

Jesse Salmi

PUTKENTAIVUTUSSOLUN ROBOTIN ASETUSAIKOJEN LYHENTÄMINEN OH- JELMALLISESTI

Kandidaatintyö
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Jyrki Latokartano
9.2020

TIIVISTELMÄ

Jesse Salmi: Putkentaivutussolun robotin asetusajojen lyhentäminen ohjelmallisesti,
Reducing setup time of tube bending robot programmatically

Kandidaatintyö

Tampereen yliopisto

Automaatiotekniikan tutkinto-ohjelma

Syyskuu 2020

Tämän kandidaatintyön tarkoituksena oli selvittää, kuinka kohdeyrityksen tuotantosolun robotin ohjelmia voitaisiin generoida automaattisesti valmistuksen nopeuttamiseksi. Yrityksellä on ollut ongelmana sopeuttaa tuotantoaan uusien kappaleiden valmistukseen niiden erilaisten geometrinen muotojen takia. Uusien kappaleiden tuotantoon siirtyminen on ollut hidasta, sillä tuotantosolun robottia on jouduttu ohjelmoimaan manuaalisesti. Tämä on vaikeuttanut tuotantoa pitkien asetusajojen takia.

Työ sisältää tutkimuskatsauksen, jossa perehdytään asetusajoihin tuottavassa teollisuudessa sekä kohdeyrityksen robottisolujen erilaisiin osiin ja niiden keskinäisiin vuorovaikutuksiin. Tutkimuksessa perehdyttiin myös robotin paikoitukseen liittyvien tietojen käsittelyyn ja robotin eri ohjelmointimenetelmiin. Työn lähteinä toimivat alojen peruskirjallisuuden lisäksi tieteelliset kirjoitukset sekä artikkelit. Näiden tietojen perusteella työn lopussa esitetään ja vertaillaan erilaisia metodeja, joiden avulla asetusajoja saataisiin lyhennettyä. Työssä löydettiin kaksi eri menetelmää, joiden avulla kohdeyritys voisi automatisoida robotin ohjelmoimisen ja näin ollen lyhentää tuotannossa esiintyviä asetusajoja. Menetelmien käyttökelpoisuutta vertailtiin teknisen haastavuuden ja kustannusten mukaan. Työssä tehtyjen alustavien tutkimusten perusteella toinen näistä kuvatuista menetelmistä osoittautui paremmaksi vaihtoehdoksi, jolla yritys pääsisi tavoitteisiinsa.

Tässä tutkimuksessa esitetyt johtopäätökset eivät ole yleistettävissä kaikkiin tuotantosoluihin, sillä tuloksia analysoitiin kohdeyrityksen valmistusprosessin sekä tuotantosolun perusteella, jotka vaihtelevat tapauskohtaisesti. Kuitenkin kirjallisuuskatsauksessa esitetyt asiat ovat sovellettavissa yleisesti, sillä tiedon keräämiseen käytettiin alan peruskirjallisuutta sekä tieteellisiä artikkeleja, jotka eivät olleet sidottuja yksittäiseen tapaukseen.

Avainsanat: asetusajaja, tuotantosolu, ohjelmointi.

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ALKUSANAT

Robottiikka on ollut pitkään oma kiinnostuksen kohteeni opiskeluaikanani. Tahdon kiittää Fennosteel Oy:ta mielenkiintoisesta aiheesta, joka mahdollisti tietämykseni avartamisen robotiikasta. Ratkaisun etsiminen oikeaan ongelmaan toi oman mausteensa työn etenemiseen. Haluaisin kiittää työnohjaajiani Jyrki Latokartanoa ja Liisa Ahaa arvokkaista neuvoista ja palautteesta. Asiantuntevat kommenttinne auttoivat minua paljon työni edistymisessä. Erityisen iso kiitos kuuluu myös kaikille opiskelutovereilleni, joiden vankkumaton tuki ovat kantaneet minua opiskelujeni parissa.

Tampereella, 21.9.2020

Jesse Salmi

SISÄLLYSLUETTELO

1. Johdanto	1
2. Asetusaikojen merkitys tuotannossa	3
3. Robotin kinematiikka	5
3.1 Teollisuusrobotin koordinaatistot	5
3.2 Koordinaattimuunnokset	8
4. Numeerinen ohjaus	10
4.1 Järjestelmän osat	10
4.2 NC-ohjelma	11
4.3 Käytön edut	12
5. Robotin ohjelmointimenetelmät	15
5.1 Johdattamalla ohjelmointi	15
5.2 Opettamalla ohjelmointi	16
5.3 Mallipohjainen ohjelmointi	17
5.4 Tekstipohjainen ohjelmointi	18
6. Asetusaikojen lyhentäminen	19
6.1 Järjestelmän kuvaus	19
6.2 Toteutusehdotuksia	21
7. Yhteenveto	23
Lähteet	26

KUVALUETTELO

3.1 Oikean käden sääntö. [10]	6
3.2 Robotin koordinaatistot. [10]	7
3.3 Pistein esitys kahden eri koordinaatiston avulla. [4]	8
4.1 Osa NC-koodin esimerkkiohjelmasta. [11]	12
6.1 Tuotantosolun putken valmistusprosessi.	20

LYHENTEET JA MERKINNÄT

CAD	engl. Computer Aided Design, tietokoneavusteinen suunnittelu
CAM	engl. Computer Aided Manufacturing, tietokoneavusteinen valmistus
CNC	engl. Computerized numerical control, tietokoneistettu numeerinen ohjaus
NC	engl. Numerical control, numeerinen ohjaus
<i>P</i>	Paikkavektori
<i>p</i>	Kiertomatriisi
<i>T</i>	Homogeeninen transformaatiomatriisi

1. JOHDANTO

Asetusajoilla on suuri merkitys yrityksen kilpailukyvyn parantamisessa. Asetusaikojen pienentämällä työstökoneiden käyttö tehostuu, tuotantoon sidottu vaihto-omaisuus pienenee ja toimitusajat ovat lyhyempiä. [12, s. 38–39] Näin ollen on tärkeää, että yritys pyrkii lyhentämään asetusaikojen. Vaikka robotiikkaa on jo pitkään sovellettu valmistavassa tuotannossa, yritysten välinen kiristynyt kilpailu on luonut lisävaatimuksia tuotannon automatisointiin. Valmistettavien kappaleiden on oltava yhä halvempia, parempilaatuisia ja sopivampia asiakkaille. Tämä on luonut tarpeen kehittää monitoimisia robotteja, jotka soveltuvat käytettäväksi dynaamisissa ympäristöissä. [15]

Tämän kandidaatintyön tavoitteena on selvittää, miten kohdeyrityksen robottisoluun asetusaikojen voitaisiin lyhentää. Asiaa on lähdetty selvittämään Fennosteel Oy:n näkökulmasta, joka toimii tämän työn toimeksiantajana. Fennosteel on pakoputkien, äänenvaimentimien ja teräsputkien tuotantoon erikoistunut yritys, jonka toimipiste sijaitsee Suomen Parkanossa. Yritys valmistaa ja toimittaa vuosittain yli 680 000 pakoputkistoa erilaisiin ajoneuvoihin. [8] Yrityksessä on tavoiteltu tuotannon tehostamista vähentämällä uuden putkityypin valmistukseen kuluva aikaa. Alustavien tutkimusten mukaan valmistussolun robotin ohjelmointi on ollut rajoittava tekijä. Haasteina on ollut putkien muodoista johtuvien geometrinen eroavaisuuksien huomioiminen ohjelmissa.

Työn alussa esitellään asetusaika ja sen merkitys tuottavassa teollisuudessa. Samalla perustellaan, miksi yrityksen on kannattavaa pyrkiä pienentämään asetusaikojen. Tämän jälkeen työssä selvitetään, kuinka robotti käsittelee ja hallitsee erilaisia paikoitukseen liittyviä tietoja. Neljännessä luvussa tarkastellaan työstökoneiden automatisoinnissa käytettyä menetelmää ja sen ohjelmien rakennetta. Viidennessä luvussa tarkastellaan robotin erilaisia ohjelmointityökaluja. Keskeisimpiä tutkimuskysymyksiä työssä ovat näin ollen:

- Mikä merkitys asetusaajoilla on yrityksen tuotannolle?
- Miten työstösolu käsittelee putken geometrisia muotoja?

- Kuinka robotin toimintoja voidaan muokata ohjelmallisesti?

Lopuksi työssä kuvaillaan erilaisia metodeja liittyen paikoituksen automaattiseen generointiin, jonka avulla robottisolun asetusaikoja saataisiin lyhennettyä. Tuloksien soveltuvuutta arvioidaan teknisen haastavuuden ja investointikustannusten mukaan. Työ on rajattu käsittelemään asetusaikojen pienentämistä ohjelmallisesti muuttaen yhden robotin toimintoja.

Työn tietolähteinä toimivat Tampereen teknillisen yliopiston ja Tampereen yliopiston tarjoamat tiedonhakupalvelut, joista tärkeimpinä ovat toimineet kirjasto, Andor ja IEEE Xplore. Työn lähteinä on käytetty niin painettuja kuin sähköisiä kirjoja. Näiden lähteiden päätarkoituksena on tarjota aihealueeseen liittyvä perustietämys. Työssä on myös käytetty aiheeseen liittyviä artikkeleita, joita on haettu alan liittyvistä julkaisuista, kuten *Industrial Robot: An International Journal* ja *Robotics and Computer-Integrated Manufacturing* aikakauslehdistä. Artikkelien avulla työhön on etsitty ajakohtaisimpia tutkimuksia aihealueista. Tämän lisäksi työssä on käytetty hyväksi Suomen Standardisoimisliiton tarjoamaa tietokantaa, jonka avulla työhön on haettu standardeja. Standardien avulla työhön on etsitty tarkasti määriteltyä tietoa.

Lähteiden oikeellisuutta ja vaikuttavuutta on arvioitu lähteen julkaisutietojen ja sitaattien määrällä. Artikkeleja käsitteleviä hakuja on rajattu Andorissa käyttämällä tarkennettujen hakujen rajaamiseen liittyviä optioita ”Koko teksti verkossa” ja ”Tieteellinen & vertaisarvioitu”. Kirjojen hakutuloksia on rajattu pääsääntöisesti rajaamalla julkaisuaikankohtaa sisällyttämään vain tämän vuosituhannen puolella julkaistut teokset.

Lähteiden hakusanoina on käytetty sekä suomen- että englanninkielisiä sanojen lisäksi lyhenteitä. Numeerisesta ohjauksesta on etsitty tietoja muun muassa hakusanoilla ”numeerinen ohjaus”, ”numerical control” ja ”CNC”. Hakukoneiden Boolean operaattoreita on käytetty rajaamaan entuudestaan hakutuloksia. Useimpiin hakuihin on käytetty katkaisumerkkiä sekä JA-operaattoria. Lähteitä on haettu sekä suomen- että englanninkielisistä teoksista. Alan ulkomaisten julkaisuiden suuresta määrästä johtuen lähteet ovat pääasiallisesti englanniksi.

2. ASETUSAIKOJEN MERKITYS TUOTANNOSSA

Valmistavien yritysten tuotantoon asetetut vaatimukset edellyttävät kasvavaa tarvetta tasalaatuisimmille tuotteille, joita voidaan tuottaa lyhyemmässä ajassa. Tuotannon suunnittelu on tärkeää, jotta tuotanto voisi tehokkaasti ja laadukkaasti tyydyttää asiakkaan tarpeet ja saavuttaa muut tavoitteensa osana yrityksen toimintaa. Yhtenä tärkeänä käsitteenä tuotannon suunnittelussa esiintyy termi asetusaika. Tässä luvussa käsitellään, mikä on asetusaika, mistä se koostuu ja miksi yrityksen kannattaa pyrkiä minimoimaan sitä.

Kun tuotantosolu vaihtaa valmistettavaa tuotetta toiseen, täytyy solun tila muokata sopivaksi, jotta uuden tuotteen valmistus olisi mahdollista. Näitä toimenpiteitä kutsutaan asetuksiksi. Asetustöitä voivat olla muun muassa kiinnittimien sekä työkalujen vaihtaminen, työpisteen puhdistaminen, työstöohjelman parametrien muuttaminen ja uuden ohjelman käyttöönoton testaaminen työstökoneella. [13, s. 60] Aikaa, joka kuluu näiden asetusten tekemiseen kutsutaan asetusajaksi. Se kuvaa siis työvaiheen valmisteluun kuluvaan aikaan ja sen kokonaiskesto lasketaan viimeisen edellisen hyväksyttävän tuotteen ulostulosta ensimmäisen uuden hyväksyttävän tuotteen ulostuloon. Asetusaika esiintyy aina vain kertaalleen tuotettavan tuotteen erää kohti. [13, s. 49][19]

Asetusaika voidaan jakaa sekä ulkoiseen että sisäiseen asetusaikaan. Ulkoinen asetusaika koostuu asetustöiden kestosta, joita voidaan tehdä ennakkoon ennen valmistuksen pysäyttämistä. Nämä työt voivat olla esimerkiksi etukäteen tehtyjä tarkastuksia työkalujen tai kiinnittimien toimintaan liittyen. Sisäinen asetusaika puolestaan koostuu asetuksista, jotka on tehtävä koneen ollessa pysähdyksissä. Esimerkki tällaisesta työstä on uuden työstöohjelman lataaminen koneelle. Yrityksen kannattaa pyrkiä vähentämään mahdollisimman paljon sisäisiä asetustöitä ja muuttamaan niitä ulkoisiksi, sillä tänä aikana tuotantoa ei tarvitse lopettaa koneiden seisottamisen takia. [17]

Asetusaikojen lyhentämisellä on havaittu olevan useita huomattavia hyötyjä yritykselle. Lyhyemmät asetusajat pienentävät tuotantoon kuluvaan aikaan, joka puolestaan

tehostaa resurssien käyttämistä. Tällöin muun muassa koneiden käyttöaste kasvaa, tuotteen läpäisy aika lyhenee ja tuotannon ohjattavuus paranee. Valmistusaikojen lyhentymisen mahdollistaa joustavammat eräkoot, suuremman tuotantokapasiteetin ja nopeammat toimitusajat. [2][20, s. 39][21]

Asetusaikojen lyhentämisellä voidaan myös vaikuttaa yrityksen kannattavuuteen ja näin ollen tulokseen. Nopeammat läpäisyajat koneiden kasvaneen käyttöasteen kanssa pienentävät tuotantoon sidottua vaihto-omaisuutta. Tällöin yrityksellä on käytössään enemmän resursseja muihin operationallisiin toimintoihin, joka puolestaan parantaa kannattavuutta. Joustavammat eräkoot ja nopeammat toimitusajat mahdollistavat pienemmät tilauskoot. Tämä johtaa puutekustannusten pienentymiseen ja kilpailukyvyyn kasvuun, jotka edesauttavat edelleen kannattavuuden kehitystä. [2][12][20, s. 39][21]

Asetusaikojen lyhentämisen on myös havaittu pienentävän yrityksen kuluja. Mukautuvat eräkoot ja toimitusajat vähentävät varastoinnin tarvetta, jolloin säästetään varastointikustannuksista [2][12][20, s. 39][21]. Pelkästään asetusaikojen lyhentämisellä voidaan työtehoa lisätä jopa 30-50 %:ia, jolloin säästetään ajallisesti merkittäviä aikoja [17]. Voidaankin siis todeta, että yrityksen on todella kannattavaa pyrkiä minimoimaan asetustöihin kuuluva aika.

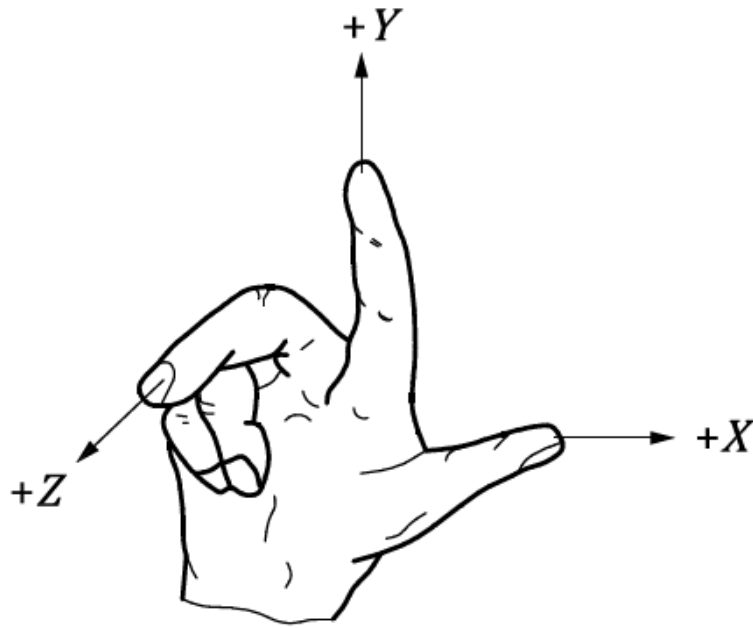
Asetusaikoja voidaan lyhentää eri menetelmin. Yhtenä menetelmänä on asetustöiden ja niihin liittyvien toimintatapojen vakioiminen, jonka on todettu lyhentävän asetusaikoja tehokkaasti. Tämä tarkoittaa muun muassa sitä, että työpisteet ovat järjestetty tarpeeseen sopiviksi ja koneiden paikoitukset sekä työkalut ovat standardoitu mahdollisimman pitkälle. Työstökoneiden asetusaikoja voidaan lyhentää automatisoimalla niitä. Automatisointi mahdollistaa nopeammat asetustyöt manuaalisiin verrattuna ja valmistuksen joustavuuden, sillä valmistuksen jatkuminen ei ole tällöin enää riippuvainen koneen käyttäjistä. [13, s. 60–61]

3. ROBOTIN KINEMATIikka

Jotta robotin liikkeet olisivat hallittuja, on sen muutettava haluttu toiminto niveliin paikkaohjearvoiksi. Tätä varten robotin on pystyttävä määrittämään sijainti ja orientaatio laskennallisesti eri lähtötiedoista. [14, s. 20] Robotin geometrisia ongelmia käsitellään kinematiikan avulla. Se voidaan jakaa sekä suoraan että epäsuoraan kinematiikkaan. Suora kinematiikka määrittää työkalun aseman, kun tiedämme robotin nivelten paikka-arvot. Epäsuoralla kinematiikalla taas määritetään tarvittavat nivelten asemat, jotta työkalu saadaan vietyä haluttuun pisteeseen avaruudessa. [5, s. 4–5] Seuraavissa alaluvuissa käsitellään millaisia koordinaatistoja robotilla on, mitä hyötyjä niiden käytöstä syntyy sekä miten koordinaatistojen tietoja voidaan siirtää toiseen koordinaatistoon.

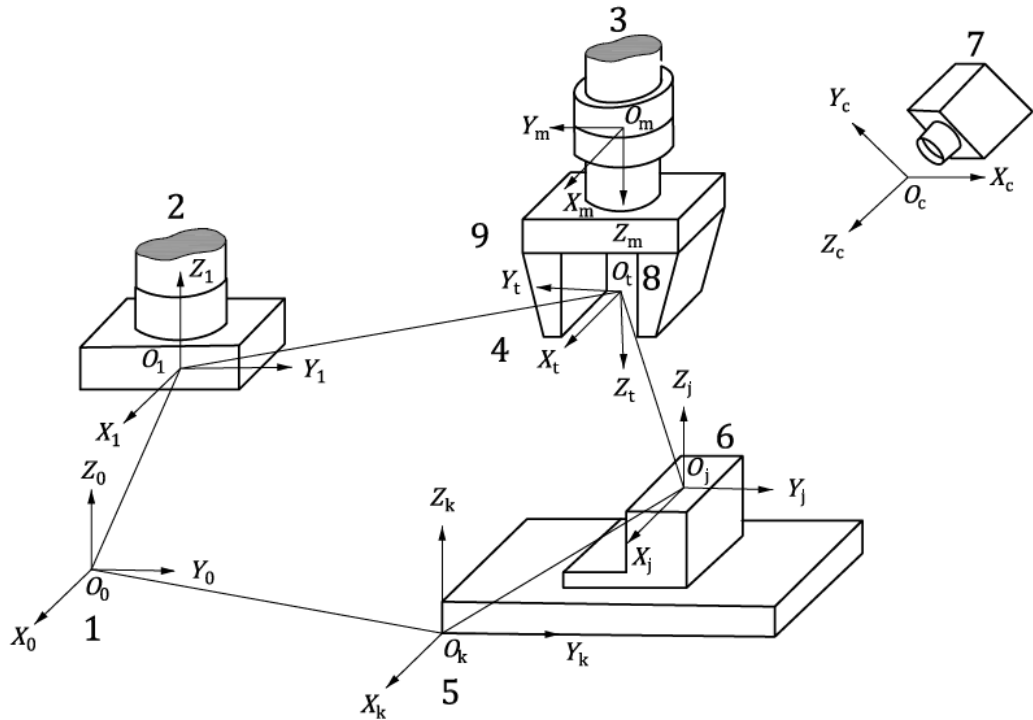
3.1 Teollisuusrobotin koordinaatistot

ISO-9787 standardi [10] määrittelee robotin koordinaatistot. Kaikki koordinaatistot ovat ortogonaalisia. Ne määritellään oikean käden säännön mukaan, joka on esitelty kuvassa 3.1. Jokainen sormi osoittaa eri akselin positiiviseen suuntaan ja ne ovat keskenään kohtisuorassa. Standardi määrittelee kahdeksan eri koordinaatistoa. [10] Näistä kolmea käytetään yleisesti teollisuusrobottien yhteydessä, jotka ovat maailma-, perus- ja työkalukoordinaatisto [14, s. 20].



Kuva 3.1 Oikean käden sääntö. [10]

Koordinaatistoja käytetään robotin liikkeen ohjaamiseen tehokkaasti. Ne helpottavat ja yksinkertaistavat erilaisten toimintojen toteuttamista, sillä jokaisella koordinaatistolla saadaan ohjattua robotin liikkeitä eri tavoilla, mikä mahdollistaa monimutkaiset liikeradat. Koordinaatistot vähentävät ohjelmointiin kuluvaan aikaan ja tekevät ohjelmista joustavampia, koska paikkakoordinaatteihin liittyvän tiedon syöttäminen on helpompaa. [11, s. 297, 301] Havainnollistavana esimerkkinä toimii uuden työkalun kiinnitys robottiin. Tässä tapauksessa asentajan tarvitsee määrittää uusi työkalukoordinaatisto sen sijaan, että koko ohjelma kirjoitettaisiin uudelleen huomioiden uudesta työkalusta johtuvat geometriset muutokset. Kuvassa 3.2 on esitetty standardin määrittelemät koordinaatistot [10].



Selite

- 1 Maailmakoordinaatisto
- 2 Peruskoordinaatisto
- 3 Liitäntälaittekoordinaatisto
- 4 Työkalukoordinaatisto
- 5 Työkoordinaatisto
- 6 Kappalekoordinaatisto
- 7 Kamerakoordinaatisto
- 8 Työkalupiste
- 9 Tarrain

Kuva 3.2 Robotin koordinaatistot. [10]

Peruskoordinaatisto on robotin alustaan kiinnitetty koordinaatisto. Sen positiivisen z-akselin suunta osoittaa robotin mekaanista rakennetta päin ja se on kohtisuorassa robotin alustaan nähden. Positiivisen x-akselin suunta osoittaa työalueen keskikohtaan. Robotin valmistajan työnä on määrittää peruskoordinaatiston origo. [10] Peruskoordinaatistoa käytetään, kun halutaan liikuttaa robotin työkalua avaruudessa, sillä se mahdollistaa työkalun liikuttamisen yhdensuuntaisesti pääakseleihin nähden. Tämä helpottaa robotin siirtämistä pitkiä matkoja. [11, s. 299]

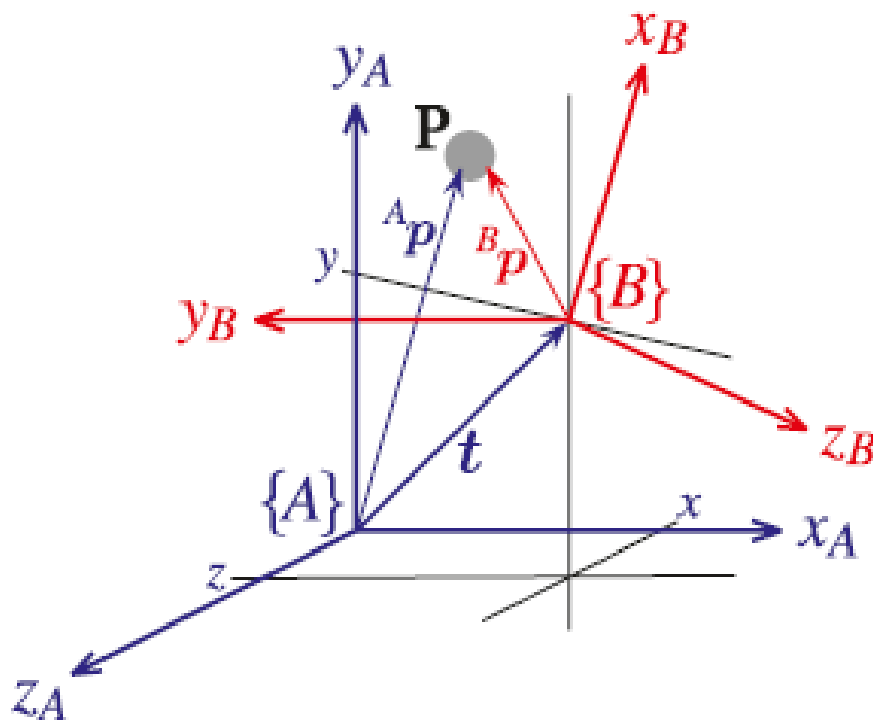
Työkalukoordinaatisto on työkalun keskipisteeseen sidottu koordinaatisto, joka on riippuvainen käytetystä työkalusta. Z-akselin suunta on yleensä työkalun kanssa samansuuntainen. [10] Työkalukoordinaatistoa hyödynnetään tarkkuutta vaativien liikkeiden ohjelmoinnissa, koska sen avulla robottia saadaan siirrettyä kohtisuorassa

työkalun keskipisteeseen nähden. Tämä mahdollistaa tarraimen liikuttamisen ilman, että työkalun orientaatio muuttuu. [11, s. 299–300]

Maailmakoordinaatisto on muuttumaton koordinaatisto, jota ei ole sidottu robotin liikkeisiin. Käyttäjä saa määrittellä koordinaatiston itse, kunhan sen positiivisen z-akselin suunta on kohtisuorassa vetovoiman kiihtyvyyksvektoria kohden ja ne ovat samalla suoralla. [10] Koordinaatiston avulla käyttäjä voi määrittellä origon mihin pisteeseen ja suuntaan tahansa, mikä helpottaa robottiohjelmien luomista [11, s. 300–301].

3.2 Koordinaattimuunnokset

Robotin käsittelemien esineiden paikkatiedot ovat aina sidottuja johonkin tiettyyn koordinaatistoon. Esineen täydelliseen asennon määrittämiseen kolmiulotteisessa maailmassa tarvitaan kuusi eri parametria, joista kolme kertovat sen paikkaa koordinaatistossa ja kolme sen orientaatiota pääakseleiden suhteen. Paikkavektoreiden avulla esineiden asento voidaan kuvata mielivaltaisen koordinaatiston suhteen. Kuvassa 3.3 on esitetty kaksi eri koordinaatistoa $\{A\}$ ja $\{B\}$ sekä piste P . Pisteiden paikka voidaan ilmaista sekä ${}^A\mathbf{p}$ että ${}^B\mathbf{p}$ paikkavektoreiden avulla. [4, s.17–18, 31–32]



Kuva 3.3 Pisteiden esitys kahden eri koordinaatiston avulla. [4]

Jotta pisteen tiedot voidaan siirtää koordinaatistosta toiseen, niiden välinen suhde paikan ja orientaation välillä on selvitettävä. Tätä varten on kehitetty erilaisia laskennallisia menetelmiä, joista yksi on homogeeninen transformaatiomatriisi. Homogeenisten transformaatiomatriisien avulla mielivaltainen piste yhdessä koordinaatistossa voidaan kuvata toisessa koordinaatistossa. Transformaatiomatriisi ${}^A\mathbf{T}_B$ B koordinaatistosta A koordinaatistoon voidaan esittää muodossa

$${}^A\mathbf{T}_B = \begin{bmatrix} {}^A\mathbf{p}_B & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (3.1)$$

jossa $\mathbf{t} = (x, y, z)$ kuvaa B koordinaatiston siirtymää ja ${}^A\mathbf{p}_B$ orientaatio suhteessa A :n koordinaatistoon. Kiertomatriisi ${}^A\mathbf{p}_B$ on ortogonaalinen, jossa $\mathbf{p} \in \mathbf{SO}(\mathbf{3}) \subset \mathbb{R}^{3 \times 3}$. Se kuvaa koordinaatiston kiertymistä x -, y - ja z -akseleiden suhteen. Transformaatiomatriiseja käytetään muuttamaan paikkavektoreiden tietoja koordinaatistoista toiseen tulon avulla. Esimerkiksi piste \mathbf{P} voidaan laskea A koordinaatistossa yhtälöllä ${}^A\mathbf{P} = {}^A\mathbf{T}_B \cdot {}^B\mathbf{P}$. [4, s. 34, 46, 58–59]

4. NUMEERINEN OHJAUS

Valmistavassa tuotannossa on tärkeää, että tarvittavat työvaiheet suoritetaan mahdollisimman nopeasti ja tarkasti. Työstettävien kappaleiden on oltava tasalaatuisia ja asetusaikojen lyhyitä. 1950-luvulla kehitetty numeerinen ohjaus (NC tai CNC, engl. numerical control tai computerized numerical control) tarjoaa tavan täyttää nämä tarpeet. [12] Ennen numeerista ohjausta työstön suoritti koneistaja käsikäyttöisillä koneilla. NC-tekniikka on kuitenkin korvannut tämän menetelmän erityisesti tuotannossa joustavuutensa, nopeutensa ja tarkkuudensa ansiosta. [20, s. 8] Vaikka menetelmää käytetään pääsääntöisesti työstökoneissa, voidaan sitä käyttää myös muiden koneiden ohjaamiseen, esimerkiksi koordinaattimittauskoneissa ja piirtureissa [11, s. 82][21].

Numeerinen ohjaus on työstökoneiden ohjaamista tietokoneen avulla. Työstökoneen tietokoneelle syötetään suoritettava ohjelma, joka sisältää halutut työstömenetelmän toiminnot. Tietokone muuttaa nämä käskyt toimilaitteiden ohjainarvoiksi ja lopulta liikkeiksi. Lopputuloksena on valmis kappale ilman ihmisen tekemiä työstövaiheita. [11, s. 76][12] Seuraavissa aliluvuissa käsitellään CNC-järjestelmien rakennetta, ohjelman sisältämiä tietoja sekä numeerisen ohjauksen tuottamia hyötyjä.

4.1 Järjestelmän osat

CNC-järjestelmästä voidaan erottaa neljä eri komponenttia, joista jokainen laite koostuu. Nämä ovat työstökone, paikoitusjärjestelmä, ohjausyksikkö ja suoritettava ohjelma. [11, s. 82]

Järjestelmän työstökone koostuu liikkuvasta työstöpöydästä, karoista, työkalumakasiineista sekä mahdollisista paletinvaihtajista. Työstökoneita voidaan luokitella niiden vapausasteiden tai karan asennon mukaan. Mitä useampi vapausaste työstökoneella on, sitä useamman eri akselin suunnassa kappaletta tai työkaluja voidaan siirtää ja kääntää. Yleisimmät koneet ovat 2-, 4- ja 5-akselisia, mutta akseleita voi olla jopa kahdeksan. Karan asennolla vaikutetaan työkalun paikkaan ja siten mahdollisiin työstömenetelmiin. Yleensä ne ovat joko vaaka- tai kohtisuorassa alustaan nähden. [11, s. 79,82–83][20, s. 14–17]

Paikoitusjärjestelmän tarkoituksena on mahdollistaa kaikki järjestelmän tarvitsemat liikkeet. Tällaisia ovat muun muassa työkalun asemointi, karan pyörimisnopeuden ja -suunnan säätäminen, työvälineen vaihdot sekä muut aputoiminnot. Systeemi koostuu toimilaitteista ja mittausjärjestelmästä. Aktuaattoreiden tarkoituksena on tuottaa mekaaninen energia, joka muutetaan halutuksi liikkeeksi vaihteiston avulla. Esimerkiksi työstöpöydän liike saadaan aikaiseksi pyörittämällä kuulamutteriruuvia, joka on kiinnitetty pöytään. Nykyaikaiset toimilaitteet ovat portaattomasti säädettäviä vaihtovirtamoottoreita, joiden avulla voimansiirron tehohäviöt saadaan poistettua ja moottoreiden pyörimisalueet asetettua laajoiksi. Mittausjärjestelmän tehtävänä on mitata luistimen hetkellistä asemaa, jota käytetään hyväksi moottoreiden säädössä sekä työkalujen asemoinnissa. Järjestelmän mittaustapa voidaan jakaa joko suoraan tai epäsuoraan mittaukseen. Suorassa mittauksessa mittalaite lukee luistimen asemaa rungon suhteen, kun taas epäsuorassa mittalaite määrittää paikan ruuvien kiertoliikkeestä. [11, s. 83–84][12][20, s. 16, 24–25]

Ohjausyksikön tehtävänä on hallita kaikkia toimintoja suoritettavan ohjelman mukaisesti ja tarjota rajapinta työstökoneen käyttäjälle. Tehtäviin kuuluu muun muassa työstöasetusten muokkaaminen, liikeratojen laskeminen, aputoimintojen käynnistäminen, kommunikointi muiden järjestelmien kanssa sekä muu looginen päättely. Ohjausyksikkö koostuu mikrotietokoneesta, erilaisista liittimistä ja käyttöpaneelistä. Mikrotietokone suorittaa kaikki järjestelmän laskutoimitukset, joita ovat muun muassa akselien liikkeiden synkronisointi ja rataohjauksen interpolointi. Nämä tiedot välitetään moottoreille ja muille laitteille liittimien avulla. Ne tarjoavat myös liityntäraajapinnan oheislaitteiden kytkemisen järjestelmään. Käyttöpaneelin tehtävänä on välittää ja vastaanottaa työstökoneen tietoja operaattorille. Paneelin kautta käyttäjä voi valita ohjaustavan, muuttaa suoritettavaa ohjelmaa tai hätäpysäyttää työstökoneen. [11, s. 84–93][12][20, s. 21–22, 26–28]

Työstökoneen suoritettava ohjelma on lista ohjeita, joka sisältää kaikki kappaleen valmistamiseen tarvittavat tiedot. Ne tallennetaan tietokoneen muistiin, josta ohjausyksikkö lukee ohjelmaa rivi kerrallaan. Ohjain muuttaa koodin servomoottoreiden käyttöyksiköiden ymmärtäviksi käskyiksi. [11, s. 93][12][20, s. 20]

4.2 NC-ohjelma

NC-ohjelmien syntaksi perustuu ISO-6983 standardiin [9]. NC-ohjelma koostuu lauseista. Yksi lause on yhden rivin mittainen ja se sisältää yhteen työvaiheeseen liittyvät tiedot aina liikekäskyistä aputoimintoihin saakka. Se on ohjelman pienin osa, jota työstökone voi kerralla toteuttaa. Yksi lause koostuu yhdestä tai useammasta sanasta. Sana on pienin käsky, jonka avulla työstökoneen toimintoja voidaan ohjata.

Sana koostuu osoitekirjaimesta ja sen jälkeen tulevista numeroarvoista. Osoitteella ilmaistaan käskyn tarkoitus ja numeroilla siihen liittyvä tieto. Osoite voi sisältää myös etumerkkejä, desimaalipisteitä tai muita erikoismerkkejä. [11, s. 93–94][12][20, s. 46] Kuvassa 4.1 on esitetty katkelma työstökoneen ohjelmasta.

```
N10 G90 G20
N20 M06 T22
N30 G00 X0 Y0 Z.1
N40 M03 S2000 F4
N50 G00 X1.5 Y1.25
N60 G01 Z-1.0
N70 G01 Z.1
N80 G00 X2.5 Y2.75
N90 G01 Z-1
N100 G00 Z0.1000
```

Kuva 4.1 Osa NC-koodin esimerkkiohjelmasta. [11]

Ohjelman sisältämä tieto on jaettavissa kahteen eri osaan: liikekäskyihin ja kyt-kentätietoihin. Liikekäskyt sisältävät liikenopeuteen ja päätepiesteeseen liittyvää tietoa. Näitä käskyjä kutsutaan G-koodiksi niiden osoitekirjaimen takia. Komentojen avulla määritellään liikkeitä, työkiertoja ja koordinaatistoja. [12][20, s. 20,46–47] Kuvan 4.1 kolmas rivi esittää pikaliikekäskyä origon koordinaattipisteisiin [11, s. 93–94].

Kytkentätiedot sisältävät aputoimintojen ohjaamiseen liittyvät toiminnot. Näitä käskyjä kutsutaan M-koodiksi niiden sisältämän etuliitteen takia. M-koodin sana rakentuu M-kirjaimesta ja kahdesta numerosta. Näillä käskyillä ohjataan lisätoimintoja, jotka eivät liity työstövaiheiden liikkeisiin, kuten esimerkiksi lastuamismesteen syöttöä taikka työkalun vaihtoa. M-koodien toiminnot ovat laitteistoriippuvaisia, joten niiden siirtäminen työstökoneelta toiselle voi vaatia muokkauksia. [9][20, s. 20,47] Kuvassa 4.1 tällaista käskyä esittää toisella rivillä oleva työkalunvaihtokomento ”M06” [11, s. 93].

4.3 Käytön edut

NC-tekniikan käytöstä on havaittu useita erilaisia höytyjä, jotka ovat:

1. lisääntynyt tuottavuus
2. korkeampi laatu
3. monimutkaisten kappaleiden valmistaminen
4. kustannusrakenteen parantuminen
5. vaaditun taitotason alentuminen
6. tuotesuunnitteluun kuluneen ajan lyhentyminen. [11, s. 107–108]

Numeerinen ohjaus on vähentänyt työvoiman tarvetta eri työstövaiheissa. Sen avulla valmistus- ja läpimenoajat ovat lyhentyneet, työstön tarkkuus parantunut sekä käyttöaste kohonnut. Nämä kaikki tekijät ovat edistäneet tuottavuuden kasvua. [11, s. 107][12][20, s. 8]

Siinä missä käsinohjattujen työstökoneiden kappaleiden eroavaisuudet ovat riippuvaisia käyttäjän taidoista, NC-koneella valmistettujen kappaleiden variaatio on huomattavasti pienempää. Poistamalla inhimillisistä tekijöistä johtuvat vaikutukset ja korvaamalla ne tietokoneohjauksella, voidaan taata eräkappaleiden samankaltaisuus, jolloin laatua saadaan parannettua. [11, s. 107][12][20, s. 10–11]

Numeerisen ohjauksen kehityksen päätavoitteena oli mahdollistaa geometrisesti monimutkaisten kappaleiden valmistus. Etenkin käsikäyttöisillä työstökoneilla on hankala ohjata liikerataa samanaikaisesti kolmen eri akselin suhteen. NC-tekniikka on kuitenkin mahdollistanut tämän, jolloin kappaleen muodosta johtuvista rajoitteista on päästy eroon. [11, s. 108][12][20, s. 8]

NC-koneet vaativat lyhyempiä asetusajoja perinteisiin koneisiin verrattuna. Tekniikalla on saatu myös vähennettyä koneiden määrää ja siten lattiatilaa, sillä nykyiset NC-koneet voivat hoitaa usean eri käsikäyttöisen työstökoneen tehtäviä. Lyhyemmät läpäisyajat, pienemmät varaosavarastot ja vähentyneet materiaalikulut ovat puolestaan laskeneet valmistukseen sitoutunutta pääomaa. [11, s. 108][12][20, s. 8]

Käsinohjattujen laitteiden käyttäjän ammattitaidolla on suuri merkitys kappaleen valmistuksessa. Koska koneen operaattorin ei enää tarvitse suorittaa työstövaiheita itse, ovat työtehtävät muuttuneet yhä enemmän koneen toiminnan valvontaan ja työn laadun tarkkailuun. Tällöin nykyisiltä koneenkäyttäjiltä vaaditaan vähemmän työstötaitoja. [11, s. 108][12][20, s. 11]

Yleensä valmiiseen tuotteeseen halutaan tehdä pieniä parannuksia. Nämä muutokset ovat helposti toteutettavissa NC-työstökoneissa, sillä suoritettavan ohjelman rivejä voidaan muokata vapaasti. Valmistettavien kappaleiden työstövaiheita voidaan myös simuloida, jolloin mahdolliset virheet ovat huomattavissa valmistuksen varhaisessa vaiheessa. Nykyiset ohjelmistot tukevat myös CAD-järjestelmien (engl. computer aided design) integraatiota parantaen ohjelmoinnin automatisointia. [11, s. 108][12][20, s. 8]

Vaikka NC-työstökoneiden käyttäminen tuo huomattavia hyötyjä, siihen liittyy tiettyjä rajoituksia, joiden takia tekniikka ei sovi kaikkiin tapauksiin. Yhtenä heikkoutena on sen korkeat investointikustannukset. Laitokselta vaaditaan korkeaa käyntiasetetta ja valmistusmääriä, jotta sijoituksesta tulisi kannattava. Lisäksi NC-koneiden huoltokustannukset ovat huomattavasti korkeampia perinteisiin koneisiin verrattuna. Tämä johtuu kalliista elektroniikasta ja huoltotyöntekijöiden tarvitsemasta korkeammasta ammattitaidosta, sillä koneiden rakenne on paljon monimutkaisempi käsikäyttöisiin verrattuna. Viimeisimpänä yrityksen on hankittava lisää työvoimaa, sillä NC-tekniikan käyttäminen vaatii ohjelmointitaitoisia henkilöitä, joita työpaikoista ei välttämättä löydy entuudestaan. [11, s. 108–109][12]

5. ROBOTIN OHJELMOINTIMENETELMÄT

Joustavuus on tärkeä ominaisuus roboteissa, jotta robotteja voitaisiin hyödyntää monissa eri tehtävissä ilman suuria rakenteellisia muutoksia. Robottien monipuolinen käyttäminen edellyttää sitä, että niiden suorittamia tehtäviä pystytään muokkaamaan. Tätä tarkoitusta varten roboteille on kehitetty erilaisia ohjelmointimenetelmiä ja ohjelmointikieliä. [1, s. 19] Niillä ohjataan robotin liikkeitä, logiikkaa ja viestejä oheislaitteille, esimerkiksi robottisolun hihnakuljettimelle [5, s. 339].

Sähkömekaaniset kytkennät tarjosivat ensimmäisen tavan robottien ohjelmoimiseen. Robotin nivelet ohjelmoitiin ajettavaksi vaihe kerrallaan kohti mekaanisia rajakatkaisimia. [1, s. 19–20][14, s. 78] Teollisuuden kasvulle on tärkeää, että robottisolut ovat joustavia, modulaarisia ja mukautuvia. Tämän lisäksi robottien tahtiaikojen ja ohjelmointisyklien on oltava mahdollisimman lyhyitä. Nämä tarpeet ovat ohjanneet kehittämään yhä helpompia robotin ohjelmointitapoja. [7] Robotin ohjelmointimenetelmät voidaan jakaa online- ja offline-ohjelmointiryhmiin. Online-menetelmissä ohjelmointi tapahtuu robottiin kytketyn ohjelmointijärjestelmän avulla, jolloin robotti on otettava irti tuotannosta ohjelmoinnin ajaksi, kun taas offline-ohjelmointi tapahtuu robotista erillään olevan ohjelmointijärjestelmän kautta. [1, s. 19][3][12] Seuraavissa alaluvuissa käsitellään näiden ryhmien ohjelmointimenetelmiä.

5.1 Johdattamalla ohjelmointi

Varhaisinta robottien ohjelmointimenetelmää kutsutaan johdattamalla ohjelmoinniksi. Menetelmä perustuu liikeratojen tallentamiseen instrumenttinauhurille. Robotin toimilaitteet vapautetaan ihmisen liikuteltaviksi, minkä jälkeen robotin käsivartta kuljetetaan haluttujen pisteiden kautta liikuttelemalla sen niveliä käsivoimalla. Robotin nivelten paikka-antureiden lukemat ja mahdollisten työkalujen tai tarrain-ten asennot tallennetaan liikkeen aikana nauhurille. Liike saadaan toistettua yhdistämällä nauhuri nivelten toimilaitteiden säätöpiireihin ohjearvoiksi ja kelaamalla nauhaa halutulla nopeudella. Johdattamalla ohjelmointi kuuluu online-ohjelmointiryhmään. [1, s. 20][14, s. 78]

Alander ja Niemi toteavat teoksessaan [1, s. 20], että johdattamismenetelmä on help-

pokäyttöinen ja helposti toteutettavissa. Se ei myöskään vaadi kalliita tietokoneita, mikä teki siitä suosituksen teollisuudessa [1, s. 20]. Vaikka johdattamalla ohjelmointi on laajimmalle levinnyt ohjelmointimetodi, on sen käytössä silti rajoituksia. Jos robotin ohjelmaan halutaan tehdä muutoksia, se joudutaan ohjelmoimaan alusta lähtien uudestaan, sillä johdattamismenetelmästä puuttuvat toistorakenteet. Tämän lisäksi ohjelmista on hankala saada tarkkoja, sillä robotin anturitietoja ei voida käyttää hyödyksi robotin ohjauksessa. Magneettinauhujen arkistointi ja käsittely tuottivat myös rajoitteita tämän menetelmän käytölle, mutta nykyisin kovalevyt ovat ratkaisseet ongelman. Menetelmä on edelleen käytössä teollisuudessa, esimerkiksi maalaus- ja hitsausroboteissa, mutta muut menetelmät ovat korvaamassa sitä useimmissa sovelluksissa. [1, s. 20][14, s. 78]

Johdattamalla ohjelmointi on tulossa takaisin voimaohjattujen robottien yhteydessä. Siinä missä vanhassa menetelmässä robotin moottoreista katkaistiin virrat ja robotin niveleitä liikuteltiin käsivoimin, uudessa metodissa moottoreissa pidettäisiin virta päällä ja liikuttamisessa käytettäisiin voimasensoreita hyväksi. Tämän avulla menetelmästä saataisiin turvallisempi ja tarkempi verrattuna vanhaan. [18] Voimaohjatut robotit ovat kuitenkin vielä verrattain uutta teknologiaa, jota ei ole vielä otettu laajasti käyttöön eri teollisuudenaloilla.

5.2 Opettamalla ohjelmointi

Opettamalla ohjelmointi -menetelmä perustuu pisteiden opettamiseen robotille. Robottia ohjataan käsiohjaimella, jonka avulla robotin niveleitä liikutellaan haluttuihin pisteisiin. Pisteiden asemat tallennetaan muistiin, minkä jälkeen niitä voidaan käyttää ohjelmoinnissa. Robotin liikkuminen, työkalujen käyttäminen ja muu logiikka ohjelmoidaan tämän jälkeen joko käsiohjaimella tai käyttämällä robotin ohjelmointikieliä. [1, s. 20][5, s. 189][14, s. 20] Opettamalla ohjelmointi voi kuulua sekä online- että offline-ohjelmointiin riippuen käytettävästä ohjelmointijärjestelmästä [1, s. 20]. Menetelmän käytön hyötyjä ovat helppo integraatio eri systeemeihin, pienet kustannukset ja lyhyet kehittämisajat [16].

Vaikka menetelmä on laajasti käytössä teollisuudessa, löytyy siitä muutamia haittapolia. Robotin käsiohjaimen käyttö vaatii koulutetun henkilön, sillä operaattorin pitää tietää, mihin koordinaatistoon robotti on konfiguroitu, ja huomioitava tämä tieto ohjelmoinnissa. Toisaalta robotin ohjelmointi voi olla haastavaa ja hidasta, jos työstettävä kappale on monimutkainen tai jos prosessi on vaikea. Lisäksi ohjelman testaaminen ja uudelleenmuokkaaminen vaativat paljon aikaa. Robotin ohjelmoijan on myös usein mentävä ohjelmoinnin aikana robotin vaara-alueelle. Viimeiseksi robotti on otettava pois tuotannosta ohjelmoinnin ajaksi, jolloin sitä ei voida käyttää

valmistuksessa. [16] Tämä ongelma ilmenee vain, jos robottia ohjelmoidaan siihen kiinnitetyn ohjelmointijärjestelmän kautta [1, s. 20].

5.3 Mallipohjainen ohjelmointi

Mallipohjainen ohjelmointi pohjautuu tuotannosta tehdyn 3D-mallin käyttämiseen robotin ohjelmoinnissa. Tässä menetelmässä robotista, sen työskentelyalueesta ja käsiteltävästä kappaleesta luodaan 3D-mallit käyttämällä CAD-järjestelmiä, joiden tuottamat simulointimallit integroidaan robotin ohjelmointijärjestelmään. Tämän jälkeen robotin ohjelmointi onnistuu ohjelmointijärjestelmän kautta tekstuaalisesti tai graafisesti. Robotin ohjelmoija voi etukäteen mallintaa ja testata robotin toimintaa. Jos robotti näyttää toimivan halutulla tavalla, voidaan ohjelma ladata fyysiselle robotille. [14, s. 81–83][16] Ohjausjärjestelmät tarjoavat useita erilaisia moduuleja, joiden avulla robottisolun mallia saadaan muokattua. Tällaisia osia ovat muun muassa kinematiikan suunnittelumoduulit ja robotin oheislaittekirjastot. [14, s. 82] Mallipohjainen ohjelmointi kuuluu offline-ohjelmointi-ryhmään [1, s. 22].

Mallipohjaisen ohjelmoinnin käytössä on havaittavissa useita erilaisia hyötyjä. Robottia ei tarvitse ottaa tuotannosta irti ohjelmoinnin yhteydessä, mikä näkyy tuotantoaikojen paranemisessa. Menetelmällä tuotetut robotin ohjelmat ovat joustavia, sillä ohjelmia voidaan muokata mistä kohtaa tahansa. Ohjelmien testaaminen on myös helppoa, sillä niiden toimintaa voidaan simuloida ennen ohjelman lataamista robotille. Tämä vähentää esimerkiksi virheitä ja parantaa turvallisuutta. Se on myös todistettu kustannustehokkaaksi ratkaisuksi massatuotannossa. [16] Mallipohjainen ohjelmointi sopii tilanteisiin, joissa tuotteen suunnittelussa käytetään CAD-järjestelmiä, tai kun robottien ohjelmointi tuotannossa on hengenvaarallista, esimerkiksi ampumatarviketeollisuus. Se soveltuu myös tapauksiin, joissa valmistusprosessi vaatii paljon paikoituspisteitä. [14, s. 82]

Ohjelmointimenetelmässä on myös tiettyjä heikkouksia, joiden vuoksi se ei sovelu kaikkiin mahdollisiin tapauksiin. Mallipohjainen ohjelmointi vaatii korkeita investointikustannuksia ja pitkiä käyttöönottoja. Robotin työalue on mallinnettava ja kalibroitava vastaamaan todellista ympäristöä aiheuttaen paljon lisätöitä ennen varsinaista ohjelmointia. Se vaatii myös korkean tason ohjelmointiosaamista, jota ei löydy useimmilta nykyaikaisten robottien operaattoreilta. He ovat usein vastuussa robotin ohjelmoimisesta, joten ohjelmointimenetelmän käyttäminen vaatisi uusien osaamisen hankkimista. [16]

5.4 Tekstipohjainen ohjelmointi

Tekstipohjaisessa ohjelmoinnissa robotin ohjelmat kirjoitetaan robotista erillään olevalla tietokoneella. Halutut toiminnot kirjoitetaan ohjelmointikielellä, jonka jälkeen valmis ohjelma ladataan robotin ohjaimelle. Ohjain kääntää ohjelman säätimien ohjearvoiksi. Ohjelmointiin käytetään yleensä robottikohtaisia ohjelmointikieliä tai korkean tason rakenteellisia kieliä. [11, s. 43] Tekstuaalinen ohjelmointi kuuluu offline-ohjelmointiryhmään [1, s. 19].

Tekstipohjaisessa ohjelmoinnissa on paljon samanlaisia hyötyjä kuin mallipohjaisessa. Robottia ei tarvitse ottaa ohjelmoinnin ajaksi pois tuotannosta, ohjelmointi on joustavaa ja ohjelmia on mahdollista testata ennen käyttöönottoa. [11, s. 296] Se sopii myös hyvin tilanteisiin, joissa robottisolu sisältää paljon oheislaitteita tai sensoreita. Erityisesti monimutkaisen logiikan ja kommunikoinnin toteuttamiseen menetelmä sopii hyvin. [6, s. 282]

Menetelmän heikkouksina on standardin ohjelmointikielen puuttuminen ja ettei työasemasta ole saatavilla tarkkaa mallia. Jokaisella laitevalmistajalla on oma ohjelmointikieli, mikä vaikeuttaa ohjelmien uusiokäyttöä ja robottien integrointia monitahoisiin järjestelmiin. Tämän lisäksi ohjelmia on muokattava ja testattava iteratiivisesti, jotta ohjelma toimii oikeassa ympäristössä. [1, s. 31][5, s. 359–340][11, s. 302]

6. ASETUSAIKOJEN LYHENTÄMINEN

Seuraavissa alaluvuissa kuvataan työstösolu ja nykyisen toteutustavan heikkoudet, jonka jälkeen esitetään erilaisia ohjelmallisia toteutusehdotuksia, joilla yritys voi vaikuttaa työstösolun ohjelmoimiseen kuluvaan aikaan. Ensimmäisessä alaluvussa selvitetään putkien valmistusprosessi, järjestelmän eri komponenttien vuorovaikutukset sekä mikä on nykyisen putkenvalmistuksen rajoittava tekijä. Toisessa alaluvussa esitetään erilaisia tapoja, joiden avulla solun uusien putkityyppien valmistukseen kuluvia asetuksaikoja voitaisiin lyhentää. Toteutustapoja vertaillaan investointikustannuksien ja teknisen haastavuuden näkökulmasta.

6.1 Järjestelmän kuvaus

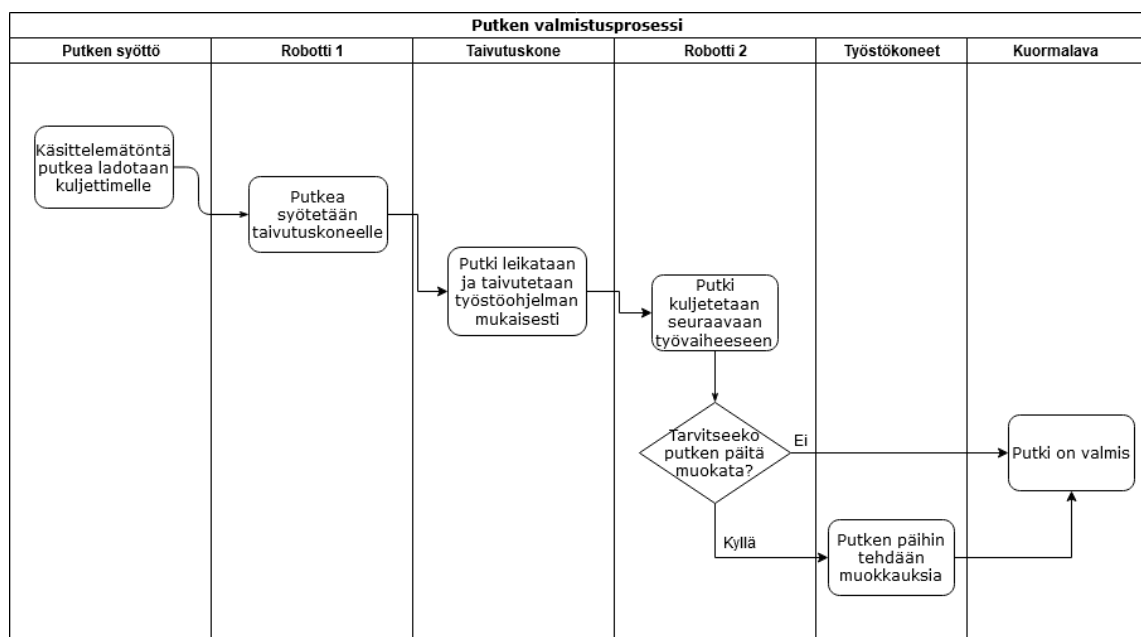
Tuotantosolu koostuu syöttökourusta, kahdesta robotista, CNC-putkentaivutuskoneesta, työstökoneista, kuormalavasta ja solun ohjaimesta. Ensimmäinen robotti on Fanucin teollisuusrobotti, jonka tehtävänä on ottaa syöttökourusta taivuttamatonta putkea ja antaa se taivuttimelle työstettäväksi. Putkentaivutuskone on numeerisesti ohjattu hydraulikalla toimiva laite. Sen tehtäviin kuuluu leikata ensimmäiseltä robotilta tuleva putki sopivan kokoisiksi osiksi ja taivutella nämä osat työstöohjelman mukaisesti.

Solun toinen robotti on Fanucin valmistama 6-akselinen teollisuusrobotti, joka on sijoitettu putkentaivutuskoneen oikealle puolelle solussa. Robotin tehtävänä on ottaa taivutuskoneen taittelema putki ja siirtää se seuraaviin työstövaiheisiin. Valmistettavasta putkityypistä riippuen putken päille saatetaan tehdä avaruksia tai muita operaatioita. Näitä varten tuotantosoluun on asetettu erilaisia työstökoneita toisen robotin lähetyville. Robotin tehtävänä on lopuksi siirtää valmis putki kappaleiden noutopisteeseen. Solun ohjain on PLC-ohjelmistolla toteutettu automaatiojärjestelmä, joka on kytketty tuotannonohjausjärjestelmään. Sen tehtävänä on kontrolloida solun muita koneita ja tarjota operaattorille käyttöliittymä, jonka kautta operaattori voi muun muassa katsoa vikailmoituksia ja hallita solun tilaa.

Solussa tapahtuva valmistus voidaan jakaa erilaisiin prosesseihin koneiden käyttötarkoitusten mukaan. Valmistusprosessi on esitetty kuvassa 6.1. Ensimmäisessä vai-

heessa operaattori syöttää taivuttamatonta putkea solun syöttökouruun. Tämä tapahtuu mekaanisesti solun ulkopuolelta. Syötettävä putki on kuusi metriä pitkä, josta saadaan valmistettua eri määrä tuotteita riippuen niiden geometrisista mitoista. Tämän vaiheen jälkeen solun pääsääntöinen valmistusprosessi tapahtuu solun ohjaimelta tulevien käskyjen mukaan. Ohjain saa tuotannonohjausjärjestelmästä tilauksia eri tyyppisistä putkista, joiden valmistusjärjestyksen ohjain optimoi sisäisellä logiikallaan. Tämän jälkeen ohjain antaa ensimmäiselle robotille käskyn siirtää putken kourulta työstökoneelle. Kun työstökone on vastaanottanut robotilta tulevan putken, ohjain kertoo koneelle putkesta valmistettavien putkityyppien tiedot. Työstökone valmistaa yhtä putkityyppiä kerrallaan lataamalla muististaan oikean ohjelman ja taittelemalla putkea koodin käskyjen mukaan.

Kun työstökone on saanut taivuteltua putken, lähettää se viestin solun ohjaimelle, joka puolestaan välittää seuraavalle robotille tiedot putkityypistä. Robotti lataa sisäisestä muististaan ohjelman ja suorittaa sen. Robotti kuljettaa putken työstölaitteille jatkokäsittelyä varten tai suoraan kuormalavalle riippuen valmistettavasta putkityypistä. Lopuksi valmiit tuotteet kerätään kuormalavalta operaattorin toimesta.



Kuva 6.1 Tuotantosolun putken valmistusprosessi.

Tällä hetkellä uusien putkityyppien valmistus perustuu takaisinmallinnukseen. Kun tuotantosolun halutaan tuottavan uudentyyppistä putkea, valmistettavasta putkesta hankitaan fyysinen kopio. Mallikappale laserskannataan, jonka avulla putkesta saadaan tehtyä tarkka 3D-malli CAD-ohjelmistolla. Mallin perusteella ohjelmistolla saadaan myös generoitua automaattisesti putkentaivutuskoneelle sopiva

työstöohjelma. Kuitenkin putkentaivutuskoneelta putkia keräävän robotin ohjelmointi perustuu tällä hetkellä opettamalla ohjelmointiin, joten ohjelman luominen ei onnistu automaattisesti vaan se joudutaan tekemään manuaalisesti. Kun tuotantosolu on valmistamassa ensimmäistä kertaa uudenlaista putkityyppiä, operaattori luo robotille sopivan ohjelman liikuttamalla robottia käsiohjaimella ja tallentamalla liikeradan sen avulla. Tämä on hidas prosessi joka lisää tuotannon asetusaikoja merkittävästi, sillä koko tuotantosolu joudutaan pysäyttämään siksi aikaa, kun operaattori opettaa robotille uuden putkityypin liikeratoja. Yritys tahtois generoida robotin ohjelmat automaattisesti, koska solussa tuotetaan vuosittain useita uusia putkityyppisiä. Ohjelmien automatisointi lisäisi solun joustavuutta ja parantaisi tuottavuutta.

6.2 Toteutusehdotuksia

Yhtenä ratkaisuna yritys voisi siirtyä käyttämään mallipohjaista ohjelmointia. Markkinoilla on saatavilla simulaatio-ohjelmia, joiden avulla työstösolusta voitaisiin luoda ohjelmalla simuloitava malli. Mallin avulla luotavia työstöohjelmia on mahdollista siirtää suoraan fyysisille koneille, mikä nopeuttaisi uusien putkien valmistusta. Ohjelmistojen käyttämisen yhtenä suurena etuna on, että robotin ohjelmia voi luoda, testata ja analysoida tuotannon ulkopuolella tuotantoa pysäyttämättä. Koko tuotantosolun toimintaa voidaan mallintaa ja validoida ennen muutoksien siirtämistä tuotantoon, jolloin tuotantoseisakkien määrä saadaan minimoitua.

Toteutustavan haittapuolina yritykselle ovat kuitenkin korkeat investointikustannukset ja käytön vaativuus. Simulaatio-ohjelmien hyödyntäminen vaatii erikoisohjelmien lisenssejä ja erikoisosaamista, joita yritykseltä ei löydy entuudestaan. Simulaatio-ohjelmaa varten koko tuotantosolusta ja siinä tapahtuvassa tuotantoprosessista on luotava tarkka simulointimalli ennen kuin simulointijärjestelmää päästään käyttämään. Malli on myös haavoittuvainen solussa tapahtuville muutoksille, sillä pienetkin vaihdokset solun komponenttien sijoittelussa joudutaan kalibroimaan uudelleen ohjelman puolella. Simulaatio-ohjelmien käyttäminen on haastavampaa opeutusohjaimen verrattuna, sillä ohjelmistojen käyttäminen vaatii erikoiskoulutusta ja tietämystä niiden toiminnasta. Ohjelmointi vaatii myös alussa paljon verifiointia, joka hidastaa tuotantoon siirtymistä erityisesti ohjelman käyttöönotossa. Toisaalta yrityksen tuotantomäärät ovat sen verran suuria, että investointia voidaan pitää kannattavana.

Toisena lähestymistapana yritys voisi käyttää nykyistä valmistusprosessiaan monipuolisemmin hyväksi ja laskea etukäteen robotille sopivat liikeradat ja tarttumapisteet. Koska valmistettavista kappaleista määritetään takaisinmallinnuksen yhteydessä jo tarkat geometriset mitat ja 3D-mallit, pystytään näiden tietojen pohjalta

määrittelemään robotille sopiva ohjelma etukäteen ilman tuotannon pysäyttämistä. Tämä ohjelma voitaisiin luoda ja ladata automaattisesti käyttäen hyödyksi tekstipohjaista ohjelmointia. Ohjelmien luomisessa voitaisiin käyttää hyödyksi sekä 3D-malleja, että työstökoneelle luotuja NC-ohjelmia. Työstökoneen G-koodista voidaan suoraan selvittää käytetyt koordinaatit M-koodin käskyistä, jotka ovat vietävissä robotille. Tietämällä työstökoneen sijainnin suhteessa robottiin, voidaan koordinaatistomuutoksilla viedä geometriset muodot robotin ohjelmalle sopivassa muodossa ilman ylimääräisiä kalibrointeja tai oheislaitteita paikannukseen liittyen. Alustavien tutkimusten mukaan solun ohjain pystyisi vastaamaan ohjelman generoimisesta, koska sillä on tieto valmistettävien putkien geometrisista mitoista ja datayhteys robottiin, jonka kautta ohjain voi lähettää robotille ohjelmia. Tällöin uusista putkityypistä johtuvat muutokset ohjelmissa voitaisiin ottaa huomioon ja siirtää tuotantoon ilman operaattorin vaikutusta.

Menetelmä sopii kohdeyrityksen tarpeisiin. Se vaatii pienemmät investointikustannukset sekä automatisoi ohjelmien generoimisen tehokkaammin verrattuna simulaatio-ohjelman käyttämiseen. Menetelmän yhtenä haittapuolena on generoimisen ylläpidettävyyden vaatimat muutokset jouduttaisiin tilaamaan luultavimmin alihankkijalta järjestelmän arkkitehtuurin takia. Jos tuotantosolun asemoinnissa tapahtuisi muutoksia, täytyisi nämä muutokset ottaa huomioon uudelleen solun ohjaimessa. Asemointimuutokset ovat kuitenkin harvinaisia, joten tämä ei ole kriittinen haitta. Alustavien tutkimusten perusteella solun sijoittelussa tai valmistusprosessissa ei tule olemaan suuria muutoksia lähiaikoina, joten tämä menetelmä sopii yksinkertaisuutensa takia paremmin yrityksen tarpeisiin kuin simulaatio-ohjelman käyttöönotto.

7. YHTEENVETO

Tämän kandidaatintyön tarkoituksena oli etsiä menetelmiä, joilla kohdeyritys voisi vähentää tuotannossa esiintyviä asetusajoja muuttamalla yhden solun robotin ohjelmia. Työ piti sisällään kirjallisuuskatsauksen, jonka alussa perehdyttiin asetusajojen merkitykseen tuottavassa teollisuudessa. Löydetyn tiedon perusteella voidaan todeta, että yrityksen kannattaa pyrkiä vähentämään asetusajoja kaikin mahdollisin keinoin. Tärkeimpiä saavutettavia hyötyjä lyhyistä asetusajoista ovat

- tuotannon resurssien käytön tehostuminen
- tuotannon ohjattavuuden paraneminen
- joustavampien eräkokojen valmistuksen mahdollistaminen
- yrityksen kulujen pienentyminen
- työtalon kasvaminen.

Yritys pystyy lyhentämään asetusajoja joko vakioimalla tuotannon toimintatapoja tai automatisoimalla koneita.

Kirjallisuuskatsauksessa selvitettiin seuraavaksi, miten robotti käsittelee paikoitukseen liittyviä tietoja. Suoraa ja epäsuoraa kinematiikkaa käytetään robotin asemoinnin määrittelyssä. Robotille voidaan määrittellä eri koordinaatistoja, joiden avulla paikoitukseen liittyvien tietojen käsittely helpottuu. Tämä mahdollistaa monimutkaisten liikeratojen toteuttamisen ja helpottaa haluttujen toimintojen suorittamista. Robotilla yleisesti käytettyjä koordinaatistoja ovat muun muassa maailma-, perus- sekä työkalukoordinaatisto. Robotti voi siirtää paikoitukseen liittyviä tietoja koordinaatistojen välillä käyttäen hyväksi transformaatiomatriisien ja paikkavektoreiden tuloja.

Työssä perehdyttiin myös numeerisesti ohjattuihin työstökoneisiin. Numeerisesti ohjatut työstökoneet ovat autonomisia koneita, jotka koostuvat työstökoneesta, paikoitusjärjestelmästä, ohjausyksiköstä ja suoritettavasta ohjelmasta. Koneiden käyt-

täminen mahdollistaa automaattisen tuotannon, joka puolestaan näkyy lisääntyneenä tuottavuutena ja tehokkaammassa kappaleiden valmistuksessa. Työstökoneiden liikkeiden ohjaamisessa käytetään G-koodia, joka koostuu käskyistä ja niihin liittyvistä parametreista. Koodilla määritellään haluttuja liikkeitä, työkiertoja sekä ohjelmointia helpottavia koordinaatioita.

Viimeisimpänä asiana kirjallisuuskatsauksessa perehdyttiin erilaisiin menetelmiin, kuinka teollisuusrobotteja voidaan ohjelmoida. Ohjelmointimenetelmät voidaan jakaa kahteen eri ryhmään riippuen siitä, täytyykö ohjelmoinnin tapahtua robotin lähettyvillä vai voidaanko se toteuttaa etänä. Erilaisia ohjelmointimenetelmiä ovat johdattamalla ohjelmointi, opettamalla ohjelmointi, mallipohjainen ohjelmointi sekä tekstipohjainen ohjelmointi, joista kaksi jälkimmäistä menetelmää soveltuvat toteuttavaksi etänä robotista. Kirjallisuudesta tehtyjen havaintojen perusteella voidaan todeta, että johdattamalla ja opettamalla ohjelmointi sopivat tilanteisiin, joissa liikeradat eivät ole monimutkaisia tai robotin ei tarvitse keskustella useiden oheislaitteiden kanssa. Ne ovat myös verrattain halpoja menetelmiä verrattuna mallipohjaiseen ja tekstipohjaiseen ohjelmointiin. Etänä toteutettavien ohjelmointimenetelmien vahvuudet ovat toisaalta vaativien toiminnallisuuksien toteuttamisen helppous sekä ohjelmien kehitys ilman tuotannon keskeyttämistä.

Työn lopussa esitettiin kohdeyrityksen tuotantoprosessi ja nykyisen toteutustavan ongelmat pitkille robotin asetusajoille. Samalla esitettiin kaksi erilaista toteutusehdotusta, joilla yritys voisi päästä tavoitteisiinsa. Ensimmäisessä toteutusmenetelmässä yritys siirtyisi ohjelmoimaan solun robotteja käyttämällä hyväksi mallipohjaista ohjelmointia kun taas toinen toteutusmetodi keskittyi kuvailemaan tekstipohjaiseen ohjelmointiin liittyvää ratkaisua.

Näistä kahdesta vaihtoehdosta yritykselle suositeltiin jälkimmäistä, sillä se täyttää yrityksen asettamat kriteerit pienemmillä muutoksilla nykyiseen valmistusprosessiin. On kuitenkin huomionarvoista, että tämä päätelmä ei ole yleistettävissä kaikkiin tapauksiin, koska menetelmien vertailu tapahtui ottaen huomioon kohdeyrityksen tarpeet, valmistusprosessin ja solun rakenteen. Nämä ovat yksilöllisiä tekijöitä, jotka vaikuttavat menetelmien soveltuvuuteen tapauskohtaisesti.

Tämän työn pääpainona oli perehtyä kohdeyrityksen ongelman taustatietoihin sekä esitellä ratkaisun toteutusehdotuksia. Työssä pystyttiin luomaan kattava kuva ase-tusaikojen merkityksestä tuottavalle teollisuudelle sekä selvittämään valmistuksessa käytettyjen koneiden paikoitukseen sekä ohjelmoimiseen liittyviä tietoja. Työn jatkamisen kannalta olisi luontevaa tutkia työssä esitettyjä toteutustapoja tarkemmin kohdeyrityksen solussa. Jatkotöissä voitaisiin tutkia ohjelmien generoimista

eri paikkatiedoista, esimerkiksi laserskannerin tai 3D-mallin tiedoista sekä vertailla tarkemmin toteutusmenetelmien hyötyjä ja heikkouksia. Ongelmaa voitaisiin myös lähteä ratkaisemaan eri näkökulmasta. Tässä työssä keskityttiin etsimään ratkaisuja ongelmaan muuttamalla yhden robotin toimintoja ohjelmallisesti, mutta on todennäköistä, että ongelmaan löytyisi ratkaisu joko tuotantoprosessiin tehtävistä muutoksista tai käyttämällä eri robotteja taikka oheislaitteita, esimerkiksi yhteistyörobotteja tai kamerajärjestelmiä. Kaiken kaikkiaan työtä voidaan pitää onnistuneena, koska kaikki sille asetetut tavoitteet saavutettiin.

LÄHTEET

- [1] J. Alander & K. Niemi, *Robottien ohjaus ja ohjelmointi*. Otakustantamo, 1987.
- [2] A. Allahverdi & H. M. Soroush, “The significance of reducing setup times/setup costs,” *European Journal of Operational Research*, vol. 187, no. 3, pp. 978–984, 2008.
- [3] M. H. Ang, L. Wei, & L. S. Yong, “An industrial application of control of dynamic behavior of robots-a walk-through programmed welding robot,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, 2000, pp. 2352–2357.
- [4] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB® Second, Completely Revised, Extended and Updated Edition*. Springer International Publishing AG, 2017, vol. 118.
- [5] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3. p. Pearson, 2004.
- [6] A. J. Critchlow, *Introduction to robotics*. Macmillan, 1985.
- [7] M. Daniele, C. Massimo, & C. Cristalli, “Manual guidance for industrial robot programming,” *Industrial Robot: An International Journal*, vol. 42, no. 5, pp. 457–465, 2015.
- [8] *Fennosteelin tarina*, Fennosteel Oy., (viitattu 16.5.2020). Saatavissa: <https://fennosteel.com/>
- [9] *Automation systems and integration – Numerical control of machines – Program format and definitions of address words – Part 1: Data format for positioning, line motion and contouring control systems*. ISO 6983, 2009.
- [10] *Robots and robotic devices – Coordinate systems and motion nomenclatures*. ISO 9787, 2013.
- [11] D. Kandray, *Programmable automation technologies: an introduction to CNC, robotics and PLCs*. Industrial Press, 2010.
- [12] H. B. Kief & H. A. Roschiwal, *CNC handbook*. McGraw-Hill, 2013, (viitattu 5.3.2020). Saatavissa: <https://www-accessengineeringlibrary-com.libproxy.tut.fi/browse/cnc-handbook#pt01>

- [13] I. Lapinleimu, V. Kauppinen, & S. Torvinen, *Kone- ja metalliteollisuuden tuotantojärjestelmät*. WSOY, 1997.
- [14] H. Lehtinen & R. Kuivalainen (toim.), *Robotiikka*. Talentum, 1999.
- [15] Y. Matsusaka, K. Oku, & T. Kobayashi, “Design and implementation of data sharing architecture for multifunctional robot development,” *Systems and Computers in Japan*, vol. 35, no. 8, pp. 54–65, 2004.
- [16] Z. Pan, J. Polden, N. Larkin, S. V. Duin, & J. Norrish, “Recent progress on programming methods for industrial robots,” *Robotics and Computer-Integrated Manufacturing*, pp. 87–94, 2012.
- [17] A. Peltonen, *Tuottava tehdas*. Opetushallitus, 1997, (viitattu 23.5.2020). Saatavissa: <http://www03.edu.fi/oppimateriaalit/tuottavatehdas/index.html>
- [18] M. Ragaglia, A. M. Zanchettin, L. Bascetta, & P. Rocco, “Accurate sensorless lead-through programming for lightweight robots in structured environments,” *Robotics and Computer Integrated Manufacturing*, pp. 9–21, 2016.
- [19] J. Trevino, B. J. Hurley, & W. Friedrich, “A mathematical model for the economic justification of setup time reduction,” *International Journal of Production Research*, vol. 31, no. 1, pp. 191–202, 1993.
- [20] H. Vesämäki, *Lastuavan työstön NC-ohjelmointi*. Teknologiainfo Teknova, 2007.
- [21] K. B. Zandin, *Maynard’s industrial engineering handbook*, 5. p. McGraw-Hill, 2001, (viitattu 5.3.2020). Saatavissa: <https://www.accessengineeringlibrary.com/browse/maynards-industrial-engineering-handbook-fifth-edition>