



jamk

Yhteistyörobotiikan tiedonkeruujärjestelmän kehittäminen

Jukka Oinonen

Opinnäytetyö, ylempi AMK
Huhtikuu 2024
Robotiikka YAMK

Oinonen Jukka

Yhteistyörobotiikan tiedonkeruujärjestelmän kehittäminen

Jyväskylä: Jyväskylän ammattikorkeakoulu. Huhtikuu 2024, 77 sivua

Robotiikan ylempi AMK tutkinto-ohjelma. Opinnäytetyö YAMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

Tiivistelmä

Teknologian kiihtyvä kehitys on aiheuttanut merkittäviä muutoksia teollisuudessa. Antureiden ja robottien avulla kerättyä informaatiota voidaan entistä yksityiskohtaisemmin tallentaa tietovarastoihin, josta sitä voidaan käyttää valvontaan ja tuotannon ohjaukseen. Yhteistyörobotit sisältävät älyä ja kykenevät mittaamaan yksityiskohtaista tietoa niiden suorittamista työtehtävistä, mutta useinkaan kaikkea hyödyllistä informaatiota ei kerätä talteen.

Työssä kartoitettiin, kuinka työn toimeksiantajan yhteistyöroboteilta voidaan kerätä tietoa niiden suorittamista tuotetestauksen työtehtävistä, kuinka tieto siirretään roboteilta tietovarastoon ja kuinka varastoitua tietoa hyödynnetään monitoroinnissa ja raportoinnissa. Työ tarjoaa yleistä tietoa robotiikasta, kuinka roboteilta voidaan kerätä dataa ja mitä hyötyä datan keräämisestä on. Työn liitteenä on Universal Robots yhteistyöroboteille kehitetty Python-ohjelma, mikä mahdollistaa yksinkertaisen monitorointinäkökulman aikaan saamisen ja antaa hyvän lähtökohdan oman tiedonkeruujärjestelmän kehittämiseksi.

Työ toteutettiin tutkimuksellisena kehittämistyönä. Työssä selvitettiin tutkimuksellisesta näkökulmasta vaihtoehtoja tiedon siirtämiseen, tallentamiseen ja jatkokäsittelyyn. Kerättyä tutkimusaineistoa hyödyntäen työssä arvioidaan, millä keinoin työn toimeksiantajan käyttämät yhteistyörobotit voivat kerätä ja siirtää dataa, kuinka datan talteenotto voidaan toteuttaa ja kuinka kerättyä dataa voidaan hyödyntää.

Työn lopputuloksena oli toimiva tiedonkeruujärjestelmä, jonka avulla yhteistyöroboteilta kyetään keräämään yksityiskohtaista mittadataa. Työssä aikaan saadun tiedonkeruujärjestelmän avulla kyetään luomaan visuaalisia ja helppolukuisia raportteja yhteistyörobottien toiminnasta. Työn aikana luotiin myös valvontanäkökulma, jonka avulla yhteistyörobottien tilaa voi seurata etänä. Tiedonkeruujärjestelmä toteutettiin siten, että sitä on helppo muokata ja laajentaa muuttuvien tarpeiden mukaan.

Avainsanat (asiasanat)

robotiikka, yhteistyörobotiikka, tuotetestaus, tiedonsiirtorajapinta, tietovarastot

Muut tiedot (salassa pidettävät liitteet)

Oinonen Jukka

Development of a collaborative robotics data collection system

Jyväskylä: JAMK University of Applied Sciences, April 2020, 77 pages.

Master's Degree in Robotics. Master's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The rapid advancement in technology has led to significant changes in industry. Information gathered with sensors and robots can be stored in data warehouses in greater detail than before, from where it can be used to monitor and control production. Collaborative robots have intelligence and are able to measure detailed information about the work tasks they perform, but rarely all useful information is collected.

The goal of the thesis was to find out how information can be collected from collaboration robots about the product testing tasks they perform, how the collected information can be transferred from the robots to a data warehouse and how the stored data can be utilized for monitoring and reporting. The thesis provides general information about robotics, how data can be collected from robots and how data collecting can be beneficial. An attachment contains a Python program developed for Universal Robots collaborative robots, which enables the creation of a simple monitoring view and provides a good starting point for developing a custom data collection system.

The work was conducted as a research-based development study. Different ways how to transfer data, store and process it are explored. Based on the collected research material the best ways to transfer and utilize data from collaborative robots of the company that commissioned the thesis are evaluated.

The end result of the thesis was a functional data collection system, which enables detailed measurement data to be collected from collaborative robots. The data collection system is used to create visual and easy-to-read reports on the operation of collaborative robots. A monitoring view was also created to allow remote monitoring the collaborative robots. The data collection system was implemented in such a way that it is easy to be modified and expanded based on future needs.

Keywords/tags (subjects)

robotics, collaborative robotics, product testing, communication interface, data warehouses

Miscellaneous (Confidential information)

Sisältö

1	Johdanto	4
1.1	Opinnäytetyön tavoitteet.....	4
1.2	Tutkimusmenetelmät.....	5
1.3	Abloy Oy	6
2	Robotiikka	7
2.1	Robotiikka Suomessa	8
2.2	Robotiikan mahdollisuudet Suomessa	11
3	Teollisuuden robottityypit	12
3.1	Nivelvarsirobotit.....	13
3.2	SCARA-robotit.....	14
3.3	Delta-robotit.....	14
3.4	Kartesiset robotit.....	15
3.5	Napakoordinaatistorobotit	16
3.6	Sylinterirobotit	17
4	Yhteistyörobotit	18
5	Teolliset vallankumoukset	21
6	Älykkäät järjestelmät	24
6.1	Kohti älykästä tehdasta	25
6.2	Signaaleista kenttäväyliin.....	28
7	Tietovarastot	31
7.1	Paikalliset tekstipohjaiset tiedostot	32
7.2	SQL-tietokannat.....	34
7.3	NoSQL-tietokannat.....	35
7.4	Pilvitietokannat	37
7.5	Apache Hadoop	38
8	Universal Robots yhteistyörobotin tiedonsiirto	38
8.1	Modbus	39
8.2	EtherNet/IP.....	39
8.3	PROFINET ja PROFIsafe.....	40
8.4	Muut Universal Robots -robottien kommunikointirajapinnat.....	41
9	Tuotetestaus	44
9.1	Yhteistyörobotilla suoritettava mekatronisen sylinterin kulutustesti	45
9.2	Yhteistyörobottien turvallisuus tuotetestauksessa	48

9.3	Yhteistyörobotilta talteen kerättävät tiedot.....	49
10	Tiedonkeruujärjestelmän toteutus.....	52
10.1	Virtuaalisen yhteistyörobotin RTDE-rajapintaan yhdistäminen Python-ohjelmointikielellä 53	
10.2	Julkiseen käyttöön valmistettu Python-ohjelma.....	55
10.3	Yhdistäminen todellisiin yhteistyörobotteihin.....	56
10.4	SQL-integraatio.....	58
11	Lopputulokset ja pohdinta	61
11.1	Jatkokehitys.....	62
11.2	Luotettavuus ja eettisyys	63
Lähteet	65
Liitteet	70
	Liite 1. Python-koodi UR5e edestakaiseen RTDE-kommunikointiin	70
	Liite 2. Tietokantaan kerätyn datan perusteella muodostettu testausraportti.....	72
Kuviot		
	Kuvio 1. Tutkimuksen ja kehittämistoiminnan risteys (Toikko & Rantanen 2009, 21).....	6
	Kuvio 2. ABLOY PROTEC ² lukkopesän sisällä olevia haittalevyjä (Abloy yleisesitys 2023, muokattu)	7
	Kuvio 3. Kaavio maiden robottitiheydestä vuonna 2021 (Graph Robot Density by country 2021 2022)	9
	Kuvio 4. Nivelvarsirobotin pääosat (Billing 2023, 118)	13
	Kuvio 5. Omron i4H-sarjan SCARA-robotti (Omron RS4-2066502 n.d.)	14
	Kuvio 6. ABB IRB 390 FlexPacker deltarobotti (ABB 2022)	15
	Kuvio 7. Feston karteeminen robotti YXML (Linear gantry YXML n.d.).....	16
	Kuvio 8. Unimation Inc. valmistama Unimate (Robot, First Unimate Robot Ever Installed on an Assembly Line, 1961 n.d.)	17
	Kuvio 9. Hudson Roboticsin laboratoriotyöskentelyyn kehitetty sylinterirobotti (SciClops Microplate Handler n.d.).....	18
	Kuvio 10. Erilaisia yhteistyörobottien rakenteita (CR-14iA/L n.d.; PreciseFlex 400 n.d.; Wyzo sidebot n.d.; PreciseFlex 100 2022; muokattu)	19
	Kuvio 11. Yhteistyörobotin viisi eri yhteistyöskentelytapaa (Bauer ym. 2016, 9, muokattu)	20
	Kuvio 12. Teollisen vallankumoukset (Collins 2016, 39).....	21
	Kuvio 13. Teollisen vallankumouksen kolme aaltoa (Evans & Annunziata 2012, 7)	23
	Kuvio 14. Kondrajettffin aaltoteorian kuusi aaltoa (Collins 2016, 41)	24
	Kuvio 15. Tiedon hyödyntäminen kasvattaa sen arvoa (André 2019, 52).....	26

Kuvio 16. 5C-arkkitehtuurin pyramidimallissa teollisen internetin infrastruktuuri kehittyä yrityksen noustessa pyramidia ylöspäin (Collins 2016, 146)	28
Kuvio 17. Tiedonsiirtoliitynnällä toteutettavia toimintoja (Liuha & Aro 2023, 148)	29
Kuvio 18. Teollisen internetin tekniset osakokonaisuudet (Collin & Saarelainen 2016, 143)....	30
Kuvio 19. Tietokannat järjestettynä suosion mukaan. Suosio laskettu internetistä tietokannan nimen käytön mukaan mm. hakukoneiden hakujen määrästä. (C# Corner).....	31
Kuvio 20. Havainnollistava esimerkki kolmesta SQL-tietokannan taulusta.....	35
Kuvio 21. Neljä yleisintä NoSQL tyyppiä (Barooah n.d., muokattu)	37
Kuvio 22. Yleiskatsaus UR-yhteistyörobotin tiedonsiirtorajapinnoista (Overview of client interfaces n.d.)	42
Kuvio 23. Mekatronisten lukitustuotteiden kulutustestilaitteen mallikuva (SFS-EN 15684:2020, 23)	46
Kuvio 24. Mekatronisen sylinterin kulutustestin toimenpiteet	47
Kuvio 25. UR5e yhteistyörobotti suorittamassa lukkosylinterin kulutustestiä	48
Kuvio 26. UR5e yhteistyörobotin ohjauspaneelin kautta nähtävä Force Copilot-ikkuna.....	50
Kuvio 27. Kommunikointirajapinnan testaus virtuaaliympäristössä	54
Kuvio 28. Päivitetty Python ohjelmalla kykeni ottamaan yhteyden kahteen eri RTDE-rajapintaan	54
Kuvio 29. RTDE-kommunikoinnin muuttujamäärittelyt, eli rajapinnan resepti	55
Kuvio 30. UR5e -robotin ohjelma laittaa muuttujien arvoja luettavaksi RTDE-rajapintaan.....	56
Kuvio 31. Palvelimella suoritettu Python-ohjelma yhdistettynä oikeisiin robotteihin.....	57
Kuvio 32. Näkymä SQL-tietokannan tauluista ja niissä olevasta datasta	60

Taulukot

Taulukko 1. Esimerkki CSV-tiedoston sisällöstä tekstimuodossa ja taulukkona.....	32
Taulukko 2. Esimerkki JSON-tiedoston sisällöstä tekstimuodossa	33
Taulukko 3. Lukitusjärjestelmiin liittyviä standardeja vaatimuksista ja testausmenetelmistä ..	45
Taulukko 4. UR5e yhteistyöroboteille määritetyt muuttujat sylinterituotteiden testauksen informaation keräämiseksi	51
Taulukko 5. CobotStatus-taulun rakenne	58
Taulukko 6. CobotData-taulun rakenne.....	59

1 Johdanto

Teknologiakehityksen aikaansaamien uusien ja kustannustehokkaiden antureiden ja robottien tuoma kilpailuetu on saanut aikaan suuria muutoksia teollisuudessa. Robottien ja automaattien avulla tuotannon jäljestä saadaan tasaista ja ne mahdollistavat ympärivuorokautisen tuotannon entistä vähemmällä henkilöstömäärällä. Tarkkojen antureiden käyttö laadun tarkistuksessa mahdollistaa tuotteiden korkean laadun varmistamisen. Laitteiden keräämää informaatiota voidaan tallentaa entistä suuremmissa määrin ja käyttää sitä hyödyksi tuotannon säätämisessä ja ohjauksessa. Tapahtuvaa kehitystä kutsutaan nimellä neljäs teollinen vallankumous, Teollisuus 4.0:ksi.

Robottien ja antureiden kaltaisten älykkäiden laitteiden toiminnasta voidaan saada yritystoiminnan kannalta myös enemmän irti kuin pelkkä tieto, mikä laitteiden tämänhetkinen tila on. Tuotantotiedon tallentamisella helposti käsiteltävään muotoon mahdollistaa historiatiedon hakemisen, jolloin tuotantoerien tiedot voidaan hakea myös jälkikäteen. Fyysisten laitteiden kehityksen lisäksi kiinnostavaa muutosta tuo myös ohjelmalliset kehitykset. Historiatiedon ja antureiden mittaamien reaaliaikaisen tiedon pohjalta, älykäs tuotannon ohjaus voi pitää tuotannon entistäkin kustannustehokkaampana. Ennen kuin tällaisia älykkäitä tehtaita voidaan saada aikaiseksi, täytyy älykkäät laitteet, kuten robotit, yhdistää jonkinlaiseen tiedonkeruu- ja tallennusjärjestelmään.

Tämän opinnäytetyön toimeksiantaja on maailmanlaajuisesti toimiva Abloy Oy. Työn tavoitteena oli tutkia Abloy Oy:n Joensuun tehtaalla toimivalle tuotetestausyksikölle yhteistyörobottien tiedonsiirtomenetelmiä ja kehittää järjestelmä, jolla voitaisiin kerätä dataa niin roboteista kuin testattavista tuotteista. Kerättyä dataa haluttiin käyttää hyödyksi tuotetestien raportoinnissa. Työssä lisäksi tutustuttiin yleisellä tasolla robotiikan kehitykseen ja yleisimpiin teollisuuskäyttöön tarkoitettuihin robottityyppeihin. Erityisesti työssä haluttiin perehtyä Universal Robots UR5e -yhteistyörobottimallin tiedonkeruu- ja siirtomenetelmiin, sillä sen mallisia robotteja Abloy Oy:n tuotetestauksessa oli jo käytössä kuusi kappaletta.

1.1 Opinnäytetyön tavoitteet

Abloy Oy:n tuotetestaus suorittaa yrityksen valmistamille tuotteille erilaisia laatutestejä, joiden tarkoitus on varmistaa, että tuotteet saavuttavat vaatimusten mukaisten korkean laadun ja kestävyden. Yhteistyörobotteja käytetään muun muassa suorittamaan pitkäkestoisia kulutustestejä,

joissa testataan kuinka tuotteet kestävät suuria määriä käyttökertoja. Opinnäytetyön tavoitteena on saada aikaan järjestelmä, jonka avulla yhteistyörobottien keräämää tietoa hyödyntäen voidaan tehdä tuotetestausta hyödyttäviä raportteja. Työn tavoitteena on kartoittaa, kuinka yhteistyöroboteilta voidaan kerätä tietoa, kuinka kerätty tieto voidaan varastoida ja kuinka varastoitua tietoa voidaan hyödyntää. Työn tavoitteet voidaan tiivistää seuraaviin tutkimuskysymyksiin:

1. Kuinka UR5e yhteistyöroboteilta suoritettavista tuotetesteistä voidaan kerätä tietoa?
2. Kuinka kerätty tieto voidaan siirtää roboteilta tietovarastoon?
3. Kuinka varastoitua tietoa voidaan hyödyntää tuotetestien raporteissa?

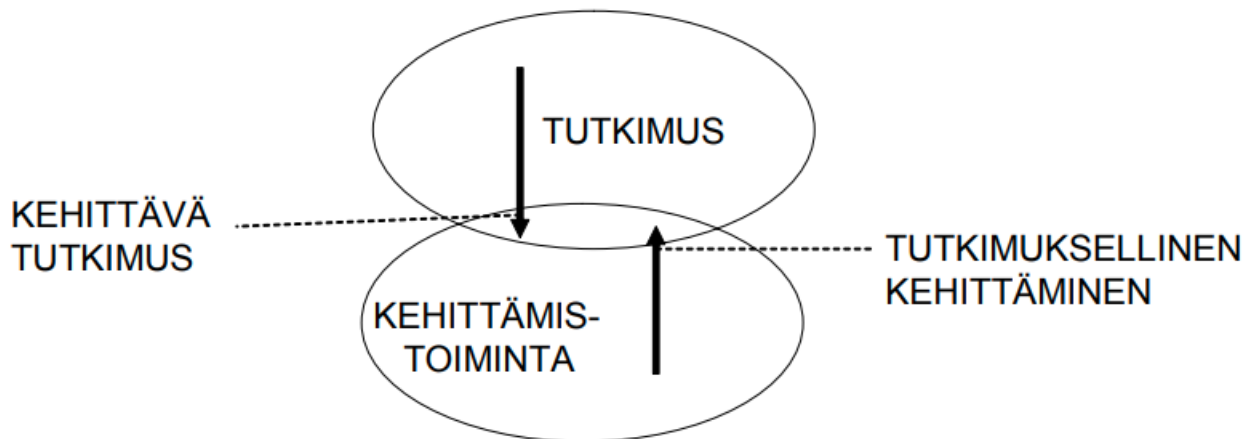
Tuotetestauksen yhteistyörobotteja käytetään monipuolisesti erilaisiin testeihin, mutta opinnäytetyöraportti rajattiin sisältämään käyttökohteena ainoastaan mekatronisten sylinterien standardin SFS-EN 15684:2020 mukaisten tuotteiden kulutustestit. Työn tavoitteena oli luoda järjestelmä, jolla saadaan yksityiskohtaista tietoa testattavien tuotteiden käyttökertojen lisääntyessä tapahtuvista tuotteiden ominaisuuksien muutoksista. Lisäinformaatio tuotteiden toimintaominaisuuksista koettiin olevan tärkeää tietoa tuotannon ja tuotekehityksen parantamiseksi. Tuotetestauksen kehittämiseksi myös itse yhteistyörobottien toiminnasta haluttiin saada lisää tietoa, kuten kuinka paljon toistoja ne kykenevät tekemään vuorokaudessa.

Opinnäytetyössä selvitetään tutkimuksellisesta näkökulmasta, minkälaisia vaihtoehtoja tiedon siirtämiseen, tallentamiseen ja jatkokäsittelyyn on. Kerättyä tutkimusaineistoa hyödyntäen työssä arvioidaan, millä keinoin toimeksiantajan käyttämät UR5e yhteistyörobotit voivat kerätä ja siirtää dataa, kuinka datan talteenotto voidaan toteuttaa ja kuinka kerättyä data voidaan hyödyntää Abloy Oy:n tuotetestauksen näkökulmasta. Työn käytännöllisessä osuudessa toteutetaan Universal Robots -yhteistyörobottien tiedonsiirtorajapintaa käyttävä ohjelma, jonka avulla robottien keräämä data tallennetaan tietovarastoon helposti hyödynnettävään muotoon, mitä puolestaan hyödynnetään entistä kattavampien tuotetestiraporttien luomisessa.

1.2 Tutkimusmenetelmät

Työ aloitetaan tutkimalla ja vertailemalla erilaisia lähestymistapoja tiedonkeruun rakentamiseen ja tietopohjaa käyttäen suoritetaan konkreettinen kehittämistyö. Toikko ja Rantanen (2009) esittä-

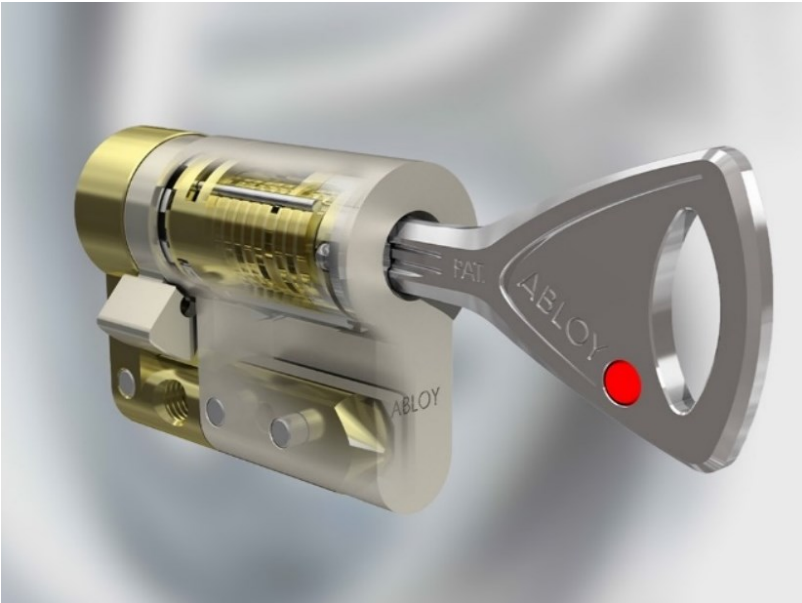
vät, että tutkimuksen ja kehitystoiminnan rajamaastossa olevaa kehittämistä voidaan lähestyä kuvion 1 mukaisesti joko tutkimuksen suunnasta tai kehittämistoiminnan suunnasta (Toikko & Rantanen 2009, 21–22). Tämän opinnäytetyön toteutuksen tavoitteena on kehittäminen tieteelliseen tietoon pohjautuen. Tutkimusta lähestytään kehittämistoiminnan näkökulmasta, joten työn tutkimusmenetelmää voidaan kuvailla tutkimukselliseksi kehittämiseksi (mts. 33).



Kuvio 1. Tutkimuksen ja kehitystoiminnan risteys (Toikko & Rantanen 2009, 21)

1.3 Abloy Oy

Helsingiläinen hienomekaanikko Emil Henriksson sai kassakoneen toiminnasta idean ja kehitti vuonna 1907 kiertyviin haittalevyihin perustuvan lukon. Keksintö sai patentin vuonna 1919 ja lukkojen tuotanto aloitettiin Helsinkiin perustetussa Ab Låsfabriken-Lukkotehdas Oy yhtiössä. Pitkä nimi myöhemmin lyheni muotoon Ab Lukko Oy ja yrityksen nimestä juonnettiin tavaramerkki ABLOY. Vuonna 1968 Joensuuhun perustettiin lukkotehdas, jossa tuotanto ja kehitystyö vieläkin jatkuvat. Abloy sulautui ruotsalaisen pääkilpailijan Assan kanssa vuonna 1994, josta muodostui konserni ASSA ABLOY. Abloy Oy:n lukot perustuvat edelleen haittalevyjen periaatteeseen, kuten esimerkiksi vuoteen 2031 asti patentoitu ABLOY PROTEC² (ks. kuvio 2). (Abloyn tarina n.d.)



Kuvio 2. ABLOY PROTEC² lukkopesän sisällä olevia haittalevyjä (Abloy yleisesitys 2023, muokattu)

Abloy Oy työllistää Suomen sisällä noin 730 työntekijää, joista suurin osa työskentelee Joensuussa, jossa ABLOY-tuotteiden fyysinen tuotanto tapahtuu. Suomen ulkopuolella Abloy Oy työllistää noin 200 ihmistä. Myyntiyksikköjä toimii noin kymmenessä maassa ja tuotteita toimitetaan yli 90 maahan. Perinteisten mekaanisten ratkaisujen, kuten lukkorunkojen, lukkosylinterien, riippulukkojen, ovensulkimien ja rakennushelojen lisäksi Abloy Oy valmistaa elektromeaanisia tuotteita, kuten sähkölukkoja, lukijoita ja oviautomaatiikkaa. Lisäksi tarjolla on myös ohjelmistoratkaisuja, kuten kulunvalvonta- ja hallintajärjestelmiä, mobiiliavaimia ja työajan seurantajärjestelmiä. Vuonna 2023 Joensuun tehtaalla robotteja oli 46 kappaletta, joita pääosin käytettiin tuotannossa. Kuusi robottia toimi yksinomaan tuotetestaustehtävissä. (Abloy yleisesitys 2023.)

2 Robottiikka

Sana ”robotti” on alun perin muodostunut tšekkiläisestä sanasta ”robotnik”, joka on tarkoittanut palkatonta työvoimaa. Sana nykyisessä merkityksessään tuli tunnetuksi vuonna 1920 Karel Čapekin kirjoittamasta näytelmästä R.U.R.: Rossum’s Universal Robots, jossa tunteettomat, mutta kuitenkin itsenäiset, robot-olennot luodaan tekemään töitä ihmisten sijasta (Hilka-Keinänen 2018). Robotti nykypäivänä on yleiskäyttöinen kone, joilla voidaan ajatella olevan joitakin ihmisenkaltaisia ominaisuuksia, kuten kyky ottaa käskyjä vastaan, tehdä päätöksiä havaintoelimien (antureiden)

perusteella ja mukautua eri työtehtäviin (Groover 2015, 221). Jopa robotin liikkuvasta osasta, joka tekee fyysisen työn, käytetään nimitystä käsivarsi.

Robotti itsessään ei kuitenkaan kykene edes yksinkertaisiin työtehtäviin, vaikka ohjausjärjestelmä olisi paikoillaan. Robotti tulee osaksi robottijärjestelmää, josta lopulta muodostuu toimiva kone. Vaikka järjestelmässä ei näennäisesti ensisilmäykseltä olisi mitään muuta kuin robotin liikkuva käsivarsi, on siihen kuitenkin lisätty työkalu, ohjausjärjestelmä, turvalaitteisto ja usein myös oheislaitteita, kuten antureita. (Lempiäinen 2023 a, 17–19).

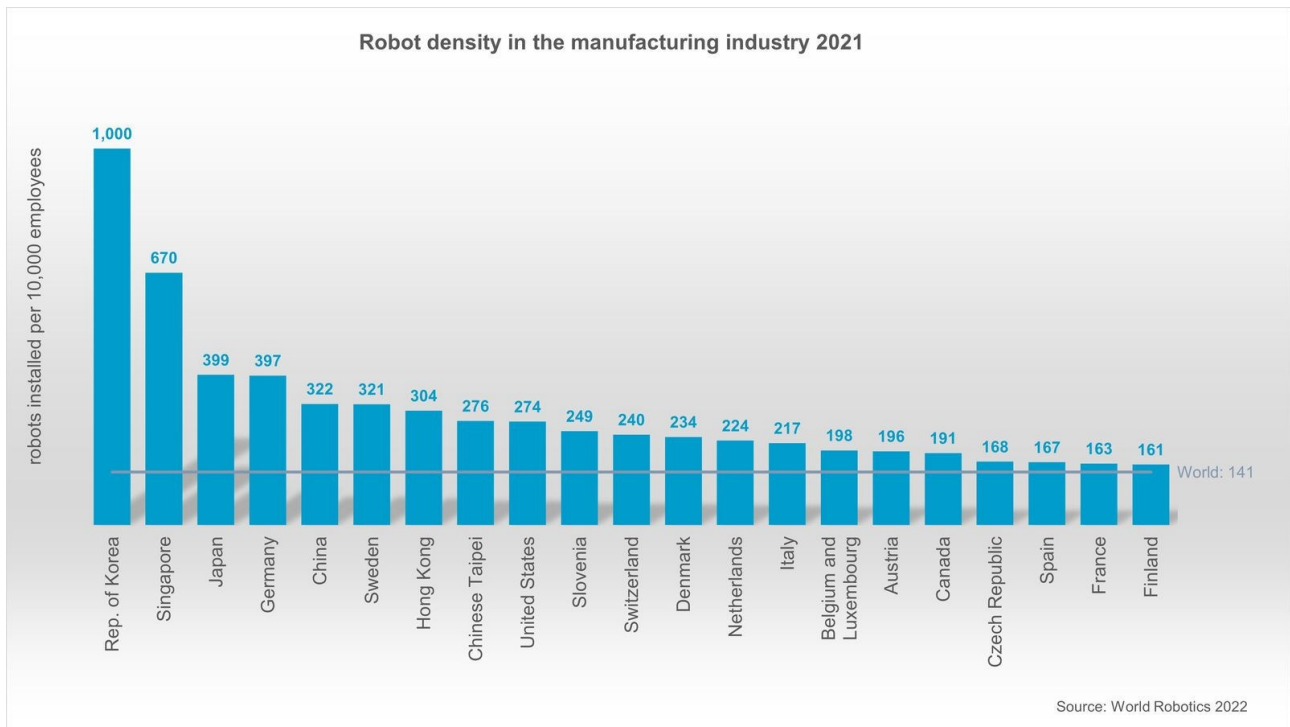
Ensimmäisten teolliseen käyttöön tarkoitettujen robottien yleistyessä niitä käytettiin niin sanottuihin kolmen D:n sovelluksiin: dull, dirty and dangerous, eli toistuviin, likaisiin ja vaarallisiin työtehtäviin. Myöhemmin robottien käyttö on laajentunut tuotantokustannusten vähentämiseksi korvaamalla perinteistä henkilöstön tekemää tuotantoa ja tekemään tuotannon laaduntarkkailua. Robotteja käytetään erityisesti pintakäsittelyn ja puhdistilaoiloihin töiden vaativuuden ja tarkkuuden vaatimuksen vuoksi (mts. 17). Nopeatoimisella robotiikalla voidaan myös saada aikaiseksi tuotettua entistä suuria tuotantoeriä. Robotiikan kehityttyä suoriutumaan hyvin yksityiskohtaisista työtehtävistä, voidaan D-kirjainten listaan lisätä sana delicate, eli hienovarainen. (Mts. 20.)

Robotisoinnin yleistymisen johtuu myös nousevista työvoimakustannuksista sekä ammattitaitoisen työvoiman pulasta. Robotiikan avulla saadaan vähennettyä vähiten houkuttelevaksi koettuja monotonisia työvaiheita ja lisättyä ylläpito- ja operointitehtäviä. Jopa pienten ja keskikokoisten yritysten kohdalla robotiikasta on tulossa välttämättömyys, koska asiakkaat odottavat tasa- ja korkealaatuisia tuotteita, sekä kykyä joustavuuteen, tuotevariaatioihin ja tarpeen tullen pieniin eräkokoihin. (Lehtinen 2023, 29.)

2.1 Robotiikka Suomessa

Ensimmäisen kerran Suomessa käytettiin teollisuusrobottia vuonna 1971 Vaasassa Strömberg-yhtiön tehtaalla käsittelemään painevalukoneen kuumia kappaleita. Samana vuonna myös Porissa Rosenlew-yhtiö otti käyttöön sen aikaisten televisioiden isojen kuvaputkien käsittelyä varten paineilmatomisia robotteja. Robotiikan alkuvaiheista asti Suomessa teollisuus on siis etsinyt roboteille käyttökohteita kilpailukyvyn kasvattamiseksi. Maailmassa yhteensä on arvioitu olevan käytössä 3,7 miljoonaa teollisuusrobottia ja suurinta osaa näistä työllistää elektroniikkateollisuus,

toiseksi eniten autoteollisuus ja sitten vasta konepajateollisuus. (Lempiäinen 2023 a, 20–25.) Kuviossa 3 on kansainvälisen robotiikkajärjestön International Federation of Robotics (IFR) julkaisema tilasto eri maiden robottien määrän suhteesta työntekijöihin vuonna 2021. Tuoreemmassa julkaisemassa vuoden 2022 tilastojen mukaan Suomessa on 168 robottia kymmentä tuhatta teollisuustyöntekijää kohti, joten robottitiheys on lähtenyt odotettuun kasvuun, vaikkakin hyvin maltillisesti.



Kuvio 3. Kaavio maiden robottitiheydestä vuonna 2021 (Graph Robot Density by country 2021 2022)

Tieteessä Tapahtuu (2015) -lehdessä kerrotaan Suomen robotiikkatutkimuksen keskittyvän perinteisen kokoonpanolinjojen sijaan erityisesti satama-, metsä- ja kaivosteollisuuden automatisointiin. Suomessa on kehitetty ilman kuljettajaa kulkevia koneita ja lisäkehittävää on erityisesti talviolosuhteiden vuoksi. (Lehtinen 2015, 42.) Automaatioväylä (2023) -lehdessä Suomen Robotiikkayhdistyksen puheenjohtaja Jyrki Latokartano kirjoittaa Suomessa robotiikan lähteneen 2010-luvun hiljaiselon jälkeen vihdoin kasvuun, ja asennettujen teollisuusrobottien määrä kasvoi 9 % edellisestä vuodesta. Suomessa robotit ovat merkittävässä roolissa raskaissa teollisuusroboteilla tehtävillä kaarihitsauskoneissa ja kevyemmissä sekä nopeammin käyttöönotettavissa yhteistyöroboteilla tehtävillä hitsaussoluissakin. (Latokartano 2023, 21.)

Samassa lehdessä Juhani Lempiäinen (2023 b), myös Suomen Robotiikkayhdistyksen hallituksen jäsen, kirjoittaa koko 2010-luvun olevan kansantaloudessa menetetty vuosikymmen, sillä Suomen teollisuus ei uudistunut merkittävästi ja investoinnit tuotantoon olivat liian alhaiset. 2020-luvulla näkymä on ollut positiivisempi, mutta koronapandemian aiheuttamat toimitukselliset ja tuotannolliset epävarmuudet ovat aiheuttaneet uutta epävarmuutta investointeihin. Lempiäisen mielestä robotiikalla pitäisi olla valtava kysyntä niin vihreän siirtymän vaatimilla uusilla tuotantoprosessien, autoteollisuuden uudistumisella sähköisiin tuotteisiin, väestön vanhentumisen ja oppivan tekoälyn kehittymisen vuoksi. (Lempiäinen 2023 b, 25).

Käyttäen pohjana IFR:n julkaisemia vuoden 2022 tilastoja, Lempiäinen (2023 a) kirjoittaa Euroopan teollisuudessa käytettävän robotiikan kituvan koko Euroopassa Aasiaan ja Amerikkaan verrattuna. Maailmanlaajuisesti 28 % roboteista on käytössä elektroniikkateollisuudessa, 26 % autoteollisuudessa ja 12 % konepajateollisuudessa. Suomen vähäinen elektrotekniikka- ja autoteollisuus selittääkin, miksi Suomessa robotiikka ei ole edistynyt samalla tavalla kuin isommissa teollisuusmaissa. (Lempiäinen 2023 a, 20.) Aasiassa yritykset tuottavat valtavat määrät kulutustavaraa koko maailmalle, joten siellä automatisointiin on valtavasti potentiaalia. Suomi on alkanut valjastaa robotiikkaa kehittyneemmissä monirobottijärjestelmissä, joissa tuotteet pyritään valmistamaan ilman ihmisen tekemiä välivaiheita. (Lehtinen 2024, 29.) Konepajateollisuuden robotisointi Suomessa puolestaan on kansainvälisesti huippuluokkaa ja 42 % Suomen uusista roboteista otettiin konepajojen käyttöön. (Lempiäinen 2023 b, 25).

Suomessa tällä hetkellä on käytössä yli kuusituhatta robottia ja niiden elinkaari tyypillisesti on yli 15 vuotta. Uusia robotteja Suomeen on vuonna 2022 IFR:n tilastojen mukaan investoitu 631 kappaletta. 85 % näistä on muun muassa hitsaukseen sopivia teollisuuskäsivarsirobotteja. Lempiäinen kirjoittaa, että yhteistyörobottien määrää ei erikseen lasketa, mutta karkeasti voidaan sanoa, että 10 % myydyistä roboteista on nykyään yhteistyörobotteja, joten uusien yhteistyörobotteja otettiin Suomessa käyttöön noin 60 kappaletta. (Mts. 22–25.)

Robotiikka on yleistä suuren kokoluokan konepajoissa, joissa robotit pääosin toimivat hitsauksessa tai kappaleenkäsittelyssä työstökoneille. Pienet ja keskikokoiset yritykset lähestyvät varovasti robotisointia, sillä sen pelätään olevan vaivalloista ja kallista. Yaskawa Finlandin puheenjohtaja Nina

Lehtinen Tieteessä Tapahtuu (2015) -lehden kirjoituksessa kertoo näkevänsä paljon valjastamattomia potentiaalia kaupan alalla, logistiikassa, rakennustuoteteollisuudessa ja elintarviketeollisuudessa. Lehtisen mielestä robottiautomaatioon voi lähteä mukaan ilman suuria ponnistuksia ja jopa osittaiset prosessien automatisoinnit lisäävät tuottavuutta. (Lehtinen 2024, 28–29.)

2.2 Robottiikan mahdollisuudet Suomessa

Suomen eduskunnan Tulevaisuusvaliokunnan 2018 julkaisussa (Linturi & Kuusi 2018) robottiikan arvioidaan vaikuttavan tulevaisuudessa merkittävästi niin kotitalouksiin kuin yrityksiin. Julkaisussa robottiikka termiä käytetään kuvaamaan monenlaisia uusia autonomisesti toimivia laitteita. Teknologiamuutoksen avulla rakennusteollisuuden ennustetaan kykenemään tarjoamaan yksilöllisiä ratkaisuja entistä taloudellisemmin, koska robotit voivat 3D-tulostaa pursottamalla seiniä esimerkiksi kipsistä ja betonista, tai tehdä tiiliverhoilu. Roboteilla voidaan jo valmistaa tiiliä, kuljettaa ne ja muurata ne seinäksi. Myös teräsrakenteita hitsaavat robotit ovat kehitteillä. (Linturi & Kuusi 2018, 278.)

Itsenäisesti työskentelevät siivousrobotit ovat olleet arkipäivää jo pitkään, mutta toistaiseksi muuten kodinhoidossa roboteista ei ole ollut hyötyä. Monimutkaisempia kodinhoidon työtehtäviä on kuitenkin jo onnistuneesti suoritettu kävelevällä ja kädellisellä robotilla, mutta itsenäisiin kotitalousrobotteihin on vielä pitkä matka. Lumen auraus voi olla myös ammatti, joka tulevaisuudessa tehdään puoli-itsenäisillä robottiajoneuvoilla. Vastaavasti myös esimerkiksi roska-autot ja maatalousajoneuvot olisivat myös robotisoitavissa. (Mts. 272–274.)

Julkaisussa kerrotaan, että jopa liikenteen robotisoimisessa on paljon potentiaalisia hyötyjä. Suomessa on noin kolme miljoonaa autoa, joista suurin osa on kansalaisten omistamia yksityisautoja, joiden käyttöaste on kuitenkin vain 4 %. Robotisoitujen yhteiskäyttöajoneuvojen, eli itsenäisesti ilman ihmisen valvontaa ajavien ajoneuvojen, käyttö vähentäisi tarvetta omistaa omaa autoa, kun samaa ajoneuvoa voisi käyttää useita ihmisiä. Itsenäisesti ajavat autot voisivat taksien tavoin tarjota kyytejä niitä tarvitseville, jolloin omaa autoa ei tarvitsisi omistaa. Suomessa ihmiset käyttävät autojen ohjaamiseen aikaa arviolta miljardi tuntia vuodessa, mikä voitaisiin valjastaa muuhun käyttöön. Samalla myös kuljettajakustannukset poistuvat ja parkkipaikkojen tarve vähenee. (Mts. 252.)

Yksilöllisten tuotteiden valmistaminen robotiikan avulla saattaa hieman yllättävästi sopia hyvin esimerkiksi vaateteollisuuteen. 3D-mittaamalla asiakas saadaan tarkat, yksilölliset mitat, jotka robotisoidussa tuotannossa voidaan ottaa huomioon yksilöllistä tilausta valmistaessa (mts. 286). Vaate-teollisuuden kehittyminen yksilöllisiin mittatilaustoihin mahdollistaisi vaatekauppojen varastojen kokojen pienentämisen, sillä valmiita tuotteita ei tarvitsisi olla heti saatavilla. Asiakas löytäisi mallin mistä pitäisi, katsoisi miltä vaate näyttäisi 3D-avataarinsa päällä ja tekisi tilauksen uudesta vaatteesta. Tällainen palvelu kuitenkin olisi alkuun kallista ja hidasta, joten nopeaa muutosta vaateteollisuuteen ei ole ainakaan heti odotettavissa.

3 Teollisuuden robottityypit

Kansainvälisen standardointijärjestön International Organization for Standardization (ISO) standardissa ISO 8373:2021 (Robotics — Vocabulary) määrittellään tarkasti, mitä robotiikan termeillä tarkoitetaan. Standardin mukaan robotti määrittellään ohjelmoiduksi toimielinmekanismiksi, jolla on jonkinasteinen kyky työskennellä itsenäisesti liikkumalla, käsittelemällä tai paikoittamalla kappaletta. Robotti terminä sisältää myös ohjausjärjestelmän, eikä tarkoita ainoastaan liikkuvaa käsi-vartta. Teollisuusrobotti määrittellään toimilaitteeksi, jossa liikkuvia akseleita on oltava vähintään kolme ja jonka ohjelmointi on mahdollista käyttäen jotakin kommunikointirajapintaa. Lisää robotiikkaan liittyviä termejä määrittellään standardissa ISO 9283:1998 (Manipulating industrial robots), esimerkiksi robottien hyötykuorma, toistotarkkuus, ulottuma, asennustavat ja työkierron nopeus. (ISO 8373:2021; ISO 9283:1998.)

Muut robottityypit, joita ei standardin ISO 8373:2021 mukaan lasketa teollisuusroboteiksi, ovat palvelurobotit ja lääkintärobotit. Palvelurobotit määrittellään olevan robotteja, jotka suorittavat joko henkilökohtaiseen tai ammatilliseen käyttötarkoitukseen ihmisille tai laitteistoille hyödyttäviä tehtäviä. Esimerkkeiksi palvelurobottien käyttötarkoituksista annetaan kappaleiden käsittely, tarkistus, valvonta, tavaroiden tai henkilöiden kuljetus, tiedon tarjoaminen, siivoaminen, fyysinen tukeminen, opastus, ruuanlaitto- ja käsittely ja siivous. Lääkintäroboteille ei puolestaan ole tarkempaa määritelmää kuin robotit, joiden käyttötarkoitus on olla lääkinnässä käytettäviä sähköisiä laitteita tai järjestelmiä. Robottia ei siis tee teollisuusrobotiksi, palvelurobotiksi tai lääkintärobotiksi sen rakenne tai ominaisuudet, niin paljon kuin robotin käyttötarkoitus ja erityisesti sen käyttöympäristö. (ISO 8373:2021.)

Teollisuusrobotit rakentuvat useasta tukivarresta, jotka liikkuvat toistensa suhteen joko pyörähtämällä tai suoraviivaisesti. Teollisuusrobottien nivelet numeroidaan rungon kiinnityskohdasta alkaen (Lempiäinen 2023 a, 19.) ja näitä tukivarsien liittymäkohtia voidaan verrata ihmisen kehossa oleviin niveliin. Usein robottien luokittelu eri kategorioihin tehdään nivelten määrän mukaan. Tukivarsien nivelten varassa muuttuvia asentoja kutsutaan vapausasteiksi. (Groover 2015, 222.) Kuusiakselisilla roboteilla on siis kyky liikkua ylös ja alas, vasemmalle ja oikealle, eteen ja taakse, sekä pyöriä vaaka-, pysty- ja sivuttaissuuntaisesti (Hänninen 2021). Robotiikan kehittyessä erilaisiin työtehtäviin aiheuttaa hyvin eri muotoisten ja näköisten robottien kehittämistä. Robottityyppejä teollisuudessa on kuitenkin pitkään ollut kuusi erilaista.

3.1 Nivelvarsirobotit

Robottityypeistä nivelvarsi- tai kiertyvänivelinen robotti (engl. articulated robot) on teollisuudessa yleisin. Noin 90 % teollisuusroboteista on kuusiakselisia käsivarsirobotteja (ks. kuvio 4). Suositun kuusiakselisen käsivarsirobotin etuna on, että sillä voi saavuttaa lähes minkä tahansa asennon, koska robotti voi liikkua hyvin monenlaisiin asentoihin. Kuusiakselisella käsivarsirobotilla runkoa kiertävä akseli on nivel 1 ja viimeinen työkalulaippaa pyörittävä akseli on nivel 6. (Lempiäinen 2023 a, 17.)



Kuvio 4. Nivelvarsirobotin pääosat (Billing 2023, 118)

3.2 SCARA-robotit

Yleisimpien kuusiakselisten käsivarsirobottien lisäksi SCARA-robotit (lyhenne engl. sanoista Selective Compliance Assembly Robot Arm) on kehitetty suoraviivaisiin kokoonpanotehtäviin. Niissä on usein neljä vapausastetta, mutta ominaista näille roboteille on niiden kyky mukautua kokoonpanotehtävään vapauttamalla vaakaliikkeen akselit asennuksen aikana. (Lempiäinen 2023 a, 17–18.) SCARA-robotti (ks. kuvio 5) soveltuu hyvin elektroniikkakomponenttien kaltaisten kevyiden kappaleiden kokoonpanotöihin. Robotin rakenteen ansiosta käsivarren koko voi olla hyvin pieni, jolloin sitä voi käyttää ahtaissa paikoissa. Tarttujan lähestymisasento on vastaava karteesisen robotin kanssa. Karteesiseen robottiin verrattuna SCARA on usein nopeampi, mutta myös hankintahinnaltaan kalliimpi. (Hänninen 2022, 121–122.)



Kuvio 5. Omron i4H-sarjan SCARA-robotti (Omron RS4-2066502 n.d.)

3.3 Delta-robotit

Delta, eli rinnakkaisrakenteiset robotit, ovat hyvin nopeita ja tarkkoja, ja ne soveltuvat erityisesti elintarviketeollisuuden pakkaus- ja lajittelutehtäviin (Lempiäinen 2023 a, 18). Tavallisesti niillä on kolme akselia, joita samanaikaisesti käyttämällä robotti voi liikuttaa tarttujaa erittäin nopeasti ja tarkasti. Akselit ovat liitetty yhteiseen alustaan nivelakselilla. Rinnakkaisrakenteisten robottien heikkoutena on kuitenkin alustaansa nähden pieneksi jäävä ulottuma (What Kinds of Industrial Robots Are There? A Guide on the Features of the Major 6 Types 2018.) Jos rinnakkaisrakenteisessa

robotissa on enemmän kuin kolme akselia, pystyy robotti tekemään myös kiertoliikkeitä, kuten kuusiakselinen ABB:n IRB 390 FlexPacker deltarobotti (ks. kuvio 6), joka pystyy deltarobotille kolmen sivuttaisliikeakselien lisäksi pyörittämään tarttujaa hyvin monipuolisesti eri asentoihin.



Kuvio 6. ABB IRB 390 FlexPacker deltarobotti (ABB 2022)

3.4 Karteesiset robotit

Karteesinen robotti (engl. cartesian robot), tunnettu myös nimillä suorakulmainen robotti, lineaarirobotti (engl. linear robot) ja portaalirobotti (engl. gantry robot), on robotti, jonka kehikon reunat ovat koko työskentelyalueen ympärillä. Karteesinen robotti (ks. kuvio 7) voi liikkua perinteisesti vain kolmella vapausasteella. Suorakulmaisia robotteja käytetään usein varastotehtävissä tuotteiden tai laatikoiden käsittelyyn. (Hänninen 2021, 115.)



Kuvio 7. Feston karteesinen robotti YXML (Linear gantry YXML n.d.)

3.5 Napakoordinaatistorobotit

Ensimmäinen kaupallinen robotti 1960-luvulla oli Unimation Inc. valmistama Unimate niminen napakoordinaatistorobotti (ks. kuvio 8). Napakoordinaatistorobotit (engl. polar configuration robot tai spherical robot) olivat ennen yleisimmin käytettyjä robotteja, mutta niiden suosio on laskenut vuosien mittaan. (Groover 2015, 224.) Napakoordinaatistorobottien liikerata on pallomainen. Ne voivat kuitenkin liikkua vain joko kääntymällä, nostamalla tai ulottamalla niiden käsivartta. Etuna napakoordinaatistoroboteilla on niiden toistotarkkuus laajoilla kiertävillä liikeradoilla. Niiden käyttö on yleisesti korvattu moderneimmilla robottityypeillä, mutta joissain tapauksissa ne vielä soveltuvat esimerkiksi paletointiin, hitsaukseen tai ruiskuvalukoneen käytön automatisointiin. (Bernier 2023.)



Kuvio 8. Unimation Inc. valmistama Unimate (Robot, First Unimate Robot Ever Installed on an Assembly Line, 1961 n.d.)

3.6 Sylinterirobotit

Sylinterirobotin (engl. cylindrical robot) toiminta-alue on nimensä mukaisesti sylinterimäinen. Niissä on karteesisen robotin tavoin ainoastaan kolme vapausastetta. Sylinterirobotti soveltuu tehtäviin, jotka eivät vaadi deltarobotin kaltaista nopeutta tai tarkkuutta. Sylinterirobottien etuna on kuitenkin niiden vähäisempi tilantarve sekä alhainen hinta. Ne soveltuvat esimerkiksi sähkökomponenttien juottamiseen tai kevyiden kappaleiden käsittelyyn. (Bernier 2022.) Sylinterirobotit (ks. kuvio 9) ja SCARA-robotit muistuttavat ulkonäöltään toisiaan, mutta sylinterirobotin akselit ovat pääosin lineaarisia akseleita. Sylinterirobotin liikkeet saavutetaan pääosin lineaariliikkeiden avulla, toisin kuin SCARA:n rotaatioliikkeiden.



Kuvio 9. Hudson Roboticsin laboratoriotyöskentelyyn kehitetty sylinterirobotti (SciClops Microplate Handler n.d.)

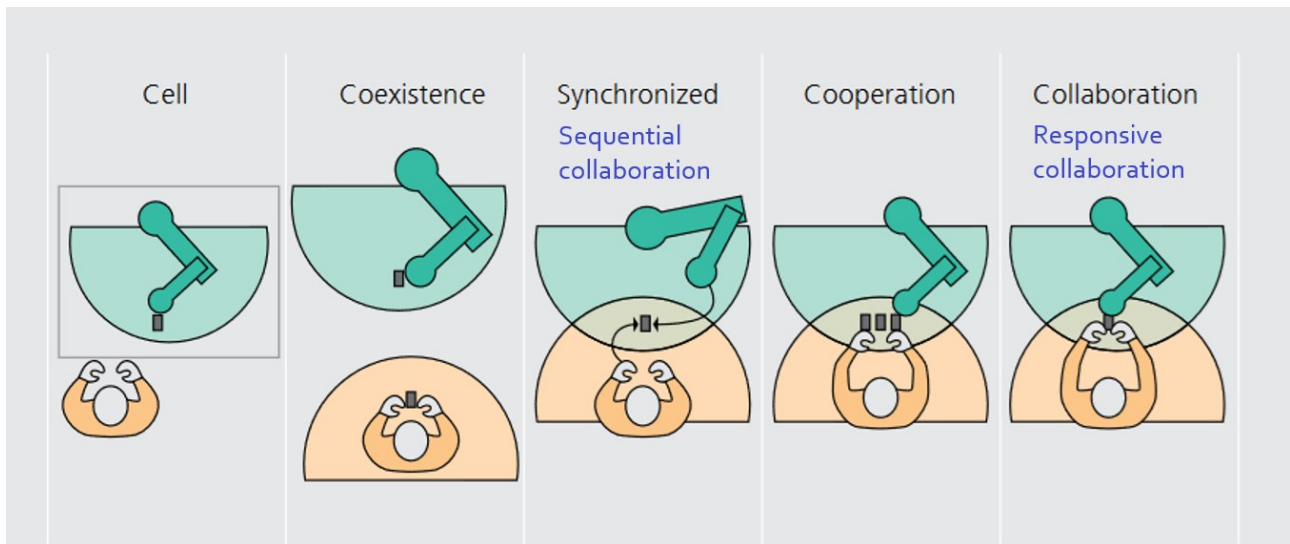
4 Yhteistyörobotit

Kansainvälinen standardointijärjestö International Organization for Standardization (ISO) määrittelee standardissa ISO 10218-2:2011 yhteistoimintarobotiksi (tunnetaan myös termillä yhteistyörobotit) sellaiset robotit, jotka ovat suunniteltu olemaan vuorovaikutuksessa ihmisen kanssa jaetussa yhteistyötilassa (ISO 10218-2:2011). Yhteistyörobotit eivät ole erillisiä omanlaisia rakenteellisia ratkaisuja, vaan robotteja, jotka ovat mekaanisesti tai ohjelmallisesti suunniteltu olemaan turvallisia läheisyydessä työskenteleville ihmisille. Kuviossa 10 on esitetty yhteistyörobotteja nivelvarsi-, SCARA-, delta- ja karteesisen robotin rakenteilla.



Kuvio 10. Erilaisia yhteistyörobottien rakenteita (CR-14iA/L n.d.; PreciseFlex 400 n.d.; Wyzo sidebot n.d.; PreciseFlex 100 2022; muokattu)

Bauer, Bender, Braun, Rally & Scholtz (2016) jakavat robotiikan yhteistyöskentelytavat viiteen eri kategoriaan. Solussa (engl. cell) toimiva robotti on usein robotti, jolla ei ole saavutettu yhteistyön vaatimaa turvallisuutta, jolloin ihminen ja robotti ovat erotettu toisistaan suoja-aidalla ja ihminen ja robotti vaihtavat kappaleita esimerkiksi kuljettimien avulla. Rinnakkaistyöskentely (engl. coexistence) saavutetaan, kun ihminen ja robotti ilman suoja-aitaa voivat turvallisesti työskennellä toistensa läheisyydessä, mutta ne eivät kuitenkaan jaa samaa työskentelyaluetta. Synkronoitu (engl. synchronized) työskentely on sitä, kun yhteistyörobotti ja ihminen jakavat yhteisen työskentelyalueen, mutta käyttävät sitä vuorotellen. Yhteistyöskentelyssä (engl. cooperation) robotti ja ihminen jakavat työskentelyalueen ja voivat käyttää sitä samaan aikaan, kuitenkin niin, että molemmat käsittelevät eri kappaleita kerrallaan. Yhteistyössä (engl. collaboration) työskentely on ihmisen ja yhteistyörobotin tiivistä yhdessä työskentelyä esimerkiksi saman kappaleen kanssa. (Bauer ym. 2016, 8–9.) Kuviossa 11 on esitetty visuaalisesti viisi erilaista robotin ja ihmisen yhteistyökeinoa. Vihreä tausta kuvastaa robotin ulottumaa ja samalla sen työskentelyaluetta ja keltainen kuvasta ihmisen työskentelyaluetta.



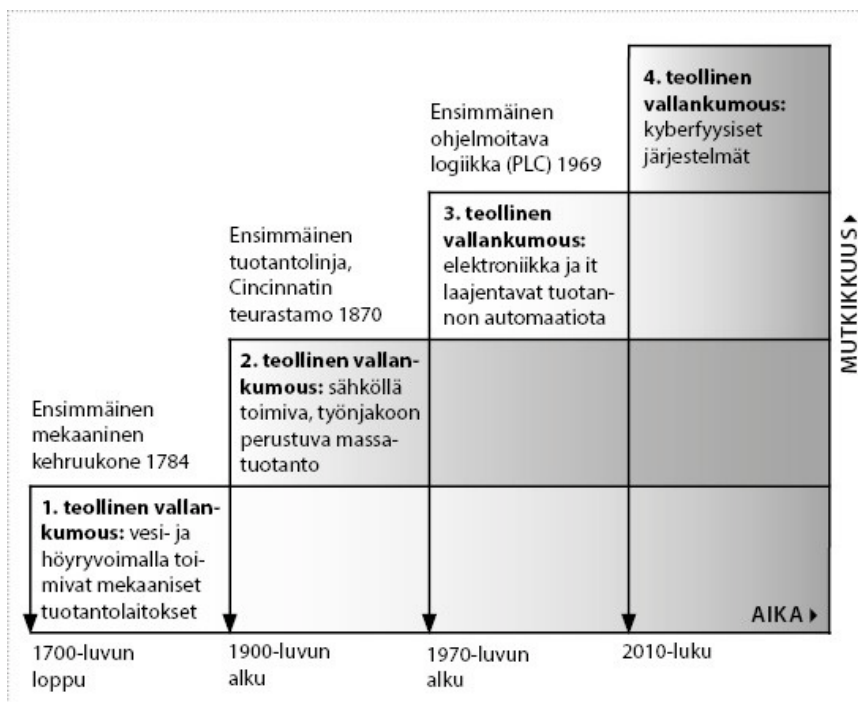
Kuvio 11. Yhteistyörobotin viisi eri yhteistyöskentelytapaa (Bauer ym. 2016, 9, muokattu)

International Federation of Robotics (IFR) käyttää kuvan oikeanpuolimmaisista yhteistyökeinosta termiä reagoiva yhteistyö (engl. responsive collaboration), mikä on Bauer ym. käyttämän yhteistyön sijasta huomattavasti selkeämpi termi. Lisäksi synkronoitu (engl. synchronized) termi on korvattu termillä vuorottainen yhteistyö (engl. sequential collaboration), mikä kuvaa selkeämmin, että työ samalle kappaleelle tapahtuu vuorottain robotin ja ihmisen kanssa. IFR:n mukaan suurin osa tämän hetken yhteistyöapplikaatioista tapahtuu joko rinnakkaistyöskentelynä tai vuorottaisena yhteistyönä. (Demystifying Collaborative Industrial Robots 2020.)

Kaikista yhteistyörobottien turvaominaisuuksista huolimatta niiden käyttöönotossa on huolehdittava asianmukaisesta työturvallisuudesta. Robotiikan ottaminen käyttöön erilaisissa työtehtävissä on huomattavasti helpottunut yhteistyörobotiikan mukana tulleen käytön yksinkertaistumisen myötä, mutta työturvallisuudesta huolehtimista ei tule unohtaa. Robotin kaltaisen työkoneneen käyttöönotossa tulee varmistaa sen käyttäjien ja ympäristössä olevien ihmisten turvallisuus. Turvallisuus varmistetaan tekemällä asianmukainen riskiarviointi ja CE-merkitään vaadittavat toimenpiteet, jonka jälkeen yhteistyörobotista vasta tulee koneturvallisuuden direktiivin 2006/42/EY mukainen.

5 Teolliset vallankumoukset

Teknologian kehittyessä teollisuudessa on tapahtunut viimeisen kolmensadan vuoden aikana useita merkittäviä kehitysaskelia, joita kutsutaan teollisiksi vallankumouksiksi (ks. kuvio 12). Ensimmäinen teollinen vallankumous (engl. Industry 1.0) alkoi Iso-Britanniassa 1700–1800-luvun vaihteessa. Silloin uudet tuotantomenetelmät, kuten höyrykoneet, yleistyivät ja muuttivat teknologiallaan teollisuutta. Toinen vallankumous (engl. Industry 2.0) alkoi uudesta teknologiahyppystä, kun autot, sähkövalot, sähkö- ja polttomoottorit yleistyivät 1800-luvun lopulla, josta alkoi teollisuuden muuttuminen nykyaikaiseen suuntaan. Kolmas vallankumous (engl. Industry 3.0) ei enää ollut fyysisten laitteiden kehittymisen tulos, vaan kyseessä oli digitaalinen vallankumous, jolloin teollisuudessa elektroniikan ja tietokoneiden käyttö yleistyi. Neljäs teollinen vallankumous (engl. Industry 4.0) on nyt menossa ja siihen kuuluu esineiden internetin (IoT), robotiikan ja tekoälyn yleistyminen. (Hänninen 2022, 17.)

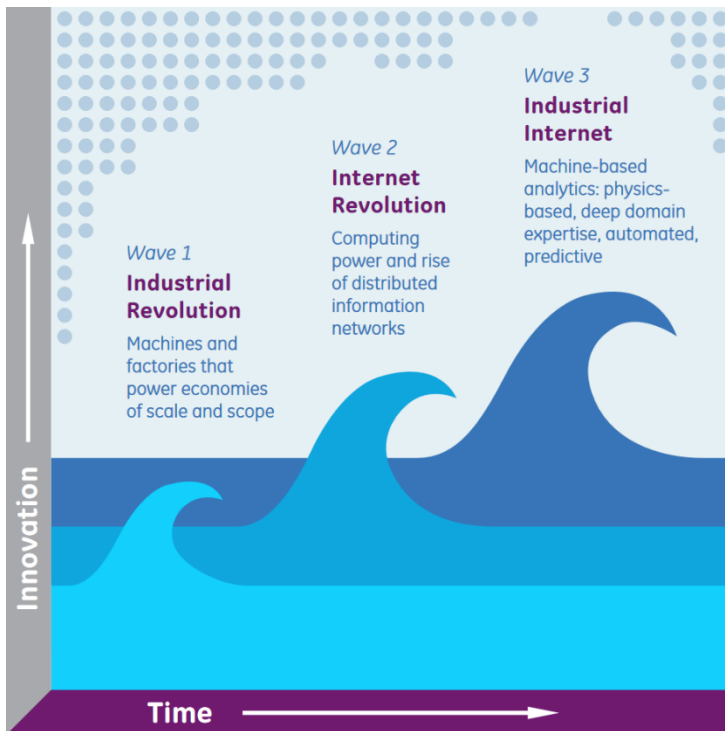


Kuvio 12. Teollisen vallankumoukset (Collins 2016, 39)

Termin *teollinen internet* (engl. industrial internet) ensimmäisenä ottivat käyttöön konsultti- ja analytikkoyhtiö Frost & Sullivan vuonna 2000 heidän julkaisemassa raportissaan. Siihen aikaan raportissa kuvatut teknologiat eivät olleet helposti tai kustannustehokkaasti toteutettavissa,

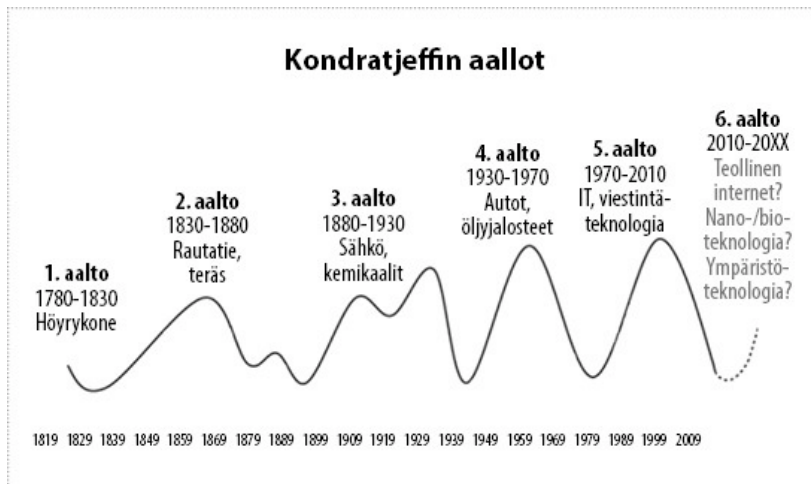
minkä vuoksi termi jäi pitkäksi käyttämättömäksi. Vuonna 2011 Saksassa Hannoverin teollisuusmessuilla professorit Kagermann, Lukas ja Wahlster esittelivät tulkintansa, kuinka teollinen internet on neljännen teollisen vallankumouksen alku. Siitä alkoi hanke Teollisuus 4.0, jota Saksan hallitus ajoi valmistavan teollisuuden vahvistamiseksi vuodesta 2012. Samana vuonna General Electornics - yhtiö alkoi aktiivisesti käyttää termiä teollinen internet, mikä merkittävästi edesauttoi termin arkipäiväistymistä. Termin nimen perusteella voisi olettaa, että sillä tarkoitetaan teolliseen ympäristöön valjastettua internettiä (tai sisäverkon intranettiä), mutta se tarkoittaa laajempaa kokonaisuutta. General Electornicsin määrittelyssä ”teollinen” tarkoittaa, että teollisesti valmistettavat koneet ja muut laitteet tulevat itsessään sisältämään jonkinasteista ohjelmistopohjaista älyä. (Collins 2016, 29–30, 37–38.) Teollinen internet soveltuu valmistavan teollisuuden toimialan lisäksi hyvin vähittäiskauppoihin, maatalouteen, metsäteollisuuteen, kaivosteollisuuteen, terveydenhuoltoon, logistiikkaan, liikenteeseen, energiaverkkoihin ja energiatuotantoon (mts. 91). Termit Teollisuus 4.0 ja teollinen internet tarkoittavat siis pitkälle samaa teknologiakehityksen hyödyntämistä, mutta Teollisuus 4.0 rajoittuu tarkoittamaan sitä vain valmistavan teollisuuden ympäristössä (mts. 30).

Teollisia vallankumouksia voidaan luokitella myös eri tavalla. Amerikkalaisen General Electric -yhtiön tulkintatapa on, että vallankumouksia on vasta kolme. Yhtiö kuvaa vallankumoukset kuvion 13 mukaisina aaltoina. Ensimmäinen aalto alkoi ensimmäisistä mekaanisista laitteista ja sama aalto kesti tietokoneiden keksimiseen asti. Toinen aalto alkoi internetin keksimisestä, jolloin kyseessä oli informaatiointensiivinen muutos. Kolmas muutos alkoi esineiden internetin (IoT) ja big datan kaltaisten muutosten tullessa, jolloin suuria datamääriä siirtyy laitteistoilta toisille ja kerättyä informaatiota käytetään älykkääseen päätöksentekoon. (Mts. 40–41.)



Kuvio 13. Teollisen vallankumouksen kolme aaltoa (Evans & Annunziata 2012, 7)

Kolmas tulkintatapa teknologiakehityksen aiheuttamiin teollisiin vallankumouksiin perustuu alkujaan venäläisen taloustieteilijän Nikolai Kondratjeffin 1900-luvun alussa esittämään teoriaansa. Kehitys kuvataan kuvion 14 mukaisesti kuutena eri aaltona. Ensimmäinen aalto alkoi samaan aikaan kuin aiemmin mainituissa tulkinnoissa, mutta talouden nousu- ja laskukausien vaihteluiden mukaan noin 50 vuoden välein alkoi uusi aalto. Vaikka aallot todellisesti lasketaan talouden eikä teollisuuden muutoksista, aaltojen huiput osuvat merkittäviin teollisuuden kehitysaskeliin. (Collins 2016, 41.)



Kuvio 14. Kondratjeffin aaltoteorian kuusi aaltoa (Collins 2016, 41)

6 Älykkäät järjestelmät

Perinteisesti dataa on analysoitu historiatietona, jolloin prosessien suoriutumista ja muutoksia on seurattu tilastollisia menetelmiä käyttäen. Historiatiedon pohjalta on voitu tehdä analyyskejä ja johtopäätöksiä, kuinka prosessia voidaan parantaa. Kyky monitoroida prosesseja on vuosien mittaan kehittynyt ja teknologian hinta laskenut, mistä on seurannut uusi mahdollisuus käsitellä valtavaa tietomäärää lähes reaaliajassa. Nopea tiedonkulku, säädettävät ohjausprosessit ja älykkäät datan analysointityökalut mahdollistavat historiatiedon ja reaaliaikaisen tiedon käyttämisen niin paremmissa prosessin ohjauksessa kuin prosessin analysoinnissa. Teollisen internetin ajatuksena on luoda ensin paljon relevanttia dataa lisäämällä antureita ja parantamalla prosessien seuranta, siten kehittää kerätyn tiedon analysointia ja lopulta saada aikaan älykäs itseään ohjaava prosessi, jossa resursseja onnistutaan käyttämään parhaalla mahdollisella tavalla. (Evans & Annunziata 2012, 3.)

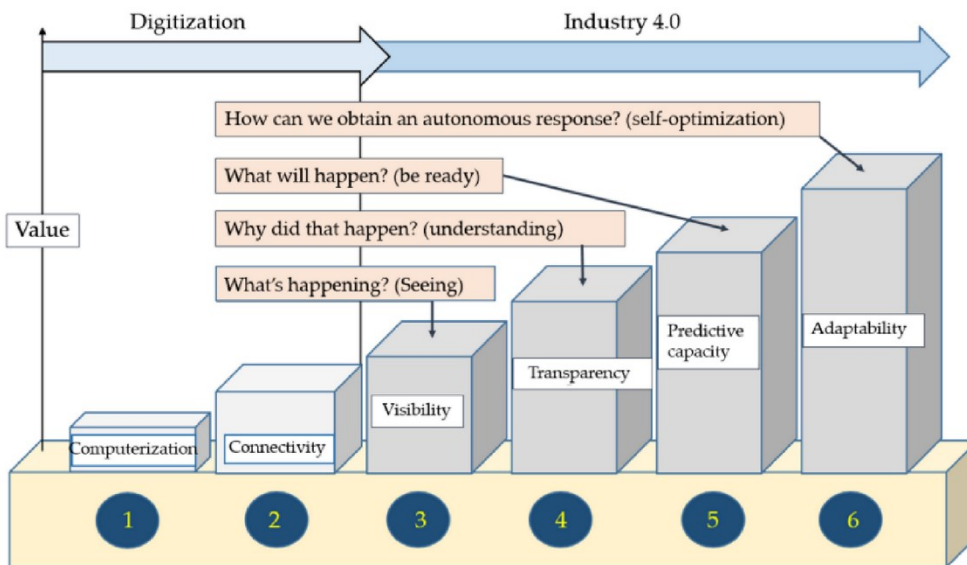
Uuden innovaatioiden aikaansaama teollinen internet kiteytyy entistä älykkäämpiin koneisiin, kehittyneempään analytiikkaan ja ihmisten välisen kommunikoinnin helpottumiseen. Kyky kerätä yksityiskohtaista diagnostista tietoa tuotannosta mahdollistaa entistä tarkemman tuotannon säätämisen. Pienistä tuotannon tehostamistoimenpiteistä kertyy usein pitkällä aikavälillä merkittäviä säästöjä. (Mts. 3.) Teollista internetiä sovelletaan jo nyt esimerkiksi etävalvontaan, ennakoivaan huoltoon, etähallintaan, optimointiin, etäpäivityksiin, älykkäisiin tehtaisiin ja autonomisiin tuotteisiin. Innovaatioiden aikaansaamia uusia sovelluskohteita tulee jatkuvasti lisää. (Collins 2016, 61.)

6.1 Kohti älykstä tehdasta

Teollisuusautomaatiossa tuotantolaitteilta kerätty dataa on käytetty etävalvontaan ja -optimointiin jo vuosikymmeniä. Teollisuuslaitoksissa sisäisten verkkoyhteyksien avulla on voitu koota sensori- ja mittadataa ja hallita keskusvalvomosta käsin laitteita etänä. Vaikka Teollisuus 4.0:n teollinen internet kuulostaa olevan juuri tätä, se itsessään vielä ei edusta teollista internetiä, vaan korkeintaan teollista intranetiä. Dataa on käytetty mekanistisena välinearvona, esimerkiksi kun valvomossa havaitaan painetta mittaavan arvon nousevan, prosessin ohjaus aukaisee venttiilin. Paineen laskiessa aiemmin mitattu arvo menettää kiinnostavuutensa, uusi arvo korvaa vanhan ja usein mitään tietoa tapahtuneesta tilanteesta ei jää talteen. Vaikka lokitietoihin olisi jäänyt merkintä, sillä ei ole pyritty tekemään mitään, korkeintaan käyttämään apuna vikatilanteen selvityksessä. (Collins 2016, 48–49.)

Datan tulee virrata antureilta keskitettyyn tietokantaan, jotta analytiikka-alusta voi hakea informaatiota johtopäätösten tekemiseksi. Datasta puristetaan analyysissä informaatiota ja informaatiosta jalostuu arvokasta tietoa. Jatkuvan analyysin avulla siirrytään hetkellisen toiminnan sijasta datan rikastamisen prosessiin. Datasta jalostetaan liiketoimintahyötyjä, kuten kestävämpiä tuotantolaitteita, tuotantokatkosten vähentämistä, prosessien optimointia, tuotteiden ja palveluiden parantamista. Perinteisen digitalisaation teollisuusautomaatio ei ole kehityksen myötä häviämässä minnekään, vaan päinvastoin sen tarve kasvaa entisestään. Teollisuusautomaation ansiosta monet koneet sisältävät jo entuudestaan antureita, joiden avulla saadaan jatkuvasti raakadataa laitteista ja niiden toiminnasta. (Mts. 48–49.)

Teknologiakehitys tapahtuu, kun laitteista, jotka jo keräävät omasta toiminnastaan dataa, tulee älykkäitä. Älykkäällä laitteella tarkoitetaan, että ne kykenevät viestimään keskenään ja ymmärtävät oman ja ympäristön tilan. Teollinen prosessi kehittyy hetkellisestä toiminnasta kohti datan rikastamisen prosessia. Kerätyn datan arvo kasvaa, kun sitä käytetään hyödyksi entistä enemmän (ks. kuvio 15). Edistysaskeleet teollisuusympäristössä näkyvät pilvipalveluiden, big datan ja data-analytiikan kautta. (Mts. 61–62.)



Kuvio 15. Tiedon hyödyntäminen kasvattaa sen arvoa (André 2019, 52)

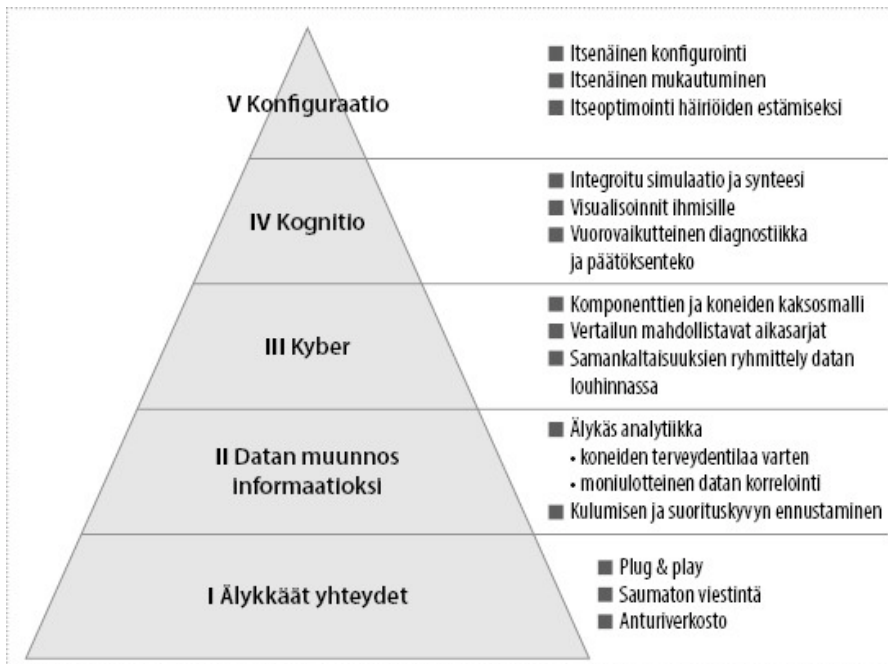
Teollisuudessa koneita ja laitteita ei usein onnistuta käyttämään niiden täyden potentiaalin mukaan, vaikka ne olisivatkin jatkuvassa käytössä. Koneet ovat usein niin monimutkaisia, että niitä käyttävien ihmisten on mahdotonta tehdä tarpeeksi tarkkoja hienosäätöjä koneen jokaiseen työvaiheeseen, joilla koneista saataisiin paras mahdollinen hyöty. Käyttöasteen menetystä tapahtuu aivan koneiden järjestelmätasolla eikä ainoastaan operaattorin hallittavissa olevalla konetasolla. Koneiden järjestelmässä tapahtuvaa parannustyötä on haastavaa osata tehdä oikein ilman kehittyneitä menetelmiä koneiden eri toimintojen ja viiveiden analysoimiseksi. Parannustyö vaatii koneiden etähallinnan mahdollistamista, antureiden lisäämistä, seurattavuuden kehittämistä ja analytiikan parantamista. Yksittäisten koneiden käytöstä saatuja ihmisten tai antureiden keräämiä kokemuksia voidaan analysoida konekohtaisesti, mutta niistä opittuja asioita voidaan käyttää kehittämään kaikkien vastaavien koneiden toimintaa. Älykäs ohjaus voi käyttää siihen yhdistettyjen laitteiden tietoja päättämään koneiden parhaat parametrit ohjaukselle. Erikseen jokaiselta koneelta kerätyn tiedon pohjalta parhaiden mahdollisen säätöparametrien löytäminen voi viedä merkittävästi enemmän aikaa, kuin kollektiivisesti usealta laitteelta keräämällä. (Evans & Annunziata 2012, 5,11.)

Älykkäät laitteet ja järjestelmät kykenevät jakamaan tietoa älykkäästi yhteiselle alustalle, jossa resurssit voidaan jakaa entistä tehokkaammin. Esimerkiksi terveydenhuollon työntekijät voivat tarkistaa, onko sairautta potevan henkilön tarvitsema hoitolaitteisto tai hoitohenkilökunta vapaana.

Tilatiedot eri koneista ja niiden komponenteista auttavat myös seuraamaan osien kulumista. Huollon reagointi nopeutuu ja varaosien säilyttäminen varastossa voi vähentyä, kun varaosat voidaan tilata vasta, kun niitä ennakoidaan tarvittavan. Oppivan analytiikan avulla ennakoivat huollot voidaan ajoittaa optimaaliseen aikaan. Järjestelmien toimintoja seuraavien älykkäiden järjestelmien avulla voidaan palautua isoista järjestelmäkatkoista entistä nopeammin tai ne voidaan jopa estää kokonaan. Jos osajärjestelmä havaitaan käyttäytyvän odottamattomalla tavalla, ongelmaa voidaan eristää entistä nopeammin ja kehittyneen diagnostiikan avulla ongelman paikantaminen helpottuu. (Evans & Annunziata 2012, 11.)

Valmistettavat tuotteet voivat sisäänrakennettujen antureiden avulla seurata itsensä toimintaa ja huollon tarvetta. Tuotteiden toimintoja voidaan säätää etänä ohjelmallisesti, eli optimoida päivityksillä tuotteiden parametrit maksimaalisen eliniän tai tuottavuuden varmistamiseksi. Ennakoivan huollon tarve selviää ajoissa, kun tuotteet voivat ilmoittaa jonkin tuotteen toiminnan osa-alueen heikkenemistä. Älykkäistä tuotteista saatava data voi olla arvokasta tuotteiden loppukäyttäjille. Tuotteiden valmistaja voi lisäliiketoimintana myydä esimerkiksi kuukausimaksullisena palveluna koosteraportteja, kuinka tuotteita on käytetty. (Collins 2016, 61–62.)

Futuristisin tavoite teollisella internetillä ja Teollisuus 4.0:lla on saada aikaan älykäs tehdas, jossa kaikki laitteet ovat huippuunsa anturoituja ja ne ovat yhteyksissä toisiinsa ja yhteiseen analytiikkajärjestelmään. Analytiikkajärjestelmän avulla jokaisen laitteen rooli on selvää tuotantoketjun kannalta ja laitteet osaavat tunnistaa mitä työvaiheita valmistettaville tuotteille tulee tehdä ja missäkin järjestyksessä. Yksinkertaisen datan jatkokäsittelyn avulla kyetään saavuttamaan älykäs ja mukautuva infrastruktuuri (ks. kuvio 16). Tuotanto automaattisesti säätää ja järjestäytyy uudelleen ihanteellisella tavalla, tehtaasta tulee mukautuva ja ihmisen mikrotason valvontaa ja säätämistä ei enää tarvita. (Mts. 86–87.)



Kuvio 16. 5C-arkkitehtuurin pyramidimallissa teollisen internetin infrastruktuuri kehittyy yrityksen noustessa pyramidia ylöspäin (Collins 2016, 146)

Älykäs tehdas optimoi itseään, vähentäen ylituotantoa ja hukka-aikaa, sekä kykenee rakentumaan uudelleen asiakkaan tilausten tarpeiden mukaisesti. Kun huippuunsa viedyn älykkään tehtaan järjestelmään tulee tuotantotilaus, hoitaa tuotannonohjausjärjestelmä tarvittavat tarviketilaukset ja aikatauluttaa sisäisen tuotantoprosessin mahdollisimman vähäisellä resurssien hukkaamisella, mikä tekee siitä myös ympäristöystävällisen. Jos tuotanto on patoutumassa, esimerkiksi ennakoivista huolloista huolimatta yksi tuotantorobotti hajoaa yllättäen, järjestelmä havaitsee tulevan ongelman ja jakaa tuotannon muille roboteille ja säättää tuotantotahtia pullonkaulojen torjumiseksi. Ihmisen vastuulle jää toiminnan valvominen, tuotannon kehittäminen ja mahdollisesti niiden työvaiheiden tekeminen, mitä automatiikalla ja robotiikalla ei ole voitu vielä järkevästi suorittaa. (Collins 2016, 86–87.)

6.2 Signaaleista kenttäväyliin

Robotit, kuten muutkin automaatiolaitteet, kommunikoivat sähköisillä signaaleilla. Yksinkertaisimmillaan tilatiedon välittämiseen käytetään I/O-signaaleja (tulo ja lähtö, engl. input ja output). I/O-signaalilla voidaan viestittää, onko jokin asia ”päällä” tai ”pois päältä” käyttämällä sähköistä viestiä johtimessa, eli johtimessa joko kulkee jännite tai sitten ei kulje. Kehittyneempi tiedonsiirtotapa on

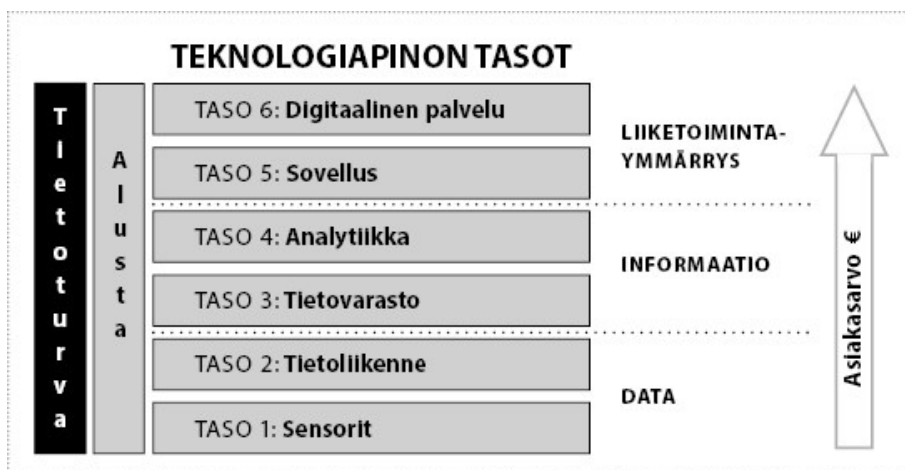
käyttää sarjaliikennetarkkaimia, kuten RS-232:sta, jolloin useiden signaalien tila voidaan välittää eteenpäin samanaikaisesti. Vielä kehittyneempi tapa on käyttää tietoliikenneverkkotekniikoita, kuten Ethernetiä, jolloin voidaan kommunikoida suuria määriä tietoa, kuten useiden laitteiden antureiden tilat. Nykyaikaisissa roboteissa käytetään kaikkia näitä kolmea tiedonsiirtokeinoa. Liuha ja Aho (2023) nostavat robottien kannalta tärkeimmiksi tiedonsiirtoliittymän toiminnoiksi kuvion 17 mukaiset toiminnot (Liuha & Aro 2023, 148.)

- Ohjelmien ylös- ja alaslataus robottiohjaimiin
- Ohjelmien käynnistys ja pysäytys
- Digitaali/analogia -tulojen ja -lähtöjen luku sekä kirjoitus
- Numero/paikkarekisteritietojen tai muuttujien luku sekä kirjoitus
- Virhetilannekoodien välitys
- Toimintatilan monitorointi sekä asetus

Kuvio 17. Tiedonsiirtoliittymällä toteutettavia toimintoja (Liuha & Aro 2023, 148)

Tiedonsiirron protokollat ja standardit ovat työkaluja, jolla tietoa vaihdetaan järjestelmien, kuten robottien ja valvomoiden, välillä. Protokollaa valitessa kannattaa ottaa huomioon sen käyttötarkoitus, suorituskyky ja tietoturva. Protokolla voi olla tarkoituksellisesti luotu luotettavaan ja virheettömään tiedonsiirtoon, tai erityisen nopeaan viestintään, tai siirtämään suuria määriä kerralla. (Collin & Saarelainen 2016, 181–182.) Collin ja Saarelainen (2016) ryhmittelevät protokollat niiden tiedonsiirron periaatteen mukaan kolmeen erilliseen ryhmään. Ensimmäinen on yleistymässä olevat julkaisija-tilaaja-mallin protokollat. Siinä dataa tuottavat järjestelmät ovat julkaisijoita, joiden data virtaa automaattisesti johonkin keskitettyyn pisteeseen, joka puolestaan välittää dataa tilaajan roolissa olevaan järjestelmään, kuten tietokantaan. Järjestelmä voi olla samaan aikaan julkaisija ja tilaaja, minkä vuoksi tällaisella tiedonsiirron periaatteella on etu ympäristössä, jossa uusia laitteita voidaan lisätä tai ottaa pois käytöstä ilman protokollan häiriintymistä. Toinen malli perustuu asiakas-palvelin-malliseen tiedon kyselyyn. Silloin palvelin lähettää dataa vain, kun asiakas pyytää sitä. Tämän etuna on, että verkko ei kuormitu paljon, kun dataa ei lähetetä pyytämättä varmuuden varalta. Kolmas malli on vertaisverkon periaate, jossa data siirtyy yhtäaikaaisesti laitteiden välillä. (Mts. 184–185.)

Collinsin ja Saarelaisen (2016) esittämässä kuvion 18 mukaisessa teknologiapinossa tiedon jalostamisen määrä on suhteessa liiketoiminnalliseen arvoon. Pinon pohjalla oleva data on yksinkertaista anturien välittämää signaalidataa, jota tietoliikennetason kautta tallennetaan tietovarastoon. Tietovarastoon voidaan tallentaa dataa muistakin lähteistä, kuten toiminnanohjauksesta ja asiakkuudenhallinnasta. Neljännellä tasolla datamassan käsittely analytiikkaohjelmistoilla, jotka pystyvät käsittelemään suuria määriä dataa ja esittämään se visuaalisesti tai muuten helposti hyödynnettävissä olevassa muodossa. Viidennellä tasolla selkeäksi muunnettu data esitetään tietokoneiden, puhelinten ja muiden laitteiden sovelluksissa tai nettisivuilla ihmisille helposti luettavasti kojelautoina ja graafisina esityksinä. Kuudes taso on palvelu, jonka avulla yrityksen asiakkaat ja toimittajat hyötyvät koko teknologiapinosta. Sillä tarkoitetaan enemmänkin arvon lisäämistä koko yrityksen liiketoimintaprosesseihin ja uusien liiketoimintamallien mahdollistamista, kuin datan jalostamista vielä entisestäänkin sovellustason jälkeen. Teknologiapinossa kolme alinta tasoa siis ovat tiedon hankkimista ja kaksi seuraavaa ovat tiedon käsittelyä. Kuvion vasemmalla puolella on huomio tietoturvasta huolehtiminen koko prosessin jokaisella askeleella. Alustalla tarkoitetaan arkkitehtuurista suunnitelmaa, jolla koko pino toteutetaan, eli suunnitelma ja yhteiset säännöt millä keinoilla data hankitaan ja käsitellään. (Collin & Saarelainen 2016, 143–144).



Kuvio 18. Teollisen internetin tekniset osakokonaisuudet (Collin & Saarelainen 2016, 143)

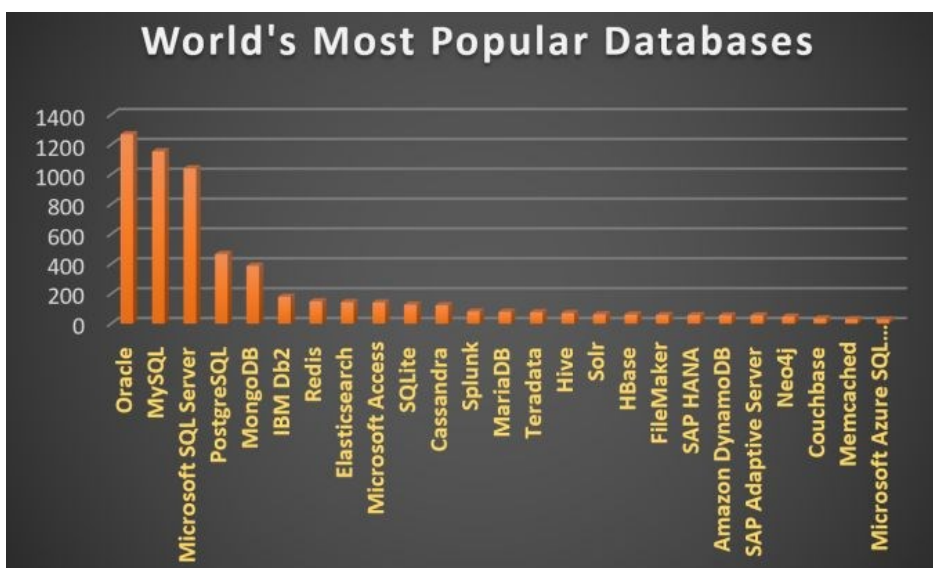
Tässä opinnäytetyössä toteutetaan tasojen 1–5 mukainen järjestelmä, jossa yhteistyörobotin sensoritasolta siirretään tietoa tietovarastoon, luodaan ohjelmisto mikä käsittelee datan ja tekee siitä työntekijöille selkeästi luettavan raportin. Työn tarkoitus ei ole luoda erityisen älykäästä ja monimutkaista analytiikkatyökalua, vaan datan pohjalta rakennetaan yksinkertainen visuaalinen esitys,

jota ihminen voi koneen sijasta analysoida. Tämä pohjatyö kuitenkin mahdollistaa älykkäämpien analytiikkatyökalujen käyttöönoton myöhemmässä vaiheessa.

7 Tietovarastot

Mikäli yhteistyörobotin antureilta ja sen laskureista halutaan kerätä tietoa, tulee tieto tallentaa jonkinlaiseen järjestelmään. Tällaista järjestelmää sanotaan tietovarastoksi. Tietovarastonjärjestelystä datasta analytiikan välineillä voidaan tehdä liiketoimintaa hyödyttäviä päätelmiä. Tietovaraston tärkeä ominaisuus on kyky skaalautumaan datamäärän kasvaessa, pystyä ottamaan vastaan monipuolisesti erityyppisiä datatyyppisiä, vastaanottamaan dataa nopeasti, sekä myös tiivistämään dataa tehokkaasti levytilan säästämiseksi. (Collin & Saarelainen 2016, 195–196.)

Tavallisesti tietovarastona käytetään tietokantoja. Tietokannat voidaan jakaa erilaisiin tietokantatyyppisiin, riippuen miten tietokanta toimii tai minkälaiseen käyttöön se on tarkoitettu. Eniten käytetyimmät tietokannat ovat Oracle, MySQL ja Microsoft SQL Server (ks. kuvio 19). Kaikki kolme noudattavat strukturoitua SQL-mallia (engl. Structured Query Language), jossa tietokantojen rakenne on määritelty tarkasti. Näihin tietokantoihin data tallennetaan riveille ja sarakkeille. (Mts. 196.)



Kuvio 19. Tietokannat järjestettynä suosion mukaan. Suosio laskettu internetistä tietokannan nimen käytön mukaan mm. hakukoneiden hakujen määrästä. (C# Corner)

Jotkut tietokannat voidaan sisällyttää useampaan kategoriaan niiden toimintatavan, käyttökohteen tai muiden ominaisuuksien mukaan, joten luokittelu ei aina ole aivan yksiselitteistä. Tässä opinnäytetyössä tehdään lyhyt katsaus yleisimpiin tietovarastotyyppeihin. On olemassa myös muita tietovarastoja kuin mitä tässä työssä mainitaan, mutta niiden käyttö on joko vähäistä tai ne eivät perinteisesti sovellu hyvin teollisuuden käyttöön.

7.1 Paikalliset tekstipohjaiset tiedostot

Yksinkertaisin keino säilöä tietoa on tallentaa se paikallisiin tiedostoihin, kuten suosittuihin CSV- ja JSON-tiedostoihin. Paikallisten tiedostojen avulla voidaan välttää palvelinten ja monimutkaisempien tietokantojen käyttö, mikä voi nopeuttaa tiedonkeruujärjestelmän toteutusta huomattavasti. Tallennettava data sijaitsee kuitenkin vain yhdellä tietokoneella, eikä sen paikallisia tekstipohjaisia tiedostoja kannata käsitellä useasta lähteestä samanaikaisesti. Jos useampi ihminen tai ohjelmisto tekee lisäyksiä tai muokkauksia tiedostoon, riskinä on jo kirjoitetun datan poistaminen tai sen päälle kirjoittaminen.

CSV-tiedostot (engl. Comma Separated Values, eli pilkulla erotetut arvot) sisältävät yksinkertaista ja usein helppolukuista tekstiä, jossa tiedostoon tallennetaan data rivi kerrallaan. Yksi rivi voi sisältää useita eri arvoja, jotka on erotettu toisistaan pilkulla tai puolipisteellä. Jokainen tiedostoon kirjoitettu rivi sisältää uutta dataa. Sisältö voi olla tekstimuotoisena hankalasti luettavissa, mutta erottelumerkin ansiosta se on helppo muuntaa taulukkomuotoon, joissa data on jaettu riveihin ja sarakkeisiin (ks. taulukko 1). (Dalik, 2012.)

Taulukko 1. Esimerkki CSV-tiedoston sisällöstä tekstimuodossa ja taulukkona

CSV-muodossa	Taulukkona		
aikaleima,anturinimi,arvo	aikaleima	anturinimi	arvo
2024-01-23 11:14:29,Lämpötila1,21.65	2024-01-23 11:14:29	Lämpötila1	21.65
2024-01-23 11:14:25,Paikka1,33.76	2024-01-23 11:14:25	Paikka1	33.76
2024-01-23 11:14:13,Paikka2,48.16	2024-01-23 11:14:13	Paikka2	48.16
2024-01-23 11:14:10,Lämpötila1,21.68	2024-01-23 11:14:10	Lämpötila1	21.68

CSV-tiedostot ovat helppokäyttöisiä ja niiden käsittely ja muokkaus onnistuu lähes jokaisella tekstinkäsittelyohjelmalla. Tiedoston sisältö on myös kompakti, eikä se sisällä tallennetun datan lisäksi mitään ylimääräisiä sisältöä. Yksinkertaisuuden haittapuolena on, että tiedostoon ei voi tallentaa mitään muuta kuin tekstiä ja numeroita, eikä niitä voi erotella toisistaan. Tallennettu tieto on vain sarja merkkejä, joten mikään ei estä kirjoittamasta tekstiä numeroiden sekaan tai toisinpäin. Myöskään mikään järjestelmä ei vahdi, että CSV-tiedoston riville kirjoitetaan oikea määrä erotusmerkkejä. Kirjoitettava data itsessään voi sisältää erotusmerkin, minkä takia yksi arvo pilkkoontuu kahteen ja riville tallentuu enemmän erotettuja arvoja muulle. Mikäli rivi sisältää tällä tavoin liikaa tietoa, taulukkomuotoon siirtämisessä tulee ongelmia, kun rivillä olevat tiedot eivät mahdu ennalta määritettyihin sarakkeisiin. Lisäksi jos CSV-tiedoston koko kasvaa hyvin suureksi, sen avaaminen ja käsitteleminen voi hidastua merkittävästi. (Dalik, 2012.)

Toinen suosittu paikallisen tiedon tallennusmuoto on JSON (engl. JavaScript Object Notation). JSON-tiedoston sisältö on selkeästi rakenneltu, minkä ansiosta se soveltuu hyvin tiedon muuttamiseen muille rajapinnoille ja tiedon välittämiseen eteenpäin muille järjestelmille. Tieto on tallennettu riveille avaimina ja arvoina (ks. taulukko 2), joten toisin kuin CSV-muodossa, jos arvot ja niiden avainparit kirjoitetaan riville erilaisessa järjestyksessä, data loppupäässä ei sekoitu väärin sarakkeisiin. Koska JSON-tiedostoon kirjoitetaan uudestaan ja uudestaan avainparit, lainausmerkit ja aaltosulkeet, kasvaa sen koko CSV-tiedostoa nopeammin. (Morris 2023.) Vaikka data on järjestelty pareihin ja JSON-muoto tukee erilaisia tietotyyppejä, mikään ei kuitenkaan estä kirjoittamasta väärää arvoa väärään avaimen pariin.

Taulukko 2. Esimerkki JSON-tiedoston sisällöstä tekstimuodossa

JSON-muodossa

```
{"aikaleima":"2024-01-23 11:14:29","anturinimi":"Lämpötila1","arvo":"21.65"}  
{"aikaleima":"2024-01-23 11:14:25","anturinimi":"Paikka1","arvo":"33.76"}  
{"aikaleima":"2024-01-23 11:14:13","anturinimi":"Paikka2","arvo":"48.16"}  
{"aikaleima":"2024-01-23 11:14:10","anturinimi":"Lämpötila1","arvo":"21.68"}
```

7.2 SQL-tietokannat

SQL- tai relaatiotietokannat ovat tunnetuimpia, perinteisiä tietokantoja, jotka käyttävät relaatiomallia ja SQL-kieltä. Termi SQL tulee sanoista *Structured Query Language*, eli strukturoitu kyselykieli. Tietokantaan tallennettu data on tietokannassa olevien taulujen sarakkeissa ja riveissä. SQL-tietokannassa täytyy määrittää tarkasti minkä muotoista dataa otetaan vastaan. Tästä syystä tallennettua tietoa on helppo lukea ja käsitellä, mutta tiedonsiirrossa voi esiintyä myös helposti haasteita, mikäli laitteilta lähetettävä tieto ei pysy tarkasti samanlaisessa formaatissa, kuin mitä tietokantaan on määritetty. (Collin & Saarelainen 2016, 196.)

Kuviossa 20 on esitetty havainnollistava kuva SQL-tietokannan taulujen rakenteesta. Siinä MeasurementTable -nimiseen tauluun on kerätty sensoridataa sarakkeeseen "Measurement". Numeerisen datan käsittely tietokannassa on nopeampaa kuin tekstin, joten sen toistuvaa kirjoittamista kannattaa välttää. Voidakseen tunnistaa mistä tauluun tuleva sensoridata on peräisin, voidaan käyttää yhdistäviä tunnisteita, kuten "MachineId" ja "SensorId". Näitä kutsutaan viiteavaimiksi. Kun dataa käsitellään, voidaan taulujen viiteavaimet yhdistää haettavaan dataan haun selkeyttämiseksi. Esimerkiksi MeasurementTable taulun sarakkeen 6 data voidaan esittää muodossa "koneelta MachineBeta anturi BackSensor mittaama arvo oli 1,10".

MeasurementTable			
Id	MachineId	SensorId	Measurement
1	1	1	1,12
2	1	3	3,57
3	2	2	2,16
4	2	3	3,74
5	1	2	2,38
6	2	1	1,10
7	2	3	3,62
8	1	1	1,13

MachineNameTable		SensorNameTable	
MachineId	MachineName	SensorId	SensorName
1	MachineAlpa	1	FrontSensor
2	MachineBeta	2	MiddleSensor
		3	BackSensor

Kuvio 20. Havainnollistava esimerkki kolmesta SQL-tietokannan taulusta

Datamäärän kasvaessa SQL-tietokannoissa ongelmaksi muodostuu kannan hidastuminen. Skaalautuvuudesta saattaa joutua huolehtimaan vaivalloisilla ja pitkässä juoksussa kalliiksi tulevilla hallintakeinoilla. (Collin & Saarelainen 2016, 197) Paljon ongelmia voidaan välttää strukturoimalla data hyvin, indeksoimalla ja skaalaamalla taulut järkevästi alusta asti. Siitäkin huolimatta SQL-tietokantojen indeksoiminen ja uudelleenjärjestely jälkikäteen on erittäin vaivalloista. (Mts. 199).

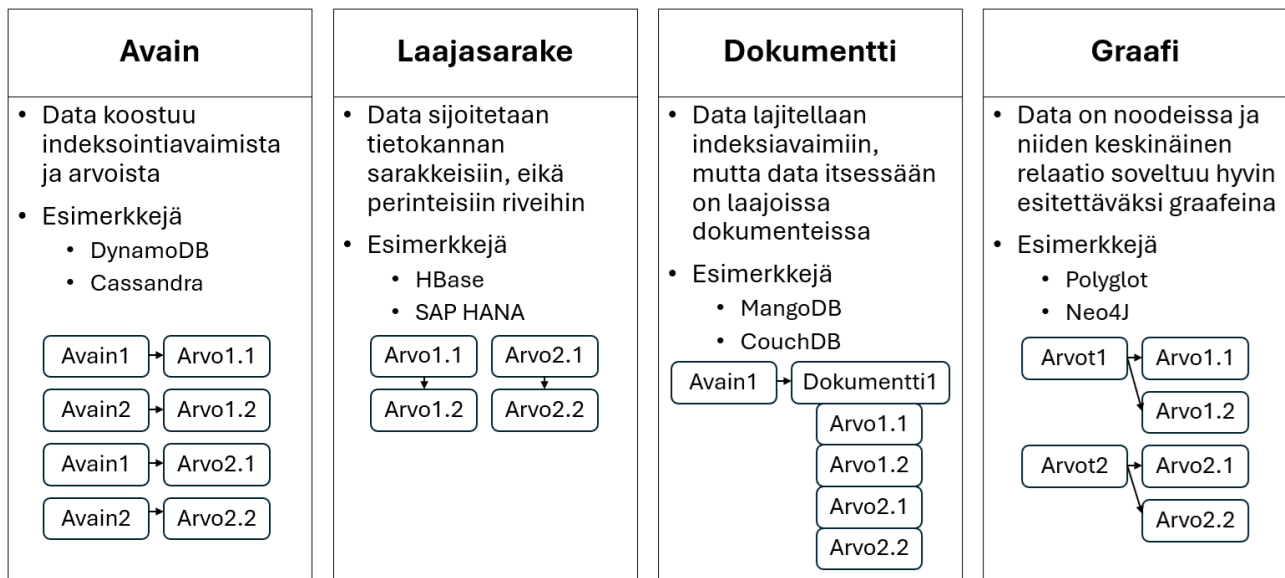
7.3 NoSQL-tietokannat

NoSQL -termi tulee lyhenteestä ei-vain-SQL (engl. Not only SQL). Toisin kuin tarkasti rakenteellisissa SQL-tietokannoissa, samaan NoSQL-tietokantaan voidaan viedä monenmuotoista dataa ilman luokitteluja ja datan täsmällistä pilkkomista tai muuttamista tiettyihin muotoihin. NoSQL-tietokantoja kutsutaan ei-rakenteellisiksi tietokannoiksi, koska ne ovat erittäin skaalautuvia ja joustavia, sekä tietokannan rakennetta ei tarvitse määritellä etukäteen, mikä helpottaa kannan käyttöönottoa ja siihen tehtäviä muutoksia. (Mts. 197.)

NoSQL-tietokannat kasvattavat suosiotaan, sillä ne soveltuvat datamäärän räjähdysmäiseen kasvuun. Datamäärän kasvaessa kirjoittaminen ja lukeminen on nopeampaa NoSQL-tietokannoissa kuin SQL-tietokannoissa. Antureiden käytön yleistyessä ja monimuotoisen datan talteenoton myötä tarve joustavalle tietovarastolle on aiheuttanut NoSQL-tietokantojen käyttöönoton kiihtyvällä vauhdilla. (Collin & Saarelainen 2016, 198.)

Vaikka NoSQL-tietokantoja kutsutaan rakenteettomiksi, todellisuudessa tiedon hakemiseksi ja käsittelyksi tallennetulla tiedolla on oltava jokin rakenne. Yleensä tiedot viedään tietotyyppin mukaan sopivan tyyppiseen NoSQL-kantaan, mutta samaan kantaan voidaan tuoda tietoa hyvin monenlaisessa muodossa, kuten mittadataa antureilta, äänitiedostoja, valokuvia, tekstiasiakirjoja tai muuta. Esimerkiksi häiriöön menneen koneen operaattori voi tehdä häiriöilmoituksen ja liittää siihen valokuvan ja lähettää sen NoSQL-tietokantaan. Metadatan avulla raportti yhdistyy tietokannassa oikeaan sensoridataan, jota häiriötilanteen syytä selvittävät voivat käyttää hyödyksi jopa etänä. (Mts. 197–198.)

Haittapuolena rakenteettomilla tietokannoilla on heikompi tiedon eheyden varmistaminen ja hakujen tekemisen monimutkaisuus. Jotkut NoSQL-tietokannat sallivat kuitenkin perinteisten SQL-hakujen suorittamisen SQL-rajapinnalla. Markkinoilla olevat NoSQL-tietokannat ovat toimintoiltaan ja ominaisuuksiltaan hyvin erilaisia, mutta ne voidaan jakaa kuvion 21 mukaiseen neljään ryhmään: avainarvoperusteiset (engl. key value database), laajasarakkeiset (engl. wide column store), dokumenttirakenteiset (engl. document database) ja graafiset (engl. graph database). (Mts. 198.) Suosituin NoSQL-tietokanta on dokumenttirakenteinen MongoDB.



Kuvio 21. Neljä yleisintä NoSQL tyyppiä (Barooah n.d., muokattu)

7.4 Pilvitietokannat

Tietovarastojen yhteydessä puhutaan usein pilvitietokannoista. Ne ovat helppo ottaa käyttöön, koska fyysistä laitteistoa ei tarvitse ostaa itselleen, vaan omaan tallennustilaan pääsee käsiksi internetin kautta. Tallennustilasta tyyppillisesti maksetaan kuukausi- tai vuosimaksua. Pilvitietokannat eivät ole oma tietovarastotyyppinsä, vaan nimellä viitataan tietovaraston sijaintiin, mihin pääsee käsiksi ainoastaan internetin kautta. Pilvipalvelin voi olla perinteinen relaatiotietokanta, kuten esimerkiksi Googlen Cloud SQL ja Microsoft Azure SQL, tai NoSQL-tietokantoja kuten Amazon DocumentDB tai Microsoftin Azure Cosmos DB.

Pilvitietokantojen tallennustila on myös erittäin edullista ja skaalautuvaa, jolloin miljoonienkin laitteiden datavirrat muuttuvat mahdolliseksi. Pilvessä sijaitsevien tietovarastojen vahvuutena on myös niiden helppo liitettävyys analytiikan, visualisoinnin ja laitehallinnan pilvipalveluihin, jolloin puhutaan hajautetusta laskennasta, eli pilven reunalla tapahtuvasta laskennasta (engl. edge computing). Pilvi ei kuitenkaan sovellu tietovarastoksi, jos sen käyttötarve ei salli viivettä datan viemisessä tai verkkoyhteydestä tai palvelusta johtuvia katkoja. (Collin & Saarelainen 2016, 202.)

7.5 Apache Hadoop

Yksi tunnetuimmista big datan tietovarastoista on Apache Hadoop. Kyseessä ei ole SQL tai NoSQL-tietokannan tyylinen keskitetty tietovarasto, vaan järjestelmä, mikä hajauttaa tallennettavan datan eri palvelimille. Järjestelmän ydin on levyjärjestelmä, joka pilkkoo saapuvaa dataa ryhmiin ja jakaa ne eri puolille palvelinrypästä, eli klusteria. Sama data jaetaan varmuuden vuoksi useaan palvelimeen, minkä ansiosta järjestelmä sietää hyvin vikaantumisia. Samasta syystä järjestelmä vaatii paljon tallennustilaa, mutta muuten komponenttivaatimukset eivät ole perinteistä tietokantapalvelinta kummempat. (Collin & Saarelainen 2016, 200.)

Big datan tallentamiseen tarkoitettut järjestelmät tarjoavat mahdollisuuden entistä älykkäämmälle datan hyötykäytölle. Valtavasta big data määrästä on perinteisillä työkaluilla hankalaa saada mitään hyödyllistä irti, mutta kun mukaan otetaan hajautetun datan prosessoinnin työkaluja, kuten Hadoop MapReduce tai Apache Spark, voidaan datamassaa analysoida tuotteliaasti. Koska big datan analytiikan käyttäminen on isolle osalle ohjelmisto- ja tuotekehittäjiä vielä toistaiseksi vieraampaa kuin perinteisten SQL-työkalujen käyttö, saataville on tullut erittäin suuri joukko Hadoop-klusterin päällä suoritettavia SQL-työkaluja, jotka tarjoavat perinteisemmän tavan käyttää SQL-kyselykielestä tuttuja hakuominaisuuksia. (Mts. 199–200.)

8 Universal Robots yhteistyörobotin tiedonsiirto

Kommunikointiväylien avulla roboteilta voidaan lukea tarkkoja käyttötietoja, kuten esimerkiksi onko robotti päällä, mikä robottiin kytketyn hätäseispiirin tila on, mikä robotin nivelten liikenoisuus, lämpötila ja virrankulutus on. Valmistajan ennakkoon määrittämien tietojen lisäksi käyttäjät voivat itse luoda omaa vapaavalintaista tietoa, kuten kuinka monta kertaa robotti on suorittanut ohjelmasyklinsä. Kommunikointiväylät ovat keino seurata robotin tietoja etänä tai tallentaa niitä johonkin järjestelmään.

OPC UA on 2008 vuonna julkaistu ja myöhemmin Industry 4.0 -liikkeen valitsema tekniikka tiedonsiirtoon (Liuha & Aro 2023, 153). Sen potentiaalista olla yleisimmin käytetty, parhaiten tuettu ja parhaiten ajan tasalla pysyvä protokolla olisi tehnyt OPC UA:n valitsemisen työn tiedonsiirtoprotokollaksi erityisen perustelluksi. Universal Robots yhteistyörobotit eivät kuitenkaan itsessään tue

OPC UA -kommunikointia ja se on vain saatavilla kolmannen osapuolen tarjoamana kuukausimaksullisena ohjelmallisena lisäpalveluna (OPC UA Client/Server n.d.), minkä vuoksi sitä ei otettu käyttöön. UR5e käyttöohjeessa robotin kerrotaan tukevan automaatiokenttäväyliä Modbus, Ethernet/IP, PROFINET ja PROFIsafe (Universal Robots 2022, 239).

8.1 Modbus

Modbus on avoin, ilmainen ja yleensä vaivaton teollisen internetin tiedonsiirron standardi. Modbus julkaistiin vuonna 1979 sarjaliikennemuotoiseksi protokollaksi, jonka viestikehys on esitelty kuviossa 5. **Virhe. Viitteen lähde ei löydy.** **Virhe. Viitteen lähde ei löydy.** Siitä on kehitetty myös Ethernetiä käyttävä versio Modbus TCP, joka julkaistiin vuonna 1999. Tavallisesti Modbus-protokollaa käytetään siirtämään jollakin kenttälaitteella mitattua dataa tietokoneeseen. Modbus-protokollan etuna on, että viestintä on tilatonta, minkä ansiosta kommunikointi on erittäin häiriösietoista. Modbus protokollana dominoi automaatiossa vahvasti, mutta erittäin heikon tietoturvan ja protokollan iän takia Modbussista yleisestä käytöstä teollisuus siirtyy hiljalleen pois. (Collin & Saarelainen 2016, 186.)

Tyypillisesti Universal Robots robotti määritetään olemaan Modbus Client (käytetään myös engl. termiä master) ja ulkoinen laite olemaan Modbus Server (käytetään myös engl. termiä slave), mutta robotti voi toimia kumpanakin. Jos robotti määritetään toimimaan Modbus serverinä, muut laitteet voivat lukea ja kirjoittaa dataa robotilta. Modbus-väylällä voi siirtää vain etumerkittömiä 16-bittisiä tietueita, mikä desimaalilukuna vastaa arvoja väliltä 0–65535. Koska siirretyt arvot ovat etumerkittömiä ja kokonaislukuja, saattaa ennen siirtoa tai sen jälkeen joutua tekemään muunnoslaskuja, mikäli halutaan siirtää tosiasiasia negatiivia tai desimaalilukuja. Tällöin voi joutua myös ohjelman sisällä yhdistämään vierekkäisiä muuttujia yhteen, mikä voi tulla lopulta hyvin työlääksi. (Modbus server 2015.)

8.2 EtherNet/IP

EtherNet/IP on 2001 julkaistu Ethernet-pohjainen tiedonsiirtoprotokolla. EtherNet/IP käyttää kommunikoinnissa CIP-protokollaa (engl. Common Industrial Protocol) ja kommunikointi tapahtuu Ethernet-lähiverkkoratkaisua käyttäen, mikä on tuttu tietokoneiden ja verkkolaitteiden kytkemisestä. (ODVA 2021.) Samalla tavalla kuin Modbus-kommunikoinnissa clientit ohjaavat

serverilaitteita, EtherNet/IP:ssä scanner-laitteet lähettävät ja hakevat dataa adapter-laitteilta. Käytännössä protokollossa scanner tarkoittaa masteria ja adapter slavea. EtherNet/IP:n etuna muihin automaatioprotokolliin on, että yksi adapteri voi olla yhteydessä useampaan scanneriin, joten scanner-laitteet voivat päästä käsiksi lukuisten adaptareiden dataan. Toisaalta vain yksi scanner-laite voi suoranaisesti kirjoittaa adapterille dataa, eli ohjata sitä, kun muut scanner-laitteet voivat vain lukea jaettua dataa. (Imhoff, 2021.)

Dataa voidaan lähettää joko explicit- tai implicit-viesteinä. Explicit-viestit ovat TCP-muotoisia viestejä, jossa adapteria pyydetään lähettämään dataa takaisin skannerille. Skannerin lähettämässä pyynnössä määritellään mitä dataa halutaan lähetettäväksi. Explicit-viestit ovat hitaampia kuin implicit-viestit, koska viestissä pitää määritellä tarkasti missä muodossa pyydetty data lähetetään. Implicit-viestit ovat UDP-muotoisia, syklistä lähetettäviä viestejä adaptereille, jonka rakenne on selkeästi määritetty ennalta. Explicit-viesteihin verrattuna data on saatavilla nopeammin ja pienemmillä paketeilla, koska viestissä lähetettävän datan rakennetta ei tarvitse määritellä viestissä, vaan molemmat osapuolet tietävät rakenteen jo ennalta. (EtherNet/IP: Implicit vs. Explicit Messaging.)

Universal Robots -robotti voidaan määrittää EtherNet/IP kenttäväylässä adapterin rooliin. Robotti tukee myös ainoastaan implicit-viestintää. Väylän päivitystaajuus voidaan määrittää välille 2 ms – 3200 ms. Vaikka EtherNet/IP-protokollassa tietokone voi olla scannerin roolissa, sen käyttö siihen tarkoitukseen on harvinaisempaa ja Universal Robots tarjoaa EtherNet/IP:n käyttöönoton ohjeissa esimerkin vain automaation ohjelmitavaan logiikkaohjaimeen. (Ethernet IP guide 2024.)

8.3 PROFINET ja PROFI-safe

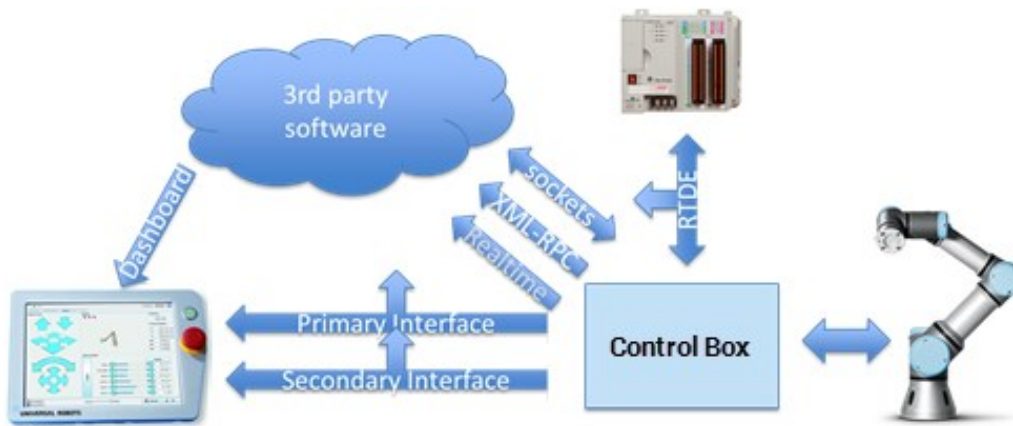
Profinet on teollisuusautomaation tarpeisiin valmistettu standardi, joka pohjautuu vanhempaan Profibus-standardiin. Suurena erona näillä kahdella on, että aiemman kaksijohtimisen parikaapelin sijasta Profinet käyttää nopeampaa Ethernet-pohjaista IEEE-standardia. Profinetiä käytetään pääsääntöisesti tehdasautomaatiossa, prosessiteknikassa ja muussa liikkeenohjauksessa. Erilaisia verkkolaitteita, kuten reitittimiä, on erityisesti valmistettu Profinet-käyttöä varten. Näiden ominaisuus on, että tavalliset Ethernetin komponentit voidaan liittää samaan verkkoon helposti. Profinetistä on tarkemmin vielä kolme protokollaa, Profinet TCP/IP, Profinet RT (engl. Real-Time) ja Profinet IRT (engl. Isochronous Real-Time). (Collin & Saarelainen 2016, 168–169.)

Universal Robots tarjoaa Profinetin käyttöön ottamiseksi ohjeet ainoastaan automaation ohjelmitavaan logiikkaohjaimeseen. Logiikkaohjaimeseen yhdistettynä väylää voidaan käyttää monipuolisesti esimerkiksi lukemaan robotin turvajärjestelmän tilatietoa, seurata ja ohjata digitaali- ja analogi-portteja, lukea robotin käsivarren nivelten ja työkalun paikkaa, sekä lähettää ja vastaanottaa vapaasti muuttujien arvoja bitti-, kokonaisluku- ja desimaalimuodoissa. (PROFINet How-To Guide E-series 2024.)

PROFIsafe protokolla sallii UR-yhteistyörobotin kommunikoinnin turvalogiikan kanssa. PROFIsafe turvaväylän avulla robotti voidaan yhdistää osaksi väylärakenteista turvajärjestelmää. Turvalogiikka valvoo robotin turvapiiriä, jolloin robotin hätäseistilaa tai hidastetun nopeuden tilaa voidaan seurata ja ohjata automaatiojärjestelmällä. (Universal Robots e-Series User Manual 2022, 243.)

8.4 Muut Universal Robots -robottien kommunikointirajapinnat

Perinteisten automaatiokenttäväylien lisäksi Universal Robots yhteistyöroboteihin voidaan yhdistää TCP/IP-protokollan (engl. Transmission Control Protocol ja Internet Protocol) avulla erilaisiin kommunikointirajapintoihin (ks. kuvio 22). Nämä rajapinnat tukevat hyvin tietokoneiden ohjelmistoilla toteutettua kommunikointia, joten niitä käyttämällä perinteisiä automaatiokenttäväyliä ja logiikkaohjaimia ei tarvita. Perinteisten väylien tavoin näiden rajapintojen avulla robotista voi hakea tilatietoja, kuten robotin työkalun sijainnin, sisäisen lämpötilan ja muuttujien arvot. Lisäksi robotille voidaan lähettää komentoja, kuten liikekäskyjä, joten liikeratojen laskenta roboteille voidaan toteuttaa esimerkiksi kokonaan tietokonetta käyttäen.



Kuvio 22. Yleiskatsaus UR-yhteistyörobotin tiedonsiirtorajapinnoista (Overview of client interfaces n.d.)

TCP/IP soketikommunikoinnissa robotti toimii client-laitteena ja yhdistettävä laite server-laitteena. Tieto vaihtuu TCP/IP protokollaa käyttäen. Kommunikointi vaatii edistynyttä ohjelmointiosaamista, koska robotit eivät oletuksena sisällä mitään asetuksia rajapinnan käyttöönotolle. Kommunikoinnin mahdollistamiseksi ohjelmistoa pitää kirjoittaa niin robotille (Universal Robots kutsuu näitä sanalla URScript) kuin palvelimena toimivalle tietokoneelle. Universal Robots tarjoaa esimerkkiohjelmat niin robotille kuin tietokoneelle, jossa C# -kielellä kirjoitetulla ohjelmalla tietokone ohjaa robottia kolmeen eri koordinaattipisteeseen. (TCP/IP socket communication via URScript n.d.)

Universal Robotsin XML-RPC-kommunikoinnissa (engl. Extensible Markup Language ja Remote Procedure Call) XML-kieltä voidaan käyttää siirtämään tietoa ja ohjaukskäskyjä tietokoneen ja robotin välillä sokettiviesteillä. Robotti voi kutsua ulkoisten laitteiden ohjelmien metodeja ja funktiota ja myös toisinpäin, joiden avulla robotin dataa voidaan nopeasti käyttää esimerkiksi monimutkaisiin laskutoimituksiin, joita robotilla ratkaistessa menisi kauan aikaa. (Overview of client interfaces n.d.) TCP/IP soketikommunikoinnin tavoin tiedonsiirtorajapinta käyttäjien täytyy rakentaa alusta loppuun itse. Universal Robots tarjoaa kuitenkin esimerkkiohjelmat robotille ja tietokoneelle, jossa C++ tai vaihtoehtoisesti Python kielellä kirjoitetulla ohjelmalla tietokone ohjaa robotin ohjelman määrittämään koordinaattipisteeseen. (XML-RPC communication n.d.)

Dashboard server -yhteydellä voidaan tehdä pitkälle samoja asioita kuin Universal Robotsin käyttöpaneelillakin. TCP/IP sokettien avulla voidaan muodostaa robotin oletusasetuksilla käytössä olevaan dashboard server -rajapintaan. Robottia voidaan ohjata lähettämällä robotin käyttöpaneelin käyttämien komentojen kaltaisten viestien lähettämiseksi robotille. Rajapinnan kautta voidaan esimerkiksi valita suoritettava robotin ohjelma ja aloittaa tai pysäyttää sen suoritus. Universal Robots ei tarjoa valmiita ohjelmia rajapinnan käyttämiseksi, mutta tarjoaa kuvia ja dokumentteja sen käyttämisestä. (Dashboard Server e-Series, port 29999 n.d.)

Universal Robots robottien primary, secondary ja real-time interface -rajapinnat mahdollistavat robotin kanssa kommunikoimisen TCP/IP sokettiviesteillä nopealla päivitystaajuudella. Näiden rajapintojen yli voidaan monipuolisesti lukea robotin tilatietoja. Kaikki kolme rajapintaa ovat Universal Robots roboteissa oletuksena päällä ja niiden kautta voidaan hakea yleisiä robotin tilatietoja ilman erikseen robotin puolelle tehtävää ohjelmointia. Primary ja secondary interfacen päivitystaajuudet ovat 10 Hz ja real-time interfacella jopa 500 Hz. Universal Robots tarjoaa ohjeita näiden rajapintojen käyttöön, mutta ei kuitenkaan suoraan tue niiden käyttöä eikä tarjoa esimerkkiohjelmiä. (Remote Control Via TCP/IP n.d.) Real-time interface -rajapinta on tunnettu myös nimellä Matlab-rajapinta, mutta sen tuki on loppunut versiosta 3.5 eteenpäin ja Universal Robots suosittelee käyttämään Real-time interfacen sijaan sen korvannutta RTDE:tä. (Realtime Client Interface n.d.)

RTDE-rajapinta (engl. Real-Time Data Exchange) kommunikoi TCP/IP yhteydellä tavallisesti 125 Hz, mutta korkeimmillaan 500 Hz -päivitystaajuudella. Rajapinta on oletuksena robotin päällä ollessa aina käytössä. Sitä voidaan käyttää yksityiskohtaisten robotin tilatietojen lukemiseen ja robotin liikepisteiden ja vapaamuotoisen datan lähettämiseen ja vastaanottamiseen. Yhdistämisen alussa robottiin yhdistävä osapuoli luo listan rajapinnalla kommunikoitavista muuttujista. Listaa kutsutaan myös reseptiksi ja se voi sisältää koordinaatti-, kokonaisluku-, desimaaliluku- ja totuusarvomuttujia missä järjestyksessä vain, kunhan siirrettävien muuttujien koko pysyy yhteensä 2048 tavussa. Mikäli prosessorin kuormittumisen takia robotin kontrolleri ei kykene lähettämään paketteja toivotulla nopeudella, kontrolleri jättää paketteja lähettämättä ja prosessorin kuorman vähentyessä lähettää muuttujien viimeisimmät arvot. Tämän vuoksi nopeasti päivittyvien muuttujien arvoja voidaan menettää. Kontrolleri kuitenkin vastaanottaa ja käsittelee kaikki sille saapuvat

paketit järjestyksessä, mutta hyvin nopean tahtiin saapuvat paketit kuormittavat robotin prosessoria. RTDE rajapinta tukee lukuisia samanaikaisia yhteyksiä, kunhan yhteyksien reseptit eivät sisällä samoja muuttujia. (Real-Time Data Exchange (RTDE) Guide 2019.) Robottien käyttäjät ovat kuitenkin raportoineet kolmen yhteyden olevan maksimimäärä ennen kuin yhteyksien päällä pysymisen kanssa alkaa muodostua ongelmia (Failures with multiple RTDE connections 2018).

9 Tuotetestaus

Tuotetestauksen tehtävänä on varmistaa, että valmistettujen tuotteiden laatu täyttää yhä tuotteille asetetut vaatimukset vuodesta toiseen. Tällaisella vuositestauksella voi esimerkiksi selvittää, että raaka-aineiden tai tuotannon muutosten seurauksena tuotteiden laatu on laskenut. Lisäksi tuotetestaus voi mitata kehitteillä olevien tuotteiden ominaisuuksia. Tällaisella tuotekehitystestauksella varmistetaan jo tuotekehitysvaiheessa olevilla tuotteilla, että valmistuneet tuotteet tulevat saavuttamaan kaikki tuotteen vaatimukset.

”Tuotteemme ja ratkaisumme ovat turvallisia, helppokäyttöisiä ja testattuja – yli sertifioitujen vaatimusten.” (Abloy yleisesitys 2023)

Riippuen lukitustuotteen ominaisuuksista ja käyttötarkoituksesta, testaaminen suoritetaan eri standardin mukaisesti. Esimerkkejä lukitustuotteiden standardeista on listattu taulukkoon 3. Tuotteiden vaatimukseen ja vaatimusten testausmenetelmiä käsittelevät standardit sisältävät määrityksiä useisiin erilaisiin testeihin, kuten kulutustesteihin, palotesteihin, olosuhdetesteihin ja erilaisiin hyökkäystesteihin.

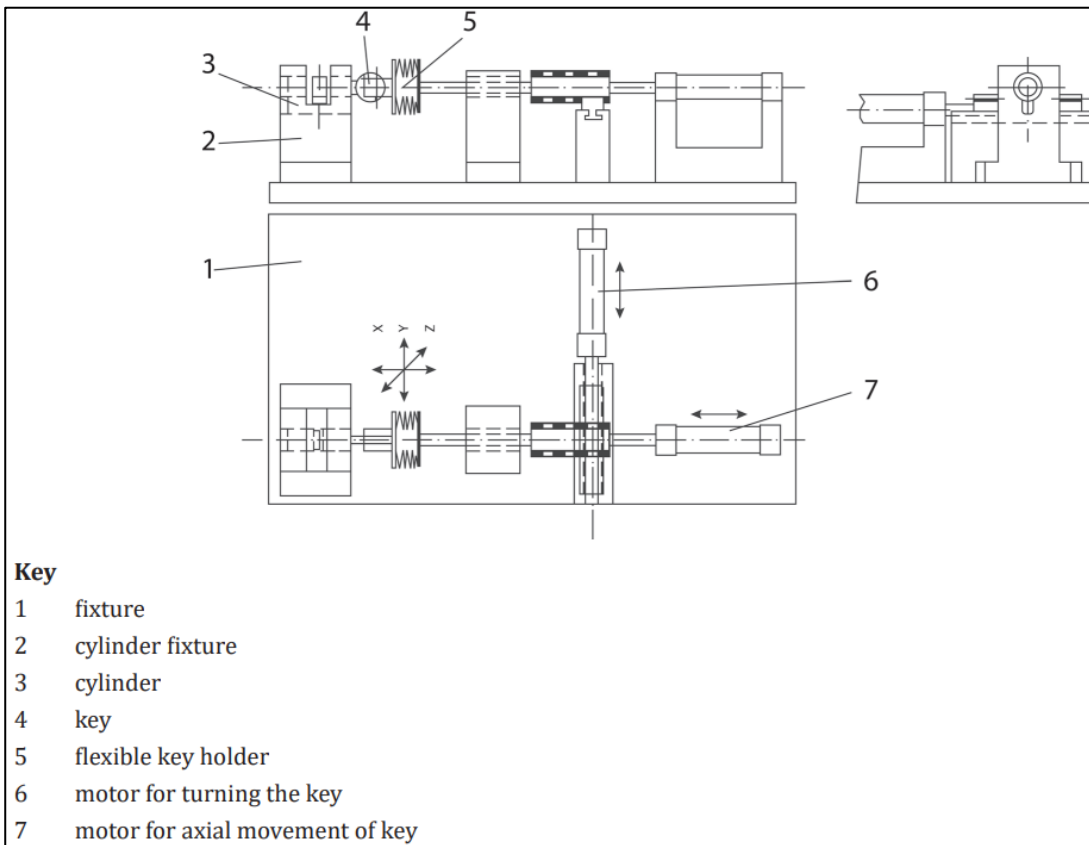
Taulukko 3. Lukitusjärjestelmiin liittyviä standardeja vaatimuksista ja testausmenetelmistä

Standardi	Tuotteet
SFS-EN 179:en	Lukot ja rakennushelat. Painikkeella tai työntölevyllä avattavat poistumisovien lukkolaitteet.
SFS-EN 1125:2008	Lukot ja rakennushelat. Vaakapuomilla avattavat poistumisovien paniikkipoistumislaitteet.
SFS-EN 12209:en	Lukot ja rakennushelat. Mekaaniset lukkorungot, salvat ja vastalevyt.
SFS-EN 14846:en	Lukot ja rakennushelat. Lukkorungot ja salvat. Sähkömekaaniset lukot ja vastalevyt.
SFS-EN 1303:en	Lukot ja rakennushelat. Avainpesät lukoilta.
SFS-EN 15684:2020:en	Lukot ja rakennushelat. Mekatroniset sylinterit.
SFS-EN 12320:2021:en	Rakennushelat. Riippulukot ja niiden korvakkeet.
SFS-EN 16864:2017:en	Lukot ja rakennushelat. Mekatroniset riippulukot.

Abloy Oy tuotetestauksen käytössä olevaa kuutta Universal Robots UR5e yhteistyörobottia käytetään monipuolisesti erilaisiin tuotetesteihin. Robottien yksi pääsääntöisistä käyttökohteista on mekatronisten sylinterituotteiden kulutustestit, mistä syystä tässä opinnäytetyön raportissa esitetään yhteistyörobotiikalla suoritettava SFS-EN 15684:2020 mukainen kulutustestaus. Kulutustestaus on tuotteen kulumiskestävyyden varmistamista käyttämällä tuotetta suuria määriä ja varmistamalla, että tuotteen normaali toiminta jatkuu lukuisista käyttökerroista huolimatta. Tämän opinnäytetyön tarkoitus on luoda tiedonkeruujärjestelmä, jonka avulla kyetään keräämään paljon robottien aikaansaamaa mittadataa. Koska kulutustestien aikana tuotetta käytetään suuria määriä, soveltuvat ne hyvin uuden tiedonkeruujärjestelmän käyttöönottoon ja testaamiseen.

9.1 Yhteistyörobotilla suoritettava mekatronisen sylinterin kulutustesti

Mekatronisten sylinterien SFS-EN 15684:2020 mukaisen kulutustestin tavoitteena on varmistaa, että tuote toimii lukuisien käyttökertojen jälkeen normaalisti. Tavallisesti tuotetta testataan joko satatuhatta tai kaksisataatuhatta kertaa, jonka aikana tuote ei saa hajota tai häiriintyä liian usein. Kuviossa 23 on esitetty mekatronisten sylinterien standardin mallikuva kulutustestilaitteen rakenteesta.



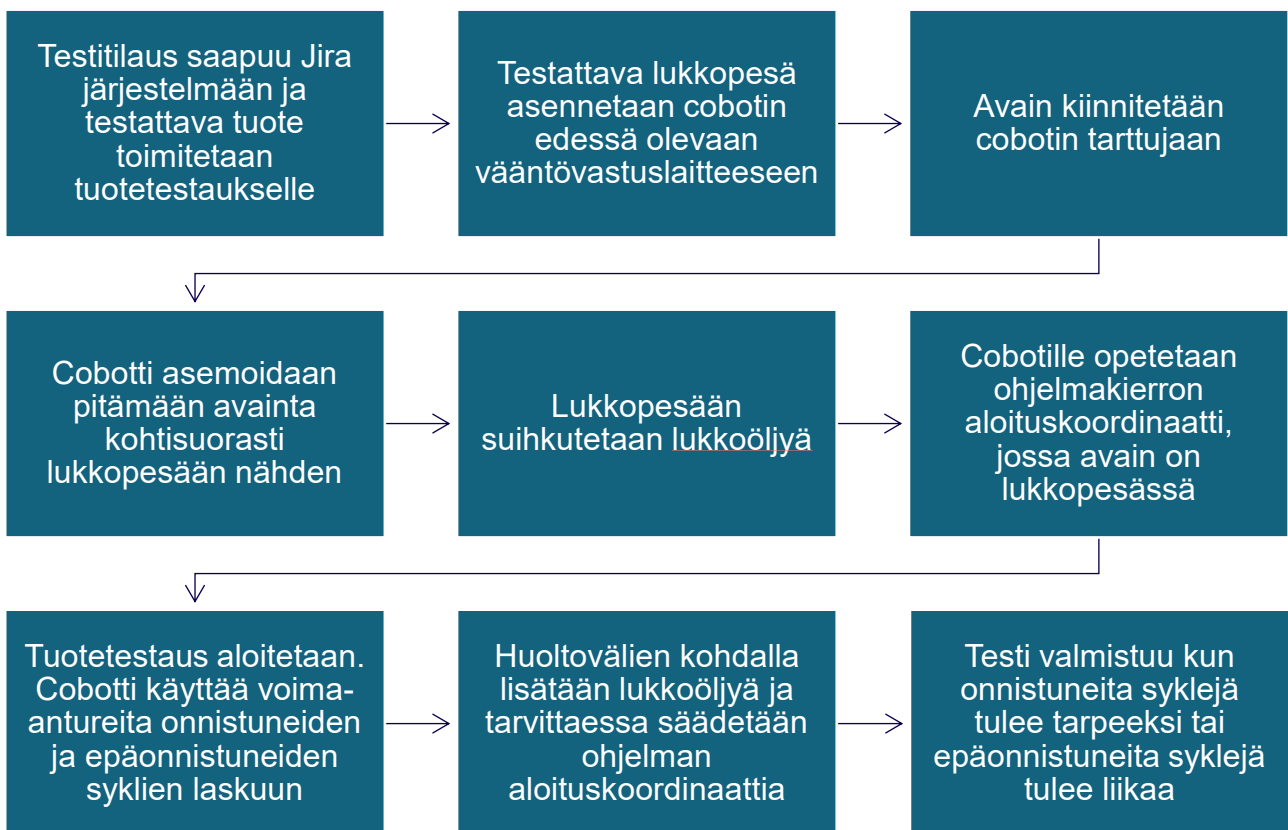
Kuvio 23. Mekatronisten lukitustuotteiden kulutustestilaitteen mallikuva (SFS-EN 15684:2020, 23)

Testin yksi sykli, eli toisinto, koostuu standardin mukaan seuraavista vaiheista:

- a) Avain tulee asettaa lukkosylinteriin tai aktivoida tuote muulla tavalla.
- b) Avainta tulee kääntää myötäpäivään 360-astetta tai sen verran kuin tuotteella on tarkoitettu. Avaimen tai muuhun käännettävään komponenttiin tulee kohdistua vähintään 0,15 Nm voima kääntöliikkeen aikana.
- c) Avain tulee poistaa tai vapauttaa lukitus muulla tavalla.
- d) Avain tulee asettaa lukkosylinteriin tai aktivoida tuote muulla tavalla.
- e) Avainta tulee kääntää vastapäivään 360-astetta tai sen verran kuin tuotteella on tarkoitettu. Avaimen tai muuhun käännettävään komponenttiin tulee kohdistua vähintään 0,15 Nm voima.
- f) Avain tulee poistaa tai vapauttaa lukitus muulla tavalla.

Syklejä saa suorittaa 3–12 kappaletta minuuttia kohden ja vain kokonaan onnistuneet syklit laske-
taan suoritetuiksi. Avain voidaan puhdistaa valmistajan ohjeiden mukaisesti ja käyttää lukkosylin-
teriin tuotteen valmistajan suosittelemaa öljyä testin alussa ja lisäksi viidentuhannen syklin välein.
Kymmentuhannen syklin välein tuotteen normaali toiminta tulee tarkistaa. Normaali toiminta

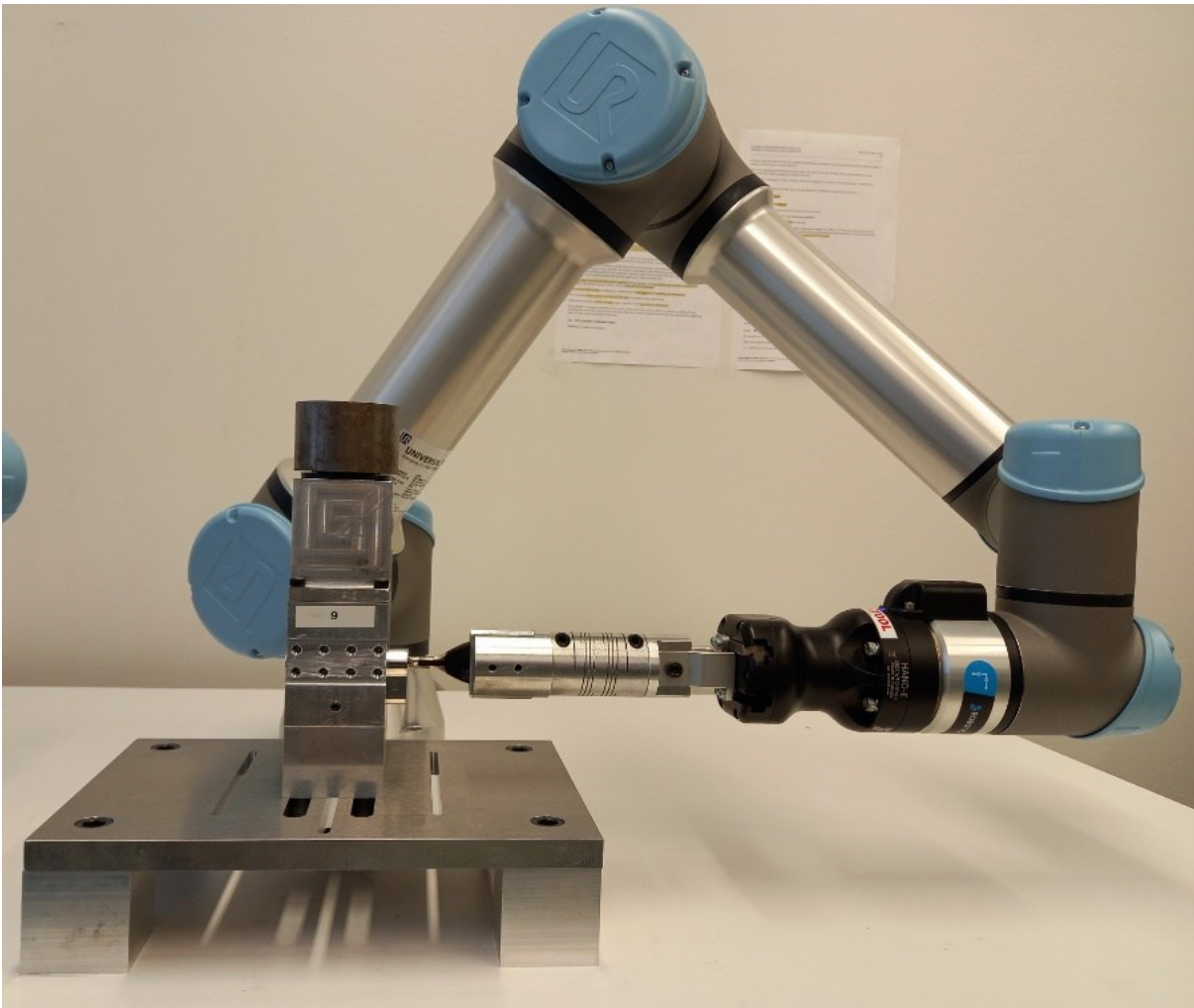
voidaan varmistaa esimerkiksi manuaalisesti käyttämällä ensin avainta, jolla ei ole kulkuoikeutta ja varmistamalla, että lukkosylinteri ei avaudu, ja sen jälkeen avaimella, jolla on kulkuoikeus, jolloin lukkosylinterin tulee avautua. Tuotteen normaali toiminta varmistetaan toistamalla tämä varmistus yhteensä viisi kertaa peräkkäin. Mikäli tuote toimii normaalisti, voidaan testiä jatkaa normaalisti. Kulutustesti on onnistuneesti suoritettu, kun valmistajan asettaman spesifikaation mukainen määrä (25000, 50000 tai 100000) onnistuneita syklejä on saavutettu. (SFS-EN 15684:2020.) Abloy Oy:n tuotetestauksen yhteistyörobotilla suorittamien kulutustestin työvaiheet on esitetty kuviossa 24.



Kuvio 24. Mekatronisen sylinterin kulutustestin toimenpiteet

Abloy Oy:n tuotetestauksen yhteistyörobottien tarttujen ja avainkiinnikkeiden suunnittelussa on huomioitu tuotetestauksen standardeissa asetetut mekaaniset ja ohjelmalliset vaatimukset. Testien aikana yhteistyörobotit valvovat niiden käsivarteen kohdistuvia voima- ja vääntöarvoja, minkä

avulla robotit kykenevät tunnistamaan onnistuneet ja epäonnistuneet syklit toisistaan. Mikäli syklin aikana voima- tai vääntöarvot nousevat odotettua suuremmaksi, tuotteen tila ei ole vaihtunut odotetun mukaisesti, sykli merkitään epäonnistuneeksi ja syklilaskurin sijasta kasvatetaan virhelaskuria. Kuviossa 25 on kuva yhteistyörobotista tekemässä standardin SFS-EN 15684:2020 mukaista mekatronisen lukkosylinterin kulutustestiä.



Kuvio 25. UR5e yhteistyörobotti suorittamassa lukkosylinterin kulutustestiä

9.2 Yhteistyörobottien turvallisuus tuotetestauksessa

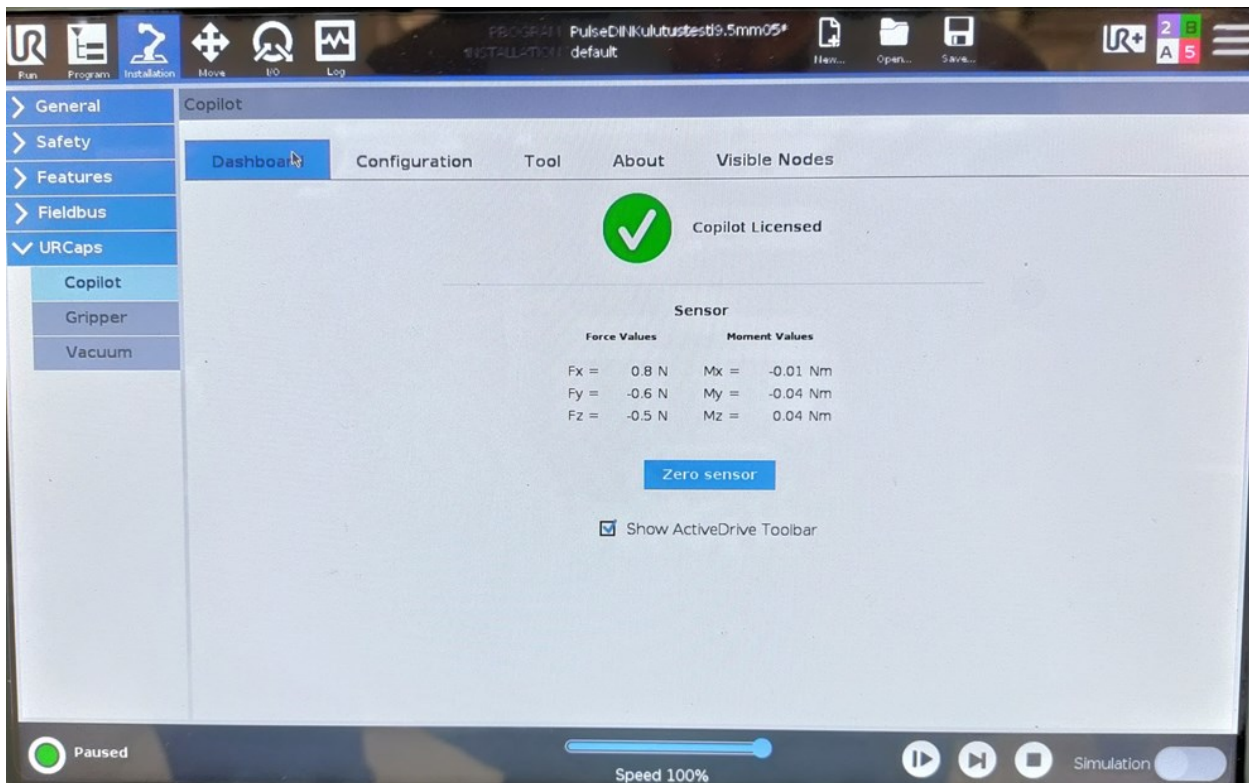
Universal Robots UR5e -yhteistyörobottien turvallisuusominaisuuksien ansiosta työtilaa, jossa yhteistyörobotit sijaitsevat, ei ole tarvinnut aidata. Samassa työskentelytilassa voidaan rinnakkaisyöskennellä yhteistyörobottien kanssa. Vaikka ihminen menisi yhteistyörobottien työskentelyalueelle, vaaraa loukkaantumisesta ei ole. Kiertoliikettä tekevien yhteistyörobottien tarttujissa ei

esimerkiksi ole teräviä reunoja, joihin vaatteet voisivat jäädä kiinni, eivätkä ne ota syklien välissä avainta kokonaan pois lukkopesästä, joten kättä ei myöskään voi laittaa testattavan lukkopesän ja avaimen väliin.

Yhteistyörobotit ovat määritetty käyttämään työskentelyssään ainoastaan niin paljon voimaa, kuin tuotteen testaamisen tehtävään vaaditaan. Yhteistyörobotteihin kohdistuva ylimääräinen vastus tai kuorma pysäyttää ne, joten ne lopettavat saman tien liikkumisen, jos ihminen osuu niihin. Kaiken lisäksi yhteistyörobottien ohjainyksiköiden hätäseispainikkeet ovat robottien välittömässä läheisyydessä, joiden avulla robottien käsivarret saadaan tarpeen tullen virrattomiksi.

9.3 Yhteistyörobotilta talteen kerättävät tiedot

Universal Robots UR5e sisältää itsessään antureita tarttujaan kohdistuvan voiman ja vääntömomentin mittaukseen (UR5e Technical Specification 2023). Antureiden mittaamat arvot siirtyvät robotin käsivarressa olevaa väylää pitkin robotin kontrollerille, jossa niitä voidaan käyttää robotin ohjelmassa. Sylinterituotteiden kulutustestien aikana robottiin tarttujaan kohdistuu robotin tarttujaan kiinnitetyn avaimen suuntaisesti kohtisuuraa voimaa ja sylinterin kiertyvän aukaisuliikkeen aikana vääntömomenttia. Universal Robots yhteistyörobotteihin erikseen myytävänä oleva Robotiq Force Copilot -ohjelmisto nopeuttaa sisäänrakennettujen antureiden arvojen hyödyntämistä robotin ohjelmoinnissa (Robotiq Force Copilot n.d.). Force Copilotin avulla voima- ja vääntöarvot saadaan näkyville robotin ohjauspaneelille kuvion 26 mukaisesti. Voima- ja vääntöarvoja voidaan Force Copilot ohjelmiston tuomien ohjelmakäskeyjen avulla helposti käyttää UR5e -robotin ohjauksessa.



Kuvio 26. UR5e yhteistyörobotin ohjauspaneelin kautta nähtävä Force Copilot-ikkuna

UR5e yhteistyörobottien kulutustestiohjelmiin luotiin taulukon 4 mukaiset muuttujat, joihin tallennettiin informaatiota syklien jälkeen. Muuttujien avulla voitiin seurata, minkälaisia voimia yhteistyörobotin tarttujaan kiinnitetyn avaimen pyörittäminen lukkosylinterissä aiheuttivat. Muuttujiin tallennetut arvot korvautuivat uusilla jokaisen syklin jälkeen. Arvokasta mittatietoa ei vielä siirretty yhteistyöroboteilta minnekään talteen, minkä vuoksi tuotteen ja robotin toiminnasta saatu informaatio ei ollut vielä erityisen hyödyllistä.

Taulukko 4. UR5e yhteistyöroboteille määritetyt muuttujat sylinterituotteiden testauksen informaation keräämiseksi

Muuttujan nimi	Selitys
Cycles	Testin aikana tapahtuneiden syklien määrä.
Errors	Testin aikana tapahtuneiden virheiden määrä.
ErrorCode	Viimeisimmällä syklillä tapahtuneen virheen numero.
TimeToMaintMins	Syklimäärän, sykliajan ja huoltovälin perusteella aika-arvio minuutteina milloin seuraava huoltoväli tapahtuu
TorqueCWMax	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana myötäpäivään mitattu maksimiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCcWMax	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana vastapäivään mitattu maksimiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCWAver	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana myötäpäivään mitattu keskiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCcWAver	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana vastapäivään mitattu keskiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
ForceTurnCWAver	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana myötäpäivään mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCWMax	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana myötäpäivään mitattu maksimiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCcWAver	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana vastapäivään mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCcWMax	Viimeisimmällä syklillä avaimen kiertoliikkeen aikana vastapäivään mitattu maksimiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForcePushAver	Viimeisimmällä syklillä avaimen työntöliikkeen aikana mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForcePushEnd	Viimeisimmällä syklillä avaimen työntöliikkeen jälkeen mitattu robotin laipan mukaisesti kohdistunut voima newtoneina.

10 Tiedonkeruujärjestelmän toteutus

Toteutettavan tiedonkeruujärjestelmän kommunikointirajapinnaksi robotin suunnalta valittiin TCP/IP-pohjainen Universal Robotsin RTDE-rajapinta. Valinta perustui siihen, että RTDE-rajapinnan käyttämisestä löytyi hyvin informaatiota ja malliesimerkkejä, sekä sen avulla pystyy lähettämään ja vastaanottamaan monipuolisesti erilaisia tietorakenteita. Lisäksi rajapinta oli vahvasti robotin valmistajan tukema ja sen käyttö oli ilmaista, toisin kuin esimerkiksi OPC UA -rajapinta. RTDE-rajapinnalla ei myöskään rajoittanut desimaalilukujen lähettämistä, kuten esimerkiksi Modbus-rajapinta, minkä ansiosta se soveltui erinomaisesti voima-arvojen keräämiseen. RTDE-rajapinta on UR-roboteissa oletusasetuksilla valmiiksi aktiivisena, joten erillisiä määrittämiä rajapinnan määrittämiseen ja käyttöönottoon robotin puolelle ei tarvinnut tehdä. Robotille tuli ainoastaan määrittää, mitkä muuttujat kirjoitetaan mihinkin RTDE-rajapinnan reseptin mukaiseen kommunikointirajapinnan rekisteriin.

Tietovarastoksi valittiin opinnäytetyön toimeksiantajan jo ennestään käyttämä Microsoft SQL-tietokanta. Valinta perustui kerättävän datan hyvää soveltuvuutta SQL:n strukturoituun taulukkorakenteeseen ja toimeksiantajan kykyyn tarjota tietokannan käyttöönotto jo valmiiksi olevilla tietokantarakenteilla. Toimeksiantajan aiemman kokemuksen ansiosta tietokannan käyttöönotto ja siihen yhdistäminen sisäverkon kautta oli toteutettavissa sujuvasti.

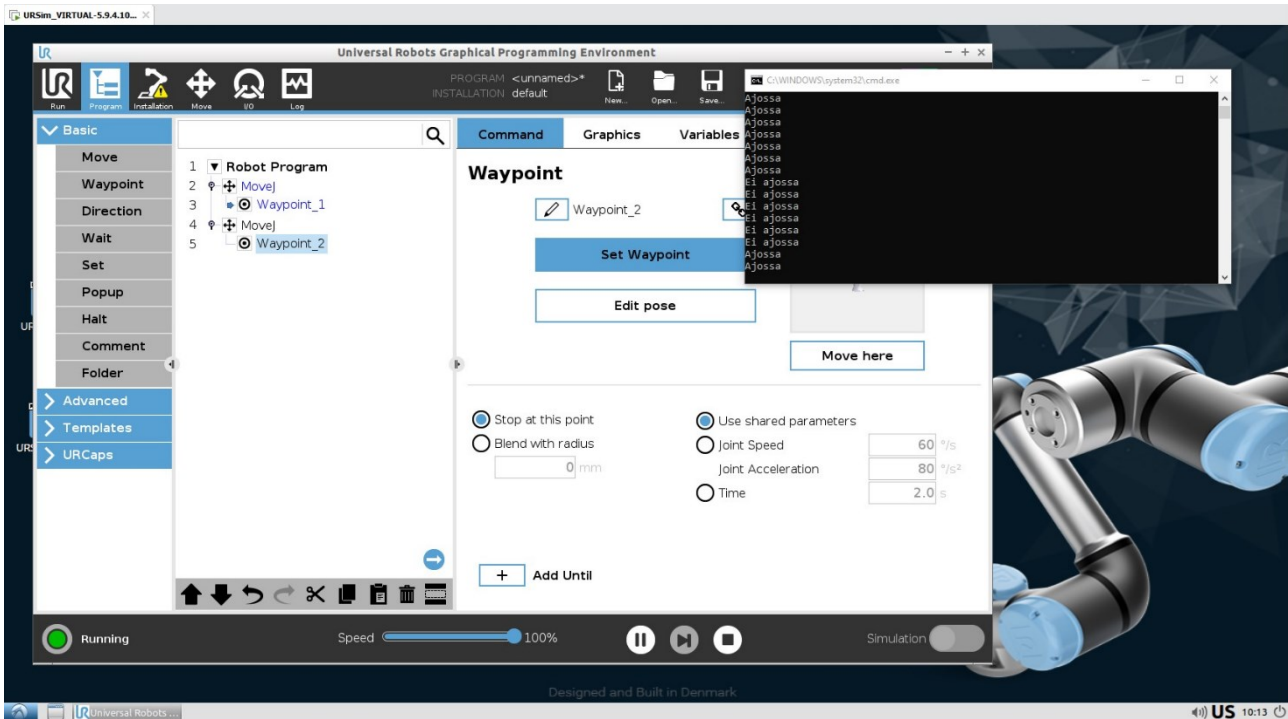
Tiedonkeruuohjelman ohjelmointikieleksi valittiin Python 3.10. Universal Robotsin RTDE-esimerkkiohjelmat ovat tehty saman ohjelmointikielen vanhemmalle Python 2.7.11 versiolle, jolloin niitä pystyi käyttämään avuksi tiedonkeruuohjelmaa rakentaessa. Python ohjelmointikieli oli myös toimeksiantajalle entuudestaan tuttua ja jokaisella Abloy Oy:n tuotetestauksen työntekijän tietokoneella oli Python 3.10 jo asennettuna. Tiedonkeruuohjelman ottaminen yleisesti käyttöön ei siis vaatisi uusien ohjelmistojen asentamista, vaan korkeintaan uusien Python moduuleiden lataamista.

10.1 Virtuaalisen yhteistyörobotin RTDE-rajapintaan yhdistäminen Python-ohjelmointikielellä

Universal Robots tarjoaa RTDE Python Client -kirjaston, jossa on toimivia esimerkkejä RTDE-rajapintaa käyttävistä ohjelmista. Kirjasto sisältää muun muassa ohjelman "record.py", jolla voidaan tallentaa robotin lähettämää dataa ja tallentaa sitä CSV-tiedostoon ja ohjelman "example_control_loop.py", jonka avulla voidaan ohjata robotin liikeratoja. Kirjaston ohjelmat tarjoavat kattavan valikoiman toimivia esimerkkejä, joita hyödyntämällä voi ohjelmoida Python-ohjelman omiin käyttötarpeisiin. (Real-Time Data Exchange (RTDE) Guide 2019; RTDE_Python_Client_Library 2022.)

Universal Robots myös tarjoaa tiedostot heidän kotisivunsa "Downloads"-osiossa nimellä "Offline Simulator - e-Series - UR Sim for non Linux" virtuaalikoneen asentamiseksi. Tämä virtuaalikone sisältää valmiiksi asennettuna ohjelman, jolla voi virtuaalisesti simuloida Universal Robotsin robotteja. Robottisimulaattorin käyttöliittymä on sama kuin oikeallakin robotilla. Simulaattori tukee useita oikean robotin ominaisuuksia, kuten tiedonsiirtorajapintoja. Tiedonkeruujärjestelmän kehittämistyön aikana oli merkittävä apu, että RTDE-rajapintaa pystyi testaamaan omalla tietokoneella ilman fyysistä robottia.

Robottisimulaattori mahdollisti paljon aikaa vievän ohjelmiston kehitys- ja testausprosessin suorittamisen ilman roboteille aiheutettua työajan menetystä. Kuviossa 27 on kuvankaappaus kehitysvaiheen alussa tehdystä Python-ohjelmasta, joka luki virtuaalikoneella olevan robottisimulaattorin RTDE-rajapintaa. Luodun Python ohjelman pohjana käytettiin Universal Robotsin tuottamaa "record.py"-esimerkkiohjelmaa. Muokattu Python ohjelma tulosti konsoliin tekstiä "Ei ajossa", kun virtuaalirobotin pysäytti simulaattorin pysäytyspainikkeella, ja tekstiä "Ajossa", kun virtuaalirobotin ohjelman laittoi takaisin päälle.



Kuvio 27. Kommunikointirajapinnan testaus virtuaaliympäristössä

RTDE-rajapintaan yhdistävää Python-ohjelman kehitystä jatkettiin, jonka seurauksena sillä pystyi lukemaan robotin tilatiedon lisäksi sen muuttujien arvoja. Tämä merkittävä edistysaskel tarkoitti, että tietokoneohjelmiston avulla yhteistyörobotilta voitaisiin kerätä laskureiden ja voima-arvojen kaltaista informaatiota. Ensimmäiset siirrettävät muuttujat nimettiin olemaan ”Tilausnumero”, ”Sykkit”, ja ”Virheet”. Python-ohjelmaa muokattiin myös sallimaan yhteys useaan robottiin samanaikaisesti (ks. kuvio 28).

```

C:\WINDOWS\system32\cmd.exe
Cobot1: Ei ajossa (pause)
Tilausnumero: 9075      Sykkit: 548      Virheet: 5

Cobot2: Ajossa
Tilausnumero: 9078      Sykkit: 5      Virheet: 0

```

Kuvio 28. Päivitetty Python ohjelmalla kykeni ottamaan yhteyden kahteen eri RTDE-rajapintaan

10.2 Julkiseen käyttöön valmistettu Python-ohjelma

Liitteessä 1 on Python-ohjelman koodi, jonka avulla saa kuvion 28 kaltaisen näkymän. Ohjelmaa pystyy käyttämään yhdistämään virtuaalisiin tai todellisiin Universal Robots -yhteistyörobotteihin. Ohjelman alussa tulee määrittää "cobotlist[]" sanakirjaan niiden robottien tiedot, joihin halutaan ottaa yhteys. Yhdistettävien robottien määrää pystyy muokkaamaan ohjelman alussa olevan sanakirjan tietueiden määrällä lisäämään tai vähentämään tarpeen mukaan, jolloin ohjelma toiminta mukautuu robottien määrän mukaisesti. Jotta liitteen 1 ohjelma toimii oikein, Python-ohjelma vaatii RTDE Python Client -kirjaston "rtde" kansion samaan tiedostonsijaintiin. Sen lisäksi kirjaston "record_configuration.xml" tiedosto tulee olla samassa tiedostonsijainnissa ja siihen tulee määrittää robotilta haettavien muuttujien tietotyypit (ks. kuvio 29), eli määrittää RTDE-rajapinnan resepti. Liitteen 1 Python-ohjelman vaatimat Universal Robotsin tuottamat tiedostot ovat ladattavista sijainnista https://github.com/UniversalRobots/RTDE_Python_Client_Library.

```

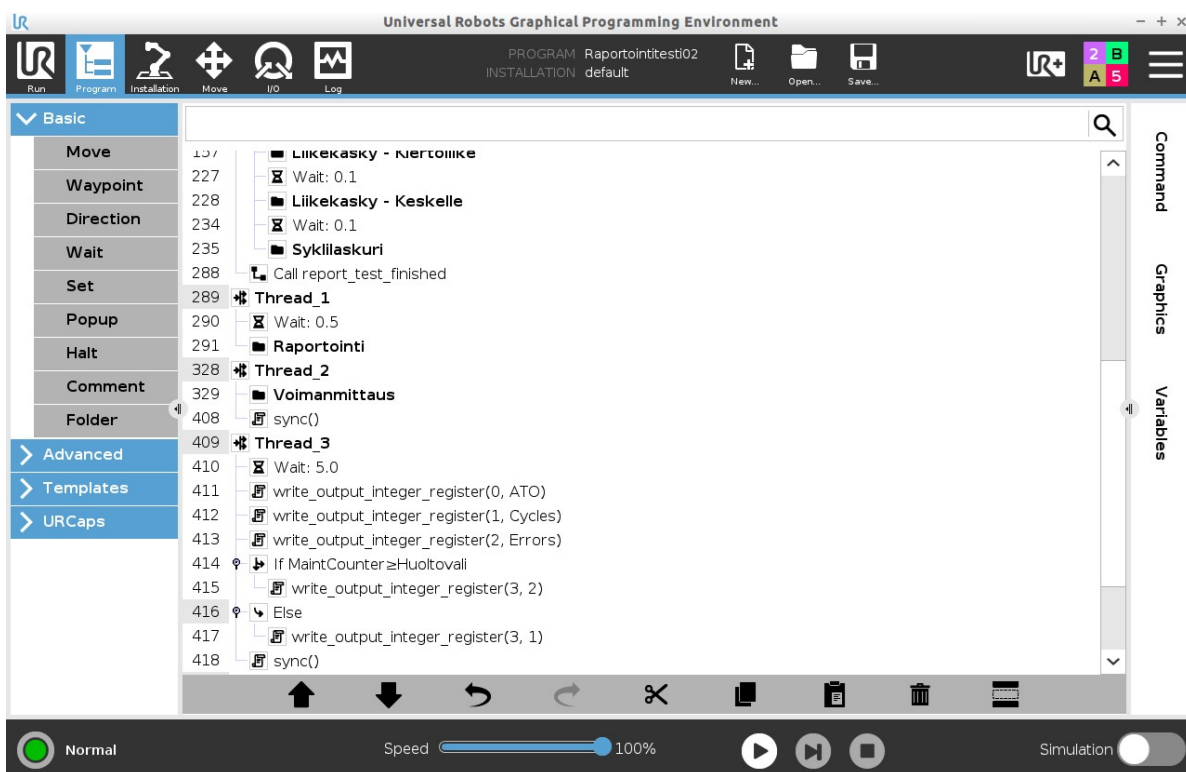
1  <?xml version="1.0"?>
2  <rtde_config>
3    <recipe key="out">
4      <field name="timestamp" type="DOUBLE"/>
5      <field name="runtime_state" type="UINT32"/>
6      <field name="output_int_register_0" type="INT32"/>
7      <field name="output_int_register_1" type="INT32"/>
8      <field name="output_int_register_2" type="INT32"/>
9      <field name="output_int_register_3" type="INT32"/>
10     <field name="output_bit_register_64" type="BOOL"/>
11   </recipe>
12   <recipe key="in">
13     <field name="input_bit_register_64" type="BOOL"/>
14   </recipe>
15 </rtde_config>

```

Kuvio 29. RTDE-kommunikoinnin muuttujamääritykset, eli rajapinnan resepti

Kun Python-ohjelma käynnistyy oikein ja se saa muodostettua yhteyden robottiin (tai robotteihin), se näyttää reaaliajassa missä tilassa robotti on ("Ajossa" tai "Ei ajossa"). Muut näytettävät arvot (tilausnumero, syklit ja virheet) pysyvät arvoina 0, kunnes robotin ohjelmassa arvoja asetetaan kuviossa 29 määritettyihin RTDE-rekistereihin. Kuviossa 30 nähtävissä olevalla robotin ohjelman rivillä 409 on viiden sekunnin välein suoritettava ohjelmasäie. Ohjelmasäikeen riveillä 411–417 siirretään arvoja RTDE-rajapinnan reseptin mukaisiin rekisteripaikkoihin. Kuviossa näkyvät "ATO", "Cycles" ja "Errors" ovat ohjelman robotin kokonaislukumuuttujia, joihin robotti määrittää arvot pääohjelmassa, mutta näiden muuttujien tilalla voi olla muitakin kokonaislukutyypisiä muuttujia

tai jopa suoraan pelkkiä arvoja. Kun robotin ohjelmaan käynnistää ja kokonaislukurekistereihin 0–3 siirtyy arvoja, tulevat ne näkyviin Python-ohjelman näkymään.



Kuvio 30. UR5e -robotin ohjelma laittaa muuttujien arvoja luettavaksi RTDE-rajapintaan

10.3 Yhdistäminen todellisiin yhteistyörobotteihin

Python-ohjelman toiminta oikeilla roboteilla testattiin asentamalla Python-ohjelma palvelintietokoneelle, josta oli muodostettu verkkoyhteys kuudelle eri robotille. Ohjelma onnistui ottamaan samanaikaisesti yhteyden robottien RTDE-rajapintaan ja se näytti robottien tilat ja luettujen rekistereiden arvot oikein. Kuviossa 31 nähtävässä näkymässä näytettiin robottien muuttujat "ATO", eli Abloy Test Order -tilausnumero, "Syklit", eli kuinka monta toistoa robotit olivat suorittaneet, ja "Virheet", eli kuinka monta toistoa robotit olivat keskeyttäneet voima-arvojen odottamattoman kasvamisen takia. Yksi yhdistettävistä roboteista oli testin aikana virrattomana (CobotJoukahainen), jonka vuoksi tilana näkyy "Yhteysvirhe".

```

Robottien tila
2023-12-13 07:14:58.381931
CobotVainamoinen: Ajossa
ATO: 9269      Syklit: 115075  Virheet: 27
CobotIlmarinen: Ajossa
ATO: 9276      Syklit: 81021   Virheet: 39
CobotAino: Ei ajossa
ATO: 9323      Syklit: 9999    Virheet: 73
CobotMarjatta: Ei ajossa
ATO: 9078      Syklit: 17898   Virheet: 28
CobotKyllikki: Huoltovali
ATO: 9371      Syklit: 5001    Virheet: 12
CobotJoukahainen: Yhteysvirhe

```

Kuvio 31. Palvelimella suoritettu Python-ohjelma yhdistettynä oikeisiin robotteihin

Kun kommunikaatio RTDE-rajapintaan yhdistävän Python-ohjelman ja robottien välillä oli todettu toimivaksi, laajennettiin siirrettävien muuttujien määrää ottamalla mukaan tuotetestien aikana kerättäviä voima-arvoja. Yhteistyörobottien ohjelmakierron aikana voima-arvoja seurataan ja ne tallennetaan ohjelmamuuttujiin ja siirretään RTDE-rajapinnalta luettavaksi jokaisen syklin lopussa. Muuttujien ja laskureiden arvoja ei haluttu ottaa talteen kesken syklin, eikä myöskään aivan jokaisen syklin jälkeen, minkä takia yhteistyöroboteille luotiin tiedonkeruuohjelman seuraama ohjelmallinen tapahtuma. Robottien ohjelmaan määritettiin, kuinka usein tiedot halutaan tallentaa. Robottiohjelmassa määritettiin, että 100 onnistuneen syklin jälkeen tehtiin ilmoitus palvelintietokoneella olevalle Python-ohjelmalle asettamalla yksi RTDE-rajapintaan määritetyistä BOOLEAN-muuttujista arvoon TOSI. Tämän muuttujan avulla tiedonkeruuohjelmalle voitiin tehdä ilmoitus, kun sykli epäonnistui, testissä saavutettiin huoltoväli, tai testi valmistui.

Syklitiedon tallentamisen käskyn antamisen jälkeen robotit määritettiin odottamaan vastausta palvelimen Python-ohjelmalta. Vastauksen saamisen jälkeen robotit jatkoivat ohjelman suorittamista. Koska tiedonkeruujärjestelmä oli testivaiheessa herkkä häiriöille ja ajoittaiset palvelimen päivitykset olisivat aiheuttaneet roboteilla suoritettavien tuotetestien pitkäaikaisia pysähtymisiä, määritettiin robotit odottamaan Python-ohjelman vastausta enintään viisi sekuntia. Viiden sekunnin odottelun jälkeen robotit määritettiin kirjoittamaan syklin tiedot paikalliseen tapahtumalokiin ja

jatkamaan, vaikka vastausta ei tullut. Paikallisesta tapahtumalokista syklin tiedot voidaan noutaa robotilta jälkikäteen siirtämällä loki USB-muistitikulle tai noutamalla se palvelimen FTP-yhteydellä.

10.4 SQL-integraatio

Palvelimella suoritettava Python-ohjelma kehitettiin tekemään robottien pyytäessä tallennus Microsoft SQL-tietokantaan. Tietokantaan perustettiin tiedon tallennusta varten kaksi erillistä taulua. Ensimmäiseen CobotStatus-tauluun (ks. taulukko 5) haluttiin tallentaa ainoastaan yhteistyörobottien tilatiedot, sykli- ja virhemäärät. Lisäksi robotit määritettiin lähettämään Python-ohjelmalle syklimäärän, sykliajan ja huoltovälin perusteella aika-arvion, milloin seuraava tuotetestin huoltoväli tapahtuu. CobotStatus-tilussa haluttiin näyttää yhteistyörobottien reaaliaikainen tilannetieto ilman historiadataa, joten Python-ohjelma määritettiin kirjoittamaan tähän SQL-tietokannan tauluun yhteistyörobottien lähettämät tiedot SQL-kielen UPDATE-komennolla, joka ylikirjoittaa tietokannassa jo valmiiksi olevat tietueet.

Taulukko 5. CobotStatus-tilun rakenne

Tietueen nimi	Tietueen tyyppi	Selitys
Cobot	Varchar (50)	Yhteistyörobotin nimi.
ATO	INT	Meneillä olevan ohjelman tunnistus, testinumero.
Cycles	INT	Tämänhetkisen testin syklien määrä.
Errors	INT	Tämänhetkisen testin virheiden määrä.
NextMaint	Datetime	Robotin laskema arvio seuraavasta huoltovälistä.

Toiseen SQL-tietokantaan määritettyyn tauluun haluttiin tallentaa tarkasti niiden syklien tiedot, joilta yhteistyörobotit tekivät Python-ohjelmalle ilmoituksen. CobotData-tiluun (ks. taulukko 6) määritettiin tehtäväksi uusi rivi jokaista ilmoitusta kohden. CobotData-tilussa olemassa olevaa dataa ei haluttu missään vaiheessa ylikirjoittaa, joten Python-ohjelma määritettiin tekemään tauluun tiedon siirtämisen SQL-kielen INSERT-komennolla, joka luo tauluun uuden rivin muuttamatta mitään taulun aiempaa dataa.

Taulukko 6. CobotData-taulun rakenne

Tietueen nimi	Tietueen tyyppi	Selitys
Timestamp	Datetime	Aikaleima
Cobot	Varchar (50)	Yhteistyörobotin nimi.
ATO	INT	Ohjelman tunniste, testinumero.
Cycles	INT	Syklien määrä.
Errors	INT	Virheiden määrä.
ErrorType	Varchar (50)	Virheen nimi tai numero. Mikäli tietue ei ole luotu virheen seurauksena, arvo on 0.
TorqueCWMax	Decimal (18, 2)	Kiertoliikkeen aikana myötäpäivään mitattu maksimiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCcWMax	Decimal (18, 2)	Kiertoliikkeen aikana vastapäivään mitattu maksimiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCWAver	Decimal (18, 2)	Kiertoliikkeen aikana myötäpäivään mitattu keskiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
TorqueCcWAver	Decimal (18, 2)	Kiertoliikkeen aikana vastapäivään mitattu keskiarvo tarttujaan kohdistuneesta vääntövoimasta newtonmetreinä.
ForceTurnCWAver	Decimal (18, 2)	Kiertoliikkeen aikana myötäpäivään mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCWMax	Decimal (18, 2)	Kiertoliikkeen aikana myötäpäivään mitattu maksimiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCcWAver	Decimal (18, 2)	Kiertoliikkeen aikana vastapäivään mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForceTurnCcWMax	Decimal (18, 2)	Kiertoliikkeen aikana vastapäivään mitattu maksimiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForcePushAver	Decimal (18, 2)	Avaimen työntöliikkeen aikana mitattu keskiarvo kohtisuoraan robotin laipan mukaisesti kohdistuneesta voimasta newtoneina.
ForcePushEnd	Decimal (18, 2)	Avaimen työntöliikkeen jälkeen mitattu robotin laipan mukaisesti kohdistunut voima newtoneina.

Kaikki kuusi yhteistyörobottia alkoivat onnistuneesti siirtämään palvelintietokoneella suoritettua Python-ohjelmiston avulla tuotetestauksesta kerättyä dataa tietokantaan. Kuviossa 32 on kuvan kaappaus Microsoft SQL Server Management Studio -ohjelman näkymästä, jossa vasemmalla on StatusTable- ja oikealla DataTable-taulu. Kerätystä datasta haluttiin pystyä luomaan tuotetestikohtaisia raportteja, joista pystyisi seuraamaan tuotteiden ja yhteistyörobottien suoriutumista testien aikana.

The screenshot shows two SQL queries and their results in Microsoft SQL Server Management Studio.

Left Window (SQLQuery13.sql):

```
SELECT TOP (1000) [Cobot]
, [ATO]
, [Status]
, [NextMaint]
, [Cycles]
, [Errors]
FROM [COBOTDATA].[dbo].[StatusTable]
ORDER BY NextMaint ASC
```

Cobot	ATO	Status	NextMaint	Cycles	Errors
1 CobotAino	9715	Ajossa	2024-01-26 09:53:28.423	171748	179
2 CobotJoukahainen	9684	Ajossa	2024-01-26 13:16:28.673	90326	238
3 CobotVainamoinen	9713	Ajossa	2024-01-26 13:42:28.080	85310	56
4 CobotKyllikki	9219	Ajossa	2024-01-28 09:24:13.120	11256	0
5 CobotIimarinen	9220	Ajossa	2024-01-28 09:32:28.533	10984	1
6 CobotMarjatta	10410	Ajossa	2024-01-30 11:57:15.687	1457	0

Right Window (SQLQuery12.sql):

```
SELECT TOP (1000) [Timestamp]
, [Cobot]
, [ATO]
, [Cycles]
, [Errors]
, [ErrorType]
, [TorqueCWMax]
, [TorqueCcWMax]
, [TorqueCWAver]
, [TorqueCcWAver]
, [ForceTurnCWMax]
, [ForceTurnCWAver]
, [ForceTurnCWMax]
, [ForceTurnCWAver]
, [ForcePushAver]
, [ForcePushEnd]
FROM [COBOTDATA].[dbo].[DataTable]
```

Timestamp	Cobot	ATO	Cycles	Errors	ErrorType	TorqueCWMax	TorqueCcWMax	TorqueCWAver	TorqueCcWAver	ForceTurnC
1 2024-01-25 14:13:10.547	CobotKyllikki	9219	11300	0	0	0.59	0.00	0.30	0.12	7.76
2 2024-01-25 14:06:28.683	CobotMarjatta	10410	1500	0	0	0.70	0.22	0.54	0.09	6.81
3 2024-01-25 14:05:06.350	CobotIimarinen	9220	11000	1	0	0.24	0.02	0.11	0.05	5.38
4 2024-01-25 13:51:48.397	CobotMarjatta	10410	1400	0	0	0.72	0.22	0.53	0.09	6.44
5 2024-01-25 13:43:40.643	CobotKyllikki	9219	11200	0	0	0.60	0.00	0.30	0.10	7.39
6 2024-01-25 13:37:07.583	CobotMarjatta	10410	1300	0	0	0.71	0.22	0.53	0.09	6.50
7 2024-01-25 13:36:07.130	CobotIimarinen	9220	10900	1	0	0.26	0.00	0.11	0.05	4.29
8 2024-01-25 13:22:26.390	CobotMarjatta	10410	1200	0	0	0.68	0.22	0.52	0.09	6.14
9 2024-01-25 13:14:10.317	CobotKyllikki	9219	11100	0	0	0.60	0.00	0.30	0.09	6.83

Kuvio 32. Näkymä SQL-tietokannan tauluista ja niissä olevasta datasta

Raporttien tekemiseksi luotiin uusi Python-ohjelma, joka ensin kysyy käyttäjältä minkä testin tiedot halutaan hakea SQL-tietokannasta ja tekee vastauksen ja DataTable-taulun rivien perusteella yhteenvedon testistä Word-dokumenttiin. Liitteessä 2 on raportinteko-ohjelman avulla luotu raportti tuotteen kulutustestistä. Liitteenä olevan raportin ensimmäinen sivu on luotu vanhemmalla ohjelmalla, joka noutaa suoritettua kulutustestin tietoja Jira-järjestelmästä. Tämän opinnäytetyön aikana tehty Python-ohjelma lisää ensimmäisen sivun jälkeiset sivut raporttiin. Tiedonkeruujärjestelmän tallentamia tietoja käyttäen ohjelma liittää raporttiin tapahtumalokin lisäksi graafisia näkymiä testin etenemisestä ja lisäksi kirjoittaa tekstinä selitykset näytettävistä graafeista. Tähän julkiseen opinnäytetyöhön liitteenä 2 olevan raportin ensimmäiseltä sivulta on poistettu Abloy Oy:n tuotetestin projektiluontoiset tiedot, mutta sivusta 2 ja siitä eteenpäin on muokkaamattomana opinnäytetyön aikana luotujen ohjelmien tuottama raportti.

11 Lopputulokset ja pohdinta

Työn lopputuloksena oli toimiva tiedonkeruujärjestelmä, jonka avulla Abloy Oy:n tuotetestaus kykenee keräämään yhteistyöroboteilta yksityiskohtaista mittadataa testattavien tuotteiden toimivuudesta ja ominaisuuksista. Tiedonkeruujärjestelmän lisäksi aikaan saatiin toimiva ohjelma tekemään kerätyn datan perusteella visuaalisia ja helppolukuisia raportteja. Työn aikana luotiin myös valvontanäkymä, jonka avulla Abloy Oy:n tuotetestauksen työntekijät voivat etänä seurata yhteistyörobottien tilaa ja tuotetestien etenemistä. Aikaan saatujen järjestelmien on koettu lisäävän tuotetestauksen tulosten laadukkuutta ja kerätyn datan avulla on jo kyetty havaitsemaan keinoja testausmenetelmien kehittämiseksi.

Merkittävin havainto kerätyn datan perusteella oli, että jokaisella yhteistyörobotilla sylinterituotteiden kulutustestien aikana voima-arvo ForcePushEnd alkaa laskemaan kulutustestin etenemisen myötä. Tämä voi tarkoittaa, että syklien kasvaessa yhteistyörobotti ei enää liikuta avainta yhtä syvälle testattavan sylinterituotetta avainkanavaan kuin testin alussa. Toinen vaihtoehto on, että testattavan sylinterituotteen avainkanavan pohja siirtyy pois päin robotista, jolloin joko tuote puristuu kasaan tai tuote liikkuu kiinnityksistään taaksepäin. Tämä voima-arvo havaittiin olevan merkittävä tuotteiden toimivuuden kannalta. Jos ForcePushEnd-arvo pääsi ajan myötä laskemaan liian paljon, tuotteet ei enää toimineet yhtä luotettavasti.

Raportointijärjestelmän avulla pystyttiin selvästi näkemään, että silloin kun ForcePushEnd-arvo alkoi laskemaan, alkoi kulutustestissä virheiden määrä kasvaa. Havainnon pohjalta yhteistyörobottien operaattoreita ohjeistettiin huoltovälien aikana varmistamaan, että ForcePushEnd-arvo on riittävän suuri ja tarvittaessa muuttamaan yhteistyörobotin koordinaattipistettä niin, että riittävän suuri voima saavutetaan. Ohjeistuksen myötä tuotteiden toimintavarmuus parani testien aikana ja yhteistyöroboteilla tehtävien sylinterituotteiden kulutustestien virheiden määrä väheni. Kerätyn datan ja sen analysoinnin ansiosta prosessiin pystyttiin heti tekemään selkeästi parantava säätötoimenpide.

Työn aikana datan keräystä varten käyttöön otettu palvelintietokonetta kyettiin hyödyntämään myös uudella tavalla. Tiedonkeruujärjestelmän luomisen ohella tehtiin myös Python-ohjelma, jota käyttäen yhteistyörobottien ohjelmat voidaan ladata palvelimelle ja jakaa takaisin roboteille. Tämä

robottiohjelmaa siirtävä ohjelma luo palvelintietokoneelta FTP-yhteyden jokaiseen yhteistyörobottiin ja lataa niistä löytyvät robottiohjelmat palvelimen tiedostonsijaintiin. Robottiohjelmien lataamisen jälkeen Python-ohjelma lataa ne ohjelmat yhteistyöroboteille, mitkä niiltä puuttuvat. Tämän avulla esimerkiksi "Avaimenkaanto01"-nimisestä ohjelmasta yhteistyörobotin käyttäjä voi yhdellä robotilla tehdä parannellun version, nimetä se olemaan "Avaimenkaanto02", ja suorittaa palvelimen robottiohjelmaa siirtävä Python-ohjelma, joka siirtää ohjelman kaikille muille yhteistyöroboteille. Luodun lisäohjelman avulla on kyetty hallitsemaan robottiohjelmaa entistä paremmin.

Opinnäytetyössä toteutettua tiedonkeruujärjestelmää ja roboteilta kerättävän datan määrää on helppo laajentaa tulevaisuuden tarpeiden mukaan. Luotu tiedonkeruujärjestelmä mahdollistaa älykkään analytiikan ottamisen käyttöön. Järjestelmästä saatavat hyödyt Abloy Oy:lle paranevat entisestään, kun lähestytään Teollisuus 4.0 mukaista älykästä tehdasta. Valittu RTDE-kommunikointirajapinta mahdollistaa parametrien ja ohjauskäskyjen lähettämisen ylemmältä järjestelmätasolta yhteistyöroboteille, joten niiden ohjaus ja säätö ylemmältä järjestelmätasolta on mahdollista toteuttaa opinnäytetyössä luodun järjestelmän pohjalta.

11.1 Jatkokehitys

Yhteistyörobottien operaattoreiden suorittama manuaalinen korjausliikkeiden tekeminen huoltovälien kohdalla voidaan korvata tekemällä se yhteistyörobottien ohjelmassa automaattisesti. Ohjelmaan voidaan pienin muutoksin lisätä esimerkiksi automaattisesti suoritettava korjaustoimenpide. Korjaustoimenpide voisi olla esimerkiksi, että yhteistyörobotit tarkistavat 100 syklin välein, onko ForcePushEnd-arvon keskiarvo pysynyt sillä tasolla millä sen pitäisi olla. Jos keskiarvo on liian pieni, yhteistyörobotti voi laskennallisesti muuttaa koordinaattipistettä eteenpäin. Vastaavasti jos keskiarvo on liian suuri, se voi tehdä korjausliikkeen taaksepäin.

Palvelimella suoritettavaa tiedonkeruuta tekevän Python-ohjelman kehitystyötä kannattaa myös jatkaa. Python-ohjelman käyttämä Universal Robotsin RTDE-kirjasto pysäyttää ja katkaisee tiedonkeruuhjelman suorittamisen kokonaan, jos robotti katkaisee RTDE-yhteyden. Tämä tilanne voi tapahtua, kun robotti käynnistyy uudestaan odottamattoman virheen tai sähkökatkon vuoksi. Vaikka Python-ohjelma toteutettiin kykenemään muodostamaan samanaikaisesti useaan yhteistyörobottiin RTDE-yhteyden, oli lopulta parempi ottaa jokaiseen robottiin yhteys omalla Python-ohjelmalla. Näin välttyttiin tilanteilta, jossa yksi robotti katkaisi RTDE-yhteyden, minkä seurauksena

RTDE-kirjaston pysäytyskomento katkaisi kaikki yhteydet. Python-ohjelmaa voi kehittää niin, että yksi ohjelman suoritus riittää ja yhteydet jokaiselle yhteistyörobotille luodaan esimerkiksi omiin ohjelmäsäikeisiinsä. Mahdollisen yhteyskatkaisun sattuessa virhe käsiteltäisiin niin, että koko ohjelma ei sulkeutuisi, vaan ohjelmäsäie kokeilisi uutta yhteydenmuodostusta hetken päästä uudelleen.

SQL-tietokannan taulujen rakenteen muokkaamista kannattaa myös harkita. "DataTable"-tauluun tulee uusia rivejä jatkuvasti, mistä syystä sen koko tulee ajan myötä kasvamaan lineaarisesti. "Cobot" ja "ErrorType" tietueet ovat määritetty olemaan tyyppiltään Varchar-tyyppiä, kun ne voitaisiin korvata suorituskyvykkäämmällä INT-tyypillä. Muutos vaatisi kahden uuden taulun tekemistä, esimerkiksi "CobotName" ja "ErrorName", joista selkokiekiset Varchar-muuttujat yhdistetään viiteavaimella.

SQL-tietokannassa olevassa "StatusTable"-taulussa yhteistyörobottien reaaliaikainen tilatieto on tuotu helposti käsiksi päästävään muotoon, mitä pystyy hyödyntämään myös muunlaisissa käyttötarkoituksissa. Esimerkiksi tilatietoja seuraava järjestelmä voisi testin tilan muuttuessa automaattisesti lähettää ilmoituksen tuotetestauksen työntekijöille sähköpostitse. Tilatietoja pystyisi hyödyntämään myös jonkinlaisessa visuaalisessa dashboard-näkymässä, mikä esimerkiksi voisi olla saavutettavissa Abloy Oy:n sisäverkosta verkkoselaimella. Näkymän avulla tuotetestauksen työntekijät pystyisivät eri laitteilla tarkistamaan, kuinka suoritettavat testit etenevät tai tarvitseeko tehdä jotain toimenpiteitä.

11.2 Luotettavuus ja eettisyys

Opinnäytetyössä pyrittiin käyttämään alkuperäislähteitä aina kun sellaisia oli saatavilla. Lähdemerkintöihin ja -viittauksiin kiinnitettiin erityistä huomiota ja lähdeaineistoista tuodun tiedon alkuperä pyrittiin aina mainitsemaan selkeästi. Omat johtopäätökset pyrittiin tuomaan esiin lähdeviittausten ulkopuolella, etteivät ne sekoittuisi lähteistä tuotuun informaatioon. Koko työn aikana noudatettiin JAMK:n eettisiä ohjeita. Työssä ei käsitelty henkilötietoja ja liitteen 2 raportissa on poistettu toimeksiantajayrityksen työntekijöiden ja tuotteiden nimet. Työssä kiinnitettiin erityistä huolta siihen, että toimeksiantajan yrityssalaisuuksia tai muuta salassa pidettävää tietoa ei tuoda esille. Työssä valmistuneen järjestelmän avulla on onnistuttu saamaan arvokasta tietoa niin tuotetes-

tauksen kuin myös tuotekehityksen avuksi, mutta tällaiset yksityiskohdat löydöksistä on opinnäytetyön raportissa häivytetty. Raportissa ei kerrota mitään tietoa toimeksiantajan lukitustuotteista, mitä ei ole julkisesti saatavilla.

Työn tutkimuskysymyksiin onnistuttiin vastaamaan kartoittamalla ensin mahdollisia vaihtoehtoja tiedonsiirto- ja tiedonkeruumenetelmistä ja sitten perustellusti ottamalla käyttöön valitut menetelmät. Työssä kehitetyn tiedonkeruujärjestelmän kaltaisen järjestelmän toteuttaminen olisi onnistunut myös muita käytettävissä olevia tiedonsiirtomenetelmiä ja tietovarastoja käyttäen. Yhteistyörobotin rajapinnan valinta perustui saatavilla olevien dokumenttien ja esimerkkien määrään. Tiedonsiirtoon käytettävä ohjelmointikieli ja tietovaraston tyyppi perustuivat pääosin toimeksiantajan toiveisiin. Vaihtoehtoisia toimivia järjestelmiä ei työssä kehitetty, joten kattavaa käytännön vertailua järjestelmien eroista ei toteutettu. Aikaan saatu järjestelmä toimi suunnitellusti ja täysin tavoitteiden mukaisesti, joten vaihtoehtoisten järjestelmien kokeiluun ei nähty tarvetta. Aikaan saadun järjestelmän heikkoutena on, että tiedonkeruujärjestelmä ei itse kykene muodostamaan uutta yhteyttä, jos yhteys robotteihin katkeaa. Automaattisen uudelleenyhdistämisen kuitenkin voi saavuttaa jatkokehittämällä järjestelmää.

Tietokantaan siirtyneet voima-arvot olivat odotetun suuruisia niin onnistuneilla yhteistyörobottien sykleillä kuin virheiden tapahtuessakin. Tietokantaan siirtyneet arvot on havaittu olevan paikkaansa pitäviä, koska järjestelmän luomien raporttien visuaalisista näkymistä pystyisi selkeästi huomaamaan tiedonsiirrossa tapahtuvat poikkeavuudet. Valmistuneen järjestelmän käyttäminen sellaisenaan muissa kuin mekatronisten sylinterituotteiden kulutustesteissä on mahdollista, mutta käytetyn tietokannan DataTable-taulun sarakkeiden nimet eivät välttämättä sellaisenaan olisi kovin kuvaavia. Järjestelmää on helppo muokata esimerkiksi tallentamaan tietoja testin tyyppin perusteella eri tietokannan tauluihin. Järjestelmää kannattaa kehittää sen monipuolisen käyttökohteiden maksimoimiseksi. Opinnäytetyön tuloksena on toimiva ja helposti eri käyttökohteisiin muokattava ja soveltuva tiedonkeruujärjestelmä, jolla jo järjestelmän alkuvaiheissa on onnistuttu saamaan aikaan työn toimeksiantajan toiminnan laatua kehittäviä tuloksia.

Lähteet

Abloy yleisesitys. 2023. Abloy Oy. Abloy Oy:n julkaisematon esittely yrityksestä vieraille ja uusille työntekijöille.

Abloyn tarina. N.d. ASSA ABLOY. Viitattu 28.3.2024. <https://www.abloy.com/global/fi/about-abloy/history>.

André, J.C. 2019. Industry 4.0: Paradoxes and Conflicts. Hoboken. John Wiley & Sons. Kuvio. Viitattu 25.3.2024. <https://janet.finna.fi>, Ebook Central.

Barooah, R. N.d. Applications That Work Best With NoSQL Database. Kuvio. Viitattu 9.3.2024. <https://www.clariontech.com/blog/applications-that-work-best-with-nosql-database>.

Bauer, W., Bender, M., Braun, M., Rally, P., Scholtz, O. 2016. Fraunhofer Institute for Industrial Engineering. Stuttgart. Viitattu 8.2.2024. [https://www.researchgate.net/publication/327744724 Lightweight robots in manual assembly - best to start simply Examining companies%27 initial experiences with lightweight robots](https://www.researchgate.net/publication/327744724_Lightweight_robots_in_manual_assembly_-_best_to_start_simply_Examining_companies%27_initial_experiences_with_lightweight_robots).

Bernier, C. 2022. Cylindrical Robots: Your Guide to the Compact Automation Solution. HowToRobot. Viitattu 30.1.2024. <https://howtorobot.com/expert-insight/cylindrical-robots>.

Bernier, C. 2023. How Spherical Robots Continue to Support Modern Manufacturers. HowToRobot. Viitattu 24.3.2024. <https://howtorobot.com/expert-insight/spherical-robots>.

Billing, M. 2023. Teollisuusrobotti. Julkaisussa Teollisuuden robotiikka. Toim. K. Välimäki & M. Niemelä. Suomen Robotiikkayhdistys ry. Viitattu 14.11.2023. <https://janet.finna.fi>, Ellibs.

Chand, M. 2023. Most Popular Databases In The World (2023). C# Corner. Kuvio. Viitattu 5.11.2023. <https://www.c-sharpcorner.com/article/what-is-the-most-popular-database-in-the-world/Images/Most%20Popular%20Databases%20In%20the%20World.jpg>.

Collin, J. & Saarelainen, A. 2016. Teollinen internet. Helsinki: Talentum Pro. Viitattu 25.9.2023. <https://janet.finna.fi>, Alma Talent.

Dalik, T. 2012. The Beginner's Guide to Understanding CSV Files. Päivitetty 14.9.2023. Viitattu 27.2024. <https://www.shopping-cart-migration.com/must-know-tips/5985-csv-what-why-and-how>.

Dashboard Server e-Series, port 29999. N.d. Universal Robots A/S. Päivitetty 5.3.2024. Viitattu 27.3.2024. <https://www.universal-robots.com/articles/ur/dashboard-server-e-series-port-29999/>.

Demystifying Collaborative Industrial Robots. 2020. IFR International Federation of Robotics. Frankfurt. <https://www.automate-uk.com/media/4jmhne5p/ifrdemystifyingcollaborativerobotsupdatev03dec2020.pdf>.

Ethernet IP guide. 2024. Universal Robots A/S. Päivitetty 14.3.2024. Viitattu 27.3.2024.
<https://www.universal-robots.com/articles/ur/interface-communication/ethernet-ip-guide/>.

EtherNet/IP: Implicit vs. Explicit Messaging. N.d. Library at AutomationDirect.com. Viitattu 29.9.2023. <https://library.automationdirect.com/ethernetip-implicit-vs-explicit-messaging/>.

Evans, P. & Annunziata, M. 2012. Industrial Internet: Pushing the Boundaries of Minds and Machines. General Electric. Viitattu 20.2.2024.
https://www.researchgate.net/publication/271524319_Industrial_Internet_Pushing_the_boundaries_of_minds_and_machines.

Failures with multiple RTDE connections. Universal Robots A/S. 2018. Viitattu 11.12.2023.
<https://forum.universal-robots.com/t/failures-with-multiple-rtde-connections/2793>.

Fanuc CR-14iA/L. N.d. Fanuc. Kuvio. Viitattu 14.2.2024.
<https://www.fanuc.eu/pt/en/robots/robot-filter-page/collaborative-robots/collaborative-cr-14ial>.

Graph Robot Density by country 2021. 2022. IFR International Federation of Robotics. Frankfurt. Kuvio. Viitattu 27.9.2023.
https://ifr.org/downloads/press2018/RobotDensities_WorldRobotics2022_1600_900.jpg.

Groover, M. 2015. Automation, production systems, and computer-integrated manufacturing. 4. p. Pearson Education. Viitattu 14.11.2023. <https://janet.finna.fi>, VLeBooks.

Hilskä-Keinänen, K. 2018. Robotti-termi syntyi tšekkiläisten maaorjien raadannasta. Yle. Viitattu 9.11.2023. <https://yle.fi/aihe/artikkeli/2018/08/07/robotti-termi-syntyi-tsekkilaisten-maaorjien-raadannasta>.

Hänninen, P. 2021. Robottiikka ja tekoäly. Tampere. Tammertekniikka.

Imhoff, J. 2021. Four Things Every Beginner Should Know About EtherNet/IP (Told by an EtherNet/IP Beginner). Real Time Automation Inc. Viitattu 29.9.2023.
<https://www.rtautomation.com/rtas-blog/four-things-beginners-should-know-ethernet-ip/>.

ISO 10218-2:2011. Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration. International Organization for Standardization ISO. Viitattu 27.9.2023. <https://www.iso.org/standard/41571.html>.

ISO 8373:2021. Robotics — Vocabulary. 3. p. Geneve: International Organization for Standardization.

Latokartano, J. 2023. Paremmalta näyttää. Julkaisussa Automaatiöväylä 2023, 6, 21. Automaatiöväylä Oy.
https://www.automaatiovayla.fi/site/assets/files/6207/automaatiovayla_6_2023.pdf.

Lehtinen, J. 2015. Robottiikka vaatii monitieteellisyttä. Julkaisussa Tieteessä tapahtuu 2015, 33, 5, 42. Viitattu 14.11.2023. <https://journal.fi/tt/article/download/52748/16435/48893>.

- Lehtinen, N. 2023. Robotista on moneksi. Julkaisussa Automaatioväylä 2023, 6, 28–29. Automaatioväylä Oy.
https://www.automaatiovayla.fi/site/assets/files/6207/automaatiovayla_6_2023.pdf.
- Lempiäinen, J. 2023 a. Teollisuuden robotiikka Suomessa. Julkaisussa Teollisuuden robotiikka. Toim. K. Välimäki & M. Niemelä. Suomen Robotiikkayhdistys ry. Viitattu 19.9.2023.
<https://janet.finna.fi>, Ellibs.
- Lempiäinen, J. 2023 b. Korjataan viime vuosikymmenen investointilamaa. Julkaisussa Automaatioväylä 2023, 6, 22–25. Automaatioväylä Oy.
https://www.automaatiovayla.fi/site/assets/files/6207/automaatiovayla_6_2023.pdf.
- Linear gantry YXML n.d. Festo Corporation. Kuvio. Viitattu 24.3.2024.
https://www.festo.com/tw/en/p/linear-gantry-id_YXML/.
- Linturi, R. & Kuusi, O. 2018. Suomen sata uutta mahdollisuutta 2018–2037. Eduskunnan tulevaisuusvaliokunnan julkaisu 1/2018. Helsinki.
https://www.eduskunta.fi/FI/naineduskuntatoimii/julkaisut/Documents/tuvj_1+2018.pdf.
- Liuha, A. & Aro, J. 2023. Aistijärjestelmät ja kommunikointi. Julkaisussa Teollisuuden robotiikka. Toim. K. Välimäki & M. Niemelä. Suomen Robotiikkayhdistys ry. Viitattu 19.9.2023.
<https://janet.finna.fi>, Ellibs.
- Modbus server. 2015. Universal Robots A/S. Viitattu 18.2.2024. <https://www.universal-robots.com/articles/ur/interface-communication/modbus-server/>.
- Morris, S. 2023. Data Storage: JSON vs. CSV. Viitattu 27.2024. <https://coresignal.com/blog/json-vs-csv/>.
- Omron RS4-2066502. N.d. Omron. Kuvio. Viitattu 11.2.2024.
<https://8z1xg04k.tinifycdn.com/images/RS4-2066502.jpg>.
- OPC UA Client/Server. N.d. Universal Robots. Päivitetty 14.3.2024. Viitattu 27.3.2024.
<https://www.universal-robots.com/plus/products/rocketfarm/opc-ua-clientserver/>.
- Overview of client interfaces. N.d. Universal Robots A/S. Sivua päivitetty 14.3.2024. Viitattu 27.3.2024. <https://www.universal-robots.com/articles/ur/interface-communication/overview-of-client-interfaces/>.
- PreciseFlex 100. 2022. Brooks Automation. Kuvio. Viitattu 14.2.2024.
<https://www.brooks.com/getmedia/b14008fa-32d0-4560-8b5f-620408847210/PreciseFlex-100.pdf>.
- PreciseFlex 400. N.d. Brooks Automation. Kuvio. Viitattu 14.2.2024.
<https://ukrobotics.com/devices/pf400-robot/>.

Product specification IRB 390. 2022. ABB. Kuvio. Viitattu 11.2.2024.

<https://library.e.abb.com/public/742fe3190890485586e156308e13fde9/3HAC066568%20PS%20IRB%20390-en.pdf?x-sign=+zUkDIGjJPYW3lxtNaF5/VJJlehP6Qchbo6cUhhKVqYsjhqrLDzLb3Ou0RxvsZmE>.

PROFINet How-To Guide E-series. 2024. Universal Robots A/S. Päivitetty 14.3.2024. Viitattu 27.3.2024. <https://www.universal-robots.com/articles/ur/interface-communication/profinet-how-to-guide-e-series/>.

Realtime Client Interface. N.d. Universal Robots A/S. Viitattu 9.12.2023. https://s3-eu-west-1.amazonaws.com/ur-support-site/16496/ClientInterfaces_Realttime.pdf.

Real-Time Data Exchange (RTDE) Guide. 2019. Universal Robots A/S. Sivua päivitetty 27.9.2023. Viitattu 9.12.2023. <https://www.universal-robots.com/articles/ur/interface-communication/real-time-data-exchange-rtde-guide/>.

Remote Control Via TCP/IP. 2015. Universal Robots A/S. Sivua päivitetty 14.3.2024. Viitattu 9.12.2023. <https://www.universal-robots.com/articles/ur/interface-communication/remote-control-via-tcpip/>.

Robot, First Unimate Robot Ever Installed on an Assembly Line, 1961. N.d. The Henry Ford. Kuvio. Viitattu 24.3.2024. <https://www.thehenryford.org/collections-and-research/digital-collections/artifact/183434/#slide=gs-450962>.

Robotiq Force Copilot. N.d. Universal Robots A/S. Viitattu 21.3.2024. <https://www.universal-robots.com/fi/plus/products/robotiq/robotiq-force-copilot/>.

RTDE_Python_Client_Library. 2022. Viitattu 21.3.2024. https://github.com/UniversalRobots/RTDE_Python_Client_Library.

SciClops Microplate Handler. N.d. Hudson Robotics. Kuvio. Viitattu 14.2.2024. <https://hudsonrobotics.com/microplate-handling-2/platecrane-sciclops-3/>.

SFS-EN 15684:2020. Building hardware. Mechatronic door furniture. Requirements and test methods. SFS Suomen Standardit ry.

TCP/IP socket communication via URScript. N.d. Universal Robots A/S. Päivitetty 14.3.2024. Viitattu 27.3.2024. <https://www.universal-robots.com/articles/ur/interface-communication/tcpip-socket-communication-via-urscript/>.

Technology Overview Series: Ethernet/Ip. 2021. ODVA Inc. Viitattu 26.9.2023. https://www.odva.org/wp-content/uploads/2021/05/PUB00138R7_Tech-Series-EtherNetIP.pdf.

Universal Robots e-Series User Manual. 2022. Dokumentin versio 9.6.179. Universal Robots A/S.

UR5e Technical Specification. 2023. Universal Robots A/S. Viitattu 21.3.2024. <https://www.universal-robots.com/media/1807465/ur5e-rgb-fact-sheet-landscape-a4.pdf>.

What Kinds of Industrial Robots Are There? A Guide on the Features of the Major 6 Types. 2018. Kawasaki. Viitattu 11.2.2024. <https://robotics.kawasaki.com/ja1/xyz/en/1803-01/>.

Wyzo sidebot. N.d. Wyzo. Kuvio. Viitattu 14.2.2024 - <https://thewyzo.com/technical-specifications/>.

XML-RPC communication. N.d. Universal Robots A/S. Päivitetty 14.3.2024. Viitattu 27.3.2024. <https://www.universal-robots.com/articles/ur/interface-communication/xml-rpc-communication/>.

Liitteet

Liite 1. Python-koodi UR5e edestakaiseen RTDE-kommunikointiin

```

import argparse
import sys
import time
import os
import datetime

sys.path.append(".")
import rtde.rtde as rtde
import rtde.rtde_config as rtde_config

#dictionary/sanakirja robottien nimista ja osoitteista
cobotlista = [
    {"name": "Cobot1", "ip": "192.168.1.1"},
    {"name": "Cobot2", "ip": "192.168.1.2"},
    # Lisaa robotteja tarvittaessa
]

#luodaan niin monta argument parseria kuin robotteja on sanakirjassa
parserlista = []
for i in range(len(cobotlista)):
    parserlista.append(argparse.ArgumentParser())

#Yhteyden muokattavat parametrit
def parametrit(parser, osoite):
    parser.add_argument(
        "--host", default=str(osoite), help="name of host to connect to (localhost)"
    )
    parser.add_argument("--port", type=int, default=30004, help="port number (30004)")
    parser.add_argument(
        "--samples", type=int, default=0, help="number of samples to record"
    )
    parser.add_argument(
        "--frequency", type=int, default=125, help="the sampling frequency in Herz"
    )
    parser.add_argument(
        "--config",
        default="record_configuration.xml",
        help="data configuration file to use (record_configuration.xml)",
    )
    parser.add_argument("--verbose", help="increase output verbosity", action="store_true")
    parser.add_argument(
        "--buffered",
        help="Use buffered receive which doesn't skip data",
        action="store_true",
    )
    parser.add_argument(
        "--binary", help="save the data in binary format", action="store_true"
    )

def yhteysmuodostus(parametrit):
    conf = rtde_config.ConfigFile(parametrit.config)
    output_names, output_types = conf.get_recipe("out")
    input_names, input_types = conf.get_recipe("in")

    con = rtde.RTDE(parametrit.host, parametrit.port)
    con.connect()
    print("Yhdistetty "+ str(parametrit))

    # setup recipes. outputs.
    if not con.send_output_setup(output_names, output_types, frequency=parametrit.frequency):
        logging.error("Unable to configure output")
        sys.exit()
    # setup recipes. inputs.
    global inputs
    inputs = con.send_input_setup(input_names, input_types)

    # start data synchronization
    if not con.send_start():
        logging.error("Unable to start synchronization")
        sys.exit()
    return con

def tilatarkastus(yhteys, argumentit):
    try:
        if argumentit.buffered:
            state = yhteys.receive_buffered(argumentit.binary)
        else:
            state = yhteys.receive(argumentit.binary)
        if state is not None:
            #dictionary/sanakirja yhteyden tiloista
            tila={}

```

```

#Robotin tila
tila['cobotstate'] = getattr(state, 'runtime_state') or 1
#Tilanusnumero
tila['cobotordernumber'] = getattr(state, 'output_int_register_0') or 0
#Sykklit
tila['cobotcycles'] = getattr(state, 'output_int_register_1') or 0
#Virheet
tila['coboterrors'] = getattr(state, 'output_int_register_2') or 0
#Vapaamuotoinen robotin tila, esimerkiksi "robotti huoltovalissa"
tila['cobotstatus'] = getattr(state, 'output_int_register_3') or 0
#Robotin output64, kattelytieto robotilta
tila['cobot64status'] = getattr(state, 'output_bit_register_64') or 0

#Selkokielinen lista robotin tilasta
if(tila['cobotstate']==1):
    tila['cobotstate']="Ei ajossa"
elif(tila['cobotstatus']==2):
    tila['cobotstate']="Robotti huoltovalissa"
elif(tila['cobotstate']==2):
    tila['cobotstate']="Ajossa"
elif(tila['cobotstate']==3):
    tila['cobotstate']="Ei ajossa (3)"
elif(tila['cobotstate']==4):
    tila['cobotstate']="Ei ajossa (pause)"

except KeyboardInterrupt:
    keep_running = False
except rtde.RTDEException:
    yhteys.disconnect()
    sys.exit()
return tila

print("\n\nAloitetaan cobottien RTDE-yhteyksien muodostus\n")
yhteyslista = []
argumenttilista = []

#Luodaan niin monta yhteyttä kuin robotin nimia on listassa
for i in range(len(cobotlista)):
    parserlista.append(argparse.ArgumentParser())
    try:
        parametrit(parserlista[i], cobotlista[i]["ip"])
        argumenttilista.append(parserlista[i].parse_args())
        yhteyslista.append(yhteysmuodostus(argumenttilista[i]))
    except:
        print("\n" + cobotlista[i]["name"] + ": Yhteysvirhe\n")
        time.sleep(2)

#Ohjelman paasilmukka alkaa tasta
while True:
    os.system('CLS')
    TimeNow = datetime.datetime.now()
    print(TimeNow)

    #Tarkastetaan tila niin monesta yhteydesta kuin robotin nimia on listassa
    for i in range(len(cobotlista)):
        try:
            Cobot = cobotlista[i]["name"]
            Status = str(tilatarkastus(yhteyslista[i],argumenttilista[i])['cobotstate'])
            Tilanusnumero = int(tilatarkastus(yhteyslista[i],argumenttilista[i])['cobotordernumber'])
            Cycles = int(tilatarkastus(yhteyslista[i],argumenttilista[i])['cobotcycles'])
            Errors = int(tilatarkastus(yhteyslista[i],argumenttilista[i])['coboterrors'])
            Status64FromCobot = bool(tilatarkastus(yhteyslista[i],argumenttilista[i])['cobot64status'])

            print("\n" + Cobot + ": " + str(Status) + "\nTilanusnumero: " + str(Tilanusnumero) + "\tSykklit: " +
str(Cycles) + "\tVirheet: " + str(Errors) + "\tLahto robotille: " + str(inputs.input_bit_register_64))

            try:
                #Asettaa robotin input64 samaan tilaan kuin robotin output64
                if(Status64FromCobot == True):
                    inputs.input_bit_register_64 = True
                else:
                    inputs.input_bit_register_64 = False
                #Paivita lahdot
                yhteyslista[i].send(inputs)
            except Exception as error:
                print("Virhe lahdon asettamisessa:", error)

        except Exception as error:
            print("\n" + cobotlista[i]["name"] + ": Yhteysvirhe\n"+str(error))

        time.sleep(1)

sys.stdout.write("\nComplete!\n")

con.send_pause()
con.disconnect()

```

Liite 2. Tietokantaan kerätyn datan perusteella muodostettu testausraportti

ABLOY

ABLOY Test Report

KÄYTTÖ RAJOITETTU / RESTRICTED

Page 1 (6)

Test Case: ATO-10472 -- Quality - Durability test 200k

Test requested by:	--
Request date:	23.02.2024
Description:	Normaali kulutustesti
Test setup:	--
Standard:	EN15684:2020
Standard value:	200k
Abloy internal value:	
Test site:	
Test type:	Durability
Component:	Mechatronic cylinder
Project:	--Quality
Product:	--
Series:	Proto
Test Equipment:	
Tested by:	--
Tested date:	28.03.2024
Test result:	Test finished on CobotKyllicki. Cycles 100000 errors 13 (0.01 %).
Measured Value:	
Resolution:	Passed

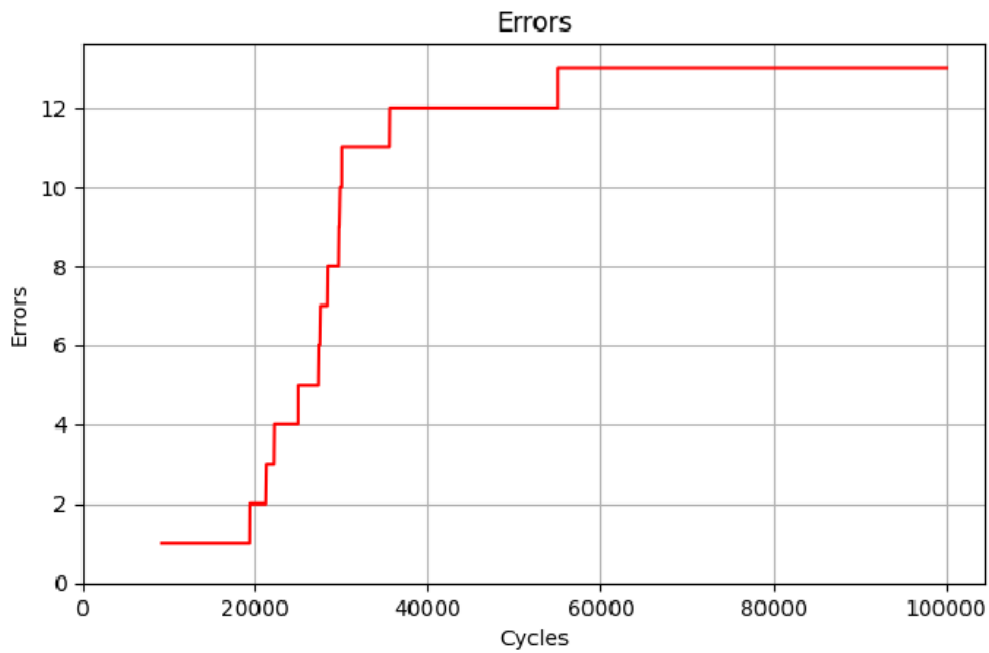
The test results relate only to the sample tested.

Cobot event log:

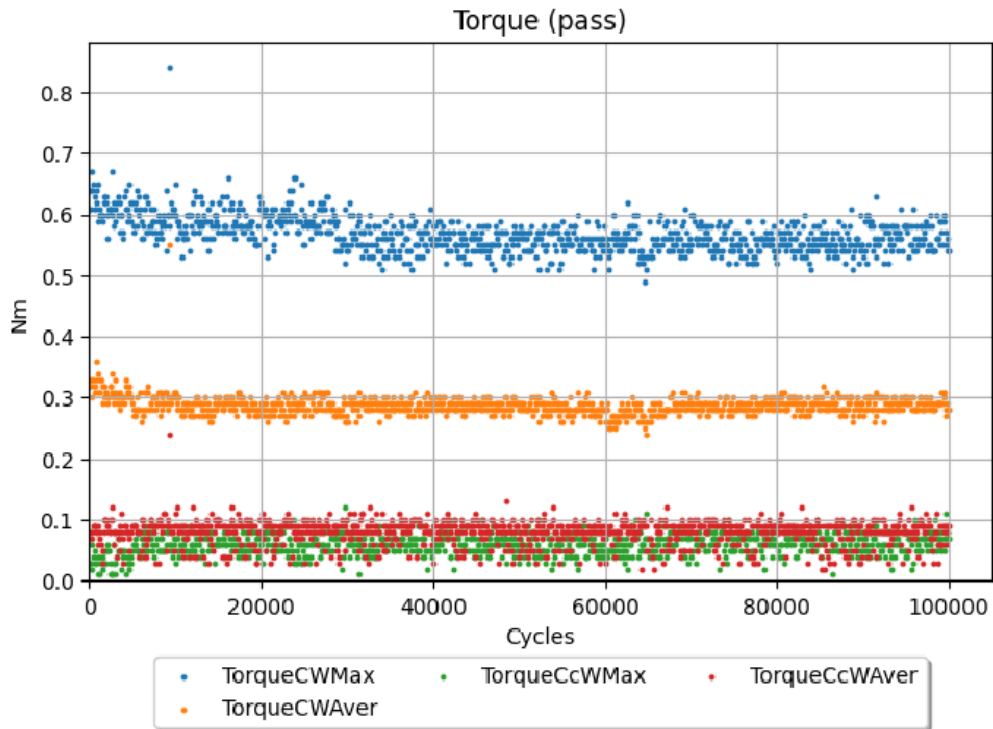
Timestamp	Cycles	Errors	Comment
2024-02-23 13:30:57	0	0	Run started
2024-02-24 15:56:58	5001	0	Maintenance started
2024-02-27 06:50:38	9390	1	Run resumed
2024-02-27 10:03:15	10001	1	Maintenance started
2024-02-27 10:03:33	10001	1	Maintenance finished
2024-02-27 10:04:34	10001	1	Run resumed
2024-02-27 10:05:42	10002	1	Run resumed
2024-02-27 10:07:10	10004	1	Run resumed
2024-02-27 10:08:14	10005	1	Run resumed
2024-02-28 07:13:34	12969	1	Run resumed
2024-02-28 17:53:02	15001	1	Maintenance started
2024-02-29 07:23:10	15001	1	Maintenance finished
2024-02-29 07:25:11	15002	1	Run resumed
2024-02-29 07:26:07	15003	1	Run resumed
2024-03-01 09:39:03	20001	2	Maintenance started
2024-03-01 10:41:48	20001	2	Maintenance finished
2024-03-04 07:44:11	23938	4	Run resumed
2024-03-04 13:18:53	25001	4	Maintenance started
2024-03-04 13:46:52	25001	4	Maintenance finished
2024-03-05 16:01:02	30001	10	Maintenance started
2024-03-06 06:54:12	30001	10	Maintenance finished
2024-03-06 06:55:11	30002	10	Run resumed
2024-03-07 08:00:41	34780	11	Run resumed
2024-03-07 09:10:28	35001	11	Maintenance started
2024-03-07 09:51:51	35001	11	Maintenance finished
2024-03-08 12:05:16	40001	12	Maintenance started
2024-03-08 12:36:49	40001	12	Maintenance finished
2024-03-09 14:50:05	45001	12	Maintenance started
2024-03-11 07:44:08	45001	12	Maintenance finished
2024-03-12 09:57:31	50001	12	Maintenance started
2024-03-12 10:35:50	50001	12	Maintenance finished
2024-03-13 12:49:03	55001	12	Maintenance started
2024-03-13 14:22:20	55001	12	Maintenance finished
2024-03-14 16:35:43	60001	13	Maintenance started
2024-03-15 07:27:34	60001	13	Maintenance finished
2024-03-18 08:16:58	64969	13	Run resumed
2024-03-18 08:27:16	65001	13	Maintenance started
2024-03-18 09:10:41	65001	13	Maintenance finished
2024-03-19 11:24:09	70001	13	Maintenance started
2024-03-19 12:25:52	70001	13	Maintenance finished
2024-03-20 15:14:48	75001	13	Maintenance finished
2024-03-20 15:23:35	75027	13	Run resumed
2024-03-20 15:24:53	75029	13	Run resumed
2024-03-20 15:26:00	75031	13	Run resumed
2024-03-21 17:29:51	80001	13	Maintenance started
2024-03-22 07:40:47	80001	13	Maintenance finished
2024-03-23 09:53:40	85001	13	Maintenance started

The test results relate only to the sample tested.

2024-03-26 09:42:34	90001	13	Maintenance started
2024-03-26 10:16:25	90001	13	Maintenance finished
2024-03-27 12:29:25	95001	13	Maintenance started
2024-03-27 12:59:13	95001	13	Maintenance finished
2024-03-28 15:11:29	100000	13	Run finished



The "Torque" graph below shows the average torque applied during rotation in newton meters. TorqueCWMax(clockwise) and TorqueCcWMax(counterclockwise) shows the maximum value of the torque applied to the gripper during rotation in newton meters. TorqueCWAver (clockwise) and TorqueCcWAver (counterclockwise) shows the average torque applied to the gripper during rotation in newton meters.



The test results relate only to the sample tested.

In the "Force" diagrams below, the perpendicular force applied to the robot's gripper is shown in newtons.

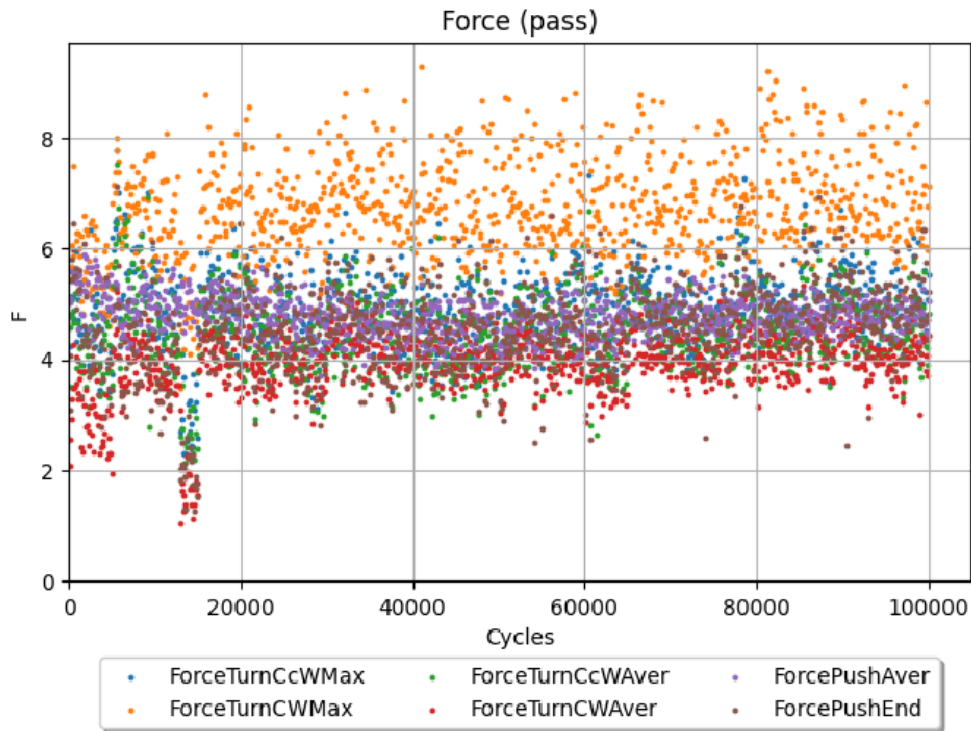
ForceTurnCwMax (clockwise) and ForceTurnCcWMax (counterclockwise) shows the maximum value of the force applied during the rotation in newtons.

ForceTurnCwAver (clockwise) and ForceTurnCcWaver (counterclockwise) shows the average value of the force applied during the rotation in newtons.

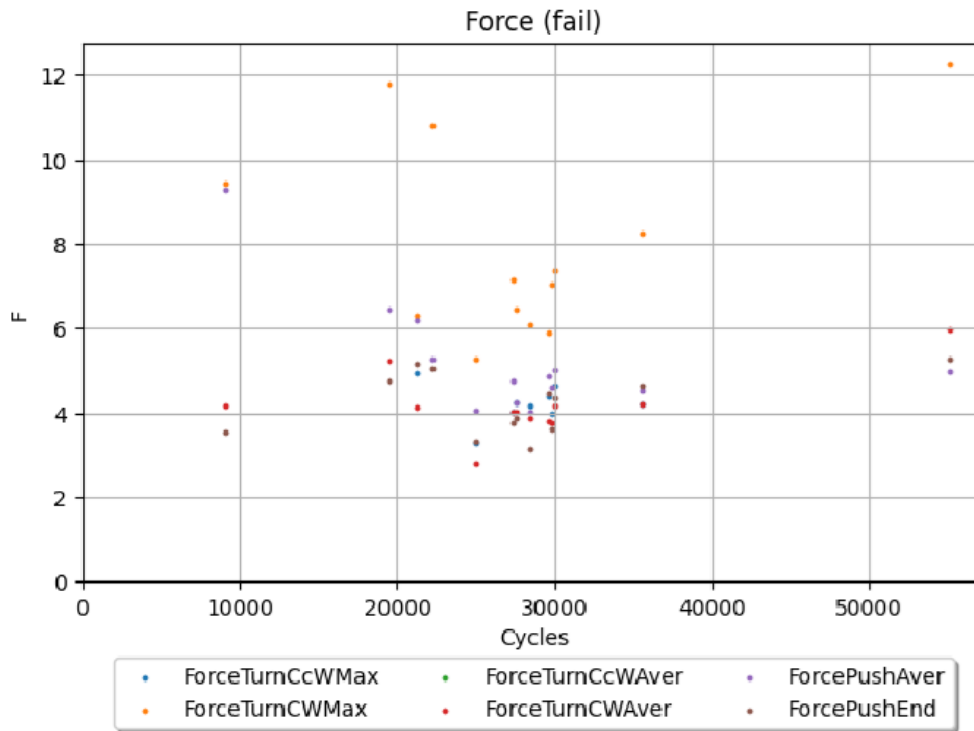
ForcePushAver shows the average force during key push movement of the key in newtons.

ForcePushEnd shows the force after the push movement of the key in newtons before the key is started to turn.

The "Force (pass)" diagram shows the force values of successful cycles and the "Force (fail)" diagram shows the force values of cycles where an error occurred.



The test results relate only to the sample tested.



The test results relate only to the sample tested.