

SATAKUNNAN AMMATTIKORKEAKOULU

Hannu-Pekka Karilahti

ANNOSTELIJAROBOTIN OHJELMOINTI PAIKKAMUUTTUJIIN
KONENÄÖN AVULLA

KONE- JA TUOTANTOTEKNIikka

Koneautomaatio

2013

ANNOSTELIJAROBOTIN OHJELMOINTI PAIKKAMUUTTUIJIN KONENÄÖN AVULLA

Hannu-Pekka Karilahti

SATAKUNNAN AMMATTIKORKEAKOULU

Tekniikka ja merenkulku Pori

Tekniikantie 2

28600 PORI

Kone- ja tuotantotekniikka

Toimeksiantaja: Vertic Zinc Wire Oy

Ohjaaja: Soini Antti

29.5.2013 PORI

Avainsanat: Robotti, anturi, etäisyydenmittaus

TIIVISTELMÄ

Päättötyökseni valittiin Vertic Zinc Wire Oy:n tarjoama annostelijarobotin ohjelmoimistyö. Aikaa vievä ja hankala pakkaustyövaihe haluttiin pois, ja työtä tekemään oli hankittu robotti. Robotin ohjelmoimisessa tuli huomioida käytön helpous, toistettavuus sekä varma toimivuus. Päättötyön oli tarkoitus toimia jatkossa myös robotin käyttöohjekirjana yrityksen työntekijöille. Tästä syystä siitä piti löytää laajat tiedot työn vaiheista, jotta mahdolliset ongelmat ja lisäykset ohjelmaan olisi mahdollista tehdä ilman apua. Ohjelmointi tehtiin suoraan robotin ohjelmointipaneelilla. Työn alkuvaihe sisälsi koordinaatistojen hallintaa, robotin varsien liikkeiden opettelu, sekä peruskäskyjen hallintaa. Alkuvaiheessa huomattiin myös lankojen laskumäärän olevan vaihteleva. Lankapinon korkeuden mittaamiseen hankittiin ultraäänianturit, jotka tekivät robotin toiminnasta varmempaa. Anturit toimivat työn loppuvaiheessa myös pakkauspakettien asemoinnin apuna. Asemointi oli tärkeää, koska lankapituudet vaihtelivat, ja sen myötä laatikkokoot. Paikkamuuttujien ohjelmointi ja käyttö helpotti työtä paljon, koska näitä muuttujia pystyi kopioimaan ja ”peilata” muihin ohjelmiin. Paikkamuuttujat mahdollistivat myös ohjelman hienosäädön, jos langat eivät heti osuneet kohteisiinsa.

Robotti on ollut mukana tehtaassa tuotannossa, ja pienten säätöjen jälkeen robotin työ on melko sujuvaa ja ongelmatonta. Pakkaustyö on jäänyt lähes kokonaan pois, ja pakkauksen viimeistely on ainut työvaihe valusta tulon jälkeen.

ABSTRACT

The aim of this Bachelor's thesis was to program a dosing robot for Vertic Zinc Wire. The company wanted to leave out the time-consuming and troublesome packing phase and a robot was purchased to do the work instead. Ease of use, repeatability and performance reliability had to be taken into consideration when programming the robot.

The Bachelor's thesis was also intended as an instruction manual for the users of the robot. Therefore it had to include extensive information of all the stages of the process/ work for the users to easily find out the problems and make additions to the program without help. The programming was made with a programming-panel. At the beginning of the work it was noticed that the amount of wires in boxes was variable. An ultrasound sensor was needed to measure the distance to the sticks, which made the landing more reliable. At the end of the work the sensors were also used in positioning the packing-boxes.

At the factory, the robot has already been in production and after some adjustments the functioning of the robot is fluent and trouble-free. The packing work has nearly been omitted, and the finishing of the box is the only work that the employees have to do after casting.

ALKUSANAT

Työ oli odotetun haastava, ja monet oikeat työtavat löytyivät vasta useiden kokeilujen ja käytännön testauksien kautta. Työtä tehdessä suurena apuna toimi Vertic Zinc Wire:n pääinsinööri Mika Lintusalo. Ilman häntä työ olisi ollut paljon vaikeampi tehdä, joten suuret kiitokset hänelle. Apuna työssä toimi myös Motomanin henkilökunta, joka puhelimitse antoi hyviä vinkkejä työn toteuttamiseen.

Robotti on siirtynyt tuotantoon, ja toivonkin, että robotti olisi suurena apuna kesän vilkkaimpina hetkinä, ja robotti todettaisiin tärkeäksi lenkiksi tehtaan kokonaisu-tuotannossa. Työ oli opiskeluni kannalta antoisa, ja toivon jatkossa kaikkea hyvää koko Vertic Oy:n henkilökunnalle.

Sisältö

TIIVISTELMÄ	2
ABSTRACT	3
ALKUSANAT	4
1 JOHDANTO	6
2 ANNOSTELUTEHTÄVÄT	7
3 LAITTEISTO	8
4 KOORDINAATISTOT	15
5 ROBOTIN OHJELMOINTI	19
6 YHTEENVETO	31
LÄHTEET	32
LIITTEET	32

1 JOHDANTO

Porilainen perheyritys nimeltä Vertic Zinc Wire tarjosi minulle tätä työtehtävää. Yrityksen perusti Matti Lohikoski vuonna 1985, hän on myös toiminut yrityksen toimitusjohtajana siitä lähtien. Yrityksellä on 2 toimipistettä, joista toinen sijaitsee Porissa Ulasoorin teollisuusalueella ja toinen Virossa. Yrityksessä työskentelee 8 työntekijää, kiireisinä kesäpäivinä yrityksessä on myös n. 4 kesätyöntekijää. Yrityksellä on yksi valulinja, jossa pystykoneella sulasta tehdään lankaa. Jatkomuokaus tehdään vetokoneilla, joissa lanka muutetaan pienempiin halkaisijoihin. Yrityksen pääinsinööriä toimii Mika Lintusalo, joka oli mukana päättötyöni eri vaiheissa.

Vertic Zinc Wire:n tuotteisiin kuuluvat mm. sinkkiruiskutuksessa käytettävät sinkkilangat ja tyhjiössä höyrystykseen käytettävät sinkkitangot. Yrityksen käytämä tekniikka on jatkuva pystyvalu, jonka on kehittänyt yhtiön hallituksen puheenjohtaja Timo Lohikoski. Yhtiön lähtömateriaalina on Outokummun puhdas Special High Grade (Zn>99,995 %) sinkkiharkko. Tuotteet sisältävät siis erittäin vähän haitallisia ainesosia kuten lyijyä (Pb) ja kadmiumia (Cd). Nykyaikaisesta tuotantotavasta ja tuotekäsittelystä johtuen myös langan pinta on puhdasta ja sileää.

Sinkkilankoja on saatavilla 1,4 - 4,76 mm paksuisia ja erilaisissa pakkauksissa liekki- ja kaariruiskutukseen. Lankoja valmistetaan eri kovuuksilla ja lujuuksilla. Sinkkitankoja valmistetaan 8 – 18 millisinä.

Sinkkitanko valetaan suoraan lopulliseen paksuuteensa, joten niitä ei käsitellä mitenkään. Tämän erityisen *VERTIC* -tuotantoprosessin ansiosta tankojen pinta on ehdottomasti täysin puhdas ilman mitään käsittelyainejäämiä. (KUVA 1.)



KUVA 1. Sinkkitangon murtopinta /1 /

1.1 Tavoitteet

Tavoitteena oli ohjelmoida yrityksen annostelijarobotti pakkaamaan kaikki sinkkilankakoot suoraan laatikkoon. Ohjelman yksinkertaisuus, luotettavuus ja toistettavuus olivat tärkeitä asioita ohjelmaa tehtäessä.

2 ANNOSTELUTEHTÄVÄT

Aikaisemmin valusta tuleva sinkkilanka pätkittiin oikean kokoisiksi leikkurilla, josta pätkät putosivat isoon laatikkoon. Isosta laatikosta työntekijä siirsi pätkät pienempiin laatikoihin. Pätkien pakkaus laatikoihin oli työlästä ja aikaa vievää. Työ sitoi yhden työntekijän pitkäksi aikaa, siksi tämä tehtävä haluttiin automatisoida. Työvaihe oli automatisoitavissa, koska robotin tekemät liikkeet olisivat samantyyppisiä ja melko yksinkertaisia.

Ensimmäisenä tehtävänä oli ohjelmoida robotille 300 mm pitkien sinkkitankopätkien annosteluohjelma. Myöhemmässä vaiheessa mukaan tuli myös 230 mm, 400 mm, ja 500 mm pätkien annostelun ohjelmointi. Ohjelmarunko oli hyvin samantapainen kaikissa ohjelmissa, joten ohjelmien teko oli ensimmäisen ohjelman jälkeen melko helppoa. Automatisointityöhön oli hankittu etukäteen Motoman UP-20 robotti, joka on ankkuroitu lattiaan. Robotin tehdasasetukset oli asennettu robottiin valmiiksi. Langan pätkimiseen ja ohjaukseen valusta robotille oli tehty siihen soveltuva katkaisin/annostelija yhdistelmälaite. Valusta tuleva sinkkilanka johdetaan ohjaimien avulla katkaisimeen, jossa katkaisun hoitaa servomootorilla varustettu leikkuri. Katkaisinta ohjataan logiikan avulla. Katkaisun jälkeen oikean pituinen sinkkipätkä liukuu kourua pitkin annostelijalle. Tästä pätkä siirretään varsinaiselle robotin kiinteälle hakupisteelle. Pätkien siirto toistetaan riittävän monta kertaa. Logiikan antaman noutoluvan jälkeen robotti vie sinkkipätkät siististi valmiiksi ohjelmoituihin laskupisteisiin.

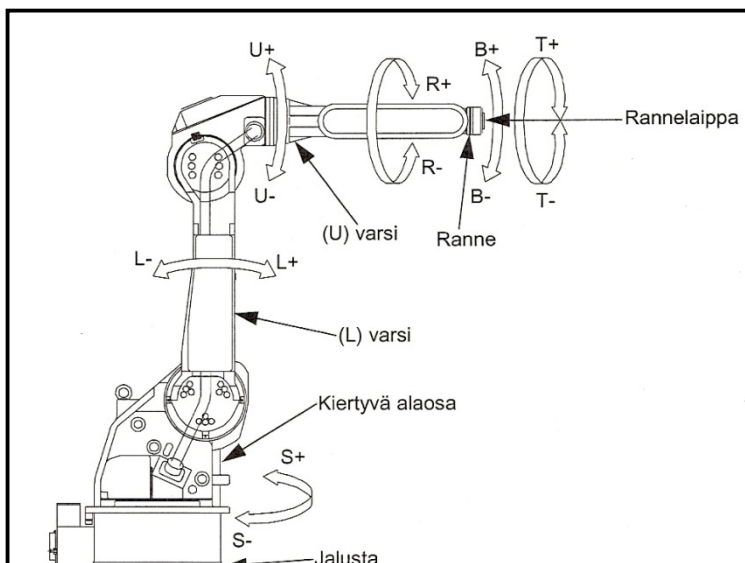
Pakkausohjelmasta oli tarkoitus saada mahdollisimman toimintavarma ja helppokäyttöinen. Ainoat työt, joita työntekijällä tulisi olla, olivat pakkauslaatikoiden vaihto sekä robotin halutun ohjelman käynnistys. Työ vaikutti haastavalta, koska koulussa olin opiskellut robotin ohjelmointia vain yhdellä kurssilla, eikä yrityksen

henkilökunnalla ollut juurikaan aiempaa kokemusta robottien ohjelmoinnista. Työ oli kuitenkin myös mielenkiintoinen, joten päätin ottaa sen vastaan.

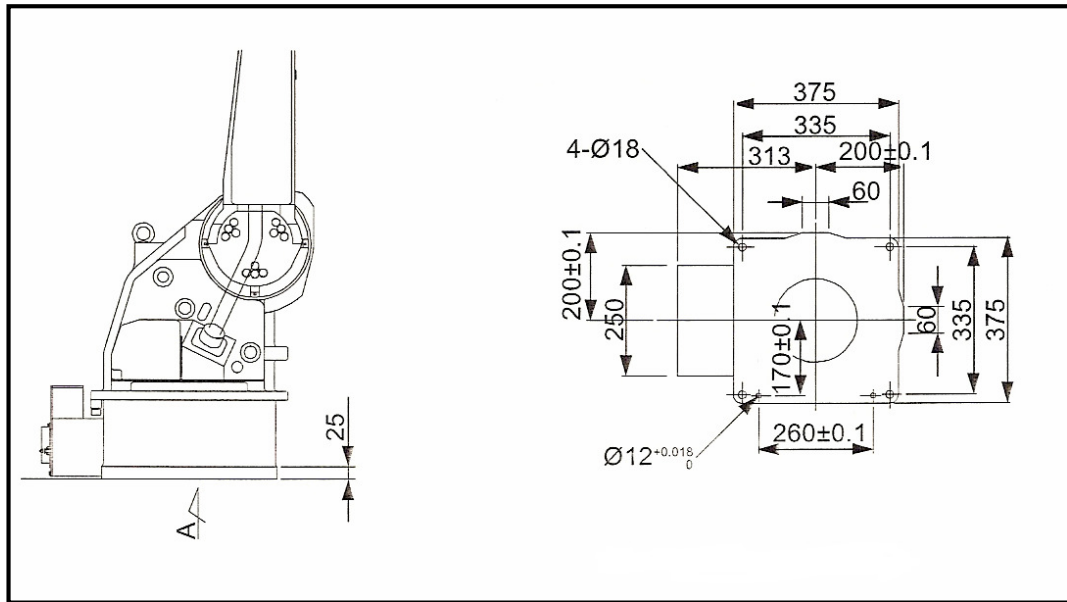
3 LAITTEISTO

Robottijärjestelmiä käytetään moniin eri tehtäviin riippuen sovelluksesta. Sovelluksen tunnistaa oheislaitteista ja robottityökaluista. Sovelluksista riippumatta robottiin voidaan liittää yksi tai useampia kappaleenkäsittelylaitteita.

Käyttämäni robottilaitteisto on nimeltään Motoman UP-20 (LIITE 1.). Sen on valmistanut Motoman Robotics Finland Oy. Robottilaitteisto koostuu 6-nivelisestä robottikäsiarvasta ja robottiohjaimesta. Robotilla on oma ohjelmisto robotin ohjelmointiin, eri tilojen seurantaan, testiajoon sekä ohjelmien lataukseen. Robotti on suunniteltu erityisesti kevyisiin hitsaus-, pakkaus- ja käsittelytehtäviin. Se pystyy nimensä mukaan käsittelemään maksimissaan 20 kg:n kuormaa viimeisen nivelen päässä. Robotin toistotarkkuus on ± 0.08 mm. Robotin mukana toimitettujen käyttö- ja turvallisuusohjeiden avulla selvitin peruskäskyt ja ohjelmointitekniikat. Lisäksi ohjeista löytyy ohjeet lähtö- ja tulosignaalien kytkennöistä. Robotti oli ankkuroitu lattiaan ja kalibroitu, joten robotti oli valmis ohjelmointia varten. Alla esitetty robotin osien nimet ja jalustan mitat (KUVAT 2. ja 3.)



KUVA 2. Robotin osien nimet ja akselit /3, s. 20 /



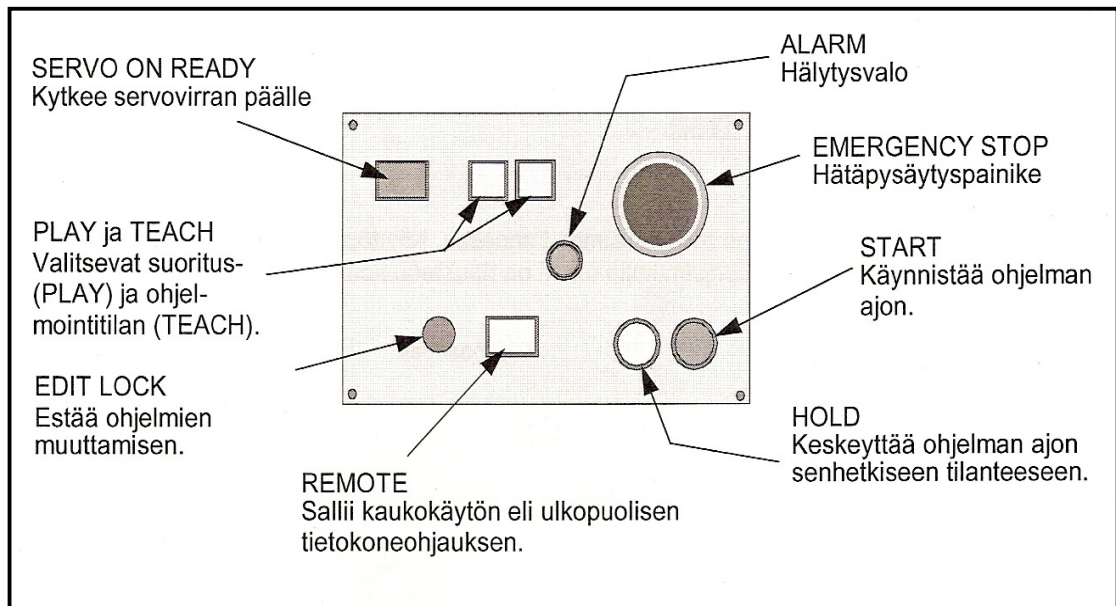
KUVA 3. Jalustan mitat /3 s, 20/

3.1 Robottiohjaimet

Robottiohjain on nimeltään Yasnac XRC. Robottiohjaimiin kuuluu käyttöpaneeli sekä ohjelmointipaneeli. Näillä paneeleilla hoidetaan robotin kaikki tehtävät, kuten käynnistykset, ohjelmointi ja seuranta.

3.1.1 Käyttöpaneeli

Käyttöpaneelista (LIITE 2) löytyy painikkeet, joilla robotin ohjelmat ajetaan. Käyttöpaneelista löytyy lisäksi päävirtakytkin joka käynnistää ja sammuttaa XRC:n. Tarvittava robotin nollaus tapahtuu päävirtakytkimen katkaisulla. Tätä tarvitaan esimerkiksi vakavan törmäyksen jälkeen. Käyttöpaneelin painikkeet on kuvattu tarkemmin seuraavalla sivulla (KUVA 4).



KUVA 4. Käyttöpaneelin painikkeet /2, s. 12/

Käyttöpaneeliin tehtiin myöhemmässä vaiheessa kaksi lisäpainiketta. Toisella painikkeella robotti pystytään kesken ohjelman nollaamaan mahdollisten ongelmien jälkeen. Toinen painike siirtää robotin täyttöohjelman suoraan toiseen laatikkoon. Tätä toimintoa käytetään, kun vain toinen laatikko halutaan täyttää. Seuraavassa näppäinten tarkempi esittely:

”PLAY” - näppäin kytkee robotin valitun suoritus-tilan, ja ”TEACH” - näppäin taas ohjelmointitilan. Ohjelmointitila on ainoa tila, jolloin robottia voidaan käsitellä sekä muokata robotin eri ohjelmia. ”SERVO ON READY” -näppäin kytkee servomootoreille virran mahdollisen hätäpysäytyksen tai liikeratojen ylittymisen jälkeen. ”REMOTE” valitsee suoritetaanko käynnistys- ja tilanvaihtotoiminnot käyttöpaneelilta vai ulkoiselta ohjaimelta. Tämä painike toimi myöhemmässä vaiheessa myös laatikon vaihto painikkeena, eli jos REMOTE - painike startin jälkeen painettiin päälle, alkoi ohjelma täyttää välittömästi 2. laatikkoa. ”EMERGENCY STOP” on ns. hätäseis-paineke, joka sammuttaa ohjaimen virran. START on ohjelman käynnistysnappula ja ”HOLD” vastaavasti ohjelman keskeytysnappula. Näppäinten lisäksi käyttöpaneelista löytyy ”ALARM” - valo, joka varoittaa mahdollisesta hälytyksestä tai suoritusvirheestä. Ohjelmointipaneelin ”RESET” - painike sammuttaa valon ja nollaa mahdolliset virheilmoitukset.. Käyttöpaneelin takaa löytyvät paikat kaikille sähkö- sekä tulo/lähtöjohtoille. Lisäksi oven takana on muistikorttipaikka, jonka avulla ohjelmat voidaan tallentaa

ulkoisille laitteistoille ja ottaa ohjelmista mm. varmuuskopiot. Käyttöpaneeli on tärkeä työväline työntekijöille, jotka robottia käyttävät. Robotin ohjelmointi oli tarkoitus viedä niin pitkälle, että lähes kaikki tarpeelliset valinnat tehdään suoraan käyttöpaneelistä, eikä itse robotin ohjelmointipaneeliin tarvitse ohjelmoinnin jälkeen juurikaan koskea.

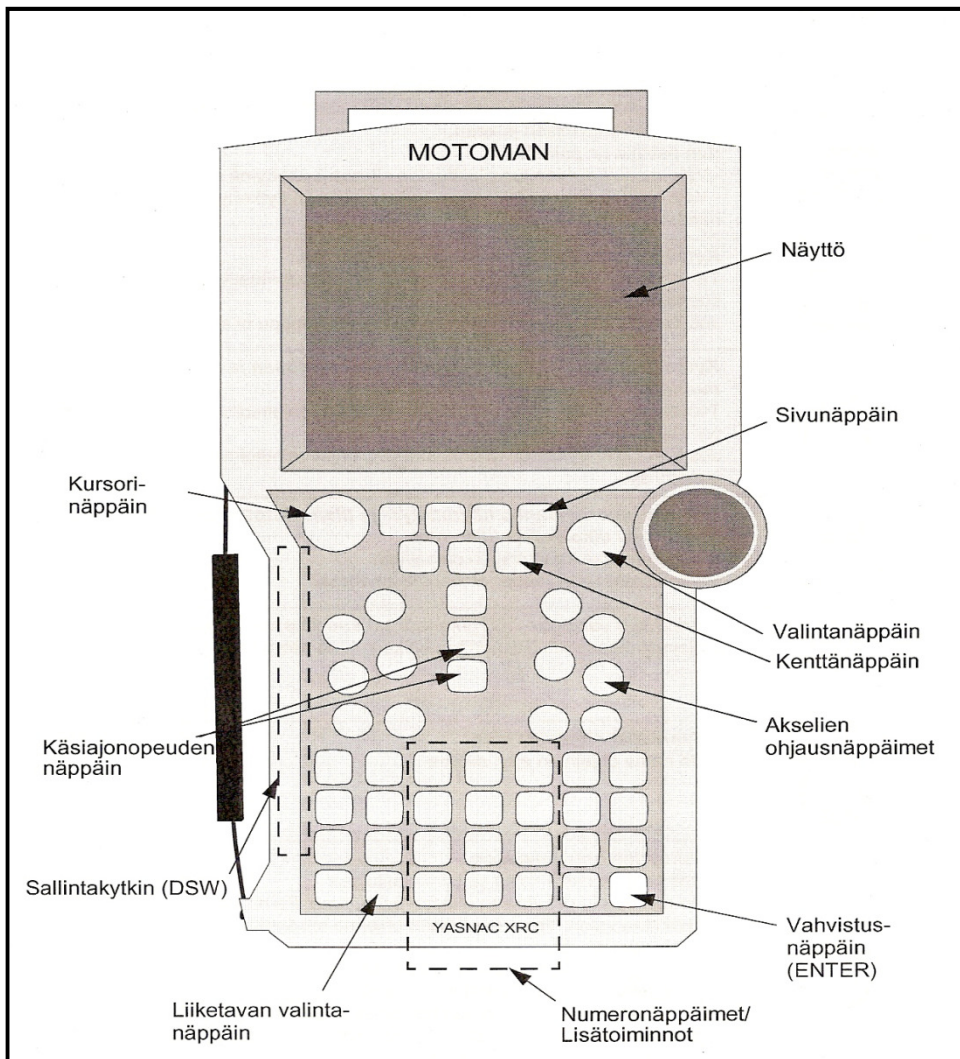
3.1.2 Ohjelmointipaneeli

Robottiohjaimiin kuuluu myös ohjelmointipaneeli. Normaalissa käyttötilanteessa ohjelmointipaneeli on ripustettuna ohjausyksikön etuseinään, jossa se roikkuu ripustuskoukussa. Ohjelmointipaneeli on robotin ohjelmoinnin kannalta tärkein työväline. Ohjelmointipaneelin näppäimillä ja painikkeilla luodaan ja muutetaan robotin ohjelmia. Lisäksi sillä voi seurata selkeästi nestekidenäytöstä ohjelmien kulkua sekä tulo/lähtösingaalien kulkua robotin ja logiikan välillä. Paneeli on varustettu turvallisuusyistä sallintakytkimellä (ns. kuolleen miehen kahva), jonka tehtävänä on tunnistaa käyttäjän olemassaolo. Jotta robotti toimisi opetustilanteessa, on kahvaa puristettava jatkuvasti keskiasennossa. Kahvassa on 3 asentoa. Kun kytkin on täysin vapautettuna, niin robotti pysähtyy. Kevyesti puristettuna robotin servomoottori käynnistyy normaalisti ja lujasti puristettuna robotti taas pysähtyy. Ohjelmointipaneelista tarkempi kuva (LIITE 3).

Suurella nestekidenäytöllä on 3 eri kenttää. Otsikkokenttä näyttää käytössä olevan ohjelman ja tilanätkkenttä näyttää ohjaimen valitut asetukset kuten valitun koordinaatiston, käsiajonopeuden, käytetyn ohjelmisivun sekä järjestelmän nykytilan. Pääkentässä näkyy ohjelmavalikko tai ohjelmoinnin aikana ohjelmoidut ohjelmarivit. Tähän kenttään syötetään ohjelma. Kaikki luodut ohjelmat ladataan robotin käyttöön ohjelmasisältökentästä. Sieltä löytyvät myös kaikki ns. aliohjelmat, jotka voidaan myöhemmin avata muokkausta varten.

Robotin liikkeiden ohjelmoinnissa tärkeimmät näppäimet ovat akselien ohjausnäppäimet, ”MOTION TYPE” -, ”ENTER” - ja kursorinäppäin. Paneelin kursorinäppäimellä voidaan ohjelmassa siirtyä eri riveille sekä siirtyä eri ohjelmiin. Enter-näppäin tallentaa käskyn, tiedon tai robotin nykyisen aseman jne. Akselin ohjausnäppäimillä ohjataan robotin haluttuja akseleita ja COORD -näppäimellä valitaan haluttu koordinaatisto, jossa robotti liikkuu. MOTION TYPE -

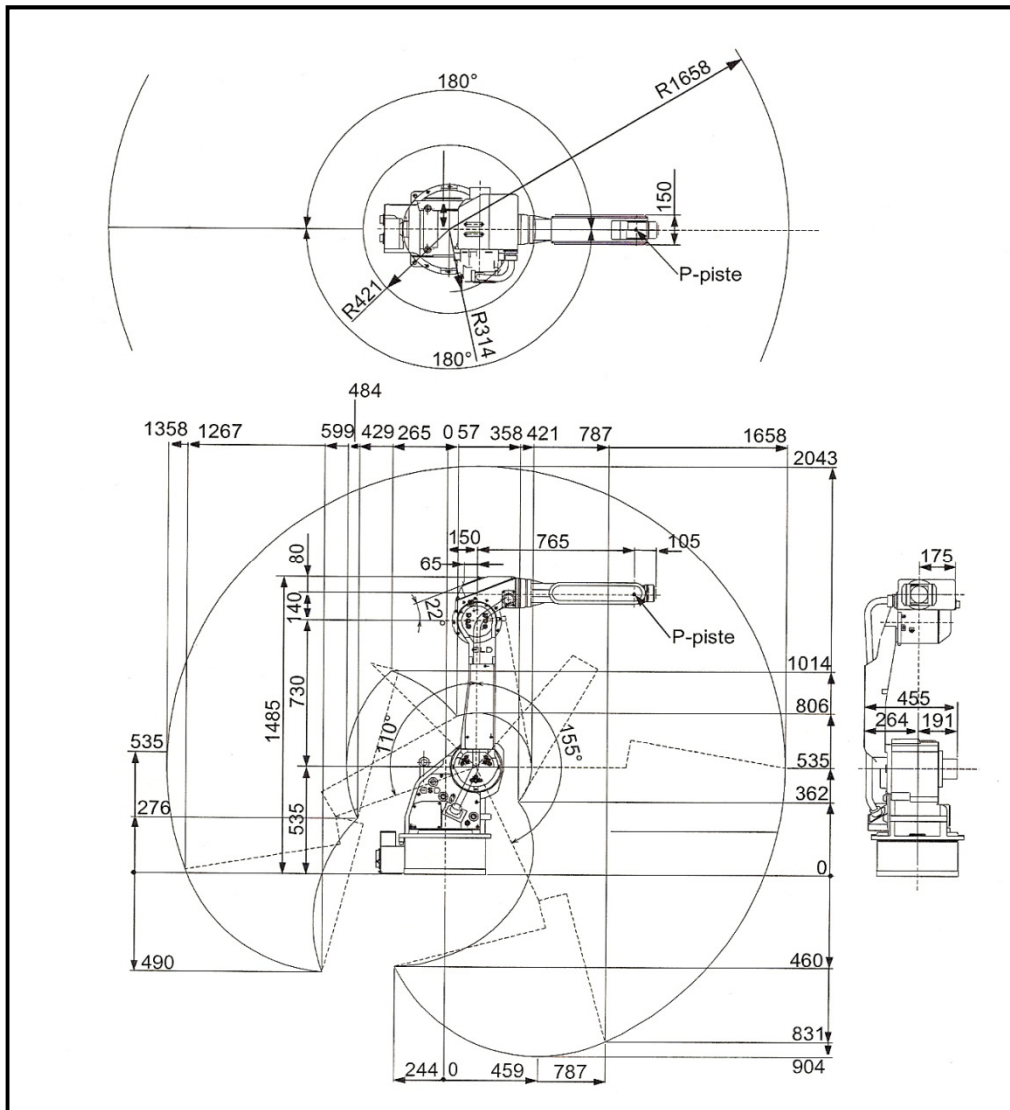
näppäimellä valitaan haluttu liiketyyppi, joita ovat mm. nivel- ja suoraliike. Alla on esitetty ohjelmointipaneelin painikkeet (KUVA 5).



KUVA 5. Ohjelmointipaneelin painikkeet /2, s. 14/

3.2 Robottikäsivarsi

Robotti on kooltaan melko pieni, ja soveltuu siksi erittäin hyvin pieniin annostelu- ja hitsaustöihin. Pienehkö koko auttoi robotin sijoituksessa, koska se ei vaatinut suurta työaluetta pienehkössä hallissa. Toisaalta laatikoita ei voinut tehdä kovin isoiksi, koska robotti ei olisi yltänyt ääripäihin. Alla on esitetty käsivarren mittoja ja maksimi ulottuvuudet (KUVA 6).



KUVA 6. Robottikäsivarren mitat ja ulottuvuudet /3, s. 21/

3.3 Robotin koura

Koura oli suunniteltava niin, että pätkät pysyisivät tukevasti liikkeen aikana kourassa ja liukuisivat siististi alas kouran auetessa. Lisäksi kouraan asennettavien antureiden tulisi olla hyvin suojassa mahdollisten törmäyksien varalta. Kun kouran optimaalinen muoto saatiin suunniteltua ja tehtyä, koura asennettiin robotin viimeisen varren päähän oikeaan asentoon. Kouran avaus on paineilmaohjattu, joten kouraan asennettiin ohjausventtiili ja paineilmaletkut. Paineilman ohjauksessa toimi normaali 2-suuntaventtiili. Sylinteri on logiikkaohjattu, joten sitä varten

asennettiin tarvittava sähköjohdotus tulo- ja lähtötietoja varten. Robotille johdotusta varten varren päälle rakennettiin kytkentärasia. Tarvittavia lähtösignaaleja ovat kouran tila, laatikon vaihtokäsky sekä laskuoperaation seuranta. Tulosignaaleja robotille ovat hakukäsky, virhekäsky sekä turva-aidan tieto. Kun johdotus oli tehty, kouran toimintoja testattiin robotin ohjaimella ja lähtö- ja tulotietojen osoitteet kirjattiin ylös. Paineilmasyylinteri, mikä tekee kouran kiinni/auki liikkeen, asennettiin robotin 1. lähtöön. Ohjelmointipaneelilla on helppo seurata tulotietojen kytkentöjä sekä asettaa lähtöpaikkoja päälle sekä pois. Koura toimi hyvin ja sen suljetusta kourasta tuli myös erittäin tiivis. Tämä on välttämätöntä ohuiden pätkien kuljetuksessa. Robotin kouran suunnittelusta, asennuksesta ja johdotuksesta vastasi Mika Lintusalo. Kourasta tekninen piirustus (LIITE 4).

3.4 Katkoja

Langan katkontaa varten oli valmiiksi tehty siihen soveltuva servomoottorilla varustettu katkoja ja valmiiden pätkien annostelulaite. Katkojassa on servomoottorin lisäksi 2 logiikkaohjattua paineilmasyylinteriä. Valusta tuleva pätkä ohjataan suoraan logiikkaohjattuun katkaisimeen. Logiikka ohjaa langan syöttönopeutta, jolla saadaan siten haluttu pätkänpituus. Katkaistu lanka putoaa ränniin, missä kuljettimena toimii moottorilla pyörivä ketju. Ketjun tarkoituksena on antaa pätkälle lisävauhtia, jotta pätkä kulkisi hyvin annostelijalle asti. Työn alussa ketjua ei ollut, ja se aiheutti syöttöongelmia annostelijalle. Ilman ketjua pätkä jumittui ajoittain kourulle ilman kourun huolellista puhdistusta. Tämä puhdistustyö haluttiin jättää pois. Langan valuttua kourun päähän paineilmasyylinteri teki kippausliikkeen ja lanka putosi varsinaiselle hakupisteelle. Annostelijan kippausyylinteri on myös logiikkaohjattu, minkä avulla voidaan määrätä mm. kerralla vietävien pätkien määrä. Pätkien kerralla vietävä määrä vaihteli melkoisesti pätkän eri paksuuksista johtuen. Varsinaisessa hakupaikassa on lisäksi toinen paineilmasyylinteri, mikä työntää pätkät laitaa vasten. Näin varmistetaan, että valmis sinkkipätkänippu on juuri oikeassa paikassa ja päät ovat tasan. Logiikka antaa pätkien työntämisen jälkeen robotille hakukäskyn, jolloin robotti käynnistää nouto-ohjelman. Kouran muodon takia hakupisteeseen piti tehdä raot, jotta langat jäisivät hyvin kouran sisälle. Jotta kaikki lankakoot pysyisivät tasapainossa kouran liikkeen aikana, piti hakupisteen kourusta tehdä liikuteltava. Pätkien mukaan kourun paikka valitaan niin, että koura tarttuu aivan langan keskeltä. Näin laskusta tulee tarkempi ja siir-

tonopeuksia pystytään kasvattamaan ilman ongelmia. Katkojasta ja noutopisteessä löytyy kuvia (LIITE 5).

3.5 Logiikka

Logiikan ohjauksessa toimii Siemensin ohjelmoitava Simatic S5-90U. Sama ohjaus on ollut jo ennestään käytössä valulinjaprosessissa, joten logiikan käyttö oli tuttua. S5-95U ja Motoman keskustelevat keskenään digitaalitulojen/-lähtöjen välityksellä. Näiden kahden ohjausjärjestelmän jännitetasot on erotettu toisistaan I/O-signaalien väliin kytketyillä välireleillä /4/. Simaticin ohjauksiin kuuluu katkojan ohjaus, sekä langan syöttönopeuden ohjaus. Muuttuva parametri langan nopeuden lisäksi on kippausmäärä hakupisteelle. Lankojen tasaukseen hakupisteessä käytettävä työntösylinteri on logiikkaohjattu. Syöttönopeus asetetaan logiikkaan lankakoon mukaan; mitä paksumpi lanka on, sitä hitaampi on valunopeus. Nopeuden ohjauksessa ovat apuna lisäksi langan yläpuolella ja alapuolella olevat sähköllä varatut metallitangot, jotka toimivat lankanopeuden raja-arvoina. Jos nopeus on liian hidas, lanka osuu alapuolella olevaan tankoon, ja syntyvä vauraus ilmoittaa logiikalle nopeuden lisäyksen tarpeen. Yläpuoli toimii vastakkaisella periaatteella.

4 KOORDINAATISTOT

Robotilla on olemassa valmiita koordinaatistoja, joilla se liikkuu tietyllä tavalla. Tällaisia ovat mm. nivel-, ja suorakulmainen koordinaatisto. Nivelkoordinaatistossa robotin kaikki akselit liikkuvat toisistaan riippumattomasti, kun taas suorakulmaisessa koordinaatistossa robotti liikkuu asennosta huolimatta x-, y- tai z-akselin suuntaisesti. Annostelutyössä tarvitaan vain kolmea suuntaa, joten liikkeiden opettelu aloitettiin tällä koordinaatistolla. Robotille on lisäksi mahdollista opettaa omia koordinaatistoja, joissa robotti liikkuu käyttäjän määrittämien koordinaattien mukaisesti.

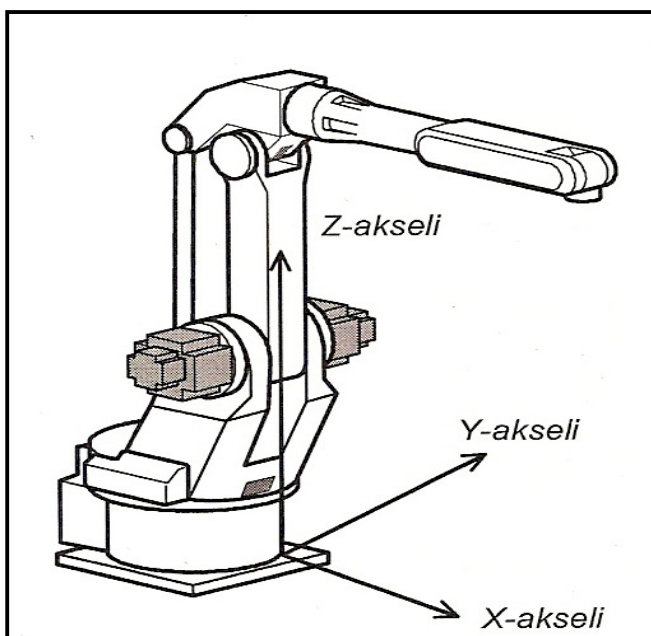
4.1 Nivelkoordinaatisto

Tässä koordinaatistossa kaikki nivelet liikkuvat toisistaan riippumattomasti. Työnsäni tätä liikettä käytetään yleisesti niiden pisteiden välillä, jossa robotti voi itse

valita nopeimman reitin. Robotin eri nivelten liikkeitä oli hyvä harjoitella tässä koordinaatistossa, koska kaikki nivelet toimivat omina yksikköinä.

4.2 Suorakulmainen koordinaatisto

Suorakulmainen koordinaatisto auttoi ohjelmoinnin alussa liikeratojen opetteluksessa. Liikkeet ovat yksinkertaisia ja suoria, joten törmäysvaara jäi pieneksi. Työn alussa ajattelin käyttää kyseistä koordinaatistoa, koska laatikot ovat suorakulmaisia, ja liikkeitä pystyisi käyttämään hyvin hyödyksi. Myöhemmin huomasin käyttäjän laatimien koordinaatistojen tehokkuuden, ja suorakulmainen koordinaatisto jäi kokonaan pois. Alla on esitetty robotin liikkeitä kyseisessä koordinaatistossa (KUVA 7).

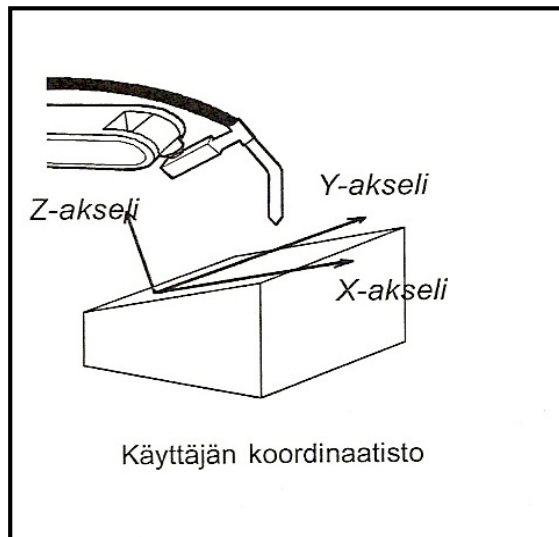


KUVA 7. Robotin liikkeet suorakulmaisessa koordinaatistossa /2, s. 39/

4.3 Käyttäjäkoordinaatisto

Käyttäjän koordinaatisto helpottaa ohjelmointia monissa tilanteissa varsinkin töissä joissa käytetään vakio paikoissa erilaisia pakkauslaatikon kokoja tai paikoituspöytiä. Toistettavissa töissä virheiden määrä pienenee ja ohjelmoinnin aikana tapahtuva käsikäyttö nopeutuu käyttäjäkoordinaattien takia. Koska käyttäjä tietää koordinaatistojen suunnat, on robottia huomattavasti helpompi ohjailta käsikäytössä. Käyttäjakohtaisessa koordinaatistossa robotti liikkuu käyttäjän määrittämi-

en koordinaattien suuntaisesti. Koordinaatteja voidaan robotille ohjelmoida 24 kappaletta. Koordinaatit ohjelmoidaan numeroituihin käyttäjäkoordinaattitiedostoihin. Ohjelmoin aluksi koordinaatiston laatikoiden pohjalle, mutta myöhemmin kouraan asennetut ultraäänianturit kasvattivat törmäysriskiä niin paljon, että lopullinen koordinaatiston asennuspaikka oli parasta ohjelmoida laatikon yläreunan tasolle. Myöhemmässä vaiheessa joka ohjelman alussa robotti teki automaattisen koordinaatiston asetuksen ultraäänianturin avulla. Tämä vähensi laatikon paikannuksessa tapahtuvia riskejä melkoisesti. Anturin toiminnasta koordinaatiston asennuksessa on kerrottu lisää kappaleessa 5.6. Alla löytyy kuva robotin liikkeistä käyttäjän koordinaatistossa (KUVA 8).



KUVA 8. Robotin liike käyttäjän koordinaatistossa /2, s. 46/

Ohjelmoituja koordinaatioita kaikista töistä löytyy yhteensä 9 kappaletta. Alla on lueteltu kaikki käytetyt koordinaatit eli toisin sanoen käytetyt laatikkokoot ja pakkausjärjestysnumero (TAULUKKO 2):

Koordinaatiston no.	Käytettävä laatikko
#1.	300mm langan 1. laatikko
#2.	300mm langan 2. laatikko
#3.	400mm langan 1. laatikko
#4.	400mm langan 2. laatikko
#5.	287mm langan 1 laatikko
#6.	287mm langan 2. laatikko
#7.	287mm langan 3. laatikko
#8.	500mm langan 1. laatikko
#9.	500mm langan 2. laatikko

TAULUKKO 2. Työssä käytettävät koordinaatit

4.4 Koordinaatiston käyttö

Jokaisessa lankakoossa on siis vähintään 2 laatikkoa, ja 287 mm:llä langalla jopa 3. Jokaisen laatikon koordinaatisto on ohjelmoitava erikseen ja erilaisia laatikkokokoja on käytössä yhteensä 5 kappaletta. Opetettu koordinaatisto ja siihen ohjelmoitavat laskupisteet on mahdollista kopioida viereiseen laatikkoon. Tämä vähensi ohjelmoimistarvetta sekä paikkamuuttujien määrää puolella. Kaikki laskupisteet, joita esim. 300 mm:ssä pätkässä oli 15 kappaletta, pystyi kopioimaan viereiseen laatikkoon. Koordinaatiston ohjelmointi tapahtuu kolmessa eri vaiheessa. Ensiksi opetetaan origo, joka sijoitetaan laatikon kulmaan. Origosta opetetaan ensin koordinaatiston x-suunta, jonka jälkeen opetetaan y-suunta. Tämän jälkeen robotti ”tietää” laatikon paikan eli koordinaatiston. Työn loppuvaiheessa kouraan asennettiin koordinaatiston automaattiseen asetusta varten **ultraäänianturi**. Ultraäänianturilla mahdollistettiin koordinaatiston automaattinen asetusta laatikon reunoille. Tämä vähensi myös työntekijöiden työtä laatikon paikoituksessa ja laatikoiden vaihtotyö helpottui siis paljon. Anturista ja sen käytöstä lisää kappaleessa 6.2.

5 ROBOTIN OHJELMOINTI

Ohjelmointi alkoi perusliikkeiden sekä eri kontrollikäskyjen opettelulla. Huomasin hyödyllisiksi liikekomennoiksi nivel- sekä lineaariliikkeen, ja näitä käyttäen tein ensimmäiset harjoitteluohjelmat. Harjoitteluohjelma piti sisällään muutamien robotille opettujen pisteiden väliset liikkeet sekä paluun aloituspisteeseen. Kun sain liikkeistä selkeitä ja turvallisia lisäsin liikkeiden nopeutta sekä lisäsin ohjelmaan muutamia kontrollikomentoja. Näitä oli mm. JUMP- käsky jolloin robotti ”hyppäsi” eri ohjelmakohtien-, tai aliohjelmien välillä sekä TIMER- käskyjä eli odotuskomentoja. Tässä vaiheessa myös aliohjelmien käyttö selkeytyi ja karkea ohjelmarunko alkoi muodostua. Alapuolella on esitetty malli ensimmäisestä ohjelmasta (MALLI 1).

Rivi	Komento	Selitys
0	NOP	Ohjelman aloitus
1	*A	Ohjelma kohdan nimi
2	MOVJ 5.0	Nivel-liike 5 % nopeudella
3	TIMER 5.0	Laskuri 5 sekuntia
4	MOVL 50.0	Lineaariliike 50mm/s
5	JUMP *A	Hyppy ohjelmakohtaan A
6	END	Ohjelman lopetus

MALLI 1. Komentokäskeyjen esimerkkejä

Robotti liikkuu ohjelmassa loogisesti riviltä seuraavalle paitsi JUMP- komennon kohdalla, jossa komennon jälkeen oleva tunnus kertoo, mihin ohjelmakohtaan siirrytään. Tämä esimerkkiohjelma loppuu siis vasta, kun käyttäjä keskeyttää ohjelman.

5.1 Robotin liikkeet ja nopeudet

Ohjelma kirjoitetaan käyttöpaneelin ohjelmasisältökenttään. Kentässä näkyy selkeästi kaikki tarvittava informaatio, kuten ohjelman eri askeleet ja liiketavat. Aina kun ohjelmoidaan askel, tallennetaan yksi liikekomento. Liiketapoja robotilla on yhteensä 4 ja kaikkien ohjelmointi tapahtuu samantyyppisellä MOVE- komennolla. MOVE- komentojen kohdalla liiketapa tallentuu liikekomentona automaattisesti, kun opetettu sijainti on määritetty ja tallennettu. Lisäksi liikkeeseen on mahdollista lisätä erilaisia määreitä ja ehtoja kuten liikkeen kesto ja nopeus. Nämä laitetaan komentoon numeroina. Liikekomennon valinta tapahtuu ohjelmointipaneelin ”MOTION TYPE” - painikkeella.

Liiketapa määrittää reitin, jota robotti seuraa ohjelma-askelten välissä. Ajonopeus on robotin käyttämä liikenopeus. Helpoin ja loogisin tapa robotin ohjelmoinnissa on ”askel kerrallaan” ohjelmointi, jolloin robotti siirtyy ohjelmoidusta pisteestä

seuraavaan rivijärjestyksessä. Myös pisteiden lisäys ohjelman väliin käy vaivattomasti.

Nivel-liike on hyvin yleinen liikekomento pakkaustyössä. Sitä käytetään aina silloin, kun robotin ei tarvitse seurata määrättyä reittiä seuraavaan askelasemaan, vaan robotti kulkee optimoitua reittiä pitkin seuraavaan asemaan. Kun robotin akselia ohjelmoidaan nivelliikkeellä, käytetään MOVJ- liikekomentoa. Komennon perässä oleva luku kertoo prosentin huippunopeudesta. Pakkaustyössä liikenopeutena riitti yleisesti 10 prosentin nopeus. Ainoastaan pätkän noudossa hakupisteeltä oli käytettävä hieman nopeampaa liikettä, jotta seuraava annostelijalta tuleva pätkä ei aiheuttaisi törmäysvaaraa. Pyrin saamaan kaikki liikkeet laatikoiden yllä nivelliikkeeksi, koska liike oli tasainen ja optimaalinen ajan kannalta. Tasaisuus oli eduksi, koska pätkät pysyivät siirron aikana kourassa paremmin. Esimerkkikäsky: MOVJ VJ=20.00 UNTIL IN#(1)= ON, eli liike seuraavaan pisteeseen 20% nopeudella, kunnes 1. tulo-osoite tulee päälle.

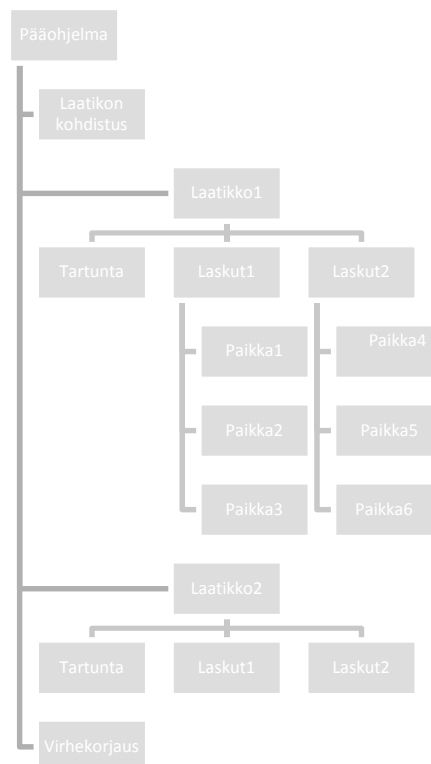
Lineaarinen liike oli käytetyin liiketapa robotille pakkausohjelmassa. Siinä robotti liikkuu lineaarisesti ohjelmoidusta pisteestä seuraavaan, ja tätä oli käytettävä varsinkin laatikon sisällä tapahtuvissa liikkeissä, jotta törmäyksiä laatikoiden laitoihin ei tulisi. Robotti muuttaa automaattisesti ranteen asentoa halutun pisteen mukaan, ja tämä on hyvin yleinen liikemuoto varsinkin hitsaustöissä. Robotin akselin lineaarisessa liikkeessä käytetään MOVL- komentoa. Komennon perään laiteaan liikenopeus numerona ja numeron yksikkö on mm/s. Esimerkkikäsky: MOVL V=50 UNTIL B001=1, eli liike pisteeseen nopeudella 50 mm/s, kunnes muistibitti 1 on 1.

IMOV- liike todettiin ohjelmoinnin loppupuolella myös hyvin hyödylliseksi liikekomennoksi. Laatikon paikoitustyössä se oli lopuksi jopa pakollinen, jotta laatikon koordinaatiston asettamisesta tulisi tarpeeksi tarkka. IMOV- liike on suoraviivainen liike nykyasemasta muuttujan määräämän matkan, johon kuuluu liikkeen suunta ja pituus. Liike on siis opetettu ja tallennettu valmiiksi tiettyyn paikkamuuttujaan ja tiettyyn koordinaatistoon. Esimerkkinä: IMOV P001 V=15 RF, eli liike paikkamuuttuja 1 koordinaattien mukaan nopeudella 15 mm/s robottikoordinaatistossa.

Muut 2 liiketapaa ovat ympyräliike (MOVC) sekä käyräliike (MOVS). Ympyräliikkeessä robotti kulkee kolmen pisteen määrittämiä kaaria pitkin, ja käyräliikkeessä kolmen pisteen muodostaman parabolisen käyrän mukaan. Näitä liikkeitä käytetään lähinnä epäsäännöllisen muotoisten kappaleiden käsittelyssä, kuten hitsaus-, leikkaus- ja maalaustöissä. Omassa työssäni näitä liiketapoja ei tarvittu.

5.2 Karkea ohjelmarunko sekä aliohjelmien käyttö

Robottiohjelman suunnittelussa piti ottaa huomioon ”kiertävä” ohjelmarunko, jotta ohjelmasta looginen ja helposti seurattava. Kiertävä runko vaati lisäksi tehokasta aliohjelmien käyttöä. Ohjelman olisi mahdollista tehdä myös yhdeksi pitkäksi ohjelmaksi, jossa lähdetään riviltä 1 liikkeelle ja edetään järjestyksessä loppuun asti, mutta tämä vaihtoehto on erittäin työläs sekä mahdollisten virheiden korjaus olisi hankalaa. Siksi suunnittelin rungon useaksi eri aliohjelmaksi eli suoritettava ohjelma liikkui eri aliohjelmien välillä muuttujien ohjaamana. Lopullinen ohjelma sisälsi 6 kappaletta aliohjelmiä, joten kaikki 4 ohjelmaa sisälsi yhteensä 24 kappaletta aliohjelmiä. Alla on esitetty ohjelman runko (KAAVIO 1).



KAAVIO 1. Yhden ohjelman karkeasti kuvattu runko

Pääohjelman käynnistyttyä ohjelma siirtyy laatikon kohdistukseen. Kohdistuksen jälkeen ohjelma siirtyy laatikko1:een. Kun robotti on tarttunut pätkiin, se menee laskupaikkaan, johon tietty muuttuja osoittaa. Tämä toistuu halutun määrän verran (muuttuja), jonka jälkeen ohjelma siirtyy toiseen laatikkoon. Sama prosessi toistuu laatikko 2:ssa. Ohjelma palaa laatikon kohdistukseen kun laatikko 2 on valmis.

5.3 Muuttujabitit ja niiden käyttö ohjelmassa

Muuttujabiteillä on useita tärkeitä tehtäviä ohjelmassa. Niiden avulla kontrolloidaan robotin ohjelmakulkua rivien- sekä eri aliohjelmien välillä. Konkreettisia toiminnan ohjauksia ovat mm. työstettävän laatikon määritys sekä laskupaikka-kierron toteutus loogisessa järjestyksessä. Motoman ohjelmasta löytyy paikkoja muuttujabiteille 99 kappaletta, joihin voidaan tallettaa luku välillä 1-99. Näitä numeroita voidaan muunnella matemaattisesti usealla eri tavalla. Käytetyt muutokset näissä ohjelmissa ovat vähentäminen ja lisääminen. Seuraavassa esimerkkinä laskupaikan määritys:

Laskupaikan ohjausbitin osoite on B004. Jokaisen tartunnan jälkeen B004:ää lisätään yhdellä. Kaikille laskupaikoille määritetään ehto, eli Paikka 1 jos $B004 = 1$, Paikka 2 jos $B004 = 2$ jne., joten kun jokaisen tartunnan jälkeen bittiin lisätään 1, ohjelma osaa kiertää laskupaikat oikeassa järjestyksessä, ja näin ohjelmoinnin määrä laskee huomattavasti.

Töissä käytin 100 kappaletta muuttujia. Ohjaustöitä ohjelmassa on paljon, joten myös muuttujia tarvitaan monta. Alla esitän luettelon kaikista muuttujabiteistä sekä niiden tehtävistä eri ohjelmissa (LUETTELO 1).

Bitti nro. Tehtävä

- B000 = 1. laatikon kierrosmäärän muistibitti. Välilaskukierroksen jälkeen bittiä lisätään aina yhdellä. Lukemaa verrataan B009:ään.
- B001 = 2. laatikon kierrosmäärän muistibitti. Välilaskukierroksen jälkeen bittiä lisätään yhdellä. Lukemaa verrataan B009:ään.
- B002 = Tartunnan seurantabitti. Kytkeytyy päälle, jos tartunta onnistuu.
- B003 = Laatikon valintabitti. Laatikoida on lankakoosta riippuen joko 2 tai 3, ja tämä bitti ohjaa laatikon valintaa.
- B004 = Laskupaikan määritysbitti. Paikka numero 1 on vuorossa, kun bitin arvo on 1, paikka numero 2, kun arvo 2 jne.
- B005 = Tartuntavaiheen ohjausbitti. Tämä bitti kytkeytyy päälle jos haku-hetkellä annostelu/katkaisulaitteessa tapahtuu virhe. Tällä ohjataan robotin toimintoja annosteluvirheen jälkeen ja lisäksi käyttäjän informointi tapahtuvista ongelmista ennen robotin työtä onnistuu helposti.
- B006 = Laatikon vaihdon ohjausbitti, eli ollessa luku 1, robottiohjelma osaa siirtyä laatikon vaihto-ohjelmaan.
- B007 = Ohjaa ohjelman virheohjelmaan, jossa tapahtuu bittien palautus edelliseen työkierron arvoihin.
- B008 = Laskupaikan määritysbitin varamuistipaikka. Koska jossain kohtaa ohjelmaa on hyvä tehdä virheen jälkeen laskubitin nollaus oikean ohjauksen takia, on laskupaikan hyvä laittaa ”varapaikkaan” palautumisen onnistumisen varmistamiseksi.

- B009 = Laskujen kierrosmääräbitti. Tähän osoitteeseen laitetaan haluttu kierrosmäärä 300mm lankojen osalta. Ohjelma vertaa tätä lukua B000:aan ja päästää ohjelman eteenpäin vasta kun bitit ovat samat.
- B010 = Laskujen kierrosmääräbitti. Tähän osoitteeseen laitetaan haluttu kierrosmäärä 400mm lankojen osalta. Ohjelma vertaa tätä lukua B000:aan ja päästää ohjelman eteenpäin vasta kun bitit ovat samat.
- B011 = Laskujen kierrosmääräbitti. Tähän osoitteeseen laitetaan haluttu kierrosmäärä 230mm lankojen osalta. Ohjelma vertaa tätä lukua B000:aan ja päästää ohjelman eteenpäin vasta kun bitit ovat samat.
- B012 = Laskujen kierrosmääräbitti. Tähän osoitteeseen laitetaan haluttu kierrosmäärä 500mm lankojen osalta. Ohjelma vertaa tätä lukua B000:aan ja päästää ohjelman eteenpäin vasta kun bitit ovat samat.
- B013 = Ohjaa B007:n tavoin ohjelmaa virheen jälkeen virheohjelmaan
- B014 = Erikoistulon no.39 eli SIN #(39) muistibitti. Työn alkuvaiheessa erikoistulo luetaan ohjelman alkuvaiheessa, ja siirretään B014 muistiin. Tämä muistibitti ohjaa laatikon valintaa, ja kun se on päällä, siirtyy robotti suoraan 2. laatikkoon.
- B015 = Virheohjausbitti. Ohjaa B007 ja B013 tavoin virhetilanteen toipumista. Käytössä 2. laatikon mahdollisissa virheissä.
- B099 = Laskuryhmien määritysbitti. Laatikoissa laskut tapahtuvat kahdessa eri ryhmässä (laskut1 ja laskut2), ja tämä bitti valitsee ryhmän.

LUETTELO 1. Muuttujabittien numerot ja tehtävät

5.4 Paikkamuuttajat ja niiden käyttö ohjelmassa

Paikkamuuttajat asetetaan ohjelmointitilassa. Toisin kuin bittimuuttajat paikkamuuttajiin asetetaan muistiin opetettuja liikkeitä tai koordinaatistopisteitä halutuissa koordinaatistoissa. Opetettuja liikkeitä ovat mm. rannekäännöt eli *IMOV*-liikkeet kohdistuksessa, ja opetettuja pisteitä ovat mm. laskupaikat laatikoissa sekä lankojen hakupiste annostelupaikassa. Koordinaatiston vapaa valinta on ohjelmointia suuresti helpottava toiminto. Sen avulla opetetut paikkamuuttajapisteet voi helposti kopioida toisiin koordinaatistoihin eli toisiin laatikoihin. Näin ollen samaa paikkamuuttujaa voi käyttää saman ohjelman jokaisessa laatikossa. Pisteitä ja liikkeitä voi opettaa myös robotin- tai peruskoordinaatistoon, eli liike tapahtuu aina samassa pisteessä. Työssä käytetään 127:ää paikkamuuttujaa. Käytetyt paikkamuuttajat on esitelty alla (LUETTELO 2).

Paikka	Tehtävä	Koordinaatisto
P000	1. koordinaatiston asetukset	Perus
P001	2. koordinaatiston asetukset	Perus
P002	3. koordinaatiston asetukset	Perus
P003	4. koordinaatiston asetukset	Perus
P004	5. koordinaatiston asetukset	Perus
P005	6. koordinaatiston asetukset	Perus
P006	Suora <u>loppulasku</u> laatikon pohjalle	#1
P007	300mm langan väliasema (2. laatikko)	#2
P008	Opetettu sivuliike kohdistuksessa	Perus
P009	Laskuvirheen jälkeinen väliasema	Robotti
P010	”Kotiasema” eli langan noudon odotuspaikka	Robotti
P011	Tartunta-asema 300mm langalla	Robotti
P012	300mm langan 1. väliasema	Robotti
P013-P020	300mm langan peruslaskut 8 kpl	#1
P021	300mm langan 2. väliasema	#1
P022-P029	300mm peruslaskujen yläasemat 8 kpl	#1
P030-P036	300mm langan välilaskuasemat 7 kpl	#1

P037-P043	300mm välilaskujen yläasemat 7 kpl	#1
P044	400mm langan 1. ja 2. laskuasema	#3
P045	Keskiasema 400mm ongelmissa	Robotti
P046-P051	400mm langan peruslaskut 6 kpl	#3
P052-P058	400mm langan välilaskut 7 kpl	#3
P059	300mm langan välinousu, jotta kouran ranne oikein	#1
P060	Ei käytössä	
P061	<u>Vino</u> lisälasku laatikon pohjalle.	#1
P062	237mm langan tartunta-asema	Robotti
P063-P068	237mm peruslaskujen yläasemat 6 kpl	#5
P069-P074	237mm langan peruslaskut 6 kpl	#5
P075-P079	237mm langan välilaskut 5 kpl	#5
P080-P084	237mm välilaskujen yläasemat 5 kpl	#5
P085	237mm langan väliasema laatikon ja kotias. välillä	#5
P086	400mm langan välipiste laatikon ja kotias. välillä	Robotti
P087-P093	400mm peruslaskujen yläasemat 7 kpl	#3
P094-P100	400mm välilaskujen yläasemat 6 kpl	#3
P101-P108	500mm peruslaskut 8 kpl	#8
P109-P116	500mm peruslaskujen yläasemat 8 kpl	#8
P117	Välipiste 500mm laatikon ja kotiaseman välillä	Robotti
P118	400mm langan tartuntapiste	Robotti
P119	Liike oikeaan päytyyn kohdistuksessa	Robotti
P120	Ranteen 90° käänö kohdistuksessa	Robotti
P121	Liike taakse reunan päälle kohdistuksessa	Robotti
P122	Ranteen 90° käänö kohdistuksessa	Robotti
P123	Hylkypiste langan tartunnan jälkeen	Perus
P124	Liike vasemman reunan päälle kohdistuksessa	Robotti
P125	Liike taakse takaseinään kohdistuksessa	Robotti
P126	Liike ylös reunan tasolle kohdistuksessa	Robotti
P127	Tartuntapaikka 500mm langalle	Robotti

LUETTELO 2. Työssä käytettävät paikkamuuttajat, tehtävä ja koordinaatisto

Lankojen laskuja laatikkoon sanotaan siis **perus- tai välilaskuiksi**. Välilaskun nimi tulee siitä, että lasku tapahtuu konkreettisesti kahden pinon väliin. **Yläasemalla** tarkoitetaan laskupaikan yläpuolella olevaa pistettä, josta koura alkaa laskea suoraan alaspäin kohti laatikon pohjaa. Tässä kohtaa myös ultraäänianturi alkaa tehdä mittaustyötä pohjan- tai lankapinokorkeuden määrittystä varten. Eri lankakokojen **väliasemat** on ohjelmoitu vähentämään liikevirheitä asemien välillä. Virheitä voisi muuten tulla mm. ranteen väärästä kiertosuunnasta tai maksimi ulottuvuuden ylittämisestä.

5.5 Pinoamisohjelman käyttö

Lankarivien määrän lisääntyessä myös pinnankorkeus luonnollisesti kasvaa. Robotin pitää siis nostaa rivin jälkeen automaattisesti tiputuskohtaa, jotta koura välttyisi törmäyksiltä. Tätä varten robotilla on mahdollisuus toteuttaa pinoamistoiminto. Siinä ideana on tallentaa robotille rivin viimeisen langan laskukorkeus ja tämän jälkeen lisätä sitä matemaattisesti tarvittava määrä ylöspäin. Pinoamisohjelman alussa robotille opetetaan laatikon pohja, joka on tietysti muuttumaton, koska laatikot kiinnitetään vakiokorkuiselle lavalle. Tämän jälkeen robotti tekee yhden ”kerroksen”, jonka jälkeen robotti nostaa laskukorkeutta käyttäjän määräämän muuttujan verran. Tämä toiminto tuntui aluksi toimivalta, mutta pitempiaikainen tarkastelu osoitti, että törmäyksiä sattuu liian usein. Syynä tähän on pinnankorkeuserot eri laskukohtien välillä muutaman laskurivin jälkeen. Langat eivät voi pudota kourasta aina saman lailla. Vaikka pudotuspaikat ovat vakiot, joihinkin kohtiin lankoja valuu enemmän ja pinnasta tulee näin hieman epätasainen. Jos tämä tapahtuu ohjelman alkuvaiheilla, on ero lopussa jo muutamia senttejä. Vakiona pysyvää nostokorkeutta ei siis pystytä määrittämään, ja lähdimmekin pohtimaan menetelmää, jolla laskumäärä voitaisiin määrittää joka laskupaikan kohdalla juuri sillä hetkellä olevan pinnankorkeuden mukaan.

5.6 Ultraäänianturien toiminta pakkaustyössä

Anturit asennettiin kouran runkoon ja johdotus lähtösignaalille tehtiin varressa olevaan rasiaan. Ultraäänianturit hankittiin ensisijaisesti välttämään laskujen aikana tapahtuvia ongelmia, sekä asentamaan automaattisesti laatikoiden koordinaatis-

tot ja helpottamaan näin laatikoiden kohdistustyötä. Ennen antureita lasku tapahtui pinoamisohjelman avulla. Se toimi kohtuullisesti, mutta melko usein sattuneet törmäykset muuttivat ajatuksia.

Ensimmäinen anturi asennettiin niin, että se näytti suoraan alaspäin. Ultraäänianturi teki laskuvaiheessa mittaustyötä, jotta törmäyksiä ei tapahtuisi, ja jotta koura pääsisi mahdollisimman lähelle pintaa korkeuseroista huolimatta. Pienen harjoittelun jälkeen huomasimme anturin hyvin tarkaksi, ja alkuun epäilemäni heijastusongelmat eivät toteutuneet. Toistotarkkuuden toleranssi oli alle 5 milliiä. Anturin toiminta on melko yksinkertainen. Aluksi anturille opetetaan keila-alue eli konkreettinen minimi ja maksimi keilapituus. Anturissa on opetusta varten TEACH - näppäin, jolla opetetaan kohteen yläpuolella halutut etäisyydet. Etäisyydet tallentuivat opetuksen jälkeen anturin muistiin, ja tehdasetukset sai haluttaessa käyttöön reset- näppäimellä. Koska robotilla on paljon aikaa mittaustyöhön, ja liiallinen nopeus suurensi toleranssia, valitsin mittausnopeudeksi melko hitaan, n. 50mm/s. Tästä syystä keila-alue oli myös kohtalaisen pieni, jotta mittausaikaan ei tuhrautuisi liikaa aikaa. Kun kohde saapuu keilan sisälle, syttyy anturin ensimmäinen led-valo päälle. Kohteen saapuessa minimi- ja maksimipisteiden keskelle, syttyy anturin toinen led-valo, ja lähtösignaali menee päälle. Tätä anturitietoa käytetään laskuohjelmassa hyväksi. Mittauslaskua tehdään aina niin kauan, kunnes tulotieto saapuu robotille, ja mittauksen pysyessä samana voidaan lopullinen lasku tehdä vakiopaikkamuuttujan määräämän matkan verran mikä työssä oli n. 100 mm.

Toinen ultraäänianturi asennettiin tekemään automaattisen koordinaatiston asennuksen. Ennen anturin asennusta laatikko piti kohdistaa tarkasti maassa oleviin merkkeihin, jotta valmiiksi opetettu koordinaatisto oli tarkasti laatikon paikalla. Jos laatikko jostain syystä jäi vinoon, oli seurauksena usein kouran törmäys laskuvaiheessa laatikon reunaan. Anturi asennettiin kouraan niin, että se osoitti suoraan eteenpäin. Myös tämä anturi todettiin erittäin tarkaksi useiden toistojenkin jälkeen. Automaattista kohdistusta varten tein pääohjelman alkuun erillisen kohdistusohjelman, jossa robotin koura meni laatikon sisään ja anturin avulla mittasi laatikon sivuja. Mittatulosten perusteella robotti teki tarvittavia vakioina pysyviä paikkamuuttujaliikkeitä asettaen koordinaatiston origon-, X-, ja Y- paikan oikein.

Tarvittavia paikkamuuttujia olivat liikkeet mittausmatkan verran sivuille ja eteen, sekä ranteen 90 asteen käännökset laatikon sivujen mukaan. Koordinaatiston asetusjärjestys on seuraavanlainen:

1. Koura siirtyy laatikon keskelle anturin osoittaessa kohtisuoraan laatikon pitkää sivua.
2. Robotti liikkuu suoraan laatikon sivua kohti, kunnes anturi ilmoittaa kohteen kytkemällä lähdön päälle. (Suora liike on paikkamuuttujaliike = mittausväli).
3. Robotti kääntää kouran 90 astetta lyhyttä sivua kohden, jonka jälkeen nostaa kouran suoraa ylös reunan tasolle. (Nostokorkeus on vakio, koska laatikot ovat samankorkuisilla lavoilla).
4. Koura tekee sivuliikkeen reunalle, jonka jälkeen robotti ottaa paikan muistiin koordinaatiston Y-suunnaksi GETS-komennolla. (Sivuliike on vakio paikkamuuttujaliike = mittausväli).
5. Koura siirtyy kohtisuoraan lyhyelle sivulle, ja tekee vastaavanlaisen mittausoperaation kuin edellä.
6. Robotti asettaa lyhyen sivun päälle koordinaatiston X-suunnan GETS-komennolla.
7. Kouran käännöt ja mittaukset nurkkaan, jonne GETS-komennolla koordinaatisto origon asetus.
8. Tämä sama prosessi toistuu toisessa laatikossa, jonka jälkeen robotti siirtyy kotiasemaan ja kuittaa käyttäjälle onnistuneen koordinaatistojen asetuksen.

Käyttäjän pitää kohdistusohjelmasta huolimatta tarkastaa ohjelman alussa, että laatikoiden pitkä sivu on suorassa linjassa. Tätä varten kohdistusta ennen robotti käy esikohdistusohjelman, jossa koura liikkuu laatikoiden keskilinjaa pitkin aina päässä pysähtyen. Näin käyttäjä saa laatikon sivun oikeaan linjaan. Tämän jälkeen 90 asteen käännökset menevät laatikon sisällä oikein, ja koordinaatisto tulee tarkkaan oikeaan paikkaan.

6 YHTEENVETO

Projekti oli kaiken kaikkiaan onnistunut pienten ongelmien jälkeen. Suurimmat ongelmat ohjelmointiin liittyen olivat laatikoiden koordinaatistojen tarkka määrittäminen, sekä virheistä toipuminen. Suurimpia mekaanisia ongelmia olivat työn alussa annostelulaitteen toiminta sekä langan syöttöhäiriöt.

Koordinaatiston pienikin heitto saattoi vaikuttaa yllättävän paljon lopputulokseen useiden laskujen jälkeen. Seurauksena koordinaatiston huonosta määrittämisestä oli epäsiistit lankapinot tai törmäykset laatikon reunoihin. Vaikka käytössä oli aina samankokoinen lava (eurolava), ja laatikoiden kiinnitys lavaan oli vakio, niin muutamia törmäyksiä sattui varsinkin toisen laatikon päädyissä. Tähän saattaa olla syynä anturin pieni toleranssi tai lavan asetus liian vinoon työn alussa. Lavapaikkojen vakiomäärittely esim. lattiakorokkeilla voisi auttaa asiaa jonkin verran.

LÄHTEET

- /1/ Vertic Oy, 2005, Vertic Oy yritysopas, Pori.
- /2/ Motoman Robotics Finland Oy, 2001, Motoman XRC käyttöohjekirja, Turku, Robia Suomi Oy.
- /3/ Motoman Robotics Finland Oy, 2001, Motoman – robottien ohjaus Yasnac XRC laitekäsikirja, Turku, Robia Suomi Oy.
- /4/ Siemensin huolto/tukisivut, Siemens – Simatic S5 ET200 Distributed I/O System MANUAL.[Verkkodokumentti]. [Viitattu 9.1.2007].
Saatavissa:
http://support.automation.siemens.com/WW/skm/pdfviewer.asp?HitsPerSite=10&ehbid=html_76%2Fehb%2F21667381.htm&lang=en&nodeid=21668860&siteid=cseus&query=S5-95U&page=1&view=new

LIITTEET