

samk



Satakunnan ammattikorkeakoulu  
Satakunta University of Applied Sciences

JUUSO LEHTI

# **Konepalvelurobotin prototyypin oh- jelmointi**

SÄHKÖ- JA AUTOMAATIOTEKNIIKAN TUTKINTO-OH-  
JELMA  
2025

## TIIVISTELMÄ

Lehti, Juuso: Konepalvelurobotin prototyypin ohjelmointi  
Opinnäytetyö, AMK  
Sähkö- ja automaatiotekniikan tutkinto-ohjelma  
Maaliskuu 2025  
Sivumäärä: 56

Tässä opinnäytetyössä perehdyttiin konepalvelun automatisointiin robotiikkaa hyödyntämällä. Työssä käsiteltiin konepalvelua, robotiikkaa sekä yleisiä asioita robottien ohjelmointiin liittyen.

Opinnäytetyön tavoitteena oli kehittää helppokäyttöinen ja laajennettavissa oleva konepalvelurobotin ohjelma, jonka avulla robotti voisi suorittaa konepalvelua mahdollisimman itsenäisesti. Tämä työ tehtiin osana laajempaa kehitysprojektia.

Robotin ohjelma kehitettiin lähes kokonaan offline-ohjelmointina hyödyntäen ABB:n RobotStudio-ohjelmistoa sekä robotin prototyypin digitaalista kaksosta. Prototyypin ohjelmoinnissa pyrittiin käyttämään hyviä ohjelmointitapoja sekä selkeää ohjelmajaottelua. Lisäksi ohjelmoitiin käyttäjäystävällinen käyttöliittymä, jonka kautta robotin operaattori pystyy hallitsemaan robotin toimintaa ja sen järjestelmässä olevia tietoja. Kaikki robotin toiminnallisuudet ohjelmoitiin RAPID-ohjelmointikieltä käyttäen. Työssä esiteltiin myös käytännön ohjelmointiesimerkkejä.

Tämän opinnäytetyön lopputuloksena oli konepalvelurobotin prototyypin ohjelma, jonka avulla yhteistyörobotti saadaan suorittamaan konepalveluun liittyviä tehtäviä operaattorin syötteiden perusteella. Alustavien käyttötestien perusteella ohjelmaan kuuluvat toiminnot toimivat oikein ja robotin työkierto sujui ongelmitta. Työssä tehty ohjelma toimii pohjana tulevalle robotin ohjelmakehitykselle.

Avainsanat: Konepalvelu, robotiikka, ohjelmointi

## ABSTRACT

Lehti, Juuso: Programming a Machine Tending Robot Prototype

Bachelor's thesis

Electrical and Automation Engineering

March 2025

Number of pages: 56

In this thesis the focus was on automating machine tending using robotics. The work covered machine tending, robotics and general aspects related to robot programming.

The objective of this thesis was to develop a user-friendly and scalable machine tending robot program that enables the robot to perform machine tending as independently as possible. This work was carried out as part of a broader development project.

The robot's program was developed almost entirely through offline programming, utilizing ABB's RobotStudio software and the digital twin of the robot prototype. The programming of this prototype aimed to follow best practices and maintain a clear program structure. Additionally, a user-friendly user interface was developed, allowing the robot operator to manage the robot's operations and system data. All robot functionalities were programmed using the RAPID programming language. The thesis also presented practical programming examples.

The outcome of this thesis was a prototype program for a machine tending robot, enabling the collaborative robot to perform machine tending tasks based on operator inputs. Initial usability tests indicated that the program's functions operated correctly, and the robot's work cycle proceeded smoothly. The program developed in this thesis serves as a foundation for future robot software development.

Keywords: Machine tending, robotics, programming

## ALKUSANAT

Haluaisin osoittaa kiitokset perheelleni, ystäväilleni sekä Satakunnan ammattikorkeakoulun RoboAI-akatemian ohjaajille, jotka ovat olleet tukenani opiskelujeni ajan. Erityiskiitokset tämän haastavan ja mielenkiintoisen opinnäytetyön toimeksiantajalle Sermatech Oy:lle sekä Jani Uusitalolle hänen syventävistä robottien ohjelmoinnin opeistaan.

# SISÄLLYS

1 JOHDANTO .....	6
2 KONEPALVELU.....	8
2.1 CNC-koneista yleisesti .....	8
2.2 Tavallisimmat CNC-koneiden perustyytit.....	9
2.3 CNC-työstökeskus.....	11
3 ROBOTIT TEOLLISUUDESSA.....	12
3.1 Teollisuusrobotit .....	13
3.2 Yhteistyörobotit.....	14
3.3 Robottien turvallisuus .....	16
4 ROBOTTIEN TYÖKALUT .....	18
4.1 Tarttujat .....	18
4.1.1 Servo-sähköiset tarttujat .....	19
4.1.2 Hydrauliset tarttujat.....	19
4.1.3 Pneumaattiset tarttujat.....	20
4.1.4 Alipainetarttujat .....	21
4.1.5 Magneettiset tarttujat .....	22
4.2 Prosessityökalut .....	22
4.2.1 Hitsaustyökalut .....	23
4.2.2 Hiontatyökalut.....	23
4.2.3 Mittaustyökalut.....	24
4.2.4 Leikkaustyökalut .....	25
4.2.5 Työkalunvaihtajat.....	26
5 ROBOTTIEN OHJELMOINTI .....	27
5.1 Robotin ohjelmasuunnittelu .....	28
5.2 Muuttujat ja ohjausrakenteet .....	30
5.3 Robotin liiketyypit .....	33
5.4 Ohjelman rakenne ja suoritus.....	35
5.5 RAPID .....	36
6 KONEPALVELUROBOTIN OHJELMAN TOIMINTA.....	38
6.1 Pääohjelma .....	41
6.2 Aliohjelmat.....	43
6.3 Käyttöliittymä .....	47
7 YHTEENVETO.....	52
LÄHTEET.....	54

## 1 JOHDANTO

Teollisuuskoneet, kuten CNC-työstökoneet ja ruiskuvalukoneet ovat suuressa roolissa nykypäivän valmistusprosesseissa. Manuaalinen konepalvelu on työntekijöille paitsi fyysisesti rasittavaa ja toistuvaa myös aikaa vievää ja mahdollisesti vaarallista. Konepalvelun automatisointi antaa kaikenkokoisille valmistajille mahdollisuuden parantaa tuottavuutta ja turvallisuutta samalla kun operatiivisia kustannuksia voidaan alentaa. Erilaisille valmistaville yrityksille automaatio on yhä tärkeämpi tekijä kilpailukyvyn säilyttämisessä. Pula osavista työntekijöistä sekä tuotantosykliden lyhentämiseen liittyvä kaupallinen paine pakottavat yritykset löytämään keinoja toimintansa laajentamiseksi ilman suuria investointeja. Yhteistyörobotit tarjoavat yhden ratkaisun teollisten prosessien, kuten konepalvelun ja hitsauksen automatisointiin. Yhteistyörobotit myös mahdollistavat nopeamman ja joustavamman käyttöönoton verrattuna perinteisiin teollisuusrobotteihin. (Universal Robots, 2023.)

Tämän opinnäytetyön tavoitteena oli kehittää helppokäyttöinen ja laajennettavissa oleva konepalvelurobotin prototyypin ohjelma sekä käyttäjäystävällinen käyttöliittymä, jonka kautta robotin operaattori suorittaa tarvittavia toimintoja. Yhteistyörobotin ja työssä tehtävän ohjelman avulla voidaan suorittaa konepalvelua itsenäisesti ilman ihmisen läsnäoloa, kunhan robotin toimintaan liittyvät paikoituspisteet ja kappaleaihioiden tiedot on määritetty ohjelmaan oikein. Prototyypin ohjelma tarjoaa hyvän pohjan ohjelmakehityksen jatkamiselle ja monien muiden toiminnallisuuksien laajuuden vuoksi tämän opinnäytetyön aihe on rajattu vain ohjelmointiin robotin toiminnan ja käyttöliittymän osalta.

Tässä opinnäytetyössä käsitellään muutamia valmistavan teollisuuden koneita, erilaisia robotteja ja niiden monipuolisia työkaluvaihtoehtoja, yleisiä asioita robottien ohjelmoinnista ja ABB:n robottien käyttämää ohjelmointikieltä sekä konepalvelurobotin prototyypin ohjelman toimintaa. Tämän työn sisältö

tarjoaa lukijalleen hyvän yleiskäsityksen robotiikasta ja robottien ohjelmoinnista sekä konepalvelun automatisoinnista. Nämä aiheet liittyvät olennaisesti työssä kehitetyn prototyypin toimintaan.

## 2 KONEPALVELU

Konepalvelulla viitataan yleisesti erilaisten teollisuudessa käytettävien tuotantokoneiden valvontaan ja käyttöön. Usein konepalveluprosessissa operaattori asettaa raakamateriaalin esimerkiksi työstö- tai ruiskuvalukoneeseen, valvoo työstötapahtumaa ja ottaa valmistuneen kappaleen pois koneesta. Koneen operaattorin tehtäviin kuuluu materiaalien käsittelyn lisäksi koneen toiminnan seuranta ja valmiiden kappaleiden laadunvarmistus. (What Is Machine..., n.d.)

Konepalvelun robotisoinnilla voidaan saavuttaa korkeampi tuottavuus sekä parantaa ihmisten työoloja. Erilaisten koneiden käyttöaste nousee, koska robotin suorittaessa materiaalien syöttöä ja poistoa, tulee työkiertoihin vähemmän katkoja, eivätkä robotit tarvitse taukoja. Robotti voidaan joissain tapauksissa myös jättää tekemään tehtäviään valvomatta, jolloin mahdollistetaan koneiden käyttö, vaikka muu henkilökunta on jo poistunut paikalta ja välttyään ylimääräisiltä henkilöstökustannuksilta ylitöiden ja lisähenkilöstön palkkaamisen osalta. (What Is Machine..., n.d.) Robotin käyttäminen yksinkertaisissa ja toistuvissa työtehtävissä vähentää onnettomuusriskejä ja vapauttaa ihmistyövoiman muihin tehtäviin (How To Robot, 2020).

### 2.1 CNC-koneista yleisesti

Tänä päivänä CNC-koneilla valmistetaan erilaisia osia lähes jokaiselle toimialalle ja useista eri materiaaleista, kuten metalleista, muoveista ja puusta. Lyhenteenä "CNC" tulee englannin kielen sanoista Computer Numerical Control. Tämä lyhenne suomennettuna tarkoittaa tietokoneen numeerista ohjausta. CNC-työstöprosessissa valmistetaan erilaisia osia kiinteistä kappaleista tietokoneohjatulla työstökoneella. CNC-koneet toimivat yleensä tietokoneavusteisella valmistuksella CAM (Computer-aided Manufacturing) tai tietokoneavusteisella suunnittelulla CAD (Computer-aided Design). CAM-ohjelmiston avulla käyttäjät voivat luoda "työpuun" työnkulun järjestämiseksi, asettaa työkalupolut ja suorittaa leikkuusimulointeja ennen varsinaista leikkausta. CAD-ohjelmistot ovat useimpien CNC-projektien lähtökohta. Markkinoilla on saatavilla

monia erilaisia CAD-ohjelmistoja, mutta kaikkia niitä käytetään suunnitteluun. Suosittuja CAD-ohjelmia ovat muun muassa AutoCAD, SolidWorks ja Rhino3D. Erilaisilla ohjelmistoilla voidaan kirjoittaa G-koodia, jota CNC-koneissa oleva ohjain pystyy lukemaan. Ohjaimen sisältämä tietokoneohjelma tulkitsee mallia ja liikuttaa työstettävää kappaletta ja/tai leikkaustyökaluja useammalla eri akselilla mallinnetun muodon leikkaamiseksi työkappaleesta. Automatisoituna leikkausprosessi on huomattavasti tarkempi ja nopeampi, kuin työkappaleiden ja työkalujen liikuttaminen manuaalisesti, mikä tehdään vanhemmissa laitteissa vivuilla ja vaihteilla. Nykyaikaisissa CNC-koneissa on useita eri työkaluvaihtoehtoja ja niillä pystytään tekemään erilaisia leikkauksia. Työstökoneessa olevien liiketasojen eli akselien määrä sekä työkalujen tyypit, joita kone pystyy koneistuksen aikana käyttämään, määrittävät kuinka monimutkaisia kappaleita pystytään valmistamaan. Nykyään erilaisia CNC-koneita ja niiden valmistajia on lukematon määrä. Lyhyesti CNC-koneet ovat työstökoneita, joilla poistetaan tai lisätään materiaalia ohjaimen G-koodilla ohjatulla tavalla. (CNC Machines, 2022.)

## 2.2 Tavallisimmat CNC-koneiden perustyyppit

CNC-koneet voidaan yleisesti jakaa viiteen eri perustyyppiin, joiden toiminta perustuu joko materiaalin poistamiseen työkappaleesta tai sen lisäämiseen. Ohessa käydään läpi lyhyesti yleisimmät CNC-konetyypit:

- Jyrsin: Manuaalisissa jyrsinkoneissa käytetään käsipyöriä ja johtoruuveja leikkuutyökalujen niveltämiseksi työstettävään kappaleeseen. Jyrsinkone liikuttaa erittäin tarkkoja kuularuuveja ohjelmoituihin koordinaatteihin. Tämän tyyppisiä koneita on suuri valikoima eri kokoisina ja tyyppinä, ja ne voivat toimia useilla eri akseleilla. (CNC Machines, 2022.)
- Sorvi: Tämän tyyppinen CNC-kone pyörittää työstettävää työkappaletta ja siirtää leikkaustyökalun kappaleeseen kiinni. Perinteisessä sorvissa on 2 akselia, mutta siihen voidaan lisätä muita akseleita lisäämään

työstön monimutkaisuutta. Työstettävä materiaali pyörii sorvin karalla ja painautuu hioma- tai leikkaustyökalua vasten, jolla saadaan tehtyä haluttu muoto. Sorveja käytetään valmistamaan symmetrisiä kappaleita, kuten palloja, kartioita tai sylintereitä. Monet CNC-koneet ovat monitoimisia ja yhdistävät erilaisia leikkaustyypppejä. (CNC Machines, 2022.)

- 3D-tulostin: 3D-tulostin on laite, jolla valmistetaan kolmiulotteisia kappaleita lisäämällä materiaalia kerroksittain. Tämä prosessi tunnetaan myös nimellä lisäävä valmistus. Tulostaminen mahdollistaa sellaisten monimutkaisten muotojen ja rakenteiden valmistuksen, joita olisi perinteisillä koneilla vaikeita tai jopa mahdottomia toteuttaa. Tulostusprosessi aloitetaan viipaloimalla työkappaleen 3D-malli erillisellä viipalointiohjelmistolla ohuiksi kerroksiksi. Ohjelmisto muuttaa tämän viipaloidun mallin G-koodiksi, jonka avulla laite tulostaa kappaleen. 3D-tulostuksessa yleisin materiaali on erilaiset muovit. (Sulkanen, 2024.)
- Leikkuri: CNC-leikkureita käytetään usein leikkaamaan kappaleita, joiden mittasuhteet ovat suuria. Materiaaleina käytetään yleensä metallia, puuta ja muovia levy muodossa. Leikkurit toimivat 3-akselisessa koordinaatistossa, jotta ne voivat leikata kolmella akselilla. Leikkureita kuitenkin löytyy jopa 6-akselisina. CNC-plasmaleikkuri käyttää levy materiaalien leikkaamiseen kiihdytettyä kuumaa plamasuihkua. (CNC-Machines, 2022.)
- Ladontakone: Ladontakoneet toimivat samalla periaatteella kuin leikkurit, mutta materiaalin poistamisen sijaan kone poimii pienillä suuttimilla komponentteja alipainetta hyödyntäen. Poiminnan jälkeen komponentit siirretään ja lasketaan haluttuun paikkaan. Näitä koneita käytetään paljon elektroniikkateollisuudessa esimerkiksi tietokoneiden emolevyjen ja muiden sähkökomponenttien kokoonpanossa. (CNC Machines, 2022.)

CNC-koneet ovat erittäin monipuolisia ja kykenevät suorittamaan monenlaisia tehtäviä. Nykyään tietotekniikkaa voidaan hyödyntää lähes kaikissa kuviteltavissa olevissa koneissa. CNC-laitteiden automaatio poistaa tarpeen

manuaaliselle ohjaukselle, sillä koneen osia ei tarvitse liikuttaa käsin halutun tuloksen saavuttamiseksi. Modernit CNC-koneet pystyvät aloittamaan raaka-aineesta, kuten teräksisestä ahiosta, ja työstämään siitä erittäin monimutkaisen osan tarkoin toleranssein ja erinomaisella toistotarkkuudella. (CNC Machines, 2022.)

### 2.3 CNC-työstökeskus

CNC-työstökeskus on monikäyttöinen koneistuslaite, joka pystyy suorittamaan tarkasti erilaisia työstötoimenpiteitä, kuten porausta, jyrshintää ja sorvausta ja saavuttamaan korkealaatuisen pinnan viimeistelyyn. Koneen toiminta ohjelmoidaan G-koodilla, jolla hallitaan sen liikkeitä ja nopeutta, jäähdytysnesteen käyttöä, työkalun valintaa sekä työstöarvoja. Työkalut voidaan kiinnittää joko pystytai vaakakaralla koneen mallista riippuen. CNC-työstökeskuksen terät jäähdytetään kierrättämällä jäähdytysnestettä laitteen sisällä. (Ye, 2023.)

Työstökeskusten erityisominaisuutena on automaattinen työkalunvaihto ja työmakasiini. Nämä mahdollistavat lyhyemmät sykliajat työstöjen välillä ja vähentävät manuaalista työtä. Automaattinen työkalunvaihto tehostaa työstöprosesseja ja tekee niistä joustavampia. (Ye, 2023.)

### 3 ROBOTIT TEOLLISUUDESSA

Valmistava teollisuus voidaan jakaa kahteen pääryhmään: prosessiteollisuuden ja kappaletavarateollisuuteen. Näistä kappaletavarateollisuus muodostaa teollisuusrobottien suurimmat markkinat, vaikka robotteja hyödynnetään laajasti koko valmistavassa teollisuudessa. Teollisuusrobotiikan käyttö alkoi valmistavassa teollisuudessa jo 1960-luvulla. Alkuvaiheessa markkinoille tuotiin hydraulikkaan perustuvia teollisuusrobotteja, mutta nämä syrjäytyivät 1970-luvun puoliväliin mennessä sähkötekniikkaa hyödyntävien robottien tieltä. Kehitys oli kuitenkin hidasta, ja robotteja käytettiin alkuun vain rajatuissa soveluksissa autoteollisuuden toimiessa erityisesti suunnannäyttäjänä. (Leinonen, 2023, s. 6.)

Teollisuusrobotiikka on kohdannut monia vastoinkäymisiä historiansa aikana. Robotit herättivät aluksi pelkoa: niiden ajateltiin vievän ihmisten työpaikkoja, aiheuttavan turvallisuusriskejä tai jopa ottavan vallan ihmisiltä. Ajan myötä asenteet ovat kuitenkin muuttuneet. Nyt, 2020-luvulla, teollisuusrobotit nähdään ihmisten työtaakkaa keventävinä ja työturvallisuutta parantavina työkaluina. Teknologiset innovaatiot ovat usein merkinneet uuden, kehittyneemmän aikakauden alkua, ja teollisuusrobotiikka on noussut valmistavan teollisuuden megatrendiksi. Robotiikan globaali kasvu on vaikuttavaa, sillä ala kasvaa vuosittain keskimäärin 15–20 %, ja vuosittain maailmassa toimitetaan noin 500 000 uutta robottia. Robotteja hyödynnetään monipuolisemmin ja yhä nopeammassa tahdissa erilaisissa työtehtävissä. Valmistava teollisuus on kokonaisuutena merkittävässä murroksessa: aiemmin manuaaliseen työhön perustunut työvoimavaltainen ala ottaa nyt suuria askeleita kohti teknologian tarjoamia mahdollisuuksia. (Leinonen, 2023, s. 6–7.)

Monet valmistavan teollisuuden työpaikat koetaan ihmisten mielestä raskaiksi, likaisiksi ja epämiellyttäviksi. Lisäksi teollisuus kärsii osaavien työntekijöiden puutteesta ja robotiikan koetaan olevan vastaus näihin ongelmiin. Teollisuuden viimeaikaisiin merkittäviin edistysaskeliin kuuluu valmistusprosessien joustavuuden lisääntyminen. Tämä kehitys on mahdollistanut aiempaa

pienempien sarjakokojen tuottamisen sekä entistä monimutkaisempien tuotteiden koneellisen valmistuksen. Robottien ohjelmointiprosessi on muuttunut käyttäjäystävällisemmäksi, ja tuotantosoluissa ja tuotantolinjoissa hyödynnettävien lisälaitteiden teknologinen kattavuus on laajentunut merkittävästi. Eri-laisten teknologioiden kuten konenäön, anturien ja robottien monipuolisten työkalujen avulla nykyaikaiset tuotantosolut kykenevät aina vain haastavampiin tehtäviin. (Leinonen, 2023, s. 6–7.)

### 3.1 Teollisuusrobotit

Teollisuusroboteiksi voidaan lukea uudelleen ohjelmitavissa olevat kappaaleenkäsittelylaitteet eli manipulaattorit, jotka ovat yleiskäyttöisiä ja joissa on vähintään kolme vapaasti ohjelmitavaa vapausastetta ja työkalu. Robotiikan alalle määritelmiä on luonut kansainvälinen robotiikkajärjestö IFR (International Federation of Robotics) ja niitä on myös standardoitu standardiin ISO 8373:2021. Näiden mukaan esimerkiksi varastoissa usein käytettävät automaattiset hyllystöhissit tai kappaletavaratuotannossa kiinteän automaation manipulaattorit eivät ole robotteja. Lähes 90 % teollisuusroboteista on käsivarsirobotteja, joilla on kuusi vapausastetta. Näiden etuna on yleiskäyttöisyys ja ne voidaan asentaa esimerkiksi lattialle, seinälle tai kattoon. Hyötykuormat, joita käsivarsirobotit pystyvät käsittelemään, vaihtelevat paljon kevyimmistä 0,5 kg painoista aina 2000 kg asti. (Lempiäinen, 2023, s. 17.)

Käsivarsirobottien lisäksi löytyy erilaisella rakenteella toimivia teollisuusrobotteja, joita ovat SCARA-, delta- sekä portaalirobotit. Suoraviivaisiin kokoonpanotehtäviin, jotka tapahtuvat ylhäältä alaspäin, on kehitetty SCARA-robotit (Selective Compliance Assembly Robot Arm), jotka omaavat 4 vapausastetta. Mukautuminen, compliance, viittaa robotin käsivarren liikkeen sopeutumiseen tuotteen sovitteen vaatimuksiin. Tämä mahdollistaa tiukkojenkin sovitteiden kokoonpanon, sillä robotti mukautuu sovitteen ohjaukseen, vaikka paikoituk- sessa esiintyisi virheitä joko robotin tai kiinnittimen osalta. Mukautuminen voidaan toteuttaa kahdella tavalla: vapauttamalla robotin vaakasuuntaisten lii- keaksien ohjaus siten, että ne sopeutuvat pystysuuntaiseen liikkeeseen ja

sovitteen etenemiseen, tai hyödyntämällä erillistä passiivista toimilaitetta, joka on integroitu työkalun rakenteeseen. (Lempiäinen, 2023, s. 17–18.)

Delta-robotti on erityisesti tarkkaan ja nopeaan liikkeeseen suunniteltu teollisuusrobotti, joka yleensä koostuu kolmesta tai neljästä rinnakkaisesta tukivaresta, joihin robotin työkalu on kiinnitetty. Robotin jokainen tukivarsi on moottorihjattu ja moottorit sijaitsevat robotin rungossa. Näiden moottoreiden avulla koordinoidaan tukivarsien liikkeitä ylös- ja alaspäin, jolloin nivelet työntyvät sisään ja ulospäin. Näiden liikkeiden avulla robotin työkalu pystyy liikkumaan X-, Y- ja Z-akseleiden suuntaisesti, jolloin muodostuu sylinterimäinen työalue. Painopisteen pitäminen paikallaan työalueen yläpuolella mahdollistaa robotille suuret käyttönopeudet ja kiihtyvyydet. Jotkin delta-robotit kykenevät jopa 300 liikkeeseen minuutissa, mutta käsittelevät pääasiassa kevyitä ja geometrisesti yksinkertaisia kappaleita. Näiden ominaisuuksiensa vuoksi ne ovat suosittuja elektroniikka-, elintarvike- ja lääketeollisuudessa tuotteiden pakkauksessa ja kokoonpanossa. (Robots Done Right, n.d.a.)

Laajalle toiminta-alueelle on kehitetty tyypillisesti kolme liikeakselia omaavat portaalirobotit. Rakenteeltaan nämä robotit ovat yksinkertaisia ja laatikkomaisia, joissa robotin käsivarsi liikkuu kiinteää ylärataa pitkin. X- ja Y-akselit mahdollistavat vaakasuuntaiset liikkeet yläraiteella, kun taas Z-akseli vastaa pystysuuntaisesta liikkeestä. Näiden robottien monipuolisuus, korkea kuormankantokyky ja ohjelmoitavuus tekevät niistä ihanteellisia tarkkuutta ja tehokkuutta vaativiin tehtäviin, kuten lavaamiseen, kokoonpanoon, pakkaamiseen ja varastointiin teollisissa ympäristöissä. (Fuyu, 2024.)

### 3.2 Yhteistyörobotit

Yhteistyörobotit ovat robottikäsivarsia, jotka on suunniteltu toimimaan turvallisesti ja tehokkaasti ihmisten rinnalla samassa työtilassa ilman erillisiä turvatekijöitä, kuten turva-aitoja tai tutkia. Toisin kuin perinteiset teollisuusrobotit, jotka pidetään yleensä ihmisistä erillään turvallisuussyistä, yhteistyörobottien anturointi ja ohjelmisto mahdollistavat tiedonvälityksen robotin

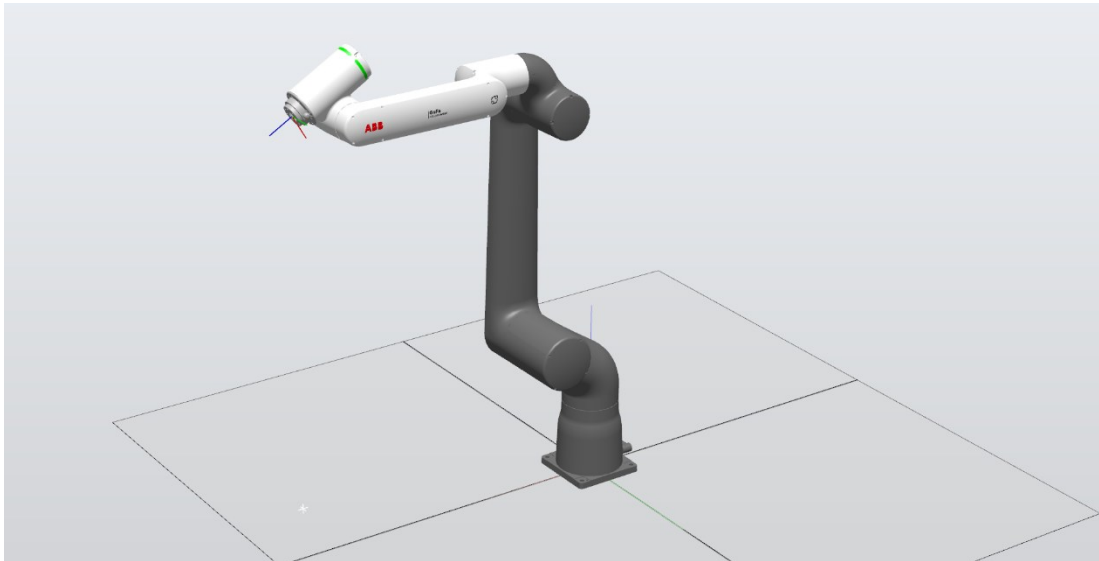
ohjausjärjestelmään mahdollisista törmäystilanteista, jolloin ihmisen ja yhteistyörobotin on mahdollista työskennellä yhdessä turvallisesti. Lisäksi näiden laitteiden muotoilu on pyöreämpää, jolloin tapaturmariski pienenee törmäysten osalta. (Lempiäinen, 2023, s. 28–29.)

Yhteistyörobottien keskeisiä ominaisuuksia ovat helppo ohjelmitavuus, joustavuus, liikuteltavuus ja kyky suorittaa erilaisia tehtäviä ihmisen kanssa. Ohjelmointi toteutetaan usein opettamalla pisteitä robotille, jolloin varsinainen ohjelmakoodi jää piiloon käyttäjältä. Yhteistyörobottien kevyt rakenne ja suhteellisen hitaat liikenopeudet, verrattuna perinteisiin teollisuusrobotteihin, tekee niistä helposti siirrettäviä eri työtehtävien välillä, eikä robotin kiinnitysalustaan ja lattiaan kohdistu suuria voimia robotin liikkeiden vuoksi. Nämä asiat mahdollistavat robotin rungon kiinnityksen lattiaan kevyillä ratkaisuilla, eikä lattia-pinta-alaa tarvita paljoa. (Lempiäinen, 2023, s. 28–29.)

Teollisuudessa yhteistyörobottien määrä on pysynyt vielä toistaiseksi maltillisena verrattuna tavanomaisiin teollisuusrobotteihin. Kansainvälisten tilastojen mukaan noin 8 % robotti-investoinneista on ollut yhteistyörobotteja ja Suomessa se tarkoittaa noin 40 kappaleen vuosittaista kauppaa. Pääasiassa yhteistyörobotteja on mennyt oppilaitoskäyttöön ja tehtaiden ensimmäisiksi roboteiksi pienempien alkuinvestointikulujen vuoksi. (Lempiäinen, 2023, s. 29.)

CRB 15000 GoFa (Kuva 1) on ABB:n kehittämä edistynyt kuusiakselinen yhteistyörobotti, joka on suunniteltu toimimaan ihmisen rinnalla turvallisesti ja tehokkaasti erilaisissa teollisissa sovelluksissa. Sen maksimikuormituskapasiteetti on 5, 10 tai 12 kilogrammaa sekä suurin ulottuma 950, 1270 tai 1520 millimetriä robotin mallista riippuen. Robotin työkalukeskipisteen suurin nopeus on 2,2 m/s, mikä tekee siitä nopeamman kuin moni muu saman luokan yhteistyörobotin. Robotin saa konfiguroitua ja ohjelmoitua FlexPendant-käsihjaimen avulla erilaisten graafisten sovellusten kautta. Ohjelmointia ja pisteiden opetusta helpottaa robotin Lead-Through-ominaisuus, jossa robotin saa viidennessä akselissa olevaa nappia painamalla opetustilaan, jolloin sitä voi liikuttaa käsin paikasta toiseen. Joustavuutensa ja monien ominaisuuksiensa

vuoksi robotti soveltuu moniin eri tehtäviin, kuten konepalveluun, kokoonpanoon, pakkaamiseen ja materiaalinkäsittelyyn. (ABB, 2023, s. 1–4.)



Kuva 1. ABB CRB 15000 10/1.52 -yhteistyörobotti simuloidussa ympäristössä.

### 3.3 Robottien turvallisuus

Kaikki robottijärjestelmät on suunniteltava turvallisuusstandardien SFS-EN ISO 10218-1:2011 ja SFS-EN ISO 10218-2:2011 mukaisesti siten, että ihmisten altistuminen vaaroille voidaan välttää. Järjestelmien turvallistaminen alkaa lähtökohtaisesti vaarojen poistamisella, mutta mahdollisuudet tähän ovat rajallisia johtuen vaarojen muodostumisesta, mikä yleensä liittyy robotin nopeisiin liikkeisiin. Järjestelmissä, joissa käytetään teollisuusrobottia, turvallistaminen tarkoittaa käytännössä robotin ja siihen liittyvien oheislaitteiden ja koneiden eristämistä ihmisestä, koska robottien nopeiden liikkeiden törmäysvoimaa ei voida kontrolloida ja estää riittävän nopeasti ja tarkasti. Lisäksi järjestelmässä olevat oheislaitteet ja robotin työkalut voivat aiheuttaa ihmiselle vaaratilanteita. Joissain tilanteissa ihmisen pääsy järjestelmään, kuten järjestelmän huollon tai robotin ohjelmoinnin aikana on välttämätöntä, mutta turvallistamista ei silloin voi toteuttaa eristämällä ihmistä vaaroilta. Materiaalin ja ihmisten täytyy aina välillä kulkea järjestelmän sisällä, joten täysi eristäminen ei ole mahdollista. Tällöin turvallisuutta voidaan parantaa erilaisilla turvalaitteilla. (Latokartano, 2023, s. 92–106.)

Yleisin tapa estää ihmisten pääsy robotin työalueelle on turva-aidat. Aitojen käyttö on kustannustehokas ja helppo turvaratkaisu tuotantosoluissa liikkumisen rajoittamiseen, mutta suunnitteluvaiheessa tulee huomioida aitojen väärinkäyttö ja tuotantoprosessissa mahdollisesti ulospäin suuntautuvat vaarat, kuten kipinät ja lentävät esineet. Robottisolujen käytössä on tärkeää varmistaa materiaalin ja ihmisten turvallinen liikkuminen soluun ja sieltä pois. Materiaalin kuljetukseen tarkoitetut aukot valvotaan esimerkiksi valoverhoilla tai anturijärjestelmillä, joilla ihmiset erotetaan materiaalista. Ihmisten kulkutiet, kuten ovet tai avoimet aukot, varustetaan turvalaitteilla, kuten turvarajakytkimillä, turvavaloverhoilla tai laserskannereilla. Turvajärjestelmän tulee reagoida antureilta tuleviin pysähtymissignaaleihin nopeasti, jotta robotti pysähtyy ennen kuin ihminen pääsee vaara-alueelle. Näiden turvalaitteiden valintaan vaikuttavat valvotavan alueen koko, ympäristöolosuhteet, turvallisuusvaatimukset, helppokäyttöisyys ja kustannukset. Kaikkien valvontaan käytettyjen anturien on oltava turvakäyttöön tarkoitettuja. (Latokartano, 2023, s. 105–112.)

Yhteistoiminnallisen robottisolun turvallisuus perustuu käytetyn robotin ominaisuuksiin ja riskien arviointiin. Jos solussa hyödynnetään yhteistyörobottia, sen suunnittelussa otetaan huomioon mahdollisuus törmäykseen ihmisen kanssa. Tällöin turvalaitteita ei välttämättä tarvita, koska robotti on suunniteltu toimimaan turvallisesti ihmisen läheisyydessä. Riskikartoituksessa keskitytään arvioimaan törmäysten aiheuttaman vahingon vakavuutta, ei niinkään törmäysten tapahtumisen todennäköisyyttä. Pelkkä yhteistyörobotin käyttäminen ei tee sovelluksesta turvallista, vaan turvallistamisessa täytyy ottaa myös huomioon robotin työkalut ja oheislaitteet sekä itse työprosessi. Yhteistyöroboteilla voidaan toteuttaa sovelluksia, joissa robottia siirretään työpisteeltä toiselle, mutta riskienarviointi täytyy toteuttaa jokaisen työkohteen osalta erikseen. Normaalissa tuotantokäytössä riskienarvioinnin vastuu siirtyy suunnittelijalta robotin käyttäjälle. Tällaisten robottisolujen turvallistamisen suunnittelussa voidaan käyttää apuna standardia SFS-EN ISO 10218-2 ja teknistä spesifikaatiota ISO TS 15066. (Latokartano, 2023, s. 112–115.)

## 4 ROBOTTIEN TYÖKALUT

Robotin työkalu kiinnitetään yleensä robotin työkalulaippaan, ja tätä työkalua kutsutaan englanniksi nimellä End-Of-Arm Tool (EOAT). Tämä termi viittaa erityisesti laippaan kiinnitettävään työkaluun erottaen sen muista työkaluista, jotka voivat sijaita robotin liiketilassa. Työkalut jaetaan pääasiassa kahteen ryhmään: tarttujat ja prosessityökalut. Tarttujat mahdollistavat robotin tarttumisen kappaleisiin, niiden siirtämisen paikasta toiseen tai liittämisen muihin osiin. Prosessityökalut sen sijaan tekevät robotista aktiivisen työstöprosessin toteuttajan. Esimerkiksi, kun robottiin kiinnitetään hitsauskolvi, se voi hitsata, ja maaliruiskun avulla siitä tulee maalari. Työkalun valinnassa tärkeitä tekijöitä ovat työkalun paino ja painopisteen sijainti. Jokaisella robotilla on määritelty hyötykuorma eli suurin sallittu kuorma, joka takaa valmistajan lupaaman nopeuden ja tarkkuuden. Hyötykuorma ilmoitetaan tyhjälle robotille, joten työkalun paino sekä kaikki muu robottiin kiinnitettävä varustus vähennetään tästä. Lisäksi työkalun painopisteen etäisyys laipasta vaikuttaa merkittävästi. Jos painopiste on liian kaukana, ranteen niveliin kohdistuva vääntömomentti rajoittaa robotin tehokasta hyötykuormaa. Näin ollen työkalun valinta vaikuttaa suoraan robotin suorituskykyyn ja toimintakykyyn. (Liuha ym., 2023, s. 202–203.)

### 4.1 Tarttujat

Nykyään markkinoilta löytyy useita erilaisia valmiita tarttujaratkaisuja eri valmistajilta. Tarttuja on usein robottijärjestelmän keskeinen osa, jolla sarjatuo- tettu robottikäsi sovitetaan tiettyyn prosessiin. Joissain tilanteissa riittää, että mekaaniseen tarttujamekanismiin lisätään sopivat sormet, mutta toisinaan tarttuja täytyy suunnitella alusta alkaen prosessiin sopivaksi. Yleisiä vaatimuksia ovat yksinkertainen rakenne, pieni koko ja paino sekä luotettava tartuntakyky. (Liuha ym., 2023, s. 203.)

#### 4.1.1 Servo-sähköiset tarttijat

Sähköiset tarttijat tarjoavat mahdollisuuden erittäin tarkkaan otteen, voiman, asennon ja nopeuden hallintaan, mikä voi olla haastavaa muille tarttujatyypeille. Sähköisiä tarttujia, joita kutsutaan myös servo-sähkötarttujiksi, käytetään laajasti monissa yhteistyörobotisovelluksissa, kuten konepalvelussa ja kappaleiden poimimisessa ja sijoittamisessa. Tarttujan sormia ohjataan sähkömoottorin avulla. Vaikka sähköisiä tarttujia on yleisimmin saatavilla kahden tai kolmen leuan kokoonpanoina, kolmileukaiset tarttijat ovat yleensä suosittuja käsiteltäessä pyöreitä tai sylinterimäisiä esineitä. (PFA Inc, n.d.)

Useimmat sähköiset tarttijat on varustettu mikroprosessorilla, joka mahdollistaa tartuntavoiman ja -nopeuden säätämisen. Näistä tarttujista usein löytyvän voima-anturin avulla saadaan kyky määrittää kullekin tehtävälle optimaalinen voima. Koska sähkömoottorin virta on suoraan verrannollinen sen tuottamaan vääntömomenttiin, tartuntavoimaa voidaan hallita tarkasti. Tarttujan sormet voidaan ohjelmoida toimimaan hellävaraisesti herkkien kappaleiden kanssa tai luomaan tiukka ote kestävämpien kappaleiden käsittelyssä. Tämä joustavuus tekee sähköisistä tarttujista monipuolisen valinnan erilaisten osien käsittelyyn. (PFA Inc, n.d.)

Sähköisten tarttujen suurin vahvuus on niiden tarkkuus. Toisin kuin perinteisemmät tarttijat, sähköiset mallit voivat minimoida leukojen liikkeen ja käyttää vain tarvittavan tilan tarttuakseen osaan. Lisäksi hyödyntämällä enkoodereita ne voivat tunnistaa, onko kappale onnistuneesti poimittu. Tartuntatilan havaitseminen on ratkaisevan tärkeää monissa sovelluksissa virhetilanteiden estämiseksi. Kun tarttijat pystyvät tekemään tämän itsenäisesti, tarve erilaisille lisäantureille ja laitteille vähenee. Tämä yksinkertaistaa järjestelmän asennusta ja vähentää kokonaiskustannuksia. (PFA Inc, n.d.)

#### 4.1.2 Hydrauliset tarttijat

Hydrauliikka on vähiten käytetty voimansiirtomenetelmä robottien tarttujissa. Hydrauliikalla toimiva tarttuja käyttää kompressoreita pumppaamaan öljyä,

mikä tuottaa tarvittavan voiman tarttujalle. Vaikka hydraulikka ei ole suosituin vaihtoehto, se tarjoaa eniten tartuntavoimaa verrattuna muihin tarttujatyyppeihin. Hydraulisten tarttujien avulla voidaan käsitellä todella raskaita kappaleita usein esimerkiksi autoteollisuudessa. (Robots Done Right, n.d.b.)

Hydraulisilla ja pneumaattisilla tarttujilla on samanlainen perustoimintaperiaate, joka perustuu mäntäjärjestelmien käyttöön. Näitä järjestelmiä on kahdenlaisia: suoratoimiset männät ja männän kiilamekanismit. Suoratoimisessa mäntäjärjestelmässä hydraulinen voima kohdistuu suoraan mäntään, joka on liitetty suoraan tarttujan leukaan tai sormeen. Tämä yksinkertainen rakenne siirtää voiman suoraan kappaleen tarttumiseen. Männän kiilamekanismi puolestaan käyttää hydraulista voimaa, joka vaikuttaa mäntään. Tämä mäntä ohjaa kiilaa, joka välittää voiman tarttujan leukoihin tai sormiin. Kiilamekanismin etuna on sen kyky kasvattaa tartuntavoimaa muuttamatta männän halkaisijaa tai siihen kohdistuvaa painetta. Tämä mahdollistaa suuremman tartuntavoiman kompaktimmassa rakenteessa verrattuna suoratoimiseen mäntärakenteeseen. (Mobile Hydraulic Tips, 2022.)

Hydraulisten tarttujien käytössä pitää myös huomioida niiden korkeat käyttökustannukset ja huoltotarpeet. Nämä tarttijat tarvitsevat suuria kompressoreja, letkuja ja öljyä, mikä lisää järjestelmän käyttökuluja ja vie ylimääräistä tilaa tuotantotiloista. Lisäksi hydrauliset voimansiirtolähteet ovat sotkuisempia ja alttiita öljyvuodoille, jotka voivat olla vaaraksi niin robotille kuin koko tuotantolaitokselle. Tämän vuoksi hydraulisia tarttujia ei voida käyttää puhdastiloissa, ja niiden käyttö rajoittuu sovelluksiin, joissa siisteys ei ole tärkeä tekijä, kuten esimerkiksi autoteollisuudessa. (Robots Done Right, n.d.b.)

#### 4.1.3 Pneumaattiset tarttijat

Pneumaattiset tarttijat hyödyntävät paineilmaa tarttujan leukojen liikkeenohjauksessa. Tämä tarttujatyyppe on tunnettu nopeudestaan ja kyvystään tuottaa vahva ote, mikä tekee niistä soveltuvia tehtäviin, joissa tarvitaan nopeita ja tarkkoja liikkeitä esimerkiksi kokoonpanotehtäviin. Lisäksi pneumaattiset

tarttumat tunnetaan kestävydestään, mikä tekee niistä sopivia toistuviin ja vaativiin teollisuusympäristöihin. Pneumaattisia tarttuvia on saatavilla monissa eri kokoonpanoissa, kuten kahden tai kolmen sormen malleina sekä kulmikkaina tarttuina, mikä mahdollistaa niiden joustavan käytön erilaisissa tartuntatehtävissä. Osa pneumaattisista tarttuista on suunniteltu tarjoamaan erityisen hellävarainen ote, mikä on olennaista herkkien materiaalien ja komponenttien käsittelyssä. (Dorna Robotics, 2023.)

Pneumaattiset tarttumat ovat suosittuja niiden helpon käyttöönoton ja edullisten kustannusten vuoksi. Koska monilla tuotantolaitoksilla on paineilmaa saatavilla jo valmiiksi, pneumaattisten tarttujen käyttö voidaan toteuttaa helposti ja kustannustehokkaasti. Toinen etu tässä voimanlähteessä on se, että voimakas tartuntavoima voidaan saavuttaa pienemmällä tarttujan koolla ja painolla. Tämä mahdollistaa toiminnan myös ahtaissa tiloissa. (Robots Done Right, n.d.b.)

#### 4.1.4 Alipainetarttumat

Alipainetarttumat ovat tehokkaita ratkaisuja tilanteisiin, joissa mekaanisten tarttujen käyttö on haastavaa. Usein ne ovat myös kustannustehokkaampia kuin monet muut tarttumatyypit. Näillä tarttuilla kappaleeseen tartutaan yleensä vain yhdestä suunnasta, ja tartunta tapahtuu imukuppien avulla, jotka on yleensä valmistettu kumista tai muovista. Näiden materiaalien ansiosta imukupit ovat hellävaraisia, eivätkä aiheuta vahinkoa kappaleen pinnalle. Tartuntavoimaa voidaan kasvattaa suurille kappaleille lisäämällä imukuppien määrää, suurentamalla imupinta-alaa tai kasvattamalla alipainetta. Kuitenkin monien imukuppien järjestelmissä on otettava huomioon turvallisuusriski: jos yksi imukuppi irtoaa, se voi aiheuttaa alipaineen häviämisen ja kappaleen putoamisen. Tämän riskin minimoimiseksi imukupit tai niiden ryhmät voidaan jakaa erillisiin alipainekammioihin, jotka toimivat itsenäisesti. (Liuha ym., 2023, s. 207–209.)

Imukupit vaativat yleensä tasaisen, sileän, puhtaan ja ilmatiiviin pinnan toimiakseen tehokkaasti. Imuvoima määräytyy paine-eron ja imupinta-alan tulosta.

Tarttujaan kohdistuvat sivuttaisvoimat on syytä pitää minimissä, koska imukuppien luoma vastus sivuttaisliikkeelle perustuu niiden ja kappaleen väliseen kitkakertoimeen. Tartuntapiste kannattaakin sijoittaa mahdollisimman lähelle kappaleen painopistettä tasapainon varmistamiseksi. Lisäksi tarjolla on alipainetarttuvia, joissa perinteisten imukuppien sijaan käytetään pesusienimäistä materiaalia. Näissä tarttujissa on usein useita tartunta-alueita, joita voidaan ohjata joko aktiivisesti tai passiivisesti. Aktiivisessa ohjauksessa alipaine voidaan suunnata halutuille alueille, kun taas passiivinen ohjaus hyödyntää vastaventiileitä, jotka sulkevat alueet, joissa alipaine ei pysy, eli kappaleeseen ei ole tartuttu kunnolla. Tällaiset alipainetarttijat kestävät paremmin pölyisiä olosuhteita tai epätasaisia kappaleen pintoja. (Liuha ym., 2023, s. 207–209.)

#### 4.1.5 Magneettiset tarttijat

Magneettitarttijat soveltuvat ainoastaan magneettisille materiaaleille. Tarttujan nostovoima riippuu useista tekijöistä, kuten kappaleen materiaalista, muodosta, pinnan laadusta, ilmaraosta ja magneetin lämpötilasta. Työkappaleen tulee tarjota riittävän laaja tartuntapinta, sillä magneettikentän voimakkuus heikkenee nopeasti ilmaraon suurentuessa. Tartunta-aluetta voidaan kuitenkin kasvattaa käyttämällä magneettiin kiinnitettäviä lisäosia, jotka on muotoiltu kappaleen geometriaan sopiviksi. Magneetin avulla kappaleisiin voidaan tarttua nopeasti, mutta irrotusta voi hidastaa jäännösmagnetismi. Kestomagneettipohjaiset tarttijat voivat olla joko sähkö- tai paineilmaohjattuja. Sähkömagneettien avulla magneettikentän suuntaa voidaan vaihtaa, mikä nopeuttaa irrotusta merkittävästi. On kuitenkin huomioitava, että sähkömagneetit lämpenevät käytössä, joten työkierron suunnittelussa on varmistettava, ettei magneetin lämpötila pääse nousemaan liian korkeaksi, koska se voi johtaa magneettisuuden häviämiseen. (Liuha ym., 2023, s. 209–210.)

#### 4.2 Prosessityökalut

Teollisuusrobotit voivat erilaisten tarttujen lisäksi käyttää myös monia erilaisia prosessityökaluja. Nykyteknologian ansiosta robotteihin voidaan liittää

monenlaisia lisälaitteita, jotka tekevät niistä entistä monipuolisempia ja soveltuvampia erilaisiin ja haastaviin työvaiheisiin. Perinteisten tarttujien avulla robotti voi käsitellä työkappaleita tarkasti ja tehokkaasti, mutta monissa tuotantoprosesseissa pelkkä tarttuminen ei riitä. Tällöin robottien kyky toimia työkaluoperaattorina, kuten hitsaajana, hiojana tai jopa mittauslaitteiden käyttäjänä, nousee merkittävään rooliin. Tällainen monipuolisuus ei ainoastaan tehosta tuotantoa, vaan avaa myös uusia mahdollisuuksia automatisoida entistä monimutkaisempia ja tarkkuutta vaativia tehtäviä. (Liuha ym., 2023, s. 213–227.)

#### 4.2.1 Hitsaustyökalut

Robottihitsaus on prosessi, jossa hitsaustyö automatisoidaan robottien avulla. Robotit suorittavat hitsauksen ennalta ohjelmoidun ohjelman mukaisesti, ja ohjelmaa voidaan muokata projektin vaatimusten mukaan. Tämä menetelmä yhdistää automaation ja edistyksellisen hitsausteknologian, mahdollistaen tarkemmat, turvallisemmat ja nopeammat hitsaukset verrattuna perinteisiin menetelmiin. Robottien käyttö vähentää materiaalihävikkiä ja lisää prosessin tehokkuutta. Lisäksi robotit voivat työskennellä haastavissa paikoissa ja suorittaa monimutkaisia hitsauksia nopeammin ja tasalaatuisemmin kuin ihminen manuaalisesti. Tämä vapauttaa työntekijöitä muuhun tuotantoon ja lisää valmistusprosessin joustavuutta. Robottihitsauksessa voidaan hyödyntää monia hitsaustekniikoita, kuten kaari-, vastus-, piste-, TIG-, laser-, plasma- ja MIG-hitsausta. Keskeistä onnistumiselle on tarkasti suunniteltujen hitsausohjelmien ja kiinnitysjärjestelmien käyttö, jotka tukevat kunkin sovelluksen erityistarpeita, kuten esimerkiksi railonseurantaa. (Kemppe, 2024.)

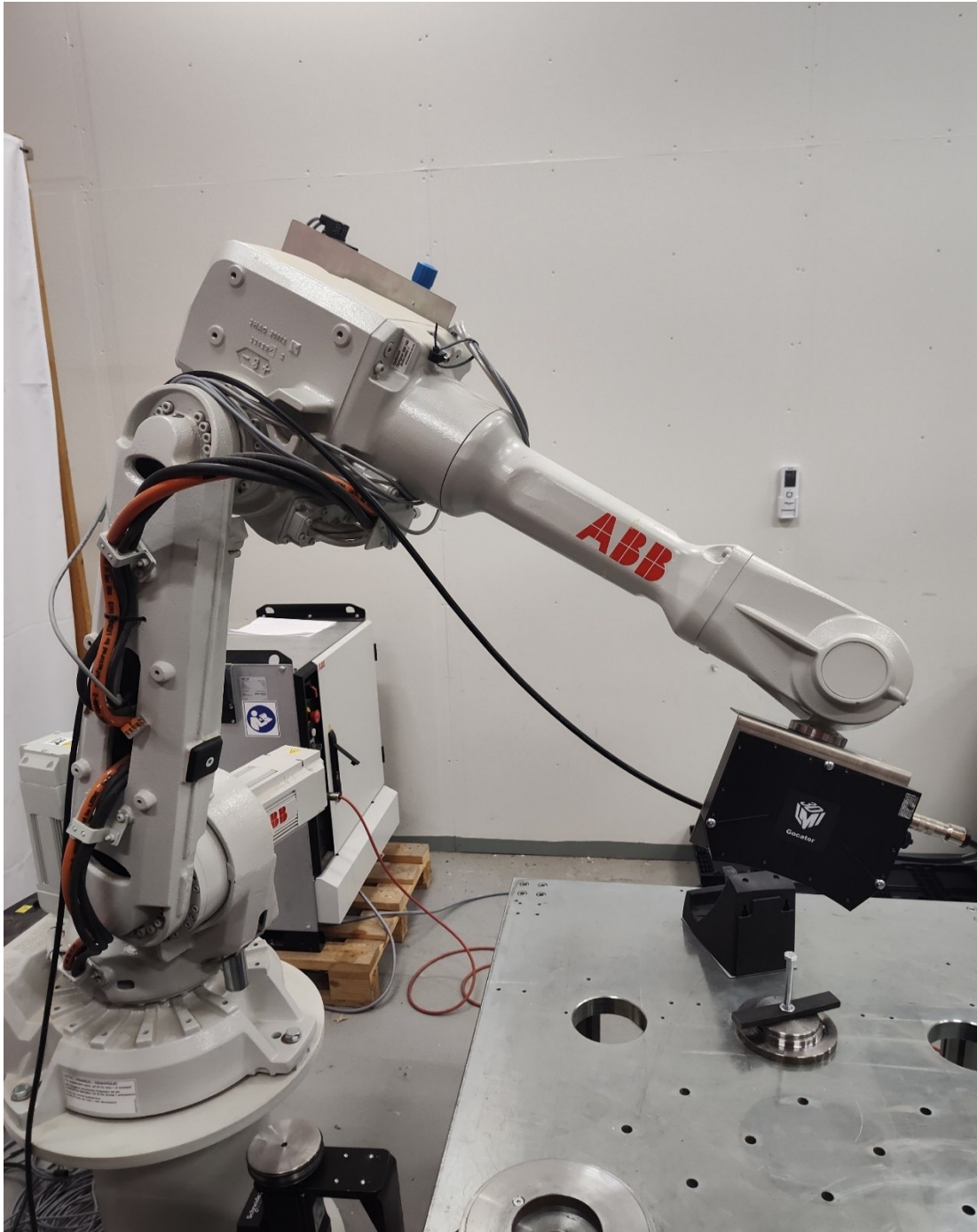
#### 4.2.2 Hiontatyökalut

Kun robotilla poistetaan materiaalia työkappaleista, tähän prosessiin kuuluvat työkalut voidaan jakaa kolmeen pääkategoriaan, jotka ovat nauhahiomakoneet, epäkeskohiomakoneet ja pyörivät työkalut kuten harjat ja laikat. Hiomatyökalujen tyypillisimpinä voimanlähteinä toimivat paineilma- tai sähkömoottorit. Paineilmamoottoriin verrattuna sähkömoottori tarjoaa monia etuja, kuten

tasaisen väännön laajalla kierrosnopeusalueella, luotettavan kierrosnopeuden, työkalun pyörytysuunnan vaihtamisen ja takaisinkytkennän moottorin virrankulutuksesta. Paineilmamoottorilla varustettujen työkalujen parametriikkuna, kontaktivoima ja pyörimisnopeus ovat huomattavasti kapeampia verrattuna sähkömoottoreilla toimiviin työkaluihin. Paineilmamoottoreiden energiatehokkuus on myös huonompi, johtuen paineilman luomisesta sähköllä kompressoria käyttäen ja sen siirtämisestä. Paineilmalla toimivien työkalujen etuna on kuitenkin kevyt paino ja mahdollisuus pienempiin fyysisiin mittoihin verrattuna sähkömoottoreilla varustettuihin työkaluihin. (Liuha ym., 2023, s. 215–216.)

#### 4.2.3 Mittaustyökalut

Robottia voidaan hyödyntää alustana erilaisille mittausprosesseille. Työkalulaippaan kiinnitetty mittalaite voidaan robotin avulla siirtää tarkasti määriteltyyn paikkaan, jolloin mittaustapahtuma suoritetaan aina samalla tavalla. Mittalaitteena voidaan käyttää esimerkiksi kameroita tai muita optisia laitteita, joilla saadaan mitattua työkappaleen muodot hyödyntäen konenäköä tai laserkeilausta (Kuva 2). Lisäksi robottiin voidaan kiinnittää kosketusmittalaitteita, joilla voidaan kappaleita koskettamalla tarkastaa niiden kolmiulotteinen muoto. (Liuha ym., 2023, s. 222.)



Kuva 2. ABB IRB 2600-20/1.65 -käsivarsirobotti, jonka työkalulaippaan on kiinnitetty GoCator-laserskanneri.

#### 4.2.4 Leikkaustyökalut

Leikkausrobotit automatisoivat materiaalin leikkaamisen työkappaleista niiden muotoilua ja jatkokäsittelyä varten. Koska leikkausten tarkkuus on tuotantoprosesseissa tärkeää, robottien korkea toistotarkkuus ja hallitut liikkeet tekevät

niistä ihanteellisia leikkaustehtäviin. Roboteilla voidaan leikata useilla erilaisilla työkaluilla, kuten vesi-, laser- ja plasmaleikkureilla sekä mekaanisilla leikkaustyökaluilla. (Robots Done Right, n.d.c.)

Leikkaustyökalujen kustannusten välillä on hyvin suuria eroja, joten leikkaustyövaiheen robotisoinnin järkevyyttä riippuu valitusta menetelmästä. Portaalirobotti soveltuu levyjen leikkaamiseen paremmin kuin käsivarsirobotti johtuen leikkaustyökalun asennon pysymisestä samana ja vain paikka levyllä vaihtuu. Käsivarsirobotin käyttäminen voi olla perusteltua suurempia ja monimutkaisempia kappaleita leikatessa. (Liuha ym., 2023, s. 222.)

#### 4.2.5 Työkalunvaihtajat

Robotin työkalulaipan ja työkalun väliin voidaan myös kiinnittää työkalunvaihtaja. Työkalunvaihtaja koostuu kahdesta osasta, jotka kiinnittyvät toisiinsa ja tämän avulla robotti voi vaihtaa oman työkalunsa kesken ohjelmakierron. Työkalunvaihtajan etuna on robotin tekemän prosessin tehokkuuden, joustavuuden ja nopeuden parantuminen. (Robots Done Right, n.d.d.)

## 5 ROBOTTIEN OHJELMOINTI

Robotin ohjelmointi tarkoittaa menetelmää, jossa robotille syötetään tarkkoja ohjeita automatisoitujen tehtävien suorittamiseksi. Ohjeet syötetään robotin ohjausjärjestelmään, joka ohjaa moottoreita tai toimilaitteita jokaisella akselilla. Ohjelma määrittää robotin toiminnan ja mahdollistaa sen suorittavan tiettyjä tehtäviä esimerkiksi valmistus-, käsittely-, logistiikka- tai pakkauslinjoilla. Robotin ohjelmoinnissa yleisimmin käytettyjä ohjelmointikieliä ovat C/C++, Python, Java ja C#. Lisäksi eri valmistajilla on omia ohjelmointikieliä, jotka on kehitetty heidän laitteilleen. Tämä tarkoittaa, että robotin ohjelmoija, joka osaa yhden valmistajan ohjelmointikielen, ei välttämättä pysty ohjelmoimaan toisen valmistajan robottia. (Automate-UK, n.d.)

Robotin ohjelmointi jakautuu kahteen pääluokkaan: online- ja offline-ohjelmointiin. Online-ohjelmoinnissa robotin käsivarsi siirretään haluttuihin asentoihin, jotka tallennetaan ja säilötään robotin järjestelmään. Offline-ohjelmointi puolestaan tarkoittaa ohjelman kirjoittamista erillisellä tietokoneella, jolla ohjataan robotin liikkeitä, minkä jälkeen ohjelma ladataan robottiin. (Automate-UK, n.d.)

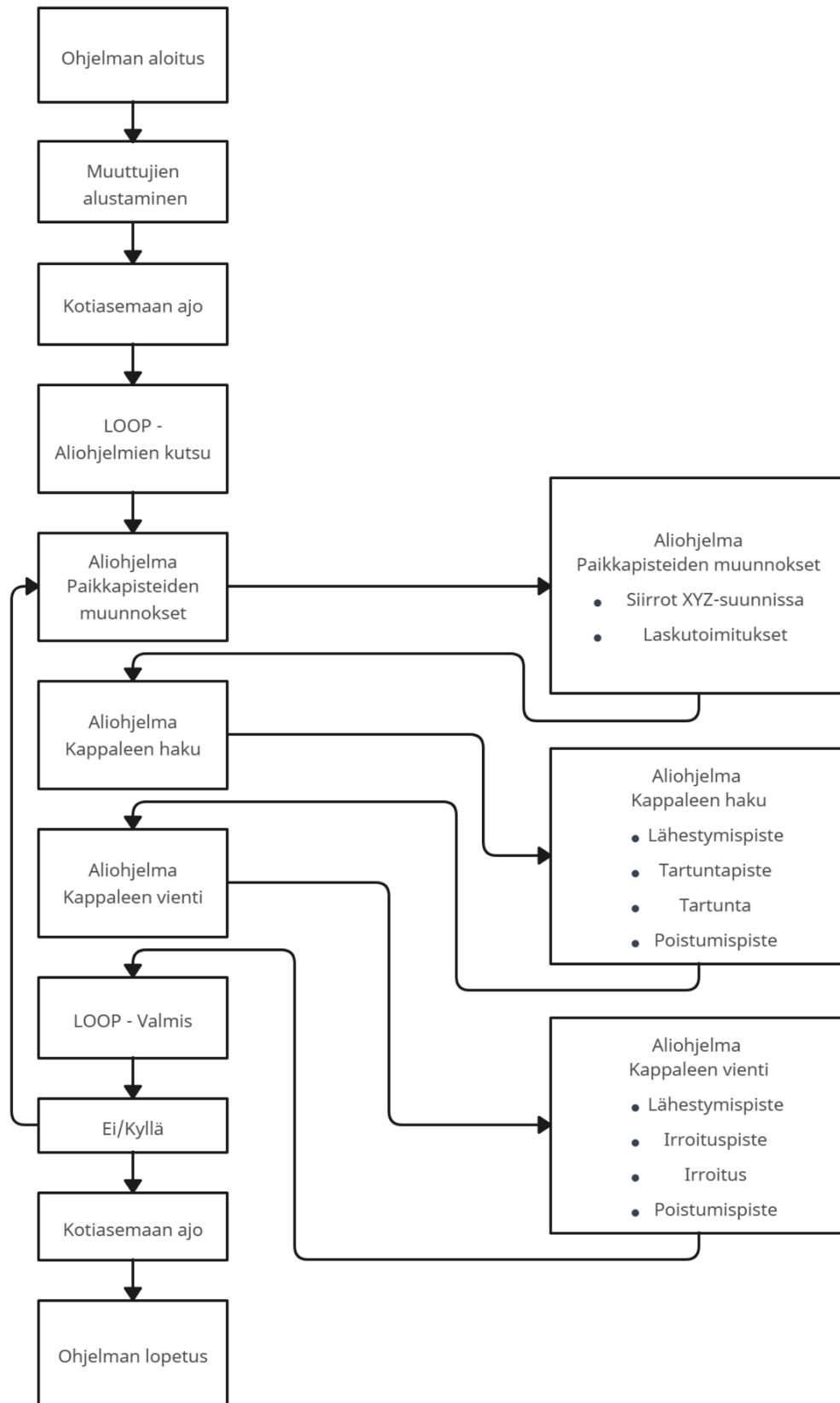
Perinteinen robottien opetusmenetelmä on ollut online-ohjelmoinnin keskeinen lähestymistapa ja ylivoimaisesti suosituin tapa ohjelmoida robotteja. Ohjelmoijat käyttävät opetuslaitetta (teach pendant), joka on ohjelmointiin tarkoitettu ohjauslaite robottien liikkeiden hallintaan. Opetuslaitteet ovat yleensä kädessä pidettäviä ja voivat olla joko langallisia tai langattomia. Robotti asetetaan opetus- tai oppimistilaan, ja laitteen näppäimillä tai komentopainikkeilla robottia ohjataan vaihe vaiheelta haluttuihin asentoihin ja liikeratoihin. Näin muodostetaan ohjelma. Ohjelmoinnissa robotin liikkeitä voidaan ohjata erilaisten koordinaattijärjestelmien avulla. (Automate-UK, n.d.)

Teollisuusrobottien online-ohjelmoinnin yksi haittapuoli on käyttökato. Ohjelmointi tapahtuu robotin itsensä avulla, mikä tarkoittaa, että robotti ei voi tuona aikana toimia tuottavasti. Offline-ohjelmointi poistaa tarpeen käyttää robotin

liikkeitä ohjelman luomiseen ja siirtää ohjelmoinnin virtuaaliseen ympäristöön. Ohjelmoija kirjoittaa koodin, mutta kaikki tapahtuu robotin virtuaalisessa kaksoiskappaleessa. Offline-ohjelmoinnin avulla ohjelmat ja liikeradat voidaan luoda suoraan CAD-malleista, jotka kuvaavat käsiteltäviä osia. Tämä menetelmä on erityisen hyödyllinen monimutkaisissa sovelluksissa, joissa manuaalinen ohjelmointi veisi paljon aikaa. Esimerkkejä tällaisista tilanteista ovat suurten, monimutkaisten osien käsittely tai tuotantoympäristöt, joissa on paljon erilaisia osatyyppejä, mutta pieniä sarjakokoja. Offline-ohjelmointi mahdollistaa tuotannon keskeytyksettömän jatkumisen, ja useimmissa tapauksissa ohjelmaan tarvitaan vain pieniä säätöjä, kun se ladataan robottiin. Tämä säästää merkittävästi aikaa erityisesti, kun siirrytään uusien osatyyppien tuotantoon. Offline-ohjelmointiin on olemassa monenlaisia lähestymistapoja yksinkertaisesta liikeradan luomisesta aina järjestelmän suunnitteluun, ohjelmointiin ja käyttöönottoon täysin virtuaalisessa ympäristössä. (Automate-UK, n.d.)

## 5.1 Robotin ohjelmasuunnittelu

Robotin ohjelmasuunnittelu kannattaa aloittaa prosessikuvauksen laatimisella riippumatta valitusta ohjelmointimenetelmästä (Kuva 3). Prosessikuvauksen avulla voidaan havaita vakavat ongelmakohdat ja helpottaa prosessin alkutilanteen hahmottamista. Selkeä prosessin havainnollistaminen edesauttaa tiimityöskentelyä ja eri tilanteiden esittelyä. Suunnitteilla olevasta prosessista tehdään nykyään usein myös simulaatiomalli halutulla robottityypillä. Näin voidaan varmistaa robotin ulottuman riittävyys haluttuun tehtävään, minkä jälkeen ohjelman tekeminen voidaan aloittaa. (Kolehmainen, 2023, s. 238–239.)



Kuva 3. Esimerkki ohjelman suunnittelun prosessikaaviosta.

Ohjelman laatiminen alkaa paikoituspisteiden, signaalien sekä muuttujien ja datan määrittämisellä ja nimeämisellä. Kaikki data dokumentoidaan selkeästi, mukaan lukien sen nimi, tarkoitus ja muut olennaiset tiedot. Samalla on tärkeää määrittää virheenkäsittelijä, eli mitä toimenpiteitä tehdään mahdollisten virhetilanteiden sattuessa. Määritetään myös tarvittavat taustaohjelmat ja laiteohjaukset sekä kommunikointirajapinnat. Varsinaisen ohjelmakoodin kirjoitus voi tapahtua joko alusta alkaen tai kierrättämällä ja mukauttamalla valmiita moduuleja. Ohjelman kulku tulee kommentoida selkeästi, jotta muutkin ymmärtävät sen kulun. Lopuksi ohjelman, oheislaitteiden ja niiden ohjauksien tekninen dokumentointi tehdään kattavasti, jotta kaikki tarvittava tieto on helposti saatavilla jatkokäyttöä ja ylläpitoa varten. (Kolehmainen, 2023, s. 239.)

## 5.2 Muuttujat ja ohjausrakenteet

Muuttujat (Kuva 4) ovat olennainen osa ohjelmaa, sillä niiden avulla robottiohjelma voidaan toteuttaa dynaamisesti. Jokaiselle muuttujalle annetaan nimi, joka toimii sen tunnisteena ja jonka avulla siihen voidaan viitata ohjelman eri kohdissa. Muuttujan arvo kuvaa muuttujan sen hetkistä sisältöä ohjelman suorituksen aikana, ja kuten nimi viittaa, arvo voi muuttua ohjelman kierron aikana. Muuttujien tyypit ja ominaisuudet vaihtelevat suuresti eri robottimerkkien mukaan, mutta yleisesti ne voidaan jakaa kolmeen pääryhmään: numeerista tietoa sisältävät muuttujat, tekstimuuttujat ja totuusarvomuttujat, jotka voivat olla joko tosia tai epätosia. Näiden muuttujatyyppeiden monipuolisuus mahdollistaa robottien toiminnan ja ohjelmien joustavan toteuttamisen. (Kolehmainen, 2023, s. 240.)

```
VAR num productCount := 0;  
VAR string connectionInfo := "Connection to ip 192.168.125.1 OK";  
VAR bool switchState := FALSE;  
VAR pos originXYZ := [0,0,0];
```

Kuva 4. Esimerkki muuttujamäärittelyistä.

Numeeriset muuttujat ovat yleisimmin käytettyjä tietotyyppisiä robottien ohjelmoinnissa ja ne mahdollistavat lukuarvojen tallentamisen ja käsittelyn. Toisin

kuin numeeriset muuttujat, merkkijonomuuttujat (String) voivat sisältää sekä tekstiä että numeroita. Merkkijonojen pituus voi olla joko kiinteä tai vaihteleva, mutta niiden rakenteena on aina peräkkäisten merkkien jono. Merkkijonot ovat keskeisiä erityisesti oheislaitteiden, kuten kameroiden, palvelimien ja työstökoneiden välisessä tiedonsiirrossa. Näiden laitteiden vastaanottama ja lähettämä tieto on usein merkkijonon muodossa. Koska merkkijono ei itsessään tunnista, ovatko sen sisältämät merkit numeroita vai muita merkkejä, sen sisältö on tarkistettava ohjelmallisesti. Tarkistukset voivat kohdistua esimerkiksi merkkijonon pituuteen tai siihen, ovatko siinä käytetyt merkit ohjelmallisesti sallittuja. Tällaiset tarkastukset ovat oleellisia virheiden ehkäisemiseksi ja tiedon oikeellisuuden varmistamiseksi ohjelmassa. (Kolehmainen, 2023, s. 241.)

Robotin ohjelmat harvoin etenevät alusta loppuun kirjoitetussa järjestyksessä, sillä robottien toiminta perustuu usein tilanteen arviointiin ja päätöksentekoon. Ohjelmissa robotti tarkistaa tiettyjen ehtojen täyttymisen ja suorittaa sen perusteella seuraavat komennot. Tällaisiin ehdollisiin suorituksiin käytetään ehtolauseita, joista yleisin on IF-lause. Robottiohjelmoinnissa IF-lauseita voidaan käyttää kahdella eri tyylillä: compact IF ja IF-ENDIF-tyyppinen ohjelmalohko. Compact IF on yksinkertaisempi ja soveltuu tilanteisiin, joissa tarvitaan lyhyitä ehtoja. Ohjelmalohkomalliset käskyt tarjoavat enemmän joustavuutta, sillä niihin voidaan lisätä myös ELSEIF- ja ELSE-lausekkeita. Näiden avulla ohjelmaan voidaan määritellä tarkemmin vaihtoehtoiset toimintamallit eri ehtojen täyttymiselle tai niiden puuttumiselle. Tämä mahdollistaa monimutkaisten toimintojen ja päätöksentekorakenteiden rakentamisen robottiohjelmiin. (Kolehmainen, 2023, s. 241.)

Robottiohjelma on usein suunniteltu kertaluonteiseksi suoritukseksi, mutta tämä suoritettava ohjelma voidaan toistaa haluttu määrä kertoja tai niin kauan kuin se on tarpeellista. Toistaminen voidaan toteuttaa ohjelman sisällä, jolloin tietyt osiot tai koko ohjelma suoritetaan useita kertoja (Kuva 5). Yleisimmät toistorakenteet ovat FOR - ENDFOR, WHILE - ENDWHILE ja LABEL - JUMP LABEL. FOR-lauseessa toistojen määrä määritellään etukäteen ja se soveltuu parhaiten tilanteisiin, joissa tiedetään tarkalleen, kuinka monta kertaa toistaminen halutaan suorittaa. Kuvan 5 esimerkissä kappaleen poimintaa toistetaan

FOR-lauseella halutun tuotemäärän verran. Käyttäjälle myös ilmoitetaan käsiohjaimeen poimitujen kappaleiden määrä. WHILE-lauseessa toistojen määrä puolestaan riippuu ehtojen täyttymisestä. Tämä rakenne toistaa toimintoja niin kauan, kunnes ehto on epätosi. Kuvan 5 esimerkissä WHILE-lauseessa käynnistyssignaalin saadessa arvon TRUE tarkistetaan, että tuotemäärä on isompi kuin 0. Tämän jälkeen kappaletta poimitaan, kunnes tuotemäärä on 0, minkä jälkeen signaali saa arvon FALSE ja ehtolause ei enää täyty. Jos robotti ei tue FOR- tai WHILE-lauseita, käyttävät ne usein LABEL / IF Ehto = Tosi -> JUMP LABEL -ratkaisua. Tässä lähestymistavassa ohjelma hyppää tiettyyn kohtaan ohjelman sisällä, ja toistaminen jatkuu niin kauan kuin ehto on täytetty. (Kolehmainen, 2023, s. 241–242.)

```

PROC ExampleFOR()

    VAR num productCount := 5;
    VAR num totalPickedProducts := 0;
    VAR string info := "The finished part amount is ";

    FOR i FROM 1 TO productCount DO
        PickPart;
        totalPickedProducts := totalPickedProducts + 1;
        TPWrite info + ValToStr(totalPickedProducts);
    ENDFOR

ENDPROC

PROC ExampleWHILE()

    VAR bool startSignal := TRUE;
    VAR num maxProductCount := 10;
    VAR num productCount := 0;

    productCount := maxProductCount;
    WHILE startSignal = TRUE DO
        IF productCount > 0 THEN
            PickPart;
            productCount := productCount - 1;
        ELSE
            startSignal := FALSE;
        ENDIF
    ENDWHILE

ENDPROC

```

Kuva 5. Esimerkki ehtorakenteiden käytöstä.

### 5.3 Robotin liiketyypit

Robottien liiketyypit ovat useimmilla robottimerkeillä samankaltaisia, vaikka poikkeuksia ja erikoisempia liikevaihtoehtoja on myös olemassa. Kolme yleistä liiketyppiä ovat:

1. Nivel-liike (joint): Nivelliikkeessä robotin työkalupiste siirretään haluttuun paikkaan ja orientaatioon samalla, kun robotin käsivarsi asetetaan tiettyyn nivelkonfiguraatioon. Nivelliikkeessä kaikki robotin akselit lähtevät liikkeelle ja pysähtyvät yhtä aikaa, mikä tekee liikkeestä yhtenäisen. Liikkeen laajuus voi vaihdella huomattavasti riippuen lähtö- ja kohdepisteiden sijainnista. Nivelliikettä käytetään usein tilanteissa, joissa robotin täytyy lähestyä toimilaitetta tai valmistautua tarkkuutta vaativaan tehtävään, kuten hitsaukseen, kokoonpanoon tai mittaukseen. Se mahdollistaa robotin joustavan liikkeen eri suuntiin, mutta liikerata ei ole tarkasti hallittu, mikä tekee siitä paremmin soveltuvan lähestymiseen kuin varsinaisiin tarkkuustöihin. (Kolehmainen, 2023, s. 234.) Tämä liiketyppi mahdollistaa robotin nivelten käyttävän suurinta mahdollista ohjelmitavaa nopeuttaan, jolloin robotin toimintaa voidaan optimoida ja sykli-aikoja lyhentää. Tämän liiketyypin ennalta-arvaamattomuuden vuoksi ohjelmoijan tulee kiinnittää erityistä huomiota liikkeen käyttöön. (Niall, n.d.)
2. Suora-liike (linear): Lineaarinen eli suoraliike tarkoittaa robotin työkalupisteen liikettä kahden pisteen välillä kulkevalla suoralla janalla. Tässä liikkeessä työkalupiste ja sen orientaatio pysyvät täsmällisesti määritetyllä liikeradalla, mikä edellyttää käsivarren akselien jatkuvaa ja tarkkaa säätelyä. Tämä tekee lineaarisesta liikkeestä vaativamman robotin ohjauksjärjestelmälle, sillä akselit työskentelevät yhdessä varmistaakseen, että työkalupiste pysyy määritetyllä reitillä. Jos työkalun orientaatio muuttuu liikeradan aikana, muutos tapahtuu yleensä tasaisesti liikkeen aikana. Tästä syystä lineaarista liikerataa ei suositella kovin pitkäksi yhdellä kertaa. Pidemmillä liikeradoilla on suositeltavaa lisätä väliin useampia paikoituspisteitä. Näin voidaan hallita käsivarren akseleiden

nivelkonfiguraatiot paremmin. (Kolehmainen, 2023, s. 235.) Tämä liiketyyppi on tarpeellinen monissa tilanteissa, kuten liikkumiseen ahtaissa tai rajoitetuissa tiloissa, joissa robotille ei voi sallia liikaa vapautta. Lineaarinen liike on myös välttämätön, kun siirretään toista pistettä tietyn matkan verran tai kuljetaan hitsausaamaa, jyrsäreitettä tai porausreikää pitkin. Lineaarisen liikkeen haittapuolena on sen hitaus verrattuna nivelliikkeeseen ja tämä johtuu siitä, että joidenkin nivelten täytyy liikkua enemmän kuin toisten, jotta roboti pysyy oikealla reitillään. Liike rajoittuu aina hitaimmin liikkuvan nivelen maksiminopeuteen, sillä jokaisella nivelmoottorilla on oma enimmäisnopeutensa. (Niall, n.d.)

3. Ympyränkaari- ja kaariliike (circular/spline): Circular eli ympyränkaari-liike muodostuu robotiikassa yleisesti kolmesta paikkatiedosta (pisteestä), jotka määrittävät kaaren (Kuva 6). Ohjelmoinnin kannalta eroja syntyy siitä, miten kaariliike toteutetaan. Käytännössä yhden kaaren ja sen liikeradan luomiseen tarvitaan yleensä 2–3 liikekäskeyä, mutta aina vähintään kaksi paikoituspistettä. (Kolehmainen, 2023, s. 235.)



Kuva 6. Esimerkki MoveC-käskyn käytöstä.

#### 5.4 Ohjelman rakenne ja suoritus

Pääohjelman rakenne tulisi suunnitella niin, että se on mahdollisimman selkeä ja helposti ymmärrettävä. Lukijan pitäisi pystyä päättelemään ohjelman perusteella robotin tehtävät ja pääasiallinen työkierto. Selkeyden saavuttamiseksi pääohjelmassa on hyvä pitää vain suoritusta kontrolloivia ehtolauseita ja niiden sisällä olevia aliohjelmakutsuja. Liikekäskyt tulisi aina sijoittaa ehtojen alaisiin aliohjelmiin. Esimerkiksi kotiasemaan ajon tai muiden liiketoimintojen ohjauksen tulisi tapahtua aliohjelmissä, jotka on liitetty pääohjelman ehtolauseisiin. Tämä tekee ohjelmasta helposti hallittavan ja mahdollistaa erillisten toimintojen tarkemman hallinnan ilman, että pääohjelma monimutkaistuu. (Kolehmainen, 2023, s. 242.)

Robotin ohjelmoinnissa toistuvat ja vakiintuneet toiminnot ohjelmoidaan usein erillisiksi aliohjelmiksi. Tällöin näitä toimintoja voidaan helposti kutsua myös uusissa ohjelmissä, mikä parantaa ohjelman ylläpidettävyyttä ja uudelleenkäytettävyyttä. Vaikka ohjelmat ovat samanarvoisia, aliohjelmaksi ohjelma muuttuu silloin, kun sitä kutsutaan toisen ohjelman yhteydessä. Pääohjelma on se ohjelma, josta robotin ohjelman suoritus alkaa, ja se käynnistää robotin toiminnot tyhjästä työmuistista. On tärkeää, että käyttäjä ja ohjelmoija ymmärtävät työmuistin roolin ja toiminnan. Työmuisti säilyttää tilan ja tiedot ohjelman suorituksen aikana, joten sen tila on tärkeää ottaa huomioon aina ohjelman käynnistyksessä ja alustaa hallitusti. Erityisesti automaattiajossa häiriönkorjaustilanteet voivat vaatia tarkempaa huomiota työmuistin tilan hallintaan, jotta virhetilanteet voidaan välttää ja ohjelman suoritus jatkuu oikein. Ohjelmien nimeämisessä on hyvä toimia yrityksen ohjelmointisääntöjen mukaan ja erottaa pääohjelma selkeästi muista aliohjelmista. (Kolehmainen, 2023, s. 242–243.)

Robottiohjaimen työmuistissa on aina vain yksi aktiivinen ohjelma, joka on valittu suoritettavaksi. Muut ohjelmat tallennetaan yleensä robottiohjaimen keskusyksikön kovalevylle tai muistikortille. Robottimerkistä riippuen ohjelman muutokset ja käskyt tallennetaan joko suoraan kovalevylle tai välimuistiin, josta ne siirretään myöhemmin kovalevylle. Robotin ohjelman suoritus tapahtuu perinteisesti rivi riviltä ylhäältä alaspäin, mikä tekee monen toiminnon

samanaikaisen suorittamisen vaikeaksi yhdessä ohjelmassa. Tämän vuoksi signaaliseuranta, logiikkatoiminnot ja kommunikointi oheislaitteiden kanssa on usein vaatinut ohjelmoitavan logiikan, eli PLC:n (Programmable Logic Controller) lisäämistä sovellukseen, jotta oheislaitteiden ohjaus voidaan toteuttaa samanaikaisesti robottiohjelman kanssa. Viime vuosina on yleistynyt robottien taustatehtävien ja moniajon (BackGround/MultiTask) käyttö. Näin roboteille voidaan ohjelmoida useita taustaohjelmia, jotka voivat olla käynnissä pääohjelman rinnalla tai jopa ilman pääohjelmaa. Tämä mahdollistaa moniajon ja erilaisten toimintojen samanaikaisen suorituksen. Robotin käsiohjaimen näytöllä näkyvä ohjelmaosuus on usein vain pieni osa koko ohjelmasta, mikä voi olla haaste aloitteleville ohjelmoijille, sillä heillä voi olla vaikeuksia ymmärtää ohjelman kokonaisuutta ja tehdä tarvittavia muutoksia sen toimintaan. (Kolehmainen, 2023, s. 243.)

## 5.5 RAPID

RAPID-ohjelmointikieltä käytetään ABB:n robottien ohjelmointiin. Se tukee tasoitettua ohjelmointikonseptia, jossa uusia rutiineja, tietokohteita ja tietotyyppejä voidaan ottaa käyttöön tietylle robottijärjestelmälle. RAPID-sovellusta kutsutaan tehtäväksi (Task) ja tehtävä koostuu joukosta moduuleja. Moduulin sisältöön lukeutuu joukko erilaisia tietoja ja rutiinimäärittelyjä. Moduulit voidaan jakaa kahteen luokkaan, jotka ovat tehtävämoduuli ja järjestelmämoduuli. Tehtävämoduulit ovat osa tehtävää tai sovellusta ja järjestelmämoduulit ovat osa robotin järjestelmää. Järjestelmämoduulit ladataan automaattisesti järjestelmän käynnistyessä ja ne on suunnattu ennalta määrättyjen järjestelmäkohtaisten tietokohteiden määrittämiseen. (ABB, n.d.)

RAPID-ohjelmointikielessä on kolme erilaista rutiinityyppiä, jotka ovat funktio, proseduri ja TRAP-rutiini. Funktio palauttaa aina tietyn tyyppisen arvon ja niitä käytetään osana lausekkeita. Proseduurit eivät palauta arvoa ja ne toimivat aliohjelmiana, joka suorittaa tietyn tehtävän. TRAP-rutiinit mahdollistavat ohjelman keskeytyksiin reagoinnin ja ne suoritetaan automaattisesti, kun rutiiniin liitetty keskeytys tapahtuu. (ABB, n.d.)

RAPID-kielessä on neljä tietokohdetyyppiä (Kuva 7), jotka ovat vakiot, muuttajat, persistentit muuttajat ja parametrit:

- Vakiot ovat tietokohteita, joiden arvo on vakio eikä muutu ohjelman suorituksen aikana. Kuvassa 7 on määritetty komentonumero, jonka arvon ei haluta muuttuvan.
- Muuttajat ovat tietokohteita, joiden arvoa voidaan muuttaa ohjelman aikana ja niiden arvo menetetään sekä alustetaan uudelleen jokaisella uudella ohjelmakierrolla. Kuvassa 7 on määritetty bool-tyyppinen signaalimuuttuja, jonka arvo vaihtelee ohjelmakierron aikana esimerkiksi anturin tilan mukaan.
- Persistentit muuttajat säilyttävät arvonsa ohjelmakiertojen välillä ja esimerkiksi silloin kun robotti sammutetaan ja käynnistetään uudelleen. Kuvassa 7 on määritetty hitsaustyökalun tiedot persistentiksi muuttujaksi, jotta työkalun arvo säilyy.
- Parametrit ovat tietokohteita, jotka välitetään muille aliohjelmille.

```

RECORD productData ! Tietue
  string Name;
  num Height;
  num Length;
  num Width;
ENDRECORD

CONST num commandNumber := 10; ! Vakio
PERS tooldata WeldingTool := [TRUE, [[0,0,0],[1,0,0,0]],[0,[0,0,0],[1,0,0,0],0,0,0]]; ! Persistentti
VAR bool signalState := FALSE; ! Muuttuja

VAR num speedList{4} := [100, 200, 300, 400]; ! Taulukko

```

Kuva 7. Esimerkki RAPID-kielen tietokohdetyypeistä.

Tietokohteet voidaan myös järjestää tietueiksi (RECORD), jotka sisältävät useita tietotyyppiä tai ne voidaan dimensioida taulukoihin ja matriiseihin. RAPID-ohjelmointikieli myös tukee askelittaista ja taaksepäin suoritettavaa ohjelmakoodia. Taaksepäin suoritettavuus on erityisen hyödyllinen ohjelman kehittämisen aikana, esimerkiksi virheidenkorjauksessa, testauksessa ja säätämisessä. (ABB, n.d.)

## 6 KONEPALVELUROBOTIN OHJELMAN TOIMINTA

Tässä opinnäytetyössä suunnitellun konepalvelurobotin prototyypin tarkoituksena on palvella erilaisia CNC-koneita mahdollisimman itsenäisesti ja mahdollistaa sen operaattorille helppo ja sujuva käyttökokemus. Prototyyppi on tehty helposti siirrettäväksi ja siinä on oma varastoratkaisu, jonka täyttämisen ja tyhjentämisen hoitaa operaattori, mutta robotti pystyy käsittelemään varastoa itsenäisesti ohjelmallisten käskyjen ja rajoitteiden mukaisesti. Robottina käytetään GoFa-yhteistyörobottia, koska liikuteltavan prototyypin turvallistaminen yhteistyörobotilla on helpompaa verrattuna perinteisiin teollisuusrobotteihin ja se mahdollistaa operaattorille kätevän tavan opettaa paikkapisteitä. Työkaluksi valittiin OnRobotin valmistama sähköinen tarttuja sen ohjelmoitavuuden ja erilaisiin tilanteisiin sopivuuden vuoksi. Tarttujaan on lisäksi kiinnitetty erillinen lisäosa varastoon liittyvien liikkeiden suorittamiseen. Suurin osa robotin ohjelmoinnista toteutettiin offline-ohjelmointina käyttäen ABB:n RobotStudio-ohjelmistoa, jonka avulla voitiin myös hyödyntää prototyypin digitaalista kaksosta. RobotStudion simuloitussa ympäristössä robotin ja ohjelmakoodin toimintaa pystyttiin kokeilemaan kauan ennen kuin varsinainen prototyyppi oli saatu kasattua, mikä nopeutti kehitystyön ohjelmointiosuutta huomattavasti ja mahdollisti robotin toimintaperiaatteiden tarkastelun.

Robotin ohjelman oikeanlaisen toiminnan kannalta operaattorin syöttämät ja opettamat tiedot ovat tärkeässä osassa. Kun operaattori on tuonut robotin paikoilleen halutun työstökoneen eteen, hänen tehtävänsä on laittaa työstettävät kappaleaihiot varastoon ja opettaa robotille paikoituspisteitä työstökoneesta ja varastossa olevista kappalejigeistä hyödyntäen GoFa-robotin Lead-Through-ominaisuutta ja erillistä opetustyökalua. Käsiohjaimessa olevan käyttöliittymän kautta operaattori syöttää ajettavien aihoiden tiedot ja määrittää ohjelman suoritusjärjestyksen sekä säätää muita tarpeellisia asetuksia. Kun operaattori on syöttänyt robotille kaiken tarpeellisen datan, hän voi käynnistää ohjelman käyttöliittymän kautta, jolloin robotti alkaa suorittaa konepalvelua itsenäisesti. Robotin työkierto koostuu muutamasta työvaiheesta. Ensin robotti avaa varaston, ottaa sieltä työstettävän aihion, laittaa sen työstökoneeseen, sulkee

koneen ovet ja käynnistää työstökoneen. Kun työstöohjelma on valmis, robotti avaa ovet, ottaa valmiin tuotteen pois ja laittaa sen takaisin varastoon.

Ohjelmoinnin selkeyttämiseksi on prototyypin ohjelman pääasialliset toiminnot jaoteltu useampaan eri moduuliin, joihin on rakennettu tarvittavat aliohjelmat ja niiden tarvitsemat datatyypit. Tässä työssä tehdyllä selkeällä ohjelmajalla helpotetaan tulevaa ohjelmakehitystä eri ohjelmoijien välillä, aliohjelmien toimintojen testaamista ja versionhallintaa. Ohessa käydään läpi lyhyesti eri moduulien toiminnot ja sisältö:

- Varastodatan hallintamoduuli: Tästä moduulista löytyy varaston toimintaan ja ylläpitoon liittyvät tietotyypit ja funktiot. Monet tiedot, kuten tuotetiedot, on määritetty RECORD-tietueella, jonka avulla useita eri muuttujia saadaan ryhmitettyä yhteen loogiseen kokonaisuuteen. RECORD-tietueiden tiedot tallennetaan listoihin, jotka on määritetty moduulin sisäiseksi, jolloin niitä ei pysty muokkaamaan muualla ohjelmassa ilman, että käyttää moduulissa olevia funktioita datan hallintaan. Näin tehtynä varastodatan hallinta pysyy selkeänä ja vähentää virheiden mahdollisuutta. Listan kokoa muuttamalla voidaan muokata järjestelmään tallennettavien tietojen, kuten tuotteiden tai kappalejigien maksimimäärää.
- Päämoduuli: Päämoduulista löytyy ylemmän tason alirutiinit, joiden kautta kutsutaan alemman tason alirutiineja. Nämä ylemmän tason rutiinit muodostavat ohjelmalle järjestyksen, jonka mukaan robotti suorittaa konepalvelua. Lisäksi itse pääohjelma eli main-proseduuri, jossa ohjelma suoritetaan, löytyy helposti tämän moduulin lopusta.
- Datamoduuli: Datamoduuli sisältää erilaisia globaaleja tietotyyppejä, kuten paikoituspisteitä ja työkaludataa. Tämän moduulin sisältö on erikseen haluttu määrittää globaaliksi, jotta sitä pystytään käyttämään kaikissa muissakin moduuleissa helposti. Lähes kaikki tämän moduulin muuttujat on määritetty persistenteiksi, koska niiden arvojen säilyminen on ohjelman toiminnan kannalta erittäin tärkeää.

- Käyttöliittymämoduuli: Kaikkien käsiohjaimessa näkyvien käyttöliittymän toimintojen proseduurit löytyvät tästä moduulista. Kun pääohjelma kutsuu päävalikko-proseduuria, pääsee operaattori liikkumaan eri valikoiden välillä, syöttämään dataa järjestelmään ja muuttamaan ohjelman asetuksia.
- Varastonhallintamoduuli: Tämä moduuli sisältää robotin varaston fyysiseen hallintaan liittyvät funktiot. Funktioissa käsitellään käytettävän varaston perusteella paikoituspisteiden muunnokset ja näihin pisteisiin liittyvät robotin liikeradat.
- Matemaattisten työkalujen moduuli: Jotta robotti pystyy suorittamaan konepalveluun liittyviä tehtäviä riittävällä tarkkuudella, on ohjelmaan rakennettu useita erilaisia matemaattisia työkaluja, jotka on sijoitettu tähän moduuliin. Näillä funktioilla pystytään laskemaan tarkkoja koordinaatistoja opetetuista pisteistä ja laskemaan työstettävien kappaleaihioiden poimintapisteiden paikkamuunnoksia robotin koordinaatistossa.
- Työstettävien kappaleiden hallintamoduuli: Robotin tarvittavat liikkeet ja niiden muunnokset kappaleiden poimintaan, työstökoneeseen asettamiseen ja sieltä pois ottamiseen kuuluvat tähän moduuliin. Liikkeet perustuvat aihoiden annettuihin mittoihin ja erilaisiin muotoihin sekä niiden sijainteihin varastossa tai työstökoneen kiinnittimien pisteisiin. Lisäksi aliohjelmat työstökoneiden ovien manuaaliseen avaamiseen sijaitsevat tässä moduulissa.
- Pisteiden opetusmoduuli: Työstökoneen ja varaston opettamiseen liittyvät proseduurit sijaitsevat tässä moduulissa. Eri paikoituspistekokonaisuuksien opettamiseen liittyvät proseduurit on ohjelmoitu näyttämään operaattorille käsiohjaimen kautta kuvalliset ja kirjalliset ohjeet, mistä ja miten tarvittavat pisteet opetetaan.
- Käyttöliittymän apumoduuli: Käyttöliittymän ohjelmakoodin pitämiseksi selkeänä ja helposti luettavana on erilaisten varastodatojen

muokkaamiseen liittyvät proseduurit sijoitettu tänne. Kun näitä prosedureja kutsutaan käyttöliittymän puolelta, kysyy ohjelma operaattorilta tarvittavat tiedot ja tallentaa ne muuttujiin. Tarpeellisten ja hyväksyttävien tietojen syöttämisen jälkeen ohjelma siirtää nämä sopivaan varastodatan funktioon, ja jos funktion suoritus onnistuu, siirtyvät tiedot varastolistoihin.

## 6.1 Pääohjelma

Konepalvelurobotin prototyypin päärutiini suorittaa kaikki tarvittavat aliohjelmakutsut tietyssä ja hallitussa järjestyksessä, jotta robotin suunniteltu konepalveluprosessi onnistuu. Päärutiinissa sijaitsee kaikki ylemmän tason aliohjelmat, joiden kautta kutsutaan niiden ehtolauseiden toteuduttua muita toiminnan kannalta tärkeitä alemman tason aliohjelmia. Tämän jaottelun ansiosta päärutiini saadaan koodimäärältään pidettyä mahdollisimman lyhyenä ja ohjelman suoritusjärjestys pysyy helposti tulkittavana.

Tässä opinnäytetyössä tehdyn robotin ohjelman päärutiinin kierto menee kuvassa 8 näkyvällä tavalla. Esimerkin yksinkertaistamiseksi siinä ei käsitellä erilaisia virhetilanteita.

```

PROC main()
  VAR bool dummy;
  VAR num nextShelfIndex;
  VAR num nextSlotIndex;
  VAR num nextProductNameIndex;
  VAR ProductNameData nextProductNameData;

  IF NOT InitializeProgram() EXIT;
  ! Force open the user interface after initialization
  HandleUserInterface \ForceShow;

  WHILE TRUE DO
    IF SystemState = RUN_PRODUCTION_STATE THEN
      shelfIndex := shelfOrder{startingShelfIndex};
      ! If there is a part in the work machine, machining program is ready, and robot is not holding part,
      ! take part from work machine and put it to storage
      IF PartInWorkMachine = TRUE AND MachiningProgramFinished() = TRUE AND HoldingPart = FALSE THEN
        dummy := PickFromWorkMachine(); ! -> DoorsOpen = TRUE + HoldingPart = TRUE + PartInWorkMachine = FALSE
        dummy := PlaceToStorage(CurrentPartShelfIndex, CurrentPartSlotIndex \NewState:=PS_FINISHED); ! -> OpenedShelfIndex = X + HoldingPart = FALSE
      ENDIF
      GetShelf shelfIndex;
      ! If the work machine is empty and robot is not holding part,
      ! try to find something from storage, place it to work machine, and start machining
      IF PartInWorkMachine = FALSE AND HoldingPart = FALSE THEN
        IF FindNextPartFromStorage(nextShelfIndex, nextSlotIndex, nextProductNameIndex) AND
           GetProductNameDataByIndex(nextProductNameIndex, nextProductNameData) THEN
          IF nextProductNameData.NcProgName < "" THEN
            dummy := PickFromStorage(nextShelfIndex, nextSlotIndex); ! -> OpenedShelfIndex = X + HoldingPart = TRUE
            dummy := PlaceToWorkMachine(); ! -> DoorsOpen = TRUE + PartInWorkMachine = TRUE + HoldingPart = FALSE
            dummy := StartMachiningProgram(nextProductNameData.NcProgName); ! -> DoorsOpen = FALSE + machining program running
          ELSE
            dummy := SetShelfProductState(nextShelfIndex, nextSlotIndex, PS_FINISHED);
          ENDIF
        ENDIF
      ENDIF
    ENDIF
    WaitTime 0.1;
  ENDWHILE
ENDPROC

```

Kuva 8. Yksinkertaistettu pääohjelmakoodi.

Kuvan 8 mukaisessa pääohjelmakoodissa edetään seuraavasti. Kun operaattori käynnistää robotin ohjelman, siirtyy ohjelmaosoitin main-proseduuriin ja ohjelma suorittaa tarvittavat muuttujien alustukset. Tämän jälkeen kutsutaan aliohjelmaa, joka pakottaa käyttöliittymän päävalikon näkyviin käsiohjaimeen. Kun operaattori on syöttänyt haluamansa tiedot ja määrittänyt ohjelman varastojen suoritusjärjestyksen, tallennetaan nämä tiedot erilliseen järjestyslistaan ja ohjelma siirtyy ikuiseen pääsilmutkaan, joka on tehty WHILE-toistolauseella. Ikuinen silmukka on ohjelmointirakenne, jossa toistorakennetta suoritetaan loputtomasti, ellei sitä erikseen katkaista. Kun käyttäjän määrittämä työkierto on saatu suoritettua, jää silmukka yhä päälle odottamaan uusia tietoja operaattorilta ilman, että robotin ohjelma täytyisi käynnistää uudelleen.

Pääsilmutkan ensimmäisessä IF-ehtolauseessa tarkastetaan, että järjestelmän tila on asetettu tuotantotilaan, minkä jälkeen shelfIndex-muuttujaan määritetään arvo järjestyslistasta. Seuraavaksi tarkastetaan, onko työstökoneessa kappaleaiho, onko työstöohjelma suoritettu, ja että robotilla ei ole kappaletta työkalussaan. Jos nämä ehdot täyttyvät, suoritetaan aliohjelmat, joissa kappale haetaan työstökoneesta ja palautetaan varastoon oikealle paikalleen.

Kun edellä mainitut ehdot eivät täyty, suoritetaan aliohjelma, joka hakee shelfIndex-arvon mukaisen varaston tiedot, jotka sisältävät esimerkiksi työstettävän kappaleen ja sen kappalejigin paikoitustiedot. Tämän jälkeen tarkastetaan, että työstökone on tyhjä, ja että robotin tarttujassa ei ole kappaletta. Jos nämä ehdot täyttyvät, suorittaa ohjelma funktiot, joissa haetaan seuraavan työstettävän aihion tiedot varastosta. Kun nämä funktiot palauttavat TRUE-arvon, tarkastetaan vielä, että työstettävän aihion työstöohjelman nimiarvo on erisuuri kuin tyhjä.

Edellisten ehtojen täytyttyä suoritetaan aliohjelmat, joissa kappale poimitaan varastosta, laitetaan työstökoneeseen ja käynnistetään työstöohjelma. Silmukan loppuun on lisätty WaitTime-käsky, joka odottaa 0,1 sekuntia. Tämä aiheuttaa viiveen pääsilman kiertoon, mikä vähentää prosessorin kuormitusta. Ylimääräinen viiveaika on kuitenkin niin lyhyt, ettei sillä ole mitään vaikutusta robotin konepalvelun sykliakoihin.

## 6.2 Aliohjelmat

Tässä opinnäytetyössä tehdyssä robotin ohjelmassa kaikki erilliset toiminnot on sijoitettu omiin aliohjelmiinsa ja riippuen aliohjelman halutusta lopputuloksesta, on niiden tekemisessä käytetty sekä funktioita että proseduureja. Tämän ansiosta työkiertoon ja käyttöliittymään liittyviä toimintoja on helppo kutsua missä vain ohjelmassa. Se myös helpottaa ohjelmakoodin kirjoittamista, hallintaa ja lukemista. Erillisille aliohjelmille on annettu kuvaavat nimet, jotta niistä näkee nopealla silmäyksellä, mitä ne tekevät, ja ne on jaoteltu omiin moduuleihinsa. Tämän opinnäytetyön kirjoitushetkellä erilaisia aliohjelmia on yli sata kappaletta yhdeksässä eri moduulissa ja niiden pituus vaihtelee muutamasta koodirivistä lähemmäs kahteensataan.

Aliohjelmat voidaan jakaa karkeasti viiteen eri pääkategoriaan:

- Robotin liikkeiden hallintaan liittyvät aliohjelmat: Näillä aliohjelmilla muunnetaan ja ohjataan robotin liikkeitä fyysisen konepalveluprosessin

suorittamiseksi. Kaikki robotin liikkeet ovat tämän opinnäytetyön kirjoitushetkellä MoveJ- tai MoveL-liikkeitä. Suurin osa yleisistä liikkeistä toteutetaan nopeina nivelliikkeinä, mutta kappaleiden poimintaan ja laskemiseen liittyvät liikkeet toteutetaan lineaariliikkeinä (Kuva 9). Kaikki työstettävien aihoiden liikuttelu varaston ja työstökoneen välillä sekä muut fyysiset toiminnot sijaitsevat näissä aliohjelmassa. Prosessiin vaadittavien paikkapisteen muunnoksien laskutoimitukset tapahtuvat myös yleensä näissä ohjelmissa. Paikkapistemuunnoksissa hyödynnetään erilaisia parametreja, jotka liittyvät esimerkiksi poimittavan tuotteen mittoihin tai varastossa olevan jigin tietoihin.

```

IF partShape = PART_SHAPE_RECT THEN ! Rectangular

! Shift picking point
pickShifted := pick;
dummy := slotIndexToPose(slotIndex, usedjig, slotPose); ! Calculate next slot origin point
pickShifted.trans := slotPose.trans;
pickShifted.rot := [0,0,1,0];
pickShifted.trans.x := pickShifted.trans.x + foundProduct.Width / 2;
pickShifted.trans.y := pickShifted.trans.y + foundProduct.Length / 2;
pickShifted.trans.z := pickShifted.trans.z + foundProduct.Height - (foundProduct.GraspCoordIn + 16);

! Shift above picking point
aboveShifted := pickShifted;
aboveShifted.trans.z := pickShifted.trans.z + shiftHighZ;

! Shift approach point
approachShifted := pickShifted;
approachShifted.trans.z := pickShifted.trans.z + shiftLowZ;

! Shift return point
returnShifted := pickShifted;
returnShifted.trans.z := pickShifted.trans.z + 5;

! Move commands
MoveJ aboveShifted,v50,z50,tCurrTool\WObj:=wTempWobj;
MoveL approachShifted,v50,fine,tCurrTool\WObj:=wTempWobj;

MoveL pickShifted,v50,fine,tCurrTool\WObj:=wTempWobj;
stop;
WaitTime 2;

! Activate gripper
!OR_2FG_GRIP 1,20,60,10,1;

WaitTime 1;
MoveL approachShifted,v50,fine,tCurrTool\WObj:=wTempWobj;
MoveL aboveShifted,v50,z50,tCurrTool\WObj:=wTempWobj;

```

Kuva 9. Osa aihion poiminnan aliohjelmasta.

- Varastohallintaan liittyvät aliohjelmat: Varastohallintaan liittyvät tietojen syöttö-, poisto- ja hakutoiminnot sijaitsevat näissä aliohjelmassa. Kaikki varastodatan hallinnat on toteutettu funktioiden avulla. Apuohjelmia käyttämällä operaattorin syötteet saadaan annettua suoraan parametreiksi näihin funktioihin, joissa tietojen oikeellisuutta vertaillaan ohjausrakenteiden avulla. Lähtökohtaisesti, jos operaattorin syötteistä tulleet parametrit ovat oikein, nämä funktiot lisäävät, poistavat tai hakevat

tietoa järjestelmän varastolistoista. Kuvassa 10 on esimerkki, jossa varaston kappalejigilistasta poistetaan halutun jigien tiedot. Funktiossa tarkistetaan, että parametrina saatu jigi-indeksi on sallittujen arvojen sisällä. Tämän jälkeen FOR-lauseella tarkistetaan, käytetäänkö poistettavaa jigia jossain hyllyssä. Jos näin on, hyllytiedoista poistetaan tuotetiedot. Tämän jälkeen järjestelmässä olevan jigilistan jigi-indeksin mukaisen alkion tiedot määritetään tyhjiksi ja järjestelmässä olevaa jigien kokonaismäärää vähennetään yhdellä. Lopuksi funktio palauttaa arvon TRUE.

```
! Delete a jig from the system based on index
FUNC bool DeleteJig(num jigIndex)
  VAR bool dummy;
  IF jigIndex < 1 OR jigIndex > MAX_JIG_COUNT RETURN FALSE;

  FOR nI FROM 1 TO SHELF_COUNT DO
    IF Shelves{nI}.JigIndex = jigIndex THEN
      dummy := DeleteShelfProducts(nI);
      Shelves{nI}.JigIndex := 0;
    ENDIF
  ENDFOR

  Jigs{jigIndex} := [ 0, 0, [[0, 0, 0],[0,0,0,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9]], 0, 0, [0,0,0], [0,0,0] ];
  Decr JigCount;
  RETURN TRUE;
ENDFUNC
```

Kuva 10. Funktio, jolla poistetaan kappalejigi robotin järjestelmästä.

- Apuohjelmat: Näissä ohjelmissa annettujen parametrien perusteella lasketaan erilaisia geometrisia laskutoimituksia, joiden avulla pystytään määrittämään koordinaatistoja haluttuihin asentoihin tai laskemaan kappaleiden paikoituksia esimerkiksi hyödyntäen ristituloa ja Hamiltonin kvaternioiden tuloa. Lisäksi apuohjelmina toimii sellaiset proseduurit, jotka kysyvät operaattorilta tarvittavia tietoja ja syöttävät annetut tiedot varastonhallintaan tai käyttöliittymään liittyvien funktioiden parametreihin.

Kuvan 11 esimerkissä lasketaan sylinterin poimintakorkeus, kun tiedetään sylinterin halkaisija ja kappalejigin kulma. Funktion parametreiksi annetaan numeroarvoina halkaisija ja kulma. Puolitetun kulma-arvon siniarvo kerrotaan kahdella, jonka jälkeen kappaleen poimintakorkeus

saadaan jakamalla halkaisija lasketulla arvolla. Funktio palauttaa lasketun poimintakorkeuden numeerisena arvona, jonka jälkeen pistettä voidaan siirtää muualla ohjelmassa halutulla tavalla.

```
! Calculate a cylinders pick height based on the diameter of the cylinder
! and the angle of the part jig
FUNC num CalculateCylinderPickHeight(num diameter, num angle)
  VAR num height;
  VAR num theeta;

  theeta := angle / 2;
  theeta := 2 * Sin(theeta);
  height := foundProduct.Diameter / theeta;

  RETURN height;

ENDFUNC
```

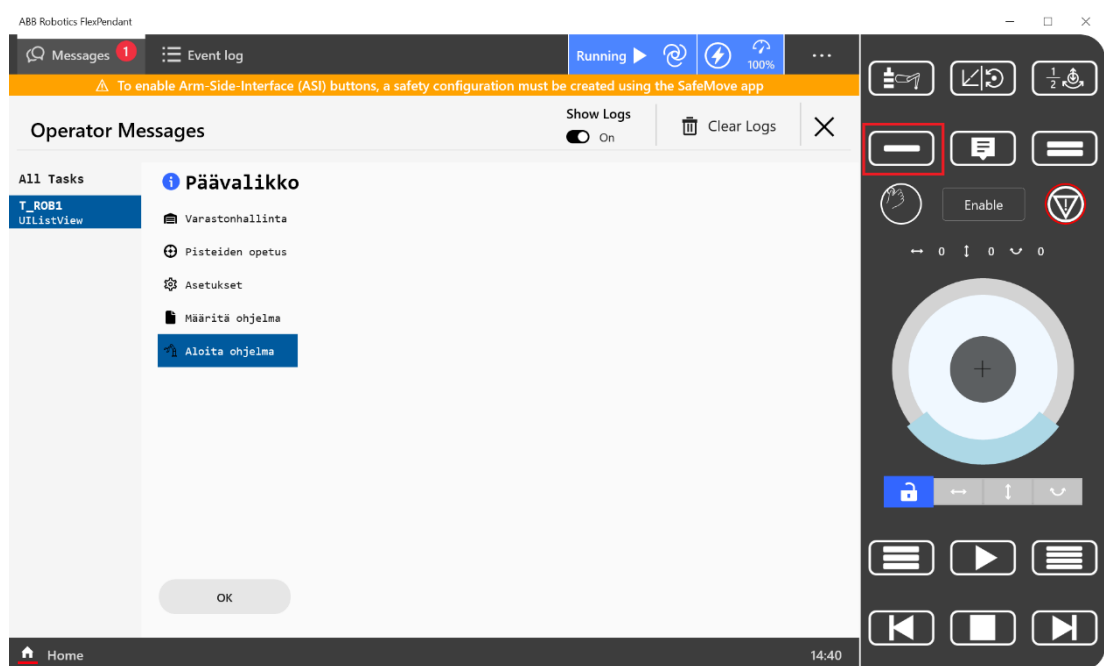
Kuva 11. Funktio sylinterin poimintakorkeuden laskemiseen.

- Paikkapisteiden opetusaliohjelmat: Opetusaliohjelmilla ohjeistetaan robotin operaattoria opettamaan konepalvelurobotin toiminnan kannalta oleelliset paikkapisteet hyödyntäen RAPID-kielestä löytyvää CRobT-funktiota. Näillä aliohjelmilla näytetään robotille, missä työstökoneen, kappalekiinnittimien ja varaston paikkapisteet ovat. Näistä opetetuista pisteistä lasketaan apualiohjelmia käyttäen tarkkoja koordinaatistoja, joita robotti pystyy käyttämään ja sitä kautta saavuttamaan riittävän tarkkuuden konepalvelun toimintoihin. Esimerkiksi työstökoneesta löytyvien kappalekiinnittimien paikoittaminen tarkasti, jotta työstettävä aihio viedään aina haluttuun paikkaan halutulla tavalla, on tärkeää työohjelman onnistumisen kannalta.
- Käyttöliittymän aliohjelmat: Näillä aliohjelmilla ohjataan käyttöliittymän eri toimintoja hyödyntäen apualiohjelmia. Näistä puhutaan tarkemmin seuraavassa kappaleessa.

### 6.3 Käyttöliittymä

Konepalvelurobotin käyttäjäkokemuksessa helppokäyttöisellä ja selkeällä käyttöliittymällä on suuri merkitys. RAPID-ohjelmointikieli tarjoaa muutamia erilaisia funktioita käyttöliittymän ohjelmointiin. Tässä opinnäytetyössä hyödynnettiin käyttöliittymän tekemiseen pitkälti näitä funktioita, kuten UICollection, UITextField ja UITextFieldDelegate. Käyttöliittymän päävalikon aliohjelmaa kutsutaan aina robotin pääruutiinin ensimmäisellä kierrolla, jolloin se aukeaa käsiohjaimessa. Operaattori saa päävalikon uudelleen näkyviin kesken ohjelmakierronkin painaessaan käsiohjaimen ensimmäistä ohjelmoitavaa painiketta. Tämä painike on ohjelmoitu asettamaan digitaalisen lähtötiedon UserMenuOn arvoksi 1 eli TRUE. Tämän lähtötiedon arvoa tarkastellaan muissa työkierron aliohjelmissa, ja jos sen arvo on 1, siirretään ohjelmaosoitin takaisin päävalikon aliohjelmaan, jolloin se taas aukeaa operaattorin nähtäväksi ja muut toiminnot pysähtyvät siksi aikaa, kun valikossa ollaan.

Päävalikon kautta operaattori pääsee hallitsemaan useita robotin järjestelmän tietoja ja suurin osa valikkorakenteista näkyy kuvan 12 mukaisina listanäkyminä.



Kuva 12. Päävalikon käsiohjainnäky.

Kuvan 12 mukaiset valikkonäkymät on toteutettu hyödyntäen UICollection-funktiota. Tämä funktio vaatii toimiakseen parametriksi jonkin listitem-nimisen tietueen, jonka tietotyyppeinä on kaksi string-tyyppistä muuttujaa (Kuva 13). Toiseen näistä voidaan määrittää sellaisen kuvatiedoston nimi, joka on sijoitettu robotin järjestelmään ja toiseen määritetään teksti, jonka haluaa nähdä valikossa. Funktioon voidaan kuitenkin lisätä monia muita valinnaisia parametreja.

```
CONST listitem mainList{5} := [ ["warehouse-solid.png", "Varastonhallinta"], ["target.png", "Pisteiden opetus"], ["settings.png", "Asetukset"], ["file-solid.png", "Määritä ohjelma"], ["robotic-arm.png", "Aloita ohjelma"] ];
```

### Kuva 13. Listamäärittely.

Kaikki käyttöliittymän aliohjelmat, joissa käytetään UICollection-funktiota, sisältävät tämän funktiokutsun WHILE-silmukan sisällä. Tämä mahdollistaa valikkonäkymien jatkuvan päivittymisen ja saumattoman liikkumisen eri valikoiden välillä. Kun operaattori tekee valinnan jossain valikossa, siirrytään seuraavan aliohjelman toimintoihin, ja operaattorin tehtyä haluamansa toiminnot, palataan takaisin siihen silmukkaan, mistä aloitettiin. Näistä silmukoista voidaan poistua käyttäjän tekemillä valinnoilla, jolloin loppujen lopuksi palataan takaisin päävalikon silmukkaan. Päävalikossa oleva silmukka lopetetaan valitsemalla ohjelman aloittaminen.

Kuvassa 14 näkyy robotin työkaluvalinnan valikkoaliohjelma. Aliohjelmaan tullessa alustetaan muuttujat ja tyhjät listitem-arvot. Erillinen toolListNames-lista on määritetty datamoduulissa globaalisti käytettäväksi, mikä sisältää erilaisten työkalujen nimet. Valikossa näytettävän listan ensimmäiseen alkioon määritetään globaalista listasta teksti ”Käytössä oleva työkalu” ja siihen lisätään käytettävän työkalun numeron perusteella työkalun nimi. FOR-silmukan avulla täytetään kaikki näytettävän listan alkiot ja sen jälkeen suoritetaan UICollection-funktio. Täältä operaattori valitsee haluamansa työkalun, jonka arvo määritetään nykyisen työkalun persistentille muuttujalle, jota käytetään robotin liikekäskyissä. Ohjelmassa mahdollistetaan käytettävien työkalujen tietojen vaihtaminen, koska konepalvelurobotti käsittelee useita erikokoisia ja -muotoisia kappaleita, joiden poimintaan tarvitaan erilaisia tarttujan sormia. Jos

käyttäjä painaa peruuttamisnäppäintä, toteutuu ehtolause ja proseduurista poistutaan takaisin asetukset-valikkoon.

```

PROC ToolSelectionInterface()

  VAR num item;
  VAR listitem toolList{7} := [ ["", ""], ["", ""], ["", ""], ["", ""], ["", ""], ["", ""], ["", ""] ];
  VAR btnres choice;

  WHILE TRUE DO
    mainButtons{1} := "OK";
    mainButtons{2} := "Peruuta";
    toolList{1}.text := toolListNames{1} + " " + toolListNames{currToolNbr};
    FOR toolIndex FROM 1 TO 6 DO
      toolList{toolIndex + 1}.text := toolListNames{toolIndex + 1};
    ENDFOR

    item := UIListView(\Result:=choice, toolList\BtnArray:=mainButtons);
    IF choice = 2 RETURN;

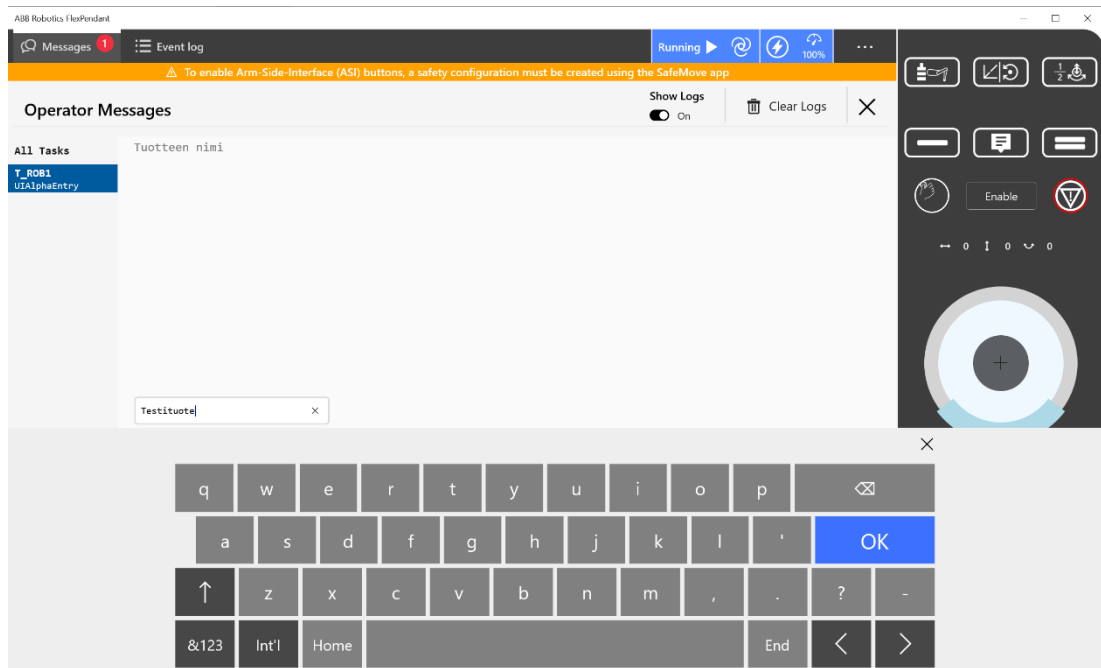
    TEST item
    CASE 1:
    CASE 2: tCurrTool := tOnRobot;
           currToolNbr := 2;
    CASE 3: tCurrTool := tOpetusTyökalu;
           currToolNbr := 3;
    CASE 4: tCurrTool := tTool1;
           currToolNbr := 4;
    CASE 5: tCurrTool := tTool2;
           currToolNbr := 5;
    CASE 6: tCurrTool := tTool3;
           currToolNbr := 6;
    ENDTTEST
  ENDWHILE

ENDPROC

```

Kuva 14. Työkalun valinta -aliohjelman koodi.

Tietyt aliohjelmat vaativat toimiakseen operaattorilta tarkempia ja erilaisissa tietotyypeissä olevia tietoja. Näiden tietojen syöttämiseksi robotin järjestelmään voidaan hyödyntää UINumEntry- ja UIAlphaEntry-funktioita sekä TPRReadFK-proseduuria. Näiden avulla voidaan avata operaattoria varten käsiyhjäimelle näkyviin esimerkiksi erilaisia viestejä, nappeja ja kosketusnäytönäppäimistö kirjoittamiseen. Kun operaattori hyväksyy syöttämänsä tiedon, palauttavat funktiot syötetyn arvon ja se määritetään haluttuun muuttujaan. Kuvassa 15 nähdään UIAlphaEntry-funktion toiminta, kun halutaan syöttää uuden kappaleen nimitiedot järjestelmään.



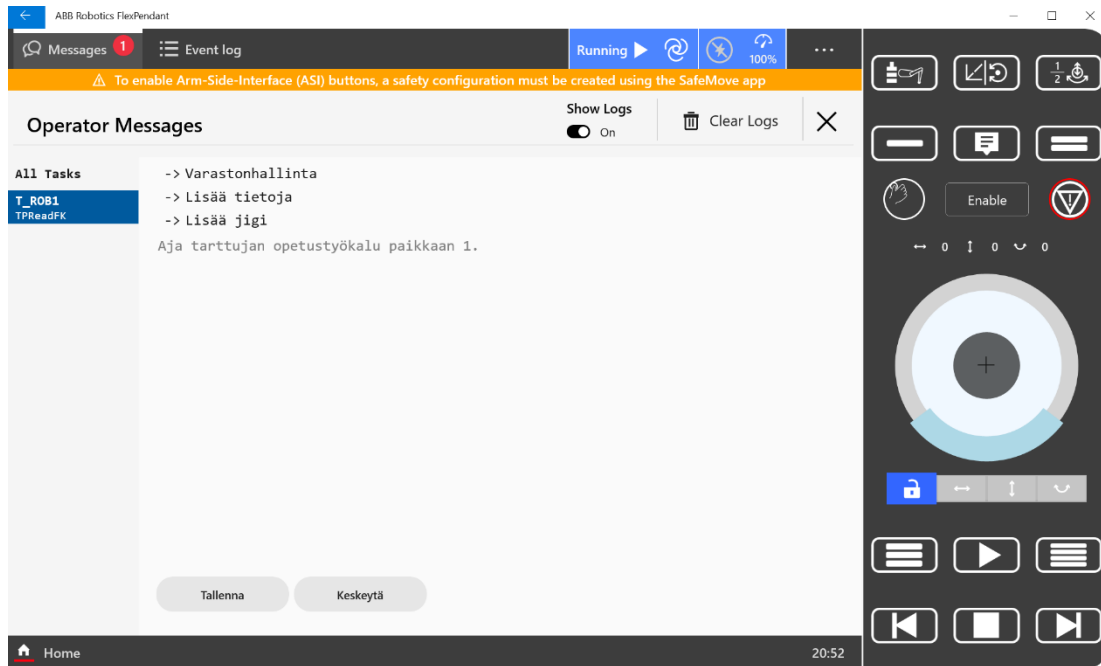
Kuva 15. Käsiohjainäkymä, kun suoritetaan UIAlphaEntry-funktio.

TPReadFK-proseduurin avulla voidaan antaa operaattorille käsiohjaimeen helposti ohjeita ja painikkeita jonkin toiminnon suorittamiseen. Seuraavassa esimerkissä, kuvassa 16, sitä käytetään robotin paikkatietojen tallentamiseen kappalejigin opetusohjelmassa.

```
TPReadFK reg4,"Aja tarttujan opetustyökalu paikkaan 1.,""Tallenna","Keskeytä",stEmpty,stEmpty,stEmpty;
IF reg4 = 1 THEN
  teachTargetX1 := CRobT (\TaskName:="T_ROB1" \Tool:=tOpetusTyökalu \Wobj:=wStorage);
ELSEIF reg4 = 2 THEN RETURN;
ENDIF
```

Kuva 16. TPReadFK-proseduurin käyttö.

Kun operaattori on valinnut uuden kappalejigin opetusvaihtoehdon käyttöliittymästä, annetaan TPReadFK-proseduurilla (Kuva 17), johon on annettu parametreiksi eri string-muotoisia tietotyyppisiä, ohjeet, mitä tehdä sekä vaihtoehdot tallentaa paikkapiste tai keskeyttää opetus. Jos operaattori on tyytyväinen paikkapisteeseen, voi hän tallentaa sen nappia painamalla, jolloin reg4 saa arvon 1. Tällöin proseduurin alapuolella oleva ehtolause täyttyy ja ohjelma suorittaa funktion CRobT, jossa robotin sen hetkinen paikkapiste määritetään teachTargetX1 rotarget -muuttujaan.



Kuva 17. Käsiohjainäkymä, kun suoritetaan TPReadFK-proseduuri.

Nämä tässä kappaleessa läpi käydyt funktiot ja proseduurit yhdistettynä listitem-tietueisiin mahdollistivat yksinkertaisen ja selkeän käyttöliittymän ohjelmoinnin.

## 7 YHTEENVETO

Tässä opinnäytetyössä käsiteltiin konepalvelun automatisointia robotiikkaa hyödyntämällä sekä perehdyttiin robottien ohjelmointiin. Tämä työ tarjoaa lukijalleen yleiskuvan konepalvelusta, erilaisista työstökoneista, erilaisista roboteista ja niiden monipuolisiin tehtäviin kykenevistä työkaluista sekä robottien ohjelmoinnista ja ohjelmistokehityksestä. Tässä opinnäytetyössä tehdyn konepalvelurobotin prototyypin ohjelma on osa laajempaa tuotekehitysprojektia.

Erilaisten työtehtävien automatisointi on teollisuuden yritysten kannalta tärkeää kilpailukyvyn ja työn mielekkyyden parantamiseksi. Työssä kehitetyn robotin ohjelman avulla voidaan vapauttaa työntekijät yksitoikkoisesta konepalvelun tehtävästä ja lisätä työstökoneiden käyttöastetta. Ohjelma toimii työn kirjoitushetkellä prototyypin ohjelmakehityksen pohjana ja se on helposti laajennettavissa kattamaan enemmän toimintoja konepalveluun liittyen.

Tässä opinnäytetyössä käytyjen pääkohtien avulla voitiin tuoda esille, mitä konepalvelun robotisointi vaatii työstökoneiden, robotin sekä ohjelmoinnin kannalta. Jotta konepalvelun robotisointi onnistuisi, vaati se perehtymistä erilaisiin työstökoneisiin, robotteihin ja niiden työkaluihin, oikeaoppiseen ohjelmistokehitykseen sekä erityisesti RAPID-ohjelmointikieleen. Näin pystyttiin luomaan mahdollisimman helppokäyttöinen ja erilaisiin tilanteisiin mukautuva robotin ohjelma, jonka avulla robotin operaattori pystyy määrittämään ohjelmaan tarvittavat tiedot ja jättämään robotin tekemään konepalvelua itsenäisesti. Edellä mainittujen seikkojen vuoksi tämä työ opetti hyvin robottien ohjelmointia ja tarjosi pohjan tämänkin prototyypin ohjelman jatkokehityksen osalta.

Robotin ohjelman ja käyttöliittymän toimintaa testattiin ensin simuloidussa ympäristössä ja sen jälkeen siirryttiin alustaviin käyttötesteihin, joissa testattiin konepalveluun liittyvien tehtävien tarkkuutta ja oikeanlaista mukautuvuutta erilaisiin tilanteisiin. Käyttötestien perusteella tämän opinnäytetyön lopputuloksena on toimiva konepalvelurobotin ohjelma, jonka datanhallinta, aliohjelmat ja muut toiminnalliset periaatteet suoritetaan ohjelmallisesti ilman virheitä sekä

noudattaen hyviä ohjelmointitapoja. Ohjelmakehityksen aikana suuren ohjelmakokonaisuuden sopeuttaminen muuttuviin ympäristöihin, työstettäviin aiheisiin ja työstökoneisiin oli haastavaa. Ohjelmakehityksen jatkuessa tämän opinnäytetyön aiheiden ulkopuolelle on tärkeää tarkastella konepalveluun liittyvien toimintojen laskennallista tarkkuutta, erityisesti paikkapisteiden opetuksen osalta. Tässä työssä tehtyjä ohjelmasosia ja käytettyä robottia voidaan tulevaisuudessa myös hyödyntää muihin teollisuudessa käytettäviin prosesseihin, kuten hitsaamiseen tai hiontaan.

## LÄHTEET

ABB. (2023). GoFa CRB 15000 Datasheet. Haettu 15.11.2024 osoitteesta <https://search.abb.com/library/Download.aspx?DocumentID=9AKK107991A8564&LanguageCode=en&DocumentPartId=&Action=Launch>

ABB. (n.d.) Technical Reference Manual RAPID Kernel. 3HAC065039-001 Revisio H.

Automate-UK. (n.d.). Robot Programming Methods. Haettu 25.12.2024 osoitteesta <https://www.automate-uk.com/our-associations/bara/expert-advice/robots-explained/robot-programming-methods/>

Automation Within Reach. (n.d.). What is Machine Tending. Haettu 11.8.2024 osoitteesta <https://www.automationwithinreach.com/what-is-machine-tending>

CNC Machines. (n.d.). What are CNC Machines. Haettu 11.11.2024 osoitteesta <https://cncmachines.com/what-is-a-cnc-machine>

Dorna Robotics. (13.10.2023). Types of Robot Grippers and Their Applications. <https://dorna.ai/blog/types-of-grippers-for-robots/>

Fuyu. (13.8.2024). Mikä on gantry- tai karteellinen robotti? Haettu 7.12.2024 osoitteesta <https://www.fuyumotion.com/fi/news/what-is-a-gantry-or-cartesian-robot/>

How To Robot. (22.4.2020). Robotic Machine Tending – Advantages and Disadvantages. Haettu 11.9.2024 osoitteesta <https://howtorobot.com/expert-insight/machine-tending-advantages-and-disadvantages>

Kemppi. (18.2.2024). Robotic Welding. Haettu 13.10.2024 osoitteesta <https://www.kemppi.com/en/blogs/robotic-welding>

Kolehmainen, P. (2023) Robottien ohjelmointi. Teoksessa Välimäki, K. & Niemelä, M. (toim.), Teollisuuden robotiikka. Suomen Robotiikkayhdistys ry.

Latokartano, J. (2023). Teollisuusrobotiikan turvallisuus. Teoksessa Välimäki, K. & Niemelä, M. (toim.), Teollisuuden robotiikka. Suomen Robotiikkayhdistys ry.

Leinonen, J. (2023) Esipuhe. Teoksessa Välimäki, K. & Niemelä, M. (toim.), Teollisuuden robotiikka. Suomen Robotiikkayhdistys ry.

Lempiäinen, J. (2023). Teollisuuden robotiikka Suomessa. Teoksessa Välimäki, K. & Niemelä, M. (toim.), Teollisuuden robotiikka. Suomen Robotiikkayhdistys ry.

Liuha, A., Salmela, A., Holamo, O-P., Asikainen, J. (2023). Työkalut. Teoksessa Välimäki, K. & Niemelä, M. (toim.), Teollisuuden robotiikka. Suomen Robotiikkayhdistys ry.

Mobile Hydraulic Tips. (24.9.2013). Hydraulic Grippers. Haettu 13.12.2024 osoitteesta <https://www.mobilehydraulictips.com/hydraulic-grippers/>

Niall, H. (n.d.). Understanding Industrial Robot Motion Types. SolisPLC. Haettu 25.12.2024 osoitteesta <https://www.solisplc.com/tutorials/industrial-robot-motion-types>

PFA Inc. (n.d.). What is an Electric Gripper. Haettu 12.12.2024 osoitteesta <https://www.pfa-inc.com/what-is-an-electric-gripper/>

Robots Done Right. (n.d.-a). What is a Delta Robot. Haettu 2.12.2024 osoitteesta <https://robotsoneright.com/Articles/what-is-a-delta-robot.html?srsltid=AfmBOorqMJqvkt7ajJnYIOTfuPdZj733iuliUaOERTXms-LrTOXdUYqiZ>

Robots Done Right. (n.d.-b). Types of Robotic Gripper Power Sources. Haettu 11.12.2024 osoitteesta <https://robotsdoneright.com/Articles/types-of-robotic-gripper-power-sources.html?srsltid=AfmBOoon57MB9lhMsrP57yboqwYxfpvfyYptFreWZSP0tjEx3kFeNEHD>

Robots Done Right. (n.d.-c). What is Robotic Cutting. Haettu 13.12.2024 osoitteesta <https://robotsdoneright.com/Articles/what-is-robotic-cutting.html?srsltid=AfmBOoqegHkU28mwVBIL4Csz5ZW2ERoxaQqzanCLY-wsnQWBiniElcqPH>

Robots Done Right. (n.d.-d). Robotic Tool Changers. Haettu 14.12.2024 osoitteesta <https://robotsdoneright.com/Articles/robotic-tool-changers.html?srsltid=AfmBOopqSwrKipGVGAd5r05qD-AwHVKBp6uS-NKGeTy15plwKYBcElo9W>

Sulkanen, A. (13.8.2024). Johdanto 3D-tulostukseen: Opas aloittelijoille. 3D-CADSOLUTIONS. <https://www.an-cadsolutions.fi/johdanto-3d-tulostukseen-opas-aloittelijoille/>

Universal Robots. (10.3.2023). Guide to machine tending with robots. <https://www.universal-robots.com/blog/guide-to-machine-tending-with-robots/>

Ye, R. (11.8.2023). What is a Machining Center: Definition, Types, Components & Applications. 3ERP. <https://www.3erp.com/blog/machining-center/>