Olli-Pekka Hakala

Robottijyrsintä 3D-mallin perusteella

Insinööri (AMK)

Konetekniikka

Kevät 2021

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Hakala Olli-Pekka

Työn nimi: Robottijyrsintä 3D-mallin perusteella

Tutkintonimike: Insinööri (AMK), konetekniikka

Asiasanat: robotiikka, jyrsintä, tietokoneavusteinen valmistus, käyttöönotto

Robottien tarkkuus on kehittynyt vuosien mittaan sellaiseksi, että niitä käytetään maailmalla jo melko yleisesti jyrsintäsovelluksiin. Myös työn toimeksiantaja Sunrob Robotics on toimittanut jyrsintäsovelluksia, mutta jyrsintäprosessin suoritustapaa ja dokumentointia haluttiin jatkokehittää. Tässä työssä keskityttiin kuvaamaan prosessi, jossa 3D-malli jalostetaan robotilla jyrsityksi kappaleeksi. Lähtökohtana oli, että työssä käytetään Fusion 360- ja RoboDK-ohjelmia.

Työ alkoi käytettäviin ohjelmiin tutustumisella. Tämän jälkeen tehtiin erilaisia käyttöönottotehtäviä, jotta robotti saatiin toimintakuntoon. Tähän vaiheeseen sisältyi muun muassa kenttäväylän konfigurointi, robotin referenssipisteen ja työalueen määrittäminen sekä painonappiohjauksen ohjelmointi.

Ohjelmapuolella pyrittiin aluksi löytämään parhaat työkalut erilaisten pinnanmuotojen jyrsimiseen Fusionissa ja luomaan näistä työstöratoja. Jatkossa päätettiin kuitenkin keskittyä enemmän tekstien ja kuvioiden jyrsimiseen kappaleen pintaan. Yrityksiltä tuli kyselyjä myös putkien aukotukseen liittyen ja tähänkin löydettiin sopiva työkalu. Työstöratojen luomisen jälkeen perehdyttiin ohjelman siirtoon robotille ja ratkaistiin tähän liittyviä ongelmia.

Yrityksen demolaitteisto mahdollisti runsaan testaamisen käytännössä ja tällä tavalla pystyttiin ottamaan huomioon asioita, jotka eivät tulleet esille simulointien aikana. Testien aikana nähtiin esimerkiksi, miten lastuamisarvot vaikuttivat työn laatuun ja havaittiin muita ohjelmiin liittyviä virheitä.

Lopputuloksena työstä jäi yritykselle yksityiskohtaiset ohjeet jyrsintäohjelman tekemiseen sekä Reference testin ja Brake testin ohjelmointiin.

KAMK • University of Applied Sciences

Abstract

Author(s): Hakala Olli-Pekka

Title of the Publication: Robot Milling Based on 3D model

Degree Title: Bachelor of Engineering, Mechanical Engineering

Keywords: robotics, milling, computer-aided manufacturing, commissioning

Over the years, the precision of robots has evolved to be used quite widely in the world for milling applications. Sunrob Robotics, the client, has also supplied milling applications, but the aim was to further develop the way the milling process was performed and documented. This thesis focused on describing the process by which a 3D model is processed into a piece milled by a robot. The premise was that Fusion 360 and RoboDK were used in the project.

The work began with an acquaintance with the programs used. Various commissioning tasks were then performed to get the robot up and running. This step included, among other things, configuring the fieldbus, determining the robot reference point and work area, and programming the pushbutton control.

In the program, the aim at first was to find the best tools for milling different surface shapes in Fusion and to create toolpaths from them. In the future, however, it was decided to focus more on milling texts and patterns on the surface of the part. Companies also sent inquiries regarding pipe cutting solutions and a suitable tool was found in Fusion 360. After the creation of toolpaths, transfer of the program to the robot was studied and related problems were solved.

The company's demo equipment enabled extensive testing in practice, and in this way it was possible to take into account issues that did not arise during the simulations. During the tests, for example, it was seen how the cutting values affected the quality of work and other program-related errors were noted.

As a result, the work provided detailed instructions for making the milling program and the company also had detailed instructions on how to program the Reference test and the Brake test.

Sisällys

Termit ja käsitteet

Aliohjelma	Toiminnallinen ohjelma, joka voi sisältää esimerkiksi liikekäskyjä tai toimilaitteiden ohjauksia väylän kautta. Voidaan ajaa itsenäisesti tai pääohjelman kautta.
CAM	Computer-Aided Manufacturing. 3D-mallin tietojen hyödyntämistä koneistuksessa, esimerkiksi työstöratojen luominen valitsemalla osia mallinnetusta kappaleesta.
Config.dat	Tekstitiedosto, jossa on määritelty robotin omia muuttujia sekä käyttäjän määrittelemiä muuttujia. Ei sisällä liikekäskyjä tai muita toimintoja.
IFR	International Federation of Robotics eli kansainvälinen robotiikkaliitto
KRL	KUKA Robot Language, KUKA-roboteissa käytettävä ohjelmointikieli
Pääohjelma	Robotin ohjelmarunko, jossa kutsutaan halutussa järjestyksessä aliohjelmia. Yritetään pitää mahdollisimman selkeänä.
Sketch	3D-mallinnuksessa käytettävä luonnos, johon voidaan piirtää ja mitoittaa halutut muodot.
Solidi	Sketch muuttuu solidiksi siinä vaiheessa, kun sille määritetään paksuus eli se pursotetaan (Extrude). Tämän jälkeen kappaleella on jokin tilavuus ja sen muotoja voidaan käyttää esimerkiksi työstöradan luomiseen.
Sps.sub	Taustalogiikka, jossa määritellään esimerkiksi ehdot painonappien toiminnalle. Robotccti tarkastaa määrättyjen muuttujien tilan aina tietyn ajan välein.
TCP	Tool Center Point. Jokaiselle työkalulle erikseen määritetty piste, joka seuraa valittua rataa. Yleensä pisteeksi valitaan työkalun kärki.

1 Johdanto

Sunrob Robotics Oy on 2006 perustettu, nykyisin Lappeenrannassa toimiva robotiikkaratkaisuja tarjoava yritys. Yritys on kehittänyt erilaisia jyrsintä-, pakkaus- ja lavaussovelluksia. Yritys käyttää sovelluksissaan pääosin saksalaisen KUKA:n robotteja.

Työn tavoitteena oli kehittää toimiva työtapa, jossa jyrsinrobotille luodaan työstörata erilaisia 3D-mallinnettuja kappaleita hyödyntämällä. Idea työhön tuli yritysten osoittamasta kiinnostuksesta robotilla suoritettavaa jyrsintää kohtaan. Alun perin suunnitelmana oli luoda 3D-malli skannaamalla kappale ja muuttamalla skannattu pistepilvi solidiksi, mutta yrityksen skanneri ei soveltunut tämäntyyppiseen käyttöön. Tämän seurauksena päätettiin ohittaa skannausvaihe ja luoda 3D-mallit perinteiseen tapaan. Yrityksellä oli voimassa olevat lisenssit Fusion 360- ja RoboDK-ohjelmiin, joten toiveena oli, että näitä käytettäisiin työn suoritukseen. Fusion 360 on Autodeskin pilvipohjainen suunnitteluohjelmisto, jolla voidaan esimerkiksi mallintaa 3D-kappaleita ja luoda työstöratoja koneistukseen. RoboDK puolestaan on yleisesti käytössä oleva simulointiohjelma robotiikkasovelluksiin.

Työssä käytetty KUKA KR120 R2500 pro -robotti oli ollut käyttämättömänä jonkin aikaa, joten työn alkuvaiheessa aikaa kului erilaisiin käyttöönottotehtäviin. Käyttöönoton aikana sähkökytkennät suoritti yrityksessä työskentelevä automaatioasentaja.

2 Teoriataustan esittely

2.1 Robotti

Mikä on robotti? Asia on määritelty eri standardeissa hieman eri tavalla. Kansainvälisen robotiikkaliiton (IFR) käyttämä määritelmä teollisuusrobotille tulee ISO 8373 -standardista, ja ydinsanoma on suomennettuna ”Automaattisesti ohjattu, uudelleenohjelmoitava, monikäyttöinen laite, jolla on kolme tai useampia vapausasteita. Voidaan asentaa kiinteästi tai liikkuvaksi.” [1.] Teollisuusroboteista puhuttaessa liikkuva robotti tarkoittaa käytännössä jonkinlaisella radalla kulkevaa robottia. Robottityypeistä tyypillisimpiä ovat juuri aiemmin mainitut teollisuusrobotit (Kuva 1) ja tämän vuoksi opinnäytetyössä keskitytään myös tällaiseen. Muita robottityyppejä ovat esimerkiksi viime vuosina pinnalle nousseet tavaran kuljetukseen sopivat mobiilirobotit sekä yhteistyörobotit eli cobotit.

Robotteja käytetään pääasiassa työtehtävissä, joissa on isot volyymit ja paljon toistoa. Myös tehtävän vaarallisuus ihmiselle voi olla perusteena robottihankintaa mietittäessä. Muita robotiikan etuja ovat tehtävien helppo muokattavuus ja pieni tilan tarve esimerkiksi tuotantolinjaan verrattuna, kasvava tuotannon tehokkuus sekä työn tarkkuus servomoottoreilla tehtävien liikkeiden ansiosta. [1.] Robottien ohjaus tapahtuu valmistajakohtaisen ohjelmointikielen avulla. KUKA:n roboteissa ohjelmointikieli on Python-pohjainen KUKA Robot Language (KRL).

Kuva 1. Yleiskuva teollisuusrobotista [2.]

2.2 Tiedostomuodot

Oikeiden tiedostomuotojen käyttäminen prosessin eri vaiheissa on tärkeää, sillä kaikki tiedostomuodot eivät sisällä esimerkiksi mittatietoja. Myös geometrian muokattavuus ja yhteensopivuus eri sovelluksien kanssa vaihtelee suuresti tiedostomuodosta riippuen. Tietyt standardimuodot, kuten 3D-mallit STEP-tiedostoina, toimivat useimmissa ohjelmissa. RoboDK:ssa voi käyttää joko valmiita työkaluja tai luoda omia.

Luodut työkalut kiinnittyvät jatkossa suoraan robotin käsivarteen, kun ne tallennetaan .tool-muotoon. Robottiohjelmaa valittaessa tiedostotyyppi määrittyy valitun Post Processorin perusteella. Esimerkiksi KUKA-robotin toiminnalliset ohjelmat ovat Source-tyyppiä (.src) ja muuttujien määrittelyyn tarkoitetut ohjelmat DAT-tyyppiä (.dat). Jos Fusionilla tehty työstörata halutaan siirtää radaksi RoboDK-ohjelmaan, voi olla tarpeen tallentaa rata G-koodina eli .tap-muodossa. G-koodiksi muuttaminen onnistuu ainakin valitsemalla Post Processoriksi ”5AXISMAKER”-nimisen vaihtoehdon. Tutkimuksen edetessä löydettiin kuitenkin toimintatapa, jolla pystyttiin minimoimaan tiedostomuotojen muutokset.

2.3 Jyrsintä

Jyrsintä on lastuavan työstön menetelmä, jossa pyörivä työkalu irrottaa materiaalia työkappaleesta. Yleisin jyrsintyyppi on pystyjyrsinkone, mutta viime vuosina myös robotteja on alettu käyttää enemmän jyrsinsovelluksissa. Jyrsintä on yleisin koneistusmenetelmä tasaisille pinnoille, mutta erimuotoisilla ja -kokoisilla työkaluilla voidaan myös esimerkiksi porata ja tehdä hammastuksia. Hyvän lopputuloksen kannalta oleellisia asioita ovat oikeat työkalut ja lastuamisarvot. [3.] Robotin etuna jyrsittäessä on käsivarren taipuminen hankaliinkin paikkoihin ilman kääntöpöytiä tai muita lisälaitteita. Heikkoutena puolestaan on robotin soveltuminen vain suhteellisen pehmeiden materiaalien jyrsintään.

2.4 Lastuamisarvot

Jyrsinnässä lastuamisarvoista tärkeimmät ovat karan pyörimisnopeus, syöttönopeus eli terän sivusuuntainen liikenopeus ja lastuamissyvyys. Kalle Juntunen on opinnäytetyössään tutkinut robottijyrsintään sopivia lastuamisarvoja [4.]. Työssä käytettiin materiaaleina mäntyä ja koivua

sekä tutkittiin, miten esimerkiksi kierrosnopeus tai lastuamissuunta vaikuttavat lopputulokseen. Myös terän kulumista ja äänentuottoa tarkasteltiin. Näitä kierrosnopeussuosituksia käytettiin suuntaa antavana ohjeena lastuamisarvoja asettaessa.

2.5 Työn näkökulma

Työtä tarkasteltiin tuotannollisesta näkökulmasta sekä pyrittiin löytämään tehokas ja selkeä työtapa, jotta eri vaiheisiin ei kuluisi liikaa aikaa. Työtä aloittaessa lähdettiin liikkeelle ajatuksella, että joitain muokkauksia täytyy varmasti tehdä aina, kun uudelle kappaleelle lähdetään tekemään ohjelmaa. Työssä tutkittiin myös, millaisia työstöratoja kummankin sovelluksen puolella kannattaa luoda. Työstä jäävien ohjeiden tuli olla selkeitä ja yksityiskohtaisia, jotta niitä voidaan käyttää jatkossa yrityksen sisäiseen ohjaukseen.

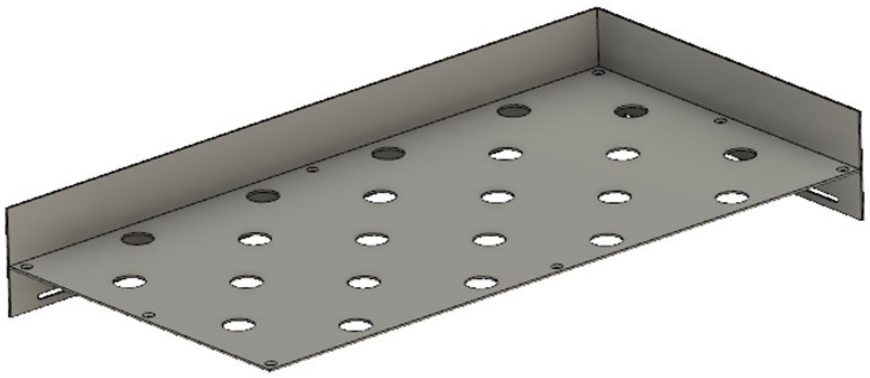
3 Tutkimus ja työn suoritus

3.1 Käytettäviin ohjelmiin perehtyminen

Simulointiohjelmiin perehtyminen alkoi jo syksyllä, mutta työn tarkka aihe oli kuitenkin pitkään epäselvä ja vasta alkuvuodesta alkoi hahmottumaan tarkemmin, minkälaisia toiveita toimeksiantajalla ja mahdollisilla asiakkailla on tutkimukseen liittyen. Syksyllä ensimmäisiä askeleita olivat työstöratojen luomiseen tutustuminen Fusionin CAM-työkaluilla. Työstöratojen luonnissa keskityttiin löytämään parhaat CAM-työkalut eri pinnanmuodoille ja tutkittiin, mikä osa geometriasta täytyi valita milläkin työkalulla. Geometrian valinnassa oli yllättävän paljon eroavaisuuksia ja oikeiden asetusten löytämiseen kului runsaasti aikaa. RoboDK:n puolella lähdettiin liikkeelle yksittäisten liikekäskyjen opettamisesta robotille ja tutustuttiin yleisiin komentoihin.

3.2 Työn aloitus

Jyrsinpöytään tehtiin runko alumiiniprofiileista ja suunniteltiin pöytälevy sekä laidat kahdelle sivulle työstettävän kappaleen paikoittamiseksi. Pöytälevyyn tehtiin ympäriinsä 50 mm reikiä, jotta työstettävä kappale oli mahdollista kiinnittää pöytään liimapuristimilla. Kuvassa 2 kokoonpanokuva pöytälevyistä.



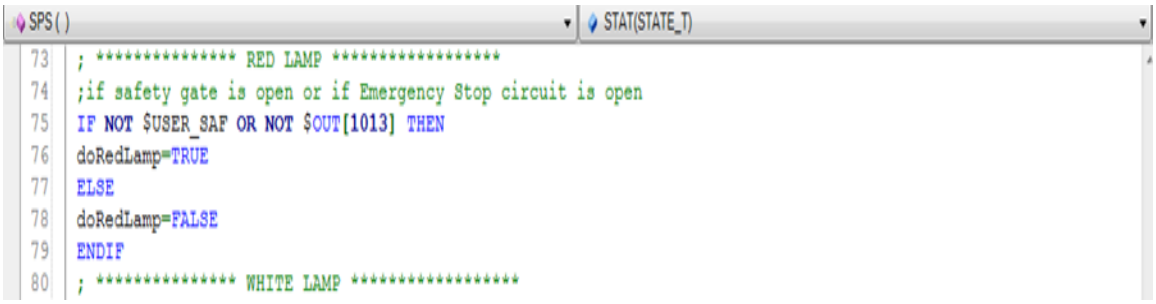
Kuva 2. Kokoonpanokuva suunnitelluista pöytälevyistä.

3.3 Käyttöönottovaihe

Seuraavaksi luotiin painonappien ohjaus sps.subiin eli taustalogiikkaan. Käytännössä tämä tarkoitti ehtojen asettamista nappien valoille ja valotornille sekä nappien ohjaamien toimintojen määrittämistä. Napit asetettiin toimimaan niin, että niillä voitiin ajaa ja pysäyttää valittu ohjelma, jos käsiohjaimesta valittiin ohjaustilaksi ”EXT” eli ulkoinen ohjaus. Ohjauskaappiin kiinnitetty valotorni puolestaan viesti käyttäjälle ohjelman tilan seuraavasti:

- Vihreän valon palaessa ohjelma on käynnissä.
- Punainen valo palaa, kun ohjelmaa ei ole valittu tai ohjelman suoritus on katkennut häiriön seurauksena.
- Kun häiriön aiheuttaja on poistettu, valkoinen valo syttyy ja valkoista nappia painamalla saadaan kuitattua virheilmoitus.

Ohjelma lähtee uudestaan käyntiin start-nappia painamalla. Kuvassa 3 on esimerkki valojen ohjauksen määrityksistä. Samaa ohjelmaa käytti myös eräs ulkomaalainen harjoittelija toisessa projektissa, joten kommentit kirjoitettiin englanniksi. $-merkki sanan edessä tarkoittaa, että kyseessä on signaali. Osa signaaleista oli määritelty Config.datiin, jolloin niitä voitiin käyttää ohjelmassa valitulla nimellä, esimerkiksi doRedLamp. Tämänkaltainen muuttujien nimeäminen auttoi ongelmatilanteissa ja helpotti muutenkin ohjelman seuraamista.



Kuva 3. Painonappien ja valojen ohjaukselle asetettiin ehdot Sps.subiin.

Painonappien toimiessa suunnitellusti keskityttiin robottiin kuuluvan SafeOperation-moduulin käyttöönottoon. Valmistajan sivuilta löytyi opastusvideot, joiden perusteella päästiin alkuun ja loput asiat selvisivät kokeilemalla. Safety configuration -asetuksiin päästiin käsiksi, kun käyttäjäryhmäksi valittiin ”Safety maintenance technician”. Asetukset löytyivät, kun painettiin robotin kuvaa käsiohjaimesta ja valittiin sen jälkeen *Configuration -> Safety Configuration*.

Näyttöön avautui ensimmäinen välilehti, johon ei tarvinnut tehdä muutoksia. Alareunassa olevaa *Global parameters* -painiketta painamalla päästiin kuitenkin valikkoon, jossa turva-alueen tarkkailu päästiin laittamaan päälle.

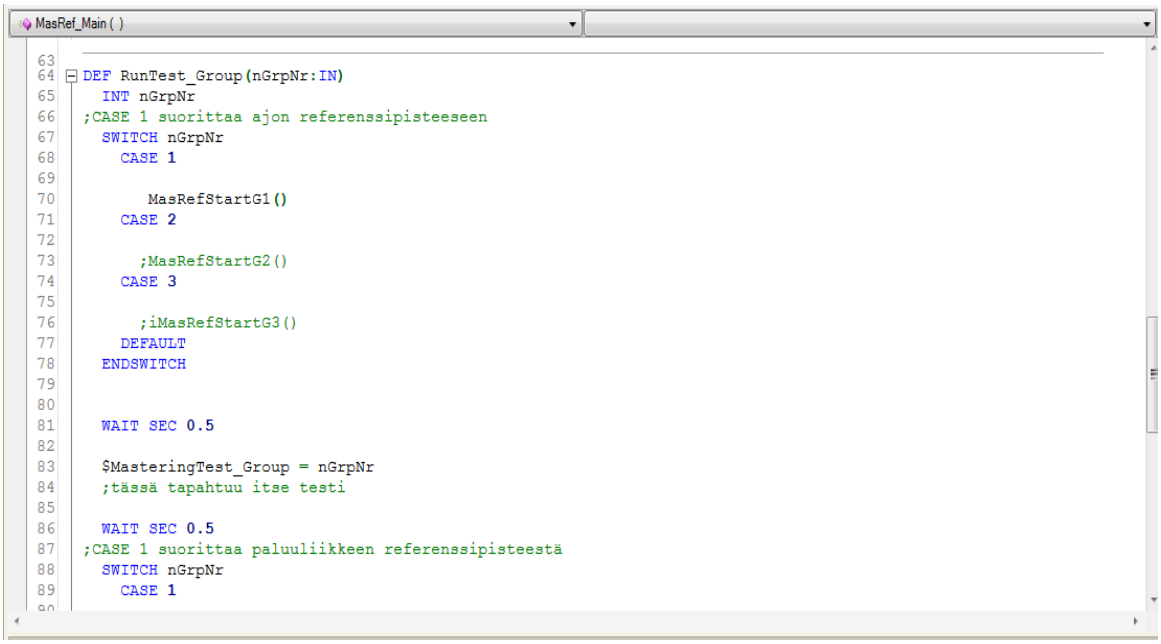
Seuraavalla välilehdellä avautui Cell configuration -ikkuna, jossa pystyi määrittämään virtuaalisen häkin, jonka sisällä robotti voi liikkua. Käytännössä tämä alue tulisi määrittää vastaamaan robotin ympärillä olevia aitoja, mutta tässä tilanteessa sellaisia ei ollut. Asetettiin siis alueen rajat sellaisiksi, että testien suorittaminen olisi mahdollista.

Seuraava välilehti oli nimeltään Monitoring spaces, ja tässä pystyi määrittämään aitojen sisään jääviä alueita joko koordinaatteihin perustuen tai akselikohtaisina rajoituksina sekä valitsemaan, olivatko ne sallittuja vai vältettäviä alueita. Sallituille alueille oli mahdollista myös asettaa rajoitettuja nopeuksia. Tällainen ominaisuus voisi sopia esimerkiksi tilanteisiin, joissa liikutaan seinien lähellä. Testiympäristössä robotin ohjauskaappi oli sijoitettu virtuaalisen häkin sisään ja yritettiin määrittää kaapin kohdalle vältettävä alue, mutta tämä osoittautui hankalaksi, sillä testejä varten jyrsimen oli liikuttava välillä aivan kaapin vieressä. Tämä ei kuitenkaan ollut olennainen asia toiminnan kannalta, joten siihen ei haluttu käyttää liikaa aikaa. Testejä ajaessa oltiin jatkossa erityisen varovaisia kaapin läheisyydessä toimiessa.

Viimeinen välilehti, johon tarvitsi tehdä muutoksia, oli nimeltään Reference position. SafeOperation-optioon kuului, että robotti kävi aika ajoin tarkistamassa oman sijaintinsa ajamalla työkaluun kiinnitetty magneetti hyvin lähelle pöydän alle asennettua reference switchiä. Kun switchin ledit sammuivat, oli magneetti tunnistusetäisyydellä ja tämä piste tallennettiin robotin referenssipisteeksi. Kyseistä pistettä käytettiin myöhemmin Mastering Reference Testiä ajaessa.

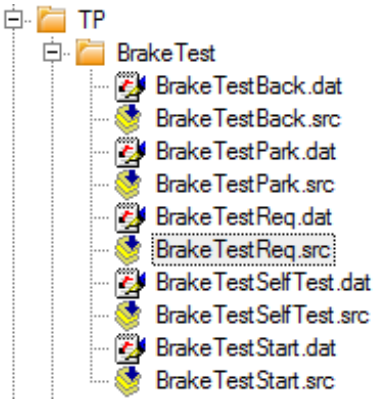
Mastering Reference Testin tutkimisessa meni melko pitkään, sillä tähän liittyviä ohjelmia oli useampia ja ohjelmakoodiakin tuhansia rivejä. Vanhojen ohjelmien avulla päästiin vähitellen perille siitä, mitä missäkin vaiheessa tapahtui. Lopulta huomattiin, että MasRef_Main- ja masrefuser-nimiset ohjelmat olivat kriittisiä toiminnan kannalta. MasRef_Main-ohjelmaan olisi mahdollista määrittää kolme akseliryhmää, joita tarkastellaan erikseen, mutta tässä kaikki akselit kuuluivat ryhmään 1. MasRef_Main ohjasi testin etenemistä ja kävi välillä ohjelmakutsujen avulla masrefuser-ohjelmassa määritellyissä pisteissä. MasRefStartG1 oli aliohjelma, jossa määritettiin polku referenssipisteeseen ja MasRefBackG1:ssä vastaavasti paluuliike kotiasemaan. Robotti tarkasti näiden välissä, oliko reference switchin aktivoiva magneetti tunnistustäisyydellä ja lähetti ohjelman päätteeksi käsiohjaimeen viestin testin onnistumisesta. Kuvassa 4 osa MasRef_Main-ohjelmasta, jossa tarkasteltavan akseliryhmän valinta oli toteutettu Switch-rakenteella.

Ohjelmakoodia voitiin kommentoida asettamalla puolipiste tekstin eteen. Tämä oli myös hyvä tapa poistaa käytöstä tarpeettomia komentoja, kuten MasRefStartG2-ohjelmakutsu.



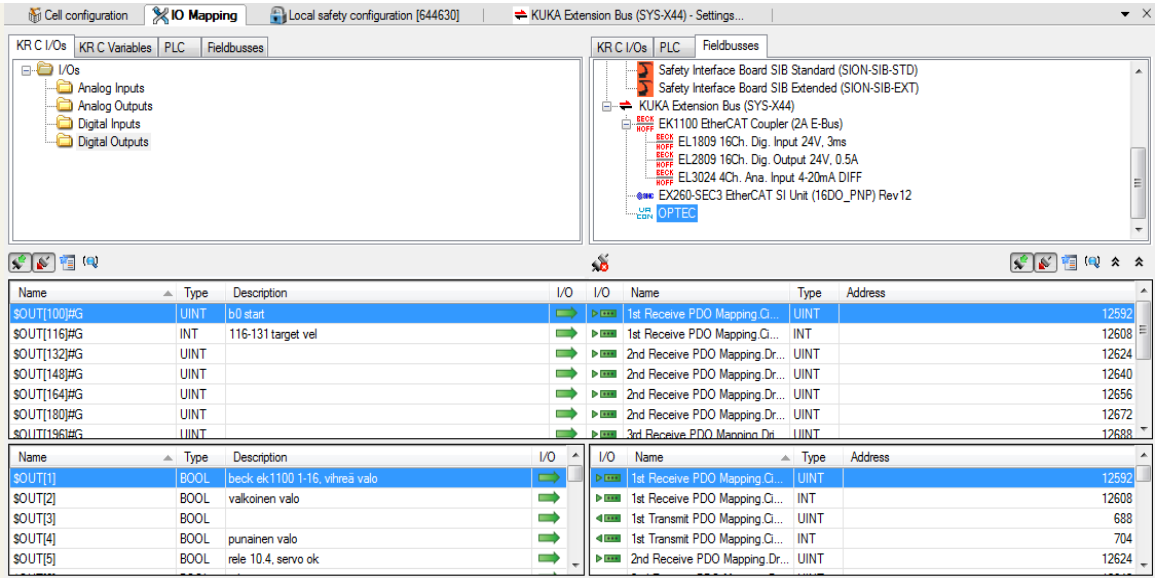
Kuva 4. Osa MasRef_Main-ohjelmasta.

Referenssitestin lisäksi SafeOperationiin kuului, että robotin täytyi varmistaa jarrujen toimivuus akselikohtaisesti tietyin väliajoin. BrakeTestiin liittyvät ohjelmat olivat valmistajan laatimia, ja ne löytyivät TP-kansiosta (Kuva 5). BrakeTestiin liittyviä ohjelmia oli paljon, mutta näistä ainoastaan BrakeTestReq-ohjelma tarvitsi kutsua pääohjelmassa. Tämä piti sisällään osan muista ohjelmista ja ehdot, milloin BrakeTest täytyi suorittaa. Jos nämä ehdot eivät täyttyneet, pääohjelmassa hypättiin testin yli. Muihin ohjelmiin ei ollut välttämätöntä koskea, mutta BrakeTestStart-ohjelmaan määritettiin testin aloituspaikka turvalliseen kohtaan. Jos testin aloituspaikkaa ei määritettäisi erikseen, robotti suorittaisi testin nykyisessä paikassaan. Ennen BrakeTestin suorittamista ensimmäistä kertaa oli tiedossa vain, että robotti kiihdyttää testin aikana täyteen nopeuteen ja tekee sen jälkeen äkkijarrutuksen, joten ei haluttu ottaa törmäysriskiä.



Kuva 5. Robotin BrakeTestiin liittyvät ohjelmat löytyivät TP-kansion alta.

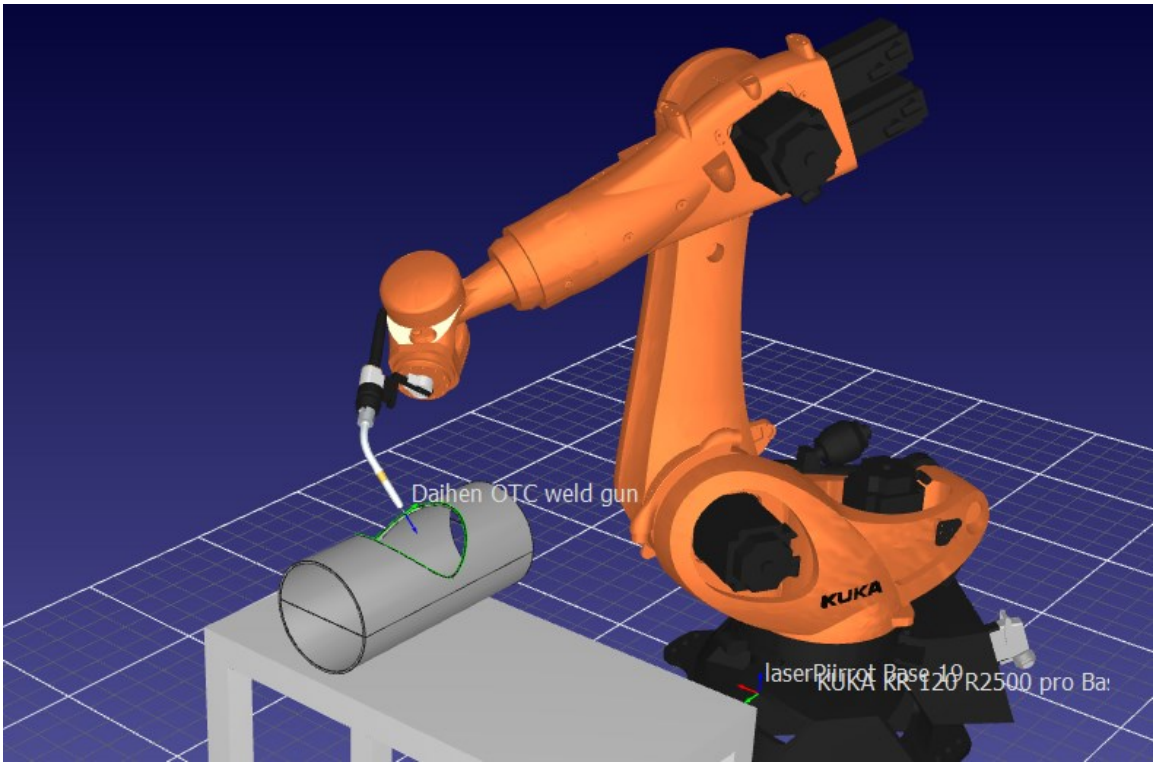
Tämän jälkeen piti saada kenttäväylä toimimaan oikein muutosten jälkeen. Aikaa kului lähinnä IO mappingin kuntoon saamisessa, tässä vaiheessa yhdistettiin robotin ja väylän signaalit vastaamaan toisiaan, jotta laitteiden välinen kommunikointi toimisi (Kuva 6). Lopulta jyrsimen ohjausta lukuun ottamatta väylä saatiin toimimaan suhteellisen kivuttomasti. Kun jyrsimen ohjaukseen liittyviin ongelmiin ei tässä vaiheessa tuntunut löytyvän ratkaisua, päätettiin jättää tämä asia vielä hautumaan ja keskityttiin työstöratojen luontiin RoboDK:lla.



Kuva 6. IO Mappingissa yhdistettiin kenttäväylään kuuluvien laitteiden signaalit vastaamaan robotin signaaleja.

3.4 Työstöradan hyödyntäminen ohjelmassa

Fusionissa mallinnettiin jyrsintäpöytä ja asetettiin se RoboDK-asemassa oikealle paikalleen robottiin nähden. Tarkat XYZ-arvot tälle saatiin selville pöydän koordinaatistoa määrittäessä. Ensimmäiset harjoitukset olivat satunnaisten liikekäskyjen luomista simulointiympäristössä ja pääohjelman luominen näitä käyttäen. Tästä eteenpäin keskityttiin jonkin aikaa lähinnä kappaleen reunojen seuraamiseen Curve follow projectia käyttämällä. Eräs yritys oli kiinnostunut putkien aukottamisesta, ja se toimitti muutamia putkiprofiileja, joita mallinnettiin testitarkoituksiin (Kuva 7). Leikkauspilliä ei ollut tässä vaiheessa saatavilla, mutta radan seuraamista pystyttiin testaamaan käytännössä, kun asetettiin robottiin kiinnitettävä piikki kulkemaan muutaman millimetrin päässä putkesta.



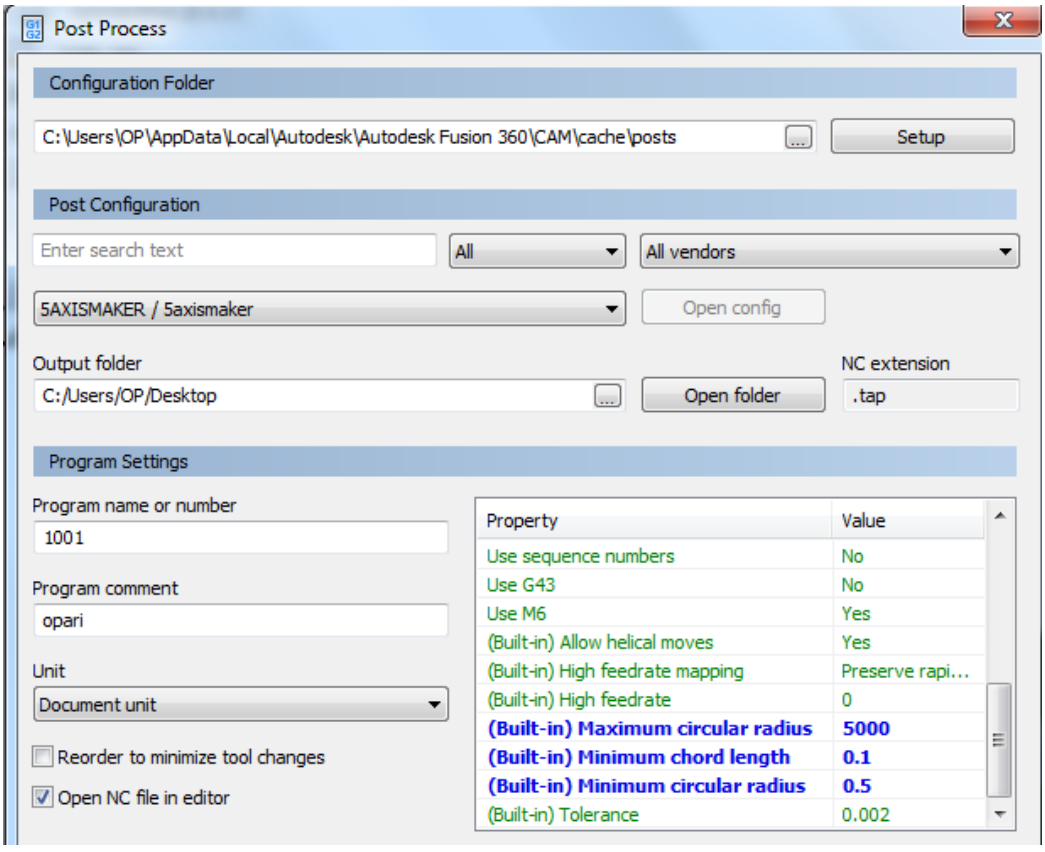
Kuva 7. Curve follow projectilla luotu työstörata, jossa hitsauspoltinta käytettiin radan testaamiseen.

Fusionin puolella mallinnettiin ensin yksinkertaisia kappaleita, kuten laatikoita ja johtokeloja ja opeteltiin löytämään oikeat asetukset, joilla robotti seurasi valittua rataa tarkasti simulaatiossa. Suurin ongelma oli aluksi se, että työkalun asento oli usein päinvastainen haluttuun nähden.

Hetken päästä huomattiin kuitenkin radalla näkyvien normaaliviivojen osoittavan työkalun asennon kussakin pisteessä ja näitä kääntämällä saatiin muutettua työkalun asentoa.

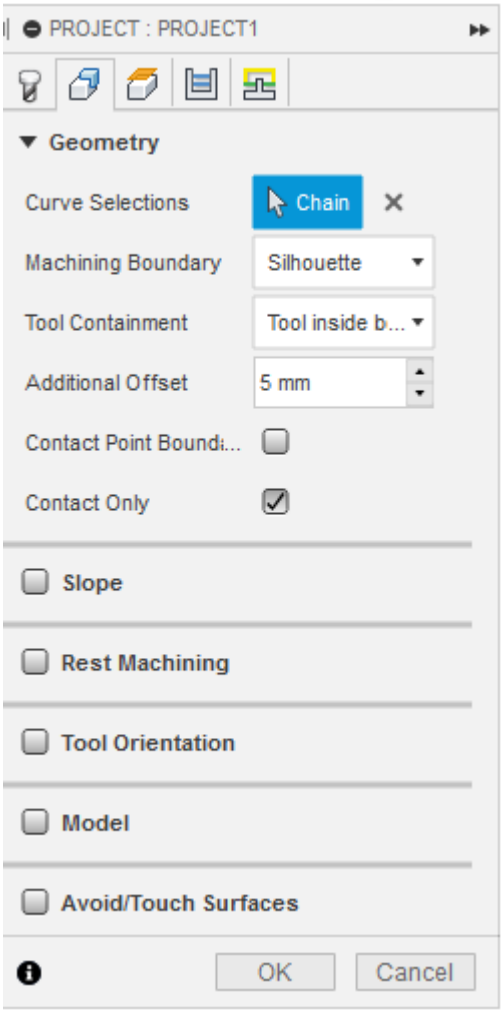
Radan tarkkuuteen ja toimivuuteen vaikuttivat lisäksi 3D-kappaleen tuonnin asetukset, jotka löytyivät *Tools -> Options -> CAD* -polkua seuraamalla. RoboDK:lla luotu rata koostui oletuksena lukuisista lineaariliikkeistä pisteiden välillä ja tämän vuoksi pisteiden tiheys nousi avainasemaan pyöreitä reunoja seuratessa. Ohjelmaa kääntäessä koodiksi huomattiin, että yhteen ohjelmaan mahtui vain noin 2500 lineaariliikettä. Jos liikekäskyjä oli radassa enemmän, RoboDK jakoi ohjelman automaattisesti osiin.

Pisteiden määrää saatiin myöhemmin vähennettyä viidesosaan, kun löydettiin tapa muuttaa osa lineaariliikkeistä kaariliikkeiksi Fusionissa. Tähän liittyvä ohjeistus löytyi Autodeskin sivuilta [5.]. Kaariliikkeet saatiin toimimaan erilaisia parametreja säätämällä, jotka on esitetty kuvassa 8. Näihin päästiin käsiksi, kun klikattiin hiiren oikealla radan kohdalla ja valittiin *Post Process.* Myös radan asetuksista Passes-välilehdeltä löytyviä tolerance- ja smoothing tolerance -arvoja säädettiin hieman alkuperäistä suuremmiksi.



Kuva 8. Fusion osasi korvata osan lineaariliikkeistä kaariliikkeillä, kun sinisellä näkyviä arvoja muutettiin.

Testien aikana teräkompensaatioon ei kiinnitetty juurikaan huomiota, mutta mittatarkkoja kappaleita jyrsittäessä on tärkeää, että työkalu kulkee oikealla puolella valittua reunaa. Jos kappaleet sisältävät vain suoria muotoja, niille voidaan tehdä työstörata RoboDK:ssa ja käyttää teräkompensaatioon Path to tool offset -toimintoa. Pyöreitä pintoja sisältävissä kappaleissa rata kannattaa puolestaan tehdä Fusionin puolella. Kompensaatio tulee usein automaattisesti, kun työstettävää geometriaa valitessa valitaan Tool containment -kohtaan *Tool inside/outside boundary* jyrsittävän puolen mukaan. Lisäksi Additional offset -kohtaan voi syöttää suuremman etäisyyden tarvittaessa. Kuvassa 9 Project-työkalulla luodun työstöradan asetuksia.

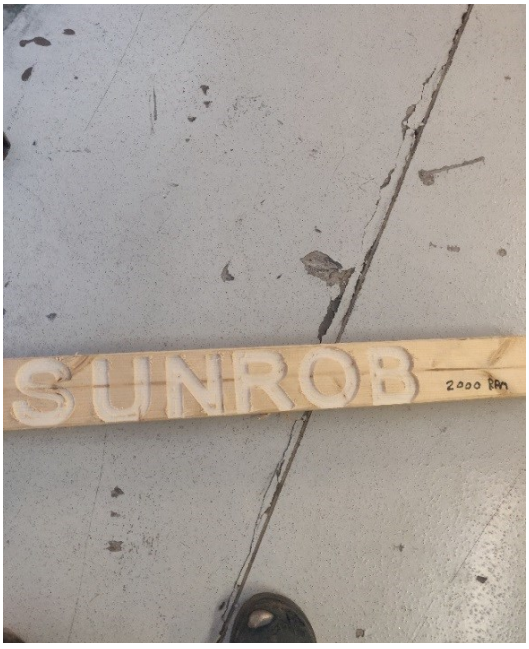


Kuva 9. Geometry-välilehdellä valittiin työstettävä rata ja työkalun sijainti rataan nähden.

3.5 Ohjelmien testaaminen käytännössä

Kun jyrsimeen liittyvät ongelmat saatiin selvitettyä, oli aika ajaa ensimmäinen testi. Testiä varten mallinnettiin hallista löytynyt lyhyt lankku ja luotiin tämän pintaan sketch, jota pystyttiin käyttämään ratana Fusionissa. Project-työkalulla saatiin luotua sopiva työstörata jyrsintään. Kalle Juntusen opinnäytetyössä [4.] oli tutkittu sopivia lastuamisarvoja robottijyrsintään, ja tätä käytettiin suuntaa antavana ohjeena. Ensimmäinen testi ajettiin kierrosnopeudella 2000 rpm ja hyvin hitaalla syöttönopeudella. Jyrsinnän jälkeä tärkeämpää tässä vaiheessa oli kuitenkin opetetun radan seuraaminen, ja tässä onnistuttiin hyvin.

Jyrsintesteissä keskityttiin jatkossa hahmottamaan karan kierrosnopeuden vaikutusta jyrsinnän tulokseen ja testejä ajettiin muutamalla eri kierrosnopeudella. Syöttönopeudesta ei saatu tilastoitua dataa, mutta huomattiin, että hidas syöttönopeus antoi usein paremman lopputuloksen kuin ”tuotantonopeudella” ajaminen. Testatuista pyörimisnopeuksista paras oli alueella 6000 - 7000 rpm, tätä korkeammalla nopeudella reunojen tikkuuntuminen oli runsasta ja alhaisilla nopeuksilla taas jyrsityn alueen pinta jäi karheaksi. Käytetty jyrsinterä oli spiraalimallinen. Tarkastelua hankaloitti testeihin käytettyjen lankkujen käyryys, jonka vuoksi uran syvyydessä saattoi olla muutaman millimetrin ero eri puolilla lankkua. Kuvissa 10 ja 11 ensimmäisten jyrsintätestien tuloksia.



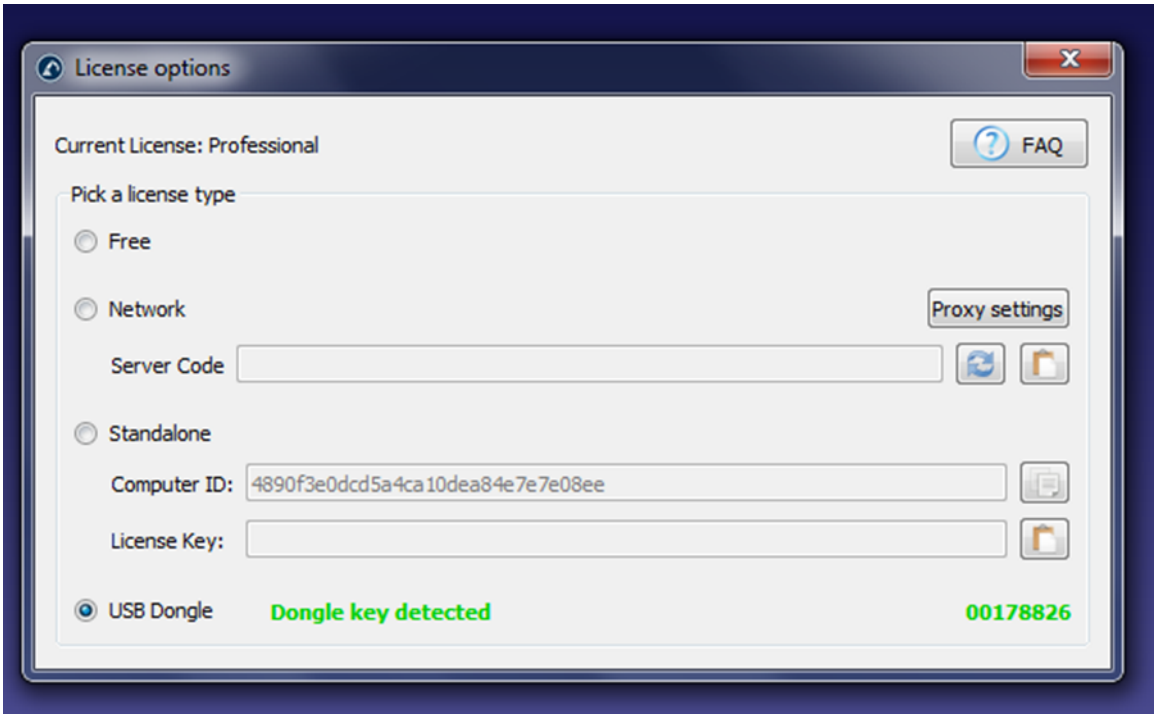
Kuva 10. Ensimmäinen jyrsintätesti 2000 rpm kierrosnopeudella ja hyvin hitaalla syöttönopeudella.

Kuva 11. Sama kuvio 10000 rpm kierrosnopeudella ja hieman suuremmalla syöttönopeudella.

4 Ohjeistus jyrsintäohjelman tekemiseen

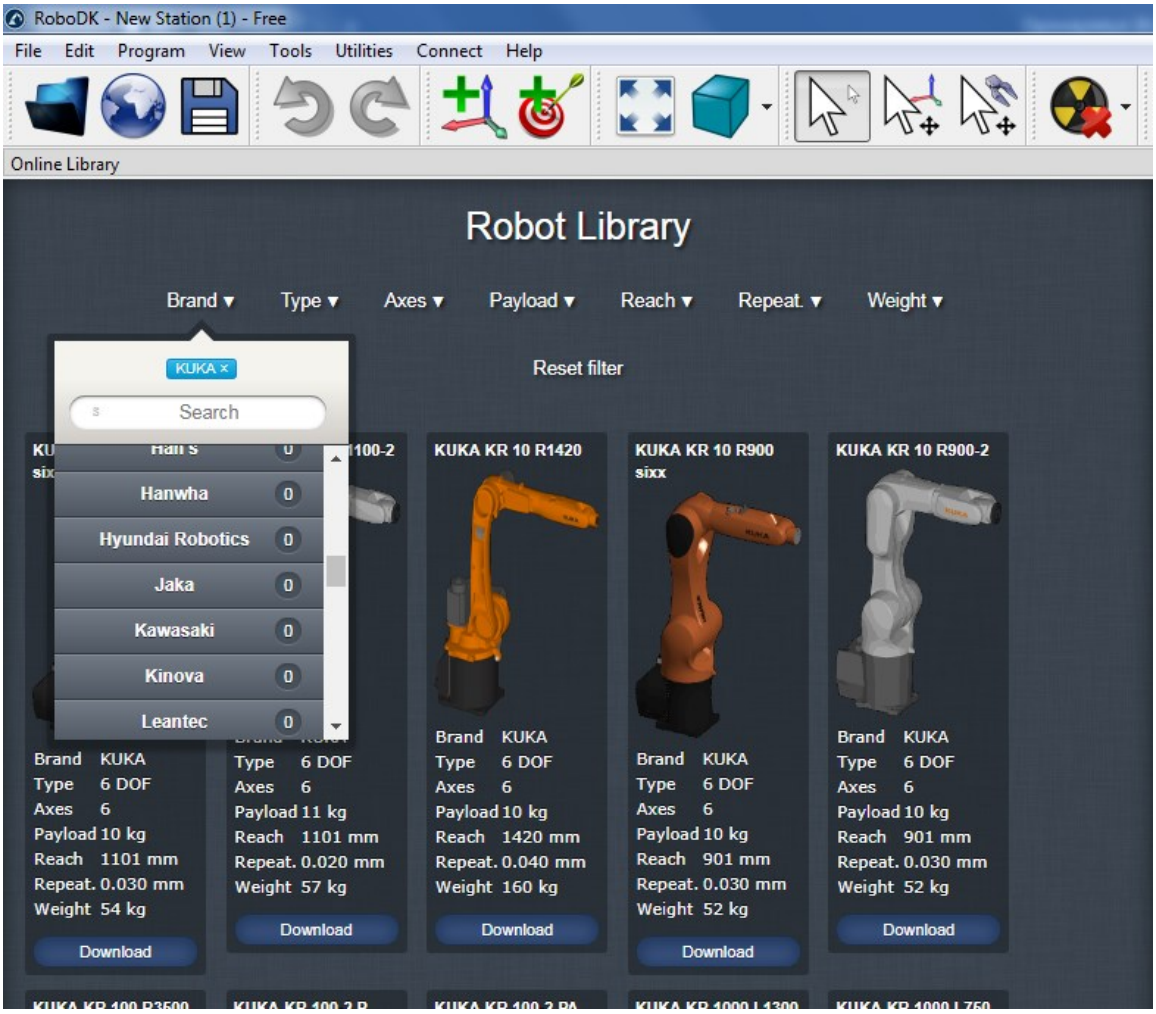
4.1 Alkuvalmistelut

Avaa RoboDK:ssa tyhjä asema ja tallenna se halutulla nimellä. Valitse yläpalkista *Help -> License* ja varmista, että lisenssityypiksi on valittu Network, Standalone tai USB Dongle käytössäolevan lisenssityypin mukaan (Kuva 12).



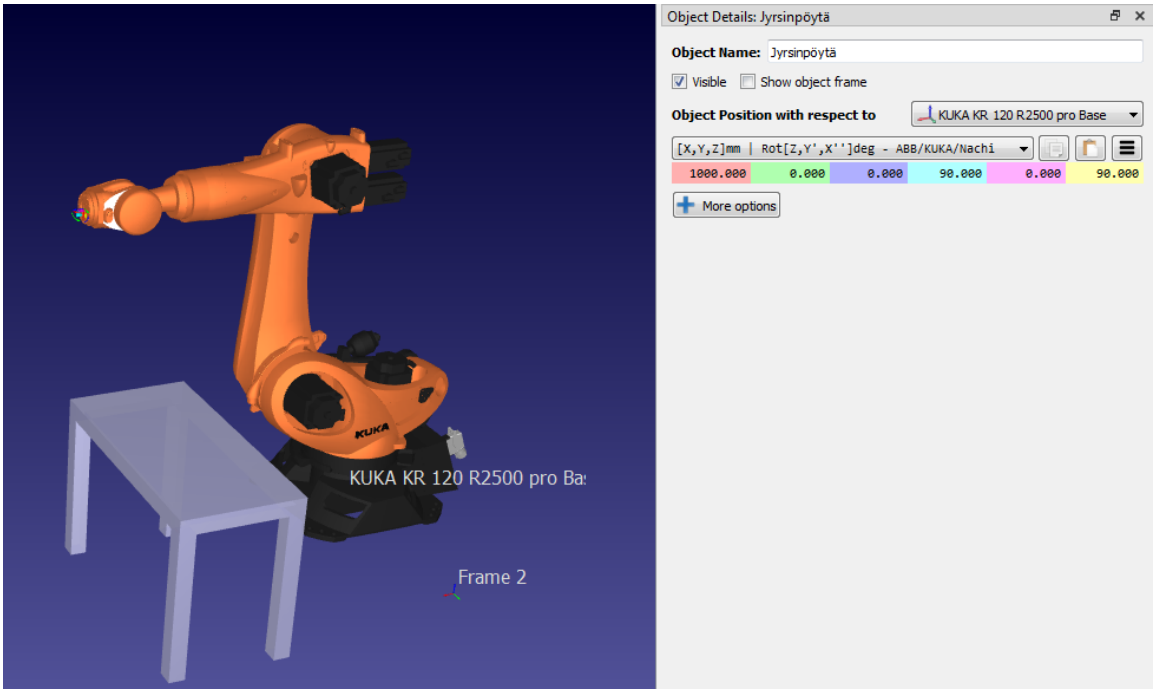
Kuva 12. Lisenssin valinta, tässä tapauksessa lisenssi on USB-tikulla.

Klikkaa *File -> Open online library* ja valitse käytettävä robotti, ikkunan yläreunassa olevat filterit helpottavat etsimistä (Kuva 13). Valitse tässä tapauksessa robotiksi KUKA KR120 R2500 pro, klikkaa *Download* ja robotti paikoittuu aseman origoon. Valitse työkalu samaan tapaan joko Online librarysta tai hakemalla työkalu tietokoneelta seuraamalla polkua *File -> Open...* ja valitsemalla työkalu. Käytä tässä tapauksessa itse luotua työkalua nimeltään piikki120mm, joka on tallennettu työpöydälle. Kyseinen työkalu on tallennettu .tool-muotoon, joten se kiinnittyy automaattisesti robotin käsivarteen.



Kuva 13. Robotin valinta online-kirjastosta.

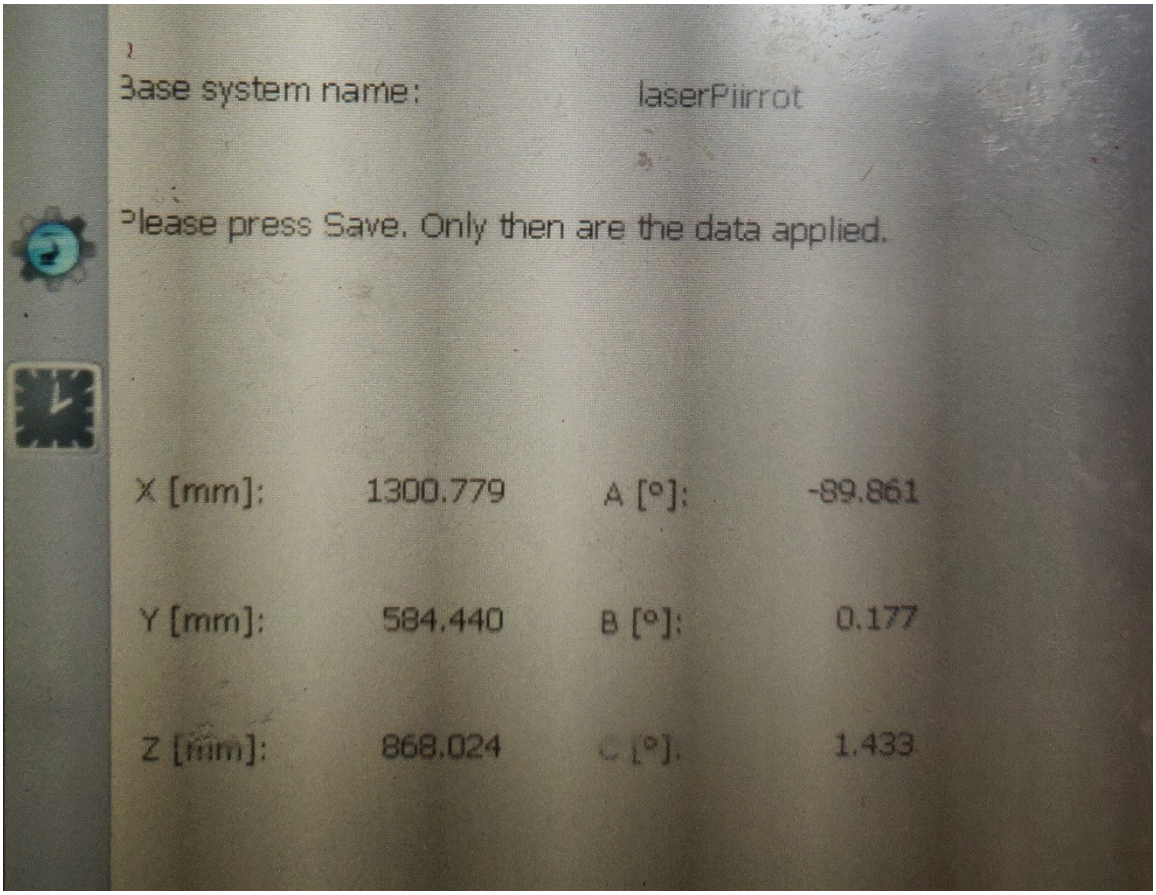
Tuo mallinnettu pöytä asemaan *File -> Open...*-polkua seuraamalla. Pöytä paikoittuu origoon, jonka jälkeen voit siirtää sen vapaaseen tilaan tuplaklikkaamalla pöytää ohjelmapuussa ja antamalla XYZ-arvoja avautuvaan ikkunaan. Luo uusi koordinaatisto pöytää varten klikkaamalla työkalurivillä *Add a Reference Frame* -painiketta, jonka jälkeen Frame 2 -niminen koordinaatisto paikottuu robotin viereen. Kuvassa 14 robotti on paikotettu origoon ja asemaan tuotu pöydän 3D-malli on siirretty vapaaseen tilaan, uuden koordinaatiston origo näkyy robotin vieressä.



Kuva 14. Robotti aseman origossa ja pöytä siirretty vapaaseen tilaan, tässä vaiheessa pöytä on vielä robotin jalustan koordinaatistossa.

Tuplaklikkaa Frame 2 -koordinaatistoa ohjelmapuussa, jolloin avautuvassa ikkunassa voit määrittää koordinaatiston paikan ja asennon robotin Base-koordinaatistoon nähden. Käännä koordinaatiston XYZ-suunnat vastaamaan oikean pöydän koordinaatistoa, tarkat kulmat tähän saat robotin käsiohjaimesta uutta koordinaatistoa luodessa. Myös pöydän koordinaatiston sijainti robotin Base-koordinaatistoon nähden näkyy käsiohjaimesta (Kuva 15). Uuden koordinaatiston kulmat eroavat jonkin verran vertailukoordinaatistosta, koska pöytä oli kiinni robotin jalustassa, joka ei ollut täysin suora. Kuvassa koordinaatiston nimi on eri kuin RoboDK:ssa, mutta sillä ei ole merkitystä toiminnan kannalta.

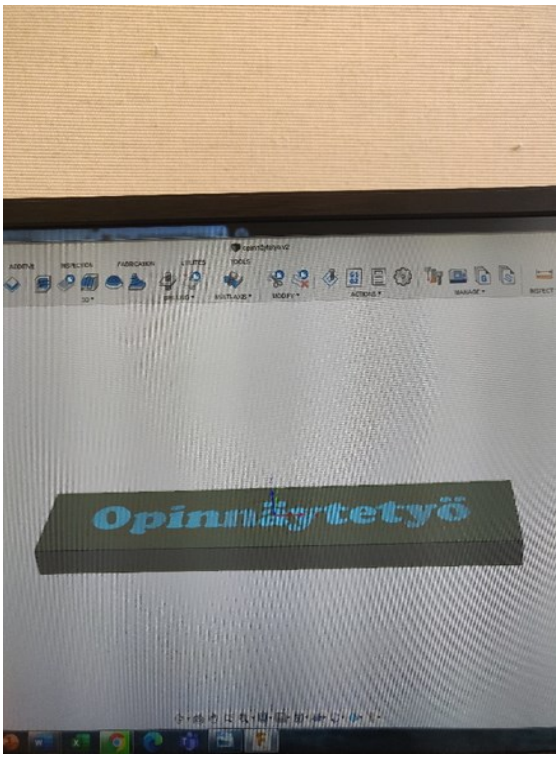
Raahaa tämän jälkeen pöytä tämän koordinaatiston alle ohjelmapuussa. Oikea koordinaatisto päivittyy näkyviin, kun object details-ikkuna suljetaan ja avataan uudestaan. Uuden koordinaatiston myötä pöytä siirtyy todennäköisesti kauas oikeasta paikasta, joten helpointa on asettaa kaikki koordinaattiarvot nollaan tässä vaiheessa. Nyt luotu koordinaatisto on 3D-mallin origossa ja voit asettaa sen oikealle paikalleen XYZ-arvoja muuttamalla. Kirjoita lopuksi työkalun ja koordinaatiston nimeen vielä halutut Tool- ja Base -numerot, esimerkiksi ”piikki120mm Tool 16”. Tämä varmistaa, että robottiohjelmassa käytetään oikeaa Tool- ja Base dataa eli robotti osaa määrittää liikkeet oikean TCP:n ja koordinaatiston mukaan. Nyt asema on luotu ja tätä vaihetta ei tarvitse enää toistaa ohjelmia tehdessä.



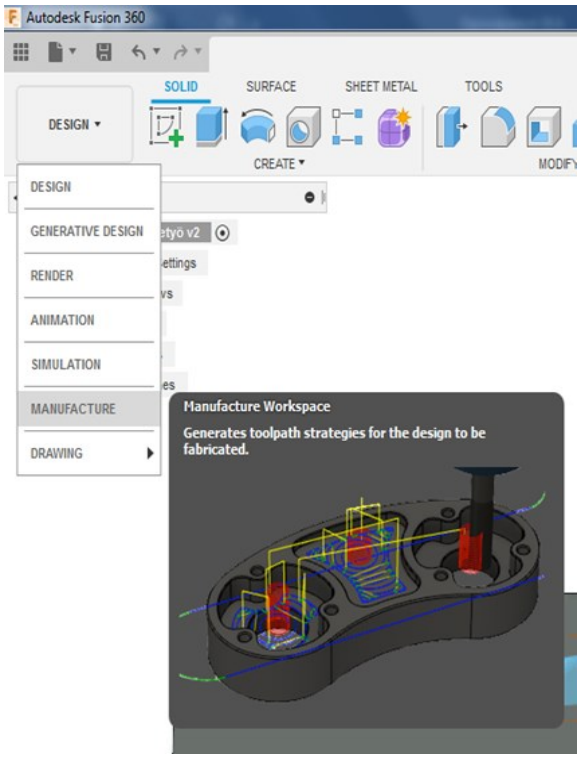
Kuva 15. Luodun koordinaatiston origon sijainti ja XYZ-suuntien kulmat Base-koordinaatiston vastaaviin nähden.

4.2 Ohjelman luominen

Fusioniin on ladattu tässä vaiheessa RoboDK Add-in Autodeskin sivuilta. Mallinna Fusionissa uusi kappale Top-tasolle ja luo sen pintaan sketch, jota käytetään jyrsittävänä ratana. Tämä voi olla esimerkiksi teksti (Kuva 16). Tallenna tämän jälkeen kappale pilveen painamalla *Save* ja luo STEP-malli työpöydälle (*File -> Export* ja valitse tyypiksi *STEP Files*), jotta sitä voidaan käyttää myöhemmin RoboDK:ssa. Siirry Fusionissa Manufacture-nimiseen workspaceen, joka on Fusionin CAM-osio (Kuva 17). Yläreunassa näkyy erilaisia CAM-työkaluja, näistä 3D-pudotusvalikon alta löytyy *Project*-työkalu, jolla voidaan luoda haluttu työstörata. Valitse tämä käyttöön, jolloin näytön oikeaan reunaan avautuu valikkoikkuna. Klikkaa Tool-kohdassa *Select* ja aukeaa työkalukirjasto, josta voit valita oikeaa työkalua vastaavan mallin. Tässä tapauksessa tällainen on vasemmassa reunassa Tutorial – Metric alta löytyvä 10mm flat end mill. Valitse työkalu tuplaklikkaamalla nimen kohdalla, ohjelmassa palataan tämän jälkeen valikkoikkunaan. Siirry seuraavaksi valikkoikkunassa Geometry-välilehdelle ja valitse sketch työstettäväksi radaksi klikkaamalla sitä. Paina *OK* valikkoikkunassa.

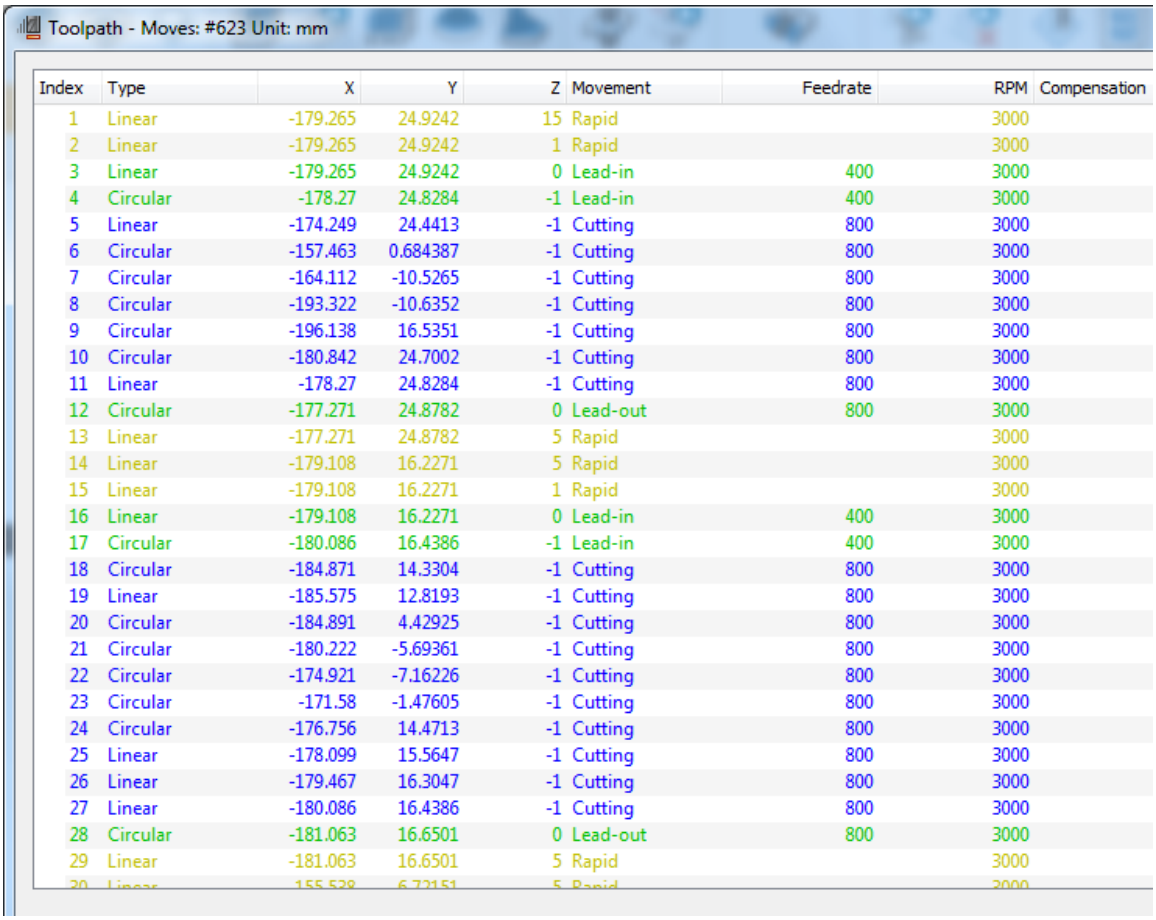


Kuva 16. Kappaleen pintaan luotu luonnos, jota käytetään työstöratana.



Kuva 17. Manufacture-tilassa voidaan luoda työstöratoja 3D-mallin muotoja tai sketchejä valitsemalla.

Ohjelmapuussa näkyy nyt Project-työstörata, jonka voit halutessasi simuloida painamalla hiiren oikealla ja *Simulate*. Klikkaa tämän jälkeen uudestaan hiiren oikealla radan päällä ja valitse *View Toolpath.* Näyttöön aukeaa ikkuna, jossa on eritelty kaikki rataan kuuluvat liikekäskyt (Kuva 18). Jos kaikki Cutting-liikkeet ovat listalla Linear-tyyppisiä, täytyy käydä muokkaamassa Post Processorin asetuksia opinnäytetyössä aiemmin esitetyllä tavalla.



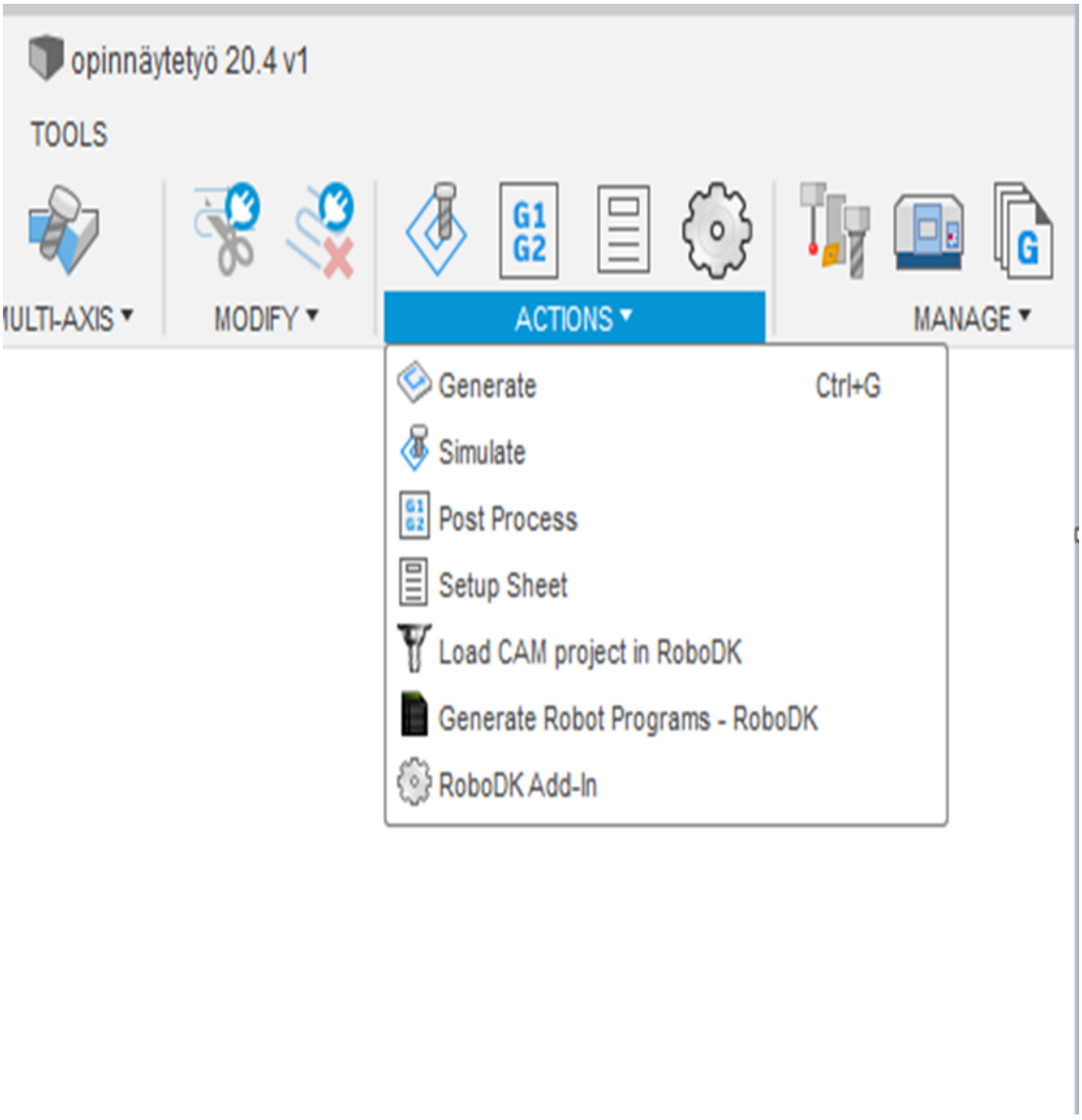
Toolpath - Moves: #623 Unit: mm

Index	Type	X	Y	Z	Movement	Feedrate	RPM	Compensation
1	Linear	-179.265	24.9242	15	Rapid		3000	
2	Linear	-179.265	24.9242	1	Rapid		3000	
3	Linear	-179.265	24.9242	0	Lead-in	400	3000	
4	Circular	-178.27	24.8284	-1	Lead-in	400	3000	
5	Linear	-174.249	24.4413	-1	Cutting	800	3000	
6	Circular	-157.463	0.684387	-1	Cutting	800	3000	
7	Circular	-164.112	-10.5265	-1	Cutting	800	3000	
8	Circular	-193.322	-10.6352	-1	Cutting	800	3000	
9	Circular	-196.138	16.5351	-1	Cutting	800	3000	
10	Circular	-180.842	24.7002	-1	Cutting	800	3000	
11	Linear	-178.27	24.8284	-1	Cutting	800	3000	
12	Circular	-177.271	24.8782	0	Lead-out	800	3000	
13	Linear	-177.271	24.8782	5	Rapid		3000	
14	Linear	-179.108	16.2271	5	Rapid		3000	
15	Linear	-179.108	16.2271	1	Rapid		3000	
16	Linear	-179.108	16.2271	0	Lead-in	400	3000	
17	Circular	-180.086	16.4386	-1	Lead-in	400	3000	
18	Circular	-184.871	14.3304	-1	Cutting	800	3000	
19	Linear	-185.575	12.8193	-1	Cutting	800	3000	
20	Circular	-184.891	4.42925	-1	Cutting	800	3000	
21	Circular	-180.222	-5.69361	-1	Cutting	800	3000	
22	Circular	-174.921	-7.16226	-1	Cutting	800	3000	
23	Circular	-171.58	-1.47605	-1	Cutting	800	3000	
24	Circular	-176.756	14.4713	-1	Cutting	800	3000	
25	Linear	-178.099	15.5647	-1	Cutting	800	3000	
26	Linear	-179.467	16.3047	-1	Cutting	800	3000	
27	Linear	-180.086	16.4386	-1	Cutting	800	3000	
28	Circular	-181.063	16.6501	0	Lead-out	800	3000	
29	Linear	-181.063	16.6501	5	Rapid		3000	

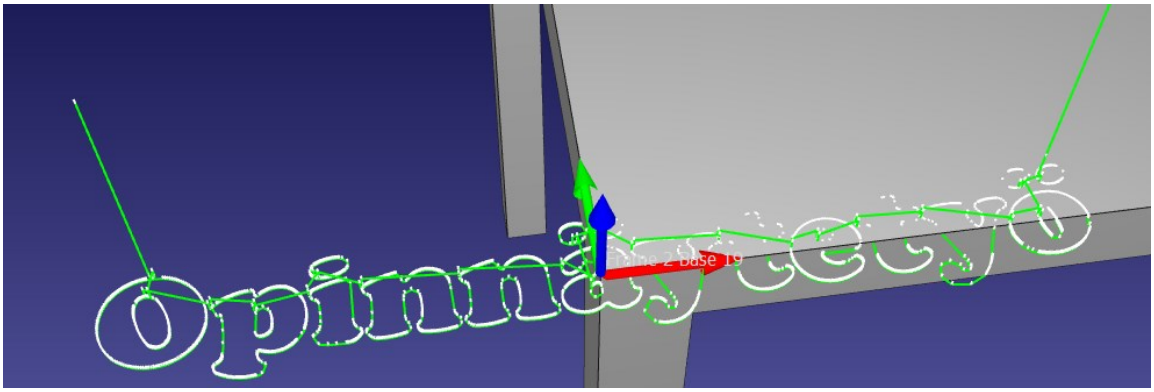
Kuva 18. Rataan kuuluvat liikekäskyt listattuna.

Avaa seuraavaksi taustalle aiemmin luotu RoboDK-asema ja klikkaa Fusionissa *Actions -> Load CAM project in RoboDK,* tämä on esitetty kuvassa 19. Rata siirtyy RoboDK:sa samaan paikkaan, kuin se oli Fusionissa, tässä tapauksessa origon lähelle. Siirretty rata näkyy kuvassa 20. Ympyränkaari-interpolaation vuoksi pisteiden määrä väheni alkuperäisestä yli 4000:sta noin 850:een.

Jos rata ei siirry RoboDK Add-Inin avulla, voit suorittaa saman vaiheen myös tallentamalla NC-ohjelman tietokoneelle ja valitsemalla tämän Curve follow projectiin radaksi (*Utilities -> Curve follow project ->* Path input-kohdassa *Select NC-file).*



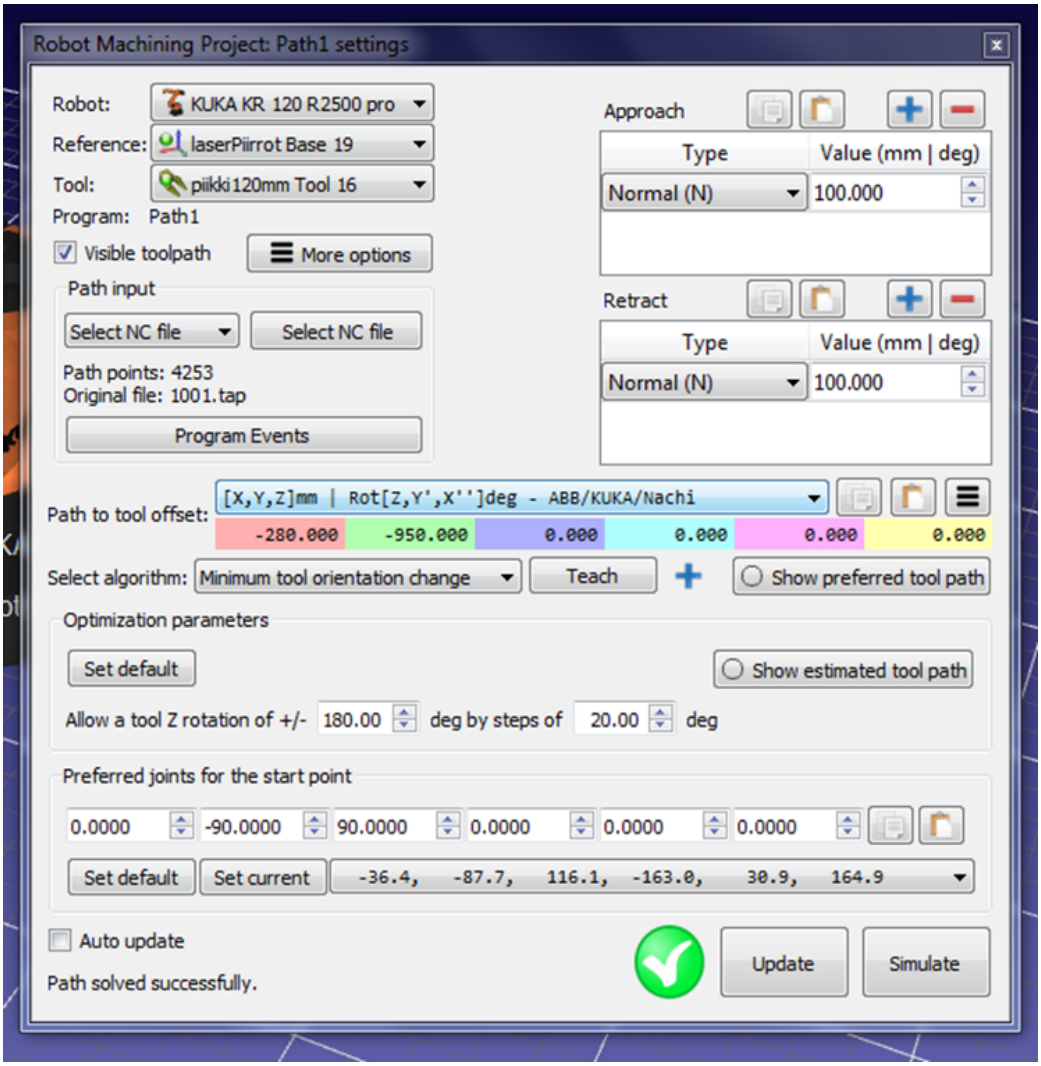
Kuva 19. RoboDK Add-inin toiminnot löytyvät pudotusvalikon alta. Valitaan näistä Load CAM project in RoboDK.



Kuva 20. RoboDK:n puolelle siirretty työstörata, valkoiset normaaliviivat osoittavat työkalun asennon kussakin pisteessä.

Tuo tässä vaiheessa mallinnettu kappale työpöydältä RoboDK:n puolelle *File -> Open...*- polkua seuraamalla. Kappale paikoittuu jälleen aluksi aseman origoon, joten raahaa se ohjelmapuussa Frame 2 -koordinaatiston alle. Nyt kappale on pöydän koordinaatistossa ja voit siirtää sen haluttuun kohtaan pöydällä. Ennen koordinaattiarvojen antamista varmista, että pudotusvalikosta on valittu ”ABB/KUKA/Nachi”. Työkappaletta siirrettäessä XYZ-suunnat määräytyvät pöydän koordinaatiston mukaan. Kun työkappale on hyvässä kohdassa, voit sulkea Object details -valikon.

Tuplaklikkaa seuraavaksi työstörataa ohjelmapuussa, jotta pääset käsiksi radan asetuksiin. Ikkunan keskeltä löytyy Path to tool offset -valikko, valitse tähän jälleen ”ABB/KUKA/Nachi”. Alla olevia koordinaatteja muuttamalla voit siirtää radan työkappaleen päälle, tässä vaiheessa koordinaatiston suunta määräytyy työkalun mukaan. Muutetut arvot tallentuvat painamalla *Update*. Radan päivittämiseen voi mennä hetki, jos rata koostuu suuresta määrästä pisteitä. Pisteiden määrä näkyy Path input -kohdassa. Päivitetyn radan pääset näkemään painamalla *Simulate*. Jos rata ei vielä osunut työkappaleen päälle, aseta uudet arvot ja toista aiemmat vaiheet. Kuvassa 21 Path settings -valikko, jossa Path to tool offset-kohtaan on asetettu oikeat koordinaattiarvot radan siirtämiseksi haluttuun paikkaan.

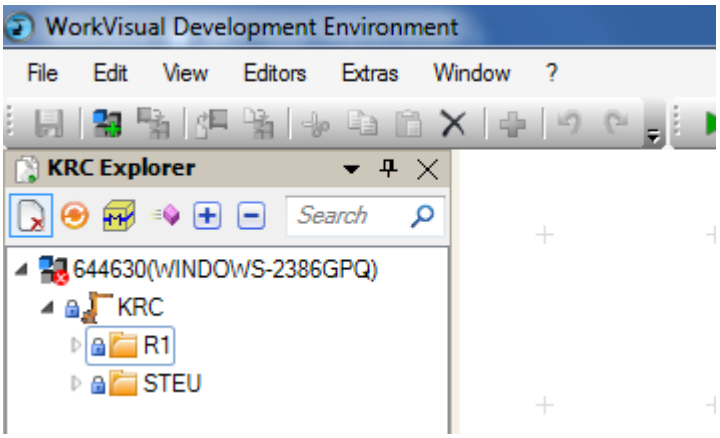


Kuva 21. Path settings -valikko, jossa pystyy muun muassa valitsemaan halutun NC-ohjelman työstöradaksi sekä asettamaan lähestymis- ja poistumisliikkeitä.

Z-arvon asettaminen offsetiksi on hieman hankalaa, sillä RoboDK:hon asetettu työkalukeskipiste eli TCP ei tässä tapauksessa vastaa täysin jyrsintapin pituutta. Tästä syystä korkeutta täytyy usein säätää vielä ensimmäisen testiajon jälkeen. Kun työstörata muuten kulkee työkappaleen päällä, se voidaan kääntää ohjelmakoodiksi. Tarkista vielä, että käytössä on oikea Post Processor. Tämä tapahtuu klikkaamalla rataa hiiren oikealla ohjelmapuussa ja valitsemalla *Select Post Processor*. Varmista, että valittuna on *KUKA KRC4*. Klikkaa tämän jälkeen radan kohdalla taas hiiren oikealla ja valitse *Generate robot program.* Koodi aukeaa uudessa ikkunassa, jossa nähdään radan koostuvan lukuisista lineaariliikkeistä radalla olevien pisteiden välillä. Tallenna ohjelma haluttuun paikkaan ja avaa WorkVisual, joka on KUKA:n oma ohjelmointisovellus.

4.3 Ohjelman siirto robotille

Tietokone on yhdistetty tässä vaiheessa Ethernet-kaapelilla robotin ohjauskeskukseen. Voit ladata robotilta käytössä olevan ohjelma tietokoneelle WorkVisualin *Browse*-toimintoa käyttämällä ja valitsemalla ohjelman, joka on aktiivinen. Yksittäisten ohjelmien siirron voit tehdä online-ohjelmointina valitsemalla Workspaceksi ”Programming and diagnosis”. Klikkaa näytön oikeassa reunassa olevasta Cells treestä tietokoneen nimeä, jolloin vasempaan reunaan KRC Explorer -ikkunaan aukeaa samanniminen kohta (Kuva 22). Klikkaile kansiorakennetta auki *KRC -> R1 -> Program*, kunnes pääset Program-kansion sisään. Kopioi seuraavaksi tietokoneelta leikepöydälle tallennettu jyrsintäohjelma, joka on muotoa .src. Siirry WorkVisualissa Program-kansion päälle ja liitä ohjelma kansioon. Ohjelma on heti käytettävissä robotilla.



Kuva 22. KRC Explorer -ikkuna, josta päästään tekemään ohjelmiin muutoksia online-tilassa.

5 Yhteenveto

Työn keskeinen tavoite oli tehokkaan työmenetelmän valitseminen robottijyrsinnän suorittamiseksi ja tarpeellisten dokumenttien laatiminen. Jyrsintään liittyvien asioiden lisäksi myös putkien aukotukseen perehdyttiin yrityksiltä tulleiden kyselyiden myötä. Todettiin Fusionin ja RoboDK:n sopivan tämänkaltaiseen työskentelyyn, vaikka esimerkiksi putkien aukotukseen liittyen olisi hyvä olla lisäksi jonkinlainen nestausohjelma, jos erilaisia variaatioita on useita. Testien myötä tultiin siihen tulokseen, että lähes kaikki työstöradat kannattaa luoda Fusionin puolella ja kääntää sen jälkeen RoboDK:n puolelle, jossa rata voidaan asettaa oikealle paikalleen ja varmistaa robotin asento työn aikana.

Yrityksellä ollut demolaitteisto mahdollisti runsaan testaamisen myös käytännössä ja tätä kautta saatiin huomioitua asioita, jotka eivät tulleet simulointien aikana esille. Yksi tällaisista oli se, että RoboDK:ssa kotiasemassa ollessaan robotin akseli 4 oli 180 astetta väärinpäin, mikä aiheutti yllättäviä liikkeitä, kun ohjelmaa testattiin oikealla robotilla. Jatkossa täytyy siis varmistaa akselien asennot ennen työn aloittamista. Myös karan pyörimisnopeuden vaikutuksesta jyrsinnän lopputulokseen saatiin tietoa, vaikka testit eivät olleetkaan täysin vertailukelpoisia.

Tämänhetkisen tiedon mukaan paras ratkaisu on luoda robotille vakituinen pääohjelma, jossa ohjataan karan pyörimistä ja jäähdytystä sekä kutsua tässä ohjelmassa työstettävää rataa aliohjelmakutsuna. Näin varmistutaan, että toimilaitteiden ohjaukset toimivat oikein. Pääohjelman voi tehdä kokonaisuudessaan RoboDK:ssa tai robotilla. Työstörataa tehdessä Fusionin puolella on syytä käydä muokkaamassa post processorin asetuksia, jotta ympyränkaari-interpoloinnin saa käyttöön. Tämä vähentää huomattavasti rataan vaadittavia pisteitä ja keventää sitä kautta ohjelmaa.

Oppimistavoitteiksi oli ennen työn alkua asetettu osaamisen kehittyminen robotiikan parissa ja simulaatiosovelluksiin tutustuminen. Työn aikana osaaminen näiden asioiden parissa kehittyi huomattavasti ja tavoitteisiin päästiin.

Lähteet

[1] Industrial Robots. https://ifr.org/industrial-robots

[2] Kr 210 R2700 Prime Kuka Robots. https://www.indiamart.com/proddetail/kr-210-r2700-prime-kuka-robots-21044129833.html

[3] Jyrsintä. https://www.ssab.fi/services/

[4] Kalle Juntunen. Lastuamisarvojen määritys robotilla tehtävälle puun jyrsinnälle, Kajaanin ammattikorkeakoulu; 2008

[5] Circular toolpaths are linearized and have no G02 or G03 arc moves in Fusion 360. https://knowledge.autodesk.com