamisen, tarkastamisen ja ohjaamisen tietokoneelta käsin.

Konenäöllä etsitään työssä koneenpanon kantta käyttämällä kolmea eri konenäön työkalua (kuva 17), joista ensimmäinen on PatMax Pattern. PatMax-toiminto etsii ympyrämäistä kuviota kaksivaiheisen prosessin avulla. Toiminnolla haetaan kohdetta suodattimen avulla mahdollisten vaihtoehtojen löytämiseksi käyttäen pisteytysparametreja. Näitä ovat esimerkiksi hyväksymisrajat, kontrasti, kuvakulma, skaalaus ja päällekkäisyydet.



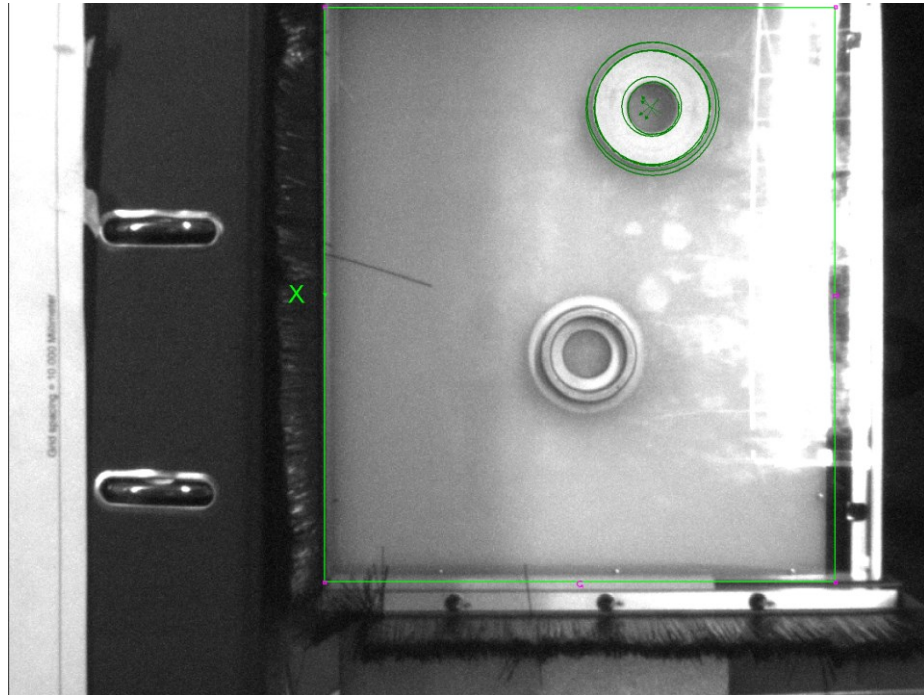
Name	Result	Type
Pattern 1	(-15.2.52.8) 216.6° score = 97.4	PatMax® Patterns (1-10)
Circle 1	Present	Circle
Blob 1	(-14.3.52.4) 148.0°	Blob

Kuva 17. Työssä käytettävät työkalut.

Jokainen parametri antaa tietyn pistearvon ja toisessa vaiheessa annetaan lopullinen pistemäärä yhdistämällä annetut arvot. Toiminto voi antaa kappaleelle hyväksytyyn arvon, vaikka osa parametreista saattaa saada huonon pisteytyksen. Tämän vuoksi hyväksymisrajaa tulee nostaa tarvittaessa korkeampaan arvoon, jotta kappaleentunnistus ei valitse ohjelmaan ulkopuolisia tai ei-toivottuja kappaleita.

PatMax-toiminnossa toimii lisäksi ympyrän paikannustyökalu sekä blob-suodatin. Paikannustyökalulla haetaan kuvasta kannen sisä- ja ulkohalkaisijaltaan samankaltaista kohdetta. Tunnistuksen tulee myös hylätä mahdollisesti väärin päin oleva kansi, jota ei ole tarkoitus poimia (kuva 18).

Blob-suodattimen avulla erotetaan väärin päin asetetun kannen kontrasti- ja pintaerot poimittavasta kappaleesta. Hyväksymisrajaa nostamalla kuvantunnistus hylkää liikaa kontrastieroja sisältävän kappaleen ja poimii oikein päin asetetun kannen kuvattavalta alueelta.



Kuva 18. Hyväksytty kappaleentunnistus.

6 Ohjelmointi

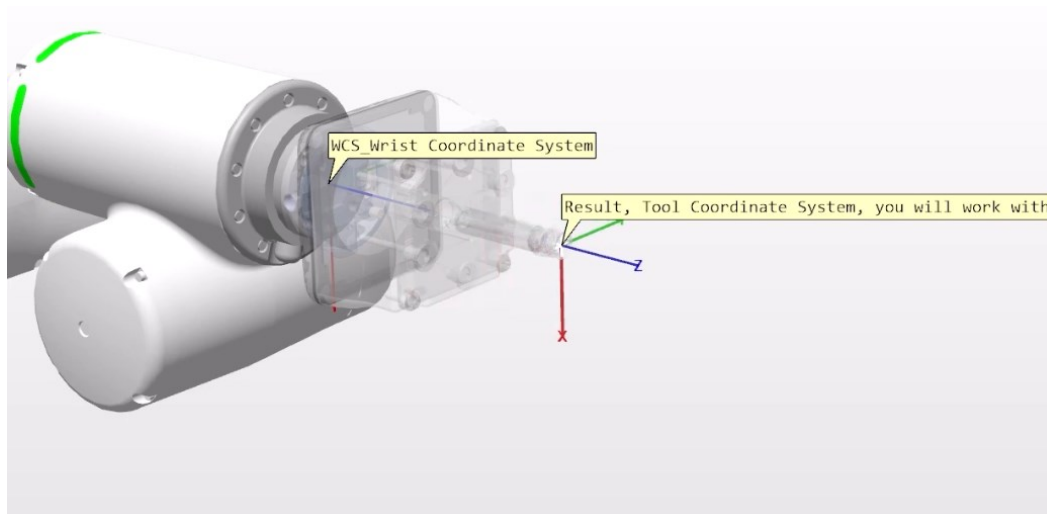
Opinnäytetyössäni, sekä yleisesti robotin ohjelmoinnissa, ennen ohjelmointia määritetään työkalukoordinaatistot, eli työkalukeskipisteet. Robotti liikuttaa ohjelmassa työkalukeskipistettä työkalukoordinaatiston pisteestä toiseen ohjelman mukaisesti. Määritysprosessit voivat vaihdella eri valmistajien roboteissa, mutta pääperiaate niissä on sama. Työkalukoordinaatisto luotiin työssä käyttäen kolmipistemenetelmään perustuvaa TCP:n määrittystä.

6.1 Koordinaatistot

Robotin koordinaattijärjestelmä on tietynlainen kehys, jota käytetään määrittämään robotin sijainti ja suunta kolmiulotteisessa ympäristössä. [13.] Koordinaattijärjestelmiä on olemassa erilaisia ja ne liittyvät toisiinsa arvojärjestyksessä, jossa kunkin koordinaatiston alkuperä määritetään asemaksi. Robotin koordinaattijärjestelmät jaetaan järjestyksessä:

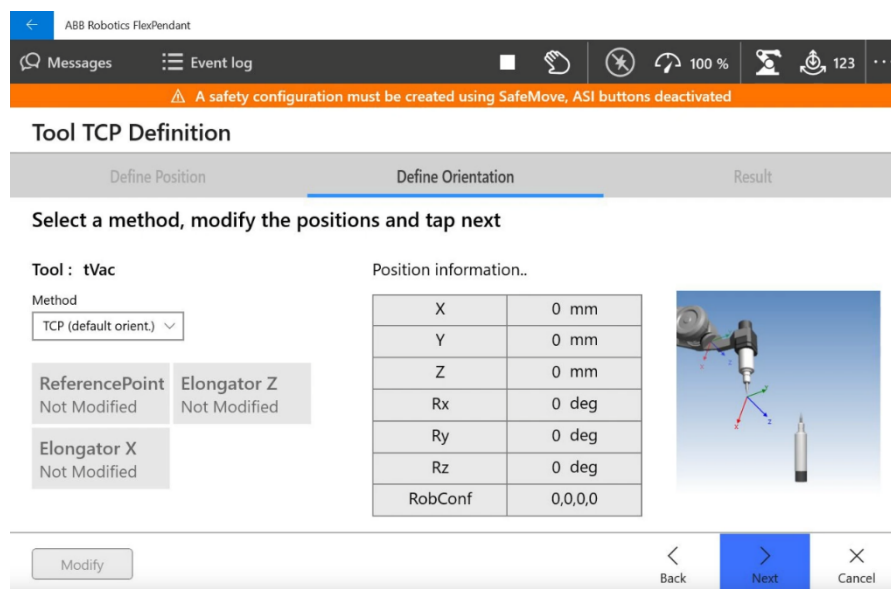
- Maailmakoordinaatisto toimii hierarkian huippuna, johon muut koordinaattijärjestelmät liittyvät.
- Task frame, eli peruskoordinaatisto, johon määritetään asemaan tai soluun sisältyvät pisteet.
- Tool-centre point (TCP), eli työkalukoordinaatisto toimii robotin työkalun keskipisteenä (kuva 19). Roboteilla on yksi ennalta määritetty työkalukoordinaatiston kiinnityspiste, mutta tarvittaessa roboteille voidaan määrittää useita eri TCP:itä, mikäli käytössä on esimerkiksi useita eri työkaluja.

Koordinaatiston määrittämisessä käytetään hyväksi valmiiksi asetettua referenssipistettä robotin käsivarren laipassa (kuva 19) sekä työkalun mitta-arvoja. Koska suoran tarttujan keskipiste on oletussuuntaisesti z-akselilla, on keskipiste helppo määrittää tarkistamalla työkalun pituus sen teknisistä tiedoista tai itse mittaamalla.

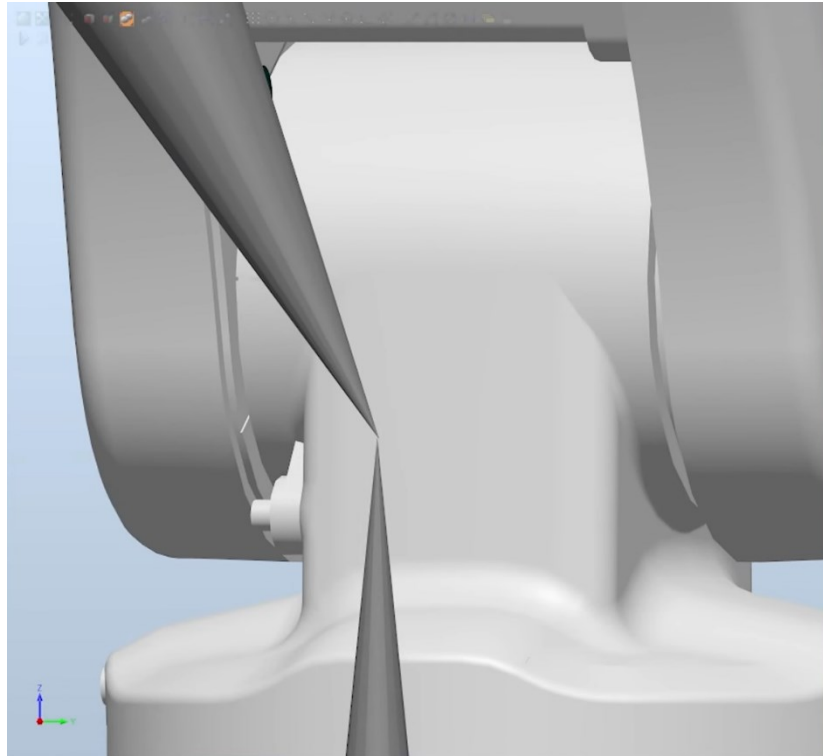


Kuva 19. Laipan ja työkalun koordinaattijärjestelmä.

TCP:n määrittäminen suoritetaan kolmea referenssipistettä käyttämällä (kuva 20). Kalibrointikärkenä toimii robotin paineilmatyökaluun asennettu piikki ja toisena kärkenä pöytäsoluun kiinnitetty piikki. Kärjet asetetaan vastakkain ja kolme referenssipistettä tallennetaan liikuttamalla robotti kolmeen eri asentoon (kuva 21).



Kuva 20. TCP:n orientaationäkymä.



Kuva 21. Visualisoitu TCP:n määrittäminen RobotStudiassa.

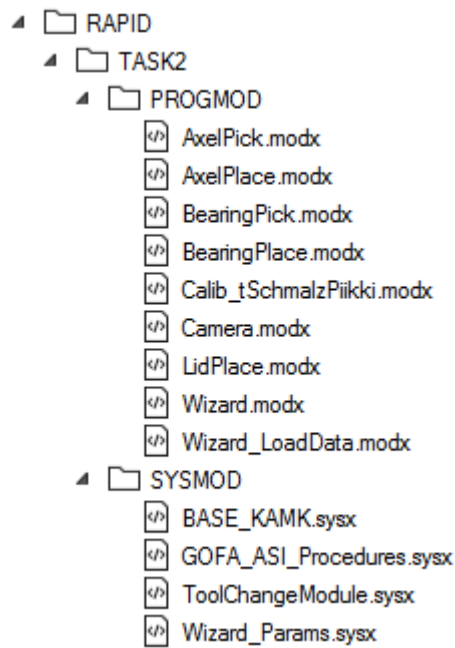
6.2 Työkalumäärittäminen

Ennen varsinaista käyttöönottoa ja ohjelmointia on robotille määritettävä asennettavan tarraimen työkalukeskipiste, massa, massakeskipiste ja hitausmomentti. Parametriarvot voidaan sijoittaa manuaalisesti luomalla uudet tarttujan tiedot, tai robotilla voidaan suorittaa automaattinen kuormantunnistus, jolloin robotti tunnistaa asennettujen työkalun massan sekä painopisteen mitausliikkeiden avulla.

Harjoituksessa tarvittavaa tarkkaa massatietoa varten on työkalumäärittäminen suoritettu automaattisella kalibroinnilla. GoFan tapauksessa automaattinen työkalumäärittäminen suoritetaan LoadIdentify -palvelurutiinin avulla, jossa eri akselit liikkuvat 90 astetta edestakaisessa liikkeessä ja robotti pysyy näin laskemaan asennettujen työkalun massan liikkeistä saadun datan avulla.

6.3 RAPID

RAPID-ohjelmat jaetaan harjoituksessa kahteen eri moduulikansioon (kuva 22). SYSMOD, eli järjestelmäkansio sisältää työkaluihin, koordinaatistoihin, ASI-menettelyihin ja Wizard-parametreihin liittyvät tiedot. PROGMOD-kansio sisältää itse ohjelmatiedostot kappaleenkäsittelyä varten. [14.] Jokaiselle ohjelman vaiheelle on luotu oma tiedosto helpottamaan lopullisen ohjelman luettavuutta ja rakentamista Wizard-näkymässä.



Kuva 22. RAPID-ohjelmapuu.

6.3.1 Laakeri ja akseli

Ohjelmassa kappaleen poimintaa ja kokoonpanoa varten on laakerille ja akselille tehty hyvin samankaltaiset ohjelmat, joissa erona on poimimispiste. Ennen itse ohjelmaa on määritettävä koordinaattipiste osan poimintaa varten, josta vastaa koodi:

```
CONST robtarget pBearingPick:=[[0,0,0],[1,0,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9]];
```

CONST-käskyssä määritetään robotin kohde, joka on nimetty laakerin tapauksessa pBearingPick. Poimintapistettä varten nimeäminen aloitetaan pienellä p-kirjaimella tunnistettavuuden vuoksi.

Pisteen jälkeen tulevat numerosarjat ovat koordinaattiarvoja, jotka luodaan käskyllä :=. Ensimmäiset kolme numeroa sisältävät robotin kohdesijainnin x-, y-, ja z-tasoissa millimetreinä. Sijainti määrittyy perustuen työkohteen koordinaattijärjestelmään tai vaihtoehtoisesti maailmakoordinaatistoon, mikäli ohjelmassa ei ole määritettyä työkohdetta.

Seuraavat neljä numeroa määrittävät paikoituspisteen orientaation kvaternioina, eli kompleksilukujen laajennuksina q1, q2, q3 ja q4. [15.] Suunta määrytyy suhteessa työobjektikoordinaatistoon tai kohdesijainnin tapaan maailmakoordinaatistoon, mikäli määrittystä ei ole.

Robotin akselikonfiguraatio määrytyy orientaation jälkeen. Konfiguraatiossa määritetään komponentit cf1, cf4 ja cf6, jotka saadaan akseleiden 1, 4 ja 6 sen hetkisten neljänneskierroksien muodossa. Konfiguraation viimeinen komponentti cfx riippuu itse robotin tyyppistä. [14.]

Viimeisessä numerosarjassa on lisäakseli sijainti. Yhteensä kuudelle akselille tehtävä määrittäminen näkyy aakkosjärjestyksessä eax_a, eax_b, eax_c ja jatkuu kirjaimien f saakka. Pyörivät akselit määrittyvät kiertoasteina kalibrointiasennosta, kun taas lineaariakselit määritetään millimetritäisyytenä kalibrointiasennosta. Kytkemättömille akseleille on annettu oletusarvo 9E9. [14.]

Rutiinin sisään luodaan itse ohjelma, joka on laakerin poimintaa varten nimellä PickBearing (kuva 23). Ensimmäisessä käskyssä liikutetaan robotti poimintapisteen yläpuolelle käyttäen Offs-käskyä. Liikekäskyssä määritetään ensiksi radan tyyppi, joka voi olla lineaarinen L, yhdistetty J tai ympyräliike C. Tämän jälkeen määritetään liikkeen nopeus yksikössä millimetriä per sekunti, joka on harjoituksessa v100 = 100 mm/s. Nopeuden jälkeen määritetään liikkeen aluekoon tarkkuus millimetreinä, joka on yhdistetyssä liikkeessä arvossa z30 = 30 mm. Mikäli kyseessä on tarkkuutta vaativa liike, kuten lineaariliikkeessä, käytetään silloin arvona fine, jossa robotti siirtyy tarkasti pisteeseen ilman poikkeamaa liikeradasta.

```

5 PROC PickBearing()
6   MoveJ Offs(pBearingPick,0,0,50), v100, z30, tZimmerLW50L\WObj:=wLuisuBearing;
7   MoveL pBearingPick, v100, fine, tZimmerLW50L\WObj:=wLuisuBearing;
8   Z_CHANGEWP 1,5,continue;
9   WaitTime 1;
10  SetDO ZG_D01, 1;
11  WaitTime 1;
12  SetDO ZG_D01, 0;
13  WaitTime 1;
14  MoveL Offs(pBearingPick,0,0,50), v100, z30, tZimmerLW50L\WObj:=wLuisuBearing;
15
16 ENDPROC
17
18 ENDMODULE

```

Kuva 23. Laakerin poiminta.

Arvojen jälkeen valitaan liikkeessä käytettävä työkalu ja työkohde. Koska ohjelmassa haetaan laakeria luisusta, on työkohteeksi valittu wLuisuBearing. Työkohteen nimeäminen aloitetaan w-kirjaimella käyttäen samaa periaatetta, kuin koordinaattipisteen nimeämisessä.

Offs-lähestymiskäskey on hyvin lineaarisen liikkeen kaltainen, jossa käytetään referenssipisteenä poimintapistettä. Pisteen jälkeiset arvot ovat x-, y-, ja z-tasoissa millimetreinä. Harjoituksessa lähestymispiste on määritetty z-tasossa 50 millimetriä ylöspäin poimintapistestä, jotta robotti voi siirtyä poimintapisteeseen turvallisesti ilman törmäystä.

Z_CHANGEWP on ZIMMERin laatima oma käskey työkalun vaihtoa varten. Käskyssä työkalukseksi valitaan HMI:ssa laakerille opetettu kappale 1, jota käskey yrittää toistaa yhteensä viisi kertaa, mikäli ohjelmassa tapahtuu virhe. Tämän jälkeen ohjelmassa siirrytään seuraavaan käskeyn.

Robotin tarrainta käskeytetään muuttamalla digitaalisia lähtöarvoja. ZG_DO1 -lähtö on tartuntaa varten, jossa arvo 1 ottaa kappaleesta kiinni. Koodissa on muutettava arvo manuaalisesti takaisin nollaan, jotta lähtöarvolla ZG_DO0 saadaan kappale vapautettua irti kokoonpanossa. Käskyjen välissä on myös yhden sekunnin odotuskäskeyt, jotta tarrain ei siirry virhetilaan.

Kokoamisen alkuun on määritetty muuttuva numero x- ja y -suuntaista liikettä sekä tehtyjä kokoonpanoja varten (kuva 24). Luvuilla määritetään koordinaatiston siirtäminen x- ja y -akselilla aina yhden kokoonpanon valmistuttua. nDist-arvoon on määritetty kahden kokoonpanon keskipisteen etäisyys toisistaan.

```
5 VAR num nXcount;  
6 VAR num nYcount;  
7 VAR num nAssembliesDone;  
8 CONST num nDist:= 80;
```

Kuva 24. Lukujen määrittäminen.

Pöydällä on x- ja y-suuntaisesti tilaa yhteensä 16 kokoonpanolle. Tämän vuoksi kansiohjelman loppuun on tehty IF-komento, joka määrittää paikoituspisteen siirtämiselle maksimiarvon 4. Komenton Incr avulla paikoituspistettä siirretään 80 millimetriä korkeintaan neljä kertaa x- tai y-akselin suuntaisesti. Maksimiarvon saavutettua ohjelma siirtää paikoituspistettä y-akselin suuntaisesti kerran, jatkaa kokoamista x-akselin suuntaisesti ja lopulta pysäyttää ohjelman, kun kaikki 16 kokoonpanoa on valmistettu (kuva 25).

```

27     Incr nXcount;
28     IF nXcount = 4 THEN
29         nXcount:=0;
30         Incr nYcount;
31     ENDIF
32
33     IF nYcount = 4 THEN
34         nXcount:=0;
35         nYcount:=0;
36         Stop;
37     ENDIF

```

Kuva 25. Paikoituspisteen siirto.

Liikkeet suoritetaan Offs-käskyillä, joissa x- ja y -akseleiden arvot saadaan yhtälöillä:

$nXcount \cdot nDist$ ja $nYcount \cdot nDist$. Muuten ohjelman suoritus tapahtuu samalla periaatteella, kuin poiminnassa.

```

23 PROC PlaceBearing()
24
25     MoveJ Offs(pBearingPlace,nXcount*nDist,nYcount*nDist,100), v200, z30, tZimmerLW50L\WObj:=wTray;
26     MoveL Offs(pBearingPlace,nXcount*nDist,nYcount*nDist,0), v100, fine, tZimmerLW50L\WObj:=wTray;
27     WaitTime 1;
28     SetDO ZG_D00, 1;
29     WaitTime 1;
30     SetDO ZG_D00, 0;
31     WaitTime 1;
32     MoveL Offs(pBearingPlace,nXcount*nDist,nYcount*nDist,100), v100, z30, tZimmerLW50L\WObj:=wTray;
33
34     !Incr nXcount;
35     IF nXcount = 4 THEN
36         nXcount:=0;
37         Incr nYcount;
38     ENDIF
39
40     IF nYcount = 4 THEN
41         nXcount:=0;
42         nYcount:=0;
43         Stop;
44     ENDIF
45
46 ENDPROC

```

Kuva 26. Laakerin asettaminen.

6.3.2 Kannen kappaleentunnistus

Kameramoduulia varten on RobotStudiassa snippet, eli katkelmatiedosto, joka on valmis koodi konenäköä ja kameranhallintaa varten (kuva 27). Valmiista katkelmatiedostosta on tarpeettomat osiot muutettu kommentteiksi lisäämällä koodirivin alkuun huutomerkki.

Periaate kameramoduulissa on lukea visiojärjestelmässä tallennettu job-tiedosto, jota kautta tunnistetaan hyväksyty kansikappale ja se poimitaan muuttamalla kappaleen sijainti pöydällä poimintapisteeksi koordinaatistossa. Ohjelmassa nimetty kamera is7600_FlexFeeder asetetaan ohjelmointitilaan, jonka jälkeen ladataan job-tiedosto kameraan ja lopuksi kamera siirtyy toimintatilaan. Oletuksena ensimmäisen käynnistyksen yhteydessä kameran valotusaika on liian lyhyt luettavaa kuvaa varten, joten ohjelmaan on myös lisätty IF-käsky nostamaan valotusaikaa.

Valotusaikakäskyn jälkeen kameralla otetaan kuva CamReqImage -käskyllä ja CamGetResult -käskyllä haetaan tunnistettavaa kohdetta kuvasta. [14.] Kansikappaleen LidGray:n tunnistettaessa ohjelma muuttaa koordinaatistoon poimintapisteeksi kappaleen sijainnin pöydällä.

```

1 MODULE Camera
2
3 CONST robtarget myrobtarget:=([-3.76, -1.18, 17.56], [0.0111454, -0.999873, -0.0111413, 0.0023
4 VAR num ExposureTime;
5 CONST string myjob:="GoFa_Lid3.job";
6 VAR cameratarget mycameratarget1;
7 !CONST robtarget myimagepos:=([[100,200,300],[1,0,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]]);
8 !TASK PERS wobjdata mywobj:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
9 !PERS tooldata mytool:=[TRUE,[[0,0,1],[1,0,0,0]],[1,[0,0,0],[1,0,0,0],0,0,0]];
10
11 PROC MoveToDetectedObject()
12   CamSetProgramMode is7600_FlexFeeder;
13   CamLoadJob is7600_FlexFeeder, myjob;
14   CamSetRunMode is7600_FlexFeeder;
15   IF ExposureTime = 8 THEN
16     CamSetExposure is7600_FlexFeeder \ExposureTime:=160;
17   ENDIF
18   CamReqImage is7600_FlexFeeder;
19   CamGetResult is7600_FlexFeeder, mycameratarget1\MaxTime:=3.0;
20   IF mycameratarget1.name="LidGray" THEN
21     wFlexFeeder.oframe := mycameratarget1.cframe;
22   ENDIF
23
24   MoveJ Offs(myrobtarget,0,0,50), v200, z30, tZimmerLW50L\WObj:=wFlexFeeder;
25   MoveL myrobtarget, v100, fine, tZimmerLW50L\WObj:=wFlexFeeder;
26   Z_CHANGEWP 3,5,continue;
27   WaitTime 1;
28   SetDO ZG_D01, 1;
29   WaitTime 1;
30   SetDO ZG_D01, 0;
31   WaitTime 1;
32   MoveL Offs(myrobtarget,0,0,50), v100, z30, tZimmerLW50L\WObj:=wFlexFeeder;
33
34
35
36   !During the first cycle, run the program until this point,
37   !then jog the tool to the desired grip position and modpos myrobtarget.
38   !MoveL myrobtarget, v100, fine, tZimmerLW50L\WObj:=wFlexFeeder;
39 ENDPROC
40
41 ENDMODULE

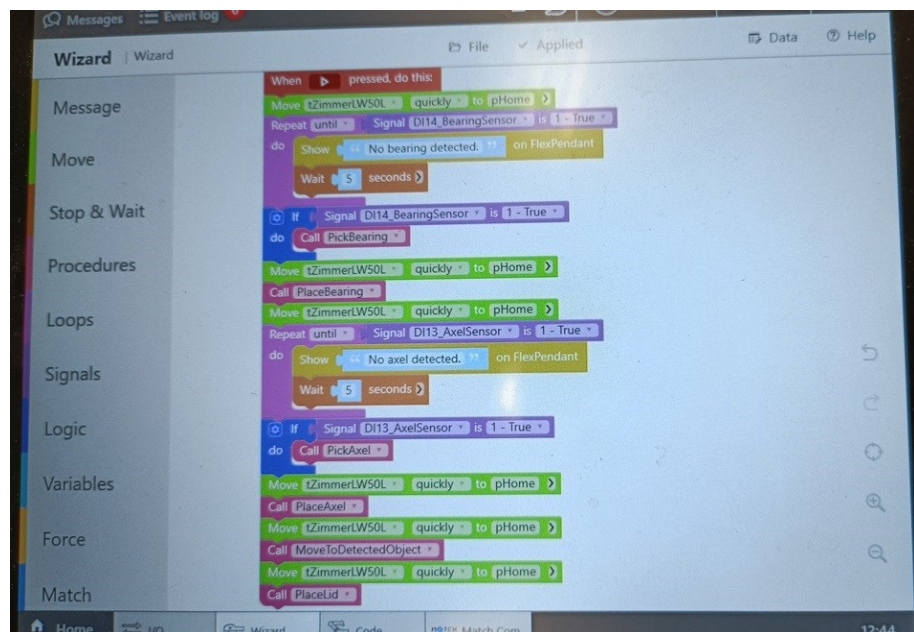
```

Kuva 27. Kameramoduuli.

6.4 Easy Wizard

Valmiin ohjelman kasaaminen tapahtuu lopuksi RobotStudiossa luotuja lohkoja lisäämällä (kuva 28). Koska Wizard ei sisällä mahdollisuutta luoda muokattuja RAPID-käskyjä, näkyvät luodut käskyt expert-koodeina. Tämä rajoittaa mahdollisuutta muokata käskyjen sisältämiä arvoja tai koordinaatteja jälkeenpäin Wizard-näkymässä, minkä vuoksi mahdolliset muutokset koodiin on suoritettava RobotStudion, tai joissain tapauksissa Skill Creator -lisäosan kautta.

Wizard-näkymässä vihreät liikekäskylohkot sisältävät kotipisteen, johon robotti ajaa aliohjelma-kutsujen välissä. Ohjelmassa on lisätty signaalin tunnistuskäskyt toistokäskyjen sisään. Mikäli anturisygnäali ei anna haluttua arvoa viiden sekunnin aikana, antaa ohjelma ongelmasta viestin käyttäjälle käsiohjaimen. Jos anturisygnäali antaa halutun arvon, kutsutaan silloin poimintaohjelmaa IF-käskyllä.



Kuva 28. Wizard-ohjelmointinäkömä valmiista ohjelmasta.

7 Yhteenveto

Tavoitteena työssäni oli luoda toimiva kappaleenkäsittely käyttäen perinteistä RAPID-ohjelmointia RobotStudiossa sekä uutta graafista Wizard-ohjelmointinäköymää. Työskennellessä opin paljon enemmän vianmäärityksestä, asentamisesta ja integroinnista. Vaikka harjoitus toi yllättäviä ja epämääräisiä haasteita työn alkuvaiheessa, pystyin silti luomaan halutun ohjelman, joka täytti itselleni asettamani tavoitteet.

Kompromissien takia työstä jäi puuttumaan ominaisuuksia, joita on mahdollista täydentää harjoitusten ja opetuksen parissa. Konenäköön liittyvää kappaleentunnistusta voidaan tulevaisuudessa parantaa ja opettaa tunnistamaan erisävyisiä kappaleita. Myös ohjelmointia voidaan jatkaa lisäämällä koodiin esimerkiksi työnkalunvaihdon suorittava aliohjelma.

Itse ohjelmointi suoritettiin suurimmalta osin perinteisesti RobotStudion puolella. Rajoittavana tekijänä Wizard-ohjelmoinnissa on mukautettujen käskyjen luominen. Olemassa olevat tai itse luodut lohkot eivät pysty suorittamaan haluttuja käskyjä samalla tavalla, kuin RobotStudiolla tehdessä. Tästä syystä esimerkiksi vaaditut mukautetut käskyt kappaleentunnistuksessa näkyvät Wizardissa niin sanottuina asiantuntijakoodeina, joita ei voi jälkikäteen muokata Wizard-näkymässä.

Etuna Wizardissa oli kuitenkin sen nopeus ja helppokäyttöisyys. Mikäli käyttäjän on mahdollista luoda mukautetut aliohjelmat RobotStudion puolella, on valmis ohjelma helppo kasata Wizardissa graafisia lohkoja liikuttamalla. Ohjelman muodostamista nopeuttaa myös valmiit peruskäskylohkot, jotka on helppo raahata haluttuun kohtaan ohjelmassa.



Kuva 29. Valmis kokoonpano vasemmalla.

Lähteet

1. Suomen Robottiikkayhdistys ry. Teollisuuden robotiikka. [1, s. 28]. Viitattu 15.1.2024
2. Chicago Tribune. Mechanical Advantage - Ensimmäiset cobotit. 1996. Viitattu 18.1.2024. [Internet]. Saatavilla: <https://peshkin.mech.northwestern.edu/cobot/chitrib/jonvan.html>
3. IFR. World Robotics 2023 Report. Viitattu 19.1.2024 [Internet]. Saatavilla: <https://ifr.org/ifr-press-releases/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas>
4. Research Reports World. Global Collaborative Robot Market Report, History and Forecast 2017-2028. Viitattu 19.1.2024 [Internet]. Saatavilla: <https://www.researchreports-world.com/enquiry/request-sample/21059074>
5. Universal Robots. CB3-tuotesarja. Viitattu 15.2.2024. [Internet]. Saatavilla: <https://www.universal-robots.com/products/cb3/>
6. Universal Robots. Universal Robotsin E-sarjan cobotit. Viitattu 16.2.2024. [Internet]. Saatavilla: <https://www.universal-robots.com/fi/tuotteet/>
7. Visual Components. Robot Simulation Software: Everything You Need to Know. Viitattu 16.2.2024. [Internet]. Saatavilla: <https://www.visualcomponents.com/resources/blog/robot-simulation-software-everything-you-need-to-know/>
8. Control Automation. Introduction to ABB Robot Programming Language. Viitattu 17.2.2024. [Internet]. Saatavilla: <https://control.com/technical-articles/introduction-to-abb-robot-programming-language/>
9. ABB. Yhteistyörobotisarja. Viitattu 17.2.2024. [Internet]. Saatavilla: <https://new.abb.com/products/robotics/robots/collaborative-robots>
10. ABB. CRB 15000 -tuotesarja. Viitattu 17.2.2024. [Internet]. Saatavilla: <https://new.abb.com/products/robotics/robots/collaborative-robots/crb-15000>
11. ABB Webshop. Match gripper starter package. Viitattu 19.2.2024. [Internet]. Saatavilla: <https://webshop.robotics.abb.com/us/zimmer-group-match-gripper-starter-package-abb-kit.html>
12. PROFIBUS. ProfiNET System Description. Viitattu 25.2.2024. [Internet]. Saatavilla: <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=51714&to-ken=4ea5554cbb80a066e805a879116ead2a759c23c3>
13. SolisPLC. Understanding Robot Coordinate Frames and Points. Viitattu 1.3.2024. [Internet]. Saatavilla: <https://www.solisplc.com/tutorials/robot-coordinate-frames-and-points>
14. ABB. Technical reference manual: RAPID Instructions, Functions and Data types. Viitattu 3.3.2024. [Internet]. Saatavilla: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RA-PID_3HAC16581-1_revJ_en.pdf

15. RobotStudio developer center. Struct Quaternion. Viitattu 3.3.2024. [Internet]. Saatavilla: <https://developercenter.robotstudio.com/api/robotstudio/api/ABB.Robotics.Math.Quaternion.html>