

TALLINNA TEHNIKAÜLIKOOL
Elektrotehnika instituut

ATR70LT

Juri Kizikov

TÖÖSTUSROBOTITE PROGRAMMEERIMINE TARKVARAGA SIEMENS NX

Magistritöö

Instituudi direktor prof. Tõnu Lehtla

Juhendaja ins. Margus Müür

Konsultant prof. Tõnu Lehtla

Lõpetaja Juri Kizikov

Tallinn 2015

AUTORIDEKLARATSIOON

Kinnitan, et käesolev lõputöö on minu iseseisva töö tulemus. Kõik selle koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Varem ei ole selle alusel kutse- ega teaduskraadi ega inseneridiplomit taotletud. Töö on koostatud litsentseeritud tarkvara abil.

Tallinn, 4.06.2015.a.

..... Juri Kizikov

ATV70LT

Tööstusrobotite programmeerimine tarkvaraga Siemens NX

Juri Kizikov, üliõpilaskood 122300AAAMM, mai 2015. – 57 lk.

TALLINNA TEHNIKAÜLIKOOL

Energeetikateaduskond

Elektrotehnika instituut, robotitehnika õppetool

Töö juhendaja: insener Margus Müür

Töö konsultant: professor Tõnu Lehtla

Võtmesõnad: Robot, post protsessor, ABB, Motoman, nurkkoordinaadid, XYZ-koordinaadid, Siemens NX, keevitus, programmi automaatne genereerimine

Referaat:

Lõputöö on 57 lehel, sisaldab 4 tabelit, 22 illustratsiooni.

Lõputöö jooksul sai uuritud võimalus programmeerida 6-teljest robotit ABB IRB1600 1,45m ja Motoman UP6 näitel kasutades kombineeritud CAD/CAE/CAM-tarkvara Siemens NX. Siemens NX on oma klassi üks levinum tarkvara tänu oma laia funktsionaalsusele. Mõned firmad, kellel on robotiseeritud keevitus, plasma- ja laserlõikus, ütlevad, et roboti ettevalmistus tööks võtab palju aega. Detaili kuju muutmisel, tuleb luua uut roboti juhtimisprogrammi detaili töötlemiseks. Antud lõputöös loodi post protsessorit Siemens NX-ile, et saaks automaatselt genereerida juhtimisprogrammi robotile vastavalt detaili muudatustele. Põhilised probleemid, mis said lahendatud olid ABB robotite jaoks kvaternionide arvutus. Kuna robotid on tavaliselt 6-teljised, aga Siemens NX võimaldab genereerida radasid 5-teljeste masinate jaoks, siis otsustati leiti kuuenda telje asukoha liikumisgradienti abil. Pakutud lahenduse nõrk koht on arvutusviivitus. Lisaks uuriti võimalust arvutada roboti iga liigendite pöördnukrasid. Antud lahendus sai realiseeritud ainult teoreetilisel kujul kuna vajab optimeerimist ja seadistamist iga konkreetse roboti jaoks. Kokkuvõtteks võib öelda, et robotit saab programmeerida kasutades teisi tarkvarasid, mis ei ole otsaliselt loodud robotite programmeerimiseks. CAD/CAE/CAM-tarkvara kasutamine annab võimalust automaatselt luua juhtimisprogrammi, aga sobib ainult robotitele, mis kasutavad tööriistana keevitus-, plasma-, vesi-, laser-, või freespea.

ATV70LT

Программирование промышленных роботов с помощью САПР Siemens NX

Юрий Кизиков, код студента 122300АААММ, май 2015. – 57 стр.

ТАЛЛИНСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ * Энергетический факультет

Электротехнический институт

Кафедра робототехники

Руководитель работы: инженер Маргус Мюур

Консультант работы: профессор Тыну Лехтла

Ключевые слова: Робот, пост процессор, АВВ, Motoman, угловые координаты, XYZ-координаты, Siemens NX

Реферат:

Дипломная работа состоит из 57 страниц, 4 таблиц и 22 иллюстрации.

В данной работе рассматривается проблема, связанная с большими затратами времени на введения новой продукции для роботизированной ячейки механической обработки.

Рассматривается возможность создания УП для робота непосредственно в САПР Siemens NX. Данная необходимость вызвана временными затратами при прототипировании и запуска продукции в производство. К примеру, с изменением конструкции кондуктора или самой детали при использовании стандартных решений необходимо заного переписывать УП для робота с самого начала. Если же использовать САПР в котором шла разработка детали и кондуктора, каковым в данном случае является NX, то необходимости заново создания управляющей программы нет, т.к. САПР пересчитает траектории автоматически. В данной работе были найдены решения по расчёту кватернионов, необходимых для программирования роботов марки АВВ, а так же было предложено решение об управлении 6-ой координаты. Т.к. NX способен генерировать траектории с 5 осями одновременно, то 6-ая ось вычисляется дифференциально. Подитоживая можно сказать, что данное решение подходит для фирм, занимающихся преимущественно прототипированием. Учитывая ограничения САПР, данное решение подходит для роботов оснащённых режущими головками или сварочным пистолетом.

ATV70LT

Programming industrial robots with Siemens NX software

Juri Kizikov, student code 122300AAAMM, May 2015. – 57 pages

TALLINN UNIVERSITY OF TECHNOLOGY * Faculty of Power Engineering

Department of Electrical Engineering

Chair of Robotics

Tutor of the work: engineer Margus Müür

Consultant: prof. Tõnu Lehtla

Key words: Robot, post-processor, ABB, Motoman, angle coordinates, XYZ-coordinates, Siemens NX

Summary:

Work contains 57 pages, 4 tables and 22 illustrations.

During this work it was created the solution for programming robots using integrated CAD/CAE/CAM-software Siemens NX. Siemens NX is complex integrated solution for industrial needs. NX allows to create 3D design, modelling, simulation and CNC-program post processing. Many companies who use robotized cells for welding and multi-axis cutting encounter with problems such as low speed of preparation for work. Sometimes the complicated program should be rewritten from very beginning even due to minor modification of the fixture or the part itself. In these cases optimal solution is to use the same software to design the part and generate the program. Siemens NX is one of several combined solutions. But it does not directly suitable for programming robots. In this work the way how to create the post processor for programming industrial robots is described. Main point what was solved is rotation angle transition to quaternions for ABB IRB controllers. Another problem was to override the limited number of supported axes in Siemens NX. It is possible to generate only 5-axis simultaneous movement using this software. But robots have 6 axes. So the last 6th axis calculation is made according to movement gradient. This solution has a drawback. It calculates the 6th angle with 1 step latency. And the last complicated part is inverse kinematics for robots. This means, that knowing the tool position it is needed to calculate joint angles. This calculation was presented purely as a theory. Due to optimization of this process for each exact robot the better way to let the controller recalculate joint angles by itself. In this work it was proved, that it is possible to create programs for robots using integrated software Siemens NX.

SISUKORD

Magistritöö ülesanne -----	8
1. Eelsõna -----	9
2. Sissejuhatus-----	10
3. Siemens NX -----	13
3.1. CAD-moodul-----	14
3.2. CAE-moodul -----	15
3.3 CAM-moodul -----	16
4. ABB IRB1600 (1.45m) -----	19
4.1 Tööriista ja tööobjekti etteantud parameetrid -----	19
4.2 Punkti koordinaadi arvutus -----	21
4.3 Trendline -----	27
4.4 Liikumiskäsud ja programmi väljastamine-----	31
4.5 Lisafunktsioonid-----	34
4.6 Juhtimisprogrammi väljastamine -----	37
4.7 Virtuaalne simulatsioon ja katsetamine -----	38
5. Yakisawa Motoman roboti programmeerimine -----	42
5.1 XYZ-koordinaadid -----	43
5.2 Nurkkoordinaadid -----	45
5.3 Liikumise interpoleerimine -----	51
6. Kokkuvõtte -----	54
Kasutatud kirjandus -----	56
Lisad -----	58
Lisa 1. ABB IRB1600 roboti spetsifikatsioon-----	59
Lisa 2. Yakisawa Motoman UP6 spetsifikatsioon. -----	60
Lisa 3. Intervjuu robotite müügiesindajaga, OMD OÜ -----	61
Lisa 4. Intervjuu robotkeevituse tööde juhiga, BLTR Masinaehitus -----	63

Lisa 5. Koordinaadi arvutus Siemens NX post protsessoris-----	64
Lisa 6. Trendline-----	66
Lisa 7. Lineaarne liikumine -----	67
Lisa 8. Juhtimisprogrammi väljastamisalgoritm -----	68
Lisa 9. DH-matriksi elemendid-----	69
Lisa 10. T-tähe markeerimine, Motoman NX100 kontrolleri juhtimisprogramm -----	70
Lisa 11. T-tähe markeerimine, ABB IRB kontrolleri juhtimisprogramm -----	71

TALLINNA TEHNIKAÜLIKOOL
Elektrotehnika instituut

KOOSKÕLASTATUD

Prof. T. Lehtla.....

..... 2015

MAGISTRITÖÖ ÜLESANNE

Juri Kizikov, üliõpilaskood 122300AAAMM

Magistritöö teema: Roboti juhtimisprogrammi genereerimine tarkvara Siemens NX abil

Ülesanne: Uurida võimalust programmeerida roboteid Siemens NX CAD/CAE/CAM tarkvara abil, luua algoritm ehk post protsessor roboti juhtimisprogrammide loomiseks.

Lähteandmed:

1. Robotid ABB IRB1600 ja Yakisawa Motoman UP6
2. Robotite kasutusjuhendid
3. Tarkvara Siemens NX kasutusjuhend
4. Artiklid roboti juhtimisprogrammi loomisest CAD/CAE/CAM tarkvarade abil
5. Artiklid robotite post protsessori loomiseks CAD/CAE/CAM tarkvarades

Lahendamisele kuuluvate probleemide loetelu:

Luu roboti post protsessor Siemens NX-i tarkvaras, et automaatselt genereerida juhtimisprogramme järgmiste parameetridega:

1. Eralda punktide väärtused ja liikumiskäskud gruppideks.
2. Realiseerida roboti liigendite, kaar- ning lineaarse liikumisinterpoleerimine.
3. Võimaldada kasutada lisafunktsioone tööriista juhtimiseks.
4. Leida lahendus arvutada roboti 6. telje asend.
5. Luua 3D-mudel ülikooli oleva roboti tööriistast.

Magistritöö esitada eesti keeles kahes eksemplaris koos eesti- ja kahe võõrkeelse referaadiga ning CD plaadil pdf failina hiljemalt 8.06.2015.

Juhendaja:

Ülesande vastu võtnud:

Insener Margus Müür

Üliõpilane J. Kizikov

1. EELSÕNA

Täna oma juhendajat Magus Müüri, kes juhendas mind lõputöö kirjutamise jooksul, andis kasulikke nõuandeid ning aitas mitmekülgsetelt keerulistelt probleemidelt lahendamisel. Täna ka Tallinna Tehnikaülikooli Energeetika teaduskonna juhendajat Tõnu Lehtlat ning teisi teaduskonna suurepäraseid õppejõudusid, kes andsid teadmisi antud alal. Kindlasti tuleb tänada Andrei Maksimovi, kes on OMD OÜ firma omanik. Tal on suured kogemused automatiseerimisalal. Andrei kirjeldas probleeme, millega tema kliendid tihti kokku puutuvad. Kindlasti tänan Valeri Kizikovi, kes on peatehnoloog BLTR Masinaehitus AS-il. Tema viitas probleemidele, millega tegelevad temale alluvad insenerid. Roboteid rakendatakse BLTR Masinaehitus AS-s keevitamisel. Täna ka Indrek Kioleini, kes andis Siemens NX-i demo litsentsi.

2. SISSEJUHATUS

Paljud ettevõtted kasutavad roboteid keevitus-, frees- ning automatiseeritud painutusmoodulites töötegemiseks. Tänu robotitele on muutunud tootmine efektiivsemaks ja kiiremaks. Reeglina teevad robotid selliseid töid, kus detailide kogus on väga suur. Roboti seadistamine tööks vajab väga palju ressursse: rakiste valmistamine, juhtimisprogrammi kirjutamine ning proovide tegemine jne.

Tootmise testimise käigus peale seadistamist vaadatakse, kas terve tootmisprotsess on stabiilne ning lõpptulemus (valmis detail) vastab tehnilistele nõuetele. Kui tuleb välja, et roboti juhtimisprogramm vajab muutmist või toodangule tehakse pisitäiendusi, siis tuleb luua uus rakis ning täiendada või kirjutada uus juhtimisprogramm. Juhtimisprogrammi koostamise lihtsustamiseks kasutatakse ka spetsiaalseid tarkvarasid, mis võimaldavad genereerida roboti juhtimisprogrammi automaatselt. Selle peab insener üle vaatama ja kontrollima, et roboti asend mingis punktis ei ole vale. Kuna detaili mudelit koostatakse tavaliselt *CAD (Computer-aided Design)*-tarkvarades, mis ei edasta infot tehtud muudatuste kohta roboti programmeerimise tarkvarale. Seetõttu tuleb kogu roboti juhtimisprogramm uuesti genereerida (mitte ainult parandus muudetud osa kohta), kuna töödeldava detaili modelleerimine ja roboti juhtimisprogrammi loomine toimub erinevates tarkvarades. Seda annab vältida, kui töödeldava detaili modelleerimine ja roboti juhtimisprogrammi loomine teha ühes tarkvaras.

Selle probleemi (detaili muutuste info kadumine) ületamiseks kasutavad mõned suured firmad nagu GM Opel GmbH (Saksamaa), OKB Sukhoi (Venemaa) kombineeritud CAD/CAE/CAM-tarkvara (Computer-aided Engineering) nagu Siemens NX. Siemens NX-i kasutatakse ka Eestis – Zircon Group OÜ, Norma AS, Eesti Energia AS ja teised.

Siemens NX võimaldab luua detaili mudeli ning koostu, teha füüsilisi katseid, luua juhtimisprogramme CNC-pinkidele ning vajaliku dokumentatsiooni nagu joonised lõpptoodangu koostamiseks, keevitamiseks, detailide valmistamiseks ning teiste protsesside jaoks. Selle tarkvara eeliseks on võimalus automaatselt kanda üle muudatusi ühest moodulist teise. Teiste sõnadega, kui detaili kuju on muudetud CAD-moodulis, siis CAM-moodul kasutab juba uuendatud kujuga detaili ning arvutab automaatselt ümber tööoperatsioonid. See kiirendab uute detailide projekteerimist ning prototüüpide valmistamist.

Siemens NX-i CAM-moodul annab võimalust genereerida ainult 5-teljelist sünkroonset liikumist. Enamus tööstuses kasutatavaid roboteid on vähemalt 6 liikumisteljega, mistõttu on vaja täiendada NX-i CAM-moodulit, et realiseerida täiusliku roboti potentsiaali.

Antud lõputöös on loodud post protsessor lisa funktsioonidega, mis võimaldab rakendada roboti 6. telge, lisa mootoreid ning roboti-kontrolleri sisendite ja väljundite programmi töö tegemiseks.

Eestis on kõige rohkem levinud just kaks robotitootjat tänu robotite funktsionaalsusele ning kiirele klienditoele. Need on ABB ja Yakisawa Motoman. Andrei Maksimovi jutu järgi on Skandinaavia riikides kõige populaarsemad robotid firma ABB omad. ABB ja Motomani robotitel on paremini realiseeritud paralleelsete juhtimiskäskude täitmine võrreldes teiste robotite juhtimiskontrolleritega. Teiseks, Fanuci robotite müügiesindust Eestis ei ole. Kui vaadata Eestis kasutusele võetuid roboteid, siis nendest on kõige rohkem just keevitusroboteid, umbes 80%, ning see arv kasvab. Seetõttu otsivad firmasid viisi, kuidas efektiivsemalt programmeerida roboteid (Lisa 3). Sama mõtte ütles välja ka keevitusroboti peatehnoloog Valeri Kizikov, kes töötab BLTR Grupp'is. Firmal on juba olemas keevitusrobot, mille programmeerimine võtab palju aega - kuni 60% terve roboti tööks ettevalmistamise protsessiks kuluvast ajast. Praegu toimub roboti programmeerimine läbi õpetuspuldi. Seetõttu on piiratud toodete valik ja partii kogus (vt. Lisa 4), mida robot on võimeline keevitama. Firma otsib lahendust, kuidas tõsta roboti tootmise efektiivsust – lühendada positsioneerimisaega punktide vahel ning luua automaatselt roboti juhtimisprogramme. Andrei Maksimov ja Valeri Kizikov ütlesid, et on vaja mingit tarkvara lõikamis- ja keevitusrobotite programmeerimiseks (Lisa 3 ja Lisa 4).



Joonis 2.1. ABB IRB-1600/1.45 (vasakul) ja Yakisawa Motoman UP6 (paremal)

Joonisel 4.1 on näidatud robotid, mida lõputöös kasutatakse Siemens NX-i post protsessori loodava meetodi katsetamiseks. ABB-st robotitest sai valitud manipulaator IRB-1600/1,45 tõstejõuga 5 kg. Selline manipulaator on olemas Tallinna Tehnikaülikooli Energeetikateaduskonnas (Lisa 1.).

Teine lõputöös kasutatud robot on Yakisawa Motoman UP6. Selle roboti kaal on ainult 130 kg ning tõstevõime on kuni 6 kg. UP6 saab tõsta detaili suurema kaaluga, aga tema tööala on väiksem kui ABB robotil.

Lõputöös on suunatud põhirõhk kaarkeevituse robotite juhtimisele ja programmeerimisele ning nende lisafunktsioonide kasutamisele, kuna need on kõige levinumad Eestis kui ka Euroopas (Lisa 3).

Lõputöö koosneb kolmest põhiosast Esimeses osas on toodud Siemens NX-i CAD-, CAE- ning CAM-moodulite võimalused ning nende kasutamine juhtimisprogrammi loomises. Teine osa kirjeldab ABB IRB1600/1.45 roboti programmeerimisviisi kasutades Siemens NX-i. See on omakorda jagatud osadeks kus on täpsemalt kirjeldatud, kuidas RAPID-programmeerimiskeeles (RAPID on ABB poolt loodud programmeerimiskeel IRB roboti juhtimissüsteemi jaoks) defineeritakse koordinaate, tööobjekti ning tööriista parameetreid ning käske. Kolmandas põhiosas on lahti kirjutatud Yakisawa Motomani programmeerimine. Selles osas vaadeldakse võimalust luua juhtimisprogrammi UP6 robotile kasutades Cartesian-koordinaate ning teoreetilisi arvutusi iga telje nurkkoodinaatide leidmiseks.

3. SIEMENS NX

Antud peatükis räägitakse kombineeritud tarkvarast tootmise automatiseerimiseks nagu Siemens NX. Seda tarkvara nimetatakse PLM-tarkvara lahenduseks. Antud lühend inglise keeles tähendab *Product Lifecycle Management* – ehk teiste sõnadega tähendab toote iga loomisetappide elutsükli juhtmine.

Siemens NX-i ajalugu algab aastal 1969, kui firma United Computing lasi turule tarkvara nimega UNIAPT. Hiljem, aastal 1973 United Computing ostis 2D CAD-moodulit nimega Automated Drafting and Machining ning integreeris selle oma tarkvarasse, mille uueks nimeks sai UNI-GRAPHICS (müüginimi Unigraphics). 1976 [1] aastal McDonnell Douglas ostis United Computing firma. Aastal 1983 [2] firma toob turule uue tarkvara nimega UniSolid, mis sai esimeseks täiesti interaktiivseks *solid* raalprojekteerimis tarkvaraks. 1991 [3] aastal McDonnell Douglasil olid rahalised raskused, ja firma müüs antud tarkvara firmale nimega EDS, mis tol ajal oli üks osa General Motorsi (GM) kontsernist. Niimoodi Unigraphics sai ühtseks CAD-lahenduseks terve GM kontserni jaoks. See tehing tegi Unigraphics-ust populaarse CAD-tarkvara [4]. Aasta hiljem kasutasid seda tarkvara juba 21 tuhat inimest. Aastal 2002 Unigraphics esitles oma uuendatud tarkvara NX, mis tähendas järgmist põlvkonda. See on koostis parimatest lahendustest, mis oli väljatöötatud Unigraphics-i enda meeskonnaga kui ka I-DEAS firmaga. Viis aastat hiljem ostis Siemens Unigraphics-i ning esitles oma uut tarkvara NX 5 koos Synchronous Technology-ga [5].

Tänu oma multi-funktsionaalsusele kasutatakse Siemens NX-i paljudel aladel nagu autode ehitus (Opel, GM), laevaehituses (Wärtsilä) ja lennukitootjad nagu Aerobus ning Sukhoi. Näiteks, Sukhoi Superjet 100 on modelleeritud kasutades ainult Siemens NX tarkvara.

Siemens NX-il on olemas ka konkureerivad tarkvarad. Põhilised konkurendid on Dassault Systems-i poolt pakutud tarkvada CATIA ning AutoDesk-ilt Inventor. Igal tarkvaral on omad plussid ja miinused. Aga üldise funktsionaalsuse poolest on need tarkvarad sarnased. Siemens NX ning Dassault Systems CATIA on kaks levinumat tippklassi CAD/CAE/CAM-tarkvara. Omavahel nad jagavad turu umbes võrdseteks osadeks. AutoDeski lahendus ei ole nii populaarne. CATIA CAD moodul põhineb tuntud SolidWorks-i põhjal ning omab suurt lisamoodulite arvu. Aga teiste moodulite võrdluses, eriti CAM-mooduli poolt, on Siemens NX parem. Antud vahet kompenseerib CATIA võimalusega lisada CAM-moodulisse lisatarkvara nagu HyperCAM, mis annab sarnast tulemust Siemens NX-ga.

AutoDesk on CAD/CAE/CAM-lahenduste turul tuntud ning heaks kiidetud tänu oma Inventor tarkvarale. Antud tarkvara ei ole nii levinud võrreldes teistega sest AutoDesk panustab arhitektuursele tarkvarale.

Siemens NX koosneb erinevatest moodulitest, mida saab jagada kolmeks põhirühmaks. Need on:

- CAD-moodul – detaili või koostu loomiseks
- CAE-moodul – mehhaanika arvutusteks ning detaili töö simuleerimiseks
- CAM-moodul – CNC-pinkide juhtimisprogrammi loomiseks

3.1. CAD-moodul

Siemens NX-i CAD-moodul koosneb mitmetest erinevatest gruppidest. Kõige esimene grupp on Solid Part. Antud moodul oli kõige esimene moodul selles tarkvaras, kui ka teistes tarkvarades. Üldiselt modelleerimisprotsess algab 2D joonestamisest. Esialgselt joonestatud kuju saab ekstruudida või pöörata ümber valitud telje. Kõik objektid antud moodulis on loodud sama loogika järgi. CAD moodulis peale 2 põhi funktsiooni on olemas ka *Boolean*-i loogika funktsioonid nagu liitumine, lahutus ning läbilõikamine. Igale detailile võib rakendada füüsilisi parameetreid, nagu materjali valik. Sellest tulenevalt võib Siemens NX arvutada massi, soojuspaisumist ja teha tugevdusarvutusi. See on hea integreerimistäide. Väga oluline omadus on ka kirjeldada mudelilt tootmisega seotatud info nagu täpsused ning tolerantsid.

Erilist tähelepanu võib pöörata paar aastat tagasi lisatud funktsioonidele, mis said nime *Synchronous Technology*, edaspidi *ST*. Antud funktsioonid võimaldavad muuta detaili kuju isegi siis, kui detailide loomise sammude ajalugu lihtsalt puudub. See juhtub näiteks, siis kui avada mudeleid, mis ei ole loodud Siemens NX-i tarkvaras või on teistes failide formaatides nagu STEP, IGES või Parasolid [6].

Peale *solid*-moodili omab Siemens NX ka lehtmetailmoodulit. Lehtmetaili moodul on erivariant tavalisest *solid*-modelleerimise moodulist. Lehtmetailist modelleerimisel kasutatakse 2 parameetrit. Need on metalli paksus – see on ühtlane ja see ei saa muutuda samal detailil erinevates kohtades. Ehk detail peab olema tehtud ainult ühe paksusest materjalist. Teine parameeter näitab materjali venivust painutamiskohas. See parameeter mõjub pinnalaotuse pikkusele. Antud moodulis saab kiirelt luua painded, erikujulisi seinu. Üks oluline võimalus – teha erikujulisi avasid, mis on lõigatud välja normaalsega antud pinnal. Antud funktsioon on väga oluline ja seda kasutatakse praktiliselt igal pool. Kuna detail lehtmetailist on esialgselt sirge, siis kõik avad on ka sirgel pinnal. Moodulis on olemas mõned lisafunktsioonid, mis lihtsustavad lehtmetaili detailide kujundamist. Kui näiteks paine ei pea olema terve serva

pikkuses, siis tavaliselt tehakse kaks väljalõige mõlema painde otstesse. See annab võimaluse materjali voolamiseks painutuse jooksul.

Siis kui detailid on tehtud, siis neid võib omavahel kokku panna koostuks. Koostude jaoks CAD-moodulis on olemas eraldi grupp nimega *Assemblies* (ing. keeles Koostud). Antud moodulis on funktsioonid detailide omavaheliseks seostamiseks. Koostamises kasutatakse geomeetrilisi ühendusalgoritme nagu panna kaks tasapinda kokku, tsentreerida joonel, tangentsiaalne ühendus jne. Kasutades algoritme võib luua parameetrilisi sidemeid teiste osadega ning edaspidi, kui on vaja muutuda ühe detaili kuju või mõõtu, siis tarkvara automaatselt uuendab koostu vastavalt loodud seostele ning uuendatud detaili kujule.

Koostude moodulis saab kasutada ka detailide raamatukogusid ning rühmitusi. Raamatukogus on salvestatud detailid mida saab korduvalt kasutada. Mutrite, poltide ning seibide kohta on mõistlikum luua detaili grupe. Kusjuures detail võib olla parametrizeeritud. Näiteks polt – insener valib õige keerme pikkuse. Tarkvara genereerib kuju automaatselt, vastavalt valitud parameetritele.

Töö kiirendamiseks Siemens NX-i CAD-moodulis on olemas veel erigrupid laevaehituseks, pressvormide modelleerimiseks, ja n.n. *Human Design*. Viimane rühm on mõeldud selliste detailide loomiseks, millega inimene puutub kokku: käepidemed, istmed jne.

Siemens NX-il on ka moodul, mis annab võimalust paigaldada juhtmeid ning voolikuid. See moodul kasutab salvestatud raamatukogudes torude ristlõiked ning nurkühendused. Samasuguselt raamatukogudes on salvestatud juhtmete parameetrid.

Siemens NX-i CAD osa on realiseeritud väga kõrgel tasemel. Selle mooduli grupid omavad väga palju erinevaid funktsioone, mida kasutades tõuseb uute toodete loomisprotsessi efektiivsus ning kiirus. Eri väärtuslik on omavaheline integreerimistase, mis võimaldab ühest moodulist teise ümbermineku andmete kaotamiseta.

3.2. CAE-moodul

Teine suur osa tarkvaras on CAE-moodul, põhineb tuntud NX Nastran algoritmidel. Nastran – see on algoritmide hulk, mis on integreeritud Siemens NX-i. NX Nastrani arvutusalgoritmid põhinevad *FEM*-arvutustel. *FEM* (ing. keeles *Finite Element Modelling*) on lõpliku arvuga elementide modelleerimine. Arvutusi, mida saab teha NX Nastran võib jagada kaheks grupiks. Esimene on mehhaanilised koormused. Kusjuures Nastran lubab ka võtta arvesse temperatuuri kõikumist protsessi jooksul. Teine on gaaside ja vedelike liikumine mingitel tingimustel [7].

NX Nastrani annab võimaluse vaadelda õhu voo temperatuuri ning suunda, õhu kiirust, et selgitada, kas antud paigaldus on piisavalt ventileeritav ning kus võib kasutada suure

võimsusega elemente. Nastran annab ka võimaluse vaadelda pressvormis oleva sula materjali voolamis- ning jahutamiskiiruste analüüsi. Lisaks tarkvara saab näidata ka neid kohti, kus pressvormis võib tekkida õhumull.

Kasutades NX Nastrani saab teha mehhaanilist analüüsi nagu koormuse ning vibratsiooni vastupidavus erinevatel tingimustel. Antud analüüs näitab kas konstruktsioon omab piisavalt tugevust ja millised on selle konstruktsiooni nõrgemad kohad. Teiseks, antud analüüs näitab ka koormuse jaotust elemendist elemendini, mis aitab konstruktoril leida õiget kohta, mida on vaja teha tugevamaks. Kusjuures analüüsi saab teha kas ainsa detaili põhiselt, või terve koostu jaoks. Koostu virtuaalseks katsetamiseks on vaja ainult lisaks määrata, kuidas koostus olevad detailid on omavahel seotud. Siis Siemens NX-i Nastran saab teada kuidas koormus jaotub detailide vahel. Gaaside ning vedelike liikumise analüüsi kasutavad mitmed firmad, kes tegelevad autode tootmisega, näiteks kui uurivad kere aerodünaamilisi parameetreid. Samal ajal mootorite konstruktorid uurivad kütuse segu kollektorist silindritesse liikumist ning resonantsi. Sellist analüüsi kasutavad ka lennukite ehitajad, kellel aerodünaamilised omadused on kriitilised. Kindlasti ka elektroonika seadmete insenerid saavad kasutada antud mooduli osa, kuna võimsatel elektroonika seadmetel on väga oluline jahutusviis ning selle efektiivsus.

Mehhaanilist analüüsi kasutavad praktiliselt kõik firmad, mis tegelevad metalltöötlemisega. Alustades Eesti firmadest nagu haagiste tootja Bestnet AS (Tiki Trailer), lõpetades Opel AG, Aerobus ja jne. Üleval oleval joonisel on näidatud kuidas jaotab koormus, ning kohad, kus koormus on maksimaalne. Lisaks võib vaadelda ka materjali deformeerimist koormuse all. Antud analüüs aitab põhjalikumalt uurida toodangu füüsikalisi parameetreid ning vastavalt vajadusele tugevdada konstruktsiooni.

Kasutades simuleerimismoodulit saab ka analüüsida mudeli liikuvate osade koormust ning vaadata tervet liikumisprotsessi. Tänu analüüsile saab kontrollida veerehõõrdejõudu laagrites, jõudu võlli otsades, kuidas võll peab vastu koormustele ning kas ei ole esinenud detailide omavahelist kokkupõrget ja kõik liikuvad osad on lubatud piirides.

Virtuaalne analüüs annab võimaluse leida ning eemaldada nõrgad kohad konstruktsioonist juba modelleerimistasandil. Omakorda kiirendab tootmist, kuna langeb vajadus toota palju prototüüpe katsetamiseks.

3.3 CAM-moodul

Viimane suur rühm funktsioone kulub CAM-mooduli alla. Antud moodulis on kombineeritud funktsioonid CNC-pinkide juhtimisprogrammide loomiseks. Siemens NX-il on hästi realiseeritud võimalus salvestada ning taaskasutada tööriistu, mis on salvestatud andmebaasis.

Aga Siemens NX sai tuntuks mitte ainult tänu andmebaasidele. Kasutades CAM-mooduli funktsioone saab automaatselt luua juhtimisprogrammi. Programmeerijale on vaja ainult määrata tööriistad, milledega ta kavatseb töödelda materjali. Siis on vaja tarkvarale ka selgeks teha, mis kujuga peab olema lõplik tulemus ning viimaseks on vaja määrata kuidas detail on kinnitatud pingis, ehk teiste sõnadega määrata koordinaatsüsteemi 0-punkti asukohta. Edaspidi pink teeb analüüsi ning jagab iga detaili kuju avadeks, mida on vaja puurida, ning aladeks, mida on vaja freesida ning leiab sise- ja väliskontuurid [8].

Üks kasulik omadus, mis on realiseeritud antud tarkvaras on võimalus kopeerida samu operatsioone teistele detailidele. See tunduvalt kiirendab juhtimisprogrammi loomist. Kui detaili töötlemisoperatsioonid on genereeritud, siis võib luua juhtimisprogrammi, ehk teiste sõnadega post protsesseerida operatsioone [9].

Post protsesseimine tähendab juhtimisprogrammi loomist konkreetsele CNC-pingile. Post protsessoris on kirjeldatud konkreetse pingi kinemaatika, parameetrid, kiirused ning kiirendused. Aga olulisem osa post protsessoris on instruksioonid ning algoritmid, mis kirjeldavad protsessi juhtimisprogrammi loomist. Post protsessis olevad algoritmid võib jagada gruppideks. Loogiliselt algoritme saab jagada kolmeks grupiks. Esimene grupp on käsud CNC-pingile, mida väljastatakse enne töötlemisprotsessi algust. Sinna kuuluvad CNC-pingi ettevalmistamiskäsud, nagu koordinaatsüsteemi valik, mõõtühikute valikud ning mõned seaded korrektorite kohta. Järgmine grupp vastutab juhtimisprogrammi sisu eest. Selles grupis on kogutud algoritmid kaadri loomiseks, kus on kirjeldatud järjekord ning mis käsud peavad olema väljastatud post protsessorina. Kui operatsioonid on lõppenud, siis väljastatakse juhtimisprogrammi lõppu. Post protsessori loomiseks pakub Siemens NX lisatarkvara nagu *Post Builder*. Antud tarkvara kasutades saab tunduvalt lihtsamini luua post protsessorit tänu kasutatavate käsude visualiseerimisele. Genereeritud koodi tasub alati virtuaalselt kontrollida, kas ei toimu kokkupõrget. Kuna juhtimisprogrammis on võimalik eksida kahes kohas, siis Siemens NX pakub n.n. ISV-meetodit, mis inglise keelest tõlkides on *Integrated Simulation and Verification* [10]. Teiste sõnadega Siemens NX CAM-moodulis on integreeritud funktsioonid CNC-pingi simulatsiooni jaoks. Tänu sellele saab kontrollida enne juhtimisprogrammi post-protsesseerimist, kas valitud CNC-pink on võimeline üldse liikuma mööda genereeritud trajektoori. Simuleeritakse just genereeritud trajektoor, mitte koodi ennast. Koodi kontrollimiseks on olemas teine funktsioon, mis laeb just post-protsesseeritud juhtimisprogrammi või selle juppi enda keskkonda ning proovib simuleerida pingi liikumisi vastavalt juhtimisprogrammi sees oleva koodi järgi. Antud võimalus annab kindlama tulemuse, kuna antud juhul saab üle vaadata ka lisaliikumisi nagu tööriista vahetustsükli ning teisi käske.

Kui detaili töödeldakse kahe või rohkema spindliga korraga, siis on kindlasti vaja veenduda, et tööriistad ei põrku üks teise vastu.

Roboti töö simuleerimiseks on vaja teha kaks põhi asja. Esimesena tuleb luua roboti 3D mudel. Näiteks, kui tahetakse simuleerida ABB roboti tööd, siis ABB pakub võimalust alla laadida oma robotite mudeleid oma kodulehelt. See on lihtsam ning mugavam, kuna sellel juhul on tunduvalt väiksem võimalus eksida roboti mõõtudega. Kui mudel on valmis, siis on vaja lisada sellele mudelile telgede asukohad ning nende parameetrid: nimi, piirangud ning kirjeldus, kas see on pöördtelg või lineaarselt liikuv telg. Kui see on tehtud, siis tuleb luua mudelile draiver. See on algoritmide hulk, salvestatud failis, mis vastutab juhtimisprogrammi koodi konverteerimise eest liikumiseks. Ilma selle algoritmita roboti mudel ei saa õigeid andmeid liikumise simuleerimiseks. Kuna selle draiveri loomine on raske ning vajab vastavalt koolitatud spetsialisti Siemensi poolt, siis antud lõputöös kasutatakse genereeritud juhtimisprogrammi testimiseks ABB poolt loodud tarkvara RobotStudio.

4. ABB IRB1600 (1.45M)

Firma ABB IRB seeria robotid on väga levinud Euroopas kui ka Eestis. Selle lõputöö jaoks sai valitud firma ABB robotitest IRB1600 (1.45m), mis on olemas Tallina Tehnikaülikooli Energeetikateaduskonna laboris. ABB robotite juhtimisprogrammide loomiseks kasutatakse programmeerimiskeelt RAPID. RAPID on väga võimas programmeerimiskeel. Selles programmeerimiskeeles on palju funktsioone, mis tunduvad lihtsustavad ning kiirendavad roboti programmeerimist. ABB robotite juhtimisprogrammid koosnevad osadest, mida võib jagada kaheks gruppiks:

- Konstantide ja muutujate grupp
- Protseduuride grupp

Esimeses grupis salvestatakse tööriista parameetreid (*tooldata*), tööobjekti asukoht (*wobjdata*) ning punktide koordinaadid (*robtarget*, *jointtarget*). Antud ploki salvestatakse ka globaalseid ja lokaalsed muutujaid. Muutujate väärtuste jagamine erinevateks gruppideks raskendab Siemens NX-i post protsessori loomist. Kuna tavalistel CNC-pinkidel juhtimiskontrolleritega Sinumerik, Heidenhain, Fanuc jne, andmete jagamist ei ole. CNC-pingide juhtimisprogrammis andmed koordinaatide kohta on salvestatud samas reas koos protsessi käsuga.

```
N10 G01 G54 X100. Y0. F1000  
N11 G01 Z-10. S3000 M04
```

Robotite juhtimisprogrammi punktide andmete ning käsude jagamine on põhjustatud punktide taaskasutamise võimalusega. See tähendab, et sama punkt võib olla kasutatud mitmes kohtades programmi sees. Globaalsed punktide koordinaate saab kasutada erinevates protseduurides.

ABB roboti koordinaadis pöördnurgad kirjeldatakse kvaternionidega. Siemens NX-i post protsessor väljastab ilma lisa arvutusteta pöördnurgad kraatides, mis kirjeldavad pöörlemist ümber X-, Y- ja Z-telje. Aga neid väärtuse on vaja ümber arvutada kvaternionideks. See tõttu peab post protsessoris olema eraldi funktsioon, mis arvutaks ümber koordinaadi andmeid nii et neid saaks kasutada ABB roboti programmeerimiseks.

Teine grupp on nii nimetatud protseduuride grupp. Antud grupis salvestatakse liikumiskäsud koos sihtpunktidega, lisafunktsioonid, loogikakontrolleri juhtimiskäsud ja muud.

4.1 Tööriista ja tööobjekti etteantud parameetrid

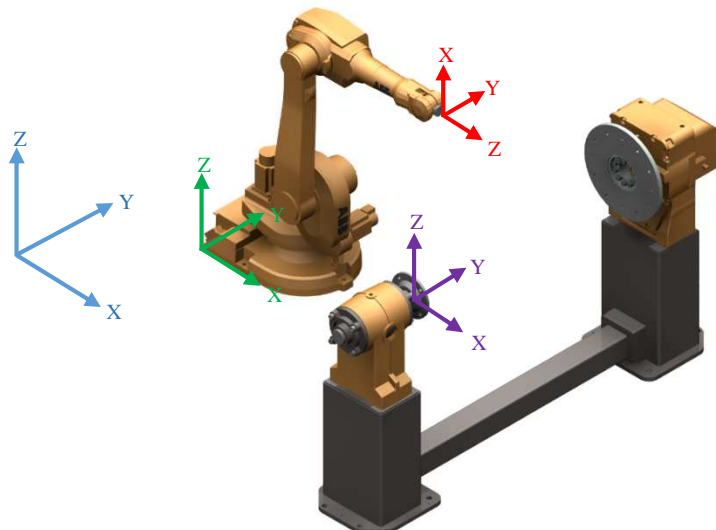
Nagu juba varem öeldud, salvestatakse esimesse gruppi erinevate konstantide parameetreid. Üks nendest on tööriista andmed. Need andmed on ABB poolt jagatud plokkideks. Tööriista kirjeldavad järgmised parameetrid [11]:

- Kas robot hoiab tööriista – loogiline väärtus (*TRUE* või *FALSE*)
- Tööriista keskpunkti asukoht X,Y,Z – reaalne arv
- Tööriista baasis orientatsioon, ümberarvutatud kvaterniodeks – reaalne arv
- Tööriista kaal – reaalne arv
- Tööriista massikeskpunkt – reaalne arv
- Momendi projektsioonid X, Y ja Z-teljele – reaalne arv

All on toodud näide, kuidas pannakse kirja tööriista parameetreid. Tööriist nimega „GRIPPER“ on kinnitatud roboti külge, tööriista keskpunkt on nihutatud 97,4 mm mööda X-telge roboti tööriista kinnitusflantsi koordinaatsüsteemis, ja nihutatud 223,1 mm võrra mööda Z-telge samas koordinaatsüsteemis. Tööriista X telg ja Z telg on pööratud 45 kraadi võrra manipulaatori koordinaatsüsteemis. Tööriist kaalub 5 kg, ja tema massi keskpunkt on nihutatud 23 mm mööda X-telge ja 75 mm mööda Z-telge ning ei ole kuskile pööratud.

```
PERS tooldata GRIPPER:= [ TRUE, [[97.4, 0, 223.1], [0.924,
0, 0.383 ,0]],[5, [23, 0, 75], [1, 0, 0, 0], 0, 0, 0]];
```

Tööriista andmed mõjutavad roboti liikumist. Tööriista keskpunkt on see punkt, mis liigub mööda programmeeritud rada antud ettenihkega. Juhtimiskontroller kasutab seda punkti roboti kiiruse ning positsiooni arvutamiseks, teised teljed võivad liikuda teiste kiirustega. Antud punkt on virtuaalne ning ei kirjelda tööriista füüsilist kuju.



Joonis 4.1. Roboti koordinaatsüsteemid

Tööobjekti andmeid kasutatakse detaili nullpunkti asukoha kirjeldamiseks. Tööobjekti andmed koosnevad järgmistest osadest:

- Kas robot hoiab seda objekti – loogiline väärtus (*TRUE* või *FALSE*)

- Kas antud tööobjekti koordinaatsüsteem on fikseeritud mingis punktis – loogiline väärtus.
- Kui robotil on välisteljed nagu pöördlaud, siis selle lisatelje nimi (näiteks „orbit_a“). Saab kasutada ainult kui tööobjekti koordinaatsüsteem ei ole fikseeritud.
- Nullpunkti X,Y ja Z koordinaadid baaskoordinaatsüsteemis.
- Nullpunkti nihkenurgad kirjeldatud kvaternionidega.
- Antud koordinaatsüsteemi sees oleva objekti nullpunkti X,Y ning Z koordinaadid.
- Antud koordinaatsüsteemi sees oleva objekti orientatsioon kirjeldatud kvaternionidega.

Üleval toodud andmed on grupeeritud RAPID programmis nii, et neid oleks lihtsam lugeda. Siin on tööobjekti koordinaatsüsteemi näide:

```
PERS wobjdata LEHT1 :=[ FALSE, TRUE, "", [ [300, 600, 200],
      [1, 0, 0 ,0] ],[ [0, 200, 30], [1, 0, 0 ,0] ] ];
```

Üleval toodud näidisel on tööobjekt nimega LEHT1, mis ei ole kinnitatud roboti külge, ning selle objekti koordinaatsüsteem on fikseeritud. Antud tööobjekti koordinaatsüsteem ei ole pööratud aga on nihutatud 300 mm mööda X-telge, 600 mm mööda Y-telge ja 200 mm mööda Z-telge. LEHT1 koordinaatsüsteemi sees oleva objekti nullpunkt on nihutatud 200 mm mööda tööobjekti koordinaatsüsteemi Y-telge, ning 30 mm mööda Z-telge. Tööriista kui ka tööobjekti defineeritakse muutuja tüübina nagu PERS, ning seda tehakse juhtimisprogrammi alguses. PERS-tüüpi muutujaid on keelatud defineerida rutiinis. Joonisel 6.1 on näidatud roboti koordinaatsüsteemide asukohad:

- Maailma koordinaatsüsteem - sinine
- roboti baaskoordinaatsüsteem - roheline
- tööriista koordinaatsüsteem - punane
- tööobjekti koordinaatsüsteem – lilla.

4.2 Punkti koordinaadi arvutus

Viimane, mida üldiselt kirjutatakse konstantide ja muutujate gruppi on roboti punktid, mida kasutatakse rutiinides. Üldiselt roboti sihtpunktide kirjeldamiseks on eeldatav kasutada lisa parameetrit nagu *CONST*, mis määrab punkti konstandina. Punkti kirjeldus koosneb järgmistest parameetritest:

- X,Y ja Z-telje koordinaadid millimeetrites – reaalne arv.
- Pöördnurgad arvutatud kvaternionideks – reaalne arv.

- Mis kvaternionide veerandis asub tööriista keskpunkt – täisnumber 0...3
- Välis- ning lisateljed koordinaadid – reaalne arv.

Viimase punkti kohta tuleb lisada, et väärtus 9E9 tähendab, et telg ei ole ühendatud robotiga ja robotikontroller ei juhi seda telge. All on toodud näidis, kuidas punkti kirjeldus näeb välja:

```
CONST robtarget p15 := [ [600, 500, 225.3], [1, 0, 0, 0],
    [1, 1, 0, 0],[ 11, 12.3, 9E9, 9E9, 9E9, 9E9] ];
```

Punkt p15 on nihutatud 600 mm mööda X-telge, 500 mm mööda Y-telge ning 225,3 mm mööda Z-telge määratud koordinaatsüsteemis. Sama punkt sama väärtustega võib olla kasutatud mitmetes erinevates koordinaatsüsteemides ja anda erineva tulemust, s.t. et punkt võib paikneda erinevates kohtades. Antud punkti kvaternionide väärtuse põhjal võib öelda, et antud punktis tööriist ei ole pööratud mööda ühtegi telge. Manipulaatori 1. ja 4. telg asuvad 90-180° piiris ning 6 telg on 0-90°. Esimene ja teine lisatelg on nihutatud 11 ühikut ja 12,3 ühikut vastavalt sellele, mis ühikutes antud telgi mõõdetakse. Mõõtühikud võivad olla kas millimeetrites või kraadides, sõltub konkreetse telje seadistest. Tuleb mainida, et punkti sees ei ole andmeid millisesse koordinaatsüsteemi see kulub. Punkti väärtused näitavad ainult punkti nihet koordinaatsüsteemi nullpunktist. Punktiga seotud koordinaatsüsteem RAPID-keeles määratakse protseduuri sees.

Siemens NX salvestab punkti asukoha hetkväärtust massiivi. Väärtuste saamiseks on võimalik kasutada muutujaid [12] \$mom_pos(0...4), kus:

- \$mom_pos(0) – X-koordinaat antud hetkel (kaadri täitmisel).
- \$mom_pos(1) – Y-koordinaat antud hetkel (kaadri täitmisel).
- \$mom_pos(2) – Z-koordinaat antud hetkel (kaadri täitmisel).
- \$mom_pos(3) – Esimene pöördkoordinaat antud hetkel (kaadri täitmisel).
- \$mom_pos(4) – Teine pöördkoordinaat antud hetkel (kaadri täitmisel).

Esimese punkti koordinaatide massiiv koosneb järgmistest elementidest:

```
[ $mom_pos ( 0 ) , $mom_pos ( 1 ) , $mom_pos ( 2 ) ]
```

Teine massiiv koosneb kvaternioni 4 väärtusest, mis kirjeldavad tööriista orientatsiooni tööobjekti koordinaatsüsteemis [13].

$$x = (x_1, x_2, x_3), \quad (4.1)$$

kus x_1 – alamkoordinaadistiku X-telje projektsioon peakoordinaatsüsteemi X-teljel,
 x_2 – alamkoordinaadistiku X-telje projektsioon peakoordinaatsüsteemi Y-teljel,
 x_3 – alamkoordinaadistiku X-telje projektsioon peakoordinaatsüsteemi Z-teljel.

$$y = (y_1, y_2, y_3), \quad (4.2)$$

kus y_1 – alamkoordinaadistiku Y-telje projektsioon peakoordinaatsüsteemi X-teljel,
 y_2 – alamkoordinaadistiku Y-telje projektsioon peakoordinaatsüsteemi Y-teljel,
 y_3 – alamkoordinaadistiku Y-telje projektsioon peakoordinaatsüsteemi Z-teljel.

$$z = (z_1, z_2, z_3), \quad (4.3)$$

kus z_1 – alamkoordinaadistiku Z-telje projektsioon peakoordinaatsüsteemi X-teljel,
 z_2 – alamkoordinaadistiku Z-telje projektsioon peakoordinaatsüsteemi Y-teljel,
 z_3 – alamkoordinaadistiku Z-telje projektsioon peakoordinaatsüsteemi Z-teljel.

Teiste sõnadega see tähendab, et alamkoordinaatsüsteemi X-telg projekteerub põhikoordinaatsüsteemi X-teljele nagu x_1 , alamkoordinaatsüsteemi Y-telg projekteerub põhikoordinaatsüsteemile on siis x_2 jne. Siit tuleb, et kolm vektorid x , y ja z võivad olla ühendatud maatriksiks [14]:

$$R = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix} \quad (4.4)$$

Üleval oli näidatud võimalus kirjeldada alamkoordinaatsüsteemi asendit rotatsioonmaatriksi kaudu. Kasutades rotatsioonmaatriksi võib leida punkti orientatsiooni järgi kvaternionide väärtused (11):

$$q_1 = \frac{\sqrt{x_1 + y_2 + z_3 + 1}}{2}, \quad (4.5)$$

kus q_1 – esimene kvaternion.

$$q_2 = \left| \frac{\sqrt{x_1 - y_2 - z_3 + 1}}{2} \right|, \quad (4.6)$$

kus q_2 – teine kvaternion.

$$\text{sign } q_2 = \text{sing}(y_3 - z_2) \quad (4.7)$$

$$q_3 = \left| \frac{\sqrt{y_2 - x_1 - z_3 + 1}}{2} \right|, \quad (4.8)$$

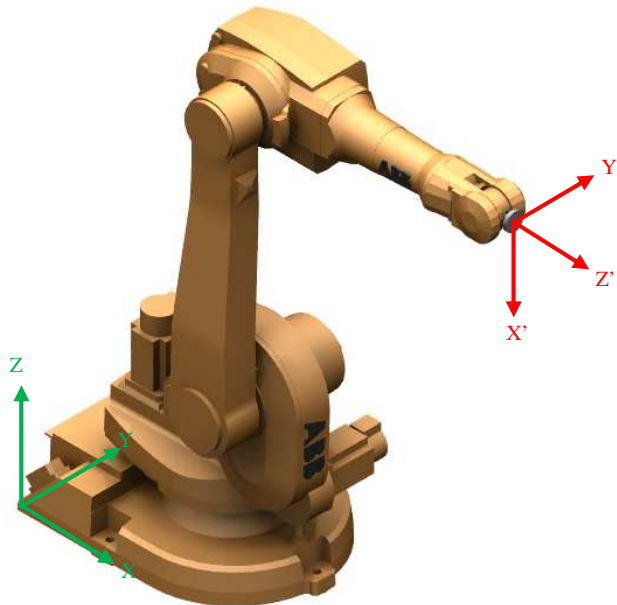
kus q_3 – kolmas kvanternion.

$$\text{sign } q_3 = \text{sing}(z_1 - x_3) \quad (4.9)$$

$$q_4 = \frac{\sqrt{z_3 - x_1 - y_2 + 1}}{2}, \quad (4.10)$$

kus q_4 – neljas kvanternion

$$\text{sign } q_4 = \text{sing}(x_2 - y_1) \quad (4.11)$$



Joonis 4.2. Koordinaatsüsteemid vastavalt näitele

Tööriista koordinaattelgede projektsioonid tööobjekti või roboti baaskoordinaatsüsteemis võib arvutada pöördnurkade kaudu. All pool on toodud n.n. elementaarsed rotatsioonimaatriksid [15]:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, \quad (4.12)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}, \quad (4.13)$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.14)$$

kus $R_x(\alpha)$ – pöörmaatriks X-telje ümber,

$R_y(\beta)$ – pöörmaatriks Y-telje ümber,

$R_z(\gamma)$ – pöörmaatriks Z-telje ümber.

Näiteks kui roboti tööriista koordinaatsüsteemi Z'-telg on suunatud sama suunda kuhu baaskoordinaatsüsteemi X-telg (vt. Joonis 4.2). Tööriista koordinaatsüsteemi Z'-telje suund langeb kokku baaskoordinaatsüsteemi X-teljega. Ehk teiste sõnadega tööriista koordinaatsüsteem on pööratud ümber Y-telje -90° võrra, või siis $\beta = -90^\circ$. Tööriista koordinaatsüsteemi X'-, Y'- ja Z'-teljed projekteeruvad baaskoordinaatsüsteemile vastavalt järgmiste valemitele: valem 6.12 kasutatakse X-telje projektsiooni arvutamiseks, valemi 4.13 – Y-telje jaoks ning kasutades valemit 6.14 arvutatakse Z-telje projektsiooni

$$X' = (\cos(-90^\circ), 0, \sin(-90^\circ)) = (0, 0, -1) = -Z$$

$$Y' = (0, 1, 0) = Y$$

$$Z' = (-\sin(-90^\circ), 0, \cos(-90^\circ)) = (1, 0, 0) = X.$$

Saavutatud võrrandite põhjal saab koostada rotatsioonmaatriksi:

$$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Kasutades üleval toodud valemiteid leitakse $q_1 \dots q_4$ väärtuseid. Valemit 4.7 tuleb kasutada q_2 märgi leidmiseks. Valemeid 4.9 ning 4.11 kasutatakse märgi leidmiseks vastavatel parameetritel q_3 ja q_4 .

$$q_1 = \frac{\sqrt{0 + 1 + 0 + 1}}{2} = \frac{\sqrt{2}}{2} \approx 0.707$$

$$q_2 = \frac{\sqrt{0 - 1 - 0 + 1}}{2} = 0$$

$$q_3 = \frac{\sqrt{1 - 0 - 0 + 1}}{2} = \frac{\sqrt{2}}{2} \approx 0.707$$

$$\text{sign } q_3 = (1 + 1) > 0 \rightarrow "+"$$

$$q_4 = \frac{\sqrt{0 - 0 - 1 + 1}}{2} = 0$$

$$\text{sign } q_4 = \text{sign}(0 - 0) = 0 \rightarrow "+"$$

Tavaliselt toimub roboti tööriista pöörlemine ümber mitmete telje detaili töötlemise ajal. Selleks on vaja tuletada elementaarsete rotatsioonimaatriksitest ühe maatriksi, mis kirjeldaks

tööriista koordinaadistiku pöörlemist määratud koordinaatsüsteemis. Sellist maatriksi nimetatakse rotatsioonmaatriks $R_{xyz}(\alpha, \beta, \gamma)$.

$$R_{xy}(\alpha, \beta) = R_x(\alpha) \times R_y(\beta) \quad (4.15)$$

$$R_{xy}(\alpha, \beta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \times \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix}$$

$$R_{xy}(\alpha, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ \sin \alpha \cdot \sin \beta & \cos \alpha & -\sin \alpha \cdot \cos \beta \\ -\cos \alpha \cdot \sin \beta & \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix}$$

$$R_{xyz}(\alpha, \beta, \gamma) = R_{xy}(\alpha, \beta) \times R_z(\gamma) \quad (4.16)$$

$$R_{xyz}(\alpha, \beta, \gamma) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ \sin \alpha \cdot \sin \beta & \cos \alpha & -\sin \alpha \cdot \cos \beta \\ -\cos \alpha \cdot \sin \beta & \sin \alpha & \cos \alpha \cdot \cos \beta \end{bmatrix} \times \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{xyz}(\alpha, \beta, \gamma) =$$

$$= \begin{bmatrix} \cos \beta \cdot \cos \gamma & -\cos \beta \cdot \sin \gamma & \sin \beta \\ \sin \alpha \cdot \sin \beta \cdot \cos \gamma + \cos \alpha \cdot \sin \gamma & -\sin \alpha \cdot \sin \beta \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma & -\sin \alpha \cdot \cos \beta \\ -\cos \alpha \cdot \sin \beta \cdot \cos \gamma + \sin \alpha \cdot \sin \gamma & \cos \alpha \cdot \sin \beta \cdot \sin \gamma + \sin \alpha \cdot \cos \gamma & \cos \alpha \cdot \cos \beta \end{bmatrix}$$

Valemi 4.16 alusel saab leida kvadratsioonide arvutuse valemid, mida kasutatakse post protsessoris roboti tööriista orientatsiooni määramiseks.

$$q1 = \frac{\sqrt{\cos \beta \cdot \cos \gamma - \sin \alpha \cdot \sin \beta \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma + \cos \alpha \cdot \cos \beta + 1}}{2} \quad (4.17)$$

$$q2 = \left| \frac{\sqrt{\cos \beta \cdot \cos \gamma + \sin \alpha \cdot \sin \beta \cdot \sin \gamma - \cos \alpha \cdot \cos \gamma - \cos \alpha \cdot \cos \beta + 1}}{2} \right| \quad (4.18)$$

$$\text{sign } q2 = \text{sign}(\cos \alpha \cdot \sin \beta \cdot \sin \gamma + \sin \alpha \cdot \cos \gamma + \sin \alpha \cdot \cos \beta) \quad (4.19)$$

$$q3 = \left| \frac{\sqrt{-\sin \alpha \cdot \sin \beta \cdot \sin \gamma + \cos \alpha \cdot \cos \gamma - \cos \beta \cdot \cos \gamma - \cos \alpha \cdot \cos \beta + 1}}{2} \right| \quad (4.20)$$

$$\text{sign } q3 = \text{sign} (\sin \beta + \sin \alpha \cdot \sin \beta \cdot \cos \gamma - \sin \alpha \cdot \sin \gamma) \quad (4.21)$$

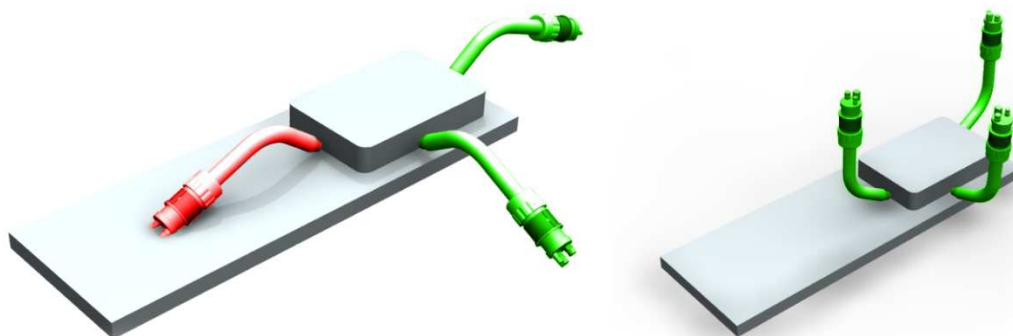
$$q4 = \left| \frac{\sqrt{\cos \alpha \cdot \cos \beta - \cos \beta \cdot \cos \gamma + \sin \alpha \cdot \sin \beta \cdot \sin \gamma - \cos \alpha \cdot \cos \gamma + 1}}{2} \right| \quad (4.22)$$

$$\text{sign } q4 = \text{sign} (\sin \alpha \cdot \sin \beta \cdot \cos \gamma + \cos \alpha \cdot \sin \gamma + \cos \beta \cdot \sin \gamma) \quad (4.23)$$

Lisas 5 on toodud Siemens NX-i post protsessori algoritm koordinaatide arvutamiseks.

4.3 Trendline

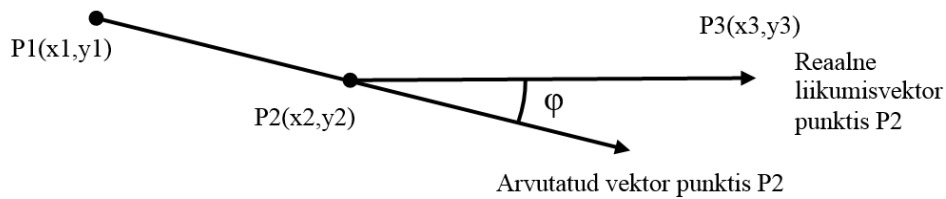
Kuna robotid on tavaliselt 6 teljelised, siis viimase telje pöördenurka saab arvutada kaudselt. Viis telge on piisav freesimiseks ja robotitele koos freespeaga ei ole mõtet arvutada kuuenda telje orientatsiooni [18]. Aga keevitusrobotitel võib juhtuda olukord, millal 6. telg peab olema rakendatud (vt. Joonis 4.3) .



Joonis 4.3. Parem pool kokkupõrge, vasakul – korrektne liikumine

Antud lõputöös on pakutud kuuenda telje orientatsiooni leidmiseks diferentsiaalne arvutusviis. Kuna Siemens NX-i süsteemi muutujate väärtused on ainult hetkväärtused, siis ei ole otseselt võimalik kasutada eelmise või tuleva punkti koordinaate. Antud probleem lahendatakse viimast punkti salvestamisega eraldi globaalsesse muutujasse. Lahutades antud punkti koordinaatidest (P2) eelmise punktide koordinaate (P1) saab leida vektori, mis kirjeldab liikumissuunda. Antud meetodi puuduseks on see, et liikumisvektor antud ajahetkel ei ole aktuaalne ja on arvutatud juba hilinemisega, kuna allpool toodud joonisel 6.4 on näidatud, et reaalne liikumisvektor punktis P2 on esitatud nagu $(x3-x2, y3-y2)$, kuid tuletatud liikumisvektor punktis P2 on $(x2-x1, y2-y1)$. Kuna reaalse liikumisvektori ja arvutatud vektori vahel olev nurk ei ületa 0.10° , siis

võib öelda, et keevituses ei ole see hilinemine oluline, ning võib lugeda arvutatud liikumisvektorit hilinemisega aktuaalseks ja sama suunaga nagu reaalne liikumisvektor.



Joonis 4.4. Arvutatud ja hetkvektori suunade erinevus.

Esimene pöördnurk kirjeldab pöörlemist ümber X-telje ehk YZ-tasapinna pöörlemist, teine pöördnurk kirjeldab pöördliikumist ümber Y-telje ehk XZ-tasapinna pöörlemist. Ainuke nurk, mis jääb puudu on see, mis kirjeldab pöörlemist ümber Z-telje ehk XY-tasapinna pöörlemine. Antud nurga arvutamiseks võib kasutada kahe vektorite omavahelise nurka koosinuse võrrandit [16].

$$\cos \gamma = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|}, \quad (4.24)$$

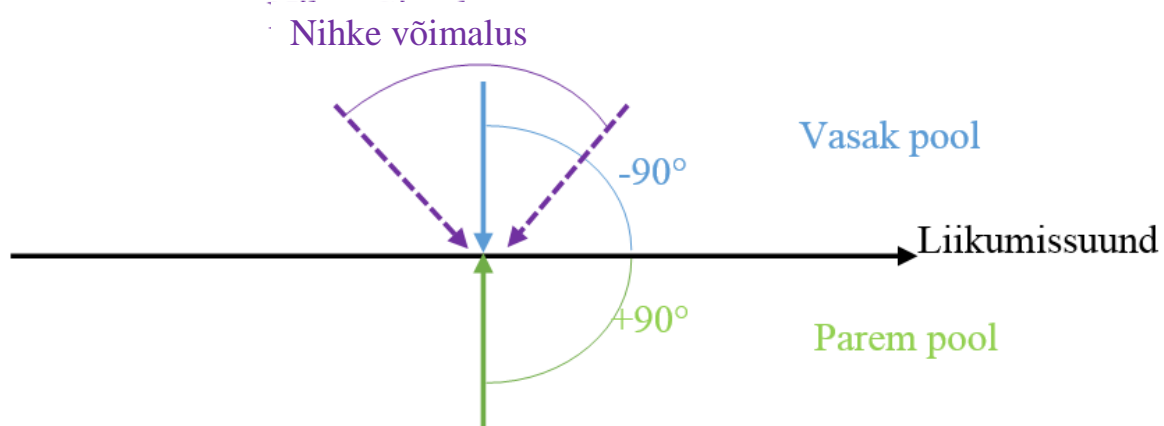
kus a – esimene vektor, mille koordinaadid on $[a_1, a_2]$,
 b – teine vektor koordinaatidega $[b_1, b_2]$.

Kasutades üleval toodud valemit 6.24 arvutatakse nurka X-telje ning liikumisvektori vahel. Liikumisevektori koordinaadid on lahutuse tulemus konkreetse hetke punkti ja eelmise punkti X- ja Y-koordinaatide vahel. Teine vektor on üksik vektor mis langeb kokku X-teljega ja omab koordinaate $[1,0]$. Kuna vaadeldav pöördnurk on XY-tasapinnas, siis Z-parameeter jääb alati 0. Nurga leidmiseks tuleb lihtsustada standardset valemit kahe vektori vahelise nurga leidmiseks. Selle tulemus on valem 4.25:

$$\gamma = \arccos \frac{(x_h - x_e) \cdot 1 + (y_h - y_e) \cdot 0}{\sqrt{(x_h - x_e)^2 + (y_h - y_e)^2} \cdot \sqrt{1^2 + 0^2}} \rightarrow \frac{(x_h - x_e)}{\sqrt{(x_h - x_e)^2 + (y_h - y_e)^2}}, \quad (4.25)$$

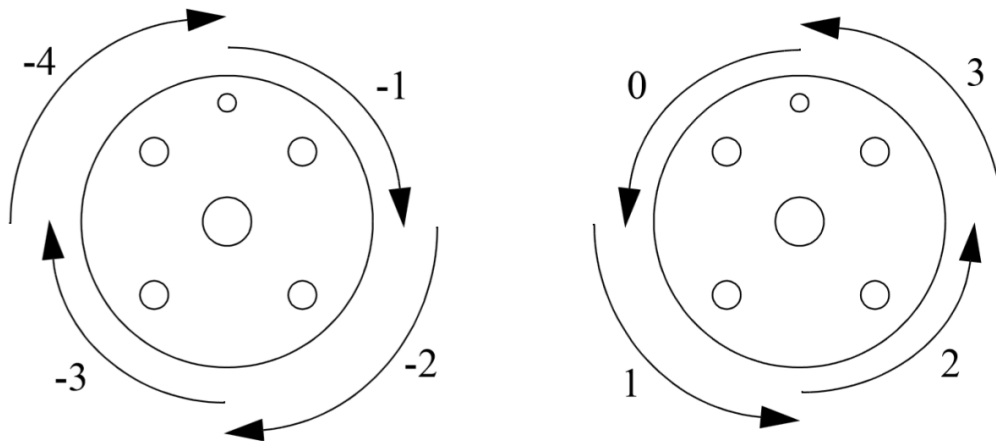
kus x_h – X-koordinaat hetkpunktis,
 x_e – X-koordinaat eelmises punktis,
 y_h – Y-koordinaat hetkpunktis,
 y_e – Y-koordinaat eelmises punktis.

Kuna nurk γ on piirides $[0^\circ;180^\circ]$, aga vajalik miinimumi vahe on $[0^\circ;360^\circ]$, siis tuleb jagada vajalik piir kaheks pooleks. Kui liikumisvektori Y-koordinaat on suurem või võrdub 0, nurk γ on piirides $[0^\circ;180^\circ]$. Kui aga liikumisvektori Y-koordinaadi väärtus on väiksem kui 0, siis nurk γ on piirides $(180^\circ;360^\circ)$.



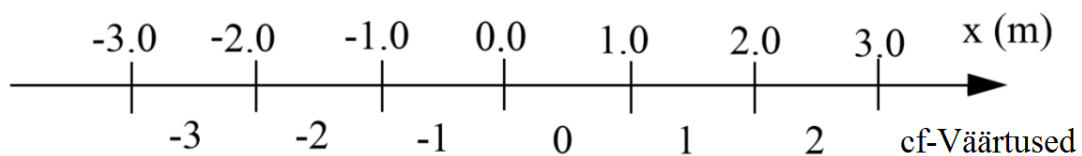
Joonis 4.5. Tööriista pool liikumisvektori suhtes, ülevaltvaade

Nüüd saavutatud pöörnurk kirjeldab pöördliikumist XY-tasapinnas. Tavaliselt keevitustööriist on suunatud liikumisvektoriga risti. Inseneril on võimalus ka sättida nihe kummalegi poole (vt. Joonis 4.5). Siemens NX ei suuda automaatselt määrata tööriista paiknemist liikumisvektori suhtes. Seda paiknemist määrab roboti programmeerija ise. Veel määrab programmeerija kaldenurga liikumisvektori normaalist. Seda nihet on vaja keevitusprotsessi optimeerimiseks. Veel ühe raskuse tekitab roboti kinemaatikamudel. Cartesean'i koordinaatsüsteemis esitatud punkti peab robot arvutama ümber oma telgede pöördnurkadeks. Selle arvutuse jooksul tekib mitu lahendust, kuidas manipulaator saab jõuda punkti. See tõttu tuleb viia sisse piiranguid, mis määraks ainult ühe õige tulemuse. Selleks kasutatakse *robtar*-i parameetrid *cf1*, *cf4*, *cf6* ning *cfx* (vt. Joonis 4.6. ja Joonis 4.7). Ette öeldes, Motomani robotitel kinemaatikamudeli lahenduse parameetrid on ette antud robotite seadistest, kuid ABB robotikontrollerid arvutavad seda iga kord uuesti välja, kui manipulaator liigub mingisse punkti. Tänu sellele ABB robot võib teha tööd nn. „üleval“ positsioonil, kui ka „alumisel“ positsioonil. Selle eest vastutavad need samad parameetrid *cf1...cfx*, mis suunavad roboti kinemaatikamudeli lahenduse õigele poole nii, et juhtimiskontroller arvutuse lõpuks saaks ainult ühe tulemuse (11).



Joonis 4.6. Kuuenda telje pöörlemissuuna määramine.

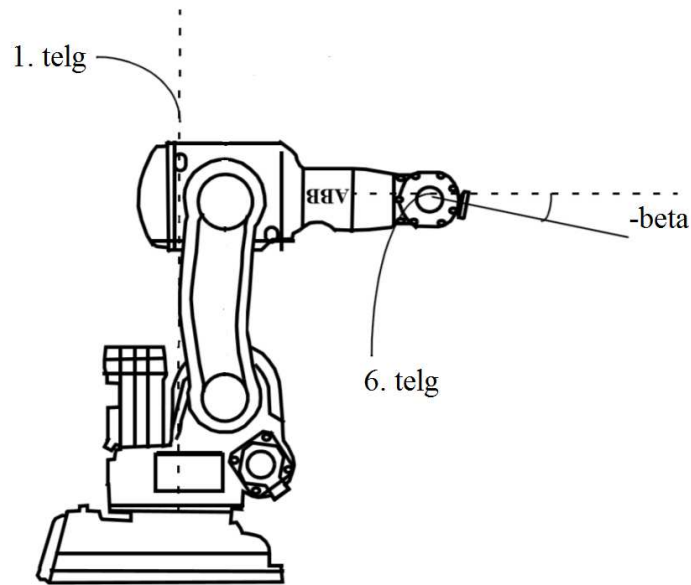
ABB robotite juhtimissüsteem võimaldab piirata asukohta lineaarsetel telgedel (vt. Joonis 4.7), samasuguselt nagu pöördtelgedel (vt. Joonis 4.6). Antud parameetrite kasutamine võimaldab juhtimiskontrollerile õiget lahendust piirtingimustel, näiteks kui telg teeb täispööre.



Joonis 4.7. Lineaarse telje parameetrid

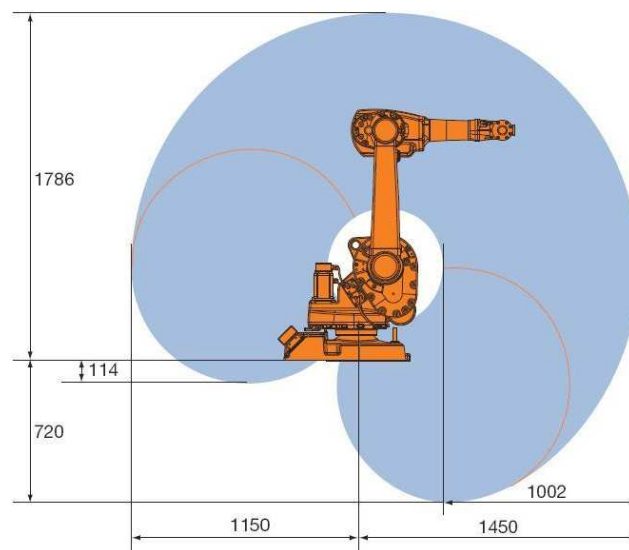
ABB IRB1600 roboti puhul protsessor arvutab välja õiget kvadranti 1., 4., ning 6. telgede jaoks ning salvestab tulemuse koordinaadi massiivi. All toodud cf1...x algväärtused.

- cf1=0 – esimene telg on esimeses kvadrantis, ehk 0° - 90°
- cf4=0 – neljas telg on esimeses kvadrantis, ehk 0° - 90°
- cf6=1 – kuues telg on teises kvadrantis, ehk 90° - 180°
- cfx=1 – IRB1600 roboti puhul, teine telg üleval positsioonil (vt. Joonis 4.8.)



Joonis 4.8. Kinemaatika mudel juhul kui $c_{fx} = 1$

Cfx parameeter üldjuhul vastutab teise telje positsiooni eest. Seda parameetrit kasutatakse ainult ABB IRB5400 5. telje määramiseks, kuna see on lineaarne. Antud juhul, kui parameeter cfx omab väärtust 1, siis on tegemist kinemaatikamudeliga nagu allpool toodud joonisel 4.9.



Joonis 4.9. ABB IRB5400 kinemaatikamudel

4.4 Liikumiskäsud ja programmi väljastamine

Andmete väljastamine toimub juhtimisprogrammi lõpus, kus vastavalt väljastamisalgoritmile pannakse kirja esiteks tööriistade, tööobjektide ning punktide andmed, ja alles siis post protsessor kirjutab faili ülejäänud info protsesside kohta. Tahaks mainida, et tööriista ega tööobjekti andmeid ei ole esimeses ega teises massiivis. Neid andmeid Siemens NX-i post

protsessor võtab juba eraldi failist, kus insener on eelnevalt salvestanud andmed robotitööriista ning tööobjekti kohta. Antud lahendus lihtsustab tööriista parameetrite parandust.

Protsesside sisu võib jagada neljaks põhigrupiks. Need grupid siis on [11]:

- Lineaarne interpoleerimine – MoveL
- Liigendite interpoleerimine – MoveJ
- Kaar interpoleerimine – MoveC
- Lisafunktsioonid – Wait, CTime jne

Lineaarsel interpoleerimisel roboti juhtimissüsteem arvutab kõiki mootorite kiiruseid nii, et tööriista keskpunkti trajektoor oleks lineaarne. See vajab ka ettenihke määramist. Antud liikumisviisil võib tekkida probleem, et üks või mõned mootorid ei saa arendada kontrolleri poolt nõutud kiirust, et pidada liikumist lineaarseks. Tavaliselt selline juhtum annab veateate. Antud liikumisviisi kasutatakse tööprotsessil, kui robot teeb sirget keevitusõmblust või freesib detaili. MoveL kasutamine roboti kontuuride vahel positsioneerimiseks ei ole efektiivne, kuna see on aeglasem ning tavaliselt vajab suuremaid mootori kiirendusi, mis omakorda võib põhjustada kiiremat reduktoore mehhaanilist kulumist. Aga programmeerija jaoks antud liikumisviis on kõige lihtsam ette kujutada võrreldes teiste liikumisviisidega, eriti liigendite interpoleerimisega. Toome ühe näite liikumise käsust lineaarse interpoleerimisega:

```
MoveL Target_50, v1000, fine, PUNANE;
```

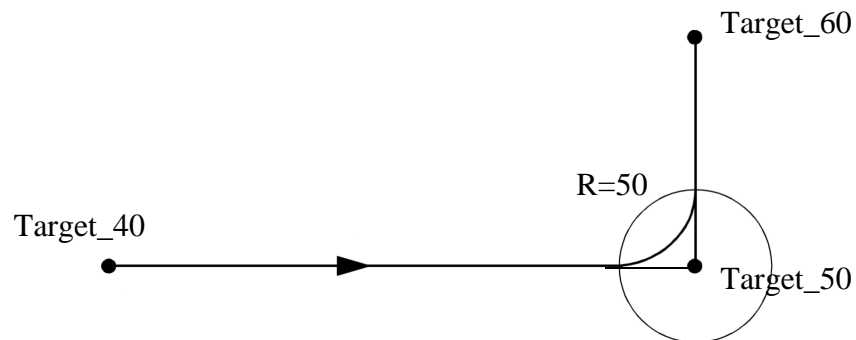
Antud näidisel kasutati liikumise käsku nagu MoveL, mis lülitab sisse lineaarse interpoleerimise juhtimiskontrolleris. Antud kaadri täitmise lõpus, robotil olev tööriist nimega *PUNANE* peab jõudma punktini nimega Target_50 kiirusega 1000 mm/min. Maksimaalne liikumiskiirus sõltub konkreetse roboti parameetritest, ning liikumisrajast, mille järgi tööriista *PUNANE* keskpunkt peab liikuma. Parameeter fine tähendab, et tööriista keskpunkt käsurea täitmise lõpuks peab jõudma täpselt sinna positsiooni, mis on salvestatud Target_50 nime all. Erivariant sellest liikumiskäsust on ArcL. Antud liikumiskäsk paned põlema keevituskaarte ning liigub samasuguselt nagu MoveL, ehk lineaarselt.

Järgmisena vaadatakse roboti manipulaatori ühenduste liikumist. Seda liikumisviisi aktiveeritakse käsuga MoveJ, inglise keelest tulnud lühendus „*move joint*“. Antud liikumises roboti juhtimissüsteem rakendab kiirendust vastavalt roboti kinemaatilisele mudelile koos koormusega – tööriistaga ning koormuse kaaluga, kui on ette antud. Liikumiskiirus aga arvutatakse „umbes“ ja see on lineaarne. Allpool on toodud näide roboti ühenduste liikumise kaadrina.

```
MoveJ Target_50, vmax, z50, PUNANE;
```

```
MoveJ Target_60, vmax, fine, PUNANE;
```


Antud kaadri täitmisel roboti juhtimiskontroller paneb manipulaatori liikuma maksimaalse kiirusega niikaua kuini robot jõuab punkti Target_50. Tööriista keskpunkt ei liigu lineaarselt punktide vahel ning kiirus ei ole täpselt arvatatud (kiirus on umbes arvatatud). Liikumises võetakse arvestusse tööriista nimega „PUNANE“ keskpunkti. Liikumine jätkub järgmisesse punkti, kui tööriista keskpunkt jõuab punkti Target_50 ümber defineeritud z50 ringkujulisse alasse, mille raadius on 50 mm (vt. Joonis 4.10)



Joonis 4.10. Roboti punktide tsoonid

Viimane liikumisviis on liikumine kaarel. Kaarel liikumiseks on vaja määrata 2 punkti – liikumise sihtkoordinaat ning kaarel olev punkt absoluutkoordinaatides samas koordinaatsüsteemis. Liikumine on iseloomustatud kiirusega, mis on samasugune nagu sirgjooneliikumise puhul ja selle ühikud on mm/s Ülejäänud parameetrid on standardsed liikumise käsu jaoks ja kirjeldatud üleval. Vaatame täpsemalt liikumist kaarega näidise peal.

```
MoveC Target_50, Target_60, v200, z5, PUNANE;
```

Antud käsu täitmisel roboti juhtimiskontroller paneb robotit liikuma kaare peal punktini Target_60. Kaare keskpunkt absoluutkoordinaatides asub punktis Target_50. Tuleb panna tähele, et kaare keskpunkt absoluutkoordinaatides on väga oluline, kuna on väga levinud variant kasutada pikkust sihtpunktist kaare keskpunkti või vastupidi. Antud näites roboti tööriist nimega PUNANE keskpunkt peab jõudma sihtpunkti määratud kiirusega 200 mm/min.

Algoritm lineaarse liikumise arvutamiseks Siemens NX-i post protsessori jaoks on toodud Lisas 7. Liikumiskaardi genereerimise algoritmid liigendite interpoleerimise ning ringinterpoleerimise jaoks on tehtud lineaarse algoritmi põhjal. Liikumiskäsu määrab Siemens NX trajektoori genereerimise jooksul. Liikumisviis edastatakse mom_motion_mode muutuja kaudu. Ettenihe, tööriista ning tööobjekti nimesid edastatakse järgmiste muutujate kaudu:

- mom_main_mcs – tööobjekti nimi,
- mom_tool_name – tööriista nimi,
- mom_feed – ettenihe väärtus mm/min.

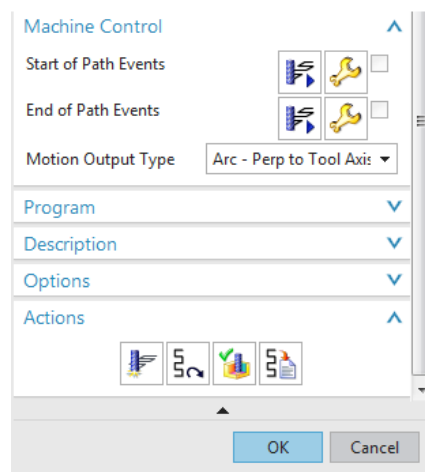
Kasutades üleval toodud muutujaid post protsessor genereerib liikumiskäsu, mille salvestab siis massiivi.

4.5 Lisafunktsioonid

Robotitööriista aktiveeritakse ning deaktiveeritakse vajadusel kasutades funktsioone. Kui see on keevitusaparaat, siis on vaja keevituskaart sisse- või välja lülitada. Kui see on automatiseeritud vakuutõste aparaat, siis on vaja lülitada sisse ning välja vaakumpumba mootorit ja määrata õhuvoolu suunda. Kui roboti tööriistana on freespea, siis on vaja kindlaks määrata freesi või puuri pöörlemiskiirust, sisse lülitamise hetk ning pöörlemissuund. Et tööriista juhtida võib kasutada programmeeritavat loogikakontrollerit, mis on ühendatud roboti juhtimiskontrolleriga näiteks Profibus-i kaudu. Profibus on ühendus mis on mõeldud Siemens loogikakontrollerite kui ka teiste seadmete omavahel ühendamiseks. Antud ühendusviis on tuntud oma laia funktsionaalsusega ning võimalustega. Kasutades Profibus-i ühendust on võimalik isegi kirjutada ümber kontrolleri juhtimisalgoritmi otse roboti juhtimiskonsoolist, kui seda võimaldab roboti juhtimiskontrolleri tarkvara. Aga enamus käskusid on standardsed funktsioonid. Antud lõputöös vaadeldakse järgmisi lisafunktsioone [17]:

- Wait – käsk oodata määratud aega sekundides, väljundi või sisendi signaali.
- Keevituse sisse- ning väljalülitamine
- TrendLine-i parameetrid – post protsessor arvutab manipulaatori kuuenda telje väärtuse.

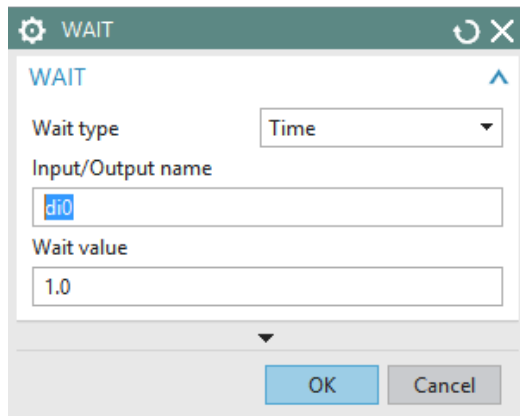
Siemens NX-i post protsessorid saavad edastada mõned kasutaja poolt määratud käsud ainult mõnedes konkreetsetes kohtades – kas operatsiooni alguses või lõpus (vt. Joonis 4.11).



Joonis 4.11. Lisafunktsioonide (UDF) valikud

Kasutaja lisafunktsioone üldiselt nimetatakse *UDF*-iks, antud lühend on inglise keelne ja ta on *User Defined Function*, mis tõlkides on kasutaja poolt defineeritud funktsioon. Siemens NX-i

lisafunktsioonide kasutamiseks luuakse graafiline kasutaja interfeiss. Näiteks kasutaja funktsioon *WAIT* näeb välja järgmiselt:



Joonis 4.12. Kasutaja funktsiooni graafiline aken

Iga lisafunktsiooni kasutajaliidet luuakse kasutades koodi (vt. allpool toodud näide). Samal koodil määratakse iga muutuja nime ning väärtuse tüüp. All toodud näide kuidas näeb välja joonisel 6.12 olev aken kui seda kirjeldada koodiga.

```
EVENT wait_command
{
    POST_EVENT "WAIT"
    UI_LABEL "WAIT"
    CATEGORY MILL
    PARAM wait_type
    {
        TYPE o
        DEFVAL "Time"
        OPTIONS "Time","DI","DO"
        UI_LABEL "Wait type"
    }
    PARAM wait_input_output_name
    {
        TYPE s
        DEFVAL "di0"
        UI_LABEL "Input/Output name"
    }
    PARAM wait_value
    {
        TYPE s
        DEFVAL "1.0"
```

```

        UI_LABEL "Wait value"
    }
}

```

Nagu toodud joonisel 6.12, kasutaja funktsioon koosneb kolmest parameetrist, mida saab valida, vastavalt vajadusele. Kõige esimene parameeter määrab mida robot peab ootama – aeg, sisendi signaal või väljundi signaal. Valikud on ette antud. ABB robotite juhtimissüsteem võimaldab ka oodata teise roboti protseduuri lõppu, oodata faili laadimist jne, aga antud töös seda ei vaadelda. Esimene parameetri muutuja omab nime *mom_wait_type* ja selle tüüp on *OPTION*, ehk tõlkides see tähendab *VALIK NIMEKIRJAST*. Võimalikud väärtused on:

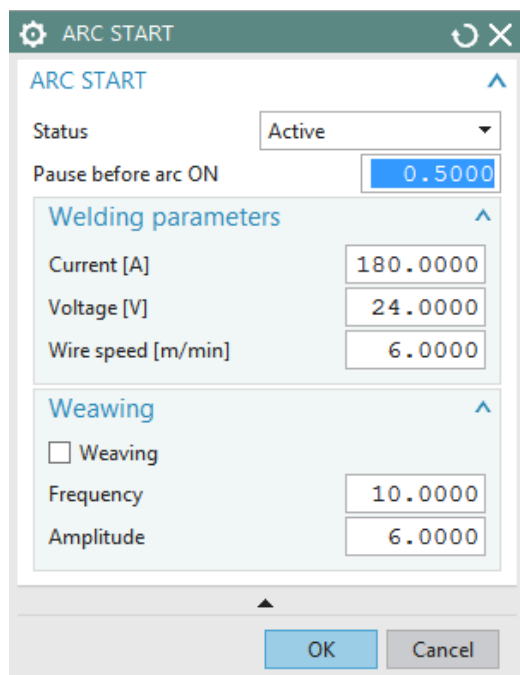
- *TIME* – aeg sekundides,
- *DI* – oodata signaali digitaalsel sisendil,
- *DO* – oodata signaali digitaalsel väljundil.

Muutuja nimega *mom_wait_input_output_name* on *s*-tüüpi (*string*) võimaldab kirjutada sinna absoluutselt kõike. Viimase muutuja nimi on *mom_wait_value*. Selle muutuja tüüp on reaalne arv, aga ühe omadusega. Siis kui edastatakse aja väärtust, siis antud muutuja täpsus on 10^{-2} s. Juhul kui muutuja ootamistüüp on väljund või sisend, siis *mom_wait_value* konverteeritakse naturaalarvuks.

Veel üks vajalik lisafunktsioon on keevituse sisse lülitamine. Antud käsk aktiveerib keevitusaparaadi. ABB robotite juhtimiskontroller lülitab kaart liikumiskäsuga. Kui liikumine toimub sisse lülitatud kaarega, siis liikumiskäsud muutuvad *MoveL*-st *ArcL*-ks ning *MoveC* muutub *ArcC*-ks. Antud kasutaja funktsioon omab mitut muutujat. Esiteks kõige olulisem muutuja on funktsiooni staatus *mom_arc_start_status*. Antud muutuja väärtus aktiveerib post protsessoris algoritmi, mis esiteks muudab liikumiskäsu liikumiseks koos kaarkeevitusega. Järgmise muutuja väärtus määrab aja, mis robot ootab enne kaare sisse lülitamist. Seda on vaja, et kaoks vibratsioon peale kiire liikumist punktini. Selle muutuja nimi on *mom_pause_before_arc_on*. Järgmised muutujad määravad kaare parameetreid keevitamiseks.

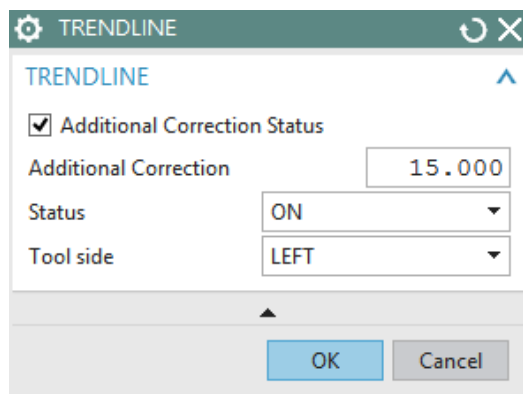
- *mom_arc_current* - kaare vool, A,
- *mom_arc_voltage* – kaare pinge, V,
- *mom_wire_speed* – traadi ettenihe m/min.

Kindlasti laia keevitusõmbluse tekitamiseks on vaja panna keevituspüstoli väikese amplituudiga liikuma. Post protsessoris on ette nähtud eraldi parameetrite grupp. Need on *mom_arc_weaving*, *mom_arc_weaving_frequency* ning *mom_arc_weaving_amplitude*. All oleval joonisel on näidatud keevitusaparaadi seadised ning valikud (vt. Joonis 4.13).



Joonis 4.13. Kaarkeevituse sisselülitamine ja parameetrid/näitajad

Antud kasutaja funktsioonid on universaalsed ja võib kasutada teiste robotite (tootjate) post post protsessori loomiseks. Selline universaalsus lihtsustab uute post protsessorite loomist. Samas ka viimane kasutajafunktsioon on universaalne. See on *TrendLine*. Antud funktsioon lülitab sisse mooduli kolmanda pöördnurka (ümber Z-telge) väärtuse arvutamiseks. *TrendLine* annab võimalust programmeerida 6 teljest sünkroonset liikumist.



Joonis 4.14. TrendLine parameetrid

Antud kasutaja funktsioonis on 4 muutujat (vt. Joonis 4.14). Need on lisa pöördnurka staatus, nihe, muutuja *Status* mis määrab TrendLine-i kasutamise vajadust. Viimane muutuja määrab tööriista suhtelise liikumissuuna.

4.6 Juhtimisprogrammi väljastamine

Antud peatükis kirjeldatakse post protsessori juhtimisprogrammi väljastamise põhimõtet. Siemens NX esialgselt edastab andmeid, mida on vaja ümber arvutada. Positsiooni muutujad

$mom_pos(3)$ ning $mom_pos(4)$ mille väärtused kirjeldavad nurkadena tööriista vektori suunda koordinaadistikus. Muutujaid $mom_pos(0...2)$ võib mitte arvutada, kuna need muutujad kirjeldavad kauguseid nullpunktist kuni sihtpunkti projektsioonini X, Y või Z-teljel ning antud väärtuseid post protsessor kasutab koordinaadi loomiseks. Muutujad $mom_pos(3)$ ja $mom_pos(4)$ on kraadides. Neid on vaja konverteerida radiaanideks, kuna üldiselt kõikides programmeerimiskeeltes trigonomeetrilised funktsioonid kasutavad väärtusi just nimelt radiaanides. Koordinaadi loomise ning massiivi salvestamise eest vastutab post protsessori algoritm nimega *PB_CMD_COORDINATE_CALCULATION*. Antud algoritm konverteerib kõik nurgad kraadidest radiaanideks kasutades võrrandid

$$\alpha_{rad} = \frac{\alpha_{deg} \cdot \pi}{180^\circ}, \quad (4.26)$$

kus α_{deg} – nurk kraadides,
 α_{rad} – nurk radiaanides.

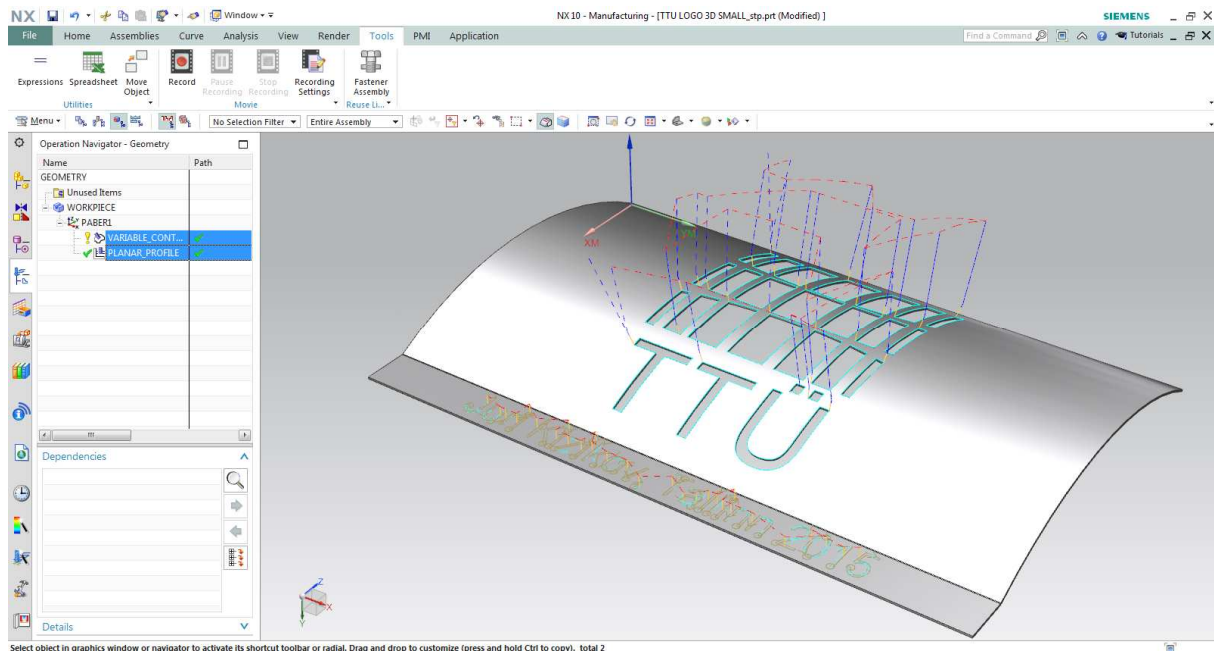
Kahjuks post protsessoris ei ole salvestatud π -väärtust ja seda on vaja välja arvutada kasutades järgmist koodi [19]:

```
set Pi [expr 4*atan(1)]
```

Üldine post protsessori töö põhimõte toimib loogika skeemi järgi, mis on toodud Lisas 8. Kui kõik punktid ning käsud on töödeldud ja salvestatud massiividesse, siis hakkab post protsessor salvestama andmeid juhtimisprogrammi faili. Kuna ABB juhtimissüsteem vajab juhtimisprogrammis tööobjekti ning tööriista parameetrite defineerimist, siis post protsessor loeb vajalikke andmeid failidest *tool_list.tool* ja *work_objects_list.wobj*, mis asuvad samas kaustas.

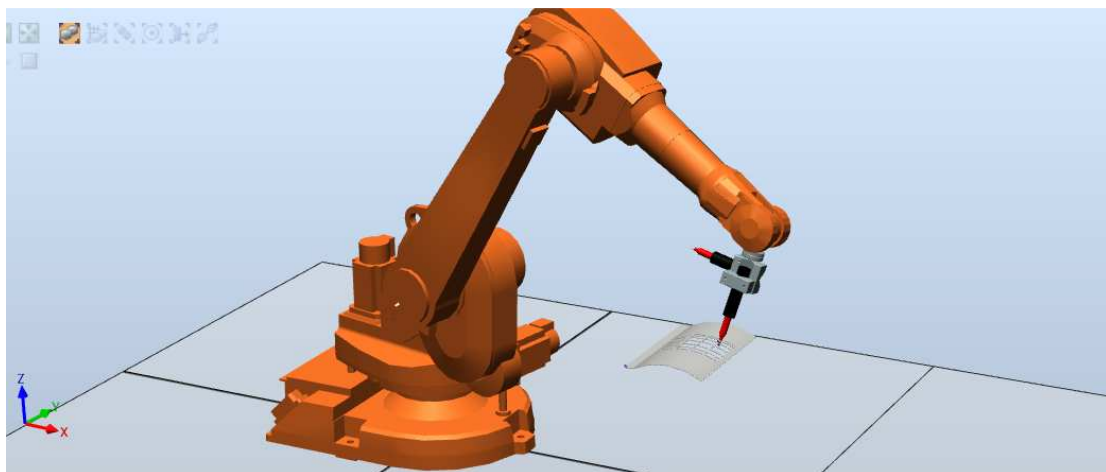
4.7 Virtuaalne simulatsioon ja katsetamine

Enne juhtimisprogrammi reaalse roboti laadimiseks, tasub alati virtuaalselt simuleerida seda. Virtuaalne simuleerimine annab võimaluse avastada programmi vigu. Simuleerimiseks sai valitud tarkvara RobotStudio, mida pakub ABB. Selleks tuleb genereerida juhtimisprogramm kasutades Siemens NX. All joonisel 4.15 on näidatud virtuaalse detaili mudel ning töötlemisoperatsioonid. Antud juhul detail on valtsitud plekk painutatud servaga.



Joonis 4.15. Siemens NX CAM moodul

Et virtuaalne tulemus oleks võimalikult sarnane reaalse katsega, on mõistlik lisada tööriista ning detaili mudelit ABB RobotStudio tarkvarasse. Selleks tuleb salvestada tööriistad ning tööobjekti mudeleid näiteks STEP-failina. Mudelite laadimisel tuleb määrata n.n. null-punkti asukoha valitud koordinaadistikus. Tööriista puhul tuleb kindlasti ka määrata *TCP* koordinaati. All oleval joonisel 6.16 on näidatud tööriista ning detaili mudelid simuleerimis keskkonnas.



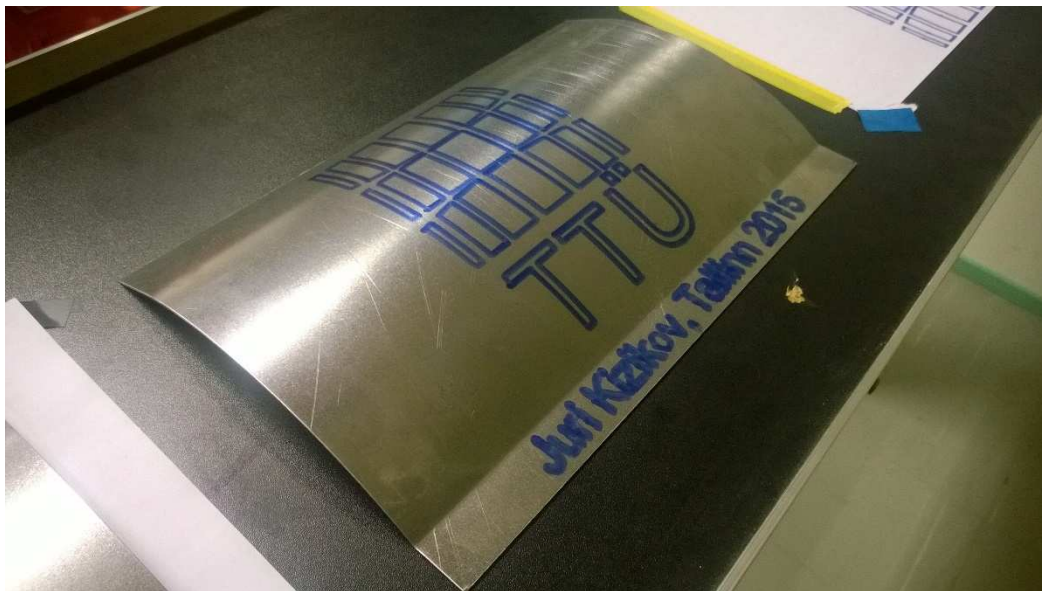
Joonis 4.16. ABB RobotStudio ning ABB IRB1600/1.45m

Tuleb mainida, et roboti tööriist oli modelleeritud tarkvaras Siemens NX. Antud tööriist koosneb osades nagu kinnitusflants, markeri suunajad, ning kinnitusklabrid. Seda tööriista saab kasutada ka Siemens NX enda olevas simulatsioonis.



Joonis 4.17. ABB IRB1600/1.45m katsetööriist (TTÜ Energeetikateaduskond)

Teiste tarkvarade kasutamine ABB robotite puhul võib anda teist tulemust, mis on reaalse robotiga erinev. Kasutades RobotStudio saab sünkroniseerida juhtimisprogrammi arvuti ning reaalsete juhtimiskontrolleri vahel. Uue juhtimisprogrammi saab laadida otse genereeritud Siemens NX-iga failist ABB RobotStudio-sse.



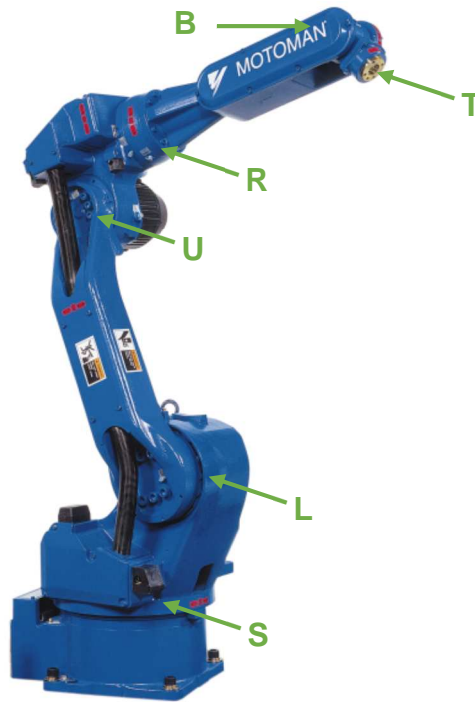
Joonis 4.18. Reaalselt valmistatud detail

Joonisel 4.18 on näidatud reaalse roboti töö tulemus. Antud detaili valmistamiseks oli automaatselt genereeritud umbes 4600 punkti. Programmi täitmise aeg kiirusega 50% määratud ettenihkest on 1 minut 32 sekundit.

Kui lühidalt võtta kokku, siis ABB IRB-seeria robotite programmeerimine on võimalik kasutades Siemens NX, aga on piiratud – ei ole võimalust otseliselt programmeerida korraga 6 telge. Siemens NX-iga ei ole otse võimalik programmeerida manipulaatorit painutuspingi jaoks ning sorteerimisrobotit. Edukalt sai realiseeritud võimalus programmeerida robotit IRB1600/1.45. Sai genereeritud ja proovitud juhtimisprogramm, mis joonestaks TTÜ logo A4 lehel kui ka mitte sirgetel pindadel. Seega võib öelda, et robotit võib praktiliselt ilma piiranguta programmeerida liikuma. Kuna tarkvara on ettenähtud trei ning freespinkide programmeerimiseks, siis võib öelda, et antud post protsessor on eriti efektiivne just siis, kui robotit kasutatakse freesimiseks. Keevitamine on ka võimalik aga piirangutega. Siemens NX ei võimalda rakendada kasutaja funktsiooni keset protseduuri ja see piirab kasutamisevõimalust.

5. YAKISAWA MOTOMAN ROBOTI PROGRAMMEERIMINE

Antud peatükis on juttu juhtimisprogrammi loomisest Yakisawa Motoman UP6 robotile, mis kasutab XRC juhtimiskontrollerit. Antud robot on üks väiksematest robotitest mida Motoman toodab. Robotil on 6 telge, mis on näidatud alloleval joonisel [21].



Joonis 5.1. Yakisawa Motoman UP6 robot

Roboti programmeerimiseks kasutatakse programmeerimiskeelt INFORM. Võrreldes RAPID (ABB) programmeerimiskeelega, võib leida ka sarnaseid funktsioone. Juhtimisprogrammi struktuurilt on need programmeerimiskeeled põhimõttest samad. Mõlematel robotitel on juhtimisprogramm jagatud kaheks gruppiks:

- Punktide koordinaatide grupp.
- Liikumiskäsude grupp.

Antud lõputöös näidatakse kahte viisi, kuidas saab salvestada Motomani roboti punkti asukohta. Esimene viis ei ole ratsionaalne, kuna vajab päris palju ressursi juhtimiskontrollerist. Selle põhimõte on iga järgnevale punktile lisada järgneva punkti vahe kasutades matemaatilist funktsiooni nagu *ADD*. Ehk teiste sõnadega see on inkrementaalne programmeerimine. Sellist programmi on väga raske muuta ning parandada juhtimiskontrolleri õpetuspuldist ning ei saa jätkata programmi keset koodi. Aga antud viis on kõige lihtsam vaadates post-protssessori kirjutamise poolt.

Teine viis on sama mis ABB post-protssessori puhul – salvestada punkti koordinaate XYZ-väärtusi ning kolm pöördnuka ühte massiivi, ning liikumiskäsk – teise. Antud koordinaadi

edastamisviis on optimaalne vaadates post-protsessori poolt kui ka roboti juhtimissüsteemi poolt. Sellist juhtimisprogrammi on võimalik korrigeerida otse juhtimiskonsoolist. Programm sellel juhul on tunduvalt paremini loetav ning iga liikumiskäsk on konkreetne.

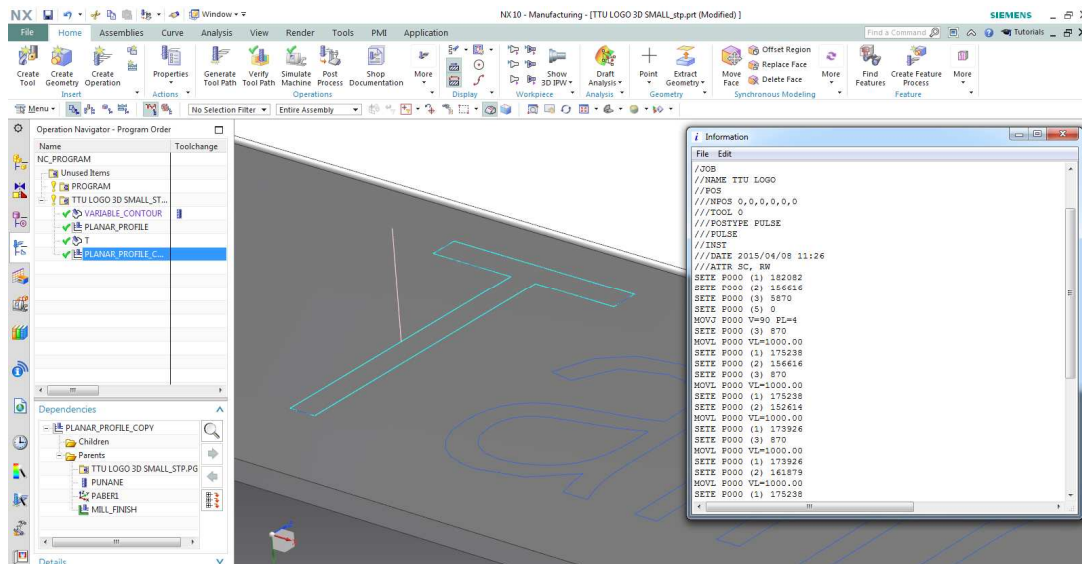
Viimane viis on kõige mahukam vaadates post-protsessori kirjutamise poolt, aga kõige lihtsam roboti juhtimissüsteemile. See on roboti iga pöördtelje nurga määramine. Üldjuhul Yakisawa Motomani juhtimissüsteem kasutab antud meetodit punkti koordinaatide salvestamiseks. Antud viis vajab palju arvutusi, kusjuures arvutuse jooksul võib tekkida mitu lahendust. Vigade vähendamiseks ning töövõimekuse tõstmiseks Motomani kontrollerid nagu XRC [22] ning kaasaegsem NX100 [23] ning kõige uuem DX200 [24] salvestavad punkte just kasutades telgede nurkkoordinaate.

5.1 XYZ-koordinaadid

Kõige lihtsam viis post protsessorile edastada sihtpunkti asukohta on arvutada seda roboti juhtimisprogrammi käskude abil. Kuna ABB ja Yakisawa Motomani juhtimissüsteemid vajavad punkti defineerimist enne käsu täitmist, siis jääb ainus võimalus edastada koordinaate koos käskudega kasutades punkti tüüpi muutujat. ning matemaatilisi funktsioone *SET*, *ADD* või *SUB* [25].

```
SETE P001 (0) 1500
SETE P001 (5) 18000
MOVJ P001 V=95
ADD P001 [0] 100000
MOVL P001 VL=2000
SUB P001 P001 'Nullida andmeid muutujal P001
```

SETE (*set element*) käsku kasutatakse siis, kui punkti koordinaati on vaja esitada absoluutkoordinaatides, *ADD* või *SUB* on diferentsiaalse programmeerimise jaoks. Näidel on hästi näha, kuidas esiteks määratakse punkti koordinaadid ja alles siis robot saab käsu liikuda punkti P001. Esiteks saab P001 X-telje väärtuse 15 mm ning Rz (pöördnurk ümber Z-telje) 180 kraadi. Siis toimub liikumine punkti P001 kiirusega umbes 95% maksimaalsest. Siis muutuja X-telje väärtusele lisatakse 100 mm. Seda nimetatakse programmeerimiseks ühe punkti kaudu. Kindlasti, võib alati kasutada erinevaid punkte, aga sellest realselt pole kasu, kuna kõik andmed, mis määravad punkti koordinaadid on salvestatud juhtimisprogrammi sees.



Joonis 5.2. T-tähe markeerimine, Siemens NX

Et paremini näidata juhtimisprogrammide erinevust juhul kui määrata punkti asukohta käskude grupi sees ja eraldi defineerimise vahel, tuleb vaadata joonist 5.2. Antud joonisel on genereeritud trajektoor „T“ tähe markeerimiseks. Juhtimisprogrammi Yakisawa Motomani jaoks saab vaadata Lisas 10 ning juhtimisprogramm ABB roboti jaoks on näidatud Lisas 11. Antud juhul Motomani juhtimisprogramm on tunduvalt väiksem, aga struktuurilt on see raskemini loetav

Antud programmeerimise viisil on oma plussid ja miinused. Selle viisi nõrgem koht on keeruline ja mahukas programm. See vajab palju ressursi roboti juhtimissüsteemilt, kuna roboti juhtimissüsteem peab ümber arvutama koordinaate kasutades matemaatilisi funktsioone. Plussiks võib nimetada punkti määramise täpsust kui määrata punkti juhtimispuldi abil.

Kui rääkida punkti täpsusest, ning punktist ise enesest, siis tuleb kindlaks teha, millest punkti koordinaadid koosnevad ja kuidas roboti juhtimissüsteem neid haldab. Kohe võib öelda, et iga muutuja Motomani juhtimissüsteemis võib olla esitatud massiivina. massiiv ise võib ka koosneda alammassiividest, näiteks punkti tüüpi muutujate massiivid. Allpool toodud tabel näitab, mis andmetest koosneb punkti muutuja [25].

Tabel 5.1. Punkti parameetrid.

Punkti parameeter	0	1	2	3	4	5
Vastab	X	Y	Z	Rx	Ry	Rz
Väärtuse tüüp	INT	INT	INT	INT	INT	INT
Täpsuse tegur	100	100	100	1000	1000	1000

Tabelis 5.1 toodud punkti parameetrite selgitus ning väärtuse tüübid. *INT* inglise keeles tähendab *integer* ehk tõlkides on täisarv. Siit tuleb huvitav punkti tüüpi muutujate omadus, ehk kuidas salvestada punkti täpsust kuni 10^{-2} millimeetrites, kui saab kasutada ainult täisarvusi. Vastuseks on siis täpsuse tegur. Iga väärtus tuleb korrutada antud teguriga et arvutada õiget impulsside arvu (5.1).

$$X_{calc} = X_{real} \cdot k_x, \quad (5.1)$$

kus X_{calc} – X-telje väärtus punkti muutujate massiivis,

X_{real} – X-telje reaalne või vajalik väärtus,

k_x – X-telje väärtuse täpsuse tegur.

Lineaarsete telgede puhul, täpsuse tegur on 100, ehk teiste sõnadega roboti juhtimissüsteem arvutab punkti asukoha 0.01 mm täpsusega. Nurga koordinaadi puhul täpsuse tegur on 1000. Pöördnurga koordinaadi täpsus on sellel juhul 0.001 kraadi.

5.2 Nurkkoordinaadid

Antud peatükis vaadeldakse kuidas saab arvutada roboti iga telje pöördnurkasid post protesseerimise jooksul. Seda arvutust nimetatakse kinemaatika mudeli pöörd lahendamiseks. See arvutus on mahukas nii matemaatika kui ka kontrolleri ressursside poolt. Lahenduse leidmine võtab palju aega, eriti siis kui tegemist on *realtime* roboti juhtimisega. Post protsessor saab koordinaatide väärtusi Cartesian'i koordinaadistikust, kuid aga robotil realselt on pöördteljed. Lihtsamaks arusaamiseks tuleb nimetada roboti telge *actuator*-iks, ning roboti tööriista keskpunkti *efector*-iks. Vaatamata arvutusraskustele, on antud programmeerimisviis kiirelt toimetav, ning vaadates juhtimissüsteemi ressursside kasutamise poolt, väga efektiivne. Pöörd lahendus praktiliselt tähendab seda, et on teada kus asub tööriist aga roboti juhtimiskontroller ei tea mis nurkade alla on pööratud iga roboti manipulaatori telg. Matemaatiliselt võib antud ülesande esitada nagu all toodud võrrandiga [26]:

$$A_1(\theta_1) \cdot A_2(\theta_2) \cdot A_3(\theta_3) \cdot A_4(\theta_4) \cdot A_5(\theta_5) \cdot A_6(\theta_6) = {}_6^0T, \quad (5.2)$$

kus $A_{1...6}$ – iga roboti telje pöördmaatriksd, *afektroid*,

$\theta_{1...6}$ – telje pöördnurk,

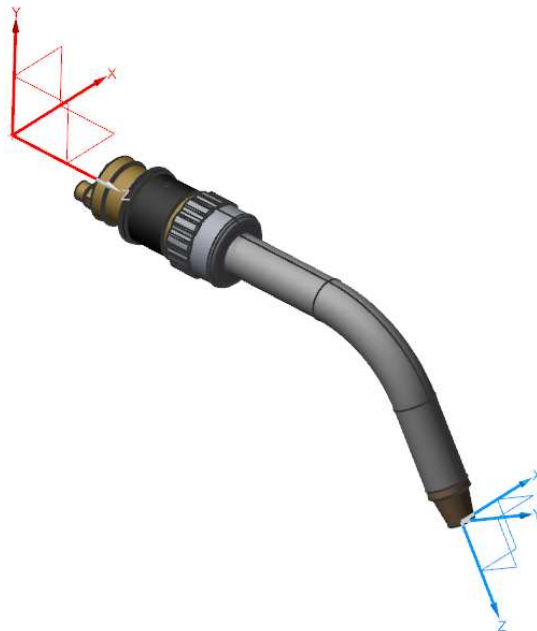
${}_6^0T$ – roboti tööriista transformatsioonmaatriks, *efektor*.

Eespool toodud valemil (5.2) kinemaatika mudeli pöörd lahenduse jooksul on teadmata vasak osa. Pöördnurkade leidmist tuleb alustada kinemaatika mudeli koostamisest. Roboti

kinemaatika mudeli loomiseks on vaja teada mitte ainult manipulaatori füüsilisi parameetreid, kui ka tööriista parameetreid. Motomani tööriista parameetreid saab salvestada eraldi faili – saab kasutada sama lahendust mida on kirjeldatud ABB robotite puhul (vt. Pt 4.1). Failis peab olema salvestatud tööriista keskpunkti asukoht roboti kinnitusplaadi koordinaadistikus.

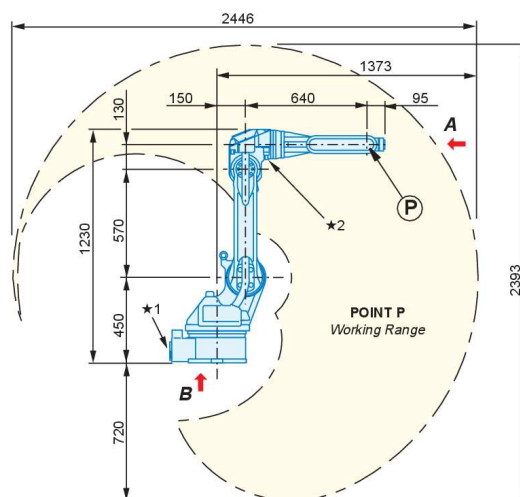
WELDGUN1, 0, -80, 260, 60, 0, 0

Kui täpsemalt vaadelda eelpool toodud näidist, siis tööriista nimetus on *WELDGUN1*, selle tööriista keskpunkt tööriista kinnitusplaadi koordinaadistikus asub 260 mm mööda Z-telge, -50 mm mööda Y-telge ning ei ole nihutatud mööda X-telge. Tööriista keskpunkti koordinaadistik on pööratud ümber X-telge 60° võrra. Visuaalselt on antud parameetrid näidatud joonisel 5.3.



Joonis 5.3. Kaarkeevituse roboti tööriist

Antud parameetreid kasutatakse edaspidi kinemaatilise mudeli arvutamiseks. Joonisel 5.3 on näidetud sinise värviga *efector*-i koordinaadistik juhul kui robotil on tööriist. Kui robotil ei oleks tööriista, siis *efector*-iks võib nimetada viimast *afector*-it. Selle koordinaadistik on punast värvi.



Joonis 5.4. Yaskawa Motoman UP6 kinemaatika mudel

Vaadates joonist 5.4 saab koostada Denavit Hartenberg-i meetodi abil parameetrite tabeli (vt tabel 5.2), mis võimaldab lahendada kinemaatika ülesande. Lahenduse lihtsustamiseks tööriista parameetreid ei arvestata.

Tabel 5.2. Yaskawa Motoman UP6 DH parameetrite tabel

Telg	θ_i	α_i	a_i [mm]	d_i [mm]
S	θ_1	-90°	150	0
L	θ_2	0°	570	0
U	θ_3	-90°	130	0
R	θ_4	90°	0	640
B	θ_5	-90°	0	0
T	θ_6	0°	-95	0

Denavit Hartenberg-i meetodi järgi koostatud tabeli (vt. Tabel 5.2) alusel võib leida allpool toodud rotatsioonimaatriksid.

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_i \\ \sin \theta_i \cdot \cos \alpha_i & \cos \theta_i \cdot \cos \alpha_i & -\sin \alpha_i & -\sin \alpha_i \cdot d_i \\ \sin \theta_i \cdot \sin \alpha_i & \cos \theta_i \cdot \sin \alpha_i & \cos \alpha_i & \cos \alpha_i \cdot d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$$A_1 = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 & 0 & 150 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_1 & -\cos \theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 570 \\ \sin \theta_2 & \cos \theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$A_3 = \begin{bmatrix} \cos \theta_3 & \sin \theta_3 & 0 & 130 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_3 & -\cos \theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

$$A_4 = \begin{bmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ 0 & 0 & -1 & -640 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

$$A_5 = \begin{bmatrix} \cos \theta_5 & \sin \theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin \theta_5 & -\cos \theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

$$A_6 = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 95 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Kui roboti iga telje rotatsioonmaatriks on tuletatud, siis võib tuletada rotatsioonmaatriksi roboti koordinaatsüsteemist roboti tööriista keskpunktini, ehk 0_6T .

$${}^0_6T = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (5.10)$$

0_6T on transformatsiooni maatriks suurusega 4x4. Kuna maatriksi elemendid on pikad, siis seda võib seda maatriksit esitada järgmisel viisil:

$${}^0_6T = \begin{bmatrix} R_{3 \times 3} & O_{1 \times 3} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.11)$$

kus R – maatriks 3x3 mis kirjeldab tööriista baasi pöördnurkasid,

O – maatriks 3x1 mis kirjeldab tööriista koordinaadi keskpunkti.

Kui vaadata sama maatriksit elementide tasemel, siis võib 0_6T transformatsiooni maatriksi kirjutada järgmisel viisil

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & O_x \\ r_{21} & r_{22} & r_{23} & O_y \\ r_{31} & r_{32} & r_{33} & O_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5.12)$$

kus elemendid võib esitada sümbolsealt. Lisas 9 on lahti kirjutatud kasutatud lühendeid, näiteks s_3 tähendab $\sin(\theta_3)$, c_6 on $\cos(\theta_6)$ jne. Antud võrrandid kuuluvad n.n. otsese-kinemaatika ülesande lahendusele. See tähendab, et robot teab iga manipulaatori pöördtelje nurgad, aga *TCP* otsa koordinaadid pole teada. Otsese- ja pöördkinemaatika mudelid on omavahel seotud, ning mõlemad on väga olulised roboti juhtimise jaoks. Pöördkinemaatika mudeli lahenduses probleemiks on see, et robot teab tööriista keskpunkti asukohta, aga peab arvutama iga liigendi pöördtelje nurga. Selleks on vaja esialgselt selgeks teha, kuhu peab roboti tööriist lõpuks jõudma.

Üldjuhul kasutatakse nurkade leidmiseks trigonomeetrilist funktsiooni *atan*. Antud töös kasutatakse *atan2*. See funktsioon on olemas ka TCL/TK programmeerimiskeeles. Selles keeles on kirjutatud Siemens NX-i post protsessorid *atan2* funktsiooni valem 5.14 on:

$$\text{atan2}(x, y) = 2 \cdot \arctan \frac{y}{\sqrt{x^2 + y^2} + x}, \quad (5.14)$$

kus x – esimene muutuja,
 y – teine muutuja.

Antud funktsioon annab tulemuse neljas kvadrantis. See on tavaline arc-tan funktsioon kahe parameetritega. Kahe parameetri mõte on saada tulemus õiges kvadrantis. Tagastatav tulemus on radiaanides, ning see kirjeldab nurka X-telje positiivse suuna ning nullpunktist arvatava punktisse suunatud vektori vahel. Allpool toodud tabelis võib vaadata, mis tulemust võiks oodata *atan2* funktsiooni kasutades.

Tabel 5.3. Atan2 tulemuste tabel.

X-param	Y-param	Tulemus
$x > 0$	$y \geq 0$	$\text{atan}(y/x)$
$x > 0$	$y < 0$	$\text{atan}(y/x)$
$x < 0$	$y \geq 0$	$\text{atan}(y/x) + \pi$
$x < 0$	$y < 0$	$\text{atan}(y/x) - \pi$
$x = 0$	$y > 0$	$\pi/2$
$x = 0$	$y < 0$	$-\pi/2$
$x = 0$	$y = 0$	\emptyset

DH-meetodi lahenduseks tuleb koostada 1x3 maatriksi. Määratud sihtpunkti X, Y ja Z koordinaatidest tuleb lahutada d_6 ja 0_6T maatriksi korrutised (vt.valem 5.16).

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} X - d_6 \cdot r_{13} \\ Y - d_6 \cdot r_{23} \\ Z - d_6 \cdot r_{33} \end{bmatrix} \quad (5.16)$$

Kuna d_6 võrdub 0 (vt. Tabel 5.2), siis x_c võrdub X, y_c võrdub Y ning z_c numbriliselt võrdub Z. Siis jäävad pöördnurgad A, B ning C. A võrdub r_{13} , B võrdub r_{23} ning C võrdub r_{33} . Siis roboti pöördtelgede nurgad saab leida järgmiste võrrandite järgi [27].

$$\theta_1 = \text{atan2}(x_c, y_c) \quad (5.17)$$

$$\theta_2 = \text{atan2}(\sqrt{x_c^2 + y_c^2 - d_1^2}, z_c - d_1) \quad (5.18)$$

$$D = \frac{x_c^2 + y_c^2 - d_1^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_3} \quad (5.19)$$

$$\theta_3 = \text{atan2}(D, \sqrt{|1 - D^2|}) \quad (5.20)$$

$$s1 = \sin \theta_1 \quad (5.21)$$

$$c1 = \cos \theta_1 \quad (5.22)$$

$$s23 = \sin(\theta_2 + \theta_3) \quad (5.23)$$

$$c23 = \cos(\theta_2 + \theta_3) \quad (5.24)$$

$$\begin{aligned} &\theta_4 \\ &= \text{atan2}(c1 \cdot c23 \cdot r_{13} + s1 \cdot c23 \cdot r_{23} + s23 \cdot r_{33}, -c1s23 \cdot r_{13} + s1 \cdot s23 \cdot r_{23} + c23 \cdot \\ &\cdot r_{33}) \end{aligned} \quad (5.25)$$

$$\theta_5 = \text{atan2}(s1 \cdot r_{13} - c1 \cdot r_{23}, \sqrt{1 - (s1 \cdot r_{13} - c1 \cdot r_{23})^2}) \quad (5.26)$$

$$\theta_6 = \text{atan2}(-s1 \cdot r_{13} + c1 \cdot r_{23}, s1 \cdot r_{13} + c1 \cdot r_{23}) \quad (5.27)$$

Kasutades üleval toodud valemeid võib arvutada roboti telgede pöördnurgad punktis. Näiteks, kui robot oleks null positsioonil, ning tööriista koordinaadistik oleks detaili koordinaadistikuga samades suundades, siis manipulaatori nurgad on esitatud tabelis 5.4:

Tabel 5.4. Roboti asend 0 punktis.

Telg	S	L	U	R	B	T
Nurk [PULSE]	12885	42070	35930	-7631	96832	67169

Antud programmeerimisviis on kindlasti kõige kiirem roboti jaoks kui vaadata andmete töötlemise poolt. Aga antud meetod ei ole optimaalne, kuna iga roboti tootja optimeerib arvutusalgortime. Seetõttu toodud arvutusmeetod on mitte efektiivne. Antud juhul pöördtelgede nurkade arvutamine on teoreetiline, kuna realiseerimisel post protsessoris võivad tekkida vead arvutustes. Teiseks, kuna post protsessoris kasutatakse konkreetse roboti kinemaatika mudelit, siis ei ole võimalik kasutada seda post protsessorit teiste robotite jaoks. Juhul kui roboti juhtimissüsteem arvutab telgede koordinaate ise ning korrigeerib vastavalt tööriista andmetele, siis post protsessori saab kasutada praktiliselt kõikide sama juhtimissüsteemiga robotitel.

5.3 Liikumise interpoleerimine

Nagu ABB robotitel, sarnaselt ka Yakisawa Motomani robotitega on olemas 3 põhi liikumiskäsu.

- MovJ – ühenduste liikumine. Antud juhul robot liigub mööda telge vastavalt programmeeritud kiirusele, sõltumatu teistest telgedest.
- MovL – lineaarne interpoleerimine. Selle käsu täitmisel robot liigutab tööriista keskpunkti lineaarselt rakendades igale teljele oma kiiruse ning kiirenduse.
- MovC – liikumine kaarega. Põhimõte on sama nagu lineaarse liikumise puhul. Vajab 3 punkti kaare defineerimiseks.

Kuna esimesed kaks liikumisviisi on lahti kirjutatud pt. 4.4 siis ei ole mõtet nendest uuesti rääkida, kuna põhimõte on sama. Erinev on aga kaar interpoleerimine. ABB roboti juhtimissüsteem vajab kaarega liikumiseks lõpp-punkti ning kaare keskpunkti. Yakisawa Motomani robotid kasutavad teist viisi – ring interpoleerimiseks kasutatakse kolm punkti, mille põhjal arvutatakse kaare parameetreid. Siemens NX post protsesseerimise jaoks on lihtsam realiseeritav just ABB robotitel kasutatav variant. Kuna post protsessori kasutusel on olemas sellised süsteemi muutujad nagu *mom_pos()* ning *mom_arc_center()*, mis tagastab kaare keskpunkti asukoha. Yakisawa Motomani kaarel liikumiseks on vaja Siemens NX post protsessoris luua eraldi funktsioon, mis väljastaks kolm punkti: kaare algpunkt, kaare lõpp-punkt ning punkt kaare keskel.

Kaarel oleva punkti arvutamiseks tuleb seadistada post protsessori nii, et ring oleks jagatud kvadrantideks ehk neljaks osaks. See on standardne käsk post protsessori loomise keskkonnas. Järgmine etapp on arvutada välja punkt, mis asub kaare peal ning jagab selle kaare võrdselt pooleks. Et seda arvutada on vaja teada kaare raadiust ning vähemalt ühte koordinaadi, antud juhul X-telje väärtust. Siis kasutades alltoodud valemit 5.28, mis kirjeldab ringi:

$$R^2 = (X - X_0)^2 + (Y - Y_0)^2, \quad (5.28)$$

kus R – kaare raadius,

X_0, Y_0 – kaare keskpunkti koordinaadid,

X, Y – kaare peal oleva punkti koordinaadid.

Kaare arvutuses ei kasutata Z-telje komponenti, kuna ringi interpoleerimine on lubatud ainult XY-tasapinnas. Üleval toodud valemist 5.28 on vaja tuletada valem, punkti Y-koordinaadi arvutamiseks:

$$Y = \pm\sqrt{R - (X - X_0)^2} + Y_0 \quad (5.29)$$

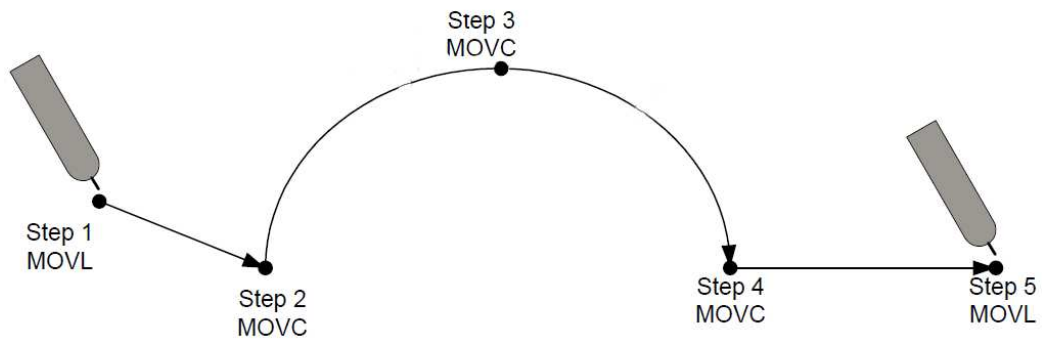
Antud juhul tekib olukord kus saavutatud tulemus ei ole selgelt defineeritud, kuna ruudus olev väärtus võib olla erinev. Et valida õiget suurust, on vaja kirjutada funktsioon, mis võrdleks Y-komponendi alguspunkti ning lõpp-punkti vahel. Kui $mom_previous_pos(1)$ on väiksem kui $mom_pos(1)$, siis tulemus on suurem kui 0. Kui aga vastupidi, siis tulemust on vaja korrutada -1. Post-protssessoris antud algoritm näeb välja järgmiselt [20]:

```

set R $mom_arc_radius
set Y [expr sqrt(R-pow(($mom_pos(1)/
-$mom_previous_pos(1))/2,2))+$mom_arc_center(1)]
if {$mom_pos(1)<$mom_previous_pos(1)} {
    set Y [expr Y*-1]
}

```

Leitud andmeid saab kasutada kaarel oleva punkti koordinaadi arvutamiseks. Ja siis kui kaar jätkub teisesse kvadranti, siis selle kvadranti kaare lõpp-punkti koordinaadid saavad järgmise kaare algpunktiks. Reaalselt aga robotile ei ole vaja nii palju punkte. Kuid kui kaar jätkub, siis punktide maht võib olla suurem. All oleva joonisel 5.3 on näidatud, kuidas toimub kaarinterpoleerimine. Punkti C0001 ja C0002 vahel roboti tööriist liigub lineaarselt isegi siis, kui mitte kasutada MOVL käsku. Motoman-i juhtimissüsteem muudab liikumisinterpoleerimist automaatselt



Joonis 5.5. Kaarinterpoleerimine

All on toodud juhtimisprogrammi lõige joonisele 5.5.

```

MOVL VL=200 PL=0 C0001 'Step 1
MOV C V=200 C0002 'Step 2, ettenihe 200cm/min
MOV C C0003 'Step 3
MOV C C0004 'Step 4
MOVL C0005 'Step 5

```

Kokkuvõtteks võib öelda, et Motomani roboti post protsessori kirjutamine on keerulisem ja vajab rohkem funktsioone andmete töötlemiseks. Programmeerimise meetodite erinevused nagu ring interpoleerimine raskendab post protsessori loomist kui ka aeglustab juhtimisprogrammi genereerimist. Teiseks, roboti pöördtelgede nurkade arvutus ei ole efektiivne. Sellisel juhul kaob võimalus kasutada sama post protsessorit, et genereerida juhtimisprogramme erinevate Motomani robotitele.

6. KOKKUVÕTTE

Tänapäeval firmad hakkavad rohkem kasutama automatiseeritud tööjõudu nagu roboteid. Robotid täidavad erinevaid ülesandeid nagu värvimine, pakkimine, sorteerimine, maisna teenindamine, keevitamine jne. Kõige rohkem roboteid kasutatakse keevitamiseks. Selliste robotite ülesandeks on kvaliteetselt keevitada masstootmis toodangut. Üldjuhul teeb robot kvaliteetsemat keevitusõmblust kui inimene. Robotkeevitamine vajab aga programmeerijat spetsiaalsete oskuste ja teadmistega keevituse valdkonnas. Vastav inimene peab tundma keevitust ja oskama seda rakendada robotiga, koostades vastavaid juhtimisprogramme robotile. Kuna neid inimesi on vähe, otsivad firmad võimalusi roboti juhtimisprogrammi automatiseerimiseks loomiseks. Seda probleemi on teadvustanud endale nii roboti kasutajad (roboti programmeerijad), kui ka roboti müüjad. Otsitakse erinevaid tarkvara lahendusi, mis võiksid luua roboti juhtimisprogramme 3D mudelite alusel, et vähendada roboti juhtimisprogrammi loomiseks ja testimiseks kuluvat aega. Selleks pakuvad roboti tootjad oma tarkvarasid, nagu RobotStudio ABB-lt. Antud tarkvarad on võimelised genereerima erineva keerukusega juhtimisprogramme. Kahjuks puudub neil võimalus uuendada automaatselt muudetud detaili 3D kujutist ning muuta roboti juhtimisprogrammi vastavalt detaili tehtud muudatustele. Enamasti peab programmeerija tooma muudatud detaili kujutise roboti programmeerimistarkvarasse ja looma roboti juhtimisprogrammi otsast peale. Sellega kaasneb tohutu ajakulu. Samas on olemas kombineeritud tarkvarad detailide modelleerimiseks, joonestamiseks ning CNC-pinkidele juhtimisprogrammi genereerimiseks. Plussideks on ajakulu kokkuhoidmine tänu suurele tarkvara moodulite omavahelisele integratsioonile. Juhtimisprogrammi saab luua praktiliselt automaatselt, inimene peab määrama tarkvaras detaili, vajadusel ka rakise ning null-punkti. Selle alusel genereerib tarkvara juhtimisprogrammi ise. Selliste tarkvarade hulka kulub ka Siemens NX. See on võimeline looma juhtimisprogramme masinatele millel on kuni 5 liikuvat telge. Kaasaegsetel robotitel on tavaliselt 6 või isegi rohkem liikuvat telge.

Lõputöös töötati välja lahendust 6-teljelist robotite juhtimisprogrammide automaatseks genereerimiseks kasutades Siemens NX tarkvara post protsessori võimlaust. Lahendust katsetati ABB IRB 1600, mis asub TTÜ Energeetikateaduskonna laboris, ja Motomani roboti peal, mis asub BLTR Masinaehituse AS-s. Sai lahendatud mõned probleemid seoses Siemens NX piiratud võimalustega programmeerida kuuendat telge. Selleks kasutati diferentsiaalset meetodit roboti küljes oleva tööriista edasise liikumissuuna määramiseks. See funktsioon sai nime *TrendLine*. Funktsiooni on kõige mõistlikum rakendada keevitusrobotite puhul.

Teiseks probleemiks oli kvaternionide arvestus ABB roboti jaoks. Kvaternionidega saab esitada vektori suunda 3D keskkonnas. ABB roboti kasutus manuaalis on ära toodud võrrandid, mis võimaldavad arvutada kvaternionide väärtused punkti pöördnurakde järgi. Neid võrrandeid sai rakendatud ABB roboti IRB1600 jaoks Siemensi NX post protsessoris.

Viimane probleem oli seotatud manipulaatori pöördtelgede nurkade leidmisega Motomani robotitele. Lõputöös lahendus näitab kuidas roboti juhtimissüsteem arvutab manipulaatori pöördtelgede nurgad nii et tööriista keskpunkt oleks määratud punktis. Selleks kasutati Denavit–Hartenberg-i meetodit. DH-tabelis pannakse kirja konkreetse roboti kinemaatika parameetrid, nagu ühenduste pikkused ning pöördtelgede omavaheline asukoht. Roboti tootjad on optimeerinud seda algoritmi nii, et punkti asukoha konverteerimine roboti juhtimissüsteemis annab tunduvalt parema tulemuse, kui arvutada seda post protsessori abil. DH arvutusmeetodit on võimalik rakendada ainult konkreetse roboti puhul, kuna iga roboti mudeli parameetrid on erinevad. See teeb post protsessorit mitte universaalseks.

6. telje programmeerimiseks, mida Siemens NX ei võimalda leida automaatselt, loodi funktsioon nimega *TrendLine*. See arvutab diferentsiaalselt pöördnurka ning on kasutatav nii ABB robotite kui ka Motomani robotite juhtimisprogrammi loomiseks. Roboti juhtimisprogrammi loomiseks olid kasutatud universaalsed algoritmid juhtimisprogrammi andmete grupeerimiseks. NX-s loodi lisafunktsioonid *WAIT* ja *TrendLine* seadistamiseks. Neid saab rakendada ainult roboti töö protseduuri alguses või lõpus. Siemens NX-s on iga operatsioon jagatud veel kolmeks loogiliseks osaks: lähenemine, tööoperatsiooni trajektoor ning eemaldumine detailist.

Post protsessori abil genereeritud juhtimisprogramme testisin nii reaalse kui ka virtuaalse roboti peal. Virtuaalse testimise jaoks kasutasin ABB tarkvara RobotStudio. Et simuleerimise tulemus oleks võimalikult reaalne, sai koostatud 3D mudelid reaalsest roboti tööriistast ning töödeldavast detailist. Probleem tekkis aga Yakisawa Motoman-iga, mille testimis tarkvara saab kasutada ainult siis, kui on ostetud robot ning seda ka ainult lisatasu eest.

Lõputöös koostatud algoritmid Siemensi NX post protsessori loomiseks on kasutatavad ka teiste tarkvarade juures. Tulevikus võib arendada edasi *TrendLine* funktsiooni, et vähendada viivtust kahe punkti vahel vektori suuna arvutamiseks. Praegu arvutatakse see eelmise ja hetkpunkti lahutuse alusel. Funktsiooni saab arvutada ka tuleva ja hetkpunkti lahutise põhjal. See võib anda täpsema tulemuse.

KASUTATUD KIRJANDUS

1. iMB koduleht <http://mbinfo.mbdesign.net/CAD1970.htm>. Vaadatud 1.05.2014. a.
2. iMB koduleht <http://mbinfo.mbdesign.net/CAD1980.htm>. Vaadatud 1.05.2014. a.
3. iMB koduleht <http://mbinfo.mbdesign.net/CAD1992-93.htm>. Vaadatud 1.05.2014. a.
4. iMB koduleht <http://mbinfo.mbdesign.net/CAD2000.htm> Vaadatud 1.05.2014. a.
5. NX Overview – Siemens Product Lifecycle Management, 2009. – 22 lk.
6. NX CAE – Siemens PLM Software 2014. – 16 lk.
7. NX Flow – Siemens PLM Software 2014. – 3 lk.
8. High productivity part manufacturing – Siemens PLM Software 2014 – 20 lk.
9. Clarke, C. Developments in the CAM software market. – TCT – 2011. – 54 lk.
10. High productivity part manufacturing – Siemens PLM Software 2014. – 20 lk.
11. RAPID Reference manual – ABB Robotics Products AB 2008. – 538 lk.
12. NX CAM 9 – Siemens PLM Software 2014. – 9 lk.
13. Намов, G. Математика. Часть I. Линейная алгебра и аналитическая геометрия: Учебное пособие - РГПУ им. А.И. Герцена – 2004 – 149 lk.
14. Lehtla, T. Robotitehnika – TTÜ Elektriainstituut ja jõuelektronika instituut – 2008 – 201 lk
15. Murray, Richard M. A Mathematical Introduction to Robotic Manipulation – CRC Press – 1994 – 456 lk.
16. Dr. Finan, Marcel B. Multiplication of Vectors: The Scalar or Dot Product – Arkansas Tech University – 2004 – 7 lk.
17. NX User Defined Features - Siemens PLM Software 2007. – 3 lk.
18. Klimov, A., Mašinin N. – Роботизированные технологические комплексы и автоматические линии в сварке – Лань – 2011 – 234 lk.
19. Tcl Build-In Commands koduleht <http://www.tcl.tk/man/tcl8.4/TclCmd/expr.htm>. Vaadatud 8.06.2014. a.
20. Tcl Build-In Commands koduleht <http://www.tcl.tk/man/tcl8.4/TclCmd/if.htm>. Vaadatud 8.06.2014. a.
21. Industrial Robot MOTOMAN-UP6 – Motoman Robotics Europe AB 2008 – 2 lk.
22. XCR Instructions Manual – Motoman Robotics Europe AB 2008 – 375 lk.
23. NX100 Robot Controller – Motoman Robotics Europe AB 2008 – 3 lk
24. DX200 Robot Controller – Motoman Robotics Europe AB 2014 – 2 lk
25. NX100 INFORM MANUAL – Motoman Electric Corporation 2004 – 297 lk.

26. Kucuk, S., Bingul, Z. – Robot Kinematics: Forward and Inverse Kinematics - InTech Europe – 2008 – 33 lk.
27. Dr. Verma, A. – FORWARD KINEMATICS ANALYSIS OF 6-DOF ARC WELDING ROBOT – G. H. Patel College of Engineering & Technology – 2010 – 5 lk.

LISAD

Lisa 1. ABB IRB1600 roboti spetsifikatsioon

Main applications

Machine tending, material handling, arc welding, cutting, dispensing, assembly, palletizing and packaging, measuring, die casting, injection moulding

Specification

Variants	Reach(m)	Payload(kg)	Armload(kg)
IRB 1600-6/1.2	1.2	6	30,5
IRB 1600-6/1.45	1.45	6	30,5
IRB 1600-10/1.2	1.2	10	20,5
IRB 1600-10/1.45	1.45	10	20,5
Number of axes	6+3 external (up to 36 with MultiMove)		
Protection	Standard IP54; opt. FoundryPlus 2 (IP 67)		
Mounting	Floor, wall, shelf, tilted, inverted		
IRC5 Controller variants	Single cabinet, Dual cabinet, Compact		

Physical

Dimensions robot base:	484 x 648
Robot height: IRB 1600-6/1.2 and IRB 1600-10/1.2	1069 mm
Robot height: IRB 1600-6/1.45 and IRB 1600-10/1.45	1294 mm
Robot weight:	250 kg

Performance (according to ISO 9283)

	6/1.2	6/1.45	10/1.2	10/1.45
Pos. repeatability (RP)	0.02 mm	0.02 mm	0.02 mm	0.05 mm
Path repeatability (RT)	0.13 mm	0.19 mm	0.06 mm	0.13 mm

Movement

Working range	1.2	1.45
Axis 1	+180° to -180°	+180° to -180°
Axis 2	+136° to -63°	+150° to -90°
Axis 3	+55° to -235°	+65° to -245°
Axis 4	+200° to -200° def. +/-190° revolution	+200° to -200° def. +/-190° revolution
Axis 5	+115° to -115°	+115° to -115°
Axis 6	+400° to -400° def. +/-288 revolution	+400° to -400° def. +/-288 revolution
Maximum Speed	6 kg	10 kg
Axis 1	150°/s	180°/s
Axis 2	160°/s	180°/s
Axis 3	170°/s	185°/s
Axis 4	320°/s	385°/s
Axis 5	400°/s	400°/s
Axis 6	460°/s	460°/s

Electrical connections

Supply voltage	200-600 V, 50-60 Hz
Power consumption	ISO-Cube at max speed 0.58 kW

Environment

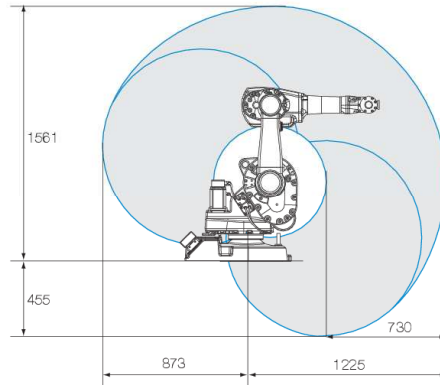
Ambient temperature for mechanical unit:

During operation	+ 5°C (41°F) to + 45°C (113°F)
During transportation and storage	- 25°C (- 13°F) to + 55°C (131°F)
For short periods (max 24h)	up to + 70°C (158°F)
Relative humidity	Max. 95% at constant temperature
Safety	Double circuits with supervisions, emergency stops and safety functions, 3-position enable device
Emission	EMC/EMI shielded

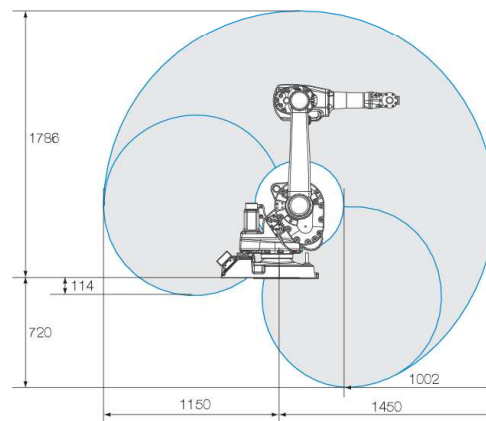
Data and dimensions may be changed without notice

Working range

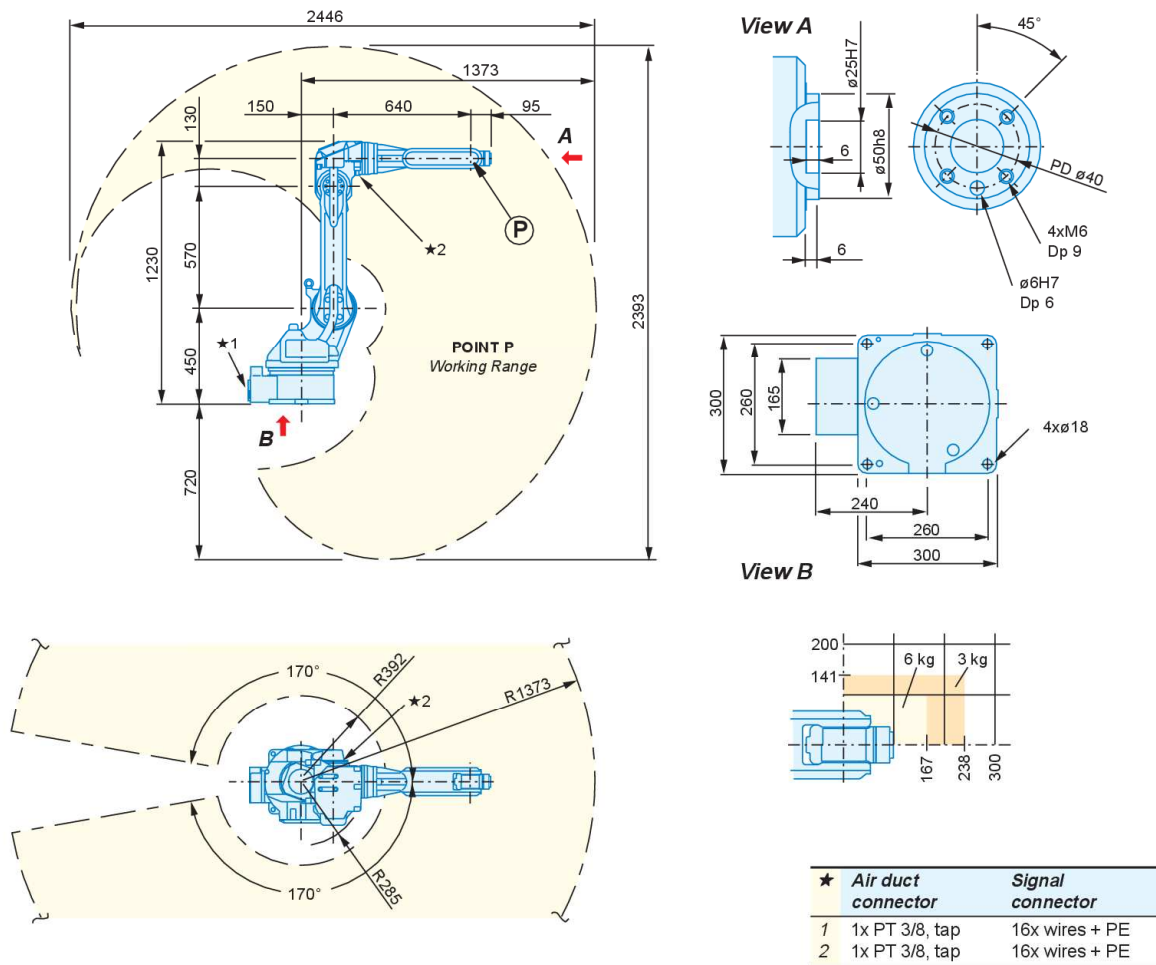
IRB 1600-x/1.2



IRB 1600-x/1.45



Lisa 2. Yakisawa Motoman UP6 spetsifikatsioon.



Specifications

Controlled Axes	6	
Payload	6 kg	
Repetitive Positioning Accuracy	±0.08 mm	
Motion Range	S-axis (turning)	±170°
	L-axis (lower arm)	+155°, -90°
	U-axis (upper arm)	+190°, -170°
	R-axis (wrist roll)	±180°
	B-axis (wrist pitch)	±135°
Maximum Speed	T-axis (wrist twist)	±360°
	S-axis	140°/s
	L-axis	160°/s
	U-axis	170°/s
	R-axis	335°/s
B-axis	335°/s	
T-axis	500°/s	

Weight	130 kgs	
Power Supply	2 kVA	
Allowable Moment	R-axis	11.8 Nm (1.2 kgf•m)
	B-axis	9.8 Nm (1.0 kgf•m)
	T-axis	5.9 Nm (0.6 kgf•m)
Allowable Inertia	R-axis	0.24 kg•m ²
	B-axis	0.17 kg•m ²
	T-axis	0.06 kg•m ²
Ambient Conditions	Temperature	0 to 45°C
	Rel. Humidity	20 to 80% RH
	Vibration	0.5G or less
Others	<ul style="list-style-type: none"> • Free from excessive electrical noise. • Free from corrosive gas or liquid, or explosive gas. 	

Lisa 3. Intervjuu robotite müügiesindajaga, OMD OÜ

Tutvustage end palun, kes Teie olete, millega tegelete?

Tere, minu nimi on Andrei Maksimov, olen müügiesindaja ning peaspetsialist firmas OMD OÜ, Meie poolt pakutud teenused on tööstusrobotite integreerimine tootmisesse ning erilahenduse masinate väljatöötlemine.

Mis roboteit Teie müüte?

Meie firma pakub roboteid tootjatelt Motoman, ABB, Hyundai ja Universal.

Aga kes pakub Fanuc-it, või siis Kuka?

Fanuc-eid Eestis ei ole, Kuka - Tehnogrupp

Palju roboteid olete müünud Eestis? Mis robotid need olid? (keevitus-, frees-, või muud robotid, mis firma omad)

Kokku on umbes 25 manipulaatorit. Enamus nendest on ABB – neid sai müüdnud 20 tk., siis 4 tk Hyndai robotit ja üks Motoman. Eestis on kõige levinum robotitootja on ABB, teine on Motoman, aga Motoman pakub ise oma roboteid ja eriti ei taha töötada edasimüüjatega, eriti nendega, kes müüvad mitme firma roboteid.

Olete müüdnud roboteid ka välismaale? Kui jah, siis kui palju, milliseid?

Enne jah, põhiliselt ABB.

Oskate öelda ABB robotite plussid ja miinused.

Klienditugi on parem, aga Motoman ja ABB on üldiselt võrdsed. RAPID on tunduvalt võimsam ja parem programmeerimiskeel erilahenduste jaoks võrreldes sama Motomani INFORM-iga. Lisaks ABB-l on parem realiseeritud multifunktsionaalsus.

Aga mis Motomani robotite plussid ja miinused.

Seda on päris raske öelda... Võib olla hinnapoliitika. Aga miinuseks on kindlasti see, et robotite CAD mudelid on tasulised ja neid võib osta ainult peale roboti ostmist.

Kuidas saab roboteid programmeerida ja kuidas enamuse teie kliente teevad juhtimisprogramme?

Praktilised kõik meie kliendid kasutavad ONLINE programmeerimisviisi – ehk otse konsoolist. 2 firmat kasutavad sellest spetsiaalset tarkvara. Need 2 firmat kasutavad roboteid 3D-metalli lõikamiseks. Tarkvara on isenesest päris kallis ja see mida pakuvad ABB või Motoman on ainult nende roboti jaoks. Ei ole mingit universaalsust.

Olete kuulnud ka võimalusest programmeerida roboteid mitte ainult tootja poolt antud tarkvaraga?

Jah, olen kuulnud. Tean kindlasti, et on võimalus konverteerida tavalise CNC pingi G-koodi roboti juhtimisprogrammiks. Selleks on isegi mõned konverterid olemas. 1 ostis värvitusroboti jaoks sellist konverterit.

Mis on teie nägemused robotite turu perspektiivid?

Praegu on umbes 80% keevitusrobotit, ülejäänud on sorterimis, loikamis või freesimisrobotid. Üks robot on painutuspingi abinõu.

Teie arvamus minu magistri lõputöö teemale?

Nii ja naa... Protode valmistamiseks on päris hea lahendus. Aga liini programmeerimine on päris raske kuna vajab spetsiaalseid funktsioone, mida võib mitte olla teie poolt valitud tarkvaras.

Lisa 4. Intervjuu robotkeevituse tööde juhiga, BLTR Masinaehitus

Tutvustage end palun, kes Teie olete, millega tegelete?

Tere, minu nimi on Valeri Kizikov, olen peainsener firmas BLTR Masinaehitus.

Mis robotit teie kasutate?

Meie kasutame keevitus robotit Motoman UP6 koos pöördlauaga.

Kuidas teie valite, mis tööd pannakse robotile?

Robotiseritud keevitust meie kasutame siis, kui on oluline keevitustäpsus ning kõrge kvaliteet või väga suured kogused detaile rutiinsete keevutsõmbustega. Kindlasti detail või õigemini öelda koost peab mahtuma ka gabariitmõõtu: 1200x680x250 mm ning kaal alla 1500 kg. Detaili pannakse robotile isegi siis, kui see on üksik tellimus, aga kõrge keevitusnõudmistega.

Kuidas teie programmeerite oma robotit?

Meie insener programmeerib seda robotit otse konsoolist.

Palju aega kulub et koostada programmi?

Aega kulub päris palju. Umbes 60% sellest ajast kulub programmi kirjutamiseks. Ülejäänud osa kulub juhtimisprogrammi parameetrite optimeerimiseks.

Mis võimalused teie näete keevitusroboti töö efektiivsuse tõstmiseks?

Kindlasti oleks hea, kui insener saaks luua juhtimisprogrammi samal ajal, kui robot töötab. Aga hetkel meil ei ole sellist tarkvara.

On teil plaanides osta veel roboteid? Kui jah, siis mis roboteid teie vajate?

Pole siia maani plaane olnud, kuna mahud ei ole niivõrd suured. Aga ausalt öeldes võiks seda robotit uuendada.

Teie arvamus minu magistri lõputöö teemale?

Meie tootmisele see sobiks. Eriti see omadus, et konstruktor saaks teha rakise mudelit ning kohe kirjutada ka juhtimisprogrammi kasutades sama tarkvara. Teiseks kindlasti tahaks mainida võimalust saavutada uuendatud programmi, kui näiteks pidime rakist muutma.

Lisa 5. Koordinaadi arvutus Siemens NX post protsessoris

```
#=====
proc PB_CMD_COORDINATE_CALCULATION { } {
#=====
# Target point calculation process #
global mom_pos A B C X Y Z q Pi mom_trendline_correction
mom_trendline_status

# Calculation Pi-value
if { ![info exists Pi] } {
    set Pi [expr 4*tan(1.0)]
}

# Calculating corrected X-,Y-,Z-coordinates
set X $mom_pos(0)
set Y [expr -1*$mom_pos(1)]
set Z $mom_pos(2)

# Converting angle degrees to radians
set A [expr $mom_pos(3)*$Pi/180]
set B [expr ($mom_pos(4)+180)*$Pi/180]
if { ![info exists mom_trendline_correction] } {
    set C 0
} else {
    set C [expr 0+$mom_trendline_correction*$Pi/180]
}

# Recalculating C angle using "TRENDLINE"
if {[info exists trendline_status] && [string match
$mom_trendline_status "ON"]} {
#    MOM_output_literal "TRENDLINE IS ON"
    PB_CMD_TRENDLINE
} else {
#    MOM_output_literal "TRENDLINE IS OFF"
}

# Calculation quaternions
set q(0) [expr sqrt((cos($B)*cos($C)-
sin($A)*sin($B)*sin($C)+cos($A)*cos($C)+cos($A)*cos($B)+1))/2]

set q(1) [expr abs(sqrt((cos($B)*cos($C)+sin($A)*sin($B)*sin($C)-
cos($A)*cos($C)-cos($A)*cos($B)+1))/2)]
    if {[expr
sin($A)*sin($B)*sin($C)+sin($A)*cos($C)+sin($A)*cos($B)]<0} {
        set q(1) [expr $q(1)*-1]
    }

set q(2) [expr abs(sqrt((-sin($A)*sin($B)*sin($C)+cos($A)*cos($C)-
cos($B)*cos($C)-cos($A)*cos($B)+1))/2)]
    if {[expr sin($B)+sin($A)*sin($B)*sin($C)-sin($A)*sin($C)]<0} {
        set q(2) [expr $q(2)*-1]
    }

set q(3) [expr abs(sqrt((cos($A)*cos($B)-
cos($B)*cos($C)+sin($A)*sin($B)*sin($C)-cos($A)*cos($C)+1))/2)]
```



```

    if {[expr
sin($A)*sin($B)*cos($C)+cos($A)*sin($C)+cos($B)*sin($C)]<0} {
        set q(3) [expr $q(3)*-1]
    }

global COORDINATE target_number
if {![info exists target_number]} {
    set target_number 10
}

set COORDINATE(0) $target_number
set COORDINATE(1) $X
set COORDINATE(2) $Y
set COORDINATE(3) $Z
set COORDINATE(4) $q(0)
set COORDINATE(5) $q(1)
set COORDINATE(6) $q(2)
set COORDINATE(7) $q(3)
set target_number [expr $target_number+10]

# Compiling full target block
global TARGET
global target_num
global axis_limits
global PLC_addresses
if {![info exists target_num]} {
    set target_num 0
}

#Redefining A-, B-, C-angles from RAD->DEG
set A $mom_pos(3)
set B [expr $mom_pos(4)+180]
set C $C*180/$Pi

#Calculating axis configurations depending on work object data and
tool data.
PB_CMD_AXIS_LIMIT_CALCULATION
set cFormat "\%.3f"
set qFormat "\%.4f"
set TARGET($target_num)

"CONST%robtarget%Target_$COORDINATE(0):=\[\[\[\
[format $cFormat $mom_pos(0)],\
[format $cFormat $mom_pos(1)],\
[format $cFormat $mom_pos(2)]\
\],\[\
[format $qFormat $q(0)],\
[format $qFormat $q(1)],\
[format $qFormat $q(2)],\
[format $qFormat $q(3)]\
\],\[$axis_limits\],\[$PLC_addresses\]\];"

regsub -all " " $TARGET($target_num) " " TARGET($target_num)
regsub -all "%" $TARGET($target_num) " " TARGET($target_num)
set target_num [expr $target_num+1]
}

```

Lisa 6. Trendline

```
#=====
proc PB_CMD_TRENDLINE { } {
#=====
global prev_X prev_Y prev_C C mom_pos upper_limit lower_limit
mom_trendline_correction
global Pi mom_trendline_side

set upper_limit [expr 400*$Pi/180]
set lower_limit [expr -400*$Pi/180]

if {[info exists prev_X] && [info exists prev_Y]} {
    # Here goes the calculation of the motion trend

    global motion_vector
    set motion_X [expr $mom_pos(0)-$prev_X]
    set motion_Y [expr $mom_pos(1)-$prev_Y]

    # Calculating angle C from cos alpha between X-axis (1,0)
    # and motion_vector()

    set C [expr
acos($motion_X/sqrt(pow($motion_X,2)+pow($motion_Y,2)))+$mom_trendli
ne_correction*$Pi/180]

    #Now C angle must be adjust according to it's position
[up/bottom]

    if {$motion_Y<0} {
        set C [expr 2*$Pi-$C]
    }

    #Correcting side +-90 deg from motion direction
    if {[string match $mom_trendline_side "LEFT"]} {
        set C [expr $C-$Pi/2]
    }

    if {[string match $mom_trendline_side "RIGHT"]} {
        set C [expr $C+$Pi/2]
    }

    # Check the angle if it is in limits
    if {$C>$upper_limit || $C<$lower_limit} {
        MOM_abort "Angle C is out of limits!"
    }
}

# Every time we need to rewrite previous X and Y values

set prev_X $mom_pos(0)
set prev_Y $mom_pos(1)
}
```

Lisa 7. Lineaarne liikumine

```
#=====
proc PB_CMD_MOVE_LINEAR { } {
#=====
# Linear motion #
global target_number COORDINATE TOOL_LIST WORK_OBJECT_LIST

PB_CMD_COORDINATE_CALCULATION

global MOTION
global motion_num

if {[info exists motion_num]} {
    set motion_num 0
}
global COORDINATE
global mom_feed_rate
global mom_tool_name
global mom_main_mcs

global mom_arc_code mom_arc_engaged

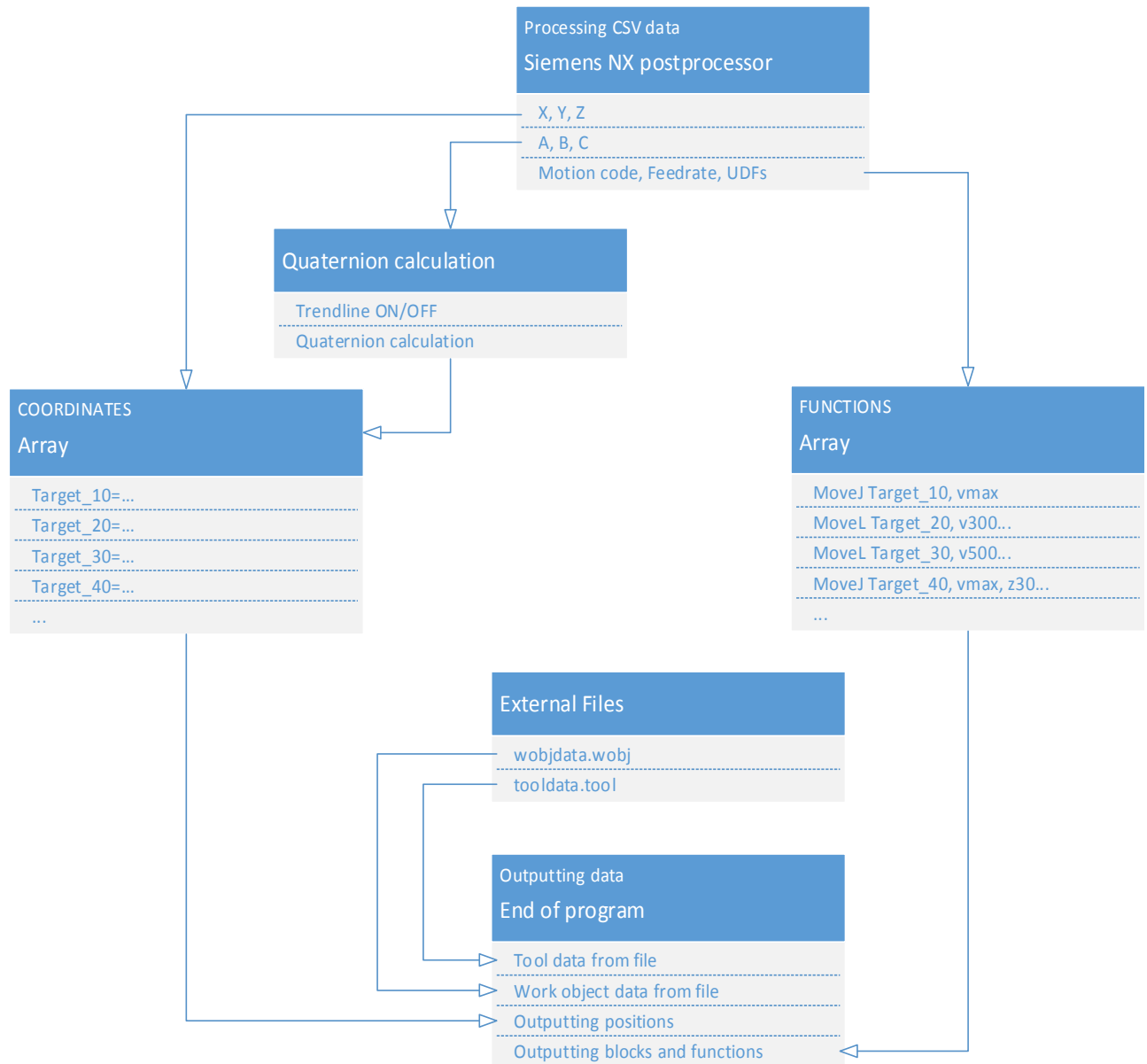
global mom_pause_before_arc_on mom_arc_current mom_arc_voltage
mom_wire_speed
global mom_pause_before_arc_off mom_arc_weaving arc_param_output

if {[info exists mom_arc_code] && $mom_arc_code==1} {
    set linear_motion "Move"
    if {[info exists mom_arc_engaged] && $mom_arc_engaged==1} {
        set linear_motion "Arc"
        if {[info exists arc_param_output] && $arc_param_output==1}
        {
            set MOTION($motion_num) "WaitTime [format "%0.2f"
$mom_pause_before_arc_on]\n\t\t"
            set TOOL_LIST($motion_num) $mom_tool_name
            set WORK_OBJECT_LIST($motion_num) $mom_main_mcs
            set motion_num [expr $motion_num+1]
            set arc_param_output 0
        }
    }
} else {
    set linear_motion "Move"
}

set MOTION($motion_num) "$linear_motion\L
Target_$COORDINATE(0),v[format %.0f
$mom_feed_rate],z1,$mom_tool_name\\WObj:=$mom_main_mcs"
set TOOL_LIST($motion_num) $mom_tool_name
set WORK_OBJECT_LIST($motion_num) $mom_main_mcs
set motion_num [expr $motion_num+1]
set mom_arc_engaged 1

}
```

Lisa 8. Juhtimisprogrammi väljastamisalgoritm



Lisa 9. DH-matriksi elemendid

$$r_{11} = s_6 \cdot ((s_5 \cdot ((s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_3 \cdot c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) - c_5 \cdot s_3 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) - c_6 \cdot ((c_5 \cdot ((s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_3 \cdot c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) + s_3 \cdot s_5 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2))))$$

$$r_{12} = c_6 \cdot ((s_5 \cdot ((s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_3 \cdot c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) - c_5 \cdot s_3 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) + s_6 \cdot ((c_5 \cdot ((s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_3 \cdot c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) + s_3 \cdot s_5 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2))))$$

$$r_{13} = - (c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1))) - c_3 \cdot s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2))$$

$$r_{21} = - ((c_6 \cdot ((c_3 \cdot s_5 + c_4 \cdot c_5 \cdot s_3))) - s_6 \cdot ((c_3 \cdot c_5 - c_4 \cdot s_3 \cdot s_5))$$

$$r_{22} = s_6 \cdot ((c_3 \cdot s_5 + c_4 \cdot c_5 \cdot s_3)) - c_6 \cdot ((c_3 \cdot c_5 - c_4 \cdot s_3 \cdot s_5))$$

$$r_{23} = s_3 \cdot s_4$$

$$r_{31} = s_6 \cdot ((s_5 \cdot ((s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + c_3 \cdot c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) + c_5 \cdot s_3 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) - c_6 \cdot ((c_5 \cdot ((s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + c_3 \cdot c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) - s_3 \cdot s_5 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1))))$$

$$r_{32} = c_6 \cdot ((s_5 \cdot ((s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + c_3 \cdot c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) + c_5 \cdot s_3 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) + s_6 \cdot ((c_5 \cdot ((s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + c_3 \cdot c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) - s_3 \cdot s_5 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1))))$$

$$r_{33} = c_3 \cdot s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2))$$

$$O_x = 570 \cdot c_1 + 640 \cdot s_3 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) - 95 \cdot c_5 \cdot ((s_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - c_3 \cdot c_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)))) + 130 \cdot c_1 \cdot c_2 - 130 \cdot s_1 \cdot s_2 - 95 \cdot s_3 \cdot s_5 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + 150$$

$$O_y = 640 \cdot c_3 - 95 \cdot c_3 \cdot s_5 - 95 \cdot c_4 \cdot c_5 \cdot s_3$$

$$O_z = 95 \cdot s_3 \cdot s_5 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - 640 \cdot s_3 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)) - 95 \cdot c_5 \cdot ((s_4 \cdot ((c_1 \cdot c_2 - s_1 \cdot s_2)) + c_3 \cdot c_4 \cdot ((c_1 \cdot s_2 + c_2 \cdot s_1)))) - 130 \cdot c_1 \cdot s_2 - 130 \cdot c_2 \cdot s_1 - 570 \cdot s_1$$

Lisa 10. T-tähe markeerimine, Motoman NX100 kontrolleri juhtimisprogramm

```
/JOB
//NAME TTU_LOGO
//POS
///NPOS 0,0,0,0,0,0
///TOOL 0
///POSTYPE PULSE
///PULSE
//INST
///DATE 2015/04/08 11:26
///ATTR SC, RW
SETE P000 (1) 182082
SETE P000 (2) 156616
SETE P000 (3) 5870
SETE P000 (5) 0
MOVJ P000 V=90 PL=4
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (1) 175238
SETE P000 (2) 156616
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (1) 175238
SETE P000 (2) 152614
MOVL P000 VL=1000.00
SETE P000 (1) 173926
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (1) 173926
SETE P000 (2) 161879
MOVL P000 VL=1000.00
SETE P000 (1) 175238
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (1) 175238
SETE P000 (2) 157868
MOVL P000 VL=1000.00
SETE P000 (1) 188926
SETE P000 (2) 157868
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (1) 188926
SETE P000 (2) 156616
MOVL P000 VL=1000.00
SETE P000 (1) 182082
SETE P000 (3) 870
MOVL P000 VL=1000.00
SETE P000 (3) 5870
MOVL P000 VL=1000.00
RET
END
```

Lisa 11. T-tähe markeerimine, ABB IRB kontrolleri juhtimisprogramm

```

MODULE BASE (SYSMODULE, NOSTEPIN, VIEWONLY)

    ! TOOL DATA
    ! tool0 is needed to go to HOME position
    !PERS tooldata
    tool0:=[TRUE,[[0,0,0],[1,0,0,0]],[0,[0,0,0],[1,0,0,0],0,0,0]];
    PERS tooldata PUNANE:=[TRUE,[[ -2.37687,-
19.1947,218.169],[1,0,0,0]],[2.5,[0,0,40],[1,0,0,0],0,0,0]];

    ! WORK OBJECT DATA
    PERS wobjdata PABER1:=[FALSE,TRUE,"",[800.257,-
488.41,501.038],[1,0,0,0]],[[0,0,0],[1,0,0,0]];

    CONST jointtarget HOME:=[[0,0,0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_10:=[[182.082,156.616,5.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_20:=[[182.082,156.616,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_30:=[[175.238,156.616,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_40:=[[175.238,152.614,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_50:=[[173.926,152.614,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_60:=[[173.926,161.879,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_70:=[[175.238,161.879,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_80:=[[175.238,157.868,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_90:=[[188.926,157.868,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_100:=[[188.926,156.616,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_110:=[[182.082,156.616,0.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
    CONST robtarget Target_120:=[[182.082,156.616,5.870],[0.0000,0.0000,-
1.0000,0.0000],[-2,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

    PROC main()
        !Going to home position
        MoveABSJ HOME,vmax,fine,tool0;

        ! PATH NAME: PLANAR_PROFILE_COPY;
        MoveJ Target_10,vmax,z10,PUNANE\Wobj:=PABER1;
        MoveL Target_20,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_30,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_40,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_50,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_60,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_70,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_80,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_90,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_100,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_110,v1000,z1,PUNANE\Wobj:=PABER1;
        MoveL Target_120,v1000,z1,PUNANE\Wobj:=PABER1;

        !Going to home position
    
```

```
MoveABSJ HOME,vmax,fine,tool0;
```

```
ENDPROC
```

```
ENDMODULE
```