



TEKNILLINEN TIEDEKUNTA

Geopolymeerin automatisoitu valmistus

Juho Kokkonen

Konetekniikan tutkinto-ohjelma

Diplomityö

Joulukuu 2018

TIIVISTELMÄ

Geopolymeerin automatisoitu valmistus

Juho Kokkonen

Oulun yliopisto, Konetekniikan tutkinto-ohjelma

Diplomityö 2019, 78 s. + 10 s. liitteitä

Työn ohjaajat yliopistolla: DI Antti Tikanmäki ja yliopistonlehtori, TkT Toni Liedes

Tässä diplomityössä on suunniteltu ja rakennettu robottisolun, jolla pystytään valmistamaan geopolymeerejä automaattisesti reseptin perusteella. Työn tavoitteena on saada valmistettua yksi geopolymeerinäyte täysin automaattisesti. Robottisolun lopullinen päämäärä on valmistaa geopolymeerejä erilaisilla resepteillä ja tehdä mittauksia valmistuneiden geopolymeerien ominaisuuksista.

Robottisolun koostuu manipulaattorista, useista mekaanisista osista, useista toimilaitteista ja koko systeemiä ohjaavasta ohjelmoitavasta logiikasta. Manipulaattorina toimii KUKA KR10 käsivarsirobotti, joka on ohjelmoitu WorkVisual-ohjelmalla. Ohjelmoitava logiikka on koodattu TwinCat3-ohjelmalla.

Työn teoriaosassa käsitellään tärkeimpiä piirteitä teollisuusroboteista ja niiden ohjelmoinnista. Tämän jälkeen perehdytään ohjelmoitavaan logiikkaan ja diplomityön kannalta oleellisiin ohjelmointikieliin. Teoriaosuuden lopussa esitellään teollisen Ethernetin ja kenttäväylien toimintaa sekä luokituksia. Käytännön osuudessa käydään läpi robottisolun vaatimukset ja suunnitelma. Tämän jälkeen esitellään toteutetut mekaaniset ratkaisut automaation toteuttamiseksi. Lopuksi käydään vielä lyhyesti läpi PLC:n ja KR10:n koodin rakenne, sekä robottisolun turvallisuuden toteutus. Tuloksien tarkastelu on toteutettu tekemällä yksi geopolymeerikoesarja diplomityön tavoitteen mukaisesti sekä tarkastelemalla valmistuksessa ilmenneitä ongelmia.

Avainsanat: geopolymeerit, automaatio, robotiikka, kenttäväylä

ABSTRACT

Automated geopolymer manufacturing

Juho Kokkonen

University of Oulu, Degree Programme of Mechanical Engineering

Master's thesis 2019, 78 p. + 10s.

Supervisor at University: M.Sc. (Tech.) Antti Tikanmäki and University Lecturer, D.Sc. (Tech.) Toni Liedes

The subject of this Master's Thesis is to design and build a robot cell to manufacture geopolymers automatically, guided by given recipe in Excel-form. The goal of this work is to manufacture one geopolymer sample with the robot cell as well as human could do it. Robot cells purpose is to create geopolymers with many different recipes and do some testing to analyze quality of created geopolymers.

Robot cell contains a manipulator, a lot of mechanical parts, actuators and logic which controls the whole system. Manipulator is KUKAs KR10 robot arm, which is programmed by WorkVisual-program. The programmable logic is programmed by Twin-Cat3-program.

At first in theoretical part industrial robots and the most important features of programming industrial robots are introduced. After that, programmable logic controllers and different programming languages to program PLCs are presented. The last theoretical part is about industrial Ethernet and fieldbuses. In practical part of the thesis is brought out the requirements of robot cell and how it is going to be executed. All designed mechanical parts are introduced and explained. At the end of practical part is shortly explained how PLCs and KR10s code has been built. Results is the last part of the thesis. In results all the problems that occurred during making test samples are discussed and the future plans are told.

Keywords: geopolymers, automation, robotics, fieldbus

ALKUSANAT

Tämä diplomityö on tehty Oulun Yliopistolla Biomimetiikka ja älykkäät järjestelmät -yksikössä 2018 joulukuun ja 2019 elokuun välisenä aikana. Diplomityö on osa Geobot-projektia, joka on EAKR hankkeen rahoittama.

Haluan kiittää professori Juha Röningiä, joka antoi minulle mahdollisuuden tähän diplomityöhön sekä koko biomimetiikka ja älykkäät järjestelmät -yksikön kaikkia työntekijöitä hyvästä työilmapiiristä ja avusta, jota olen saanut. Haluan erityisesti kiittää teknistä ohjaajaani Antti Tikanmäkeä, joka on ideoinut ja auttanut keksimään hyviä ratkaisuja diplomityön suhteen. Lisäksi haluan kiittää ohjaajaani Toni Liedestä hyvästä opetustyöstä Oulun yliopistolla, mikä on luonut hyvän pohjan nykypäivän tekniikan oppimiselle sekä siitä, että löysin tämän erittäin mielenkiintoisen diplomityön. Kiitos kuuluu myös Oulun yliopiston työpajalle ja Fablabille, joissa henkilökunta on ollut erittäin avuliasta ja ammattitaitoista.

Lopuksi haluan vielä kiittää vaimoani Susannaa ja ystävääni sekä työkaveriani Jesse Mattilaa, jotka ovat olleet mukana koko opiskeluaajan ja tukeneet kaikissa opintojen haasteissa.

Oulu, 19.10.2019

Juho Kokkonen

SISÄLLYSLUETTELO

TIIVISTELMÄ	
ABSTRACT	
ALKUSANAT	
SISÄLLYSLUETTELO	
MERKINNÄT JA LYHENTEET	
1 JOHDANTO	8
2 GEOPOLYMEERIT	11
2.1 Yleistä geopolymeereistä	11
2.1 Geopolymeerien vaikutus ympäristöön.....	11
2.2 Geopolymeerien valmistus ja käyttö.....	12
2.3 Geopolymeerien tutkimus Oulun yliopistossa	13
3 TEOLLISUUSROBOTIT	15
3.1 Teollisuusrobottien historiaa	15
3.2 Teollisuusrobotin määrittely	15
3.2.1 Lineaarinen robotti.....	17
3.2.2 SCARA-robotti	17
3.2.3 Nivelletty robotti (Articulate robot).....	18
3.2.4 Parallel-robotti	19
3.3 Tyypillisimmät teollisuusrobottien käyttökohteet.....	19
3.3.1 Hitsaus	20
3.3.2 Auton rungon kokoaminen	20
3.3.3 Kappaleiden siirtely ja paikoitus	21
3.4 Teollisuusrobottien ohjelmointi.....	22
3.4.1 Robotin koordinaatisto ohjelmoinnissa	22
3.4.2 Työkalun määrittäminen robotille	23
3.4.3 Online- ja offline-ohjelmointi.....	24
3.5 Teollisuusrobottien kommunikointi	25
4 OHJELMOITAVAT LOGIIKAT JA KENTTÄVÄYLÄT	27
4.1 PLC:n komponentit ja rakenne.....	28
4.2 PLC:n ohjelmointi.....	29
4.2.1 ST-ohjelmointi (Structured text, ST).....	30
4.2.2 Tilakaavio (Statechart, SC).....	30
4.3 Ethernet	31
4.4 Reaaliaikainen systeemi (RTS).....	32

	6
4.5 Reaaliaikaisten kenttäväylien luokat	33
4.6 Kenttäväylien standardit.....	35
4.6.1 EtherCAT.....	36
4.6.2 PROFINET	38
4.7 Profinet IRT:n ja EtherCAT:in vertailu.....	39
5 AUTOMAATION JA MEKANIIKAN SUUNNITTELU.....	41
5.1 Robottisolun vaatimusmäärittelyt	41
5.2 Geopolymeerien automatisoidun valmistuksen suunnitellut vaiheet.....	42
5.3 Mekaanisten osien suunnittelu	42
5.3.1 Robottisolun pohja.....	43
5.3.2 Jauhon syöttölaite	44
5.3.3 Sekoituspurkkien paikoitus.....	47
5.3.4 Korkkien paikoitus.....	48
5.3.5 Ränni jauhonsyötössä	49
5.3.6 Nestepumppu	49
5.3.7 Tarttujat.....	50
5.4 Toimilaitteiden ja IO:n valinta	52
5.4.1 Manipulaattori.....	52
5.4.2 Lineaariyksikkö	53
5.4.3 Laitetoimittajan valinta.....	53
5.4.3 Servojen valinta	57
6 LOGIIKAN OHJELMOINTI JA KOMMUNIKOINTI	60
6.1 Käyttöliittymä.....	60
6.2 PLC:n logiikka.....	60
6.3 ADS-viestintä ja reseptin luku	63
6.5 KUKA:n ja PLC:n FsoE-kommunikointi.....	64
6.6 Turvallisuuden toteutus robottisolussa.....	64
7 Testaus ja tulokset	67
7.1 Testaus	67
7.2 Tulokset	68
7.3 Jatkokehitys	70
YHTEENVETO	72
LÄHDELUETTELO	74
LIITEET	

MERKINNÄT JA LYHENTEET

6-DOF	Kuusi vapausastetta (eng. Six degrees of freedom)
AAM	Alkaliaktivoituneet materiaalit (eng. Alkali-activated materials)
CAD	Tietokoneavusteinen suunnittelu (eng. Computer Aided Design)
CNC	Numeerinen ohjaus (eng. Computerized Numerical Control)
CPF	Communication profile families
CPU	Proessori (eng. Central processing unit)
FBD	Toimilohkokaavio- ohjelmointikieli (eng. Function Block Chart)
FsoE	Safety-Over-EtherCat- protokolla
I/O	Sisään- ja ulostulo (eng. Input/Output)
ICT	Tieto- ja viestintäteknikka
IEC	International Electrotechnical Commission
IL	Ohjelista- ohjelmointikieli (eng. Instruction List)
IP	Internet Protocol
LD	Tikapuukaavio- ohjelmointikieli (eng. Ladder diagram)
OLP	Off-line- ohjelmointi (Offline programming)
PLC	Uudelleen ohjelmoitava logiikka (eng. Programmable logic controller)
RAM	Keskusmuisti (eng. Random-access-memory)
ROM	Lukumuisti (eng. Read-only-memory)
RTS	Reaaliaikainen systeemi (eng. Real time System)
SC	Tilakaavio- ohjelmointikieli (eng. Statechart)
SFC	Sekvenssikaavio (eng. Sequential function chart)
SFD	Erotusmerkki (eng. Start of Frame Delimiter)
ST	Rakenteinen teksti- ohjelmointikieli (eng. Structured text)
TCP	Työkalun keskipiste/Transmission Control protocol

1 JOHDANTO

Vuonna 1970 mekatroniikan ajateltiin olevan mekaniikan ja elektroniikan yhdistämistä. Silloin oli jo useita mekatronisia tuotteita, kuten automaattisesti avautuvia ovia, myynti-automaatteja ja itsestään keskittäviä kameroita. Vaikka 1970-luvulla mekatroniikka oli suhteellisen yksinkertaista, se oli alku kehittyneelle ohjausteknologialle. 1980-luvulla tietotekniikka tuli mukaan, kun mikroprosessorien suorituskyky parani ja tietotekniikkaa aloitettiin käyttämään kehittyneimmissä mekatronisissa tuotteissa. NC-koneet (numerical control machines) ja teollisuusrobotit pystyttiin tekemään pienemmässä koossa ja moottorien ohjausjärjestelmistä tuli suosittuja mekatronisia tuotteita teollisuudessa. (Harashima ym. 1996, s. 1)

Nykypäivänä teollisuus on täynnä robotiikkaa ja automaattisia järjestelmiä, jotka ovat kehittyneet jatkuvasti tietokoneiden laskentatehon kasvaessa. Tarve uusille automaatio- ratkaisuille on jatkuvaa, ja robottien vaatimukset ovat kasvaneet koko ajan. Robottien korvataksa aina vain vaativampia ihmisten töitä, roboteilta vaaditaan enemmän ihmisten aisteja, kuten näköä ja tuntoa. Anturien mahdollisuus mitata erilaisia suureita mahdollistaa myös kokeellisen tutkimuksen automatisoinnin, jossa voidaan tehdä samaa testiä pienillä variaatioilla, ja mitata tarvittavat tulokset testien välissä. Automatisoinnin esteenä on yleensä ennemminkin ihmisen tietämättömyys automaation nykypäivän mahdollisuuksista kuin itse tekniikka.

Samaan aikaan, kun mekatroniikka on kehittynyt, myös insinöörien työ on helpottunut. CAD-ohjelmistojen (Computer-aided design systems) kehittyessä laite- ja tuotesuunnittelu ovat nopeutuneet ja helpottuneet. Enää ei tarvita erikseen piirtäjää, suunnittelijaa ja insinööriä, vaan nämä työnkuvat on voitu yhdistää. CAD-ohjelmien avulla suunnitteluvirheet voidaan havaita aikaisessa vaiheessa ja tuotteiden valmistus on helpottunut. Myös 3D-tulostus, CNC-koneet, laserleikkurit ja muut mekatroniset laitteet ovat kehittyneet ja yleistyneet huomattavasti. Työstökoneiden yleistyessä myös niiden hinnat ovat laskeneet, sekä niiden käyttö on helpottunut. Tavalliset harrastelijatkin voivat ostaa 3D-printtereitä tai CNC-koneita ja oppia käyttämään niitä sujuvasti. Tämä on osaltaan nopeuttanut ja helpottanut insinööriratkaisujen kehittymistä.

Tämän diplomityön tavoitteena on suunnitella robottisolun, jolla pystytään tekemään alka-liaktivoitujen materiaalien eli geopolymeerien kokeellista tutkimusta. Ensisijaisesti robottisolun pitää valmistaa geopolymeerejä sille annetun reseptin mukaisesti. Tämän lisäksi robottisolun tulee olla helposti liikuteltava. Robottisolun liikuteltava alusta tulee suunnitella joustavaksi siten, että alustaan ja sen ympärille on helppo lisätä uusia toiminnallisuuksia. Robottisolun ei ole tarkoitus tulla lopulliseen versioonsa diplomityön aikana, vaan tavoitteena on onnistua tekemään annetun geopolymeerireseptin mukainen koesarja onnistuneesti.

Geopolymeerit ovat olleet viime vuosina paljon esillä niiden sementtiä korvaavien ominaisuuksiensa vuoksi, sekä niiden valmistuksen vähäisien hiilidioksidipäästöjen takia. Geopolymeerien monipuolisten kemiallisien ominaisuuksien takia, niissä riittää tutkimista pitkäksi aikaa. Geopolymeerien valmistus käsin on työlästä ja ergonomisesti haastavaa. Tutkimustyötä nopeuttaakseen ja helpottaakseen on Oulun yliopistossa aloitettu Geobot- projekti, jossa tavoitteena on suunnitella robottisolun, joka valmistaa geopolymeerejä ja tekee geopolymeerinäytteille erilaisia mittauksia. Projektin tavoitteita ovat uudenlaisen tutkimusympäristön luominen geopolymeerien automaattiseen valmistukseen, robotiikan hyödyntäminen geopolymeerien valmistuksessa, geopolymeerireseptien kehityksen automatisoiminen sekä myös löytää uusia ja tehokkaampia tapoja hyödyntää teollisuudessa syntyviä sivuvirtoja.

Robottisolun suunnittelu aloitetaan perehtymällä geopolymeerien käsin valmistuksen vaiheisiin. Kun tarvittaviin työvaiheisiin on perehdytty, voidaan aloittaa automatisoinnin suunnittelu. Automaation suunnittelussa tutkitaan ensin teollisuudessa käytettyjä automaattioratkaisuja vastaavanlaisissa sovelluksissa ja arvioidaan, mitkä ratkaisuista voisivat sopia automatisoitavaan systeemiin. Tämän diplomityön automaattioratkaisut toteutetaan pääosin teollisuusrobotilla, ohjelmoitavalla logiikalla sekä servo- ja askelmoottoreilla. Valmistusmenetelminä diplomityössä käytetään hyväksi 3D-tulostinta ja laserleikkuria. 3D-tulostus ja muovilevyjen, kuten akryyli- ja POM-levyjen, laserleikkaus ovat halpoja valmistusmenetelmiä ja soveltuvat hyvin toimilaitteiden ja erilaisien osien prototyyppien tekemiseen ja jopa valmiiden osien valmistukseen. Kestävyyttä vaativat osat tilataan työpajalta.

Diplomityö tehdään Oulun yliopiston biomimetikka ja älykkäät järjestelmät -yksikössä (BISG). BISG-yksikössä on yhdistetty tietotekniikkaa ja biologiaa. BISG:n tutkimusalueisiin kuuluvat muun muassa tiedon louhinta, koneoppiminen, robotiikka ja tietoturva. Näitä BISG:ssä sovelletaan esimerkiksi teollisuuden valmistusprosessien optimointiin ja liikkuvien robottien ympäristömonitorointiin. Tutkimusryhmän perinteisten tutkimussuuntien lisäksi BISG pyrkii myös tarjoamaan uudenlaisia avauksia parempaa ICT:tä ja bioteknologiaa varten. Nopea ICT:n ja tietotekniikan kehittyminen on avannut uusia menetelmiä, joiden avulla tutkimusalueet voidaan yhdistää konkreettisemmin. (Oulun yliopisto, 2019.)

2 GEOPOLYMEERIT

Tässä kappaleessa esitellään, mitä geopolymeerit ovat ja miksi niitä tutkitaan. Kappaleen lopuksi kerrotaan hieman niiden valmistuksesta sekä esimerkkejä mihin niitä voitaisiin käyttää.

2.1 Yleistä geopolymeereistä

Alkaliaktivoitunut materiaali (AAM) eli geopolymeerit, ovat saaneet paljon huomiota viime aikoina. Niillä on osittain mahdollista korvata tavallisen sementin käyttö rakennusmateriaalina. Tutkimuksien mukaan geopolymeereillä on jopa parempia mekaanisia ominaisuuksia kuin tavallisella sementillä valmistetuilla betonilla. Geopolymeereistä voidaan saada tulenkestäviä tai todella kevyitä, tuoden sille paljon uusia käyttömahdollisuuksia. (Provis J. 2018, s. 40)

Geopolymeeristä valmistettävien sideaineiden, sementtien ja betonien löydyttyä laaja tieteellinen kiinnostus on herännyt. Geopolymeerien monipuolisen kemian ansiosta niille on etsitty paljon erilaisia käyttötapoja. Ensimmäinen geopolymeerejä koskeva tutkimus tehtiin vuonna 1972 Cordi-Géopolymère -tutkimuslaboratoriossa Ranskassa. Nyt vuoteen 2019 mennessä on julkaistu satoja tutkimuksia ja patenteja koskien geopolymeerejä. Geopolymeerien ominaisuuksia ja käyttöä tutkitaan hyvin monella eri tieteenalalla. (Davidovits, 2007, s. 4)

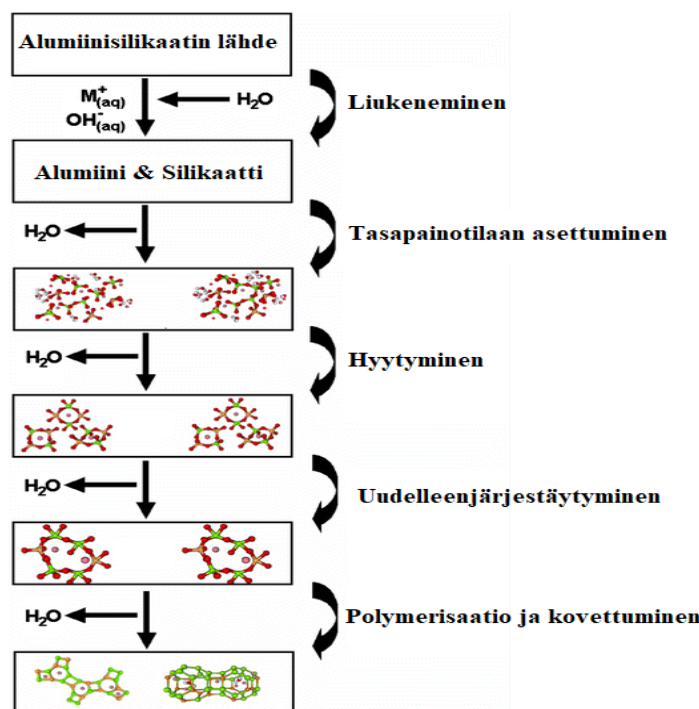
2.1 Geopolymeerien vaikutus ympäristöön

OPC on maailmanlaajuisesti käytetyin sideaine betonin valmistuksessa. Tavallisen OPC-sementin valmistuksessa syntyvien hiilidioksidipäästöjen sekä sen valmistusmäärän takia sillä on merkittävä vaikutus maailmanlaajuisiin hiilidioksidipäästöihin. Vuonna 2016 arvioitujen hiilidioksidipäästöt sementin valmistuksessa ovat 8 % koko maailman päästöistä, vaikka päästöt ovat koko ajan olleet laskussa. (Luukkonen T. et al., 2017, s. 21) Päästöt ovat silti todella suuria sementtiteollisuudessa, ja näiden päästöjen vähentämiseksi on mietitty sementtiä korvaavia vaihtoehtoja. Jatkuvien tutkimuksien myötä on huomattu, että geopolymeerit ovat todella lupaava korvaava vaihtoehto sementille. Vaikka geopolymeerit tuskin pystyvät täysin korvaamaan sementtiä, ne voisivat korvata

sementin paikallisesti kokonaan ja tätä kautta hiilidioksidipäästöt vähenevät huomattavasti. Jättemateriaalien ja sivuvirtojen hyötykäyttö geopolymeerien valmistuksessa vähentää kaatopaikoille päätyvän materiaalin määrää. (Illikainen et al., 2018, s.64)

2.2 Geopolymeerien valmistus ja käyttö

Geopolymeerien valmistukseen on kaksi tapaa: kuivan jauheen yhdistäminen veteen (one-part mix) tai jauheen sekoittaminen nestemäiseen aktivaattoriin (two-part mix). Käytetyin tapa geopolymeerien valmistuksessa on jauheen sekoittaminen nestemäiseen aktivaattoriin. (Provis J. 2018, s .41) Geopolymeerien valmistuksessa voidaan käyttää useita teollisuuden mineraalisia sivuvirtoja osana alkaliaktivointia. Geopolymeerin syntymisen vaatimuksena on, että käytettävät materiaalit sisältävät reaktiivista piitä ja alumiinia. Reaktiivista alumiinia ja piitä sisältäviä materiaaleja kutsutaan yleisesti alumiinisilikaateiksi. Alkali-aktivoinnin aikana alkuaineet liukenevat ja muodostavat uusia yhdisteitä ja materiaali kovettuu betonin kaltaisesti. Kuvassa 1 on selkeä kuvaus geopolymeerin muodostumisesta. Parhaita raaka-aineita geopolymeerien valmistukseen ovat korkealämpötilaprosesseissa syntyvät teollisuuden sivuvirrat, kuten metalliteollisuuden kuonat ja energiateollisuuden tuhkat. (Illikainen ym. 2018, s. 62)



Kuva 1. Geopolymeerien kovettuminen perustuu piin ja alumiinin kemiaan. Silikaatti- ja aluminaattiryhmät liukenevat alkalisessa hydrolyysissä. (Muokattu lähteestä Duxson et al. 2007)

Metalliteollisuuden jätteenä syntyvää masuunikuonaa ja energiateollisuudesta polttotuotteena syntyvää lentotuhkaa on tutkittu paljon OPC:n korvaajana sekä betonin lisäaineena. Näistä tutkimuksista on saatu lupaavia tuloksia. Toinen potentiaalinen vaihtoehto geopolymeerien valmistuksessa ovat mineraalivillat, kuten kivivilla ja lasivilla. Juho Yliniemen vuoden 2016 tutkimus koskien mineraalivillojen alkaliaktivointia osoittaa, että mineraalivilloilla valmistetuilla geopolymeereillä saadaan aikaan hyvinkin lujia materiaaleja. Esimerkiksi lasivillasta valmistetulla geopolymeerillä saatiin kokeissa puristuslujuudeksi 48.7 Mpa. (Yliniemi J. et al., 2016, s.9)

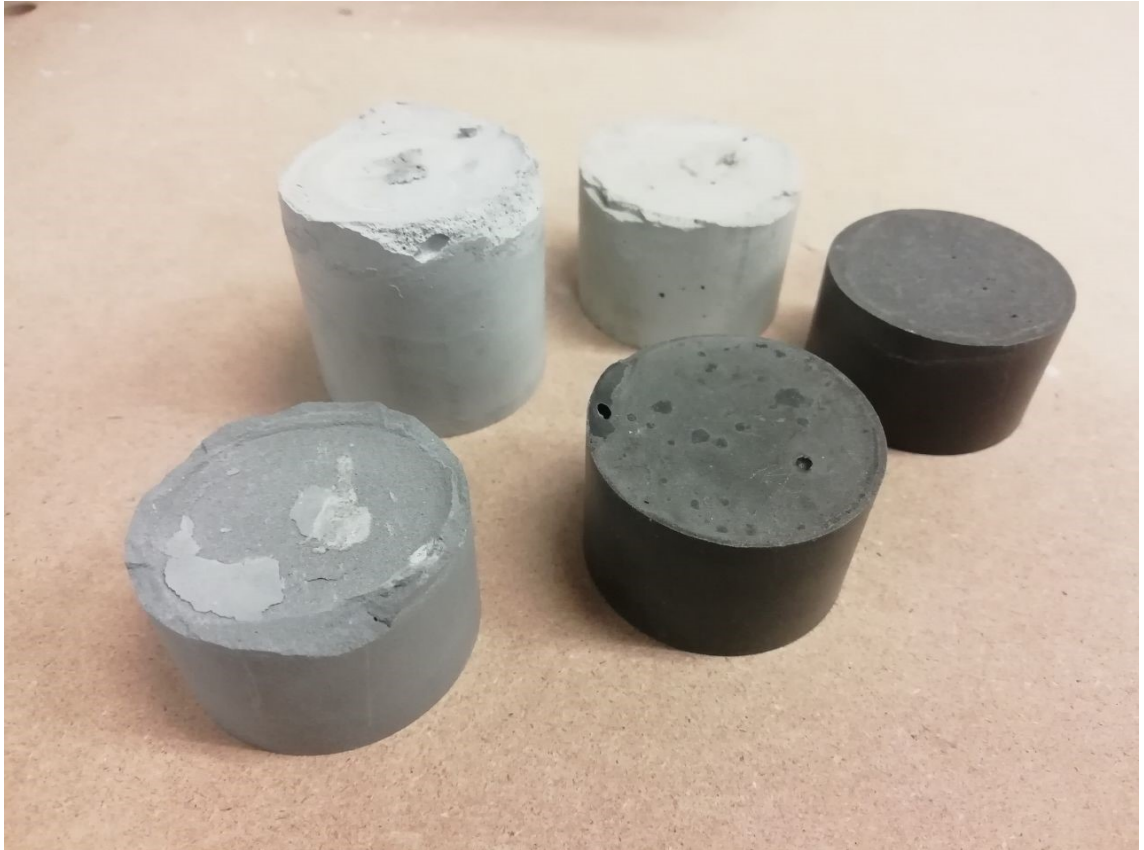
Geopolymeereistä voidaan sen monipuolisen kemian ansiosta valmistaa betonia, joka on kestävä ja lujaa, sekä samalla erittäin painavaa, kevyttä tai jopa taipuisaa. Nämä ominaisuudet tekevät geopolymeeristä kiinnostavan materiaalin erikoisovelluksiin, kuten eristemateriaaleiksi tai muovia korvaaviksi materiaaleiksi. Geopolymeeristä voidaan valmistaa myös betonia, jolla on todella hyvä happojen ja kemikaalien kestävyys. Tämä ominaisuus tekee materiaalista käyttökelpoisen muun muassa viemäriputkiin, maatalouden käyttöön tai perustuksiin happamassa maaperässä. (Illikainen S. et al., 2018, s. 64-65)

Vaikka on jo löydetty paljon käytännöllisiä geopolymeerireseptejä, niissä on vielä paljon kehitettävää. Pienet muutokset sekoitettavassa reseptissä tai geopolymeerin kuivausprosessissa muuttavat geopolymeerin ominaisuuksia huomattavasti. Provisin mukaan tulevaisuuden geopolymeeritutkimuksessa tulisi keskittyä muun muassa reseptien optimoimiseen, testausmenetelmien standardoimiseen ja geopolymeerien käytännön kokeiluihin korvaten käytössä olevia materiaaleja. (Provis J. 2018, s. 47)

2.3 Geopolymeerien tutkimus Oulun yliopistossa

Oulun yliopiston kuitu- ja partikkelitekniikan professori Mirja Illikaisen johtaman tutkimusryhmän tutkimus keskittyy teollisuuden sivuvirtojen kierrätykseen ja hyödyntämiseen geopolymeereissä ja muissa sementtiä korvaavissa sideaineissa. Tutkimusryhmään kuuluu viisitoista henkilöä, jotka tavoittelevat kansainvälisesti huipputasoisia tieteellistä tutkimusta. Tämän lisäksi tutkimustuloksia pyritään kaupallistamaan yhteistyössä yritysten kanssa. Tutkimusryhmä on ensimmäisen viiden vuoden aikana julkaissut yli 30 tieteellistä artikkelia ja tehnyt yhteistyötä yli 40 yrityksen kanssa. (Illikainen S. et al., 2018, s. 64)

Vuonna 2018 on aloitettu GEOBOT-projekti, jossa tavoitteena on hyödyntää robotiikkaa geopolymeerien tutkimisessa. Geobot-projekti toteutetaan Oulun yliopiston kuitu- ja partikkelitekniikan -yksikön, sekä biomimetikka ja älykkäät järjestelmät -yksikön yhteistyönä. Kuvassa 2 on kaksi Oulun yliopistolla valmistettuja geopolymeerinäytteitä.



Kuva 2. Oulun yliopistolla valmistettuja geopolymeerinäytteitä.

3 TEOLLISUUSROBOTIT

Tässä kappaleessa esitellään mitä teollisuusrobotit ovat. Lisäksi käydään läpi erilaisia teollisuusrobotteja ja minkälaisia ominaisuuksia niillä on. Lopuksi esitellään teollisuusrobottien käyttöön ja ohjelmointiin liittyvät oleelliset asiat.

3.1 Teollisuusrobottien historiaa

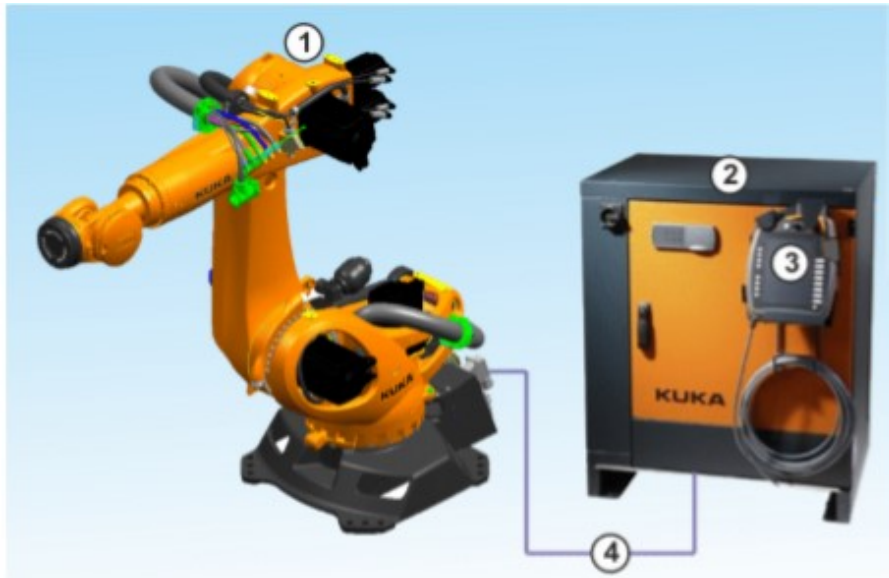
Numeerisesti ohjattavat jyrsinkoneet, tunnetummin NC-koneet, tulivat teollisuuteen 1950-luvulla. Ensimmäinen NC-kone esiteltiin MIT:ssä (Massachusetts Institute of Technology) Bostonissa vuonna 1951 ja sen kehittämistä oli johtamassa ruotsalainen J. Parson. Tuolloin NC-koneiden ohjelmointi oli vielä hyvin alkeellista. NC-kone oli alku robotiikalle sekä läpimurto, joka avasi uusia mahdollisuuksia työstökoneiden kehitykselle. (Wallén J., 2008, s. 8; Siciliano & Khatib 2016, s. 2)

Unimation oli ensimmäinen teollisuusrobotteja valmistava yritys, joka aloitti toimintansa 1960-luvulla. He vierailivat lukuisissa tehtaissa selvittääkseen teollisuuden tarpeita robotiikalle. Vuonna 1961 teollisuusrobotista tuli ensimmäinen prototyyppi ja se asennettiin General Motors:lle. Kyseisellä robotilla pystyttiin tekemään yhtä tehtävää kerrallaan ja elinikä robotilla odotettiin olevan 3 vuotta. Yksi tärkeimmistä askeleista teollisuusrobottien yleistymiseen oli se, kun Ford ilmoitti olevansa kiinnostunut asentamaan 2000 Unimationin robottia tehtaalleen. Tämän jälkeen useat muutkin amerikkalaiset yritykset huomasivat teollisuusrobottien potentiaalin ja teollisuusrobotit alkoivat yleistyä tehtaissa (Wallén J., 2008, s. 9)

3.2 Teollisuusrobotin määrittely

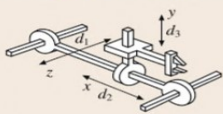
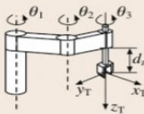
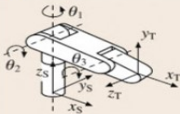
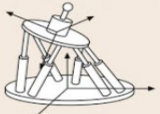
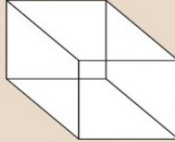


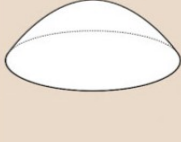




ISO 8373:2012 -standardi määrittelee teollisuusrobotin näin: ”An automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes, which can be either fixed in place or mobile for use in industrial automation applications.” Yksinkertaisuudessaan teollisuusrobotti koostuu kontrollerista eli robotin ohjaimesta, ohjelmointilaitteesta ja manipulaattorista (Kuva 3). (Saukkoriipi J., 2019, s.10-11). Teollisuusrobottia voidaan ohjata joko ohjelmointilaitteella tai kontrollerilla. Lisäksi ainakin Universal Robot:n valmistamia robotteja voidaan ohjata myös käsin liikuttelemalla. Robotin ohjaimessa on useimmiten käyttöjärjestelmänä Windows tai Linux. Esimerkiksi

KUKA:n roboteissa on Windowsin käyttöjärjestelmä ja Universal Robot:in roboteissa on Linux. Teollisuusrobotin ohjelmointi tapahtuu robotin ohjelmointilaitteella tai ulkoisella tietokoneella, josta ohjelmakoodi ladataan robotille. Useimmiten ohjelmoinnissa käytetään kumpaakin menetelmää. Teollisuusrobotin ohjelmoinnista lisää kappaleessa 3.4.



Kuva 3. KUKA teollisuusrobotti: manipulaattori (1), robottiohjain (2), ohjelmointilaite (3) ja johdotus (4). (KUKA 2015, s. 17) (Kuva julkaistaan KUKA Robotics:n luvalla.)

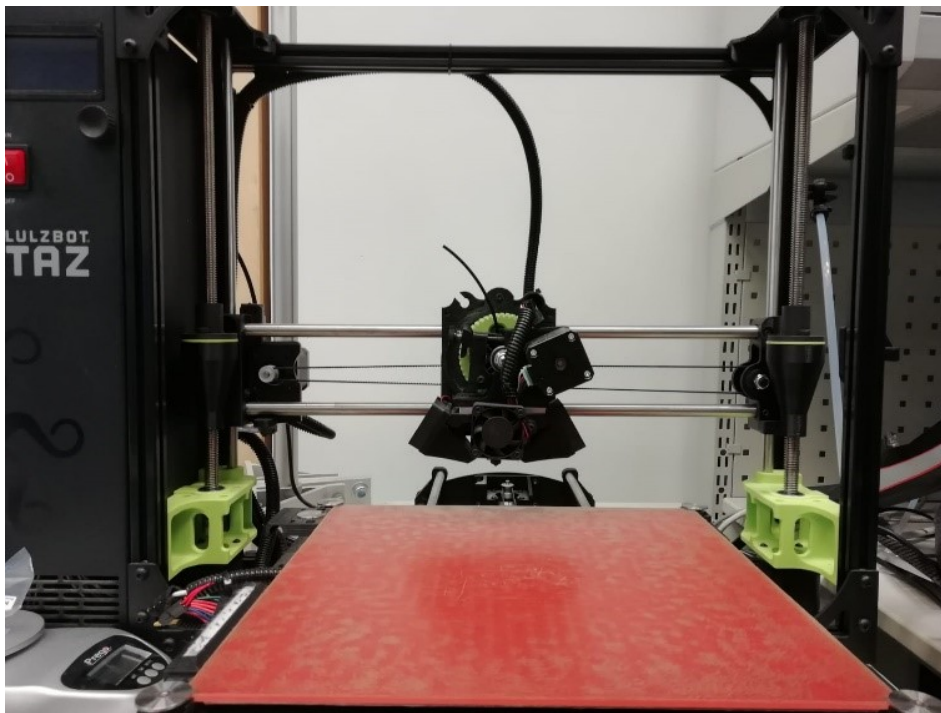
Teollisuusrobotit voidaan jakaa niiden itsenäisten akselien lukumäärän avulla sekä niiden mekaanisen rakenteen avulla. Teollisuusrobottien päämalleja on neljä. Kuvassa 4 on esitelty kaikki neljä päämallia: Lineaarinen malli, SCARA, nivelletty malli ja rinnakkainen malli. (Siciliano & Khatib 2016, s. 1392-1393)

Category	Gantry (or Cartesian)	SCARA	Articulate	Parallel
Robot main axes structure	 3 prismatic joints	 1 prismatic and 2(3) revolute joints	 3 revolute joints	 Typically with 3, 4 or 6 prismatic axis
Workspace shape				
Technical example				

Kuva 4. Erilaiset teollisuusrobottimallit. (Siciliano & Khatib 2016, s. 1393) (Kuvan julkaistaan Springer:in luvalla.)

3.2.1 Lineaarinen robotti

Lineaarisilla roboteilla on kolme lineaarista akselia, jolloin työalue on suorakulmainen särmiö. Lineaariset robotit ovat tyypillisesti 3D-tulostimia (kuva 5) ja CNC-koneita, joita löytyy lähes kaikista työpajoista. 3D-tulostimet ja CNC-koneet eivät välttämättä kuitenkaan mene teollisuusrobottien kategoriaan, koska ne eivät yleensä ole kovin monikäyttöisiä. Liikealue mahdollistaa robotin käytön myös esimerkiksi lavojen täytössä tai erilaisissa annostelusovelluksissa. Erityisesti lääketeollisuudessa ja erilaisissa laboratorioissa voidaan nähdä lineaarimallisia annostelurobotteja.



Kuva 5. Lulzbot TAZ Pro 3D-tulostin.

3.2.2 SCARA-robotti

SCARA-robotilla (Selective Compliance Assembly Robot Arm) on yksi lineaarinen akseli ja 2-3 pyörähdysakselia. SCARA-robotit (kuva 6) ovat parhaimmillaan kappaleiden poimimisessa ja siirtelyssä pisteestä toiseen niiden liikealueen, nopeuden ja tarkkuuden ansiosta. Pyörähdysakselit pyörivät Z-akselin ympäri, eli robotti liikkuu sujuvasti X-Y -tasossa. Lineaarinen akseli kulkee Z-akselin suuntaan ja sitä käytetään kohteiden poimintaan tai paikalleen laittoon.



Kuva 6. SCARA-robotti (FANUC 2019) (Kuva julkaistaan FANUC:in luvalla).

3.2.3 Nivelletty robotti (Articulate robot)

Nivelletty robotti koostuu yleensä kuudesta nivelestä, mutta myös jopa kymmennivelisiä robotteja on saatavilla. Kuvan 7 robotin kuusi akselia mahdollistavat robotin ajamisen jokaiseen pisteeseen ja asentoon robotin työtilassa. Näin robotti kykenee kulkemaan tarkkoja ja monimutkaisia liikeratoja, mikä vaaditaan esimerkiksi hitsauksen automatisoimiseen. Verraten muihin robottityyppeihin, nivelletyillä roboteilla on todella korkeat maksimikuormat ja niillä on hyvä ulottuvuus. Siksi niitä käytetäänkin paljon teollisuudessa. (Siciliano & Khatib 2016, s. 1393).



Kuva 7. ABB IRB1600 6-DOF robotti. (ABB 2019) (Kuva julkaistaan ABB:n luvalla)

3.2.4 Parallel-robotti

Parallel-mallisista roboteista yksi tunnetuin esimerkki on Delta-robotti (kuva 8). Delta-robotti on jäykkä rakenteeltaan ja sen hitaudet ovat pieniä, mikä mahdollistaa robotin tarkkuuden ja suuren nopeuden. Delta-robotti voi saavuttaa jopa 100m/s^2 kiihtyvyyden. Delta-robotteja käytetään pääasiassa poimi ja pudota- tehtäviin, joissa tarvitaan suuria nopeuksia ja kuormat ovat suhteellisen pieniä. (Siciliano & Khatib 2008, s. 966)



Kuva 8. Delta-robotti. (ABB 2019) (Kuva julkaistaan ABB:n luvalla.)

3.3 Tyypillisimmät teollisuusrobottien käyttökohteet

Teollisuudessa pyritään jatkuvasti automatisoimaan mitä haastavampia tehtäviä ja uusien robottien tarve kasvaa samaa vauhtia. Tuotanto teollisuudessa on nopeutunut huomattavasti, kun robotit ovat korvanneet useat tärkeät tehtävät. Tehokkuuden lisäksi myös tapaturmien määrä on vähentynyt, kun tapaturmille altistavia tehtäviä on automatisoitu teollisuusroboteilla. Teollisuusrobottien ensimmäiset käyttökohteet koneistuksessa ja autoteollisuudessa ovat nykypäivänä edelleen tärkeimpien automatisoitujen työtehtävien joukossa. Tyypillisimmistä teollisuusrobottien käyttökohteista esitellään muutama esimerkki tärkeimmistä nykypäivän teollisuuden robotiikan sovelluksista Sicilianon ja Khatibin (2008, s. 969-974) mukaan.

3.3.1 Hitsaus

Hitsaus on teollisuudessa yksi tärkeimmistä kappaleiden liittämismenetelmistä. Käsintehty hitsaus vaatii paljon taitoa hitsaajalta ja pienetkin virheet saattavat johtaa vakaviin seurauksiin. Näistä syistä sarjatuotannon hitsaustehtävät on siirretty useissa tehtaissa robottien tehtäväksi. Modernit teollisuusrobotit pystyvät hitsauksessa äärimmäisen hyvään toistotarkkuuteen ja nopeaan työhön. Teollisuusrobottien toistotarkkuus nykypäivänä on tyypillisesti noin 0.1 mm luokkaa ja nopeudet yltyvät jopa 8 metriin sekunnissa. Tyypillisesti hitsauksessa käytetään 6-DOF -käsivarsirobotia, jolla voidaan ajaa hyvinkin monimutkaisia liikeratoja eri asennoista.

3.3.2 Auton rungon kokoaminen

Auton rungon kokoaminen sisältää paljon raskaita ja vaarallisia työvaiheita. Metallilevyjen käsittely, asettelu, hitsaus ja auton kekkien siirtely käsin on hyvin vaarallista ja fyysisesti vaativaa. Nykyään robotit pystyvät siirtelemään satojen kilojen osia kevyesti ja tarkasti, hitsaten osat suurella toistotarkkuudella erittäin nopeasti. Tyypillisesti autojen valmistuslinjasto koostuu useasta linjasta, joiden toimivuus riippuu toisistaan (kuva 9). Tällainen jatkuva valmistus usean linjan yhteistyöllä vaatii erittäin suurta toimintavarmuutta. Keskiarvo virheiden tiheydelle auton valmistuslinjoilla on ilmoitettu olevan noin yksi virhe 50 000 tunnin välein.



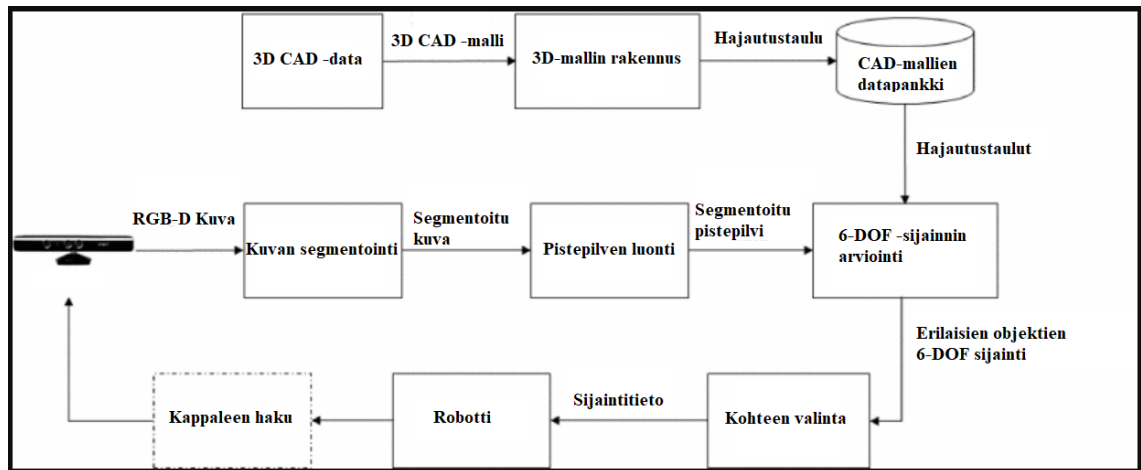
Kuva 9. Autoteollisuuden linjasto (Siciliano & Khatib 2016, s. 1396) (Kuva julkaistaan Springer:in luvalla.)

3.3.3 Kappaleiden siirtely ja paikoitus

Yksinkertaisuudessaan robottisolun suunnittelussa pyritään löytämään sopiva suhde poimittavien kappaleiden paikoituksen ja tarvittavien anturien kanssa. Yleensä pyritään siihen, että poimittavat kappaleet saapuvat robotille aina samassa asennossa samoille paikoille. Tämä toteutetaan siten, että poimittavat kappaleet laitetaan säiliöihin tai laitteisiin, jotka järjestävät poimittavat kappaleet aina samalla lailla tärinän, painovoiman tai jonkin erityismekanismin avulla.

Monimutkaisen muotoisten kappaleiden asentoa on usein vaikea hallita. Niiden paikoitukseen voidaan käyttää esimerkiksi 3D- tai 2D-kameroita ja konenäköä. Yksi tapa käyttää 3D-kameraa satunnaisen kappaleen asento ja sijainti kolmiulotteisessa avaruudessa, on käyttää kappaleen CAD-mallia ja 6-DOF-sijainnin arviointia. 2D-kameroita voidaan käyttää paikoitukseen, jos paikoitettavat kappaleet ovat samassa tasossa. 2D-kuvista voidaan opettaa tietokoneelle, minkälaisia muotoja kuvasta ollaan etsimässä. Kappaleen muoto kertoo sen asennon ja sijainnin 2D-kuvassa. Kamera täytyy aina kalibroida esimerkiksi kalibroitiruudukolla, jotta kuvan mittasuhteet saadaan selville.

Kappaleen kolmiulotteisen asennon ja -paikan tunnistuksessa on kolme osaa (Kuva 10): CAD-mallin data, 6-DOF-sijainnin arviointi ja kohteen valinta. Kinect-sensori ottaa RGB-D-kuvia ja otetut kuvat segmentoidaan, jonka jälkeen segmentoidusta kuvasta muodostetaan pistepilvi. Pistepilven ja CAD-mallin avulla pystytään tunnistamaan mikä kappale on kyseessä sekä missä pisteessä ja asennossa kappale on. (Wu C., S. Jiang & Song K. 2015, s. 1645-1646) Tämän jälkeen tieto pisteen sijainnista ja asennosta annetaan robotille. Tämän sijaintitiedon perusteella robotin tai ohjelmoijan täytyy laskea pisteen todellinen 6-DOF-sijainti suhteessa työkalun keskipisteeseen (TCP), jolloin robotti voi hakea kappaleen työkalullaan.



Kuva 10. Kappaleen tunnistus 3D-kameralla. (Muokattu lähteestä Wu C., S. Jiang & Song K. 2015, s. 1646)

3.4 Teollisuusrobottien ohjelmointi

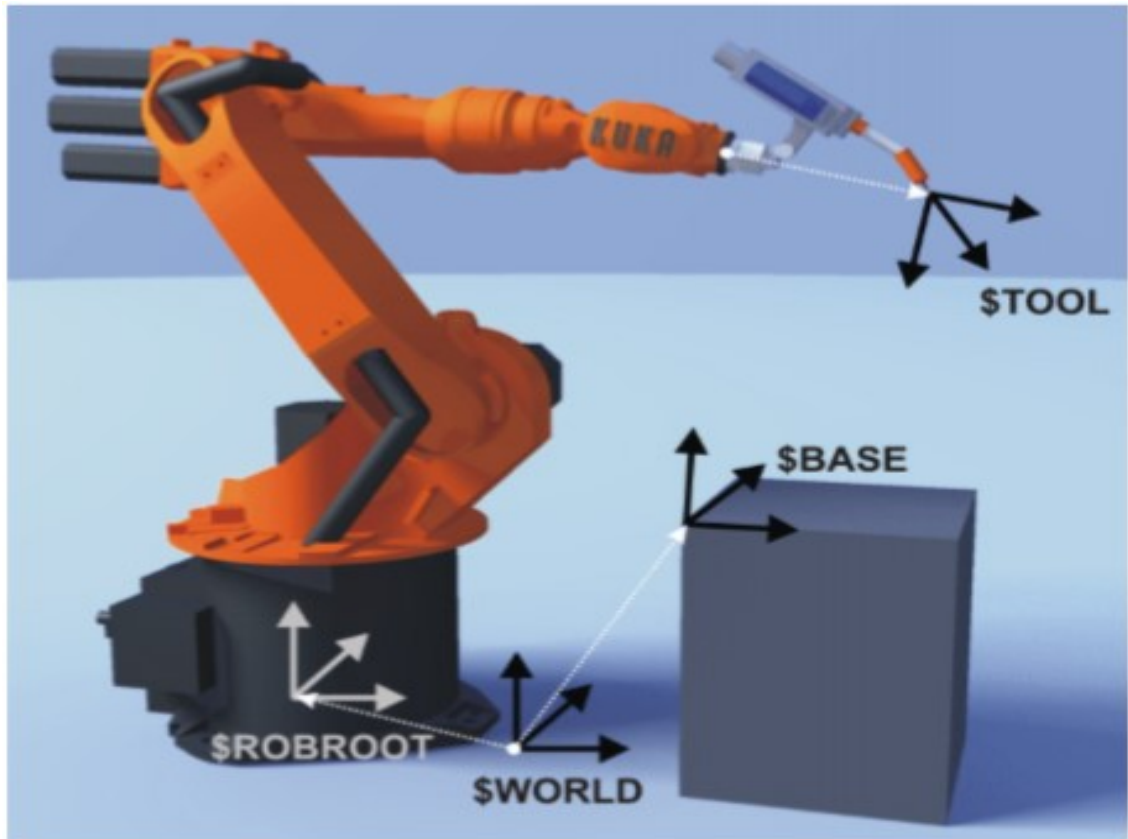
Teollisuusrobotteja voidaan ohjelmoida useilla eri tavoilla. Riippuen robotin valmistajasta, niiden ohjelmointiympäristöt vaihtelevat paljon. Teollisuusrobottien ohjelmoinnissa on kuitenkin tietyt periaatteet, jotka täytyy toteuttaa helposti käytettävän ja muokattavan koodin syntymiseksi. Tässä kappaleessa esitellään tärkeimpiä elementtejä teollisuusrobottien ohjelmoinnin kannalta. Tärkeimmät elementit teollisuusrobotin ohjelmoinnista on poimittu KUKA:n manuaaleista.

3.4.1 Robotin koordinaatisto ohjelmoinnissa

6-DOF-käsivarsirobotit käyttävät kolmiulotteista karteesi-koordinaatistoa, jossa määritellään koordinaatiston sijainti X-, Y- ja Z-akseleilla sekä asento näiden akseleiden ympäri. Robotilla on aina kantakoordinaatisto (Kuvassa 11 \$ROBROOT), joka on tavallisesti robotin kiinnityskohdan keskellä. Muut koordinaatistot määritellään tavallisesti kantakoordinaatiston suhteen. Erityisesti työalueen koordinaatistosta on apua, jos työalueet ovat etäällä toisistaan tai työalue on eri asennossa kuin kantakoordinaatisto. Kun työalue tai -piste muuttuu jälkikäteen, työalueiden koordinaatistojen avulla robotin ohjelmoija pystyy vaihtamaan työpisteen paikkaa lisäämällä koordinaatistoon siirtymän tai asennon muutoksen. Tämä nopeuttaa ohjelmointia huomattavasti, kun muita muutoksia ei vaadita.

Koordinaatistojen tarkkaan määrittämiseen voidaan käyttää robotin ohjelmointilaitetta. Ohjelmointilaitteissa on yleensä valmis työkalu, joka kertoo ohjeet koordinaatiston määrittämiseen. Origin piste tallennetaan ohjelmointilaitteella ja sen jälkeen tallennetaan

vielä kaksi pistettä työtasolla. Yksi piste valitaan X-akselilta ja toinen Y-akselilta. Kolmen pisteen avulla tietokone laskee automaattisesti koordinaatiston paikan ja asennon. Koordinaatiston tarkkaan määrittämiseen voidaan käyttää robotin työkaluna terävää kärkeä, joka helpottaa tarkkoihin pisteisiin ajamista (Kuva 12).

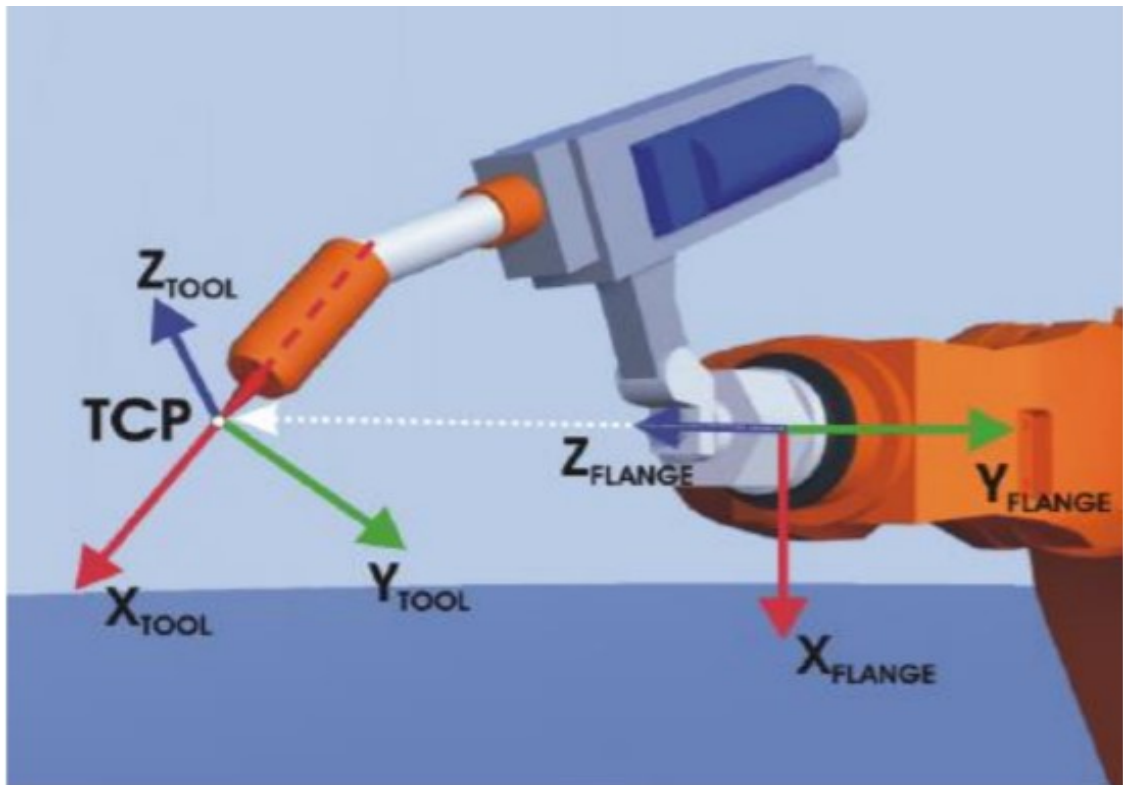


Kuva 11. Robotin koordinaatistot. (KUKA 2015, s. 64) (Kuva julkaistaan KUKA Robotics:n luvalla.)

3.4.2 Työkalun määrittäminen robotille

Työalueiden koordinaatistojen lisäksi on oleellista määrittää myös käytettävien työkalujen TCP:t sekä työkalujen painopisteet ja massat. Yleensä nämäkin työkalun määrittelyt voidaan tehdä robotin ohjelmointilaitteella. Työkalun koordinaatiston määrittämiseen vaadittu tarkkuus vaihtelee riippuen sovelluksesta. Jos esimerkiksi työkalua halutaan pyörittää TCP:n ympäri tietyssä pisteessä, niin pienikin virhe määritetyssä pisteessä vaikuttaa kiertämiseen kriittisesti.

Työkalun massan ja keskipisteen määrittäminen taas vaikuttaa oleellisesti robotin suorituskykyyn ja liikkeen sulavuuteen. Robotin servojen säätöparametrit muuttuvat kuorman mukaan. Jos kuorma on määritelty väärin, robotin tarkkuus ja nopeus heikkenee ja se aiheuttaa värähtelyä, joka näkyy liikkeissä.



Kuva 12. Työkalun koordinaatisto. (KUKA 2015, s. 125) (Kuva julkaistaan KUKA Robotics:n luvalla.)

3.4.3 Online- ja offline-ohjelmointi

Robottien ohjelmointikielet ja -ympäristöt on jaettu online- ja offline (OLP)-ohjelmointiin. Online-ohjelmoinnissa käytetään robotin omaa ohjelmointilaitetta, jolloin työskentely tapahtuu robotin läheisyydessä. OLP-ohjelmoinnissa ei kosketa robottiin ol- lenkaan, vaan robottia ohjelmoidaan etänä OLP-ohjelmistolla, jolloin robotin toimintaa ei tarvitse keskeyttää (Nilsson K. 1996, s. 29).

Offline-ohjelmistoympäristöön voidaan siirtää suoraan realistisia CAD-malleja robotin työympäristöstä ja simuloida robotin toimintaa visuaalisesti realistisessa ympäristössä. Offline-ohjelmoinnilla voidaan saavuttaa merkittäviä säästöjä tuotteesta ja sarjakoosta riippuen, koska robottia voidaan ohjelmoida keskeyttämättä tuotantoa. Ideaalissa tilanteessa OLP-ohjelmistolla tehty koodi voitaisiin ladata robottiin ja aloittaa tuotanto uudella koodilla välittömästi. Käytännössä kuitenkin kappaleiden geometria ja työalueet työsolussa vaativat koordinaattien kalibroinnin (Pan Z. ym. 2011, s.91).

Kaikilla isoilla teollisuusrobottien valmistajilla on omat offline- ja online-ohjelmistot. Taulukosta 1 nähdään tunnetuimpia offline-ohjelmistoja.

Taulukko 1. Tunnetuimpia offline-ohjelmistoja (Pan Z. ym. 2011, s. 91).

	Ohjelmisto	Yhtiö
Yleisiä OLP-ohjelmistoja	Robomaster	Robomaster
	Robsim	Camelot
	Cosimir	Festo
OLP-ohjelmistoja robotivalmistajilta	Robotstudio	ABB
	KUKA-Sim	KUKA
	Roboguide	Fanuc
	MELFA WORKS	Mitsubushi
	Pc-ROSET	Kawasaki
Akateemisia/Vapaan lähteen OLP-ohjelmistoja	MATLAB	Useita MATLAB-pohjaisia ohjelmistoja
	Coppelia Robotics	V-REP
	RoboDK	RoboDK

3.5 Teollisuusrobottien kommunikointi

Teollisuudessa teollisuusrobotti on harvoin ainut laite automatisoidussa systeemissä. Tällöin robotti ja sen ympärillä olevat laitteet tarvitsevat rajapinnan, jonka kautta ne pystyvät jakamaan informaatiota ja toimimaan yhdessä. Perinteisenä rajapintana on aina voitu käyttää rinnakkaista I/O:ta, jossa jokainen signaali vedetään erillisellä johdolla. Monimutkaisemmissa systeemeissä tällainen kommunikointitapa on kuitenkin työläs kytkeä ja johtojen merkkkaus on hidasta.

Nykyaikana teollisuudessa pyritään käyttämään Ethernet-pohjaisia kenttäväyliä automaation kommunikointiin, jossa kaikki data kulkee kahden laitteen välillä yhden Ethernetin kautta. Kenttäväylä yhdistää automaation systeemin kaikki laitteet kuten moottorit, robotit ja sensorit. Teollisuuden automaatiossa kenttäväyliltä vaaditaan usein reaaliaikaisuutta servomoottorien ohjaukseen ja turvalaitteiden nopeaan vasteeseen. Tuotannossa pienet

virheet ja viiveet turvalaitteissa tai servo-ohjauksessa voivat aiheuttaa taloudellisia vahinkoja ja jopa henkivahinkoja. Kuitenkin ylemmän tason kommunikointi voidaan usein toteuttaa käyttämällä standardia Ethernet-laitteistoa tukevaa kenttäväylyä. Kenttäväyliä on lukuisia erilaisia ja eri laitevalmistajat suosivat eri kenttäväyliä. Osa kenttäväylyistä on kuitenkin avoimia ja ne ovat voidaan yhteensovittaa myös muiden kenttäväylien kanssa. Teollisuusrobotit pystytään usein konfiguroimaan useille eri kenttäväylyille. Kenttäväylyistä kerrotaan lisää kappaleissa 4.3-4.6

Teollisuusrobotit voivat olla hengenvaarallisia laitteita, jos ihminen joutuu niiden työalueelle. Tämän takia turvastandardit ovat teollisuusroboteilla hyvin tarkkoja. Teollisuusrobotteja ei voi lainkaan ajaa automaattiajolla ennen asianmukaisia turvakytkentöjä. Tavallisesti kaikki teollisuusrobotin turvakytkennät voidaan tehdä suoraan robotin ohjaimen. Jos automaattiosysteemissä on kuitenkin useita laitteita, jotka pitää kytkeä turvapiiriin, on järkevää toteuttaa turvapuoli kenttäväylän kautta. Tällöin kaikki laitteet on kytketty samaan kenttäväylään ja kaikki laitteet saadaan helposti kytkettyä samoihin turvalaitteisiin yhdellä turvakytkennällä.

4 OHJELMOITAVAT LOGIIKAT JA KENTTÄVÄYLÄT

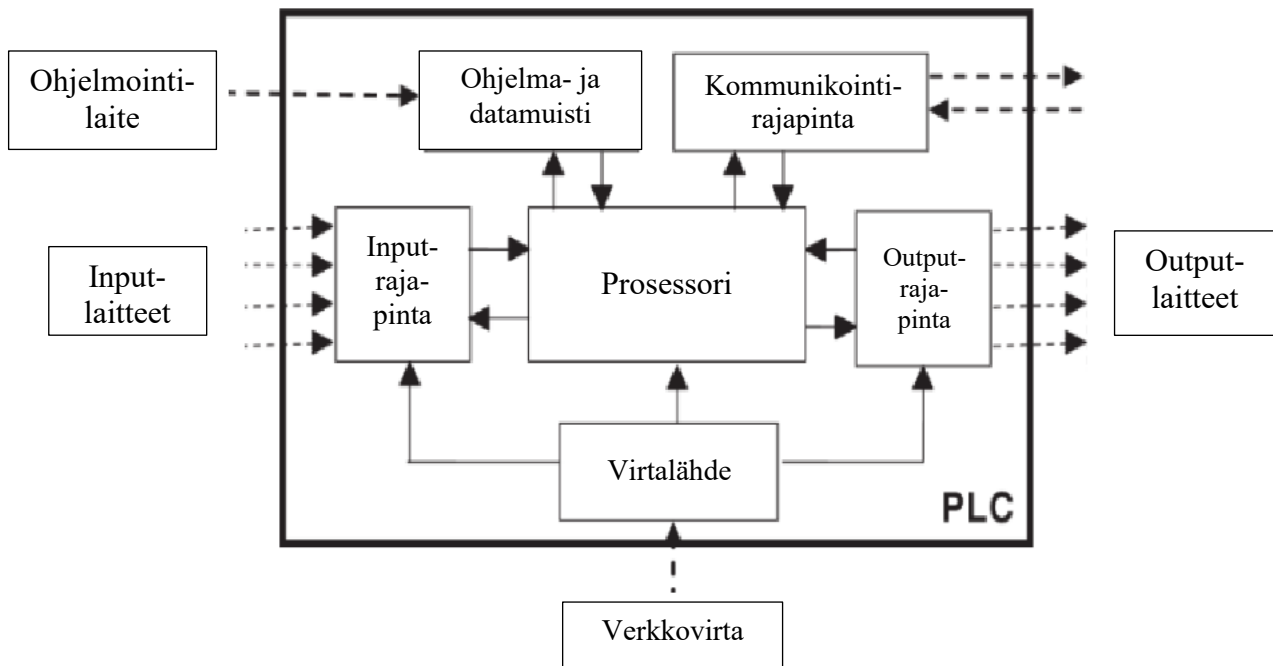
Tässä kappaleessa perehdytään ohjelmoitavan logiikan eli PLC:n perusrakenteeseen ja toimintaan sekä PLC:n erilaisiin ohjelmointikieliin. Ohjelmointikielistä perehdytään ST-ohjelmointiin (Structured text) sekä Beckhoffin ohjelmointiympäristön TwinCat:in omaan tilakaavioon, jotka ovat työn kannalta oleelliset ohjelmointikielet. Tämän jälkeen esitellään standardisoitu Ethernet, joka toimii myös kenttäväylien viestinnän pohjana. Kenttäväylistä esitellään niiden eri luokat sekä erilaiset vaihtoehdot. Koska kenttäväyliä on hyvin paljon erilaisia niin niistä esitellään nimeltä vain käytetyimmät. EtherCat ja Profinet esitellään tarkemmin, koska ne ovat työn kannalta oleellisimpia. Kappaleen lopuksi vertaillaan vielä esiteltyjen kenttäväylien eroavaisuuksia.

Ensimmäinen PLC kehitettiin vuonna 1969 ja nyt ne ovat laajalti käytössä itse tehdyissä pienissä sovelluksissa sekä myös isommissa teollisuuden sovelluksissa. PLC on mikroprosessoriin pohjautuva ohjain, jolla voidaan suorittaa toimintoja, kuten logiikkaa, ajastimia ja laskentaa. PLC käyttää ohjelmoitavaa muistia käskyjen tallettamiseen. Näiden ominaisuuksien avulla se pystyy ohjaamaan laitteita ja prosesseja. (Bolton W. 2019, s. 4) PLC:t ovat samantyyppisiä tietokoneisiin verrattuna. Ero näiden välillä on se, että tietokoneet on optimoitu laskentaan ja tuottamaan kuvaa näytölle, kun taas PLC on optimoitu ohjelmoitujen tehtävien suorittamiseen ja teollisuusympäristöön. PLC:t ovat myös reaaliaikaisia, toisin kuin tietokoneet. PLC:t on suunniteltu kestämään tärinää, lämpötilan vaihteluita, kosteutta ja melua. Ne ovat helposti ohjelmoitavia ja niiden ohjelmointikieli on vaivaton oppia.

Kenttäväylät tarkoittavat teollisuuden verkkoa, jota käytetään laitteiden väliseen kommunikointiin ja ohjaukseen. Yhteinen tekijä kenttäväylillä on se, että ne kaikki käyttävät standardisoitua Ethernetia pohjanaan. Automaatiosysteemin reaaliaikainen ohjaus PLC:llä tarvitsee aina reaaliaikaisen kenttäväylän. Reaaliaikaisia teollisuus-Etherneteja, kuten myös muita kenttäväyliä, on tarjolla useita erilaisia erilaisiin tarpeisiin. (Tunis S. 2015, s.11)

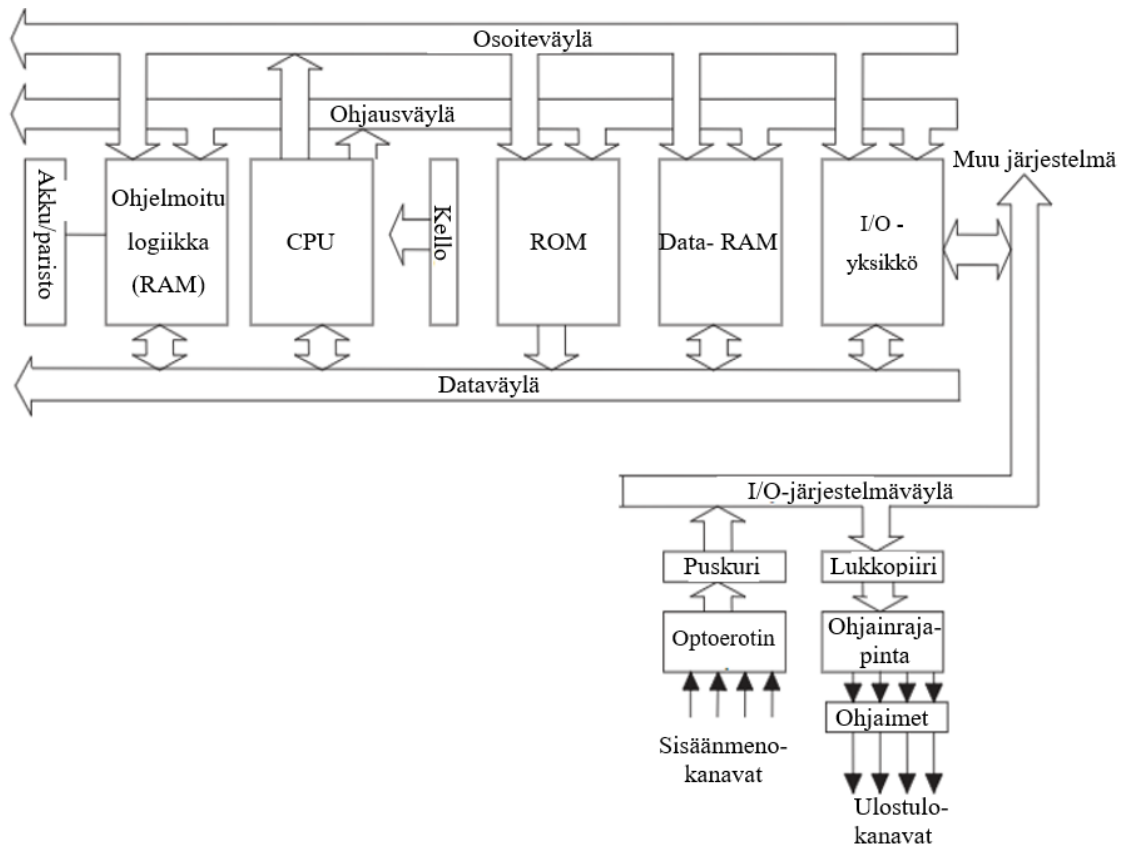
4.1 PLC:n komponentit ja rakenne

Tyypillisessä PLC-systeemissä perustoiminnalliset osat ovat prosessori (CPU), muisti, virtalähde, input/output (I/O) rajapinta, kommunikointirajapinta ja ohjelmointilaite (kuva 13).



Kuva 13. Tyypillisen PLC:n laitteisto. (Muokattu lähteestä Bolton W. 2019, s. 4)

Kuvasta 14 nähdään PLC:n perusrakenne. CPU ohjaa ja prosessoi kaikki operaatiot PLC:ssä. PLC:n prosessorissa on kello, joka määrää PLC:n suoritusnopeuden. PLC:n sisällä liikkuva data on digitaalista ja datan kulkureittejä kutsutaan väyliksi. CPU käyttää dataväylää (data bus) lähettämään dataa eri toiminnallisten yksiköiden välillä. Dataväylällä digitaaliset viestit kulkevat 8 bitin suuruisina. CPU käyttää osoiteväylää (address bus) etsimään dataa osoitteen osoittamista muistipaikoista ja jokaisella muistipaikalla on oma uniikki osoitteensa. Ohjausväylää (control bus) käytetään ohjaamaan PLC:n sisäisiä toimintoja olipa kyse sitten IO-viesteistä tai ajoitusviesteistä synkronisointia varten. Järjestelmäväylää (system bus) käytetään IO-kommunikaatioon. (Bolton W. 2019, s. 5-6)



Kuva 14. PLC:n perus rakenne. (Muokattu lähteestä Bolton W. 2019, s. 5)

PLC:llä on kahden tyyppistä muistia, ROM-muistia ja RAM-muistia. ROM-muisti on varattu käyttöjärjestelmälle ja pysyvälle datalle, jota CPU käyttää. RAM-muistia taas käytetään datan käsittelyyn ja PLC:lle ohjelmoituun ohjelmaan, jota PLC:n ohjelmoija pystyy muokkaamaan. (Bolton W. 2019, s. 6)

4.2 PLC:n ohjelmointi

PLC:n ohjelmointiin on useita erilaisia tyyliä. Erilaiset ohjelmointitavat ovat syntyneet helpottamaan insinöörejä, joilla ei ennestään ole ollut tietämystä ohjelmoinnista. Suurin osa PLC-valmistajista ovat kehittäneet uusia helpottavia tapoja ohjelmoida PLC:tä. Luokuisien erilaisten ohjelmointikielien variaatiot osoittautuivat kuitenkin ongelmallisiksi. Tämän takia, vuonna 1993 IEC (*International Electrotechnical Commission*) julkaisi IEC 61131-3 -standardin, joka määrittelee PLC-ohjelmoinnissa käytettävät ohjelmointikielien. (IEC 61131-3, 2003, s. 8)

IEC 611631-3 Standardiin kuuluvat ohjelmointikielet:

- Tikapuukaavio (Ladder diagram, LD),
- Ohjelista (Instruction List, IL),
- Toimilohkokaavio (Function Block Diagram, FBD),
- Sekvenssikaavio (Sequential function chart, SFC),
- Rakenteinen teksti (Structured text, ST).

Näiden ohjelmointikielien lisäksi PLC-valmistajilla on vielä nykyäänkin omia ohjelmointikieliä. Seuraavissa kappaleissa on PLC:n ohjelmoinnista kaksi esimerkkiä. Toinen esimerkeistä on ST-ohjelmoinnista ja toinen on IEC 611631-3 standardin ulkopuolisesta ohjelmointikielestä, TwinCat:in tilakaaviosta.

4.2.1 ST-ohjelmointi (Structured text, ST)

ST-ohjelmointikieli on tekstimuotoinen ohjelmointikieli ja se muistuttaa vahvasti Pascal-ohjelmointikieltä. ST-ohjelmointi on PLC:n ohjelmointikielistä kaikista helposti lähestyttävimmän insinööreille, joilla on jo ennestään ymmärrystä ohjelmoinnista. ST-ohjelmointi sopeutuu monimutkaiseenkin logiikkaan ja se pysyy selkeänä, jos koodi rakennetaan hyvin.

4.2.2 Tilakaavio (Statechart, SC)

Tilakaavio on Beckhoffin TwinCat-ohjelmistosta löytyvä ohjelmointikieli, joka ei kuulu IEC 61131-3 -standardiin. Se on graafinen ohjelmointikieli ja kuten monissa muissakin graafisissa ohjelmointikielissä, ohjelmointi itse dokumentoi toimintaansa hyvin. Tilakaavion toiminta on hyvin lähellä SFC-ohjelmointia, mutta se on helppolukuisempi. Se toimii myös hyvin tilakoneena selkeyden ja helpon ohjelmoinnin takia. Monimutkainen koodi saattaa näyttää tilakaaviossa epäselvältä, joten se ei ole aina vaihtoehto. Suurimpia etuja tilakaaviossa on se, että siitä näkee ohjelman kulun suoraan ja ulkopuolinenkin pääsee nopeasti perille koodista.

4.3 Ethernet

IEEE 802.3 on virallinen Ethernet-standardi. Ethernet koostuu standardista laitteistosta ja viestintäprotokollista, jotka toimivat yhdessä siirtäen dataa tietokoneiden välillä. Ethernet muodostuu neljästä osasta: kehys, MAC-protokolla, viestintälaitteisto sekä kaapelit ja muu laitteisto, jolla digitaaliset Ethernet-signaalit saadaan välitettyä. (Spurgeon C. 2000, s. 23) Tämän kappaleen tarkoitus on luoda pohjaa kenttävylien ymmärtämiselle. Tässä kappaleessa keskitytään Ethernet II -kehyyksen sisältöön.

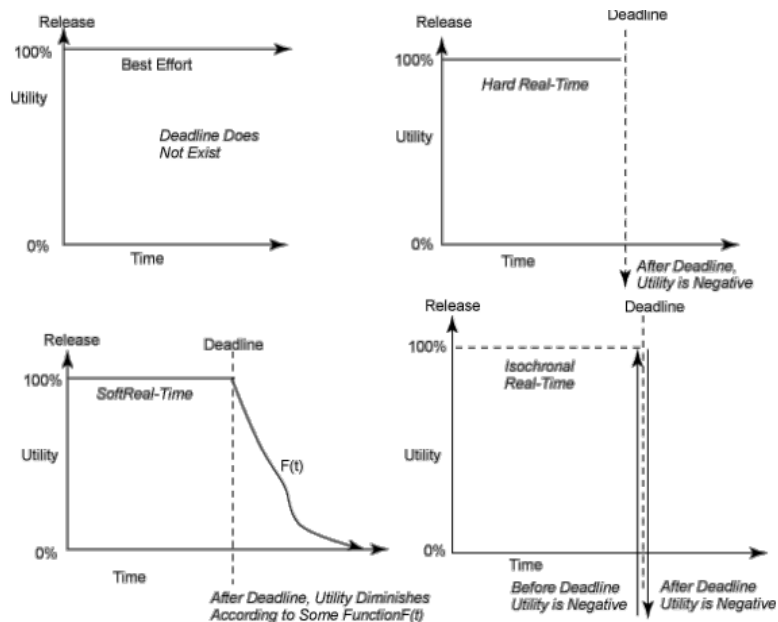
Ethernet-verkossa dataa siirretään pienissä paketeissa, joita kutsutaan kehyksiksi. Ethernet-kehyksiä on erilaisia, mutta käytetyin näistä on Ethernet II -kehys. Kuten kuvasta 17 nähdään, Ethernet II -kehys koostuu kahdeksasta eri alueesta. Ensimmäinen 64 bitin alue sisältää alustuksen ja erotinmerkin, jonka jälkeen viestin oikea sisältö alkaa. Uusissa Ethernet-systeemeissä viestintä on jatkuvaa eikä kehyksen alustukselle ole tarvetta. Kuitenkin yhteensopivuusongelmien välttämiseksi vanhempien Ethernet systeemien kanssa, alustus ja erotinmerkki kulkevat vielä kehyksen mukana. Alustuksen jälkeen tulee MAC-osoitteet (Media Access Control addresses). MAC-osoitteet kertovat minne viesti on menossa ja mistä se on tulossa. Length/type-osio kertoo mikä protokolla on kyseessä. Sen arvon ollessa alle 1523 se ilmaisee dataosion pituuden. Data-osio sisältää viestin datan, joka on minimissään 46 tavua ja maksimissaan 1500 tavua pitkä. 46 tavun minimipituus johtuu siitä, että viestin pitää kestää tarpeeksi pitkään, jotta kaikki verkon laitteet ehtivät saada viestin oikeassa aikarajassa. Lopuksi on CRC-osio (Cyclic Redundancy Checksum), joka päättää kehyksen. CRC on polynomi, joka muodostuu kehyksen sisällöstä. Sekä kehyksen lähettäjä, että vastaanottaja muodostavat CRC-luvun ja näitä vertaamalla saadaan selville, pääsikö viesti ehyenä perille. (Spurgeon C. 2000, s. 24-25)

64 bittiä	48 bittiä	16 bittiä	46-1500 tavua	32 bittiä
Alustus	Kohdeosoite	Lähdeosoite	Protokolla/Pituus	Tarkistus/Lopetus (CRC)

Kuva 15. Ethernet II -kehyyksen sisältö. (Spurgeon C. 2000, s. 24)

4.4 Reaaliaikainen systeemi (RTS)

RTS on tietokoneen ohjaama systeemi, joka on vuorovaikutuksessa siihen kuuluvien laitteiden kanssa. RTS:llä on tiukat ajalliset vaatimukset tämän vuorovaikutuksen kanssa. RTS-systeemeissä ajastuksen luotettavuus on tärkeämpää, kuin datan kulkeminen systeemin läpi. RTS-systeemejä on erilaisia riippuen niiden aikavaatimuksista. Tunis S. esittelee kolme erilaista RTS-luokkaa: soft-RTS, hard-RTS ja isochronal/firm-RTS (kuva 16). (Tunis S. 2009, s. 21-22)



Kuva 16. RTS- luokat. (Tunis S. 2009, s. 22)

Hard-RTS-systeemeissä viestien ajoituksessa ei saa olla yhtään virheitä. Systeemeissä, joissa vaaditaan hard-RTS:ää, viestin viiveet saattavat aiheuttaa peruuttamatonta ja suurta vahinkoa ympäristölle tai ihmiselle. Esimerkiksi teollisuusrobotin turvapiiri täytyy olla täysin viiveetön.

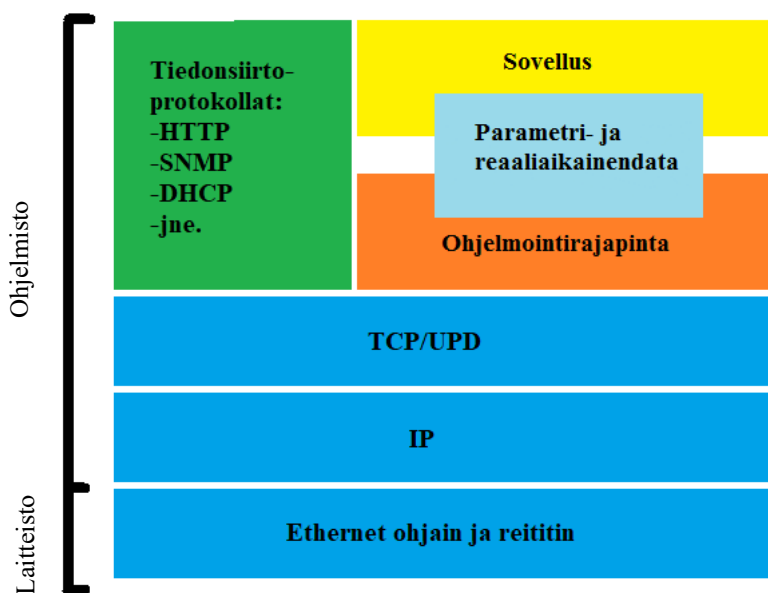
Soft-RTS-systeemeissä ei voida luottaa viiveettömän viestin saapumiseen, vaikka viestin viivästymiset eivät olekaan toivottuja. Soft-RTS sopii systeemeihin, joissa halutaan nopea vaste, mutta viivästymiset eivät aiheuta peruuttamatonta vahinkoa. Tällaisessa systeemeissä ajoituksen epäonnistuminen johtaa vain hitaampaan vasteeseen ja laskee järjestelmän laatua.

Isochronal/firm-RTS:ssä viestinsaapumisajalla on toleranssi, jonka sisällä viestin saapuminen saa olla. Yleensä isochronal-RTS toteutetaan siten, että aikainen vastaus menee puskuriin odottamaan oikeaa aikaikkunaa. (Tunis S. 2009, s. 22).

4.5 Reaaliaikaisten kenttäväylien luokat

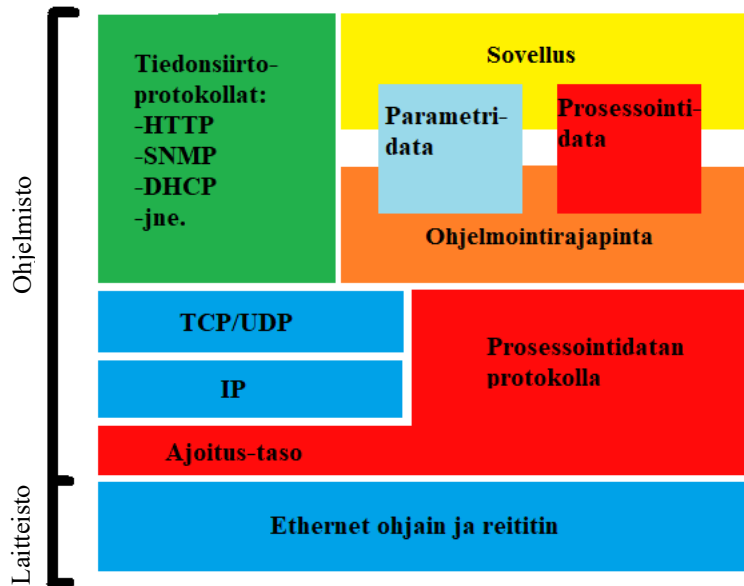
Rostan M. jakaa verkkodokumentissaan reaaliaikaiset kenttäväylät eli teollisuus-Ethernet teknologiat kolmeen luokkaan niiden reaaliaikaisuuden ja teknisten lähestymistapojensa perusteella. Esimerkit luokista B ja C ovat yksinkertaistettuja, sillä samaan luokkaan kuuluvilla kenttäväylillä on paljon eroja. Reaaliaikaisten kenttäväylien luokitukset antavat kuitenkin hyvän yleisymmärryksen miten erilaiset kenttäväylät on toteutettu.

Luokka A (kuva 17) käyttää standardia Ethernet-laitteistoa ja myös standardia TCP/IP-protokollaa (Transmission Control Protocol/Internet Protocol). Joissakin luokan A ratkaisussa on kuitenkin silti hieman muokattuja TCP/IP-protokollia suorituskyvyn parantamiseksi. Luokan A kenttäväylissä reaaliaikaisuus on rajoittunut reitittimien viiveiden, verkon muun ruuhkan ja erityisesti standardien protokollien takia.



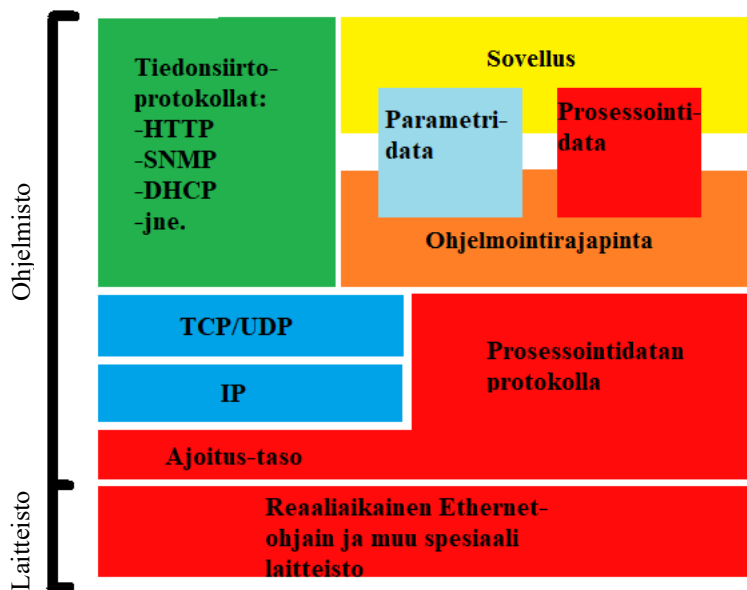
Kuva 17. Luokan A orjalaitteen tavanomainen rakenne. (Rostan M. 2009, s. 3)

Myös luokan B ratkaisussa käytetään standardia Ethernet-laitteistoa, mutta TCP/IP-protokollat on korvattu datan prosessointiprotokollalla (kuva 18). Datan prosessointiprotokollan avulla data saadaan suoraan Ethernet kehykseen. TCP/IP-protokollan korvaamisen ansiosta arvaamattomat viiveet saadaan poistettua. TCP/IP-tasot säilytetään yleensä kuitenkin rajapinnaksi ajallisesti vähemmän tärkeälle viestinnälle.



Kuva 18. Luokan B orjalaitteen tavanomainen rakenne. (Rostan M. 2009, s. 4)

Välttääkseen ennalta arvaamattomia viiveitä luokka C ei käytä lainkaan standardeja Ethernet komponentteja. Laitteisto on kokonaan vaihdettu käyttökohteeseen muokatulla laitteistolla (kuva 19).



Kuva 19. Luokan C orjalaitteen tavanomainen rakenne. (Rostan M. 2009, s. 5).

Taulukossa 2 on esitelty tunnetuimpia teollisuuden kenttäväyläratkaisuja sekä niiden luokka. Lisäksi taulukossa on väyläratkaisujen kehittäjät sekä väyläratkaisua käyttävä laitteistomittaja.

Taulukko 2. Reaaliaikaisten kenttäväylien luokat. (Tunis S. 2009, s.26)

Teollisuus-Ethernet teknologia	Organisaatio	Yhtiö	Luokka
ODVA	Pen DeviceNet Vendors Association	Rockwell Automation	A
Modbus TCP	Modbus-IDA	Schneider Electric	A
EtherCAT	EtherCAT Technology Group	Bechhoff Automation	C
SERCOS III	SERCOS international		C
Profibus/Profinet	PROFIBUS & PROFINET International (PI)	Siemens	A-C
CC-Link IE	CC-Link Partner Association (CLPA)	Mitsubishi Electric	C
Ethernet Powerlink	Ethernet Powerlink Standardization Group (EPSG)	B+R	B

4.6 Kenttäväylien standardit

Kenttäväylien kansainvälinen standardisointi on aina ollut haastavaa. Kenttäväylien innokkaimpina kehitysvuosina, alkaen 1985 luvulta, alkuperäinen tavoite on ollut vain yksi ja ainut kattava kansainvälinen kenttäväylä. Kenttäväylän standardoimisessa on kuitenkin ilmennyt ongelmia ja yhden ainoan kenttäväylän sijasta on standardisoitu useampi vaihtunut kenttäväylä. Standardin IEC 61 158 hyväksymisen myötä vuonna 2000 erimielisyydet kenttäväylistä saatiin sovittua. Vuonna 2008 standardi päivittyi IEC 61784 versioon, jossa kenttäväylät jaettiin CPF:iin eli kommunikointiprofiiliperheisiin (*Communication Profile Families*). Seuraavassa listauksessa on osa standardiin kuuluvista kenttäväylistä, joista esitellään automaatiassa paljon käytetyt EtherCAT ja Profibus/Profinet. (Felser M. 2005, s. 1123)

IEC 61784 Standardiin kuuluvia kenttäväyliä:

- CPF-2: ControlNet (Ethernet/IP),

-CPF-3: Profibus/Profinet,

-CPF-4: P-Net,

-CPF-12: EtherCAT,

-CPF-13: Ethernet Powerlink,

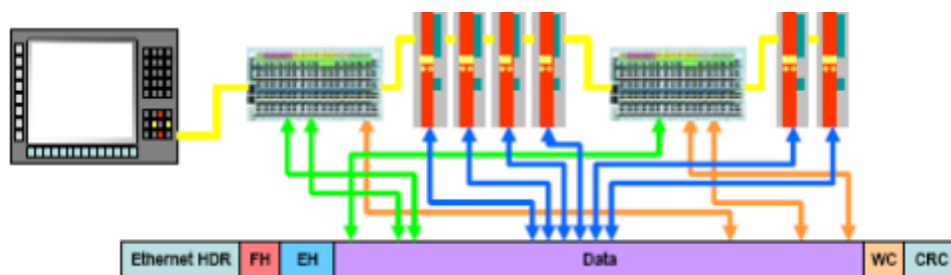
-CPF-15: Modbus-RTPS,

-CPF-16: Sercos.

4.6.1 EtherCAT

EtherCAT-kenttäväylä on Beckhoffin kehittämä ja sen kehityksessä on keskitytty viiteen eri ominaisuuteen. Ensimmäinen ominaisuus on, että kaikki Ethernet-portilliset ohjaimet voisivat olla EtherCAT-isäntiä ilman erityisliitäntöjä. Toisena kehityksen lähtökohtana on ollut se, että EtherCAT voisi toimia kaikkien Ethernet-laitteiden ja -protokollien kanssa. Tällainen joustavuus mahdollistaisi EtherCAT-laitteiden integroimisen ilman suuria muutoksia valmiiseen systeemiin, joka käyttää eri väyläratkaisua. Kolmantena lähtökohtana on ollut se, että kaiken tyyppiset I/O:t voisivat toimia orjalaitteina. Lisäksi EtherCAT-datan siirtonopeus tulee olla niin nopea kuin Ethernet-kaapelin kaistanleveys sallii. Viimeisenä kehityksen lähtökohtana on ollut alle sadan mikrosekunnin sykli aika, mikä mahdollistaa hyvän ja tarkan servo-ohjauksen. (Jansen D. & Buttner H. 2004, s. 17).

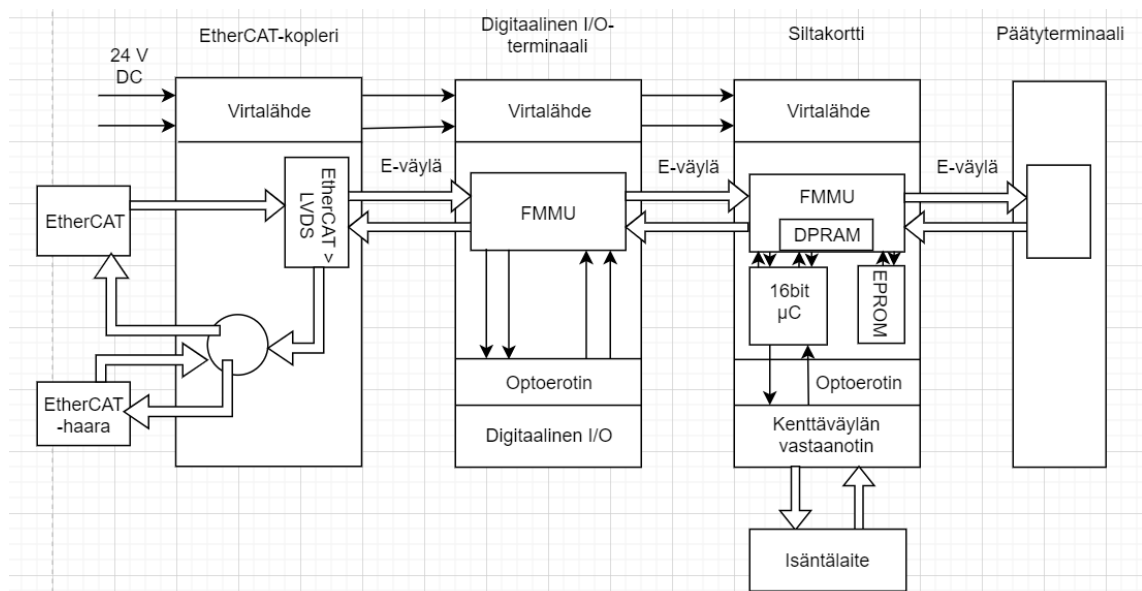
EtherCAT-väylä on yksinkertaistettuna yksi iso Ethernet-laite, joka lähettää ja vastaanottaa viestejä. Tämä Ethernet-laite sisältää suuren määrän EtherCAT-orjia, jotka prosessoivat tulevat viestit suoraan ja ottavat viestin vastaan tai lähettävät sen eteenpäin seuraavalle EtherCAT-orjalle. Viimeinen orja lähettää kokonaan prosessoidun viestin takaisin isännälle vastauksena saapuneeseen viestiin. Yksi lähetetty viesti eli Ethernet-kehys sisältää viestin kaikille väylässä oleville laitteille (kuva 20). Yhdessä viestissä voi olla useampi komento, joissa jokaisella on oma laitekohtainen osoitteensa. (Jansen D. & Buttner H. 2004, s. 17; Rostan M. 2009, s. 132)



Kuva 20. EtherCAT viestin kulku. (Rostan M. 2009, s. 132).

Kuvassa 21 on Beckhoffin tavallinen PLC:hen liitettävä EtherCAT-kopleri, jonka perään on kytketty I/O-terminaali, siltakortti ja päätyterminaali. EtherCAT-koplereita voidaan laittaa useita sarjaan liittämällä ne Ethernetilla toisiinsa. EtherCAT-koplerin logiikka osaa ohjata signaalin tarvittaessa takaisin ohjaavalle PLC:lle tai lähettää viestin eteenpäin seuraavalle koplerille. EtherCAT-koplerien välinen viestintä tapahtuu EtherCAT-signaaleilla. Kopleri taas välittää viestin sen perässä oleville terminaaleille E-väylän (E-bus) kautta.

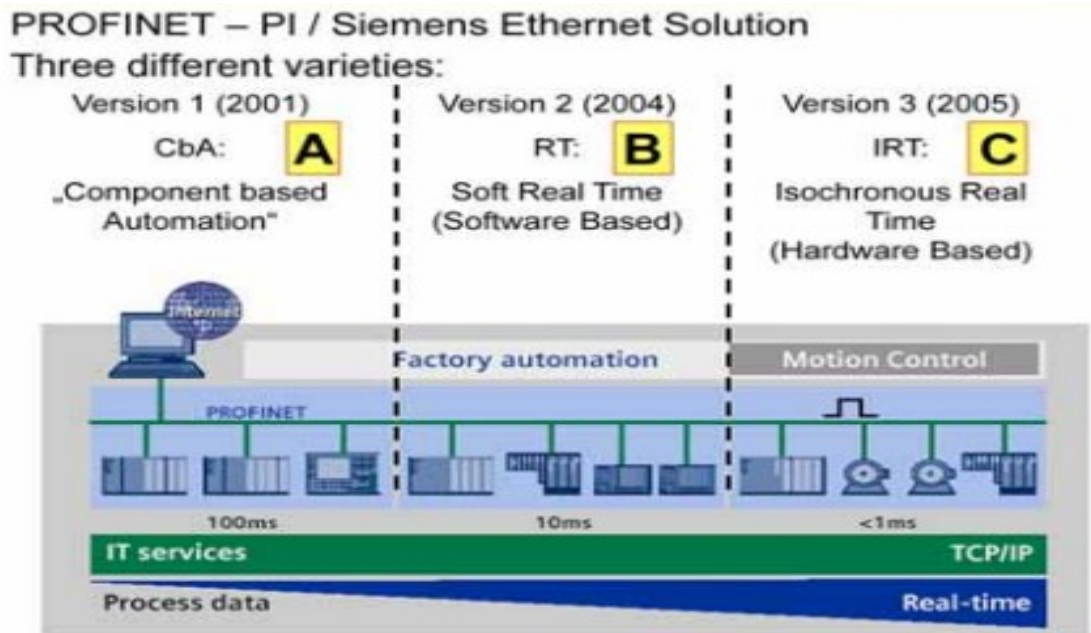
Kuvan 21 esimerkissä EtherCAT-signaali saapuu EtherCAT-koplerille, jossa Ethernet signaali muutetaan LVDS-signaaliksi E-väylää varten. I/O -terminaaliin saapuvat signaalit menevät FMMU-kortille (Fieldbus Memory Management Unit). FMMU-kortti tunnistaa osoitteen avulla mikä osa saapuvasta viestistä on tarkoitettu kyseiselle terminaalille. Kuva 21 on hyvä esimerkki EtherCAT-laitteiden liittämisestä toiseen eri kenttäväylää käyttävään systeemiin. Toinen terminaali on siltakortti, joka mahdollistaa EtherCAT-laitteiden yhdistämisen toisen isäntälaitteen kanssa. FMMU on tässä tapauksessa kytketty RAM:in kautta mikrokontrolleriin, joka suorittaa protokollan vaihtamisen toiseen kenttäväylään. Lopussa on lopetusterminaali, joka lähettää saapuvan signaalin takaisin. (Jansen D. & Buttner H. 2004, s. 18)



Kuva 21. EtherCAT:n toiminta. (Beckhoff, 2019)

4.6.2 PROFINET

Profinet on teollisuuden Ethernet-teknologia, joka tarjoaa useita kommunikointiprofiileja, riippuen käyttökohteen reaaliaika-vaatimuksista. Sen on kehittänyt Siemens ja sitä hallitsee Profibus & Profinet International. Profinetistä on kaksi pääversiota: Profinet CbA ja Profinet I/O (kuva 22).



Kuva 22. Profinet-versiot (Nykyään RT ja IRT kuuluvat Profinet I/O:n alle). (Rostan M. 2009, s. 5)

Profinet CbA (Component basen Automation) on luokan A väyläratkaisu ja se julkaistiin vuonna 2001. Profinet CbA on kenttäväylä eri ohjaimien välille. Lisäksi sen tarkoitus on mahdollistaa Profibus-väylän yhdistäminen standardiin Ethernet-väylään. Profinet CbA:an sisältyy komponenttipohjainen ohjelmointiympäristö, jossa muuttujat voidaan yhdistää graafisesti kommunikointilinkkeihin. (Tunis S. 2009, s. 41)

Profinet I/O sisältää kolme erilaista vaihtoehtoa reaaliaikaiseen kommunikointiin. Kategoria yksi on luokan B lähestymistapa, jota kutsutaan myös Profinet RT:ksi. Kategoriat kaksi ja kolme ovat taas luokan C lähestymistapoja. Yhdessä ne muodostavat Profinet IRT:n. **Kummassakin kategorioissa sykli aika on jaettu** kahteen osaan, toinen reaaliaikaiselle kommunikoinnille ja toinen muulle viestinnälle. Kategoriasa 3 kommunikoinnin

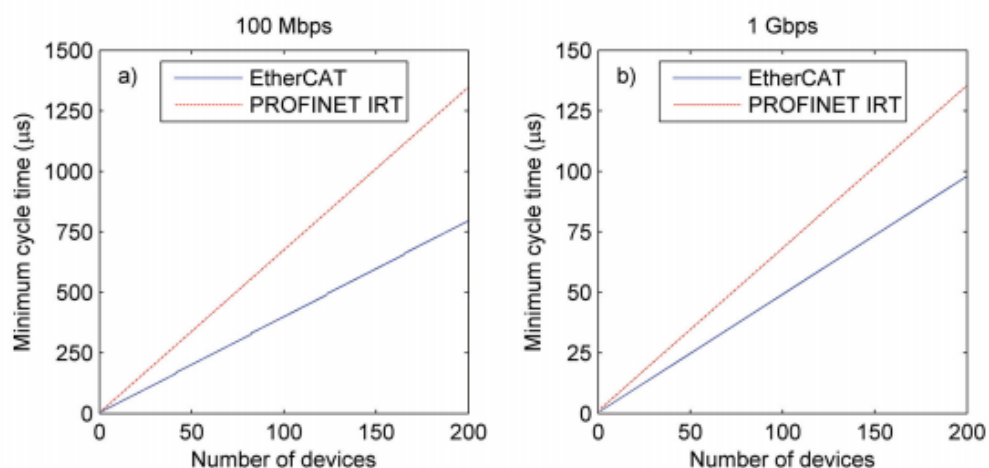
järjestys ja reitit ovat tarkasti määriteltyjä ja optimoituja käyttökohteen systeemiä suunniteltaessa. Tämä antaa sille paremman suorituskyvyn, kuin kategoria kaksi, mutta toisaalta sen muokkaaminen vaatii uuden systeemin suunnittelua. (Tunis S. 2009, s. 42)

4.7 Profinet IRT:n ja EtherCAT:in vertailu

Kenttäväylien vertailussa voidaan tarkastella esimerkiksi reaaliaikaisuutta, laitteiston hintaa ja monipuolisuutta. Koska kenttäväyliä on todella paljon ja vertailtavia aiheita vähintäänkin yhtä paljon, tässä työssä vertailu kohdistetaan EtherCAT:in ja Profinet IRT:n välille.

Gunnar P.:n konferenssijulkaisussa on vertailtu Profinetin IRT:n ja EtherCAT:in reaaliaikaista suorituskykyä (Kuva 23). Julkaisussa esitelyjen tuloksien perusteella EtherCAT suoriutuu nopeammin reaaliaikaisuuden testeistä, kuin profinet IRT. Syyksi julkaisussa esitetään EtherCAT:in protokollaa, jossa yksi viesti sisältää käskyjä usealle orjalaitteelle, kun taas Profinet IRT lähettää eri viestin joka orjalle. Testien tuloksista nähdään, että kun orjalaitteiden määrä lisääntyy, Profinet IRT:n suoritus aika on hitaampi kuin EtherCAT:in.

EtherCAT:in ja Profinet IRT:n reaaliaikaisuuden eroilla ei ole käytännön merkitystä suurimmassa osassa teollisuuden sovelluksissa. EtherCAT ja Profinet voidaan kuitenkin yhdistää samaan automaatiojärjestelmään erilaisten siltakorttien avulla.



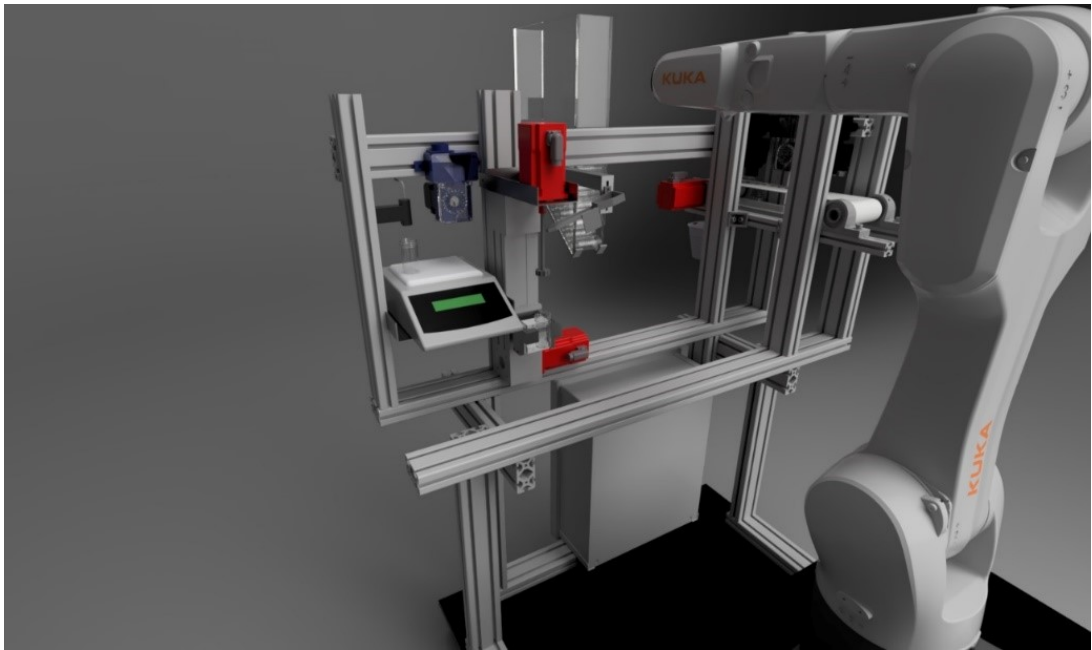
Kuva 23. Profinet IRT:n ja EtherCAT:in syklijajan vertailu. (Gunnar P., 2008, s. 413)

Kenttäväylien valinnassa ei ole pelkästään kyse siitä, kuinka reaaliaikainen se on, vaan kuinka se vastaa sovelluksen tarpeisiin. Esimerkiksi Modbus TCP voi toimia halpana ja

hyvänä ratkaisuna, vaikka sillä ei ole hyvää reaaliaikaisuutta, jos tätä ominaisuutta ei tarvita. Modbus TCP ei tarvitse paljoa laskentatehoa ja se käyttää tavallisia reitittämiä (Tuis S. 2009, s.55). Profinet taas tarjoaa joustavan teknologian, joka sopii joka tarpeeseen.

5 AUTOMAATION JA MEKANIIKAN SUUNNITTELU

Tässä kappaleessa käydään läpi kattavasti geopolymeerejä valmistavan robottisolun vaaditut ominaisuudet, mekaaniset osat, toimilaitteet sekä miten automaatio on suunniteltu ja toteutettu. Robottisolulla tiedetään olevan paljon toiminnallisia vaatimuksia ja tarkoituksena on kehittää solua eteenpäin myös diplomityön jälkeen. Tämän takia kaikista suunnitelluista ratkaisuista on pyritty saamaan mahdollisimman monikäyttöisiä ja yksinkertaisia. Kuvassa 24 on robottisolun 3D-malli.



Kuva 24. Robottisolun suunniteltu 3D-malli.

5.1 Robottisolun vaatimusmäärittelyt

Diplomityön osalta vaatimuksena on rakentaa ja suunnitella liikuteltava robottisolu, joka valmistaa yhden geopolymeeri-koesarjan yhtä hyvin tai jopa paremmin, kuin ihminen sen pystyy tekemään. Valmistettavan koesarjan resepti annetaan Excel-taulukon muodossa, josta robottisolu ottaa ohjeet sekoitusten tekemiseen. Geopolymeerin valmistukseen kuluva aikaa ei ole määritetty. Robottisolun mekaaninen rakenne tulee olla joustava siten, että siihen voidaan lisätä uusia ominaisuuksia ja parannuksia jatkoa varten.

Yhden geopolymeerisarjan valmistukseen tarvitaan kahta erilaista jauhoa ja nestemäistä aktivaattoria. Näiden ainesosien syöttämistarkkuudelle ei ole määritelty rajoja, mutta syötettyjen ainesosien määrien tulee tulla ilmi tuloksissa. Itse sekoituksessa jauho tulee syötää tasaisesti sekoituksen aikana ja sekoittimen tulee sekoittaa seosta kauttaaltaan.

5.2 Geopolymeerien automatisoidun valmistuksen suunnitellut vaiheet

Geopolymeerin valmistukseen käytettävä jauhe ja neste syötetään erillisiin purkkeihin. Robotti hakee purkkisäiliöstä purkin puntarille, jossa nesteen pumppaus purkkiin tapahtuu. Puntari on kytketty IPC:hen USB-liitännällä, josta python-ohjelma lukee puntarin dataa ja lähettää ADS:llä painolukeman TwinCatille. Tämän jälkeen nestepurkki viedään lineaarikuljettimelle sekoitukseen, jossa purkki asetetaan pneumatiikkasyylinterillä toimivaan puristimeen. Purkin on liikuttava sekoittimen suuntaisesti siten, että sekoitus tapahtuu tasaisesti sekä purkin pohjalta että pinnalta. Liike toteutetaan lineaariyksiköllä, jota liikutetaan servomootorilla. Nestepurkki jää odottamaan jauhoa sekoittimelle.

Seuraavaksi robotti hakee toisen purkin puntarille jauhosyöttimen alle ja jauhon määrän rajoitus tapahtuu samalla periaatteella kuin nesteen syötössä. Kun jauhoa on purkissa tavoitettua, robotti vie ja kaataa jauhon syöttörauniin. Jauho valuu sekoituksen aikana tasaisesti rännistä sekoitettavaan nesteeseen tärhistimen avulla.

Kun geopolymeerisekoitus on valmis, robotti hakee geopolymeerisekoituksen tarttujasta ja vie sen telineeseen odottamaan seuraavaa geopolymeeriä. Jos sekoitettavaa tuotetta vaihdetaan tai valmistuksessa pidetään tauko, sekoitin puhdistetaan sekoittamalla vesipurkkia.

5.3 Mekaanisten osien suunnittelu

Diplomityö sisältää useiden erilaisten mekaanisten osien suunnittelua ja valmistusta, sekä erilaisten toiminnallisten ratkaisujen kehittämistä. Osien valmistuksessa ja suunnittelussa on pyritty halpoihin ratkaisuihin, monipuoliseen käyttöön ja toistettavuuteen. Kaikki 3D-mallit on tehty Autodesk Fusion 360 -ohjelmistolla ja osien valmistus on toteutettu 3D-tulostamalla ja laserleikkaamalla. Haastavammat ja kestävyyttä vaativat osat on valmistettu Oulun yliopiston työpajalla koneistamalla.

5.3.1 Robottisolun pohja

Diplomityön valmisteluissa on suunniteltu robotille ja muulle laitteistolle tukeva ja liikuttava pohja. Suunnittelussa käytetään apuna Turun KUKA-koulutuskeskuksessa käytettyä robotin alustaa, joka on suunniteltu KR10:n alustaksi. Valmiin mallin perusteella painoksi on pyritty saamaan noin 400 kiloa. Mitoiltaan alusta halutaan EUR-lavan kokoiseksi ja liikealue halutaan maksimoida pöytäkorkeudella. Pöytäkorkeudeksi mitattiin noin 800 mm ja KR10:n manuaalista löytyvän kuvan 25 mukaan KR10:n paras liikealue on 400 mm korkeudessa. Tästä on laskettu alustan korkeudeksi noin 400 mm. Alustan lopullinen korkeus on kuitenkin 440 mm, koska KR10:n ohjain ei muuten mahdu korkeen alle piiloon. Alustaan on lisäksi lisätty reikiä alumiiniprofiilikehikkoa varten. KR10:n alustaan on lisätty myös mahdollisuus asentaa renkaat.

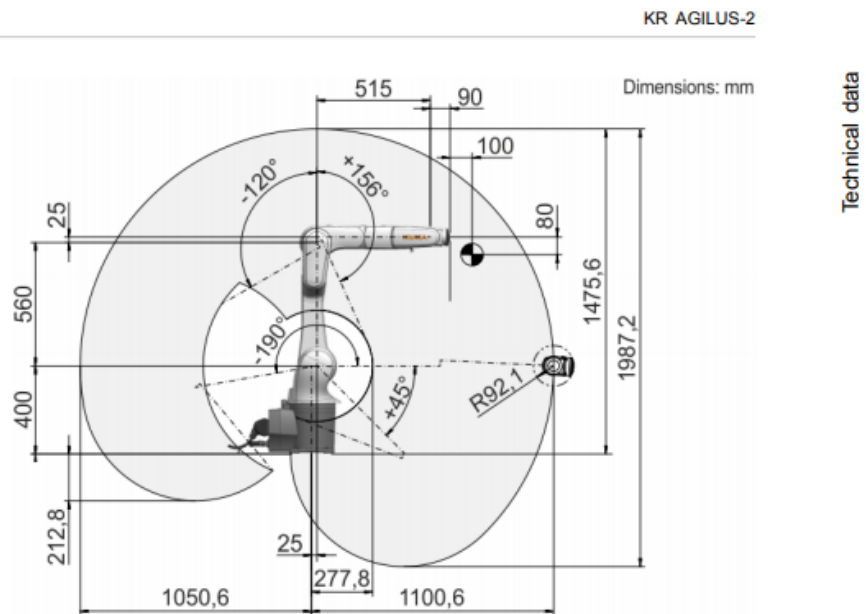


Fig. 4-29: KR 10 R1100-2, working envelope, side view

Kuva 25. KUKA KR10 liikealue (KUKA, 2019).

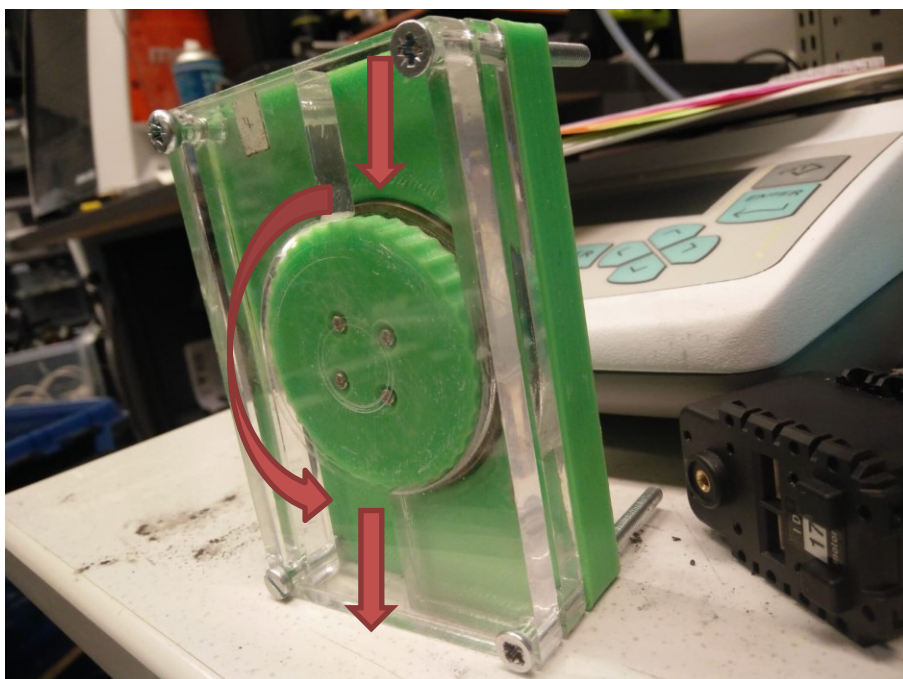
KR10:n alustan päärakenteet on valmistettu vesileikkurilla 20 mm teräslevystä. Korokkeen suojalevyt edessä ja takana ovat 3 mm teräslevyä ja ne ovat irrotettavia. Kaikki 20 mm teräslevyt on hitsattu yhteen. Pinnoitteena on käytetty mustaa, iskunkestävää ja kiiltävää maalia. KR10:n alusta on valmistettu kokonaisuudessaan Oulun yliopistolla. Alustan lopulliset mitat ovat 1000x1200x440 mm. Kuvassa 26 robotti on asennettuna valmistetun alustan päälle.



Kuva 26. Robottisolun alusta.

5.3.2 Jauhon syöttölaite

Jauhon syöttölaitteen suunnittelu on aloitettu alustavasta syöttölaitteen mallista, jonka kesäharjoittelija Teemu Laurila on tehnyt (kuva 27). Alustavan mallin lisäksi syöttölaitteelle on etsitty muita vaihtoehtoja teollisuudessa ja laboratorioissa käytetyistä ratkaisuista. Eri-laisista ratkaisuista on tehty hintavertailu ja kaikki järkevät ratkaisut ovat osoittautuneet huomattavan kalliiksi. Tästä syystä parhaaksi vaihtoehdoksi on valikoitunut Teemu Lau- rilan suunnitteleman mallin kehittäminen eteenpäin. Muovinen jauhosyötin on testauk- sien perusteella toimiva, halpa, helposti muokattava ja nopeasti kopioitava.

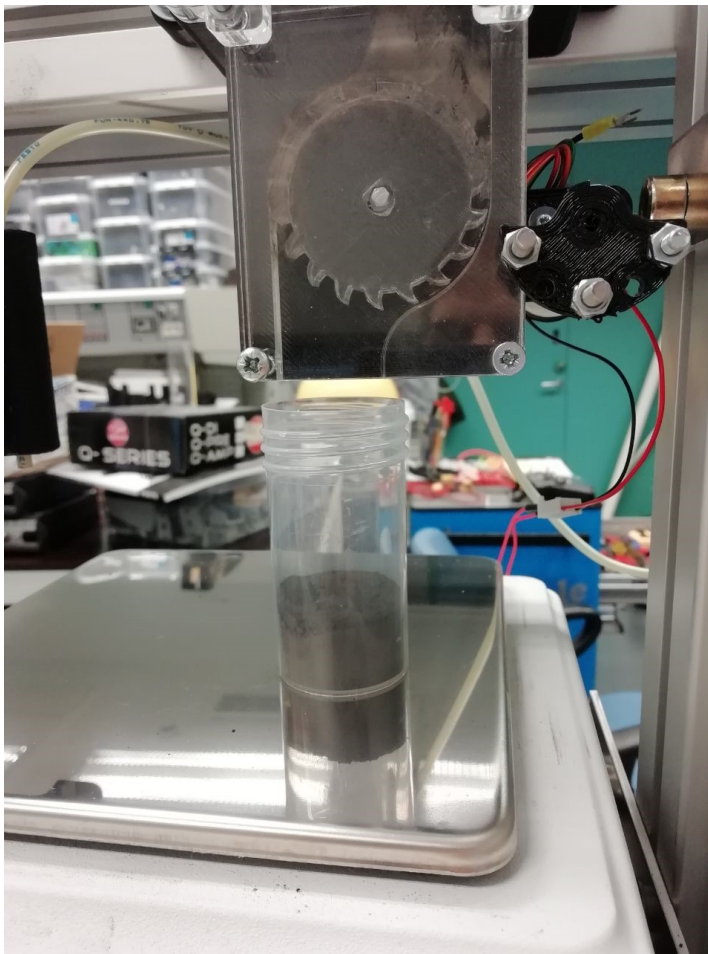


Kuva 27. Syöttölaitteen ensimmäinen versio

Kesäharjoittelijan tekemän syöttölaitteen mallissa toimintaperiaate perustuu pyörivään rattaaseen, jonka koloihin jauho tippuu laitteen yläpuolella olevasta aukosta. Rataan pyöriessä jauhe tippuu alas rataan kolojen kokoisina annoksina. Diplomityössä käytetyssä syöttölaitteessa käytetään samaa toimintaperiaatetta ja rakennetta.

Syöttölaitteen ensimmäisen version suurimpia näkyviä ongelmia ovat jauheen tarttuminen hammasrataan koloihin, jolloin jauheen syöttö loppuu nopeasti. Jauheen tarttuminen koloihin johtuu siitä, että jauho pakkautuu helposti paakuiksi, 3D-tulostettu pinta on epätasainen sekä rataan kolot ovat liian pieniä.

Ensimmäisen ja viimeisen mallin välissä on ollut useita erilaisia kokeiluja. Lopulta on päädytty kuvassa 28 olevaan malliin. Suurimmat muutokset syöttölaitteessa ovat hammasrataan materiaalin vaihtuminen PLA-muovista akryyliin, kolojen kasvattaminen, hampaiden muodon muuttaminen ja DC-moottorilla toimivan täristimen lisääminen.

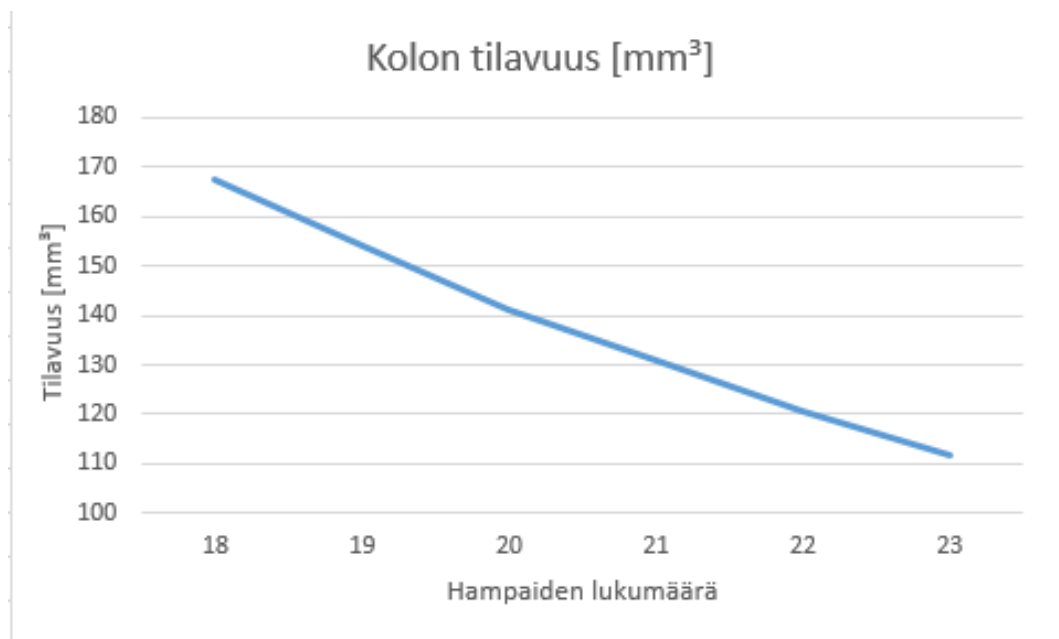


Kuva 28. Syöttölaitteen lopullinen versio (20 hampaan ratas).

Syöttölaitteen hammasratas voidaan vaihtaa tarpeen mukaan. Syöttölaitteen hammasrataan mitoituksella voidaan vaikuttaa siihen, millä tarkkuudella jauhoa voidaan syöttää ja

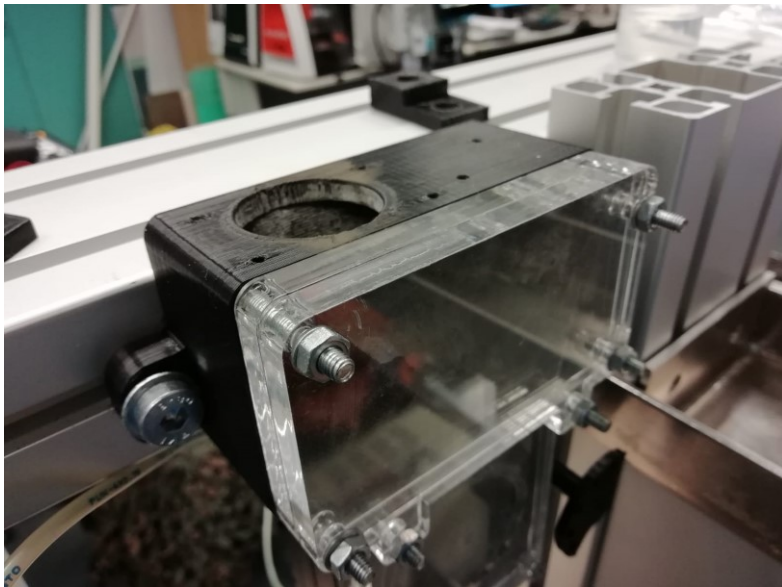
voiko syötettävä materiaali olla hienojakoista jauhoa vai karkeaa rietta. Hienojakoisen jauhon syöttöön soveltuu paremmin ratas, joka kaivaa jauhoa (kuva 30) ja rakeiden syöttöön soveltuu paremmin tavallista hammasratasta muistuttava ratas.

Syöttölaitteen tarkkuutta voidaan muokata muuttamalla hampaiden lukumäärää. Kuvan 29 kuvaajasta nähdään, miten rattaan kolojen tilavuus muuttuu hampaiden lukumäärän suhteen. Hampaan tilavuudesta voidaan laskea arvio yhden annoksen massasta, jos tiedetään syötettävän jauhon tiheys. Testeissä on kuitenkin huomattu, että 23 hammasta on maksimi määrä, sillä jauho tarttuu ahtaisiin koloihin helposti. Testeissä käytettiin jauhetta masuunikuonaa.



Kuva 29. Kolon tilavuus suhteessa hampaiden lukumäärään.

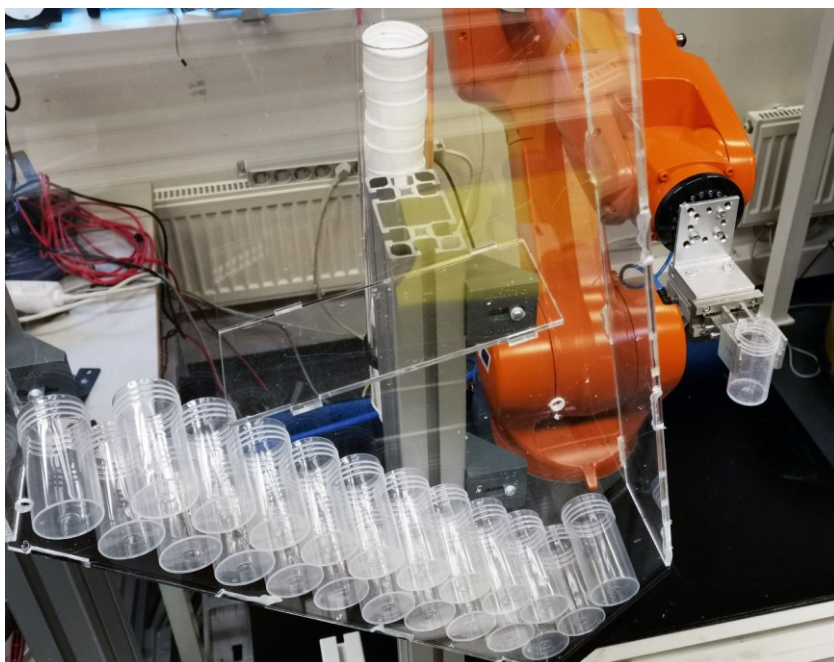
Myös syöttölaitteen jauhosäiliöstä suunniteltiin useita versioita, joista lopulta päädyttiin kuvan 30 mukaiseen ratkaisuun. Jauhosäiliön päälle on tehty irrotettava kierresovite, johon voi laittaa esimerkiksi kahden litran limsapullon. Tarvittaessa kierresovite voidaan vaihtaa isommalle astialle sopivaksi. Testeissä huomattiin, että jos jauhon paino tulee suoraa rattaalle, jauho pakkautuu helpommin ennen ratasta olevan syöttöaukon suulle. Tästä syystä pullon suuaukko on asetettu säiliön reunalle.



Kuva 30. Syöttölaitteen säiliö.

5.3.3 Sekoituspurkkien paikoitus

Sekoituspurkkien paikoitukseen on suunniteltu kuvan 31 mukainen säiliö, johon mahtuu noin sata purkkia. Jatkoa ajatellen säiliön sivussa on paikat kiinnitystapeille, joiden avulla säiliötä voidaan laittaa useita rinnan. Säiliö valmistettiin 3D-tulostimella ja laserleikkamalla akryyliä. Osat valmistettiin Oulun yliopiston Fablabissa ja BISG:n laboratoriossa. Testauksissa ei tullut yhtäkään jumittumista, kun robotin liike oli ohjelmoitu sopivaksi. Kuitenkin varmuuden vuoksi säiliöön tulee asentaa anturi, jotta voidaan huomata purkin mahdollinen puuttuminen hakupaikalta. Robotti kohdistaa purkit vielä erikseen säiliön jälkeen.

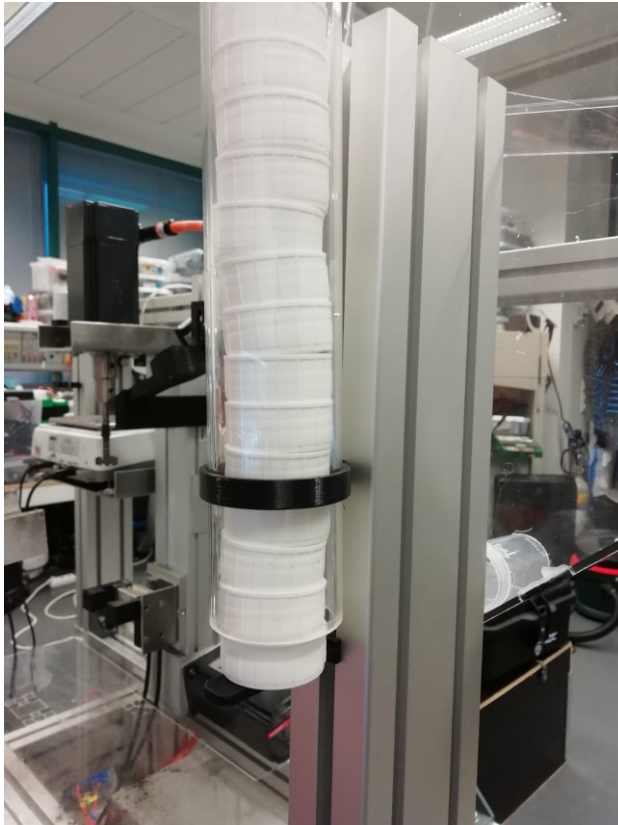


Kuva 31. Sekoituspurkkien säiliö.

Purkkien paikoituksen jatkokehityksenä on mietitty konenäön käyttöä. Diplomityön aikana on selvitetty myös, onnistuuko kirkkaiden ja läpinäkyvien purkkien asennon tunnistaminen konenäöllä. Koe suoritettiin SICK:n toimesta trispectorP1000 3D-kameralla. Koeksessa käytettiin kirkasta läpinäkyvää pulloa. Tulokset näyttivät siltä, että konenäkö kyseisellä kameralla saattaisi toimia, mutta konenäköä ei haluttu lisätä robottisoluun ainakaan tässä vaiheessa.

5.3.4 Korkkien paikoitus

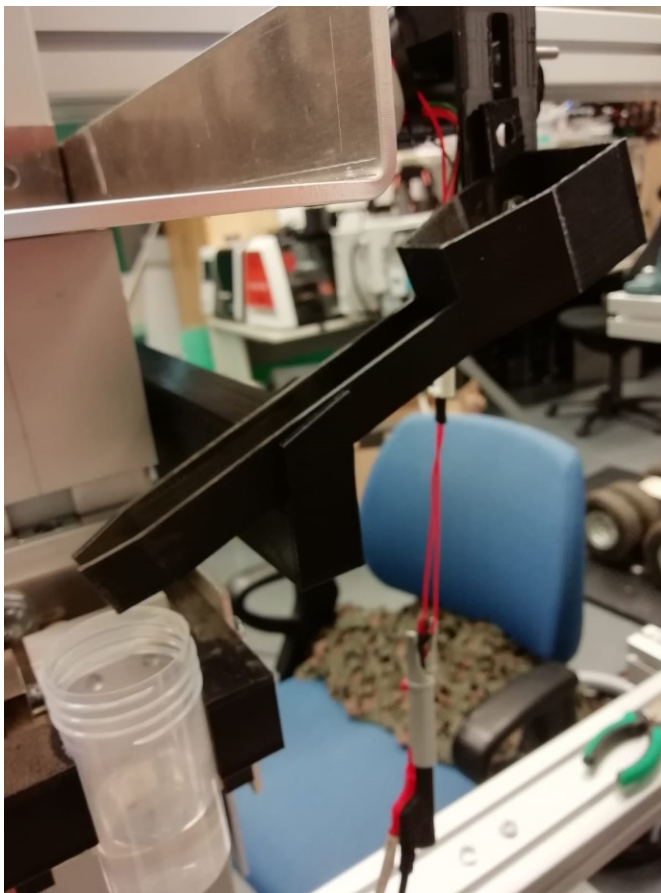
Korkkien paikoitukseen käytetään kirkasta muoviputkea (kuva 32). Testeissä putki on toiminut todella hyvin, vaikka sen sisähalkaisija on ollut hieman liian iso. Tästä johtuvan virheen takia robotti ei ole aina saanut korkkia kiristettyä riittävästi sekoituksen valmistuttua.



Kuva 32. Korkkien paikoitusputki.

5.3.5 Ränni jauhonsyötössä

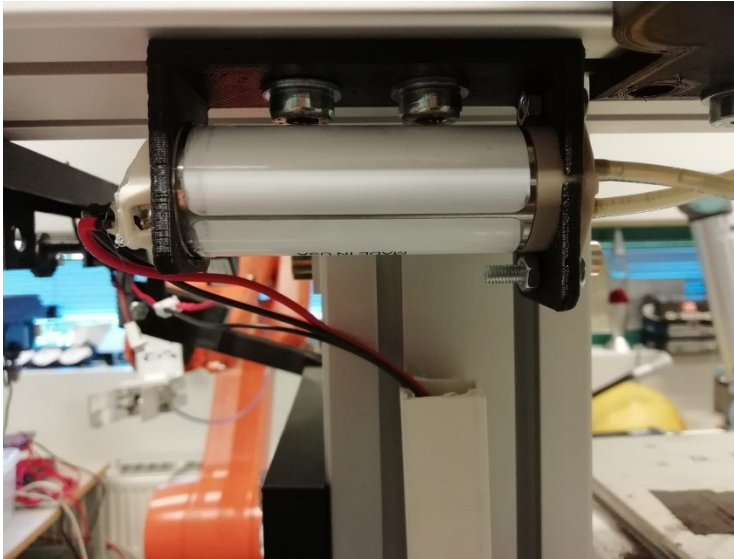
Sekoitusta tehdessä jauho syötetään tasaisesti sekoitettavan nesteen sekaan. Tähän on suunniteltu ränni (kuva 33), joka on valmistettu 3D-tulostamalla. Rännin varteen kiinnitetään DC-moottori, jossa pyörii epäkeskeinen paino ja se toimii rännin tärhistimenä. Tärinän amplitudia ja taajuutta voidaan muokata muuttamalla epäkeskeisen painon massaa tai moottorin pyörimisnopeutta. Jauhon syöttämisen sulavuus riippuu useista tekijöistä, kuten rännin tukipisteistä ja tärinän amplitudista sekä taajuudesta. Rännin pää tulee tukea, jotta jauho tippuu purkkiin sujuvasti. Tärinän amplitudilla ja rännin kaltevuudella voidaan säätää jauhon syöttönopeutta. Sopivilla asetuksilla syöttö toimii todella luotettavasti.



Kuva 33. Syöttöränni.

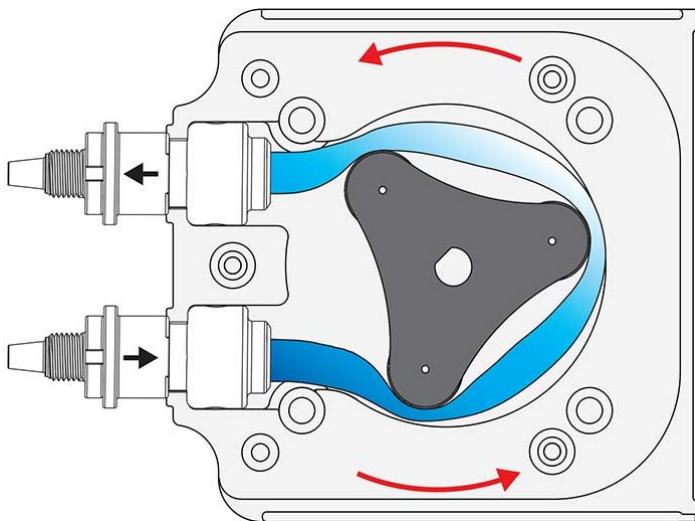
5.3.6 Nestepumppu

Robottisolun nestepumppuna on toiminut The Lee Companyn LPG-sarjan pumppu (kuva 34), jonka yhden pumppauksen tilavuus on 175 μl . Pumppu pumppaa maksimissaan viiden hertsin taajuudella.



Kuva 34. The Lee Companyn LPG-sarjan pumppu.

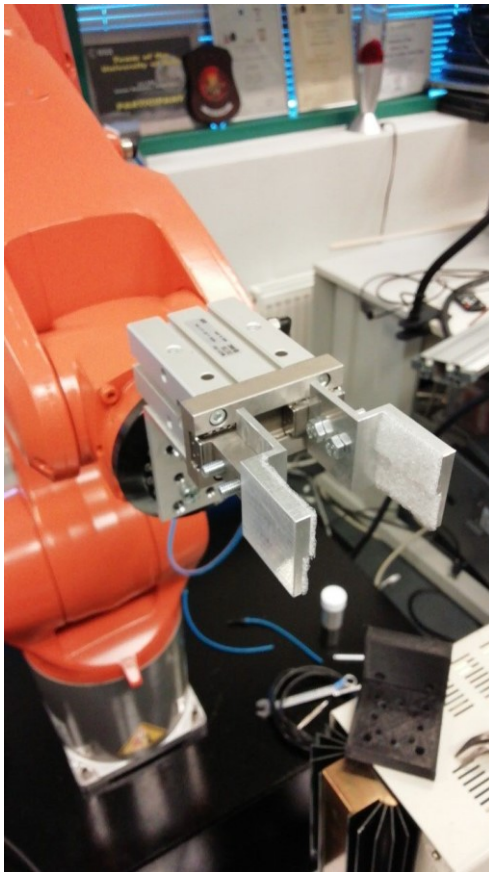
Kokeissa huomattiin, että pumppu ei toimi kunnolla vettä viskoosisemmilla nesteillä, eikä siksi soveltunut kaikille aktivaattoreille. Jatkokehityksessä robottisoluun otetaan käyttöön *peristaltic pump* -mallinen pumppu, joilla myös viskoosisten nesteiden pumppaus on mahdollista. Peristalttinen pumpun toiminta on kuvattu kuvassa 35.



Kuva 35. Peristalttisen pumpun toiminta (Pump & Systems, 2019).

5.3.7 Tarttujat

Robotin tarttujat on valmistettu alumiinista Oulun yliopiston pajalla. Tartuntakappaleet on suunniteltu siten, että niillä voidaan tarttua tiukasti sekä purkista että korkista, puristamatta kumpaakaan liikaa. Puristava toimilaite on SMC:n mallin MHZ2 paineilmalla toimiva tarttuja. Kuvan 36 kuvassa on tarttuja kokonaisuudessaan.



Kuva 36. Robotin tarttuja.

Lineaariyksikön tarttujan (kuva 37) tartuntakappaleet on valmistettu 3D-tulostimella muovista. Lineaariyksikön tarttuja on SMC:n MGQM 16-10 -mallinen kaksitoiminen pneumaattikasyylinteri. Viiden baarin paineella sylinterin puristusvoima on noin sata newtonia, joka varmistaa purkin paikallaan pysymisen sekoituksen ja korkin kiristyksen aikana.



Kuva 37. Lineaariyksikön tarttuja.

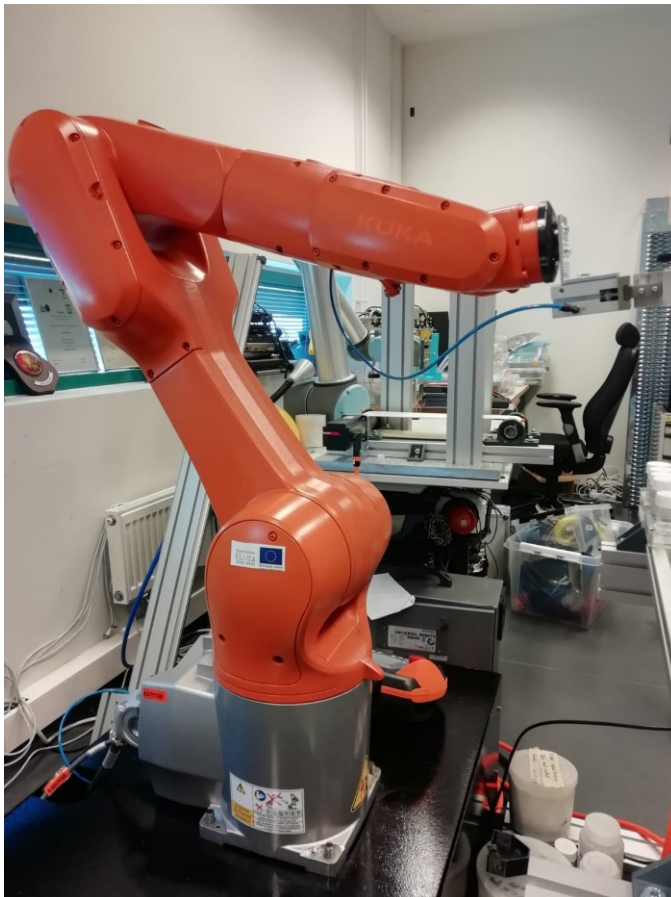
5.4 Toimilaitteiden ja IO:n valinta

Toimilaitteiden valinnassa on mietitty robottisolun lisäksi niiden mahdollista käyttöä muissa projekteissa ja jatkokehityksessä. Tämän takia esimerkiksi moottorit ja servo-ohjaimet on ylimitoitettu vaativampien käyttökohteiden varalle. Osa valinnoista on tehty sen perusteella mitä yliopistolta on löytynyt valmiiksi, sillä näin on säästetty kuluissa.

5.4.1 Manipulaattori

Manipulaattorilta vaaditaan robottisolussa 1.5 m ulottuvuutta ja 0.1 mm toistotarkkuutta. Manipulaattori valittiin kahdesta vaihtoehdosta, jotka löytyivät valmiiksi Oulun yliopistolta. Toinen vaihtoehto oli Universal Robot:in UR10 ja toinen KUKA:n KR10. Kumpikin soveltui käyttökohteeseen hyvin.

Robottisolun manipulaattoriksi valikoitui KUKA KR10 R1100-2 käsivarsirobotti (kuva 38), jonka ohjaimena toimii KR C4 compact. Robotilla on kuusi vapausastetta ja se painaa 54 kiloa. Sen nimellinen maksimikuorma on kymmenen kiloa ja sen maksiminopeus on 2.5 m/s. Robotin ohjelmointiin käytetään WorkVisual 5.0- ohjelmistoa sekä ohjelmointilaitetta.



Kuva 38. KUKA KR10.

5.4.2 Lineaariyksikkö

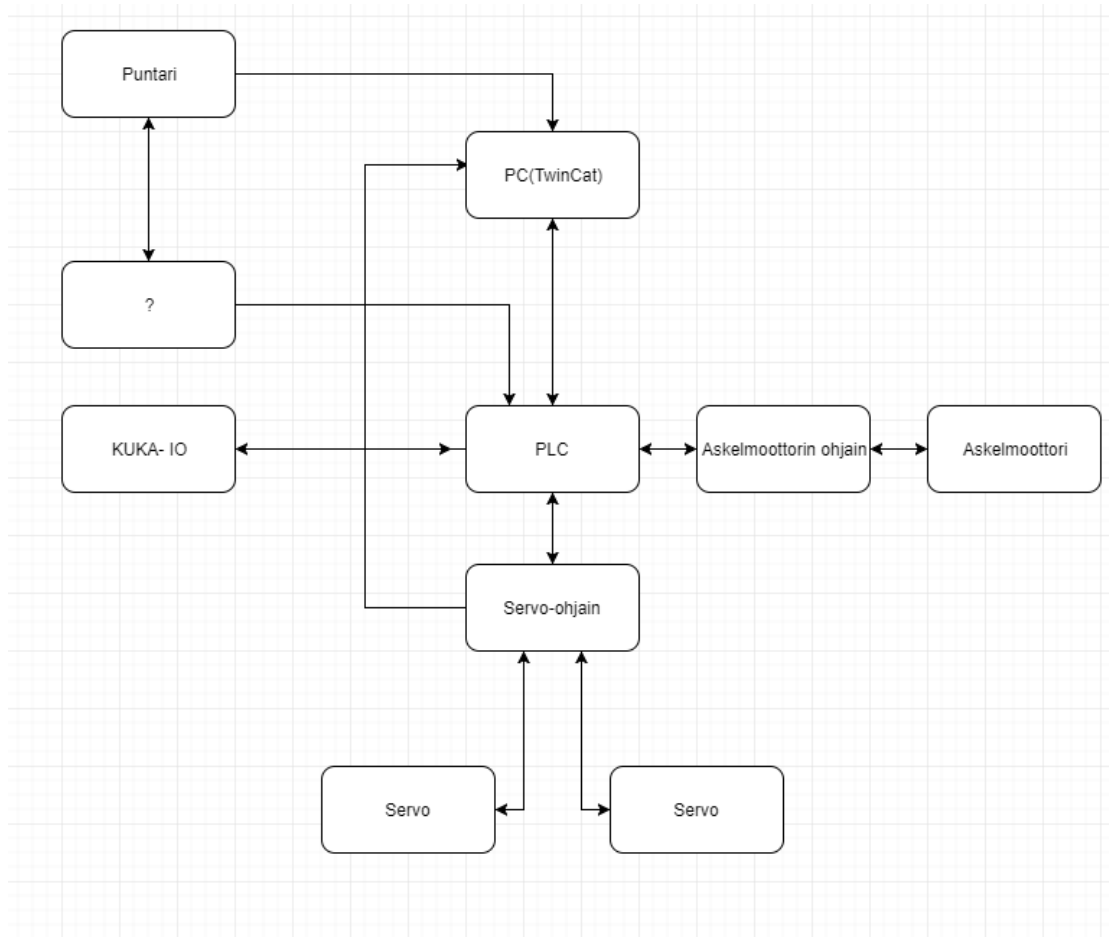
Lineaariyksikön tehtävänä robottisolussa on mahdollistaa tasainen sekoitus sekä purkin pohjalta että pinnalta samalla mahdollistaen robotille muita rinnakkaisia toimintoja. Lineaariyksikön valinnassa on mietitty mekaanisten osien suojaamista mahdollisilta happamilta roiskeilta ja mineraalijauholta. Lineaariyksiköksi valikoitui ROLLCO:n RHL (kuva 39), joka toimii hihnavedolla. RHL-lineaariyksikössä rattaat ja hihna on suojattu kokonaan ympäröivällä alumiinirakenteella. Se on myös helppo kiinnittää alumiiniprofiiliin. Lineaariyksikön liikkuma-alueeksi on valittu 210 mm.



Kuva 39. RHL-lineaariyksikkö (ROLLCO, 2019)

5.4.3 Laitetoimittajan valinta

Tarvittavien ohjauslaitteiden etsiminen aloitetaan piirtämällä suurpiirteinen kartta (kuva 40) tarvittavista komponenteista ja niiden yhteen liittämisestä. Kartan pohjana on kappaleessa 6.2 esitelty suunnitelma, josta on jätetty muutama yksityiskohta huomioimatta. Ensimmäinen suunnitelma on toteutettu alustavien tietojen pohjalta, ja se on muokkautunut lopulliseen versioon vasta, kun laitetuottaja on saatu valittua.

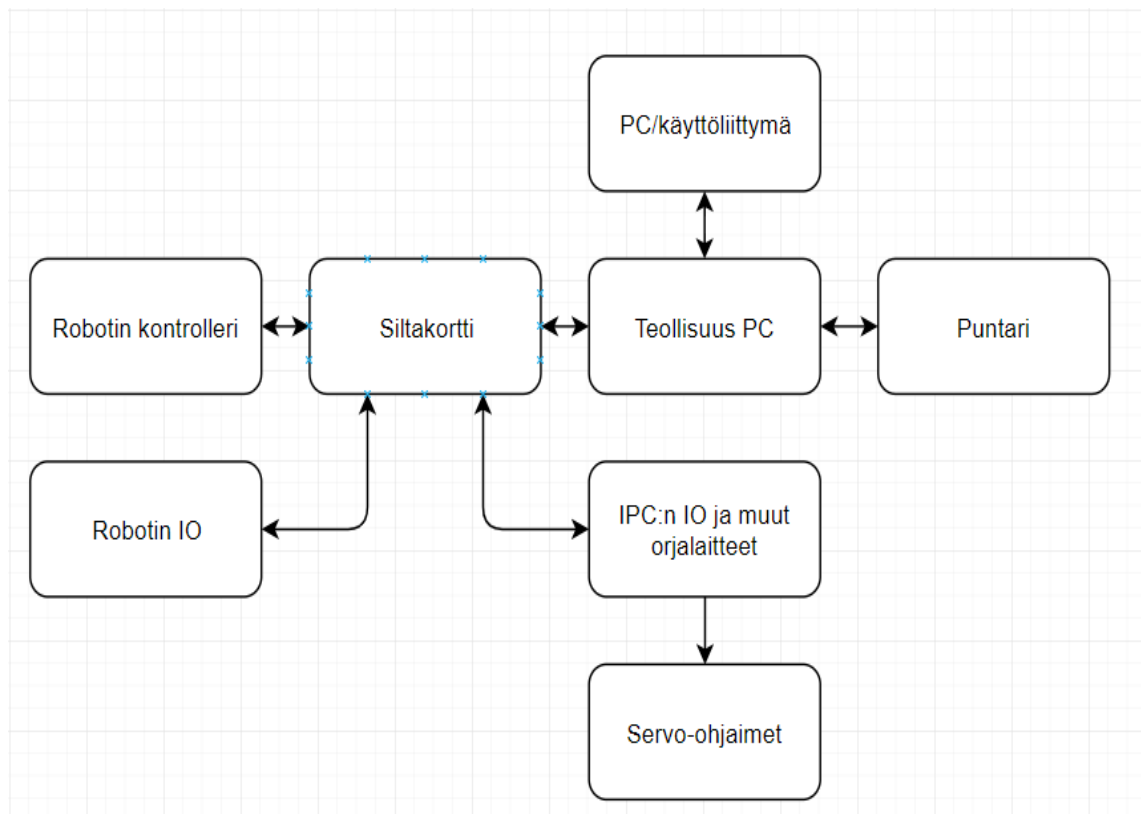


Kuva 40. Ensimmäinen suunnitelma toimilaitteiden ja IO:n kommunikoinnista.

PLC:n, terminaalien, askelmoottorien ja servojen valinta on toteutettu hintavertailulla. Työssä on lähdetty vertailemaan Siemensin, Rexrothin ja Beckhoffin laitteita keskenään. Ennen vertailua on etsitty Beckhoffilta sopivat laitteet, joiden löytymiseen on pyydetty apua Beckhoffin henkilökunnalta (Liite 1). Kun sopivat Beckhoffin laitteet ovat löytyneet, kilpailijoilta on pyydetty tarjous vastaavanlaisista tuotteista. Tarjouksien saavuttua tuotteiden ominaisuuksia ja hintoja on vertailtu.

Sähköpostikeskustelujen pohjalta on käynyt ilmi, että Rexroth on keskittynyt paljon liikkeenohjaukseen, eikä heiltä löydy kovin pienitehoista IPC:tä lainkaan tällä hetkellä. Tämän takia laitteiston hintakin on korkea. Vuoden 2019 aikana Rexrothilta on tulossa myös pienempiä IPC:itä. He eivät myöskään myy askelmoottoreita tai niiden ohjaimia. Näiden tuloksien perusteella työhön ei valittu Rexrothia.

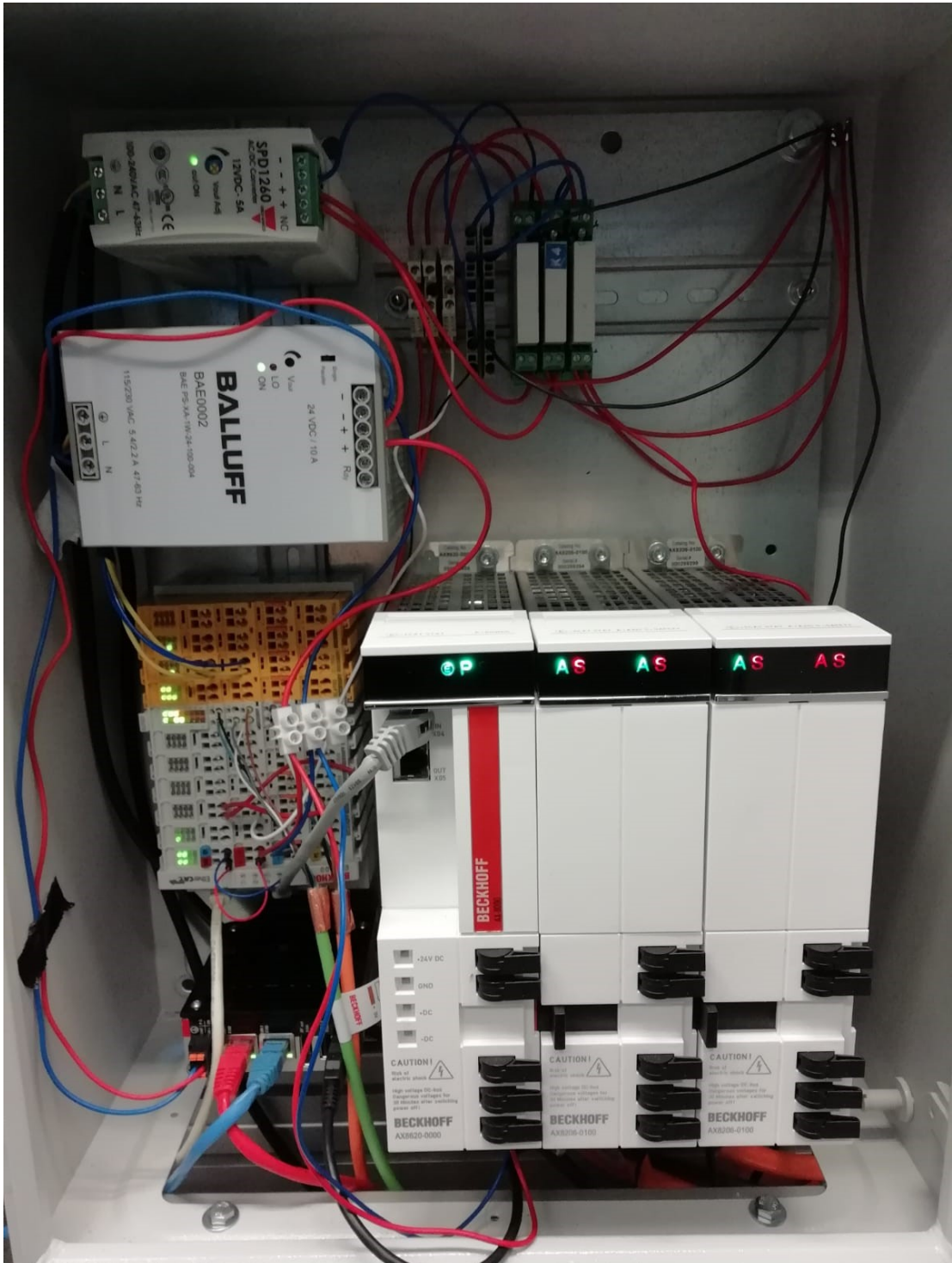
Vertailussa Beckhoff ja Siemens ovat hinnaltaan ja toiminnaltaan lähellä toisiaan. Yksi suurin ero Beckhoffin ja Siemensin välillä on kenttäväylä. Beckhoff käyttää EtherCAT:ia ja Siemens Profibus:ia. Robottisolun reaaliaikaisuudelle ei ole tarkkoja vaatimuksia, mutta Beckhoff on kuitenkin jo entuudestaan hieman tutumpi diplomityöntekijälle, joten laitetoimittajaksi valitaan Beckhoff. Laitetoimittajan löytymisen jälkeen on tehty suunnitelma tarvittavista komponenteista ja niiden yhteen liittämistä. Kuvassa 41 on yksinkertaistettu kuva robottisolun laitteiden kommunikoinnista. Liitteessä 5 on tarkka kuva, miten eri komponentit on sijoitettu systeemiin.



Kuva 41. Lopullinen suunnitelma toimilaitteiden ja IO:n kommunikoinnista.

Alkuperäisen suunnitelman mukaan tarkoituksena on ollut käyttää perinteistä PLC:tä ja käyttää erillistä PC:tä käsittelemään Excel-taulukkoa ja puntarilta tulevaa dataa. Myöhemmin on ymmärretty, että systeemistä saadaan kompaktimpi ja yksinkertaisempi, jos PLC korvataan teollisuustietokoneella (IPC), jossa teho riittää tekemään puntarilta tulevan datan käsittely. Toinen suuri muutos alkuperäiseen suunnitelmaan on Beckhoffin ja KUKA:n välinen kommunikointi, jossa kommunikointi ja turva toteutetaan EtherCAT:in kautta. Tämän takia KUKA:n ja Beckhoffin laitteiston välille täytyy laittaa EL6695-1001 siltakortti, jolla pystytään yhdistämään KR10:n IO ja turvabitit Beckhoffin laitteiston

kanssa. KUKA:n ja Beckhoffin välisestä kommunikoinnista on tarkempaa tietoa kappaleessa 6.4. Kuvassa 42 on lopullisen robottisolun sähkökaappi, jossa Beckhoffin laitteet ja virtalähteet ovat kytkettyinä.

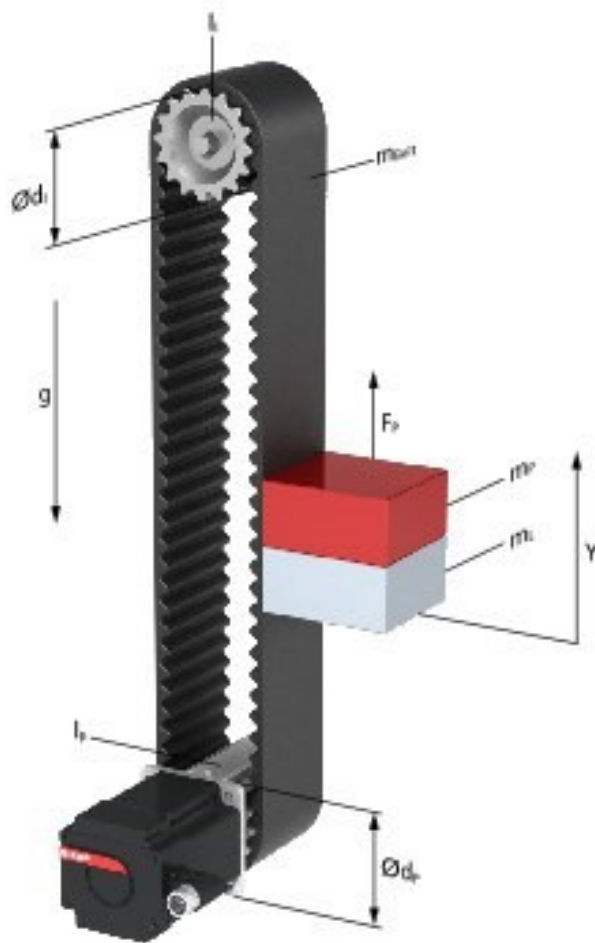


Kuva 42. Beckhoffin laitteet sähkökaapissa.

5.4.3 Servojen valinta

Moottorien valinnassa on pyritty siihen, että sama askelmoottori- ja servomalli sopivat robottisolun kaikkiin käyttökohteisiin. Servomoottoreita on suunniteltu käytettäväksi lineaariyksikössä sekä sekoittimessa. Askelmoottoreita on tarkoitus käyttää jauhon syöttölaitteissa. Servomoottorien mitoitus on tehty TwinCatin lisäosalla Motion Designerillä.

Motion Designerista löytyy diplomityössä käytettävää hammashihnalla toimivaa lineaariyksikköä täysin vastaava malli (kuva 43). Malliin syötetään hammasrattaiden ulkohalkaisijat, hitausmomentit, kuorman massa, hihnan massa ja kitkakerroin (Kuva 44). Hammasrattaiden ulkohalkaisijat löytyvät lineaariyksikön manuaalista ja nostettava maksimikuorma saadaan, kun lineaariyksikön kelkan, sylinterin ja täyden sekoituspurkin massat lasketaan yhteen. Hammashihnan painoa ei löytynyt, joten se arvioitiin sadan gramman painoiseksi. Hammasrattaiden hitausmomentin laskeminen onnistuu Motion Designerin valmiilla hitausmomenttilaskurilla.

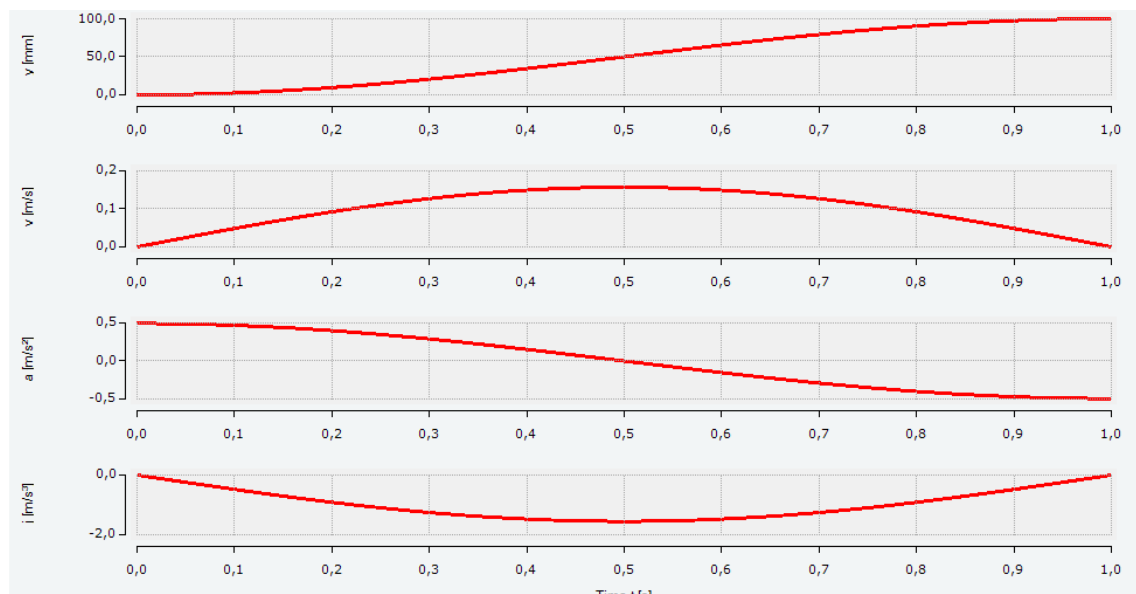


Kuva 43. Linearikuljettimen malli.

Settings			
Configuration		Vertical	▼
Diameter drive pulley	$[\Phi d_p]$	56	mm ▼
Inertia drive pulley	$[I_p]$	0,8	kgcm ² ▼
Diameter idler pulley	$[\Phi d_i]$	56	mm ▼
Inertia idler pulley	$[I_i]$	0,8	kgcm ² ▼
Load mass	$[m_L]$	1,5	kg ▼
Belt mass	$[m_{belt}]$	0,1	kg ▼
Friction coefficient	$[\mu]$	Self defined	▼
		0	
Efficiency	$[\eta]$	90	% ▼
Gravitation	$[g]$	9,81	m/s ² ▼

Kuva 44. Moottorin mitoitus

Mitoituksen jälkeen Motion Designer kysyy liikeprofiilin tietoja, joiden perusteella se piirtää kuvaajan profiilista (Kuva 45).



Kuva 45. Moottorin liikeprofiili.

Lopuksi Motion Designer antaa moottorivaihtoehdot (kuva 46). Moottoriksi valikoituu AM8031-1C20, sillä sen maksimikierronnopeus on sopiva sekoittimeen ja vääntö riittää lineaarikuljettimeen. Kyseisen moottorin maksimikierronnopeus on 3000 rpm ja nimel-

lismomentti on 1.34 Nm. Inertiasuhde on huono lineaarikuljettimeen, mutta sopivilla säädöillä ja pienellä nopeudella lineaarikuljetin kulkee sulavasti. Vaihde parantaa inertiasuhdetta, mutta se vie paljon tilaa, eikä suurille nopeuksille ole tarvetta.

Motor type	Nom safety	Max safety	Inertia ratio	Max speed	M
AM8021-1Bxx-0000	1.09	5.52	105.55	1.2%	
AM8021-1Dxx-0000	1.10	5.51	105.55	.6%	
AM8022-1Dxx-0000	1.74	8.61	55.91	1.0%	
AM8022-1Exx-0000	1.75	8.62	55.91	.6%	
AM8023-1Exx-0000	2.61	13.09	37.92	1.0%	
AM8023-1Fxx-0000	2.62	13.12	37.92	.6%	
AM8031-1Dxx-0000	3.03	12.50	30.62	1.2%	
AM8031-1Fxx-0000	3.07	12.49	30.62	.7%	
AM8032-1Dxx-0000	5.21	16.72	16.80	2.3%	
AM8032-1Exx-0000	5.18	23.97	16.80	1.3%	
AM8032-1Hxx-0000	5.19	23.86	16.80	.7%	
AM8033-1Exx-0000	6.96	21.82	11.59	2.4%	
AM8033-1Fxx-0000	7.00	34.25	11.59	1.3%	
AM8033-1Jxx-0000	7.04	35.35	11.59	.8%	
AM8041-1Dxx-0000	5.19	14.83	13.10	2.3%	
AM8041-1Exx-0000	5.36	18.79	13.10	1.3%	
AM8041-1Hxx-0000	5.23	18.78	13.10	.7%	
AM8042-1Exx-0000	8.92	19.85	7.14	2.9%	
AM8042-1Fxx-0000	8.91	31.23	7.14	1.5%	
AM8042-1Jxx-0000	8.91	38.65	7.14	.9%	
AM8043-1Exx-0000	12.26	23.66	4.93	3.0%	

Kuva 46. Moottorivaihtoehdot.

6 LOGIIKAN OHJELMOINTI JA KOMMUNIKOINTI

Tässä kappaleessa esitellään pintapuolisesti robottisolun automaation logiikan toteutusta ja rakennusta. TwinCat:sta esitellään yksi esimerkki, joka kuvaa koko robottisolun koodin rakennetta. PLC:n logiikan lisäksi esitellään muutama python-kirjasto, joita käyttämällä puntarin, Excelin ja logiikan välinen kommunikointi on saatu toimimaan. Lopuksi esitellään, miten turvapuolen logiikka toteutetaan TwinCat:lla, sekä miten KUKA:n turvapuoli saadaan yhdistettyä TwinSafen turvalogiikkaan.

6.1 Käyttöliittymä

Käyttöliittymä (Liite 6) on toteutettu TwinCat3:n visualization-työkalulla. Käyttöliittymä on tehty simulointivaiheessa, joten siinä on visuaalisia elementtejä kuvaamassa moottorien ja syöttölaitteiden toimintaa. Käyttöliittymässä on myös useita nappeja, jotka ovat käytössä vain testausvaiheessa. Tärkeimmät painonapit käyttöliittymässä ovat START, PAUSE, READY ja RESTART. START-napilla geopolymeerien valmistus alkaa, PAUSE-napilla ohjelma menee tauolle, READY-napilla kuitataan hätäseis ja tehdään alustukset ja RESTART-nappi lopettaa ohjelman ja aloittaa ohjelman alusta.

6.2 PLC:n logiikka

Robottisolun ohjaus on toteutettu kokonaisuudessaan PLC:n puolella. Robotilla on ohjelmoitu ainoastaan liikefunktiot, joita PLC kutsuu logiikan mukaan.

Kuvassa 47 on robottisolua ohjaavan koodin pääohjelma, joka suorittaa alustuksen ja käynnistää robottisolun toimintaa ohjaavan tilakoneen (*fbTilakone*). Alustuksessa alustetaan turvapiiri (*fbInit_Estop*), joka tässä tapauksessa on yksi hätäseis-nappi.

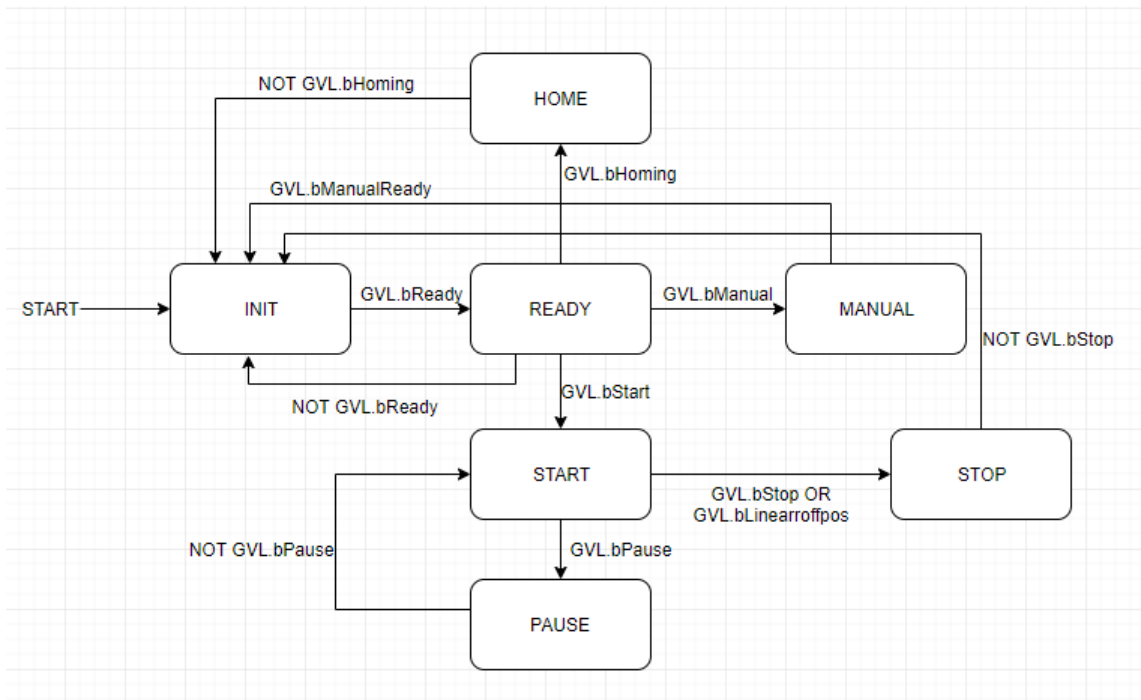
```

1  CASE iState OF
2      0: //Alustus
3          fbInit_Estop();
4
5      IF fbInit_Estop.bDone THEN
6          iState:= 10;
7      END_IF
8
9      10: //Käynti
10         fbTilakone();
11
12         IF GVL.bRestart THEN
13             iState := 20;
14         END_IF
15
16         IF SAFETY.g_xTS_COMerror THEN
17             iState := 20;
18         END_IF
19
20         20: // Uudelleen alustus
21             iState := 0;
22
23     END_CASE
24

```

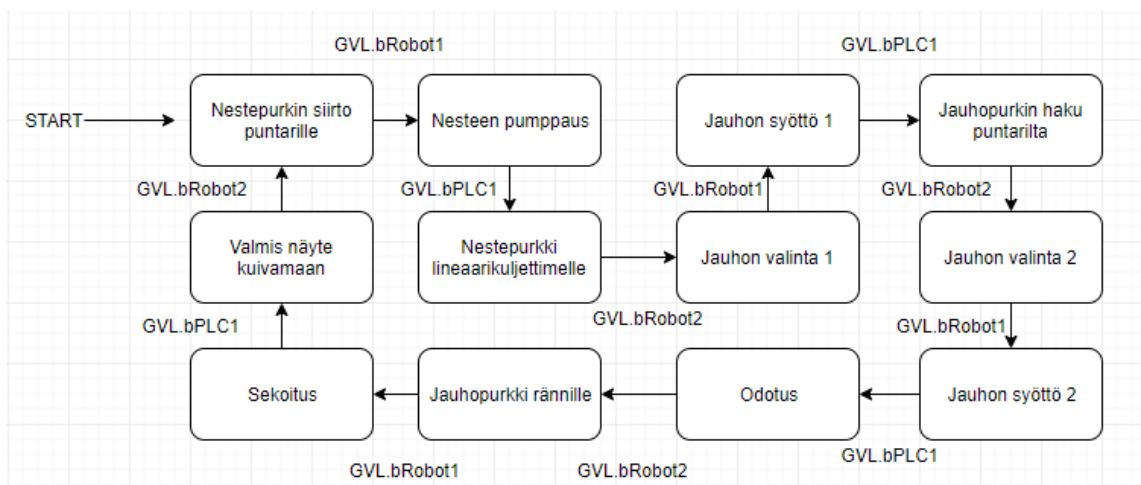
Kuva 47. Robottisolun pääohjelma.

Pääohjelmana toimii tilakone, joka on ohjelmoitu tilakaaviolla (Liite 3). Tätä tilakonetta ohjataan käyttöliittymällä. Kuvassa 48 tilakone on piirretty helppolukuiseen muotoon. Tilojen väliin on merkattu muuttuja, joka aktivoi tilojen vaihtumisen. Tilakoneen käynnistyttyä kaikki muuttujat alustetaan INIT-tilassa. INIT-tilasta siirrytään READY-tilaan, jossa myös hätäseis-napin kuittaus tapahtuu. READY-tilasta voidaan mennä muihin tiloihin, kuten MANUAL- ja START-tilaan. MANUAL-tilassa voidaan ohjata sekoitinta ja syöttölaitteita manuaalisesti. START-tilassa tapahtuu geopolymeerien valmistus. Lisäksi tilakoneesta löytyy PAUSE-, STOP ja HOME-tilat. PAUSE-tila pistää sekoitusohjelman tauolle. STOP-tila pysäyttää geopolymeerin valmistussyklin. HOME-tilassa lineaarikuljetin ajaa kotiasemaan, jos se on jäänyt ylös. Valmistussykli ei käynnisty, jos lineaarikuljetin ei ole kotiasemassa.



Kuva 48. Robottisolun tilakone.

START-tilan sisällä on vielä toinen tilakone (kuva 49), jossa on kaikki geopolymeerin valmistukseen liittyvät vaiheet. Liitteessä 4 on koko tilakone TwinCatissa, jossa keltaisten laatikoiden sisällä on Beckhoffin laitteiden toimintoja ja vihreissä laatikoissa on odotustiloja tai logiikkaa liittyen robotin liikkeisiin. KR10 ja PLC kertovat IO:n välityksellä milloin siirrytään seuraavaan vaiheeseen. bRobot-muuttujat ovat KR10:n output-viestejä ja bPLC-muuttujat ovat PLC:n output-viestejä. Keltaisten laatikoiden sisään voidaan laittaa rinnakkaisia ohjelmia, kuten kuvan 49 esimerkissä sekoitus-laatikossa on lineaarikuljetinta ohjaava ohjelma ja sekoitinta ohjaava ohjelma.



Kuva 49. Geopolymeerin valmistuksen tilakone.

Viimeinen vaihe robottisolun logiikkaketjussa on toiminnallisten osien koodi, mitkä on ohjelmoitu ST-ohjelmointikielellä. Esimerkkinä tästä on Liite 7, missä on peristalttisen pumpun askelmoottoria ohjaava koodi.

6.3 ADS-viestintä ja reseptin luku

Painodatan lukeminen ja lähettäminen sekä Excelin käsittely on toteutettu pythonilla. Painodatan lukeminen on toteutettu sarjaportin kautta pyserial-kirjastolla. Luettu data lähetetään IPC:lle ADS-protokollalla pyads-kirjastoa käyttäen. Kuvan 50 esimerkissä on ADS-portin avaus PLC/IPC:n ja PC:n välille. Portin avaamiseen tarvitaan portin numero ja PLC/IPC:n AMS net ID. TwinCat3:ssa toimii portti 851 ja TwinCat2:ssa portti 801. AMS net ID löytyy PLC:ltä. Portin avauduttua ADS:llä voidaan lukea PLC:n muuttujia tai kirjoittaa niihin uusia arvoja. Kuvan 51 esimerkissä PLC:n iPythonLmass-muuttujaan kirjoitetaan muuttuja Lmass.

```
pyads.open_port()
plc = pyads.Connection('172.18.232.146.1.1', 851)
plc.open()
print('Port 851' + ' is open...')
```

Kuva 50. Yhteyden luonti PLC/IPC:hen.

```
plc.write_by_name('GVL.iPythonLmass', Lmass, pyads.PLCTYPE_DINT)
```

Kuva 51. Muuttujan kirjoittaminen PLC:lle Pythonilla.

Excelin käsittelyyn on käytetty useampaa kirjastoa, jotka ovat slwt, xlrd ja openpyxl. Näiden kirjastojen avulla voidaan lukea käyttäjän kirjaamat tavoitemassat jauholle ja nesteelle Excel-tilukosta, jotka voidaan taas lähettää PLC:lle pyads-kirjaston avulla. Lopuksi pythonilla luetaan PLC:ltä paljonko jauhoa ja nestettä todellisuudessa on syötetty ja merkataan tulokset Exceliin. Kuvassa 52 on esimerkkinä Excel-tilukosta viiden geopolymerisekoituksen tavoitemassat ja niiden todelliset punnitut massat.

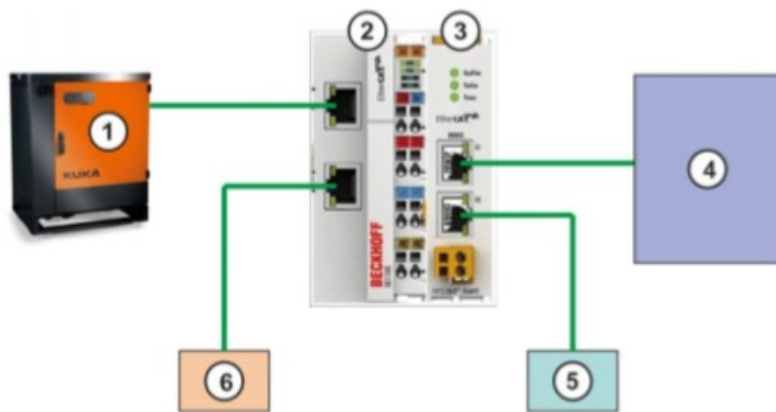
	A	B	C	D	E	F	G	H	I
1	Sample	Powdertype	Liquidtype	Mass of powder	True Mass of powder	Mass of liquid	True Mass of liquid	Date and time	
2	1			15	15,04	15	15,52	2019-06-10 14:19:26	
3	2			20	20,14	15	15,36	2019-06-10 14:23:50	
4	3			23	22,89	20	21,26	2019-06-10 14:28:24	
5	4			25	25,08	20	20,45	2019-06-10 14:33:02	
6	5			25		17			
7									

Kuva 52. Tulokset Excel-tilukossa.

6.5 KUKA:n ja PLC:n FsoE-kommunikointi

Beckhoffin turvalogiikka toteutetaan TwinSAFE-tuoteperheellä, jossa on käytössä turvallisuusprotokolla Safety over EtherCAT. Turvallisuussovellukset ohjelmoidaan ja konfiguroidaan TwinCat-ohjelmistolla graafisesti. TwinCat:ssa ohjelmoitu turvaohjelma ladataan TwinSAFE PLC:lle, jonka kautta kaikki TwinSAFE-komponentit kommunikoivat kenttäväylän isännän kanssa. TwinSAFE-komponentit voivat käyttää tiedonsiirtoon EtherCAT-väylän lisäksi myös muita väyläratkaisuja, jolloin Beckhoffin turvalaitteet voidaan lisätä suurimpaan osaan teollisuuden automaatiojärjestelmistä.

Tässä diplomityössä tarkoituksena on yhdistää KUKA KR10-robotin turvalogiikka TwinSAFE:n turvalaitteisiin yhdellä Ethernet-liitännällä. Tällöin turvasignaali täytyy tuoda Beckhoffin PLC:ltä FsoE:lla (Safety-over-EtherCAT protokolla) KR10:lle ja kumpikin laite tulee olla kenttäväylän isäntiä. Tätä varten KUKA on valmistanut siltakortin EL6695-1001, joka avulla FsoE-kommunikointi robotin ja PLC:n välillä onnistuu. Myös Beckhoff valmistaa vastaavanlaista korttia. Tällä siltakortilla IO saadaan vaihdettua robotin ja PLC:n logiikan välillä. Kuvasta 53 nähdään kuinka EL6695-1001 siltakortti tulee kytkeä systeemiin.



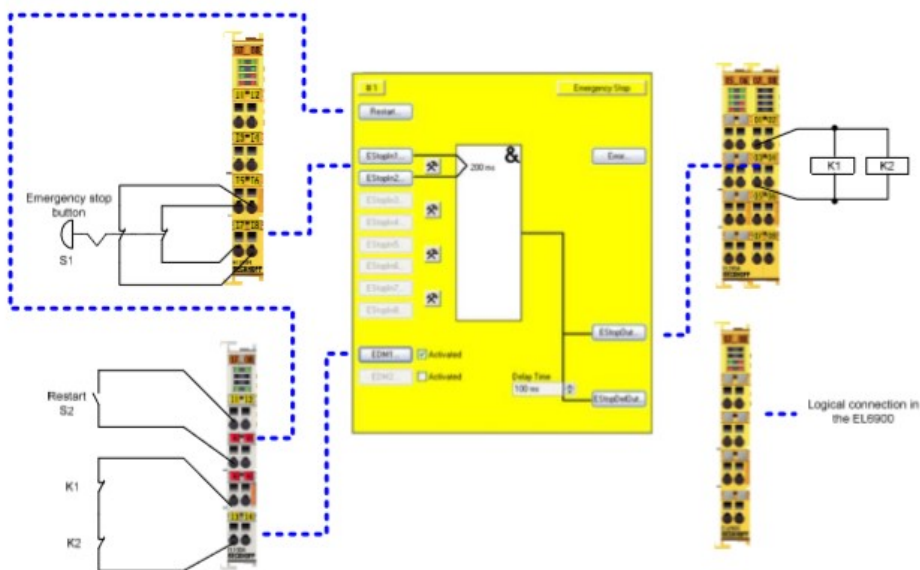
Kuva 53. KRC4 compact (1), EK1100 EtherCAT-liitin (2), EL6695-1001 (3), IPC/PLC (4), Turvaterminaalit ja muut terminaalit (5) ja Muita EtherCat- orjia KR C4:lle (6).

6.6 Turvallisuuden toteutus robottisolussa

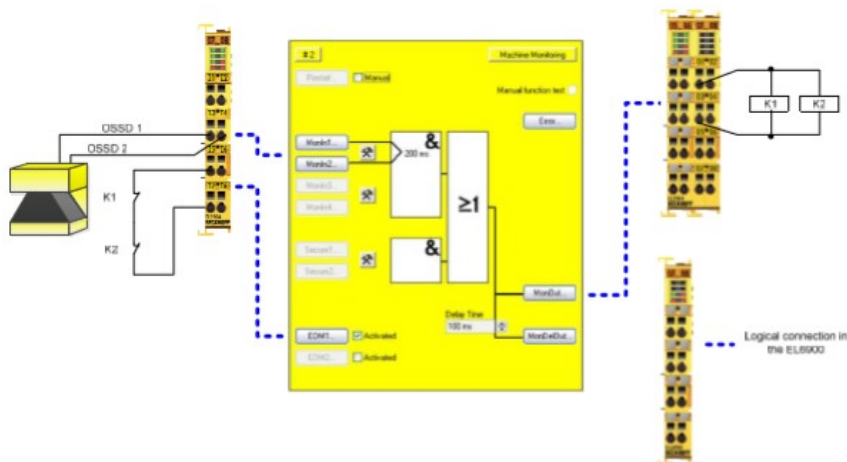
Tämän diplomityön versio geopolymeerejä valmistavasta robottisolusta on tehty BISG:n laboratoriossa koekäyttöön. Koekäyttöversiossa robottisolun turvallisuutta ei ole suunnit-

teltu turvastandardien mukaisesti. Turvallisuus on toteutettu kahdella hätäseisnapilla, toinen robotin ohjelmointilaitteessa ja toinen kytkettynä turva-PLC:n turvalogiikkaan. Kuvassa 54 on Beckhoffilta saatu ohje hätäseis-napin kytkentään. Erona diplomityön robotisolun ja kuvan 54 ohjeen välillä on, että kuitausnappi on toteutettu käyttöliittymän kautta.

Robottisoluun on suunniteltu jatkokehityksenä sivuille asennettavat turva-skannerit ja eteen yksi laserverho. Kuvasta 55 nähdään laserskanneriin tarvittava logiikka ja ohjeet sen kytkemiseen. Laserverholle voidaan tehdä samanlainen kytkentä ja logiikka kuin laserskannerille.

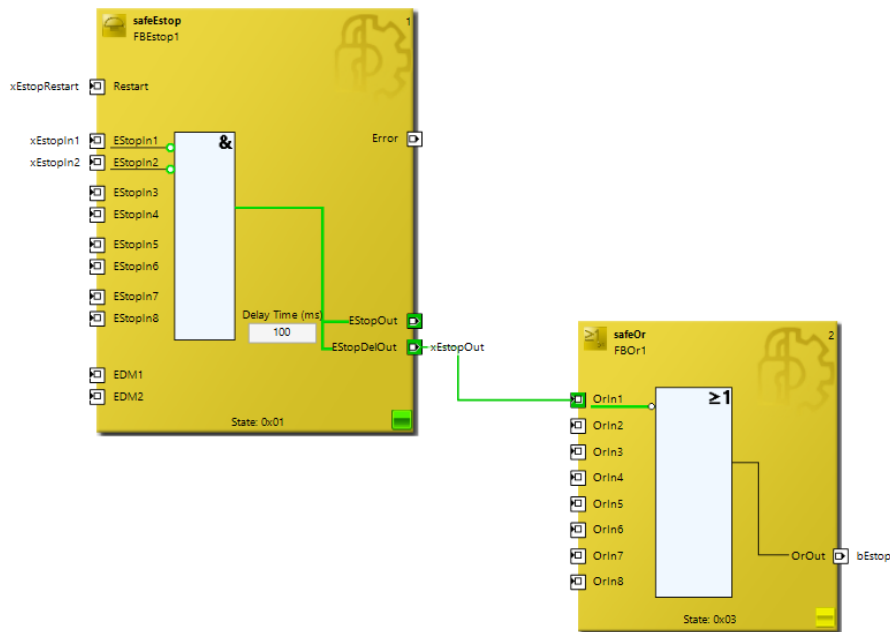


Kuva 54. Robottisolun hätäseis-napin kytkentä ja logiikka. (Beckhoff, 2019) (Kuva julkaistaan Beckhoffin luvalla.)



Kuva 55. Laserskannerin turvakytkentä. (Beckhoff, 2019) (Kuva julkaistaan Beckhoffin luvalla)

TwinSafe:n logiikka ohjelmoidaan aina valmiilla graafisilla elementeillä, joissa on kytkentöihin vaadittavat sisäänmenot ja ulostulot. Kuvassa 56 robottisolun hätäseis-napin muuttujat EstopIn1 ja EstopIn2 on kytketty hätäseisnappiin ohjeiden mukaan. Tässä sovelluksessa hätäseis-nappi on aukeava, joten hätäseis muuttujiin on pitänyt lisätä NOT-portit.



Kuva 56. Turvalogiikka TwinCatissa.

Kuvasta 56 ja 57 nähdään minne mitkään muuttujat on kytketty turvalogiikassa. Custom FsoE Connection- muuttujat ovat KR10:n turvabittejä, jotka TwinCat löytää siltakortin kautta. Kaikki KR10:n turvabitien selitykset ja vaaditut kytkennät löytyvät KUKA:n manuaaleista. Myös servojen turvabitit on kytketty samaan nappiin (Drive13-14).

Variable Mapping					
Variables	Group Ports	Replacement Values	Max Start Deviation		
Assigned Variable	Direction	Alias Port	Port Name	Instance Name	Function Name
xEstopRestart	input	xEstopRestart.In (TwinSafeGroup1)	Restart	FBStop1	safeEstop
xEstopIn1	input	Term 10 (EL1904) - Module 1 (FSOES).InputChannel1 (TwinSafeGroup1)	EStopIn1	FBStop1	safeEstop
xEstopIn2	input	Term 10 (EL1904) - Module 1 (FSOES).InputChannel2 (TwinSafeGroup1)	EStopIn2	FBStop1	safeEstop
xEstopOut	output	Custom FSoE Connection_1.Safe Data Byte 0[0] (TwinSafeGroup1)	EStopDelOut	FBStop1	safeEstop
		Custom FSoE Connection_1.Safe Data Byte 0[1] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[2] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[3] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[4] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[5] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[6] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 0[7] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[1] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[2] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[3] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[4] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[5] (TwinSafeGroup1)			
		Custom FSoE Connection_1.Safe Data Byte 1[6] (TwinSafeGroup1)			
Drive 13 (AX8206-0100-0101) - Device (Safety).Axis A STO (TwinSafeGroup1)					
Drive 13 (AX8206-0100-0101) - Device (Safety).Axis B STO (TwinSafeGroup1)					
Drive 14 (AX8206-0100-0101) - Device (Safety).Axis A STO (TwinSafeGroup1)					
Drive 14 (AX8206-0100-0101) - Device (Safety).Axis B STO (TwinSafeGroup1)					
Term 11 (EL2904) - Module 1 (FSOES).OutputChannel1 (TwinSafeGroup1)					
bEstop	output	bStop.Out (TwinSafeGroup1)	OrOut	FBOr1	safeOr

Kuva 57. Turvalogiikan muuttujat.

7 TESTAUS JA TULOKSET

Tässä kappaleessa käsitellään diplomityössä suunnitellun ja valmistetun robottisolun toimintavarmuutta ja käytettävyyttä. Tämän lisäksi pohditaan, miten robottisolua voidaan kehittää eteenpäin ja mitkä ovat robottisolulle suunnitellut tavoitteet jatkossa. Lopuksi esitellään yksi valmistettu resepti ja sillä aikaansaadut tulokset.

7.1 Testaus

Robottisolun perustoimintojen, kuten esimerkiksi syöttimien, robotin liikkeiden ja sekoituksen toimintavarmuutta testattiin enimmäkseen käyttämällä vettä ja masuunikuonaa. Tästä ei synny oikeaa geopolymeeriä, mutta näin pystyttiin materiaalien huonosta saatavuudesta huolimatta tekemään useita testejä.

Perustoiminnot sujuvat robottisolussa hyvin. Robottisolun kevyen rakenteen takia pienet töytäisyt voivat kuitenkin liikuttaa alumiiniprofiileita hieman. Tästä voi seurata jonkin pisteen liikkuminen liian paljon, jolloin robotti tekee virheen. Tällöin robotti ei esimerkiksi saa purkkia säiliöstä. Alumiiniprofiilien ja muiden kiinteiden osien tukevaan kiinnitykseen tulee kiinnittää lisää huomiota.

Testeissä esiintyi yksi robottisolun toiminnan pysäyttävä ongelma, joka liittyi PLC:n logiikkaan. PLC:n logiikan tilakone on koodattu StateChart-koodikielellä ja jostain syystä PLC kadottaa tilansa satunnaisesti, jolloin robottisolun toiminta pysähtyy. Tämän ongelman syy pitää jatkossa selvittää tai tilakone on koodattava uudelleen eri kielellä, kuten ST:llä.

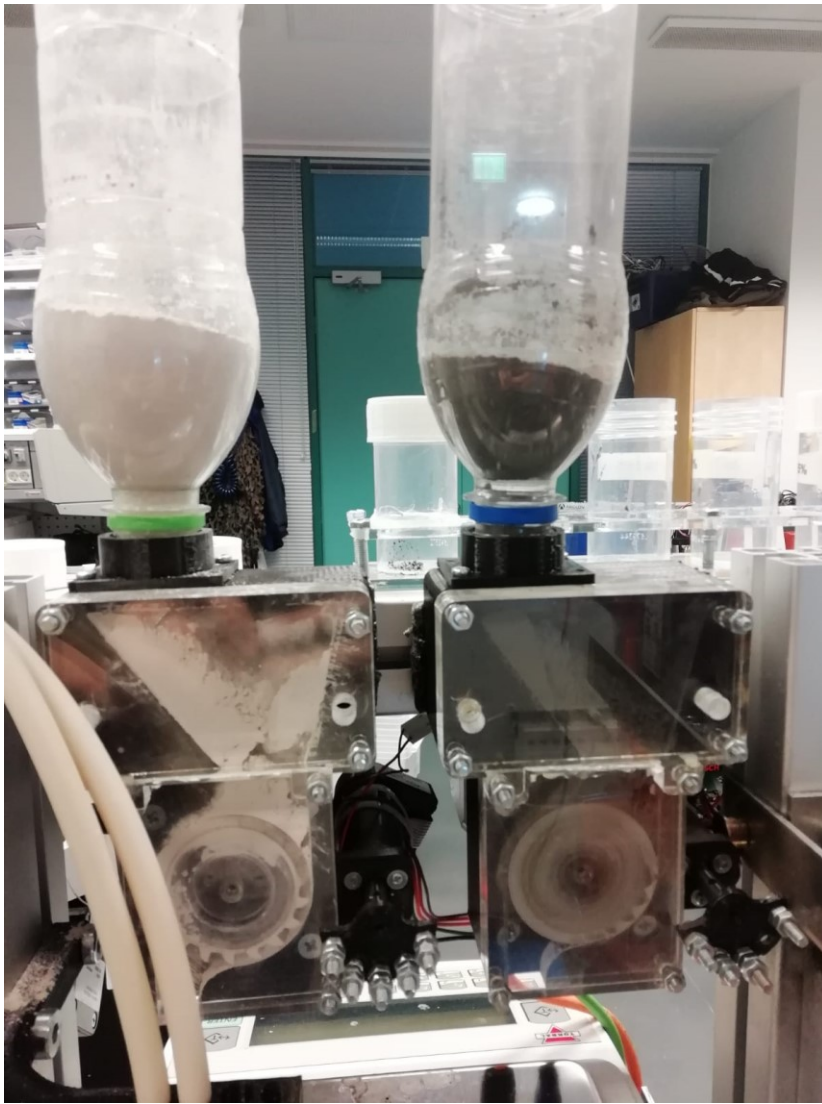
Testeissä huomattiin, että robottisolun puhtaana pysymiseksi tarvitaan lisää toimintoja. Jauhosyöttimet eivät olleet tarpeeksi tiiviitä ja tärinän takia hieno hiekka pääsi putoamaan väleistä, joista sitä ei ollut tarkoitus syöttää. Lisäksi syöttörännin säiliö oli liian pieni. Jos jauhoa syötettiin yli 30 grammaa, osa hiekasta tippui tärinän takia. Toisaalta sekoituspurkkien koko rajoittaa sen, paljonko jauhoa ja nestettä voidaan sekoittaa purkissa ja 40 grammaa masuunikuonaa on maksimi, mitä purkkiin on järkevää syöttää. Tämäkin tietysti riippuu syötettävän jauhon tiheydestä.

Robottisolun ja sekoituksien puhtauden takia robottisoluun pitää vielä lisätä pesutoimintoja. Sekoituksen jälkeen sekoitin jää likaiseksi. Likaista sekoitinta on kokeiltu pyörittää suurella kierrosnopeudella vesiastiassa ja ylimääräinen lika näyttäisi liukenevan hyvin. Tätä tarvitsee kuitenkin jatkossa testata vielä erilaisilla resepteillä. Ränniin jää myös hie-man jäämiä masuunikuonasta, mutta määrät ovat niin pieniä, ettei tällä ole nähty olevan merkittävää haittavaikutusta. Jauholaadun vaihdossa voidaan ajaa puhdistustoimenpiteenä muutama ylimääräinen näyte solun läpi, jotta vanha jauho poistuisi hukkanäytteiden mukana.

Sekoituksen testauksessa on kokeiltu muutamaa eri jauhoa ja myös oikeaa aktivaattoria veden sijasta. Tämän avulla pyrittiin selvittämään, kuinka kauan sekoituksen tulee kestää ja minkälainen sekoitusliike on sopiva. Tässä testissä huomattiin, että jauhон laatu määrää sekoituksen keston ja jauhon syöttönopeuden. Pienemmän tiheyden omaavat jauhot vaativat hitaampaa syöttönopeutta ränniltä ja pidempää sekoitusaikaa. Raskaammat jauhot taas sekoittuivat huomattavasti nopeampaa ja syöttönopeuteen ei tarvinnut kiinnittää niin paljoa huomiota. Jauholaadun muuttamiseen liittyy myös muita haasteita. Esimerkiksi syöttölaitteita ohjaavaan koodiin, joka laskee rattaan pyörimismatkan, pitää koodata laskukaava, jossa laskettavan jauhон tiheys muuttuu jauhovalinnan myötä. Samalla tavalla tulee koodata sekoitusaika ja rännin syöttönopeus jauholaatu huomioon ottaen. Sekoituk-sen parametrien muuttamisen lisäksi sekoituksessa voitaisiin kokeilla erilaisia sekoitti-men päitä. Näissä kokeissa huomattiin myös, että alkuperäinen nestepumppu ei toiminut aktivaattoreilla, jotka olivat viskoosisempia, joten pumppu korvattiin uudella peristalti-sella pumpulla (6.3.6).

7.2 Tulokset

Ensimmäiseen koesarjaan haluttiin käyttää kahta jauhoa, joten yhden jauhosityttimen rin-nalle asennettiin toinen samanlainen syöttölaite (kuva 58). Jauhoiksi on valittu sopivan raskaat ja helposti sekoittuvat jauhot, jotta syöttäminen ja sekoitus onnistuisivat vaivatta. Testeihin on otettu käyttöön myös uusi peristalttinen pumppu, joka on todettu toimivaksi. Kuvan 59 Excel-taulukosta nähdään koesarjan resepti ja taulukkoon on täydennetty myös syöttimiltä tulleet oikeat massat.

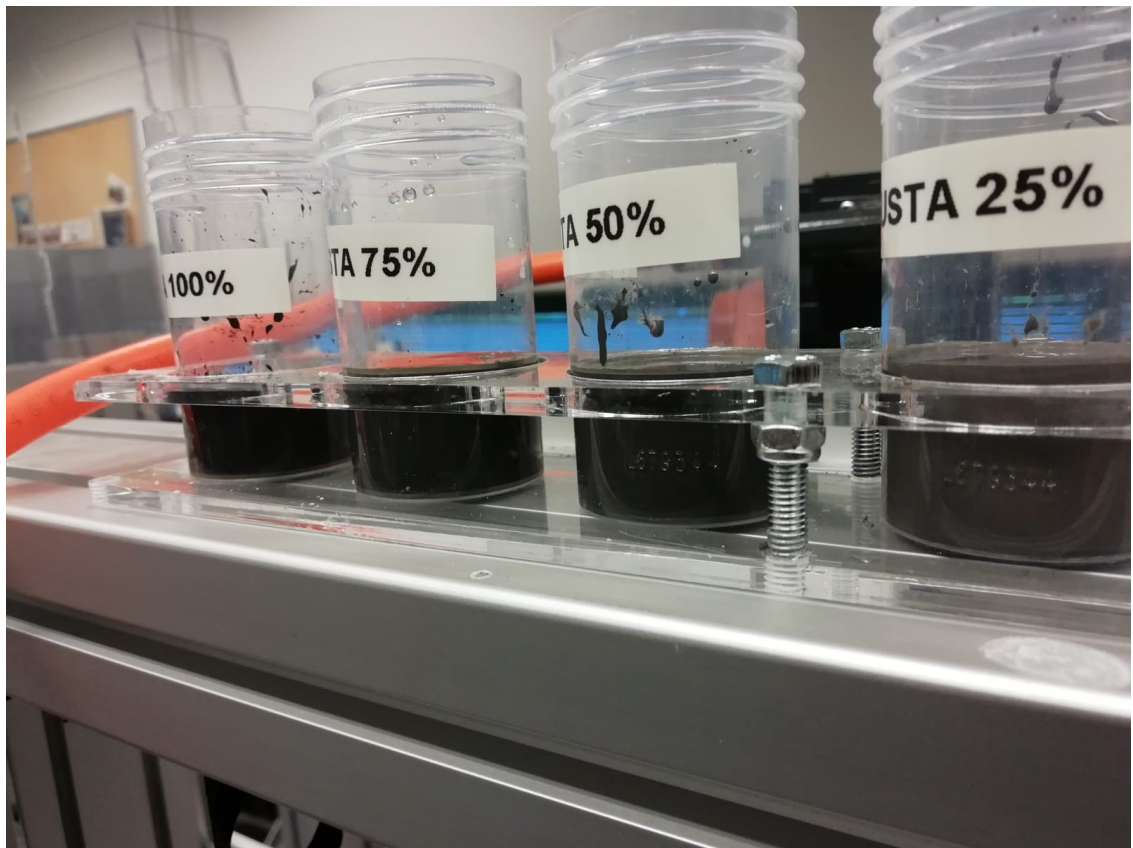


Kuva 58. Kaksi jauhosyötintä valmiudessa.

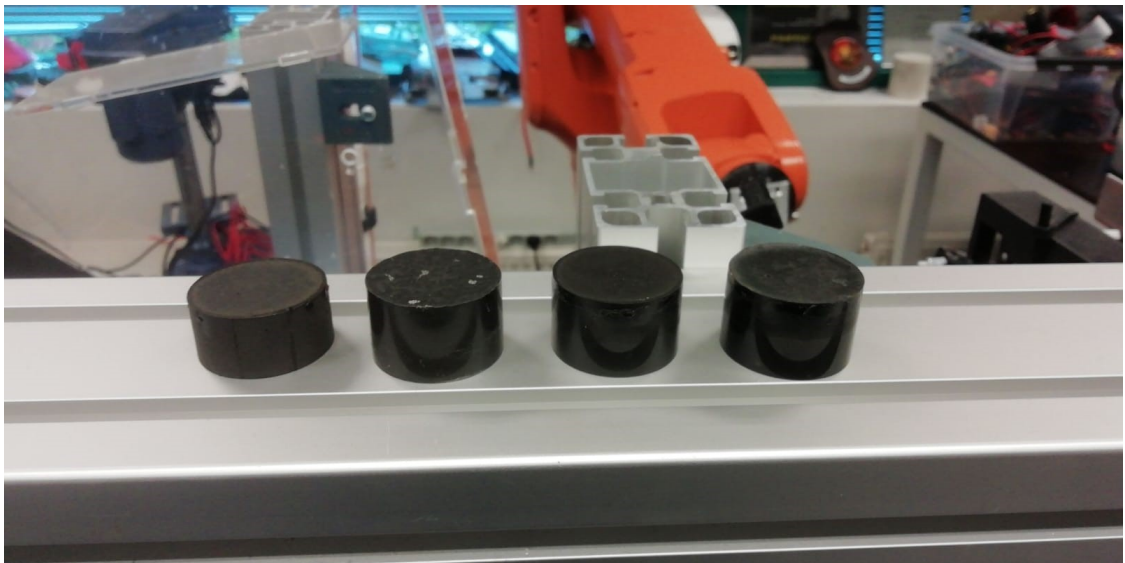
Sample	Powder1 (GD#1)	Powder2	Real mass of powder 1	Real mass 2	Liquid 1	Real mass of Liquid 1	Date
1	32		32,04		8	8,01	9.9.2019 12:52
2	21	7	21,08	7,21	12	11,95	9.9.2019 13:04
3	14	14	13,92	13,94	12	12,06	9.9.2019 13:15
4	7	21	5,78	20,98	12	12,19	9.9.2019 13:26
5	0		0		0	0	

Kuva 59. Valmistettujen geopolymeerinäytteiden reseptit ja punnitut massat.

Kuvassa 60 sekoitukset ovat kuivamassa. Vasemmalla on näytekappale, jossa on pelkästään mustaa jauhoa, ja kuvassa oikealle mentäessä mustan jauhon osuus vähenee. Kuvassa 61 on samat näytteet samassa järjestyksessä ja ne ovat kovettuneita lopulliseen muotoonsa.



Kuva 60. Geopolymeeriäytteet juuri sekoitettuna.



Kuva 61. Valmiit geopolymeerinäytteet kovettuneena.

7.3 Jatkokehitys

Jatkokehityksessä tullaan ensimmäisenä keskittymään testauksissa tulleisiin puutteisiin ja ongelmiin. Erityisesti erityyppisten ja rakenteisien jauhojen syöttäminen ja sekoitus tulisi onnistua. Myöhempää käyttöä ajatellen jauhosyöttimiä tarvitaan myös lisää. Useamman jauhosyöttimen käyttö vaatii solulta isoja muutoksia, koska puntarin pinta-ala on

liian pieni useammalle kuin kahdelle syöttimelle. Jatkoa varten on suunniteltu liikkuvaa puntaria, joka kulki jauhosyöttimien alla lineaarikuljettimella.

Jatkokehitykseen kuuluvat myös geopolymeereille tehtävien mittauksien ja kokeiden integroiminen robottisoluun. Ensimmäisenä tavoitteena on mitata sekoituksen jähmettymiseen kuluva aika. Markkinoilta löytyy useita malleja betonin jähmettymisajan mittaukseen ja näitä samoja laitteita voidaan käyttää myös geopolymeereihin.

Lisäksi robottisolun turvallisuuden kehittäminen on yksi tärkeä jatkokehityskohde. Tämänhetkisessä versiossa robottisolu pysäyttää toiminnot törmäystilanteissa ja hätäseisnapista, joita on kaksi. Robottisolun sisälle menemistä ei kuitenkaan ole estämässä mikään, joten tähän tarvitaan vielä laserverhoja tai –turvaskannereita. Näillä voidaan estää robottisolun sisälle meneminen sen toiminnan aikana.

YHTEENVETO

Diplomityön tavoitteena oli tehdä robottisolun, joka valmistaa hyvälaatuisia geopolymeerinäytteitä annettujen reseptien perusteella. Työn tarkoituksena oli helpottaa uusien toimivien geopolymeerireseptien löytämistä ja erilaisien reseptien ominaisuuksien tutkimista.

Työn teoriaosassa käytiin läpi robottisolun suunnitteluun ja valmistukseen liittyvä teoria. Ensiksi esiteltiin paremmin, mitä robottisolulla on tarkoitus valmistaa, jonka jälkeen siirryttiin teknisempään osioon. Robottisolun suunniteltiin teollisuusrobotin ehdoilla, joten teoria aloitettiin esittelemällä erilaisia robottimalleja. Samalla käsiteltiin erilaisia sovelluksia, joissa ne toimivat parhaiten. Tämän jälkeen esiteltiin robottisolussa käytettävän 6-DOF-teollisuusrobotin tärkeimpiä käyttökohteita. Teollisuusrobotti-osion lopuksi käytiin läpi sellaisia tärkeimpiä ohjelmointiin liittyviä ominaisuuksia, jotka tukevat robotin ohjelmointia ja mahdollistavat sille hyvän pohjan.

Teoriaosan toisena isona kokonaisuutena oli ohjelmoitava logiikka ja laitteiden välinen kommunikointi. Tämä aihealue oli hyvin laaja, joten alue rajattiin vain työn kannalta oleelliseen teoriaan. Ensiksi esiteltiin ohjelmoitavan logiikan perustietoja ja toimintaa, jonka jälkeen siirryttiin sen ohjelmointiin. PLC:n ohjelmoinnista käsiteltiin vain työn kannalta oleelliset ohjelmointikielet. Tämän jälkeen siirryttiin Ethernetiin, reaaliaikaisiin systeemeihin ja kenttäväyliin. Ethernetistä ja reaaliaikaisesta systeemistä kerrottiin perustiedot, jonka jälkeen siirryttiin kenttäväyliin. Niistä tuotiin esille erilaiset standardisoidut kenttäväylät ja käytiin läpi niiden reaaliaikaisuus. Tämän jälkeen esiteltiin työn kannalta oleellisimmat kenttäväylät.

Teoriaosan jälkeen siirryttiin työn suunnittelu- ja toteutusvaiheeseen. Ensiksi esiteltiin työn suunnitelma sekä tavoitteet. Tämän jälkeen esiteltiin suunnitelman toteutus sekä miten eri laitevalintoihin päädyttiin. Robottisolun pohjaksi valmistettiin 20 mm teräslevystä vesileikkaamalla painava alusta, jota oli kuitenkin helppo liikuttaa pumppukärryllä. Muita toiminnallisia osia varten rakennettiin alumiiniprofiilikehikko, johon oli helppo kiinnittää kaikki toimilaitteet ja mekaaniset osat. Toiminnallisten osien valmistuksessa ja protoilussa käytettiin paljon hyödyksi 3D-printteriä ja laserleikkuria. Kestävämät osat, kuten moottorien kiinnikkeet, valmistettiin Oulun yliopiston työpajalla.

Seuraavaksi siirryttiin toimilaitteiden ja laitetuimittajien valintaan. Robotiksi valittiin KUKA:n KR10-teollisuusrobotin, sillä se oli jo valmiina käytettävissä ja se vastasi kokoluokaltaan robottisolun tarpeita. Kaikki muut toimilaitteet täytyi kilpailuttaa ja samalla vertailla valmistajien välisiä eroja. Lopulta PLC:ksi valikoitui Bechhoffin C6015 teollisuus-PC ja täten myös kaikki moottorit sekä niiden ohjaimet ja I/O valittiin Beckhoffilta. Laittevalinnoista tehtiin ennen tilausta kaavio, jossa oli suunnitelma siitä, miten laitteet kytketään toisiinsa ja miten ne tulevat kommunikoimaan keskenään.

PLC:n logiikka ohjelmoitiin TwinCat3-ohjelmistolla. Puntarin ohjaus ja kommunikointi PLC:n kanssa toteutettiin Pythonilla käyttäen pyads- kirjastoa. Diplomityön PLC:n koodin rakenne esiteltiin ja myös pyads-kirjaston käytöstä annettiin havainollistava ja yksinkertainen esimerkki. Yksi suurimmista haasteista laitteiden kommunikoinnissa oli robotin ja PLC:n välinen kommunikointi ja turvapiirin yhdistäminen kompaktisti Ethernet-kaapelin välityksellä. Tähän käytimme KUKA:n valmistamaa siltakorttia, jolla pystyimme yhdistämään KR10:n vaatimat turvabitit turva-PLC:n bitteihin. Sama siltakortti mahdollisti I/O:n jakamisen robotin ja PLC:n välillä.

Diplomityön lopuksi tehtiin valmiilla robottisolulla muutama valmis geopolymeeri ja käsiteltiin robottisolussa ilmenneitä haasteita ja ongelmia. Suurimmat ongelmat geopolymeerien valmistuksessa olivat jauhojen erilaiset vaatimukset. Erirakenteiset jauheet vaativat syöttölaitteilta useita ominaisuuksia, kuten tärinän lisäystä tai rakenteellisia muutoksia. Jauhot myös sekoittuivat eri lailla, jolloin niille tulee ohjelmoida erilaiset sekoitusohjelmat. Toisena ongelmana oli robottisolun siisteys ja puhtaana pysyminen.

Robottisolun suunnittelu, rakentaminen ja ohjelmointi on ollut haastava projekti. Siihen sisältyi paljon mekaniikkasuunnittelua, ohjelmointia ja jonkin verran sähkökytkentöjä. Diplomityö opetti paljon mekatronisen kokonaisuuden suunnittelusta, laitteiden välisen kommunikoinnin haasteista sekä ongelmakohtien ennakoinnista.

Lopputuloksena diplomityössä päästiin tavoitteisiin suunnitella ja rakentaa geopolymeerejä valmistava robottisolu, joka hakee valmistettavien geopolymeerien reseptit annetusta Excel-taulukosta. Valmistetut geopolymeerit olivat hyviä, vaikka valmistus ei vielä onnistunutkaan kaikilla raaka-aineilla. Robottisolun kehittämistä jatketaan kuitenkin Geobot-projektin merkeissä. Tulevat tavoitteet on integroida mittauslaitteistoa ja lisätä jauhosyöttimien määrää robottisoluun.

LÄHDELUETTELO

ABB, 2019. Industrial Robots [Verkkodokumentti]. Saatavissa: <https://new.abb.com/products/robotics/industrial-robots> [Viitattu 17.6.2019]

Beckhoff Automation, 2019. Principle of operation [Verkkodokumentti]. Saatavissa: <https://www.beckhoff.com/english.asp?ethercat/aufbau.htm> [Viitattu 22.5.2019]

Beckhoff Automation, 2019. Application guide TwinSAFE [Verkkodokumentti]. Saatavissa: <https://download.beckhoff.com/download/document/automation/twinsafe/applicationguidetwinsafeen.pdf> [Viitattu 17.7.2019]. 282 s.

Bolton W., 2009. Programmable Logic Controllers. 5. Newnes: Elsevier, 400 s. ISBN 978-1-85617-751-1

Davidovits, J., 2015. Geopolymer Chemistry & Applications. 4. Saint-Quentin: Institut Géopolymère, 644 s. ISBN 978-2-951-48209-8

Degotignie J.-D., 2005. Ethernet-Based Real-Time and Industrial Communications. Proceedings of the IEEE , 93 (6), S. 1102-1117

Duxson, P., Fernández-Jiménez, A., Provis, J.L., 2007. Geopolymer technology: the current state of the art. Journal of Materials Science, 42 (9), S. 2917-2933

Felser M., 2005. Real-Time Ethernet – Industry Prospective. Proceedings of the IEEE, 93 (6), S.1118-1129

FANUC, 2019. SCARA Robots [Verkkodokumentti]. Saatavissa: <https://www.fanuc.eu/de/en/robots/robot-filter-page/scara-series/selection-support> [Viitattu 16.5.2019]

Field J., 2004. PROFINET – scalable factory communication for all applications. Teoksessa: IEEE International Workshop on Factory Communication Systems. Vienna: IEEE. S. 33-38. ISBN:0-7803-8734-1

Gunnar P., 2008. A performance analysis of EtherCAT and PROFINET IRT. Teoksessa: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg: IEEE, S. 408-415. ISBN 978-1-4244-1505-2

Harashima F., Tomizuka M. & Fukuda T., 1996. Mechatronics- 'what is it, why, and how?' An editorial. IEEE/ASME Transaction of mechatronics, 1 (1), S. 1-4

IEC 61131-3, 2003. Programmable controllers – Programming languages. 2. painos. International Electrotechnical Commission IEC: 143+ 67 s.

Illikainen S., Kinnunen P. & Yliniemi J., 2018. Geopolymeerien maailmanvalloitus. Materia, 5, S.62-65

ISO 8373, 2012, Robots and robotic devices – Vocabulary. 2. painos. International Organization for Standardization: 38 s.

Jansen D., Buttner H., 2004. Real-time ethernet the EtherCAT solution. Computing & Control Engineering Journal, 15 (1), S. 16-21

KUKA, 2019, KUKA System Software 8.3 [verkkodokumentti]. Saatavissa: <http://www.wtech.com.tw/public/download/manual/kuka/krc2ed05/Operating%20and%20Programming.pdf> [Viitattu 21.5.2019]

Luukkonen T., Abdollahnejad Z., Yliniemi J., Kinnunen P., Illikainen M., 2018, One-part alkali-activated materials: A review. Cement and Concrete Research, 103, S. 21-34

Matilainen S., 2015. Harvesteripäiden testauslaitteiston automatisointi, Oulu: Oulun yliopisto, Konetekniikka, S. 85

Nilsson K., 1996. Industrial robot programming, LUND: Lund Institute of Technology, Department of automatic control, S. 185

Oulun yliopisto, 2019. Biomimetikka ja älykkäät järjestelmät [Verkkodokumentti]. Oulu: Oulun yliopisto. Saatavissa: <https://www.oulu.fi/bisg-fi/> [Viitattu 8.2.2019]

Oulun yliopisto, 2019. Kuitu- ja partikkeliteknikka [Verkkodokumentti]. Oulu: Oulun yliopisto. Saatavissa: <https://www.oulu.fi/pyokui/> [Viitattu 8.2.2019]

Pan Z., Polden J., Larkin N., Van Duin S. & Norrish J., 2011. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing* 28 (2), S. 87-94.

Provis J. 2018. Alkali-activated materials. *Cement and Concrete Research*, 114 , S. 40-48

Pumps & Systems, 2019, Advantages of Peristaltic Pumps in Metering Applications, Saatavissa: <https://www.pumpsandsystems.com/advantages-peristaltic-pumps-metering-applications> [Viitattu 19.7.2019]

Rostan M., 2009, Industrial Ethernet Technologies, Saatavissa: http://www.ethernet.org/pdf/english/Industrial_Ethernet_Technologies.pdf [viitattu 29.5.2019]

Saukkoriipi J., 2019. Design and implementation of robot skill programming and control, Oulun yliopisto, Konetekniikka, s. 84

Siciliano B. & Khatib O., 2016. *Handbook of Robotics*. 2. Heidelberg: Springer, 2227 s. ISBN 978-3-319-32550-7

Siciliano B. & Khatib O., 2008. *Handbook of Robotics*. 1. Heidelberg: Springer, 1611 s. ISBN 978-3-540-23957-4

Spurgeon C., 2000. *Ethernet: The Definitive Guide*. 1. Sebastopol: O'Reilly Media, 501 s. ISBN 1-56592-660-9

Tunis S (2009) *Real-Time Industrial Ethernet in Machine Automation Systems*, Tampere: University of Technology, Automation Technology, S. 89.

Wallén J., 2008, *The History of the Industrial Robot*, Linköping University, Department of Electrical Engineering, S. 18

Wu C., Jiang S. & Song K., 2015. CAD-based pose estimation for random bin-picking of multiple object using a RGB-D camera, 15. Busan: International Conference on Control, Automation and Systems (ICCAS), S. 1645-1649.

Yliniemi J., Kinnunen P., Karinkanta P., Illikainen M., 2016, Utilization of Minearal Wools as Alkali-Activated Material Precursor, *Materials*, 312 (9), S. 1-12.

LIITTEET

Liite 1. Beckhoffin laitteistoluettelo

Liite 2. Muu laitteisto

Liite 3. Laitteiston kytkentäkaavio

Liite 4. Pääohjelman tilakone

Liite 5. Valmistuksen tilakone

Liite 6. Käyttöliittymä

Liite 7. Peristalttista pumppua ohjaava koodi

Beckhoff Laitesluettelo

Laitteistovalmistaja Beckhoff Automation GmbH. Tuotteiden tarkemmat tiedot löytyvät valmistajan sivuilta.

Teollisuus-PC

C6015-0010

Ultra control cabinet PC

-Proessori: Intel Atom E3845, 1.91GHz, 4 cores

-RAM: 4GB DDR3L

-Suojaus: IP 20

-Kiinnitys DIN-kiskolla

-Microsoft Windows 10 IoT Enterprise

I/O-laitteisto

EK1100 x2 *EtherCat Coupler for E-Bus-terminal*

EL1008 x3 *8-Channel Digital Input Terminal*

EL2008 x3 *8-Channel Digital Input Terminal*

EL9011 x2 *Bus End Cap*

LIITE 1 (2)

Safety-laitteisto

EL6910	x1	<i>TwinSAFE Logic Terminal (Safety-PLC)</i>
EL1904	x1	<i>4-Channel Digital Input terminal, TwinSafe</i>
EL2904	x1	<i>4-Channel Digital Output terminal, TwinSafe</i>

Askelmoottorit ja ohjaimet

AS2021-0D10	x4	<i>Stepper motor</i>
EL7037	x8	<i>Stepper motor terminal</i>

Servomoottorit ja servo-ohjaimet

AM8031-1C21-000	x2	<i>Servomotor</i>
AX8620-0000	x1	<i>Power supply module</i>
AX8206-0100	x2	<i>Double-axis module (Servo drive)</i>

Muu Laitteisto

Manipulaattori

KR 10 R1100-2

KUKA, 10kg

Virtalähteet

BAE0002

BALLUFF, 24 VDC / 10 A

SPD1260

CARLO GAVAZZI, 12VDC / 5A

Nestepumput

114ST

Watson-Marlow, Peristaltic pump

LPGA1251618D

The Lee Company, Liquid Pump

Lineaariyksikkö

RHL80

Rollco, Lineaariyksikkö

Puntari

AG4000

*Torbal, puntari***Tarttujat**

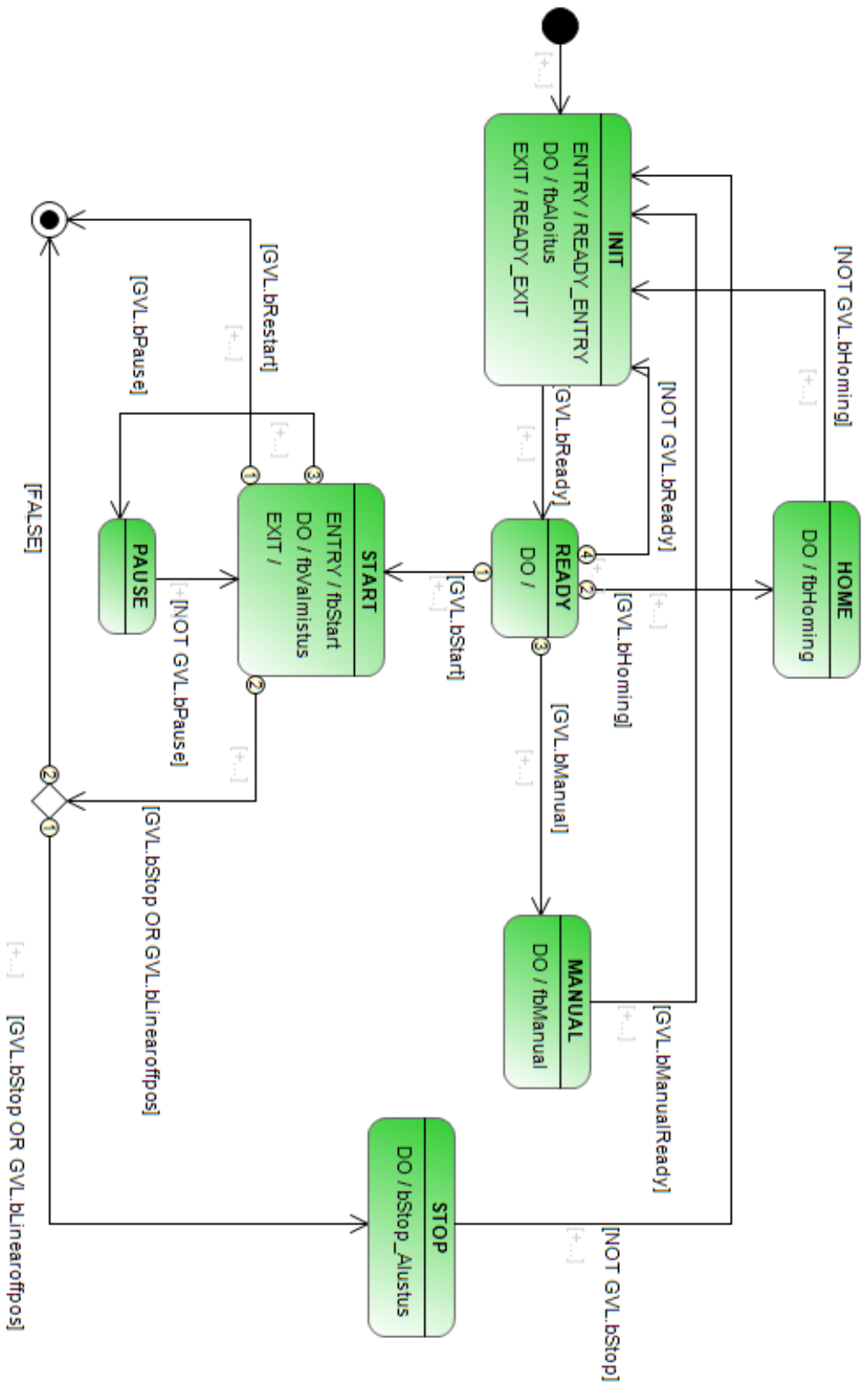
MHZ2-25S

SMC, paineilma tarttuja

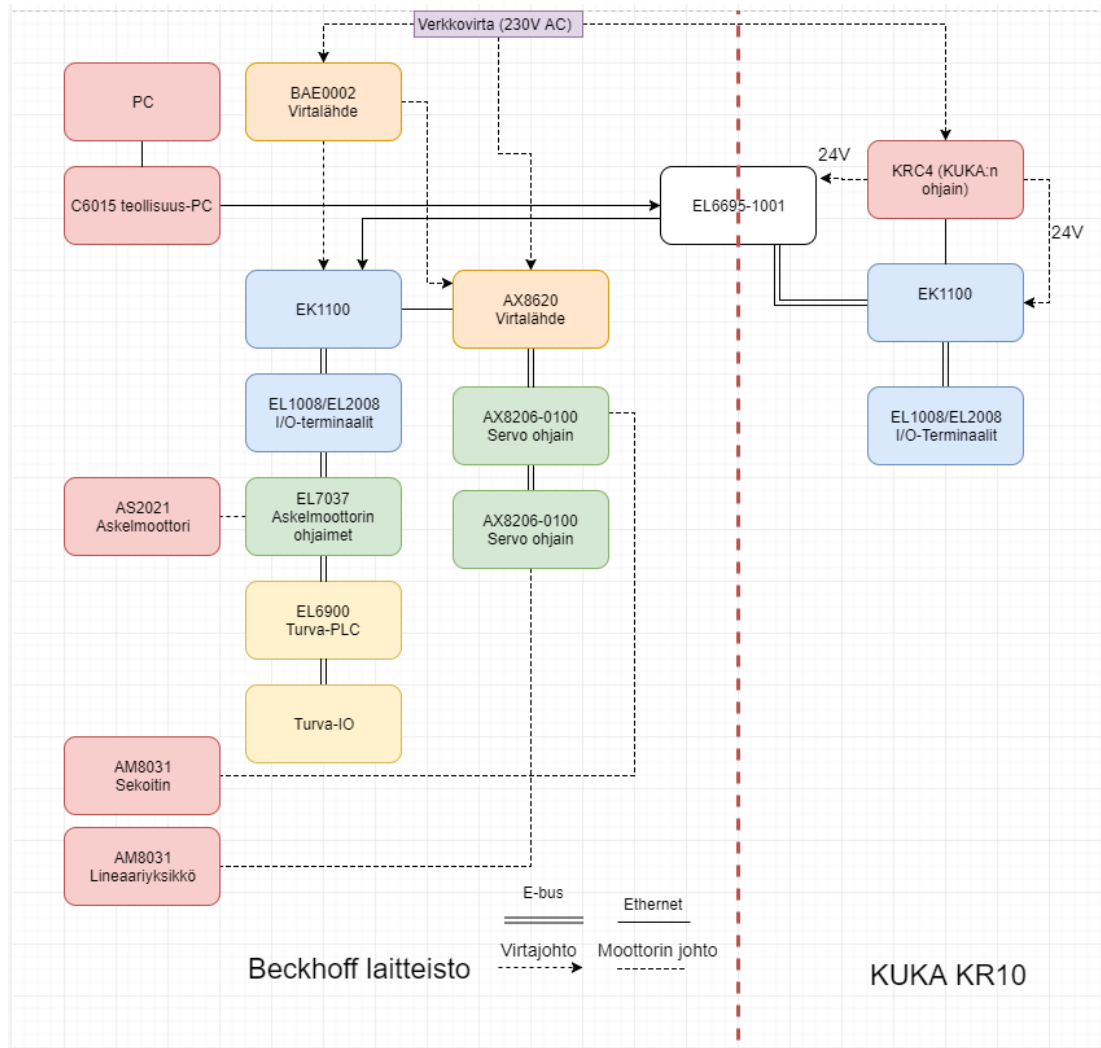
MGQM 16-10

SMC, paineilma sylinteri

LIITE 3



LIITE 5



LIITE 6

PAUSE START HOME

READY STOP RESTART

TARE

Parameters

Set mass of the liquid
% 2f g

Set mass of the powder
% 2f g

Set mixers speed of rotation
% 2f rpm

MANUAL

% 2f mm

Mixer

Clean the tube

Powder dose
% 2f

Lnmass

Smass

sample

Visualization

%S

Scale: % 2f g



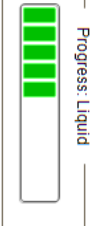
Torque of mixer: % 2f Ncm

Progress: Liquid

Progress: Powder

% 2f rpm

% 2f mm



LIITE 7 (1)

```

1  GVL.AxisE.ReadStatus();
2
3  McPower(   Axis := GVL.AxisE,
4            Enable := TRUE,
5            Enable_Negative := TRUE,
6            Enable_Positive := TRUE);
7
8  CASE iState OF
9    0: //Initializing
10     GVL.bTare := TRUE;
11     rCount := 0;
12     bFix := FALSE;
13     iState := 10;
14     fbTare(IN:=FALSE);
15
16    10: //Start Drive reset
17     IF NOT GVL.bTare THEN
18         McReset(Axis:=GVL.AxisE, Execute:=TRUE);
19         IF McReset.Done THEN
20             McReset(Axis:=GVL.AxisE, Execute:=FALSE);
21             iState := 20;
22         ELSIF McReset.Error THEN
23             iState := 999;
24         END_IF
25     END_IF
26
27    20: //Wait for Drive to Reset
28     IF McPower.Status THEN
29         iState := 30;
30     END_IF
31
32    30: // Move command
33     GVL.sStatus := 'Pumping Liquid';
34
35     IF bFix THEN
36         rLiquid := rLmassFix/0.25;
37         rDistance := -120 * (rLiquid);
38     ELSE
39         rLiquid := GVL.iLmass1/0.25;
40         rDistance := -120 * rLiquid;
41     END_IF
42     McRelative(Axis := GVL.AxisE,
43               Acceleration := 1440,
44               Deceleration := 1440,
45               Velocity := 720,
46               Jerk := 2250,
47               Distance := rDistance,
48               Execute := TRUE);
49     IF McRelative.Done AND NOT McRelative.Error THEN
50         McRelative(Axis := GVL.AxisE, Execute := FALSE);
51         iState := 50;
52
53     ELSIF McRelative.Error THEN
54         iState := 999;
55     END_IF
56

```

LIITE 7 (2)

```

57      50:
58      bClock3 := TRUE;
59      fbPuntariTimer(IN:=bClock3, PT:=T#8S);
60      GVL.sStatus := 'Checkin the weight of liquid';
61
62      IF fbPuntariTimer.Q THEN
63
64          GVL.iTruemass := ((DINT_TO_REAL(GVL.iPythonTruemass))/100);
65          GVL.iLmass := ((DINT_TO_REAL(GVL.iPythonLmass))/100);
66          GVL.iLmass1 := ((DINT_TO_REAL(GVL.iPythonLmass1))/100);
67          GVL.iSmass1 := ((DINT_TO_REAL(GVL.iPythonSmass1))/100);
68          GVL.iLmass2 := ((DINT_TO_REAL(GVL.iPythonLmass2))/100);
69          GVL.iSmass2 := ((DINT_TO_REAL(GVL.iPythonSmass2))/100);
70
71          IF GVL.iLmass1 - GVL.iTruemass >= 0.15 THEN
72              rLmassFix := GVL.iLmass1 - GVL.iTruemass;
73              bFix := TRUE;
74              bClock3 := FALSE;
75              fbPuntariTimer(IN:=FALSE);
76              iState := 30;
77          ELSE
78
79              bFix := FALSE;
80              bClock3 := FALSE;
81              GVL.iTruemass := 0;
82              fbPuntariTimer(IN:=FALSE);
83              rLiquid := 0;
84              rLmassFix := 0;
85              GVL.bLmass1Ready := TRUE;
86              iState :=0;
87              GVL.bPLC1 := TRUE;
88          END_IF
89      END_IF
90
91      999: //Error handle
92      GVL.sStatus := 'Error on AxisE';
93      GVL.bStop := TRUE;
94      IF McRelative.Error THEN
95          GVL.iError := McRelative.ErrorID;
96      ELSIF McReset.Error THEN
97          GVL.iError := McReset.ErrorID;
98      ELSIF McPower.Error THEN
99          GVL.iError := McPower.ErrorID;
100     END_IF
101 END_CASE

```