

TEEMU PYÖRNY

CAM-PROSESSI TARKKUUSVALMIS- TUKSESSA

Tekniikan ja luonnontieteiden tiedekunta
Diplomityö
Lokakuu 2019

TIIVISTELMÄ

Teemu Pyörny: CAM-prosessi tarkkuusvalmistuksessa
Diplomityö
Tampereen Yliopisto
Konetekniikan tutkinto-ohjelma
Lokakuu 2019

Tarkkaa koneistusta vaativan kappaleen tietokoneavusteinen valmistus (englanniksi Computer-Aided Manufacturing, CAM) ja siihen liittyvä prosessi on yksi yrityksen toiminnan tärkeimmistä osa-alueista, sillä optimoitua prosessia voidaan käyttää kilpailuetuna ja varmistaa valmistuksen onnistunut toteutus. Diplomityössä esitetään erilaisten CAM-prosessien rakenteita ja käsitteitä. Työn tavoitteena on selvittää optimoidun CAM-prosessin rakenne esimerkitapaukseen sekä tarjota erilaisia työkaluja Sandvikin ja muiden yritysten CAM-prosessien tehostamiseksi. Työssä kerätään lukuisten haastattelujen avulla yleisiä ongelmia CAM-prosesseihin liittyen, joita käytetään lähtökohdina ratkaisumallien luomisessa potentiaalisten työkalujen avulla.

Työ koostuu kahdesta osasta, jotka ovat kirjallisuustutkimus ja soveltaminen. Kirjallisuustutkimuksessa käydään läpi CAM-prosessien koostumusta sekä tarkkuusvalmistukseen liittyviä osa-alueita. Soveltamisessa käytetään hyväksi kirjallisuuskatsausta sekä kerätään tietoa CAM-prosesseihin ja niiden ongelmakohtiin liittyen useiden haastatteluiden avulla neljässä eri yrityksessä. Sandvikin nykytilasta tehdään yksityiskohtainen prosessikuvaus ja kerätään ylös siihen liittyvät ongelmakohtat. Lisäksi potentiaaliset työkalut esitellään ja niiden käyttöä kokeillaan tunnistettujen ongelmakohtien ratkaisemisessa.

Suurimpana ongelmakohtina ovat prosessisuunnittelun toteutus, suunnittelijoiden tekemiin tuotoksiin kuuluva aika sekä manuaalisten operaatioiden aiheuttamat pullonkaulat. Yritykset miettivät tapoja, joilla olemassa olevista työkaluista saadaan paras mahdollinen hyöty. Työkaluja ovat muun muassa tiedonhallintaohjelmisto Teamcenter sekä suunnitteluohjelmistot. Tärkeimpinä ratkaisukeinoina ovat prosessisuunnittelun työkalu Manufacturing Process Planner, joka on osa Teamcenteriä. Myös NX Open ohjelmointirajapintojen hyödyntäminen osana suunnitteluohjelmistoa NX osoittautuu potentiaalisesti ratkaisutavaksi automatisoida kappaleen geometrian, valmistuksen sekä koordinaattimittauskoneen ohjelman tekemisen operaatioita. NX Openia pidetään potentiaalisena kaikkien haastateltujen yritysten tahoilta. Diplomityön konkreettiset liiketoiminnalliset hyödyt eivät ole tiedossa sillä työssä luotua sisältöä ei ole ehditty vielä ottamaan käyttöön yrityksissä. Työn avulla eri yritykset saavat tietoa uusista työkaluista, joiden avulla olemassa olevia CAM-prosesseja voidaan tehostaa.

Avainsanat: Tietokoneavusteinen valmistus, CAM, tarkkuusvalmistus, valmistusprosessi, ohjelmointirajapinnat, API, NX Open, CMM

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

ABSTRACT

Teemu Pyörny: CAM-process in precision manufacturing
Master's Theses
Tampere University
Mechanical Engineering
October 2019

Computer-Aided Manufacturing (CAM) process, in case of a product that needs precise machining, is one of the most important areas in company's functionality because an optimized process can be used as a competitive edge and makes sure that manufacturing is being implemented successfully. This thesis represents structures and concepts of various CAM-processes. The main goal of the thesis is to discover the structure of an optimized CAM-process in case of an example case and to provide different tools to Sandvik and other companies for enhancing their CAM-processes. This thesis gathers general issues related to CAM-processes based on numerous interviews. Those issues are being used as a starting point for creating solution models with potential tools.

The thesis consists of two parts that are literature review and application. Literature review looks through which parts CAM-processes consist of and what kind of areas there are in precision manufacturing. Application part takes benefit of the literature review and gathers information related to CAM-processes and their issues through interviews at four different companies. Current situation of the CAM-process at Sandvik is being visualized and different issues are being marked down. Moreover, potential tools are being introduced and tried out in order to solve those recognized issues.

The biggest recognized issues are computer-aided process planning implementation, time caused by designer operations and bottlenecks because of manual operations. Companies are thinking about ways for taking the most of existing applications. For example, product lifecycle management application Teamcenter and designing software are common applications. The most desirable solutions are following: a process planning tool called Manufacturing Process Planner, which is part of Teamcenter. Also, taking advantage of NX Open application programming interfaces as a part of designing software NX turns out to be a potential way to automatize operations for part geometry modeling, manufacturing method planning and coordinate-measuring machine program planning. NX Open proves to be a potential solution method according to all interviewed companies. Specific financial benefits are not being recognized yet because companies haven't had time for taking advantage of the thesis yet. The most important general level purpose of this thesis is to provide information to different companies about new solution tools that they can consider starting to use in order to enhance existing CAM-processes.

Keywords: Computer-aided manufacturing, CAM, precision manufacturing, manufacturing process, application programming interfaces, API, NX Open, CMM

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

ALKUSANAT

Tämä diplomityö on tehty yhteistyössä Tampereen Yliopiston ja Sandvikin kanssa kehittämään olemassa olevaa tilannetta CAM-prosesseihin liittyen. Tarkoituksena on selvittää CAM-prosessien rakenteet, ongelmakohdat sekä kerätä erilaisia ratkaisutapoja ongelmakohtien ratkaisemiseksi ja näin ollen CAM-prosessien tehostamiseksi.

Kiitokset yliopistolle ja Sandvikille kaikkien tarpeellisten resurssien tarjoamisesta. Kiitos myös Sandvikin sekä muiden haastateltujen yritysten henkilöstölle työn kannalta erittäin hyödyllisen tiedon kertomisesta. Erityiskiitokset diplomityön ohjaajana ja toisena tarkastajana toimineelle Andrei Loboville sekä Pasi Julkuselle, joka tarjosi diplomityön aiheen Sandvikilla.

Tampereella, 28.10.2019

Teemu Pyörny

SISÄLLYSLUETTELO

1. JOHDANTO	1
1.1 Kohdeyityksen nykytila.....	1
1.2 Työn tausta	2
1.3 Työn tavoitteet ja tutkimuskysymykset	2
1.4 Työn rakenne	3
2. CAM-PROSESSI	5
2.1 CAM-prosesseissa käytettäviä käsitteitä	5
2.1.1 CAD	5
2.1.2 CAM.....	7
2.1.3 CAPP	8
2.1.4 NC-koodi ja postprosessori	9
2.1.5 CNC-kone.....	10
2.1.6 CMM	11
2.1.7 DNC.....	13
2.1.8 MRL	14
2.2 CAM-prosesseja kirjallisuudessa.....	16
2.3 Ohjelmointirajapinnat ja työkalut.....	19
2.4 Yhteenveto.....	22
3. TARKKUUSVALMISTUS	24
3.1 Toleranssit	27
3.1.1 Muototoleranssit	31
3.1.2 Suuntatoleranssit	33
3.1.3 Sijaintitoleranssit.....	34
3.1.4 Heittotoleranssit	36
3.1.5 Profiilitoleranssit.....	37
3.1.6 Toleranssit eri standardeissa	38
3.1.7 Toleranssien merkitseminen Siemens NX:ssä	41
3.2 Pinnankarheus	45
3.2.1 Pintamerkit.....	48
3.2.2 Pinnankarheidet eri standardeissa.....	52
3.2.3 Pinnankarheuksien merkitseminen Siemens NX:ssä	54
3.3 5-akselinen koneistuskeskus.....	58
3.4 Yhteenveto.....	62
4. ESIMERKKITAPAUSET	64
5. CAM-PROSESSIN METODOLOGIA.....	66
6. TOTEUTUS	77
6.1 CAM-prosesseja ja kehityskohteita muissa yrityksissä	78
6.2 Ohjelmointirajapintojen hyödyntäminen.....	87
6.3 CMM ohjelmien automaattinen generointi	94
7. TULOKSET JA ANALYYSI	99
8. KESKUSTELU	106

9. YHTEENVETO JA KEHITYSKOhteet	108
LÄhteet	111
LIITE A: LUVUN 6.2 CAD PUOLEN Koodi	115
LIITE B: LUVUN 6.2 CAM PUOLEN Koodi	118
LIITE C: LUVUN 6.2 CMM PUOLEN Koodi	124
LIITE D: SANDVIKILLE TEHTY SUUNNITELMA	127

KUVALUETTELO

Kuva 1.	<i>DNC:n tarkoitus, mukaillen lähdettä (Jiang 2011).....</i>	13
Kuva 2.	<i>MRL Siemensin ohjelmassa Teamcenter.....</i>	15
Kuva 3.	<i>Lähteen (Ismail, Lin et al. 2003, s.184) mukainen CAM-prosessi.....</i>	16
Kuva 4.	<i>CAM-prosessi perustuen lähteeseen (Siemens 2019).....</i>	18
Kuva 5.	<i>Siemensin oppaat NX Openiin, perustuen lähteeseen (Siemens 2019)</i>	20
Kuva 6.	<i>NXOpenin esimerkkikomento.....</i>	21
Kuva 7.	<i>Tarkkuusvalmistuksen koneistustarkkuudet ja luokittelu, mukaillen lähteitä (Dornfeld, Lee 2008, s. 6, Zhang, To et al. 2015, s. 77)</i>	25
Kuva 8.	<i>Toleranssien merkitseminen piirustukseen, mukaillen lähdettä (Casas Fernández, Sordo et al. 2006, s. 7).....</i>	29
Kuva 9.	<i>Yleisimmät lisätunnukset, mukaillen lähdettä (Chang, Wysk et al. 2006, s. 57).....</i>	30
Kuva 10.	<i>Muototoleranssit, mukaillen lähdettä (Pere 2009, s. 20-82).....</i>	32
Kuva 11.	<i>Suuntatoleranssit, mukaillen lähdettä (Narayana 2006, s. 234).....</i>	33
Kuva 12.	<i>Sijaintitoleranssit, mukaillen lähdettä (Pere 2009, s. 20-82)</i>	35
Kuva 13.	<i>Heittotoleranssit, mukaillen lähdettä (Pere 2009, s. 20-82)</i>	36
Kuva 14.	<i>Profiilitoleranssit.....</i>	38
Kuva 15.	<i>ASME:n ja ISO:n merkintätapojen eroavaisuuksia, mukaillen lähteitä (Rufe 2013, Simmons, Phelps et al. 2012).....</i>	40
Kuva 16.	<i>Toleranssipiirteen lisääminen NX-piirustukseen.....</i>	42
Kuva 17.	<i>Toleranssikehyksen liittäminen NX-piirustukseen.....</i>	43
Kuva 18.	<i>Toleranssipiirteen lisääminen NX-malliin.....</i>	44
Kuva 19.	<i>Eri valmistusmenetelmillä saatavia pinnankarheuksia, perustuu lähteeseen (Pere 2009, s. 21-16).....</i>	46
Kuva 20.	<i>Pinnankarheuden määrittämisen suuret, perustuen lähteeseen (Narayana 2006, s. 242)</i>	47
Kuva 21.	<i>Pintamerkit, mukaillen lähteitä (Pere 2009; Pohanish 2016).....</i>	49
Kuva 22.	<i>Pintakuvion ilmoittaminen, perustuen lähteeseen (Pere 2009, s. 21-30)</i>	50
Kuva 23.	<i>Pintamerkkien merkitsemistapoja, perustuen lähteeseen (Pere 2009)</i>	52
Kuva 24.	<i>Pintamerkin eroavaisuus SFS:n ja ISO:n kesken, perustuen lähteeseen (Pere 2009, s. 21-64).....</i>	53
Kuva 25.	<i>ASME:n, DIN:n ja ISO:n täydellisten pintamerkkien esittäminen NX:ssä.....</i>	53
Kuva 26.	<i>Pinnankarheuden lisääminen NX-piirustukseen</i>	55
Kuva 27.	<i>Pinnankarheuden lisääminen NX-malliin</i>	56
Kuva 28.	<i>Pinnankarheuden määrittäminen NX-piirustuksessa</i>	57
Kuva 29.	<i>5-akselisten koneistuskeskusten yleisimmät konfiguraatiot, perustuu lähteeseen (Shinde, Bhole 2015)</i>	59
Kuva 30.	<i>Hydrostaattinen liikuttaminen, perustuu lähteeseen (Shinde, Bhole 2015)</i>	61
Kuva 31.	<i>MRL:n osa-alueet.....</i>	66
Kuva 32.	<i>NX:n sisällä suoritettavat vaiheet</i>	68
Kuva 33.	<i>CMM ohjelman vaiheet</i>	70
Kuva 34.	<i>CAM-prosessi kokonaisuudessaan</i>	72
Kuva 35.	<i>Tasowheelin prosessi.....</i>	79
Kuva 36.	<i>Salon Konepajan prosessi</i>	82
Kuva 37.	<i>Nometin prosessi</i>	85
Kuva 38.	<i>Pallon muodostamiseen tarvittava koodi</i>	88
Kuva 39.	<i>Journalin avulla tehtyjä muotoja</i>	89

Kuva 40.	<i>CAM-puolen kappale.....</i>	<i>91</i>
Kuva 41.	<i>Kappale CMM ohjelmaa varten.....</i>	<i>92</i>
Kuva 42.	<i>Tilanne kappaleen ja mittauskoneen resurssien määrittelyn jälkeen</i>	<i>97</i>
Kuva 43.	<i>Esimerkkitapauksen ideaalinen CAM-prosessi.....</i>	<i>99</i>

LYHENTEET JA MERKINNÄT

API	Ohjelmointirajapinta, Application Programming Interface
ASME	Amerikkalainen standardisointiliitto, American Society of Mechanical Engineers
CAD	Tietokoneavusteinen suunnittelu, Computer-Aided Design
Calypso	Mittausohjelman tekemisessä käytettävä ohjelma
CAM	Tietokoneavusteinen valmistus, Computer-Aided Manufacturing
CAPP	Tietokoneavusteinen prosessisuunnittelu, Computer-Aided Process Planning
CMM	Koordinaattimittauskone, Coordinate-Measuring Machine
CNC	Tietokoneistettu numeerinen ohjaus, Computer Numerical Control
CNC-kone	Tietokoneistettu numeerisesti ohjattu kone
CR	Kontrolloitu säde, Controlled Radius
DIN	Saksalainen standardisointi-instituutti
DMIS	Mittausstandardi, Dimensional Measuring Interface Standard
DNC	Suora numeerinen ohjaus, Direct Numerical Control
EDM	Kipinätyöstö, Electrochemical Discharge Machining
FEA	Äärellisten elementtien analyysi, Finite Element Analysis
GD&T	Geometrinen mitoitus ja tolerointi, Geometric Dimensioning and Tolerancing
G-koodi	Työstökoneen toimintoja esittävä koodi
ISO	Kansainvälinen standardisointiliitto, International Organization for Standardization
Journal	Työkalu käyttäjän tekemien operaatioiden tallentamiseen NX:ssä
LMC	Vähimmäismateriaalin vaatimus, Minimum Material Condition
MMC	Maksimimateriaalin vaatimus, Maximum Material Condition
MMS	Tuotannonohjausjärjestelmä, Manufacturing Management Software
MRL	Valmistusresurssien kirjasto, Manufacturing Resource Library
M-koodi	Työstökoneen toimintoja esittävä koodi
NC-koodi	Numeerisesti kontrolloitu koodi, Numerical Controlled Code
NURBS	Tarkkojen liikeratojen mahdollistava periaate, Non-Uniform Rational B-Spline
NX	Siemensin suunnitteluohjelmisto tuotekehityksen suunnitteluun ja valmistuksen tarpeisiin
NX Open	Siemensin ohjelmointirajapinnat
PMI	Tuotteen ja valmistuksen tieto, Product & Manufacturing Information
P-profiili	Rakenneprofiili
Ra	Profiilin aritmeettinen keskipoikkeama, Mean Roughness Index
RFS	Riippumatta materiaalin koosta, Regardless of Feature Size
R-profiili	Karheusprofiili
SFS	Suomen Standardisointiliitto
ST	Statistinen tolerointi, Statistical Tolerancing
TED	Teoreettisesti tarkat mitat, Theoretically Exact Dimensions
VC	Virtuaalinen olosuhde, Virtual Condition
W-profiili	Aaltomaisuusprofiili

1. JOHDANTO

Johdanto esitetään alaotsikoiden avulla, joissa kerrotaan diplomityön teettävästä yrityksestä, tarpeista työn taustalla, tavoitteista, rajauksesta, tutkimuskysymyksistä ja tämän diplomityön rakenteesta. Johdannon tavoitteena on antaa lukijalle hyvä yleiskuva diplomityön sisällöstä. Johdannon luettuaan lukija pystyy päättämään, onko työn sisältö tarpeeksi kiinnostava tai soveltuva koko työn lukemiseen. Diplomityö on tehty yhteistyössä Tampereen Yliopiston ja Sandvikin kanssa. Diplomityö on kuitenkin luotu siten, että siitä pystyvät hyötymään Sandvikin lisäksi myös muut valmistusprosessien kehittämistä kiinnostuneet yritykset.

1.1 Kohdeyrityksen nykytila

Sandvik on vuonna 1862 perustettu noin 42000:n työntekijän kansainvälinen korkeatasoisen teknologian teollisuuskonserni, jonka pääasiallisina tarkoituksina on parantaa asiakkaiden tuottavuutta, vakautta ja turvallisuutta. Sandvik investoi jatkuvasti tutkimustyöhön ja tuotekehitykseen tavoitteenaan olla johtava yritys seuraavaksi esitetyillä alueilla: Sandvik tarjoaa erilaisia työkaluja ja järjestelmiä metallin leikkaukseen teollisuudessa mahdollistaen kannattavamman valmistuksen. Kaivos- ja urakointiteollisuuksille Sandvikilla on kalustoa, koneita, palveluja ja teknisiä ratkaisuja. Lisäksi Sandvikilla on paljon edistyneitä tietotaitoa ruostumattomista teräsmateriaaleista, erikoisoseosmateriaaleista sekä mahdollisuus tarjota tuotteita teolliseen lämpökäsittelyyn. (Sandvik 2019)

Tämä diplomityö tehdään Sandvikin kaivos- ja urakointiteollisuuteen keskittyvälle alueelle nimeltään Sandvik Mining and Rock Technology. Tämä noin 15500 henkilöä työllistävä osa Sandvikia tarjoaa muun muassa seuraavia tuotteita ja palveluja: Paikallaan pysyviä sekä liikuteltavia murskaimia ja seuloja, malminetsintälaitteita ja tarvikkeita, mekaanisen rouhinnan laitteita, porakalustoa, porakoneita, iskuvasaroita, purkutyökaluja, rikotuspuomeja ja automaatiota. Lisäksi tuotteisiin kuuluu maanalaisia lastauskoneita, dumppereita, poraus- ja pultituslaitteita sekä maanpäällisiä porauslaitteita. Tämä nopeasti kasvava yrityksen liiketoiminta-alue tarjoaa siis ratkaisuja sekä yksittäisiin tuotteisiin

että koko toiminnan hallintaan tarkoituksenaan olla omalla alueellaan johtava toimittaja. (Sandvik 2019)

1.2 Työn tausta

Sandvikin eräänä tärkeänä periaatteena on oman toiminnan jatkuva kehittäminen. Tämän diplomityön tärkeimpänä motiivina on tarve parantaa tuotteiden suunnittelun ja valmistuksen välisen prosessin kulkua, mahdollistaen sujuvamman toiminnan tuotteen elinkaaren alussa. Tietokoneavusteinen valmistus (CAM) on ollut yrityksessä puheenaiheena jo pitkään, mutta mitään merkittäviä toimenpiteitä tai tuloksia ei ole dokumentoitu prosessin kehittämiseen liittyen. Kaikilla CAM-prosessin aikana osallisina olevilla työntekijöillä ei ole selkeää kuvaa koko prosessista ja tästä syystä on hyödyllistä teettää helposti lähestyttävä dokumentti, jonka luettuaan CAM-prosessin vaiheet ja resurssit tulisivat tutummiksi lukijalle.

Tämä diplomityö liittyy pääosin uusien porakoneiden komponentteihin liittyvään suunnitteluun ja valmistukseen. Sandvikin porakoneilta vaaditaan tarkat toleranssit täyttäviä komponentteja, jonka johdosta valmistusvaiheessa on syytä saada jo ensimmäisellä yrityksellä toleranssien sisällä pysyviä tuotteita. Yritys on investoinut uusiin koneistuskeskuksiin, joiden avulla voidaan valmistaa nyt myös 5-akselista työstöä vaativia komponentteja. Saman tehtaan alueella tapahtuva valmistus yksinkertaistaa eri työntekijöiden välistä kommunikointia ja tilanteisiin reagoimista. Diplomityössä paneudutaan tarkkuuteen liittyen toleranssien käyttöön suunnittelussa ja koneistuksessa sekä laadunvalvonnassa käytettävän mittauskoneen luomisprosessiin.

1.3 Työn tavoitteet ja tutkimuskysymykset

Kirjallisuudesta löytyy monia erilaisia työkaluja ja menetelmiä CAM:iin liittyen mutta työssä käytettävän esimerkkitapauksen ja yrityksen käytössä olevien resurssien johdosta on oleellista valita yhdeksi fokusalueeksi esimerkkitapausta tukevat ja parantavat apuvälineet. Tämän johdosta esimerkkitapaukseen esitetään ideaalinen kuvaus käytettävästä CAM-prosessista. Sandvikilla on käynnissä CAM pilottiprojekti mutta siinä on kerrottu vain yleisellä tasolla tilanteesta, eikä ole saatu vielä mitään tuloksia. Tässä työssä ei ole tarkoituksena tutkia CAM-prosessia pelkästään esimerkkitapauksessa, vaan aihepiiriin liittyvää tietoa kerätään vieraillemalla kolmessa muussa valmistuksen parissa toimivassa yrityksessä. Muiden yritysten kohdalla suoritetaan vapaamuotoisia haastatteluja aiheesta. Haastattelujen pohjalta kerrotaan useiden yritysten CAM-pro-

sessien toteuttamisesta sekä havainnoiduista ongelmista ja kehityskohteista. Ongelma-kohtien tunnistamisen jälkeen työssä paneudutaan potentiaalisiin ratkaisukeinoihin. Ratkaisukeinojen toiminnot selitetään ja joissain tapauksissa havainnollistetaan.

Seuraavana tavoitteena on tutkia koordinaattimittausohjelman luomista. Diplomityössä tunnistetaan minimivaatimukset automaattisesti generoitavaan ohjelmaan sekä testataan siihen liittyvän apuvälineen käyttämistä. Todistetusti toimivaa ratkaisumallia ei synny mutta työn luettuaan lukija tietää, millaisia asioita tarvitsee olla määriteltynä automaattisesti generoitavassa ohjelmassa. Diplomityön viimeisenä tavoitteena on tunnistaa potentiaalisesti toimiva ratkaisu sulavaan tiedonsiirtoon eri osapuolten välillä. Eri osapuolia ovat suunnittelijat, valmistuksen parissa työskentelevät henkilöt sekä toisen osapuolen yritykset, esimerkiksi alihankkijat ja asiakkaat. Sulavan tiedonsiirron ratkaisumallien pohjalla on eri osapuolten kanssa käydyt haastattelut kehityskohteista sekä kirjallisuudesta tehty tutkimustyö aihepiiriin liittyen.

Tutkimusongelman pohjalta luodut tutkimuskysymykset ovat:

1. Mikä on ideaalinen CAM-prosessi tarkkuusvalmistuksessa ja mistä osista se koostuu?
2. Mitä CAM:iin liittyviä ongelmakohtia on ja miten ne ratkaistaan?
3. Mitä tarvitaan, että tuotteen 3D-mallista voidaan automaattisesti generoida koordinaattimittausohjelma?
4. Kuinka voidaan toteuttaa sulava tiedonsiirto eri osapuolten välillä?

1.4 Työn rakenne

Työ alkaa kirjallisuusselvityksellä luvuissa kaksi ja kolme, missä kerätään kirjallisuudesta löytyvää tietoa työn kannalta oleellisista asioista. Kirjallisuusselvityksessä tutkitaan aluksi CAM-prosessiin liittyviä käsitteitä, jonka jälkeen esitetään kirjallisuudessa olevia malleja prosessien rakenteille. Tämän jälkeen tutustutaan ohjelmointirajapintojen käsitteeseen, joka on tässä työssä pääosin NX Open. Ohjelmointirajapintoja seuraa toleranssien ja pinnankarheuden teoriaosuudet, joiden seurauksena esitetään niiden merkkäminen NX:n avulla sekä merkkäustapojen eroavaisuudet eri standardien kesken. Viimeisenä kirjallisuusarvion osana käydään läpi 5-akseliseen koneistuskeskukseen liittyvää

teoriaa, vertailua 3-akseliseen koneistuskeskukseen sekä potentiaalisia parannusehdotuksia yleisiin koneistuskeskuksissa ilmeneviin ongelma-kohtiin.

Luvussa neljä esitellään työhön liittyvä esimerkkitapaus mahdollisimman tarkasti ja kerrotaan, millaisia vaatimuksia siihen liittyvälle prosessille on asetettu. Viidennessä luvussa käydään esiteltyyn esimerkkitapaukseen liittyvä prosessi yksityiskohtaisesti läpi tunnistuen prosessin kulussa vaadittavia resursseja. Tämän lisäksi havainnollistetaan erilaisia tiedonsiirrossa käytettäviä tietotyyppisiä, jotta voidaan nähdä kuinka monessa muodossa tieto liikkuu prosessin aikana. Esimerkkitapauksen CAM-prosessiin liittyvät ongelmat kohdat tunnistetaan ja ratkaisuehdotukset esitellään luvun lopussa.

Kuudennessa luvussa laajennetaan käsitystä CAM-prosesseista haastatteleamalla eri yrityksiä heidän prosesseistaan sekä niihin liittyvistä ongelma-kohtista. Haastattelujen pohjalta voidaan tunnistaa kaikista yleisimmät ongelmat tämänhetkisessä tilanteessa ja niiden ratkaisemiseen voidaan paneutua sen jälkeen. Luvun seuraavassa osiossa pyritään ratkaisemaan olemassa olevia ongelma-kohtia muun muassa NX Openia hyödyntäen. Kirjoittajan esittämiä parannusehdotuksien perustellaan kirjallisuudesta löytyvien esimerkkitapausten kautta, jonka johdosta työssä pyritään välttämään todennäköisesti turhaksi osoittautuvia ratkaisumalleja. Luvun viimeinen osa liittyy mittauskoneohjelmien luomisen tutkimiseen ja siihen tunnistettujen kehityskohteiden ratkaisemiseen.

Luvussa seitsemän esitetään vastaukset tutkimuskysymyksiin ja analysoidaan vastauksien onnistuneisuutta sekä potentiaalisuutta. Luvussa kahdeksan ja yhdeksän tarkistetaan työn yhdenmukaisuus vertaamalla sitä kirjallisuuteen ja kerrotaan, miten työ on laajentanut alkuperäisessä kirjallisuuskatsauksessa saatua tietoa. Lisäksi kommentoidaan työn onnistuneisuutta. Lopuksi kootaan yhteen, millaista kehitystyötä tulevaisuudessa olisi tarpeellista tehdä aihepiiriin liittyen ja esitetään keinoja toteuttaa tunnistettuja kehityskohteita sekä Sandvikilla että kolmessa muussa yrityksessä.

2. CAM-PROSESSI

Tässä luvussa käydään alalukujen avulla läpi, minkälaisia käsitteitä ja vaiheita kuuluu valmistettavien tuotteiden CAM-prosesseihin. Ensimmäisenä määritellään prosessissa tarvittavat peruskäsitteet, jotka ovat oleellisessa osassa tämän luvun sisällön ymmärtämisestä. Tämän jälkeen käydään läpi olemassa olevia malleja CAM-prosesseille. Lopuksi esitetään erilaiset ohjelmointirajapinnat ja työkalut CAM-prosesseihin liittyen, jotka ovat potentiaalisia apuvälineitä prosessien tehostamiseen. Tämän luvun tutkimisen jälkeen ollaan selvillä nykytilasta ja sitä voidaan käyttää hyväksi työn myöhemmässä vaiheessa, jossa toimintaa pyritään kehittämään.

2.1 CAM-prosesseissa käytettäviä käsitteitä

Tässä alaluvussa käydään läpi kirjallisuudessa käytettäviä käsitteitä ja kerrotaan niiden merkitykset. Käsitteiden pohjalta helpotetaan prosessin läpikäymisen ymmärtämistä ja varmistetaan, ettei prosessien kuvauksissa tule selittämättä jääneitä termejä. Alaluvun luettuaan lukijalla on hyvät lähtökohdat alkaa sisäistämään erilaisien CAM-prosessien sisältöjä ja myöskin mahdollisuus tarkistaa käsitteitä, joiden merkitykset eivät enää ole muistissa prosessien kuvauksia lukiessa.

2.1.1 CAD

Tietokoneavusteinen suunnittelu (englanniksi Computer-Aided Design, CAD) voidaan määrittää monella eri tavalla mutta Chang et al. (1991, s. 32) esittää helpoimmin lähestyttävän kuvauksen CAD:lle. Kuvauksen mukaan CAD voidaan yksinkertaisimmillaan määrittää tietokoneen avulla tehtäväksi suunnitteluksi, jossa tietokonetta voidaan käyttää sekä esitys- että analyysivaiheissa. Samansuuntaisen määrittelyn antaa myös Pere (2009, s. 9-1), jonka mukaan CAD tarkoittaa tietokoneen ja CAD-ohjelman käyttämistä suunnittelijan työkaluna. Määrittelyjen kautta pyritään antamaan idea siitä, että tietokoneen avulla voidaan hoitaa suurimmaksi osaksi suunnittelu, tuotoksien havainnollistaminen sekä työn luotettavuuden arvioiminen. CAD:n käyttämisen ymmärtämiseksi on tärkeää tietää, millaisista osista se koostuu ja mitä niillä tarkoitetaan.

CAD systeemi koostuu kolmesta merkittävimmästä osasta, jotka ovat laitteisto, systeemin ohjelmisto sekä suunnitteluohjelmisto. Laitteiston tehtävänä on tukea ohjelmistojen funktioita ja sen vaatimustaso riippuu käytettävän systeemin kompleksisuudesta. Sys-

teemin ohjelmisto toimii käyttöliittymänä laitteiston ja suunnitteluohjelmiston välillä valvoen suunnittelutyössä olevia prosesseja ja mahdollistaen ohjelmistoon sisältyvien osien käytön. Suunnitteluohjelmistoa voidaan pitää CAD:n sydämenä, sillä se sisältää suunnittelussa käytettävät ohjelmat ja työkalut. Tämän seurauksena syntyy myös erot eri CAD systeemien välillä, sillä kilpailevat suunnitteluohjelmistojen tekijät yritykset tarjoavat usein yhteensopivuutta vain itse kehittämilleen tuotteilleen. (Chang, Wysk et al. 1991, s. 34-35).

Business Advantagen (2008, s. 46) tutkimuksen mukaan vuonna 2018 viisi eniten käytettyä suunnitteluohjelmistoa ovat AutoCAD (35 %), SolidWorks (25 %), Inventor (13 %), PTC Creo (11 %) ja Revit (8 %). Diplomityön kohdeyrityksessä käytössä oleva NX oli tutkimuksessa kymmenentenä viiden prosentin osuudellaan. Tutkimuksessa myös todetaan, että suunnitteluohjelmien avulla tehtävien 3D-mallien käyttöaste on 92 %, joten nimenomaan 3D-mallien tekeminen on kaikista yleisin toimenpide suunnitteluohjelmistojen käytössä. Pere (2009, s. 9-1) kertoo, että suunnitteluohjelmistojen avulla saadaan CAD:iin liittyen mallinnettua ja muokattua haluttuun lopputulokseen viittaavia digitaalisia dokumentteja. Näillä dokumenteilla tarkoitetaan yleisimmin kappaleen 3D-mallia ja työpiirustusta. Näiden lisäksi CAD-ohjelmistolla voidaan suorittaa analysointia suunnitellun kappaleen hyväksymiseksi.

Eräs eniten käytetty analysointimenetelmä on nimeltään äärellisten elementtien analysointi (englanniksi Finite Element Analysis FEA). Sitä käytetään esimerkiksi seuraavissa tapauksissa: Kompleksisten struktuurien staattiset ja dynaamiset analyysit, lämmönjohtumisen ja lämpötilan aiheuttaman kuormituksen analysointi. FEA:ssa kompleksinen runko jaetaan peruselementteihin, jotka on tehty yhdestä materiaalista ja koostuvat geometrisista muodoista. Ratkaisemisessa käytetään kaikille elementeille tiettyjä yhtälöitä, jonka jälkeen saadaan varmistettua kappaleen geometrian sopivuus (Chang, Wysk et al. 1991, s. 50-51).

Kurowski (2013) esittää FEA:n käyttämistä suunnitteluvaiheessa. Suurimpana erona CAD- ja FEA-malleissa on se, että CAD-malleissa tarpeellinen tieto on valmistusvaiheessa käytettävä tieto kun taas FEA-malleissa pääpainona on verkon tekeminen ja sen laskeminen järkevässä ajassa. Verkon tekemisellä tarkoitetaan kappaleen pinnan jakamista laskennassa käytettäviksi elementeiksi. Tämän seurauksena CAD-malleista voidaan poistaa piirteitä, kuten lyhyitä reunoja ja tasoja, jotka eivät vaikuta menetelmän tuloksiin ja yksinkertaistavat verkon luomista. Tämä tarvitsee kuitenkin aina harkittua päätöksentekoa. FEA ohjelmissa on yleensä työkaluja verkon sopivuuden tarkastamiseen mutta käyttäjällä on aina vastuu hyväksyä käytettävä verkko. FEA-menetelmä voi

olla osana CAD-ohjelmaa tai sitten erillisenä ohjelmana. Joka tapauksessa menetelmää käytetään suunnittelun aikana ja se auttaa päätöksien tekemisessä.

2.1.2 CAM

CAM voidaan määritellä monella tavalla, mutta yleisimmin tunnettu määritelmä voidaan kertoa seuraavasti: Pere kirjoittaa teoksessaan (2009, s. 9-1), että CAM voidaan määritellä olevan CAD:n pohjalta luodun tiedon siirtämistä ja hyödyntämistä kappaleen työstövaiheessa. Tarkkuusvalmistuksessa käytetään tietokoneistettuja numeerisesti ohjattuja (englanniksi Computer Numerical Control, CNC) koneita, joiden ohjelmoinnissa lähtötietona käytetään CAD-ohjelmalla tehtyjä mitoitettuja malleja. CAM on siis menetelmä, jonka avulla suunnitteluvaiheessa syntynyt informaatio liikkuu etukäteen suunnitellun prosessin avulla valmistusvaiheeseen. CAM:n tavoitteiden toteutumisen arvioinnissa voidaan määritellä olevan kolme pääsääntöä: Tuotantovaiheen ja markkinointivaiheen välisen ajan pieneneminen, laadun parantuminen sekä valmistuskustannuksien vähentäminen (Novak-Marcincin, Barna et al. 2013, s. 264).

Valmistusprosessina CAM on erityisesti kohdistettu tapauksiin, joiden sisällä tehtävissä operaatioissa käytetään hyväksi tietokoneita. Valmistettaessa tarkkuutta vaativia ja monimutkaisia tuotteita, CAM-prosessissa kulkee paljon tietoa, jota ei pysty dokumentoimaan ja hyödyntämään tehokkaasti ilman tietokoneiden tarjoamaa tukea. Valmistavassa teollisuudessa yleisesti toivottuna lopputuloksena on pystyä suunnittelemaan koko tuotteen kulkema prosessi käyttämällä työkalua, jonka avulla voidaan mallintaa kaikki huomiointia tarvitsevat asiat (Novak-Marcincin, Barna et al. 2013, s. 263-264). CAM-prosessin vaiheissa on osallisena monen eri roolin työntekijöitä, käytettäviä ohjelmia sekä toimipiteitä. Nämä esitetään ja käydään yksityiskohtaisesti läpi diplomityön myöhemässä vaiheessa.

Muun muassa CAM-työkalujen käytöstä ja trendeistä kertovan Business Advantagen (2018) tutkimuksen mukaan CAM:n käyttöaste on kasvanut neljällä prosentilla vuoteen 2016 verrattuna. Vuonna 2018 CAM oli käytössä 31:llä prosentilla vastanneista käyttäjistä. Tutkimukseen vastanneista henkilöistä 38 prosenttia työskenteli valmistukseen liittyvällä sektorilla. Tulevaisuuden kasvupotentiaalia tutkittaessa 35 prosenttia käyttää tai suunnittelee käyttävänsä CAM-työkaluja tulevan vuoden aikana ja 37 prosenttia tulevan

kolmen-viiden vuoden aikana. Yhteenvetona CAM:n nykytilanteesta on se, että sillä on korkea käyttöaste mutta siitä johtuen varsin pieni kasvupotentiaali.

2.1.3 CAPP

Tietokoneavusteinen prosessisuunnittelu (englanniksi, Computer-Aided Process Planning, CAPP) on luotu helpottamaan prosessia valmistukseen tuleville kappaleille kahden eri menetelmän avulla. Menetelmät ovat nimeltään variantti lähestymistapa sekä generatiivinen lähestymistapa. Variantissa lähestymistavassa etsitään valmiita prosesseja vastaavanlaiselle, esimerkiksi samassa tuoteperheessä olevalle, kappaleelle ja sitä kautta saadaan kattava pohjatieto käsiteltävää prosessia varten. Generatiivisessa lähestymistavassa puolestaan pyritään helpottamaan täysin uuden kappaleen prosessin suunnittelua. Systeemi osaa päätellä joitain operaatioita ja valintoja perustuen kappaleen geometriaan. Generatiivinen systeemi on vaikeampi kehittää ja se vaatii kappaleelta paljon tietokoneella luettavissa olevaa tietoa. (Yusof, Latif 2014, s. 77)

CAPP systeemin toteuttamisessa on syytä ottaa huomioon seuraavat asiat: valmistuksessa käytettävät komponentit, koneistettava volyyymi ja tuoteperheiden määrä. Komponenttien avulla saadaan selville, kuinka montaa laitetta tuotteen valmistuksessa käytetään. Monipuoliset koneistuskeskukset ovat kykeneviä valmistamaan kokonaan jotkut tuotteet, jolloin muita laitteita ei tarvita. Volyymien avulla voidaan valita mitä laitteita on kannattavinta käyttää. Tuoteperheiden avulla voidaan päättää, mikä menetelmä valitaan CAPP:lle. Mikäli tuoteperheitä on monta ja samankaltaisia kappaleita on vähän, kannattaa käyttää generatiivista lähestymistapaa. Jos taas tuoteperheitä on kohtalainen määrä, variantti lähestymistapa on kaikkein soveltuvin (Chang, Wysk et al. 1991, s. 476-478). CAPP systeemissä tarvitaan kaikenlaista tietoa, jotta määritettyä prosessia voidaan perustella ja pitää luotettavana. Taulukko 1 esittää CAPP:n erilaiset tietotyypit ja niiden selvitykset.

Taulukko 1. CAPP:n tietotyypit, mukailen lähdettä (Chang et al. 1991, s. 518-527)

Tietotyyppi	Selvitys
Työkappale	Geometria, koneistettavat pinnat
Tuotteen laatu	Toleranssit, pinnanlaatu, geometriset mitat
Koneistuksen operaatiot	Erilaisilla koneistustavoilla saavutettavat tulokset
Koneen työkalut	Käytettävissä olevat työkalut, teho
Työstö	Millaisia pintoja työkaluilla voidaan koneistaa
Pitimet	Pitimien soveltuvuus tarkkuuksiin ja nopeuksiin
Piirteiden erottelu	Sääntöjä, joiden avulla voidaan määrittää tuotteen geometria
Operaation valinta	Millä operaatiolla saadaan haluttu lopputulos
Operaatioiden järjestys	Määritetään perustuen tiedettyihin rajoitteisiin
Kone-työkalu valinta	Katsotaan millä koneella ja sen työkalulla päästään vaatimuksiin
Työkalun valinta	Millä työkalulla saadaan koneistettua vaadittu pinta

Jigi ja kiinnitys	Valitaan kappaleelle ja siihen kohdistuville operaatioille soveltuvat komponentit
Koneistusparametrit	Mahdollisimman hyvien parametrien valinta tuottavuuteen
Metatieto	Perustelut älykkään ja oppivan systeemin takana

Taulukon pohjalta voidaan todeta, että CAPP:n avulla prosessisuunnittelija käy koko prosessin läpi ja pystyy suunnittelemaan siihen vaadittuja resursseja. CAPP:n tarkoituksena on muodostaa ketju tulevasta raaka-ainemateriaalista aina valmiiseen, viimeisteltyyn lopputulokseen. CAPP:n käyttäminen valmistusprosessissa voidaan todeta olevan CAD:n ja CAM:n välissä. Tämän seurauksena CAPP tarjoaa siis linkin suunnittelun ja valmistuksen välille. CAPP:n onnistuneen käytön tuloksena lopputuloksen valmistumiseen kuluu vähemmän aikaa ja siitä tulee laadukkaampi koska kaikki operaatiot on suunniteltu etukäteen. (Yusof, Latif 2014)

2.1.4 NC-koodi ja postproessori

Mallinnettujen resurssien ja operaatioiden pohjalta suunniteltu tieto tarvitsee muuttaa sellaiseen muotoon, jota työstökone pystyy ymmärtämään. Numeerisesti kontrolloitu (englanniksi Numerical Control, NC) koodi on tietomuoto, jonka avulla suunniteltua dataa voidaan alkaa muokkaamaan sopivaksi simulointi- ja koneistusvaihetta varten. Yleisimmät NC-koodin komennot on esitetty G- ja M-koodien avulla. G-koodilla pystytään kertomaan työstökoneelle, millainen toiminto on tarkoitus suorittaa. Koodin rakenne koostuu kirjaimesta G, jonka jälkeen tulee kaksi numeroa väliltä 00 ja 99. Tämä yhdistelmä sisältää siis maksimissaan sata funktiota, joiden avulla työstökoneelle voi välittää tietoa. M-koodilla on samanlainen rakenne kuin G-koodilla, mutta M-koodin avulla konetta ohjataan esimerkiksi pyörimissuunnan, jäähdytysjärjestelmän ja työkappaleen tarttumisen suhteen. G-koodilla tyypillisiä toimintoja ovat interpolointi, määrätty kiertokulku ja työstöterän kompensoinnit. Taulukko 2 kuvaa NC-koodissa olevia funktioita ja niiden vaihteluvälejä. (Lai 2014, s. 1515-1516)

Taulukko 2. NC-koodissa olevia funktioita, mukailen lähdettä (Lai 2014, s. 1516)

Funktio	Tunnus	Vaihteluväli
Ohjelman nimi	O	4
Komennon numero	N	4
Valmistava funktio	G	3
Koordinaatti	X, Z, U, W, R, K, I	5 tai 3
Syöttö	F	4
Ensisijaisen akselin funktio	F	4
Kierteen leikkauskulma	Q	0~360°
Työkalun funktio	T	4
Kirjava funktio	M	2
Keskeytys	P tai X	5
Aliohjelman numero	P	4
Toistokerrat	L	2

Kuten esitetystä taulukosta huomataan, NC-koodissa on monenlaisia osa-alueita ja niiden avulla pystyy tekemään kaikki tarvittavat operaatiot. Työstökoneesta riippuen NC-koodissa voi olla monia funktioita tai ainoastaan yhdenlaisia funktioita, kuten pelkästään G-koodi. Tästä voidaan tehdä johtopäätös siitä, että NC-koodin tarvitsee olla soveltuva olemassa oleviin olosuhteisiin ja resursseihin nähden. Ainoastaan koodin luominen ei riitä, sillä CNC-koneita on erilaisia ja koodia tulkitaan eri tavoin riippuen koneen ominaisuuksista. Tämän vuoksi käytetään työkalua, jolla luomisen lisäksi varmistetaan koodin oikeellisuus ja välitetään se oikeanlaisessa muodossa määritetyille CNC-koneelle. Tätä työkalua kutsutaan nimellä postprosessori. (She, Lin et al. 2012, s. 638)

Kirjallisuudessa kehitetään jatkuvasti uudenlaisia ja paranneltuja postprosessoreja erityyppisiin tilanteisiin ja näin ollen samaa postprosessoria ei ole optimaalista käyttää kuin parannellun version kehittämiseen saakka. She, Lin et al. (2012) kehittää postprosessoria, joka ei yleiseen tapaan vain generoi sopivaa koodia, vaan pystyy myös muuntamaan toisen tyyppistä NC-koodia halutun laiseksi koodiksi tai sitten takaisin suunnitteluvaiheessa tuotettavaan tietomuotoon. Tällaista postprosessoria kutsutaan nimellä käänteinen postprosessori. Kyseinen ominaisuus tuo lisää joustavuutta datan muokkaamiseen ja mahdollistaa olemassa olevan NC-koodin hyödyntämisen tehokkaammin. Cheypoca, Lekthamrong et al. (2010) puolestaan kehittää postprosessoria yleispäteviin tapauksiin, joissa käytetään vain G-koodia. Tämän perusteena on se, että postprosessorien tarjoajat tekevät ne siten, että vain tietyntyypiset koneet ovat tuettuna. Tang, Yin et al. (2018) esittää kehitetyn postprosessorin CNC-koneelle, jossa optimoidaan kulma-asentoa. Tässä tapauksessa kehitystä ennen ei ollut sopivaa postprosessoria käytetyille koneelle ja fokusalueelle. Postprosessorit ovat siis tärkeässä roolissa tiedon siirtymisessä.

2.1.5 CNC-kone

Tietokoneistettuja numeerisesti kontrolloituja (englanniksi Computer Numerical Control, CNC) koneita kutsutaan CNC-koneiksi. Suunnittelu- ja valmistustieto tulee näihin koneisiin postprosessoituna NC-koodina, joka toimii koneistuksen säätöketjuna. Näiden koneiden avulla saadaan realisoitua tuotettu data ja valmistettua suunniteltu lopputulos. CNC-koneita on monenlaisia ja erilaisilla koneilla pystyy työstämään eri tilanteissa vaadittuja operaatioita. Esimerkkejä CNC-koneista ovat jyrsin, sorvi plasmaleikkuri ja koneistuskeskukset. CNC-koneissa paikantaminen voidaan hoitaa kahdella eri tavalla: Muutosta tutkivassa tavassa otetaan vertailupisteeksi aina edellinen sijainti ja merka-

taan, kuinka paljon liikutaan siihen pisteeseen katsottuna. Absoluuttisessa tavassa puolestaan kaikki paikat lasketaan absoluuttiseen nollopisteeseen, eli muuttumattomaan referenssipisteeseen, vedoten (Chang, Wysk et al. 1991, s. 236).

CNC-koneita voidaan arvioida seuraavien tekijöiden avulla: Tarkkuus, toistettavuus, karan ja akselin moottorin hevosvoimat, akselien lukumäärä, työstöalue ja koneen ohjaimessa olevat ominaisuudet. Koneen tarkkuus aiheutuu ohjaimen ja laitteen tarkkuuksista. Ohjaimen tarkkuus tarkoittaa ohjaimen pienintä tunnistettavaa pituutta ja laitteen tarkkuus puolestaan työstökoneessa olevaa virhettä. Näitä virheitä aiheuttavat koneen osissa olevat toleranssit, työkalujen poikkeumat ja lämpötilasta johtuvat virheet. Toistettavuudella tarkoitetaan sitä, kuinka lähelle kone pääsee edellisiä tuloksia. Karan ja akselin tehojen avulla saadaan selville, millaisia materiaaleja ja kuinka nopeasti voidaan koneistaa. Akselien lukumäärä kertoo, kuinka kompleksisia kappaleita koneella voidaan tehdä. Koneessa ja ohjaimessa voi olla seuraavia ominaisuuksia: työkalun, paletin ja työstettävän kappaleen vaihto, graafinen käyttöliittymä, interaktiivinen operaatio-ohjelmointi (Chang, Wysk et al. 1991, s. 213-214).

Kuten CNC-koneiden arviointiperiaatteista voidaan huomata, CNC-koneilla on keskenään suuriakin eroavaisuuksia ja näin ollen niiden hankintavaiheessa on suunniteltava tarkkaan, millaisesta koneesta on eniten hyötyä yritykselle. Suunnittelussa voidaan vertailla esimerkiksi seuraavia asioita: Koneen käytön omaksumiseen, koneen asentamiseen, kappaleen koneistamiseen kuluvat ajat. Lisäksi koneen, työkalujen, käyttäjien koulutukseen vaadittavat kustannukset on syytä arvioida. Ajan ja hinnan lisäksi kannattaa arvioida myös, minkälaisiin tarkkuuksiin koneella päästään, minkälaisia kappaleita voidaan koneistaa ja kuinka itsenäisesti kone suoriutuu koneistuksesta. Itsenäisyydellä tarkoitetaan sitä, kuinka paljon käyttäjän tarvitsee olla tekemisessä koneen kanssa kappaleiden koneistuksessa. Onnistuneen ja pätevän arvioinnin jälkeen yritys voi luotettavien perusteluiden avulla tehdä päätöksen oikeanlaisen CNC-koneen hankinnasta. (Zębala, Plaza 2014)

2.1.6 CMM

Kappaleen valmistuksen jälkeen tarvitsee usein tarkastaa, täyttääkö kappale vaaditut toleranssit ja muut laadulliset vaatimukset. Tarkastelua varten käytettävä laite on usein koordinaattimittauskone (englanniksi Coordinate-Measuring Machine, CMM), jonka avulla voidaan tutkia kappaleita tarvittavalla tarkkuudella. CMM on ollut tarkkuutensa ja käytettävyytensä ansiosta yksi tärkeimmistä laadunhallintatyökaluista jo monen vuoden ajan. CMM:n avulla kappaleita voidaan tutkia koskettamalla ja ilman kosketusta mutta

kaikista yleisin laadunhallintatapa on käyttää kappaletta koskettavaa pallomaista anturia, joka on kiinnitettyä CMM-koneeseen. Anturi on yleisin siitä syystä että sillä päästään jopa mikrometriä osien tarkkuuksiin riippuen käyttöolosuhteista. Yksittäinen CMM vaatii käyttöönottoa varten pääomaa tuhansista satoihin tuhansiin euroihin, joten koneiden hankkimiselle tulee olla perusteltu tarve. (Mian, Al-Ahmari 2014, s. 78)

CMM:n kehitystä tutkittaessa voidaan tunnistaa seuraavia vaiheita: Manuaalinen suunnittelu, tietokoneohjelman avulla suunnittelu ja ammattilaisohjelman eli nimenomaan CMM-mittauksiin luodun ohjelman avulla suunnittelu. Yleisin menetelmä nykypäivänä on tietokoneohjelman avulla suunnittelu. Suunnittelutavasta riippumatta yhteinen elementti on tutkittava kappale, josta voidaan tutkia mittaustavasta riippuen geometriaa, toleransseja tai niiden yhdistelmiä. Tutkimisprosessi koostuu muutamasta avainelementistä, jotka ovat radan suunnittelu, törmäyksien välttäminen, luoksepäästävyys analyysi, työkappaleen asetus sekä mitta-anturin konfigurointi. Kokonaisien tutkimuksien suunnittelu sisältää kaikki mainitut avainelementit ja niiden suorittaminen on kannattavaa optimoida. (Stojadinovic, Majstorovic et al. 2016, s. 327)

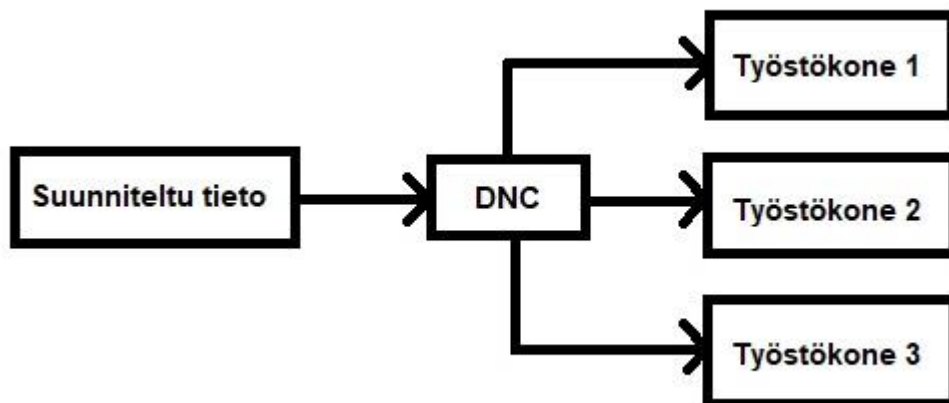
Tutkimisen optimoinnissa radan suunnittelulla voidaan ensisijaisesti nopeuttaa tutkimisprosessiin kuluva aikaa mutta myös minimoida osien kulumista. Ratojen luomisessa on tunnistettu erilaisia lähestymistapoja, joista kirjallisuudessa mainitaan muun muassa geneettinen algoritmi ja raaka voima. Lähestymistapoja voidaan valita tutkittavien kappaleiden perusteella. Esimerkiksi prismaattisissa eli ei-pyörähdyssymmetrisissä kappaleissa sopiva lähestymistapa on geneettinen algoritmi, sillä se on tehokas ja pystyy ratkaisemaan myös kompleksisia ongelmia. Geneettisessä algoritmista määritetään aluksi tutkittava mittaajärjestys jonka jälkeen valitaan tärkeimmät mittauskohdat lähempään tarkasteluun. Seuraavana tutkitaan, saadaanko järjestyksiä vaihtamalla ja yhdistelemällä mittauksiin kuluva aikaa optimoitua. Optimoinnin lopuksi turhat operaatiot poistetaan. Raan voiman periaatteessa generoidaan ja arvioidaan kaikki mahdolliset vaihtoehdot, jonka johdosta se soveltuu yksinkertaisiin ongelmiin. Ensiksi määritellään kaikki mitattavat piirteet, jonka jälkeen tutkitaan kaikki mahdolliset mittaajärjestykset ja valitaan paras. (Mian, Al-Ahmari 2014)

CMM ohjelman voidaan sanoa olevan ohjenuora, joka mahdollistaa CMM-tutkimisen. CMM ohjelman teossa tarvitsee huomioida, ovatko CMM-koneet piirre- vai tietoperusteisia. Piirreperusteisissa mittauskoneissa tunnistetaan tutkittavan kappaleen muotoja ja verrataan niitä asetettuihin rajoitteisiin, jotka on syötetty järjestelmään. Tietopohjaisissa koneissa CMM-koneet osaavat lukea itse rajoitteita mallinnetusta kappaleesta ja käytettävistä standardeista, joten niissä ei erikseen tarvitse syöttää rajoitteita. CMM ohjelman

luomisen ja toteuttamisen automatisointia pyritään kehittämään tehokkuuden lisäämiseksi. Automatisoidussa CMM ohjelmassa mallinnetun kappaleen ja käytettävän CMM-koneen ominaisuuksista tulee löytyä kaikki tarvittavat tiedot, jotka vaaditaan CMM ohjelman tekoon. Näihin asioihin perehdytään diplomityön myöhemmässä vaiheessa. (Stojadinovic, Majstorovic et al. 2016)

2.1.7 DNC

Suora numeerinen ohjaus (englanniksi Direct Numerical Control tai Distributed Numerical Control, DNC) toimii valmistavassa teollisuudessa suunnittelun ja työstökoneen välisessä tiedonsiirrossa. Kun suunnittelija on laatinut valmistuksen edellyttämät parametrit, joita ovat esimerkiksi valitut työkalut, kiinnitys, työkaluradat ja aikataulut, voidaan postprosessoitu NC-koodi lähettää työstökoneelle koneistusta varten. Tuotannossa voi olla useita erilaisia työstökoneita joilla on keskenään erilaisia ominaisuuksia, joten onkin ensiarvoisen tärkeää, että luotu tieto siirtyy soveltuvalle työstökoneelle. Kuva 1 esittää DNC:n tarkoituksen. (Jiang 2011, s. 3388)



Kuva 1. DNC:n tarkoitus, mukailen lähdettä (Jiang 2011)

Ylläoleva kuva havainnollistaa sen, missä kohtaa valmistusprosessia DNC sijaitsee ja mikä on sen tarkoitus. Jos suunnittelija luo tiedon, joka soveltuu vain työstökoneelle 2, DNC:n tehtävänä on siirtää se kyseiselle koneelle. Voi myös olla, että koneet ovat keskenään identtisiä, jolloin DNC:n tehtävänä on siirtää tieto vapaana olevalle koneelle. DNC:n ominaisuuksiin kuuluu valmistuksen tilan päivittäminen tuotannonohjausta varten ja sen avulla voidaan tunnistaa jos lähetetyssä tiedossa on ongelmia esimerkiksi työkalujen suhteen. DNC:lle asetetaan yleensä seuraavia vaatimuksia: vakaus, joka tarkoittaa luotettavaa laitteistoa, pieniä epäonnistumistodennäköisyyksiä ja luotettavaa tiedonsiir-

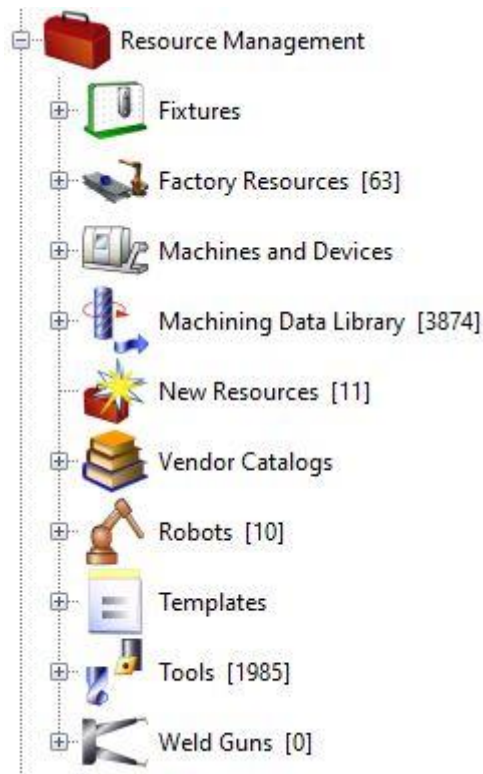
toa. Toisena vaatimuksena on keskittäminen, jolla tarkoitetaan usean työstökoneen toimimista DNC:ssä sujuvasti. Kolmantena vaatimuksena on tehtaan layoutiin sopiva kommunikointietäisyys DNC:n ja työstökoneen välillä. (Ni, Gao 2013)

DNC koostuu kommunikointiverkosta ja kommunikointiohjelmistosta. Kommunikointiverkko rakentuu normaalisti kontrolloivasta tietokoneesta, digitaalisista työkaluista, käyttöliittymästä, kääntämoduuleista ja datanvälitysvälineestä. Kommunikointiohjelmisto puolestaan koostuu kolmesta toiminnallisesta moduulista: Tiedostojen lähetysohjelma lähettää DNC:hen tulevan tiedon työstökoneelle, jonka johdosta siinä kulkee paljon tietoa. Tämä voi aiheuttaa käyttöliittymän hetkittäistä käyttämättömyyttä. Tiedostojen vastaanotto-ohjelma on vastuussa työstökoneelta tulevan tiedon välittämisestä niitä kontrolloivalle taholle, jolloin työstön tilaa voidaan seurata. Etäohjausohjelma mahdollistaa työstökoneelle tulevien ohjelmien kutsumisen työstökoneen käyttäjän toimesta. Tämän johdosta käyttäjän ei tarvitse poistua työstökoneen äärestä seuraavaa ohjelmaa käsitellessä. DNC on siis valmistusprosessin osa, jossa varmistetaan suunnitellun tiedon kulkeutuminen ja sopivuus työstökoneelle. (Jiang 2011, s. 3388-3389)

2.1.8 MRL

Valmistusresurssien kirjasto (englanniksi Manufacturing Resource Library, MRL) tarkoittaa paikkaa, joka sisältää kaikki valmistusprosesseja tukevat tiedot. Nämä tiedot ovat sekä yrityksen sisällä olevia resursseja tai sitten ulkopuoliselta taholta, esimerkiksi työkalujen myyjältä, saatavia tietoja. MRL:n tarkoituksena on tarjota mahdollisimman yksinkertainen tapa päästä käsiksi kaikkiin tarvittaviin tietoihin ja mahdollistaa sujuvan valmis-

tusprosessin suunnittelu. Tämän lisäksi MRL:n käyttäjän tulisi pystyä lisäämään ja poistamaan resursseja vaivattomasti kirjaston ajan tasalla pitämiseksi. Kuva 2 esittää esimerkin MRL:n rakenteesta Siemensin ohjelmassa Teamcenter. (Siemens 2019)



Kuva 2. MRL Siemensin ohjelmassa Teamcenter

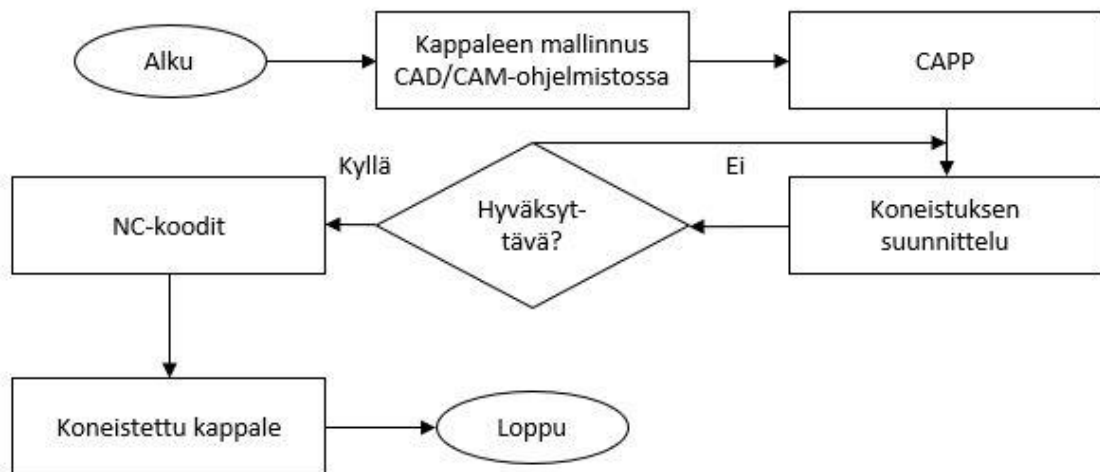
Kuten ylläolevasta kuvasta käy ilmi, MRL:n rakenne voi koostua kuvassa näkyvistä osaluista jotka ovat: kiinnittimet, tehdasresurssit, koneet ja laitteet, koneistusdatakirjasto, myyjien katalogit, robotit, mallipohjat, työkalut ja hitsauslaitteet. Koska MRL sisältää massiivisen määrän dataa, on se tarpeellista rakentaa vaiheittain. Tärkeimpänä vaiheena voidaan pitää valmistuksessa käytettäviä resursseja, sillä niiden määrittely on välttämätöntä yrityksen tuotannolle. Kuvassa tällaisia resursseja ovat kiinnittimet, koneet ja laitteet sekä työkalut. Tämän jälkeen voidaan määrittellä loput resurssityypit sitä mukaa kun tulee tarvetta. Tehdasresursseilla voidaan mallintaa virtuaalinen malli tuotannosta simulointia varten. Koneistusdatakirjasto pitää sisällään toteutuneiden koneistuksien parametrit ja niiden tulokset, jonka johdosta se kasvaa jatkuvasti. Myös työkalujen määrä kasvaa ajan kuluessa, sillä sinne poimitaan myyjien katalogeista tarvittavia työkaluja.

MRL:n avulla valmistuksen suunnittelusta tulee siis yksinkertaisempaa ja sujuvampaa. (Yang, Zhang et al. 2008)

2.2 CAM-prosesseja kirjallisuudessa

Kirjallisuudessa eri tahoilla on monia erilaisia tapoja määritellä, mistä osista CAM-prosessit koostuvat. Vaikka eri tahoilla olisi samanlainen rakenne prosesseissa, voi niiden toteutustapoja verrattaessa tulla merkittäviäkin eroavaisuuksia käytössä olevien resursien ja toimintatapojen seurauksina. Tässä alaluvussa käydään läpi kirjallisuudessa olevia määritelmiä ja toteutustapoja CAM-prosesseihin liittyen. Kohdeyrityksen CAM-prosessia tarkkuusvalmistuksessa käydään läpi käytettävän prismaattisen esimerkkitaupauksen kautta työn luvussa 5.

Ismail, Lim et al. (2003) esittää monimutkaisen mekaanisen kappaleen mallinnuksen ja koneistuksen CAM-prosessin avulla. Kappaleena on laskutelineen yhdistimen hela, jonka tarkoitus on siirtää laskutelineeseen kohdistuvia kuormituksia lentokoneen runkoon. Käytettävänä CAD/CAM-ohjelmistona on UniGraphics. Valmistettavan kappaleen materiaali on alumiini ja 3-akseliseen koneistukseen kuuluva aika on noin 850 minuuttia. Kuva 3 havainnollistaa käytettävän CAM-prosessin rakenteen.



Kuva 3. Lähteen (Ismail, Lin et al. 2003, s.184) mukainen CAM-prosessi

Ylläolevan kuvan mukaan ensimmäisenä askeleena on kappaleen CAD-mallin luominen, jonka jälkeen kohdassa CAPP suunnitellaan prosessin rakenne ja siihen vaadittavat resurssit. CAPP-vaiheessa tapahtuu lähteen mukaan seuraavin asioita: Sisäistetään mallinnetun kappaleen rakenne, tunnistetaan kaikki koneistettavat piirteet huomioineen, valitaan koneistukseen tulevan materiaalin muoto, tunnistetaan peruselementit, hahmotel-

laan operaatioiden järjestys ja niissä käytettävät työkalut, hahmotellaan kiinnitys, hahmotellaan operaatiokohtaiset työstöparametrit ja valmistetaan lopullinen CAPP-dokumentti. CAPP vaatii työntekijältä tietämystä ja kokemusta sekä suunnittelusta että valmistuksesta. (Ismail, Lim et al. 2003)

Rakenteen ollessa selvillä voidaan suunnitella koneistus. Lähteessä mainitaan tähän vaiheeseen liittyen työkalujen, operaatiotyyppien ja koneistusparametrien lopullinen valinta. Valinnoissa käytetään hyväksi ohjelman toimintojen sekä eri työkalujen parametrien arvoja. Tämän jälkeen koneistuksen suunnittelussa syntynyt tieto simuloidaan ja ajetaan postprossessorin läpi. Tätä ei esitetä kuvassa mutta se tapahtuu Koneistuksen suunnittelu- ja Hyväksyttävä? -kohtien välissä. Jos hyväksyntää ei tule, siirrytään takaisin koneistuksen suunnitteluun ja tehdään tarvittavia muutoksia. Hyväksynnän jälkeen on muodostettu koneistuksessa käytettävät NC-koodit. Kuvassa tätä ei näy, mutta ennen koneistettua kappale -vaihetta koodit ladataan CNC-koneelle. Tämän jälkeen kappale koneistetaan ja saadaan CAM-prosessi päätökseen. Lähteen perusteella saa varsin hyvän kuvan CAM-prosessista, mutta siinä ei esitetä yksityiskohtaisesti kaikkia prosessin vaiheita (Ismail, Lim et al. 2003)

CAM-prosessia esitetään teoreettisella tasolla lähteessä (Yusof, Latif 2014). CAPP:n kuvaaminen on lähteen pääasiallinen tarkoitus, mutta CAM-prosessista on tehty kuvaus, johon CAPP on sijoitettu. Kuvauksessa prosessin järjestys koostuu seuraavista vaiheista: **Idean luominen, CAD, CAPP, CAM, CNC, Valmis tuote**. Tämä kuvaus on esitetyiltä osin yhtenevä aikaisemman tapauksen kautta mutta esitystapa on vieläkin yleisemmällä tasolla. Lähteen CAM-prosessia kuvataan seuraavasti: Ensimmäisenä valmistettavasta kappaleesta kehitetään idea, jonka pohjalta suunnitellaan 3D-malli kuvaamaan kappaleen muotoa ja piirteitä. Tämän jälkeen luodaan CAPP-dokumentti, jossa otetaan huomioon prosessin myöhemmissä vaiheissa tapahtuvat operaatiot. Seuraavana suunnitellaan kappaleen koneistus, jonka jälkeen suunnittelutieto menee työstökoneelle. Lopulta suunniteltu kappale on koneistettu ja prosessi saadaan päätökseen. Esitetystä lähteestä saa hyvin yleisellä tasolla kuvan CAM-prosessista.

Siemens (2019) esittää oman näkemyksensä CAM-prosesseista ja havainnollistaa siihen ratkaisua yrityksen tarjoamia tuotteita hyödyntämällä. Motivaationa CAM-prosessin kehittämiseksi kerrotaan olevan yleinen tilanne, jossa prosessin eri osa-alueet, kuten tuotteen suunnittelu, valmistuksen suunnittelu ja tuotanto, eivät ole yhdistettyinä toisiinsa.

Tämän johdosta aiheutuu ongelmatilanteita kun ei tiedetä, onko uusin tieto muuttumattomasti käytössä olevaan osa-alueen tietoon nähden. Kuva 4 esittää Siemensin kuvauksen CAM-prosessista.



Kuva 4. CAM-prosessi perustuen lähteeseen (Siemens 2019)

Ylläolevan kuvan tulkinnassa prosessi kulkee vasemmalta oikealle, noudattaen esitettyjä osa-alueita. Alussa, aikaisempia lähteitä mukaillen, kappaleen geometria suunnitellaan CAD/CAM -ohjelmistolla. Tämä mahdollistaa CMM ohjelman tekemisen ja kappaleen CAPP:n luomisen (ylläolevassa kuvassa nimellä Process planning). Kun CAPP on valmis tai tarpeeksi pitkällä, voidaan määrittellä käytettävät kiinnitykset ja työkalut ja siirtyä CAM-suunnitteluun. Samalla kun valmistusta suunnitellaan, voidaan hyödyntää työkalukirjastossa (ylläolevassa kuvassa nimellä Tool libraries) olevia työkaluja ja nopeuttaa tätä vaihetta. CAD/CAM -suunnittelun jälkeen saadaan muodostettua postprosessoitu ja simuloinnilla validoitu NC-koodi, jota ylläolevassa kuvassa ei esitetä. (Siemens 2019)

Seuraavana vaiheena siirrytään tuotannon puolelle, jossa kappale lopulta koneistetaan. Ennen koneistuksen mahdollistamista tarvitsee selvittää CNC-koneiden tilanne (ylläolevassa kuvassa kohta Shop floor data access) ja ladata NC-koodi DNC:n avulla määritellyyn CNC-koneeseen. Kun NC-koodi on oikealla koneella, koneistajan tehtävänä on validoida eli varmistaa lopullinen CNC-koneella toteutettava ohjelma (ylläolevassa kuvassa nimellä Shop floor program validation). Samanaikaisesti kone valmistetaan koneistusta varten tuomalla ja tarkistamalla ohjelmassa määritellyt työkalut (ylläolevassa kuvassa nimellä Tool management). Lopulta kappale koneistetaan määritettyjen resurssien mukaisesti kohdassa SINUMERIK CNC. SINUMERIK on Siemensin käyttöliittymä, jonka avulla CNC-koneen käyttäjä näkee visuaalisesti tuotetun NC-koodin. Koneistuksen jäl-

keen valmiille kappaleelle suoritetaan vielä tehdyn CMM ohjelman mukainen tutkinta (yläolevassa kuvassa nimellä CMM Inspection execution), jonka tuloksena saadaan haluttu lopputulos ja prosessi on suoritettu. (Siemens 2019)

Siemensin mallissa esitetään myös CAM-prosessiin kuuluvia työntekijöitä ja heidän tehtäviään. Työkalujen hallinnasta ja muiden resurssien suunnittelusta vastaava työntekijä (Siemensin mallissa nimellä Tooling Author) on vastuussa MRL:n ylläpidosta mahdollistaen prosessin myöhemmissä vaiheissa tietojen nopean hyödyntämisen. Tehtäviin kuuluu yrityksen resurssien määrittäminen ja päivittäminen MRL:ään, kokoonpanojen tekeminen esimerkiksi työkaluille sekä suositeltujen syötön ja nopeuksien kirjaaminen. Prosessisuunnittelijan tehtävänä on suorittaa CAPP MRL:n resurssien pohjalta ja näin ollen tarvittavan tiedon siirtäminen eteenpäin suunnittelijoille. Suunnittelija (Siemensin mallissa NC programmer) pääsee käsiksi tarvittaviin resursseihin. Tämän pohjalta hän määrittää CNC-koneen, luo koneistuksen parametrit NC-koodin avulla ja tekee työohjeet. Työkaluvastaava (Siemensin mallissa Tool setter) pääsee käsiksi MRL:ään ja valmistelee CNC-koneen varastosta löytyvien resurssien avulla. Lopuksi koneistaja käyttää NC-koodia ja työohjeita kappaleen koneistuksessa. Siemensin CAM-prosessin kuvaus on kokonaisuutena monipuolinen ja selkeä. (Siemens 2019)

2.3 Ohjelmointirajapinnat ja työkalut

Ohjelmointirajapinta (englanniksi Application Programming Interface, API) tarkoittaa tekoälyn hyödyntämistä, internetiin liittyvää teknologiaa ja digitaalista tuotetta, jonka avulla yrityksen liiketoimintaa saadaan tehostettua. API:t koostuvat tyypeistä, metodeista, ja paketeista, joita tarjoavat palvelujen kehittäjät. API tyyppejä ovat julkinen, partneri ja privaatti. Julkisessa tyyppissä kehitetty tieto on kaikkien käytettävissä, partnerityypissä puolestaan kehittäjän ja yhteistyökumppaneiden käytössä, privaattityyppi tarkoittaa vain kehittäjän yrityksen käytössä olemista. Metodeilla tarkoitetaan muun muassa eri ohjelmointikieliä (Java, Python, C++), joiden avulla tieto saadaan luotua. Paketeilla tarkoitetaan puolestaan metodien sisällä olevia osia, jonka komennot keskittyvät tiettyyn osaluueeseen, kuten esimerkiksi CAM:iin. (Meng, Steinhardt et al. 2018, s. 296-304)

Näiden olemassa olevien apuvälineiden avulla sovellukset voivat hyödyntää eri lähteissä kehitettyä tietoa, jonka johdosta sitä ei tarvitse kehittää uudelleen käytettävään tilanteeseen. Jos tilanteeseen ei löydy sopivaa ratkaisua, sellaisen voi tehdä itse ja tallentaa myöhempää käyttöä varten. Tämän johdosta API:en käyttämisellä voi myös hankkia itselleen markkinaetua joko kauppaamalla kehitettyä tietoa tai käyttämällä sitä vain omaan tarkoitukseen. Tarkoituksena on siis nopeuttaa ja tehostaa prosessien kulkua. API:en

käyttöön otossa kysytään yleisimmin seuraavia asioita: Mitä sillä voi tehdä, miten sen käyttöönotto vaatii, millainen rakenne sillä on ja kuinka sillä voi ohjelmoida? Näitä kysymyksiä voidaan käyttää hyväksi esiteltäessä API:en ominaisuuksia. (Meng, Steinhardt et al. 2018)

Siemens (2019) tarjoaa CAD/CAM ohjelmaansa NX monenlaisia julkisia ohjelmointityökaluja API:en hyödyntämiseen. NX Open on julkinen API:en kokoelma, jonka avulla voi luoda eri ohjelmointikielien avulla sovelluksia. Nämä sovellukset voivat esimerkiksi automatisoida monimutkaisia ja toistettavia tehtäviä. SNAP on helposti opetettava ohjelmointityökalu, jonka avulla voidaan automatisoida yksinkertaisia toistettavia operaatioita. Knowledge Fusion on työkalu, jonka avulla kappaleisiin voi lisätä erilaisia ominaisuuksia sääntöjen avulla. GRIP on myös automatisointiin liittyvä apuväline. Menuscript mahdollistaa ohjelman valikkojen muokkaamisen ja lisäämisen käytettävyyden parantamiseksi. Block UI Styler mahdollistaa ikkunoiden rakentamisen, joista voi lukea ja joihin voi syöttää tietoa. Tässä työssä keskitytään NX Openin ominaisuuksiin ja hyödynnetään niitä. Kuva 5 Esittää Siemensin tarjoamat oppaat NX Openin sisältämään kokonaisuuteen liittyen.



Kuva 5. Siemensin oppaat NX Openiin, perustuen lähteeseen (Siemens 2019)

Ylläolevassa kuvassa huomataan, että soveltuvia metodeja ovat C/C++, Java, Python ja .NET. Lisäksi ohjeita löytyy GRIP:n ja CAE-alueiden käyttöön. Ohjelmointikielikohtaiset viittausohjeet löytyvät merkityksineen avaamalla linkin, jossa lukee Reference Guide. Ylläolevassa kuvassa on laajennettu NX Open Python, jonka alla kyseiseen kieleen liittyvä viittausohje on. Kun viittausohje on avattu, käyttäjä näkee käytettävissä olevia paketteja

ja niiden kuvauksia. CAM-prosessia tutkittaessa voidaan Python-kielen referenssihjeesta valita CAM:iin liittyvä paketti. Paketin sisältä näkee CAM:iin liittyviä komentoluokkia, joilla tarkoitetaan tiettyjen komentojen sisältämää kokonaisuutta. Komentoluokkien sisältä löytyy niihin liittyvät komennot ja kirjoittamismuodot koodauksessa. (Siemens 2019)

Kun tiettyä komentoa käytetään, voidaan se havaita selkeästi koodin rakenteesta. Kaikki CAM-pakettiin liittyvät komennot on nimetty seuraavalla tavalla: NXOpen.CAM.X.Y, missä X tarkoittaa käytettävien luokkien nimiä ja Y tarkoittaa käytettävää komentoa. Jos luokan sisällä on toinen luokka, se tulee nimeämisessä ensimmäisen luokan jälkeen pisteellä erotettuna. Kaikki luokat tulevat esittämistavassa kohtaan X. Jokaisen luokan sisältä löytyy tiedot siitä, minkälaisia ominaisuuksia, menetelmiä ja listauksia siihen kuuluu. Tämän lisäksi jokaisesta ominaisuudesta löytyy yksityiskohtainen opas sen käyttämiseen ja siitä saatavaan tietotyyppiin. Tietotyyppiä ovat esimerkiksi numerot ja tekstit. Kuva 6 esittää esimerkkipomennon NXOpenissa. (Siemens 2019)

GetBoolValue

CAMAttribute.GetBoolValue

Get the feature attribute boolean value for a boolean type attribute

Signature `GetBoolValue()`

Returns:

Return type: `bool`

New in version NX8.0.0.

License requirements: None.

Kuva 6. NXOpenin esimerkkipomento

Kuten edellisestä kappaleesta ja ylläolevasta kuvasta huomataan, komennon käyttämisen oppaassa esitetään ensiksi komennon nimi ja kerrotaan, millaisesta komennosta on kyse. Seuraavaksi esitetään komentoon viittaaminen ohjelmoinnissa. Lisäksi havainnollistetaan, millaisia ominaisuuksia komento palauttaa ja millaisia lisenssejä se vaatii käyttämisen mahdollistamiseksi. Komennot voivat palauttaa oletuksena myös ohjelmassa olevien toisien komentojen arvoja, mutta nimeämiseen tällä ei ole vaikutusta. Kuvassa oleva komentoon viitataan ohjelmoitaessa kokonaisuudessaan nimellä

`NXOpen.CAM.CAMAttribute.GetBoolValue`. Komento on siis osa CAM attribuutin määrittelyä ja se palauttaa Boolean arvon, joka on Pythonin ohjelmointikielessä joko True tai False (suomeksi tosi tai epätosi). Ohjelmointikielestä riippuen arvona voi olla myös 1 ja 0, missä 1:n merkitys on tosi ja 0:n epätosi. NXOpenin avulla voidaan siis kehittää olemassa olevaa versiota CAD/CAM ohjelmasta NX. Lokakuussa vuonna 2019 ohjelmointikielen Python referenssiopas NX:n versiolle 12.0 löytyy osoitteesta (https://docs.plm.automation.siemens.com/data_services/resources/nx/12/nx_api/custom/en_US/nxopen_python_ref/index.html).

2.4 Yhteenveto

CAM-prosessista puhuttaessa kuulijalla voi helposti tulla mieleen pelkästään CAM:n sisältö, joka rajoittuu kappaleen valmistuksen mallintamiseen CAD/CAM ohjelmaa hyväksikäyttäen. Todellisuudessa tämä on vain osana CAM-prosesseja. Lukijalle halutaan antaa hyvät lähtökohdat CAM-prosessien rakenteen sisäistämiseen, jonka johdosta aluvussa 2.1 käydään läpi CAM-prosesseissa yleisesti käytössä olevia käsitteitä. Käsitteet sisäistettyään lukijalla on tarvittavat valmiudet lähteä tutkimaan aihepiiriin liittyvää tietoa. Pelkkien käsitteiden avulla prosessia ei voi ymmärtää vaan sitä varten käsitteet on laitettava järjestykseen ja konteksteihinsa. Prosessien yhteydestä voidaan kuitenkin huomata, että selitetyjä käsitteitä esiintyy jokaisessa variaatiossa, jonka seurauksena prosessien ymmärtäminen on yksinkertaista.

Kun CAM-prosesseja käydään läpi, voidaan huomata niiden olevan keskenään osittain erilaisia. Tämä johtuu siitä, että erilaisia tuotteita ja resursseja käyttävät yritykset haluavat hyödyntää mahdollisimman paljon omia resurssejaan, jonka johdosta erilaisista lähestymistavoista on tarkoituksena valita paras mahdollinen omaan toimintaan liittyen. Erilaisissa lähestymistavoissa huomionarvoista on se, että jos samalla palveluntarjoajalla on ratkaisuehdotuksia koko CAM-prosessin sisältöön, painottavat ne helposti omien tuotteidensa käyttämistä. Esimerkiksi Siemensin prosessikuvauksesta tämä käy ilmi. Toisaalta, jos yritys on kykenevä toteuttamaan CAM-prosessinsa saman palveluntarjoajan tuotteilla, yksinkertaistaa se prosessien toteuttamista ja varmistaa kaikkien resurssien yhteensopivuuden. Lukijan kannattaa myös pitää mielessään, että vaikka lähestymistavoissa esitettäisiin todella paljon erilaisia komponentteja, eivät ne kaikki ole

välttämättömiä. Esimerkiksi ilman työstökoneen jatkuvaa yhteyttä järjestelmään CAM-prosessi voidaan suorittaa alusta loppuun.

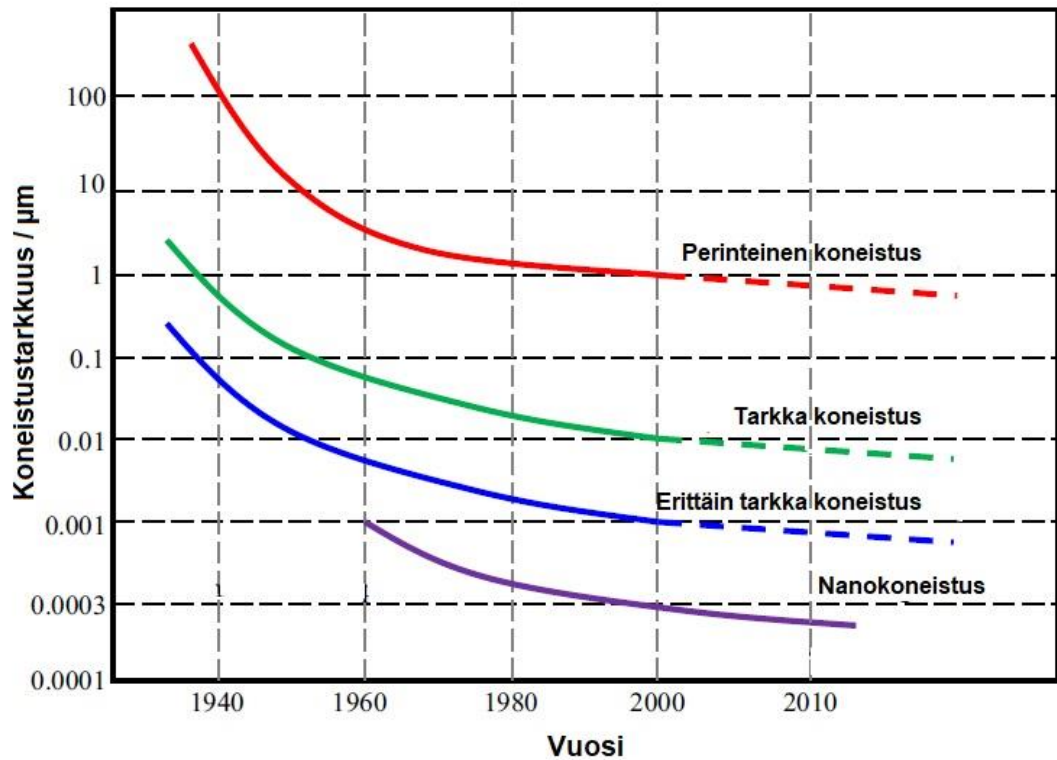
Kun CAM-prosesseja aletaan muokkaamaan ja optimoimaan yrityksen omaa käyttöympäristöä silmälläpitäen, on hyvä saada käsitys erilaisista ohjelmointirajapinnoista ja työkaluista. Näiden avulla pystytään lisäämään ja modifioimaan tietoa, jota tarvitaan CAM-prosessin läpiviemiseksi. Tällaisen muokkaamisen avulla halutaan yksinkertaistaa ja nopeuttaa prosessien läpivientiä ja sitä kautta tehostaa yrityksen toimintaa. Apuvälineiden rakenteita ja ominaisuuksia käydään läpi, jonka johdosta lukija voi mahdollisesti huomata jonkinlaisen potentiaalisen ratkaisutavan tämänhetkiseen ongelmaansa. Lisäksi alaluvussa 2.3 käytävä asia toimii pohjana työssä päädyttäviin tuloksiin, jotka ovat ratkaisuja toiminnan tehostamiseksi kohdeyrityksessä. Luku kaksi tarjoaa siis hyvät lähtökohdat CAM-prosessien oppimiselle ja potentiaalisen kehitystavan tunnistamiselle.

3. TARKKUUSVALMISTUS

Hienomekaniikka (englanniksi Precision Engineering) on laaja kokonaisuus, joka on syytä määritellä ennen kuin ruvetaan tarkastelemaan tarkkuusvalmistusta. Ajankohdasta ja kysyttävästä henkilöstä riippuen hienomekaniikan määrittelyjä on monenlaisia, aina yleiseltä tasolta erittäin spesifeihin rajauksiin. Hienomekaniikka voidaan määritellä olevan nykyisen teknologian maksimaalista hyödyntämistä ensimmäisten joukossa, tähdäten kohti seuraavaa desimaalitarkkuutta. Toinen määritelmä hienomekaniikalle on suunnittelun, kehityksen ja mittauksen kattaminen korkean tarkkuustason komponenteille. Tämä sisältää myös työkalujen, mittalaitteiden ja systeemin suunnittelun tarkkuuden olemisten tärkeimpänä tarkastelukohteena. Suurin osa määritelmistä viittaa pieniin ja tarkkoihin kappaleisiin, joiden mitat on toleroitu erittäin tarkasti. Hienomekaniikka voidaan jakaa moneen eri osa-alueeseen, joista tässä työssä on tarkastelun kohteena tarkkuusvalmistus. (Dornfeld, Lee 2008, s. 1-2)

Tarkkuusvalmistus eroaa Dornfeldin ja Leen kirjoittaman teoksen (2008, s. 2-4) mukaan hienomekaniikasta seuraavalla tavalla: Kun hienomekaniikassa keskitytään kappaleessa oleviin kokoihin, tarkkuusvalmistuksessa paneudutaan puolestaan kappaleen luomiseen ja valmistamiseen. Tarkkuusvalmistuksen voidaan siis määritellä olevan suunnitellun kappaleen valmistamista tutkiva alue, jonka tarkastelun keskipisteenä on tarkkuus. Historian saatossa tarkkuusvalmistuksen määritelmä on pysynyt samana, mutta eri tarkkuusluokkien lukuarvot ovat muuttuneet jatkuvasti pienemmäksi.

Perustuen diplomityön aikana tehtyihin haastatteluihin ja omakohtaisiin havaintoihin, voidaan arvioida tämänhetkisessä konepajateollisuudessa liikuttavan suurilta osin tarkkuusvalmistuksen teoriassa olevalla perinteisen koneistuksen alueella. Jos nykytilan koneistustarkkuuksia skaalataan niin tarkkana koneistuksena voidaan pitää jopa kymmenien mikrometrien arvoja. Tämän johdosta tarkkuusvalmistuksen teorian ja nykytilassa käytettävien käsitteiden välillä syntyy usein eroavaisuuksia. Diplomityön teoriaosuudessa kerätyn tiedon tarvitsee perustua kirjallisuudessa olevaan, julkaistuun materiaaliin. **Lukijan on tärkeää ymmärtää että seuraavaksi esitettävässä kuvassa olevia käsitteitä ja tarkkuusluokkia ei ole skaalattu tämänhetkiseen konepajateollisuuden vaan se perustuu puhtaasti tarkkuusvalmistuksen teoriaan.** Kuva 7 esittää tarkkuusvalmistuksen lukuarvot ja luokittelun ajan saatossa.



Kuva 7. Tarkkuusvalmistuksen koneistustarkkuudet ja luokittelu, mukailien lähteitä (Dornfeld, Lee 2008, s. 6, Zhang, To et al. 2015, s. 77)

Esitetystä kuvasta voidaan havaita tarkkuusvalmistuksen jakautuvan neljään eri kategoriaan. Kategoriat ovat perinteinen, tarkka, erittäin tarkka ja nanokoneistus. Kuvaa tarkastelemalla huomataan, että ajan kuluessa tapahtuneen tarkkuuden pienenemisen johdosta kategorioissa saattaa tulla sekaannuksia jos ei tiedetä viitattavan tiedon ajankohitaisuutta. Esimerkiksi jos puhutaan vuonna 1950 olleesta erittäin tarkasta koneistuksesta, kyseessä onkin nykypäivänä eri kategoria, tarkka koneistus. Kategorioiden tarkkuuksien parantuminen johtuu erilaisten osa-alueiden parantumisesta, jotka vaikuttavat muun muassa kappaleen pinnankarheuteen ja suunnittelussa oleviin toleransseihin. Näistä kerrotaan lisää myöhemmissä alaluvuissa (Zhang, To et al. 2015, s. 77-79)

Perinteisessä koneistuksessa voidaan puhua aina satojen mikrometrien tarkkuudesta mikrometrin tarkkuuteen, mikä tekee siitä kaikista laajimman tarkkuusalueen kattavan koneistustavan. Perinteisen koneistuksen avulla voidaan tehdä muun muassa kulkuvälineiteollisuuden, optiikan runkojen tai pidikkeiden, erilaisten laitteiden ja koneiden tarvitsemia osia. Käytettäviä laitteita ovat esimerkiksi sorvit, jyrsimet, hiomakoneet ja CNC-koneet. Tarkassa koneistuksessa puolestaan tarkkuudet ovat muutamien mikrometrien ja mikrometrin sadasosien välisellä alueella. Tyypillisiä tarkalla valmistuksella tehtäviä kohteita ovat esimerkiksi laakerit, venttiilit, vaihteet ja linssit. Tarkassa koneistuksessa

käytetään tarkkoja hiomakoneita, sorveja ja muita koneita, joiden avulla päästään tavoiteltuihin tarkkuuksiin. (Dornfeld, Lee 2008, s. 4-6)

Erittäin tarkassa koneistuksessa liikutaan mikrometrin kymmenyksien ja nanometriä kymmenyksien rajaamalla alueella. Huomionarvoinen tosiasia on, että perinteiseen koneistukseen verrattuna tarkkuudet ovat 100 kertaa suuremmat koneistuksessa ja 1000 kertaa tarkemmat pinnanlaadussa. Tällä koneistustavalla voidaan valmistaa erittäin tarkkoja X-Y -pöytiä tai esimerkiksi materiaalien kovuuksia testaavan timanttiterän säteitä. Timanttipinnoitteisella terällä koneistavat erittäin tarkat sorvit ovat tyypillisin esimerkki käytettävistä koneista (Zhang, To et al. 2015, s. 77-79). Nanokoneistuksen tarkkuus on nanometrin ja sen kymmenyksien rajaama alue. Menetelmää ei ole tutkittu niin paljon kuin edellä mainittuja, mutta nanokoneistuksella on potentiaalia muun muassa autoteollisuudessa, biolääketieteessä, terveydenhoidossa sekä lentokoneiteollisuudessa. Valmistettavia tuotteita ovat gyroskoopit, painesensorit ja lähettimet. Nanokoneistusta hyödyntäviä valmistusmenetelmiä ovat kipinätyöstö (englanniksi Electrochemical Discharge Machining, EDM) sekä todella tarkat koneistuskeskukset. Tällä hetkellä nanokoneistusta ei voida toteuttaa alhaisilla kustannuksilla tai hyödyntämään tarpeeksi tarkkaa toistettavuutta, mutta tulevaisuudessa nanokoneistusta voidaan käyttää enemmän jatkuvan tutkimustyön johdosta (Rajurkar, Levy et al. 2006, s. 654-657).

Kilpailu on tarkkuusvalmistuksen kohdalla väistämätöntä, joten valmistajan pitää täyttää seuraavat asiat kilpailukykyensä turvaamiseksi ja parantamiseksi: Automaattisessa kokoonpanossa poistetaan sovittaminen ja siirrytään kerralla onnistuvaan kokoonpanoon. Komponenttien vaihtokelpoisuuden tulee olla korkealla tasolla. Laadunvalvontaa pitää parantaa jatkuvasti tarkemman koneistuksen kautta ja samalla vähentää perinteistä tarkkailua, joka sopii paremmin epätarkemmalle valmistukselle. Valmistettavilla komponenteilla tulee olla pidempi väsymis- ja kulumisaika kuin aikaisemmin. Lisäksi kappaleiden pakkaamista ja pienoiskoossa rakentamista kannattaa kehittää. Kaiken kaikkiaan tärkeimpänä asiana on se, että edellä mainittujen asioiden toteuttaminen ei vaadi pitkää odottelua tai valmistusjärjestelmän massiivista uudelleensuunnittelua. (Dornfeld, Lee 2008, s. 7-8)

Tarkkuusvalmistuksessa keskitytään tässä työssä osa-alueeseen tarkka koneistus, sillä työssä käytettävä esimerkkitapaus liittyy kyseiselle tarkkuusalueelle. Lisäksi työssä käsitellään ensisijaisesti esimerkkitapauksesta johtuen 5-akselista työstöä. Tämän joh-

dosta on syytä määritellä ja esittää tarvittavat tiedot käsitteistä toleranssit, pinnankarheus ja 5-akselinen koneistuskeskus. Käsitteiden määrittely tehdään tarpeeksi laajasti, jotta diplomityössä tarkkuusvalmistukseen liittyvä teoria tulee tutuksi.

3.1 Toleranssit

Tarkkuusvalmistukseen liittyy vahvasti toleranssien ymmärtäminen sillä tarkkuusvalmistuksen avulla koneistetussa kappaleessa on erityisen tärkeää pysyä toleranssirajojen sisällä. Nykyään korkeakouluissa opetetaan toleransseja mutta yleisesti ottaen toleransseihin ei paneuduta välttämättä tarpeeksi syvällisesti tarkkuusvalmistusta silmälläpitäen. Diplomityön eräänä tavoitteena on muistuttaa, ettei toleransseja voi pitää itsestäänselvyytenä ja että niiden oikeaoppinen käyttäminen on ensiarvoisen tärkeää kun kappale halutaan valmistaa tarkasti kerrasta oikein. Toleranssien käyttöön liittyvät rajoitukset ja niiden esittäminen NX:n avulla esitetään tässä kokonaisuudessa.

Kappaleita suunnitellessa ja kokoonpantaessa halutaan olla varmoja siitä, että kaikki saadaan toimimaan kerralla. Tämän johdosta kappale suunnitellaan siten, että sen piirteille asetetaan toleransseja. Toleranssi voidaan määritellä olevan kappaleiden mitoissa olevaa sallittua mittavaihtelua (Pere 2009, s. 20-1). Kappaleen valmistusvaiheessa ei ole kannattavaa, tai joskus edes mahdollista, koneistaa täysin täsmäävää dimensiota. Tämän johdosta käytetään toleransseja määräämään sallitut heitot. Toleranssien suuruuksia tarkkailtaessa voidaan todeta, että mitä pienempi toleranssin arvo on, sitä tarkempi-laatuinen kappaleenpinnan tarvitsee olla. Pituusmitoissa toleransseja on yksi- ja kaksipuolisia. Yksipuolinen toleranssi, esimerkiksi $1.00^{+0.05}_0$, asettaa sallitun heiton mitan ylä- tai alapuolella kun taas kaksipuolinen toleranssi, esimerkiksi 1.00 ± 0.05 , määrää heiton molemmiin puolin mittaa. (Chang, Wysk et al. 1991, s. 19-20)

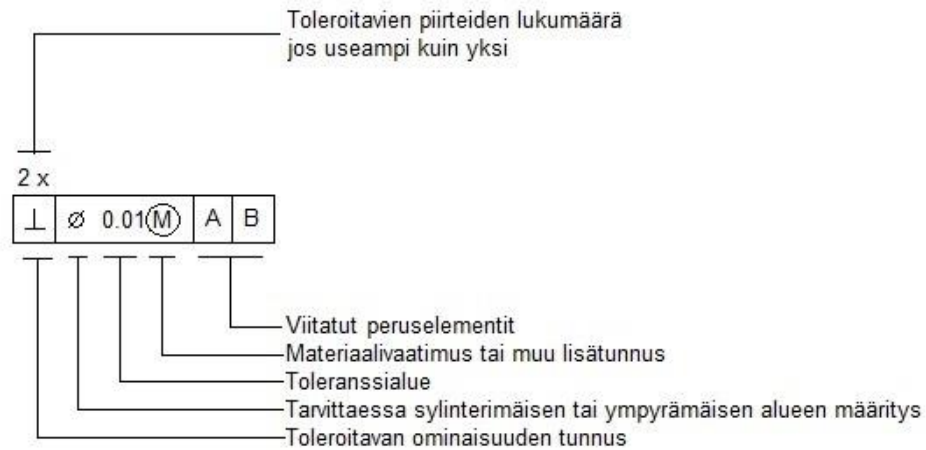
Valmistuksen ja laadun parissa työskentelevillä insinööreillä tulee jatkuvasti vastaan kappaleiden mitoitettuja piirustuksia ja malleja, joissa on kerrottu myös toleransseista. Kaksi suurinta ongelmaa teknisissä piirustuksissa kerrotaan olevan perusteellisuus ja yksiselitteisyys. Tulkintavirheet voivat jopa tuplata valmistus- ja tutkimiskustannukset, joten toleranssien oikeaoppisella käyttämisellä voidaan säästää aikaa, kustannuksia ja esittää tieto helposti tulkittavalla tavalla. Geometrinen mitoitus ja tolerointi (englanniksi Geometric Dimensioning and Tolerancing, GD&T) on merkintätapa, joka lisää merkittävästi yksiselitteisyyttä ja yhtäläisyyttä piirustuksissa. Sitä voidaan pitää yhteisenä kielenä insinöörien välillä, jotka työskentelevät tuotteen, siihen liittyvän prosessin ja laadun parissa. Kolme suurinta hyötyä GD&T:n käyttämisellä ovat toleranssien kasautumaton esit-

täminen, selkeästi määritelty tarkasteluprosessi ja bonustoleranssin olemassaolo materiaaliehtojen myötä. Maailmanlaajuisella tasolla on olemassa monia eri standardeja toleransseihin liittyen ja eroavaisuuksia löytyy, mutta seuraavaksi käydään läpi yleisesti käytössä olevat merkinnät. (Chiabert, Lombardi et al. 1998, s. 29-33)

GD&T on kieli, jota voi lukea lineaarisella logiikalla. Piirustuksen lukemisen alussa on syytä löytää alkukohta, joka on yleensä ensimmäinen peruselementti. Seuraavana kannattaa etsiä toinen ja tarvittaessa myös loput peruselementit, jotta kappaleen perusolemus selviää. Tämän jälkeen lukijalla tulisi olla tietämys erilaisten toleranssityyppien merkkaukstavoista ja niiden symboleista. Suurin osa piirustuksen lukemisesta on siis yksittäisten piirteiden ja toleranssien tunnistamista. Lukija voi myös päätellä piirustuksen eri piirteiden järjestystä tutkimalla, moneenko peruselementtiin niissä viitataan. Jos piirteessä viitataan vain ensimmäisenä määriteltyyn peruselementtiin, piirre liittyy todennäköisesti kappaleen suunnittelun alkuvaiheeseen.

Pituusmittojen toleranssit merkitään mitan viereen, mutta geometristen toleranssien kanssa merkintä tehdään mahdollisimman lähelle sitä kuvaavaa mitta. Merkintätapana käytetään toleranssikehystä (englanniksi Feature Control Frame), joka on jaoteltu siten, että sen sisällä ja vieressä on kaikki tarvittava toleranssitiieto mittaan liittyen. Toleranssikehyksen vasemmassa reunassa on piirteiden lukumäärä ja toleroitavan ominaisuuden tunnus. Keskiosassa puolestaan määritellään tarvittaessa alueen muoto ja mitan lisäehto. Toleranssialue on aina esitetty keskiosassa. Oikeassa reunassa on puolestaan

tarvittaessa esitetty viitattavien peruselementtien tunnuksia. Kuva 8 havainnollistaa toleranssien merkitsemistapaa ja sen sisältämiä osia. (Casas Fernández, Sordo et al. 2006, s. 7)



Kuva 8. Toleranssien merkitseminen piirustukseen, mukailen lähdettä (Casas Fernández, Sordo et al. 2006, s. 7)

Ylläolevassa kuvassa huomataan oikeassa reunassa merkattuna **peruselementit**, joihin toleranssin määrittelyssä voidaan viitata. Peruselementit (englanniksi Datum features) ovat kappaleen pintoja, joiden avulla rakennetaan kuvitteellisia keskiviivoja tai tasoja. Näiden avulla mitataan kulmaa tai sijaintia. Peruselementit voidaan valita monista syistä, kuten esimerkiksi valmistuksen optimoimisen perusteella. Ensisijaisen peruselementin pitää rajata kappaleen muotoa tarvittavalla tarkkuudella ja olla pinta, joka on kosketuksessa muihin kappaleisiin kokoonpanovaiheessa. Tämä varmistaa kappaleen sopivuuden sitä kokoonpantaessa. Toissijaiset peruselementit valitaan siten, että ne määrittävät mittauspinnan piirteelle, joka liitetään tai on samansuuntainen kokoonpanossa olevan vastineen kanssa. Hyvinä toissijaisina peruselementteinä pidetään tiukan toleranssin omaavia muotoja. Toissijaisten peruselementtien pitää aina viitata ensisijaiseen peruselementtiin ja nyrkisääntönä toleranssin arvolle on kolminkertainen arvo ensisijaiseen peruselementtiin nähden. (Meadows 2009, s. 23-27)

Ennen peruselementtien merkitsemistä toleranssikehyksessä on lisätunnus, joka ilmoittaa tarvittaessa käytettäviä tarkennuksia tolerointiin. Kaikista yleisimmin käytössä olevia

lisätunnuksia ovat vähimmäismateriaalin vaatimus, maksimimateriaalin vaatimus ja siirretty toleranssialue. Kuva 9 esittää yleisimpien lisätunnuksien symbolit. (Chang, Wysk et al. 2006, s. 57)

- Ⓛ Vähimmäismateriaalin vaatimus
- Ⓜ Maksimimateriaalin vaatimus
- Ⓟ Siirretty toleranssialue

Kuva 9. Yleisimmät lisätunnukset, mukailten lähdettä (Chang, Wysk et al. 2006, s. 57)

Vähimmäismateriaalin vaatimus (englanniksi Minimum Material Condition, LMC) tarkoittaa olosuhdetta, jossa kappaleessa oleva piirre sisältää rajoissa sallitun minimimäärän materiaalia. Reikien tapauksessa käytetään maksimihalkaisijaa ja akseleiden kanssa minimihalkaisijaa. Kappale, jonka olosuhteena on LMC, on kevyin mahdollinen ja siinä on tarkoituksena estää paksumpien seinien materiaalin vähentäminen. LMC:n eräänä tavoitteena on siis kappaleen lujuuden lisääminen. LMC:tä käytetään esimerkiksi valamiseen menevien kappaleiden piirustuksissa sallien tarpeeksi suuren materiaalmäärän siivoamisen koneistuksella (Meadows 2009, s. 48)

Maksimimateriaalin vaatimus (englanniksi Maximum Material Condition, MMC) on tilanne, jossa kappaleen piirre sisältää maksimimäärän materiaalia rajoitteiden salliessa. Esimerkiksi reikien kanssa halkaisija on mahdollisimman pieni ja akselin tapauksessa puolestaan mahdollisimman suuri. MMC-tilassa oleva piirre on siis painavin mahdollinen tapaus ja se sallii epäyhtenäisen ilma alueen liitettävien piirteiden kesken. MMC luo virtuaalisia olosuhteita, jotka voidaan esittää kiinteänä kokona esimerkiksi rei'issä ja urien syvyyksissä. (Pere 2009, s. 20-118 - 20-119)

Virtuaalinen olosuhde (englanniksi Virtual Condition, VC) tarkoittaa huonointa mahdollista liittämisolosuhdetta, jossa ollaan asetetun toleranssin ylä- tai alarajalla riippuen materiaaliolosuhteesta ja piirteestä. MMC:n tapauksessa käytetään akseleissa toleranssin ylärajaa ja rei'issä toleranssin alarajaa. LMC:n kohdalla toleranssit menevät puolestaan toisin päin. Virtuaalisen olosuhteen avulla varmistetaan, että kappale sopii kokoonpanoon myös huonoimmassa mahdollisessa tapauksessa. (Meadows 2009, s. 9)

Siirrettyä toleranssialuetta käytetään kokoonpanoissa, joissa kaksi kappaletta eivät ole kosketuksissa toisiinsa (Chang, Wysk et al. 2006, s. 58). Reiän tai kierteen kulma-

asentoa voidaan tutkia pulttiin tai ruuviin nähden, joihin se on tarkoitus liittää kokoonpanovaiheessa. Toleranssialue siirretään liitettävän pinnan yläpuolelle siten, että joissain tapauksissa liitettävän osan maksimipaksuus on tarkasteltavana sovitemittana. Jos käytetään toleranssimitoitusta riippumatta piirteen koosta, tarkkuus pysyy vakiona. Mitoitettaessa siirrettyä toleranssialuetta tarkastelun kohteena on ainoastaan kaksi kappaletta. (Rufe 2013, s. 152)

Riippumatta piirteen koosta (englanniksi Regardless of Feature Size, RFS) tarkoittaa tilannetta, jossa geometrinen toleranssi pysyy aina samana, jos sitä ei erikseen mainita. RFS:n avulla pyritään säilyttämään yhtäläisyyttä ja sen tyypillisiä käyttökohteita ovat pyörivät osat ja tiivisteisiin kosketuksissa olevat pinnat (Meadows 2009, s. 47). Rufe (2003, s. 152) kirjoittaa, ettei kaikissa standardeissa ole erillistä symbolia RFS:lle ja että se on oletettu tieto. Ainoastaan LMC ja MMC voi muuttaa piirteen toleranssin suuruutta. RFS:n selvittäminen auttaa piirustuksen yleistarkkuuden hahmottamisessa ja nopeuttaa piirustuksen tulkittamista, sillä RFS:n kautta merkintöjä ei ole niin paljon.

Teoreettisesti tarkat mitat (englanniksi Theoretically Exact Dimensions, TED) ovat eräs huomionarvoinen osa kappaleen mitoitus- ja tolerointia. Elementtejä ja elementtiryhmiä, joiden määrittely on tehty sijainnin, suunnan tai muodon mukaan voidaan antaa teoreettisesti noudatettavia arvoja. Näitä mittoja kutsutaan nimellä TED. TED-mittoja ei itsessään toleroida ja niiden merkintätapa eroaa tavallisista mitoista siten, että TED-mittojen ympärillä on kehys. Tyypillinen käyttötilanne on vaatimukset, jotka ovat riippuvaisia toisistaan. Toisin sanottuna vaatimukset, joita voidaan pitää yhtenä toleranssialueena. TED-mittoja ei siis merkata toleranssikehyksiin vaan ne ovat mittoja, joihin toleranssien merkintäkehyksessä viitataan. (SFS-EN ISO 1101 2017, s. 51-52)

3.1.1 Muototoleranssit

Muototoleransseja ovat suoruuksuus, tasomaisuus, lieriömäisyys ja ympyrämäisyys. Muototoleransseille yhteistä on se, ettei niiden yhteydessä saa viitata peruselementteihin. Pin-

nan tai tason poikkeamaa geometrisesti ideaalista muodosta kutsutaan muototoleranssiksi. Kuva 10 esittää kirjallisuudesta löytyvät muototoleranssit ja niistä käytetyt merkin-
nät. (Narayana 2006)



Kuva 10. Muototoleranssit, mukailen lähdettä (Pere 2009, s. 20-82)

Suoruutta voidaan käyttää kuvaamaan pinnan viivaelementtejä tai halkaisijan lävistäjää viitattuna keskiviivan suoruuteen. Pinnan rajaamisessa se asettaa jokaisen viivaelemen-
tin olevaksi toleranssialueen sisällä. Täydellisesti suora pinta tarkoittaa pintaa, jolle on vedetty suora viiva, jonka jokainen piste on kosketuksissa tarkasteltavan pinnan kanssa. Toleranssialue koostuu kahdesta yhdensuuntaisesta viivasta, jotka ovat toleranssialue-
een suuruusella etäisyydellä toisistaan. Maksimi- ja vähimmäismateriaalin vaatimusta on sallittua käyttää silloin, kun suoruus ei kohdistu pinnan rajaamiseen ja se on määritellyn virtuaalisen olosuhteen täyttävä suure. (Rufe 2013, s. 155-156)

Tasomaisuus muodostaa toleranssialueen, jota rajaa kaksi yhdensuuntaista tasoa. Ta-
somaisuuden vaatimuksen täyttymiseksi toleranssialueen pinnan pitää olla kokonaisuus-
dessaan rajaavien tasojen välissä. Tasomaisella pinnalla ei ole halkaisijaa, sillä piirre
kuvaa nimenomaan tasoa, eikä esimerkiksi lävistäjää. Tämä tarkoittaa myös sitä, ettei
tasomaisuudella ole kokoa, maksimi-, eikä vähimmäismateriaalin vaatimusta. Ainoat
asiat pinnan tasomaisuuden määrittelyssä ovat siis tasomaisuuden geometrinen tunnus
ja geometrinen toleranssi. Keskiviivan määrittelyssä puolestaan LMC ja MMC on sallit-
tuja. Tyypillisiä sovelluskohteita tasomaisuudelle ovat tiivisteiden pinnat, liitettävät pinnat
ja välillä myös peruselementeissä käytettävät pinnat. (Rennels 2003, s. 566)

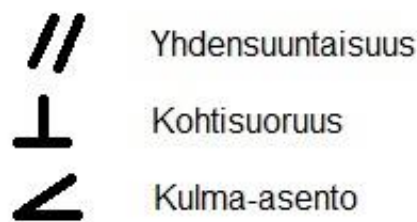
Lieriömäisyys on tasomaisuuden tapaan kolmidimensioinen muotokontrolli ja tolerans-
sialueilla on paljon yhteistä. Lieriömäisyyden toleranssialueen muodostaa kaksi sama-
keskistä lieriötä, jotka ovat toleranssin suuruusella etäisyydellä toisistaan. Ympyrämäi-
syyden, suoruuden ja kaventumisen kolmidimensioista muodonhallintaa voidaan kutsua
lieriömäisyydeksi. Lieriömäisyys kontrolloi pintaa, joten sillä ei ole kokoa, eikä LMC ja

MMC ole sallittuja. Lieriömäisyyttä tutkiessa voidaan mitata heiton, suoruuden ja halkaisijamitan arvoja. Mittaukset tulee tehdä useista kohdista lieriömäistä pintaa tarkan kuvan saamiseksi (Meadows 2009, s. 109-111)

Ympyrämäisyys kuvaa poikkileikkauksen muotoa ja sen toleranssialueena on kaksi samakeskistä ympyrää, jotka ovat toleranssialueen suuruisella etäisyydellä toisistaan (Casas Fernández, Sordo et al. 2006, s. 12). Meadows (2009, s. 113-115) puolestaan kertoo kappaleen ympyrämäisyyttä tutkiessa ilman mittaussuunnitelmaa seuraavaa: Mittaajan on otettava tarkasteluun tarpeeksi monta poikkileikkausta, jotta hän voi olla vakuuttunut kappaleen ympyrämäisyydestä. Jo yhden poikkileikkauksen kohdalla toleranssialueen ylittäminen tarkoittaa sitä, ettei tutkittava kappale täytä ympyrämäisyyden asettamia rajoja. Ympyrämäisyyden tutkimisessa epävarmuustekijöitä ovat muun muassa tutkittavien poikkileikkausten määrä ja jokaisen ympyrän mittauspisteiden lukumäärä. LMC ja MMC pitäisi erikseen vahvistaa jokaiselle tutkittavalle poikkileikkaukselle, joten niitä ei ole järkevää käyttää ympyrämäisyyden yhteydessä.

3.1.2 Suuntatoleranssit

Suuntatoleransseja ovat yhdensuuntaisuus, kohtisuoruus ja kulma-asento. Niille yhteistä on vaadittu peruselementtiin viittaaminen sekä sallittu LMC ja MMC symbolien käyttö. Periaatteessa kaikki kolme rajaavat kulma-asentoa, mutta yhdensuuntaisuudessa kulma-asento on 0 astetta ja kohtisuoruudessa 90 astetta. Eri symbolien käytöstä on lukijan kannalta kuitenkin se hyöty, että tiedetään kuinka suuren kulman arvoa rajataan (Rufe 2013, s. 157). Kuva 11 esittää suuntatoleranssit ja niitä kuvaavat symbolit.



Kuva 11. Suuntatoleranssit, mukailen lähdettä (Narayana 2006, s. 234)

Yhdensuuntaisuutta voidaan käyttää sekä pinnoissa että lävistäjissä. Pinnoissa yhdensuuntaisuus kontrolloi kulman lisäksi myös alaluvussa aikaisemmin määritellyä tasomaisuutta. Toleranssialueen muodostaa kaksi yhdensuuntaista viivaa tai tasoa, jotka ovat toleranssialueen suuruisella etäisyydellä toisistaan. Toleroidun pinnan jokaisen pisteen

on oltava määritellyn toleranssialueen sisällä, jotta pinta täyttää yhdensuuntaisuuden vaatimuksen. Yhdensuuntaisuus kontrolloi tasomaisuutta siltä osin kun asetettu toleranssi näyttää, mutta se vaatii tutkittavan tason olevan määrätyn kokoinen, jotta tarkastelu olisi mahdollista toteuttaa tasomaisuudelle (Rufe 2013, s. 157)

Kohtisuoruudessa on sama periaate kuin yhdensuuntaisuudessa, mutta referenssikulmana on 90 astetta. Tarkasteltavana suureena kohtisuoruudessa on pinnan, keskiviivan tai akselin oikeansuuruinen kulma verrattavaan peruselementtiin tai toiseen akseliin. Toleranssialueen muodostavatkin kaksi kohtisuoraan referenssitasoon nähden olevaa viivaa tai tasoa, jotka ovat tietyllä etäisyydellä toisistaan. Kohtisuoruudessa viitataan usein useampaan peruselementtiin. (Chang, Wusk et al. 2006, s. 66)

Kulma-asennossa voidaan pintojen ja viivojen lisäksi kontrolloida keskitasoa. Asteluvut, joihin kulma-asento keskittyy poikkeavat suurimmaksi osaksi yhdensuuntaisuudesta ja kohtisuoruudesta, mutta koska kulma-asentoa voidaan käyttää yleisenä suuntatoleranssin symbolina, sitä voidaan myös käyttää asteluvuissa nolla ja 90. Toleranssialueen muodostaa siis kaksi yhdensuuntaista tasoa tai viivaa, jotka ovat tietyssä kulmassa peruselementtiin tai referenssiviivaan nähden ja toleranssialueen suuruisella etäisyydellä toisistaan. Kulma-asentoa käytetään usein muun muassa yhden pinnan suuntaamisessa ja reikien tai akselien paikan parantamisessa. Lisäksi useampaan peruselementtiin viittaaminen on yleistä kulma-asentoa tarkasteltaessa (Meadows 2009, s. 136-139)

3.1.3 Sijaintitoleranssit

Sijaintitoleransseja ovat paikka, samakeskisyys, sama-akselisuus ja symmetrisyys. Peruselementtiä voidaan käyttää jokaisessa sijaintitoleranssityypissä mutta paikkatoleranssi on ainoa, jossa peruselementin tarve ei ole välttämätöntä. Samakeskisyydellä ja

-akselisuudella on sama symboli, mutta niitä käytetään erilaisissa tapauksissa, jotka tarkennetaan alla olevan kuvan jälkeen. Kuva 12 esittää sijaintitoleranssit ja niitä kuvaavat symbolit. (Pere 2009, s. 20-82)



Kuva 12. Sijaintitoleranssit, mukailten lähdettä (Pere 2009, s. 20-82)

Paikkatoleranssi on yksi arvokkaimmista ja monimuotoisimmista toleranssityypeistä geometriseen määrittelyyn ja sen vuoksi se on eniten käytetty toleranssityyppi. LMC, RFS ja erityisesti MMC hyödyttävät paikkatoleranssin sallittujen toleranssialueiden määrittelyssä. Toleranssialueena voidaan pitää absoluuttisen sijainnin poikkeamaa, joka on toleranssialueen suuruinen. Virtuaalisen olosuhteen tarkastelun huomioon ottaminen on tärkeää erityisesti MMC:n hyödyntämisessä sillä sijaintitoleranssi ei saa ylittää virtuaalisen olosuhteen rajapintoja. (Rufe 2013, s. 160-161)

Samakeskisyyys kontrolloi pyörähdyssymmetrisien kappaleiden mediaanipisteitä ja samaan keskipisteeseen sijoitettujen piirteiden yhteneväisyyttä. Täydellisessä samakeskisyydessä kaikkien tutkittavien keskipisteiden sijainti on sama. **Sama-akselisuudessa** puolestaan tutkitaan kappaleiden akseleiden yhteneväisyyksiä. Muodostettavien toleranssialueiden tyyppi on joko lieriömäinen ja sama-akselinen referenssiakselin kanssa tai pallomainen ja sama-akselinen peruselementin keskipisteen kanssa. Yhtä tai useampaa peruselementtiä pitää käyttää samakeskisyyden ja sama-akselisuuden määrittelyssä. Materiaaliolosuhteita LMC ja MMC ei saa puolestaan käyttää tarkastelussa. Toleranssialueen muodostaa pisteet tai akselit, jotka ovat toleranssialueen suuruisella etäi-

syydellä määritellystä sijainnista. Samakeskisyyden avulla kontrolloidut kappaleet voi hylätä vain silloin kun tiedetään tarkasti kontrolloidun piirteen ja referenssipisteen välinen relaatio. Tämä voidaan varmistaa erilaisten mittausten avulla. (Meadows 2009, s. 208)

Symmetrisyys on periaatteessa samanlainen työkalu kuin samakeskisyyys, mutta sitä käytetään eri tyyppisiin piirteisiin. Symmetrisyydessä kahden tai useamman piirteen mitauspisteet ovat kaikki samalla peruselementin akselilla tai keskitasolla. Useimmiten symmetrisyyttä käytetään leveyden keskipisteen ja peruselementin keskitason yhteydessä. Toleranssialueeksi on määritelty toleranssin suuruinen alue peruselementin keskitason ympärillä. Materiaaliolosuhteita ja halkaisijan merkintää ei ole sallittua käyttää mutta peruselementtiin pitää viitata. Tutkittavan pinnan poikkileikkauksissa on eroavaisuuksia kuoppien ja kohoumien muodossa, mutta poikkileikkausten keskipisteet muodostavat pistejoukon, jonka jokaisen pisteen pitää täyttää toleranssiehto, jotta kappale voidaan hyväksyä (Meadows 2009, s. 215)

3.1.4 Heittotoleranssit

Heittotoleransseja pidetään yhdistelmätoleransseina, jotka kontrolloivat funktionaalisia suhteita peruselementin akselin ja yhden tai useamman piirteen välillä. Tyypillisiä heittotoleranssin tarkastelukohteita ovat peruselementtien ympärille kootut tai peruselementteihin sopivassa kulma-asennossa olevat piirteet. Tämän johdosta onkin luonnollista, että peruselementteihin pitää viitata heittotoleransseja käytettäessä. Kuva 13 esittää heittotoleranssien nimet ja niistä käytettävät merkinnät. (Chang, Wysk et al. 2006, s. 72)



Kuva 13. Heittotoleranssit, mukaillen lähdettä (Pere 2009, s. 20-82)

Heittoa (englanniksi Circular Runout Tolerance) käytetään kaksiulotteisessa pinnanhallinnassa. Useimmiten heittoa käytetään halkaisijoiden ympyrämäisyyden ja sama-akselisuuden tutkimisessa. Sitä voidaan käyttää myös tasomaisessa pinnassa kontrolloimaan pinnan ympyrämäistä muotoja ja niiden suhdetta peruselementtiin. Tämän toleranssityypin käytössä pitää huomioida kappaleen kokorajoitukset, jotta tarkastelussa ei tule ristiriitoja suunnitellun kappaleen kanssa. Toleranssin arvo tarkastetaan täyden mittarin liikkeen menetelmällä, jossa tutkitaan halkaisijan muutosten arvoja. Kaksiulotteisuudesta

johtuen poikkileikkauksia pitää tutkia tietty määrä tuloksen luotettavuuden varmistamiseksi. Jos yksikin poikkileikkaus ei täytä toleranssiehtoja, tutkittavan piirteen toleranssiehto ei täyty. Heitossa toleranssialueena on toleranssin etäisyydellä mallinnetusta pinnasta olevat kuvitteelliset ympyrät (Chang, Wysk et al. 2006, s. 72-74)

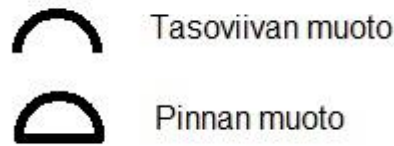
Kokonaisheitto (englanniksi Total Runout Tolerance) on kolmiulotteisessa pinnanhallinnassa käytetty apuväline. Jos kokorajoitukset ovat tiukemmat kuin kokonaisheitto, kontrolloi se ainoastaan halkaisijan samakeskisyyttä. Tässä tilanteessa ympyrämäisyys, suoruus ja kaventuminen rajoitetaan halkaisijan kokotoleranssilla. Jos taas halkaisijan kokotoleranssi on löysempi kuin kokonaisheitto, rajoittajana on kokonaisheitto. Toleranssialueen muodostaa kaksi tutkittavan pinnan ympärillä olevaa yhdensuuntaista muotoa, jotka ovat toleranssin suuruisella etäisyydellä toisistaan. Molemmat heittotoleranssit rajaavat pintaa, jonka johdosta materiaaliolosuhteita LMC ja MMC ei saa käyttää. Kuitenkin, peruselementteihin viittaaminen on vaadittu tieto. Heittotoleransseja mitataan tutkittavan kappaleen pyöriessä akselinsa ympäri (Meadows 2009, s. 195-200)

Määriteltyjä toleranssityyppejä arvioitaessa on syytä miettiä kustannuksia ja luotettavuuksia. Sama-akselisuutta voidaan mitata sijainnin, samakeskisyyden ja heittotoleranssien avulla. Sijainnissa huomioidaan materiaaliolosuhteet geometrisen toleranssien jälkeen, jonka johdosta se on epätarkin, mutta myös halvin vaihtoehto. Samakeskisyydessä tarvitsee tehdä mittauksia hylkäämisen perustana, jonka johdosta se on tarkempi mutta vaikeampi toteuttaa kuin sama-akselisuuden mittaaminen sijainnilla. Heitto on helppo toteuttaa ja se sopii hyvin pyörähdyssymmetrisille piirteille mutta ympyrämäisyyden tarkan vaatimuksen seurauksena kappale on vaikeampi ja kalliimpi valmistaa. Kokonaisheitto on kaikista vaikein ja kallein tapa, mutta tarkastelu on luotettavinta (Meadows 2009, s. 212)

3.1.5 Profiilitoleranssit

Profiilitoleransseja ovat pintaprofiili ja viivaprofiili. Profiilitoleransseissa yhteistä on se, että on sallittua, joskaan ei pakollista, viitata peruselementtiin. Lisäksi LMC ja MMC saa-

daan ottaa huomioon vain koon peruselementin määrittämisessä. Piirteiden tapauksessa LMC ja MMC on kiellettyjä käytettäväksi. Kuva 14 esittää profiilitoleransseista käytetyt nimet ja merkintätavat. (Pere 2009, s. 20-82)



Kuva 14. Profiilitoleranssit

Tasoviivan muoto on kaksiulotteinen suure, joka tarkastelee poikkileikkauksia. Ympyrämäisyyden tapaan tasoviivan muotoa tarkasteltaessa jokaisen poikkileikkauksen tulee olla toleranssialueen sisällä, jotta ehto täyttyy. Tarkasteluun valitaan joko mittaus suunnitelmassa oleva määrä poikkileikkauksia tai tarpeeksi monta poikkileikkausta asian toteutukseen. Toleranssiväli tasoviivan muototoleranssissa voi olla toiseen suuntaan suurempi kuin toiseen, mutta joka tapauksessa toleranssivälin rajaa kaksi samanmuotoista tutkittavalla etäisyydellä olevaa käyrää mitattavasta viivasta. Toleranssityyppi voi myös olla yksisuuntainen, jolloin toinen rajoittava viiva on yhtenevä alkuperäisen viivan kanssa (Chang, Wysk et al. 2006, s. 65)

Pinnan muodossa rajataan mielikuvitusmittojen sijaan oikean piirteen poikkeamia perusmitasta. Perusmittoja käytetään kuvaamaan kokoa, muotoa, kulma-asentoa ja sijaintia. Tasoviivan muototoleranssin tapaan pinnan muodossa toleranssialue voidaan yhtä suurien plus ja miinustoleranssin sijaan määrittää epäsuorasti. Alaraja voi olla lähempänä perusmittaa kuin yläraja ja toisinpäin. Ylä- ja alarajoina on kuitenkin samanmuotoiset pintaelementit, jotka määrittävät toleranssivälin. Peruselementtien käyttö hyödyntää joissain tapauksissa toistettavuutta ja antaa suuntaa kappaleen sopivuudelle kokoonpanovaihetta varten (Meadows 2009, s. 147-151)

3.1.6 Toleranssit eri standardeissa






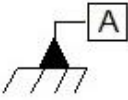
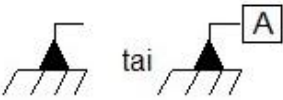
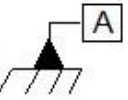





Toleransseja määriteltäessä pyritään noudattamaan tietynlaista merkintätapaa, jotta dokumentointi olisi säännönmukaista ja ymmärrettävää. Maantieteellinen sijainti, yrityksen kansainvälisyys ja käytäntöinä olevat toimintaperiaatteet ovat muun muassa syitä sille, että yrityksillä on käytössään erilaisia standardityyppejä. Yleisiä käytössä olevia standardityyppejä ovat Suomen Standardisointiliitto (SFS), kansainvälinen standardisointiliitto

(ISO), saksalainen standardointi-instituutti (DIN) ja American Society of Mechanical Engineers (ASME). Standardityyppien lyhenteen jälkeisellä numerosarjalla ja nimellä pystyy löytämään tilanteeseen sopivan standardin ja sitä kautta varmistamaan oikeanlaiset merkintätavat. (Casas Fernández, Sordo et al. 2006, s. 375-377)

SFS:n toleransseihin liittyviä toleranssistandardeja tarkasteltaessa voidaan todeta, että ne on nykyään tehty perustuen ISO-standardeihin. Tämän johdosta näiden kahden standardityypin välillä ei ole eroavaisuuksia merkintätavoissa (SFS-EN ISO 1101 2017). Casas Fernández et al. (2006, s. 376) havainnollistaa teoksessaan, että myös uudemmat DIN-standardit toleranssien suhteen on tehty perustuen ISO-standardeihin. Näiden huomioiden seurauksena tarkastellaan siis eroavaisuuksia ISO:n ja ASME:n toleranssistandardien välillä.

Rufe (2013, s. 151) esittää ISO:n ja ASME:n eroavaisuutena symbolien käytössä muun muassa sen, ettei ISO:ssa käytetä symbolista merkintää maksimi- ja vähimmäismateriaalin rajalle, peruselementin kääntymiselle eikä tangenttitalolle. Erilaista merkintätapaa

käytetään myös epätasaisesti suhtautuvasta profiilista sekä peruselementtien merkinnästä. Kuva 15 esittää erilaiset merkintätavat näiden kahden standardityypin välillä.

Symbolin nimi:	ASME	ISO
Maksimimateriaalin raja		Ei ole
Vähimmäismateriaalin raja		Ei ole
Tangenttitaso		Ei ole
Peruselementin kääntyminen		Ei ole
Epätasaisesti suhtautuva profiili		UZ
Peruselementtien merkintä		 tai 
Kontrolloitu säde	CR	Ei ole
Statistinen tolerointi		Ei ole
Mitat		
TED-mitat		

Kuva 15. ASME:n ja ISO:n merkintätapojen eroavaisuuksia, mukailen lähteitä (Rufe 2013, Simmons, Phelps et al. 2012)

Vähimmäismateriaalin raja tarkoittaa toleranssin tai toleranssien asettamaa rajoitusta piirteen materiaalin sisässä. Sitä voidaan käyttää vain peruselementteihin viitattaessa ja näyttää, että peruselementti on peräisin virtuaalisesta olosuhteesta. Maksimimateriaalin rajassa on sama periaate, mutta se tarkastelee piirteen ulkopuolella olevia rajoituksia. Peruselementin kääntyminen tarkoittaa sitä, ettei peruselementin asento ole kiinnitetty piirustuksessa olevaan tietoon, vaan se voi vapaasti kääntyä toleranssialuetta pitkin kytkeytyen näin koko piirteeseen. Tangenttitaso tarkoittaa nimensä mukaisesti toleroitavan pinnan tangenttina olevaa tasoa, mutta se ei ota kantaa muodonhallintaan. Tangenttitaso merkataan lisätunnuksen kohtaan toleranssikehyksessä. Epätasaisesti suhtautuva

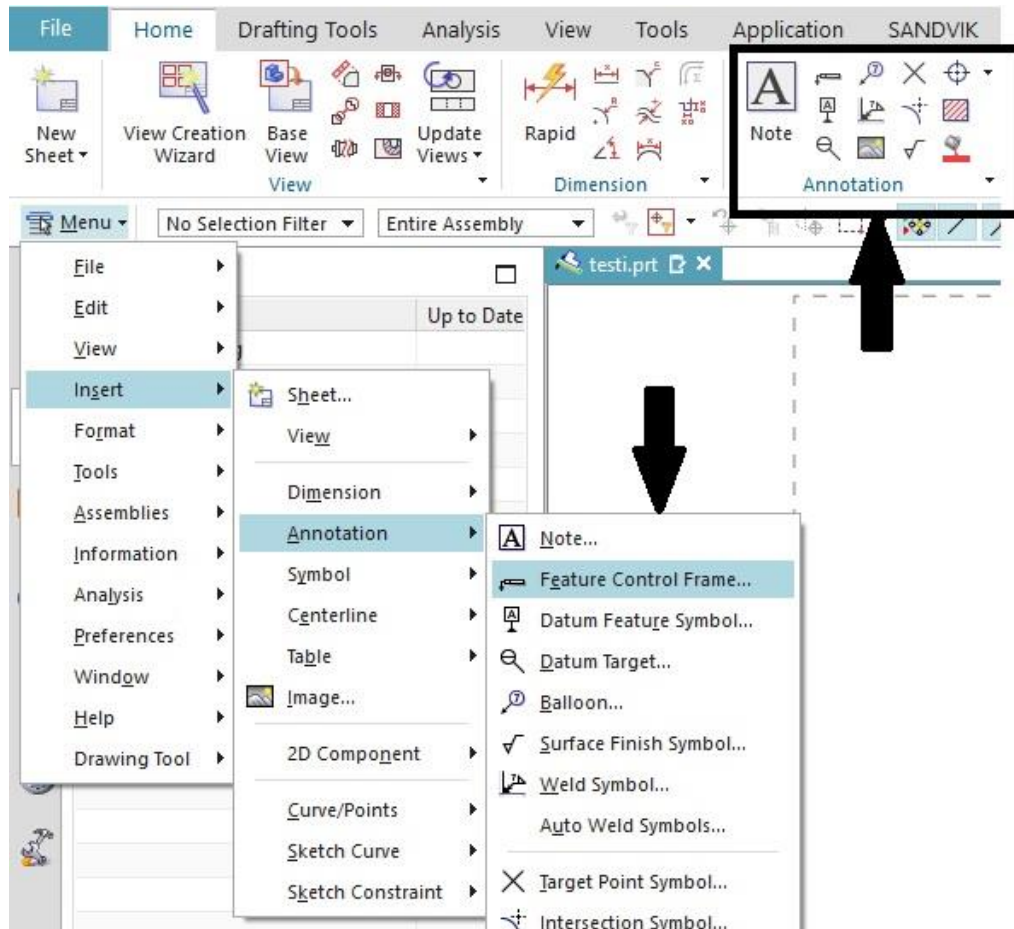
profiili tarkoittaa yksisuuntaista toleranssia tai toleranssiväliä, joka ei ole yhtä suuri tarkastelupinnan molemmilla puolilla. Merkintäpaikka on tangenttitason tapaan toleranssikehyksen lisätunnuksen kohdassa (Rufe 2013)

Simmons et al. (2012, s. 251) esittää määritelmät kontrollisäteelle ja statistiselle toleroinnille. Kontrollisäteellä (englanniksi Controlled Radius, CR) tarkoitetaan toleranssialuetta, joka on määritelty kahden samakeskisen kaaren väliin. Kaaret ovat tangentteja vierekkäisille tasoille. Kappaleen ääriviivan ja puolikuun muotoisen toleranssialueen pitää olla käyrä, jossa ei ole täyskäännöstä ja jonka kaikkien pisteiden on oltava toleranssialueen sisällä. Symboli laitetaan merkittäessä ennen toleroitavan säteen dimensiota. Statistinen tolerointi (englanniksi Statistical Tolerancing, ST) indikoi komponentteihin liittyvät toleranssit kokoonpanoon. Esimerkkinä voi olla kokoonpanon toleranssi, joka saadaan käyttämällä kaavaa. Kaavassa puolestaan käytetään arvoina yksittäisiä toleransseja, jotka havaitaan ST-merkinnästä. Symbolin sijainti toleranssikehyksessä on toleroidun dimension jälkeen.

3.1.7 Toleranssien merkitseminen Siemens NX:ssä

Suunnittelu- ja mallinnusohjelmassa Siemens NX toleranssien merkitsemiseen on kaksi tapaa: Toleranssit voidaan merkitä työpiirustukseen tai suoraan 3D-malliin yrityksen toimintatavasta riippuen. Piirustuksia voidaan luoda NX:n sovelluksella nimeltä Drafting. Kun sovellus on auki ja kappaletta aletaan toleroimaan, on tarvetta lisätä peruselement-

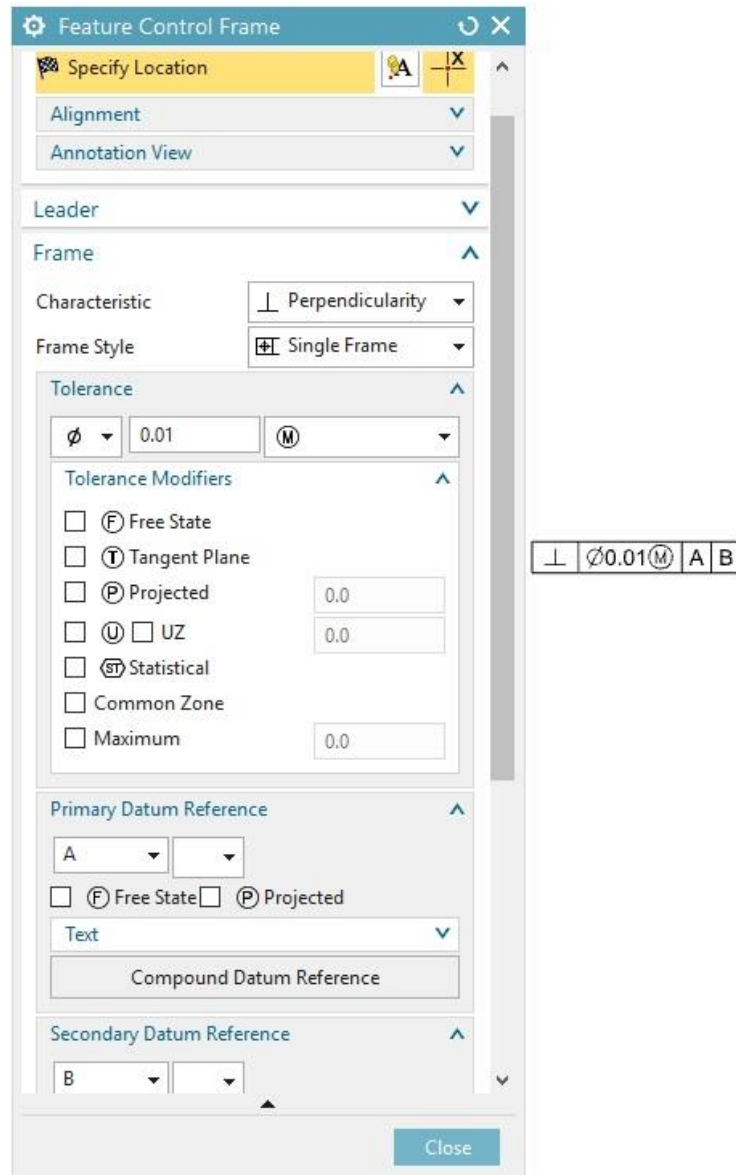
tejä ja toleranssikehyksiä. Lisääminen käy joko valitsemalla Home-välilehden Annotation-osuudesta halutun piirteen tai valitsemalla sen polkua Menu-Insert-Annotation pitkin. Kuva 16 visualisoi kerrotut lisäämistavat NX:ssä. (Siemens 2019)



Kuva 16. Toleranssipiirteen lisääminen NX-piirustukseen

Esitettyssä kuvassa on toleransseihin liittyen nähtävissä peruselementti (Datum Feature Symbol) ja toleranssikehyks (Feature Control Frame). Peruselementtiä lisättäessä tarvitsee vain määrittellä sijainti, liitettävä pinta ja symbolina oleva kirjain. Tästä prosessista ei

ole tarvetta näyttää visuaalista selvennystä. Toleranssikehyksen liittämässä puolestaan on enemmän huomioitavia kohtia ja niitä havainnollistaa Kuva 17.

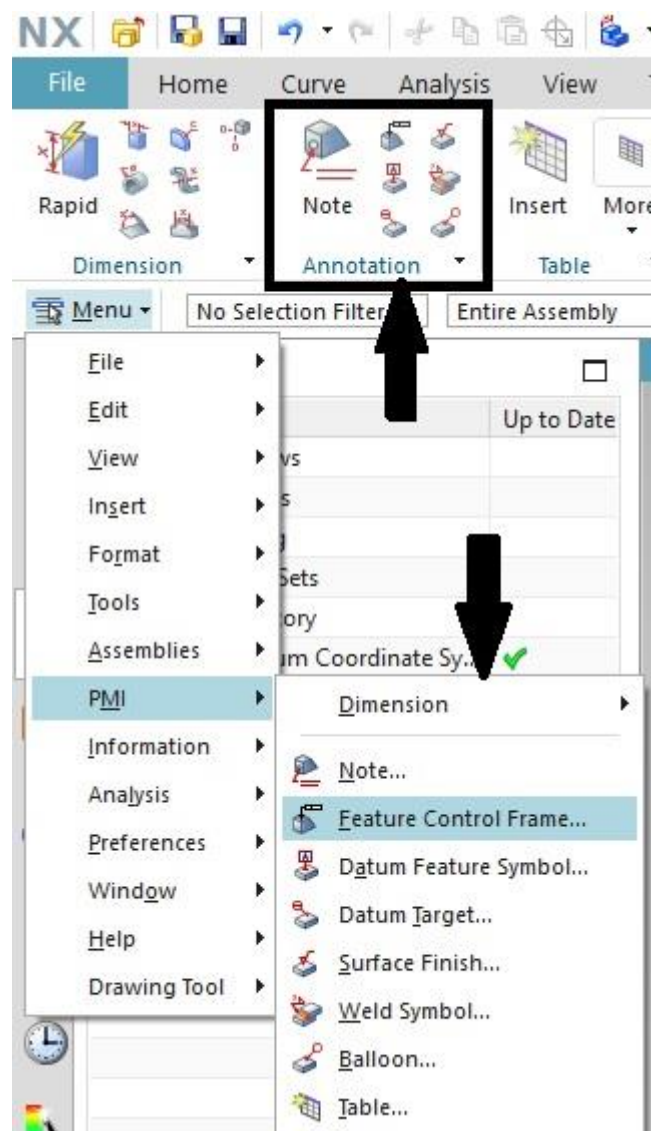


Kuva 17. Toleranssikehyksen liittäminen NX-piirustukseen

Sijainnin ja viitattavan pinnan jälkeen määritetään toleranssityyppi, joihin kuuluvat aikaisemmin esitetyt muoto-, suunta-, sijainti-, heitto- ja profiilitoleranssit. Kuvassa valittuna on kohtisuoruus. Seuraavaksi valitaan toleranssikehyksen tyyppi, jonka jälkeen päästään lisäämään toleranssiin kohdistuvia ehtoja. Toleranssikehyksen keskiosaan valitaan toleroitavan suureen tyyppi, toleranssiarvo, ja mahdollisesti käytettävät menetelmät. Kuvassa valittuna halkaisija ja maksimimateriaalin vaatimus. Toleranssikehyksen oikeaan

reunaan valitaan viitattavat peruselementit ja niihin liittyvät lisähuomiot. Kuvassa viitataan peruselementteihin A ja B. Määritelty toleranssikehys näkyy ylläolevan kuvan oikeassa reunassa.

Toleranssiominaisuuksien lisäämisellä NX:n 3D-malliin voidaan lähestyä tilannetta, jossa voidaan kokonaan luopua työpiirustusten käytöstä. Lisäksi 3D-mallissa olevan toleranssin kautta voidaan mahdollistaa mallin monipuolisempi käyttö esimerkiksi koordinaattimittauksien parissa. Toleranssiominaisuuksien lisääminen käy joko valitsemalla PMI-välilehden Annotation-osuudesta halutun piirteen tai valitsemalla sen polkua Menu-PMI pitkin. Kuva 18 visualisoi kerrotut lisäämistavat NX:ssä. (Siemens 2019)



Kuva 18. Toleranssipiirteen lisääminen NX-malliin

Peruselementtien ja toleranssikehysten määrittelemiseen käy tästä eteenpäin samalla tavalla kuin 2D-piirustusten tapauksessa. Ainoana eroavaisuutena on pinnan valitseminen,

joka onnistuu kappaletta kääntelemällä sen sijaan että valitsisi piirustuksessa olevan parhaan näkymän ominaisuuden lisäämiselle. Siemens NX:n avulla voidaan lisätä toleranssipiirteitä onnistuneesti riippumatta käytettävästä standardista, sillä ohjelmasta löytyy kaikkien standardityyppien vaatimat ominaisuudet esittämistapaan. (Siemens 2019)

3.2 Pinnankarheus

Kappaletta suunniteltaessa ja virtuaalista mallia luodessa käytetään ideaalisia geometrisia pintoja, sillä niiden avulla on helpointa luoda visuaalinen esitys lopputuloksena olevasta kappaleesta. Valmistustekniikat eivät pysty tuottamaan ideaalista pintaa, vaan niiden tuotoksina on makro- ja mikrogeometrisia poikkeamia. Makrogeometrisia poikkeamia kutsutaan muotovirheiksi ja mikrogeometriset poikkeamat ovat osana pinnankarheutta. Pinnankarheus tarkoittaa pienijakoisia pinnan epätasaisuuksia, jotka sisältävät yleensä epäsäännöllisyyksiä. Epätasaisuudet ovat tuloksena käytetyistä valmistusmenetelmistä tai muista vaikutuksista ja nämä epätasaisuudet otetaan huomioon suunnitteluvaiheessa asetettujen rajojen puitteissa. Pinnankarheuksia optimoitaessa tulee useimmiten mieleen mahdollisimman sileä pinta, mutta esimerkiksi metalliruiskutettavat osat vaativat onnistuakseen epätasaisista pintaa. Pinnankarheutta tarkasteltaessa käytetään seuraavaa logiikkaa: Mitä pienempi lukuarvo pinnankarheudella on, sitä tasaisempi on kappaleen pinta. (Pere 2009, s. 21-1 - 21-2)

Pinnankarheuteen vaikuttava koneenrakennuksen kehitys kasvattaa jatkuvasti tarvetta määritellä pinnat entistä yksityiskohtaisemmin. Aikaisempaan verrattuna eroavaisuuksia voidaan havaita koneistusvaiheessa seuraavissa asioissa: Suuremmat nopeudet ja kiihtyvyydet, suuremmat kierrosluvut, vähäisemmät huoltotyöt jne. Pinnankarheuden tarkentumisen seurauksena kappaleiden väsymiskestävyys ja korroosion vastustuskyky ovat parantuneet. Lisäksi parempi pinnanlaatu vaikuttaa liitettävien kappaleiden väliseen kit-

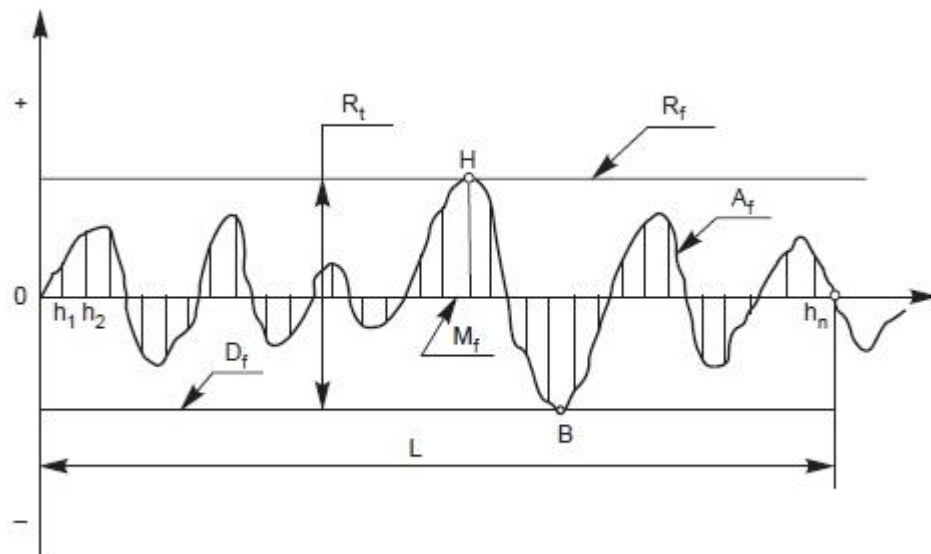
kaan vähentäen sitä (Narayana 2006, s. 242). Erilaisien työstötapojen soveltumisaluista pinnanlaatua silmälläpitäen on kirjallisuudessa monia eri versioita, mutta Kuva 19 esittää suuntaa-antavan käsityksen työstötavoista ja niiden tarkkuusalueista.

VALMISTUSMENETELMÄ	PINNANKARHEUS (R_a) μm													
	0,006	0,012	0,025	0,05	0,1	0,2	0,4	0,8	1,6	3,2	6,3	12,5	25	50
Hiekkavalu Kuorimuottivalu Kokilivalu Painevalu Tarkkuusvalu														
Metallin ruiskupuristus Muovin ruiskupuristus														
Tarkkuustaonta Muottitaonta														
Pursotus Muotovalssaus Silovalssaus														
Levyn syväveto Vetäminen														
Pituussorvaus Tasosorvaus Pistosorvaus														
Höyläys Pistäminen Kaavinta														
Poraus Avarrus Upotus Kalvinta														
Lieriöjyrsintä Otsajyrsintä Avennus Viilaus														
Pyöröhiointa Tasohiointa														
Kiillotushionta Elektrolyyttinen kiillotus Kipinätyöstö														
Laahinta Hiveltäminen Hiertäminen Kiillotushiertäminen														
Suihkutus Rummutus														
Polttoleikkaus Sahaus Leikkaus														

Kuva 19. Eri valmistusmenetelmillä saatavia pinnankarheuksia, perustuu lähteeseen (Pere 2009, s. 21-16)

Kuvasta voidaan huomata, että eri työstötavat soveltuvat eri tarkkuuksia vaativiin tilanteisiin ja niiden soveltuvuusalueiden laajuus vaihtelee keskenään. Pohanish (2016, s. 278) esittää vastaavantyyppisen kuvan, mutta siinä on jaoteltu jokainen työstötapa tyy-

pillisen ja epätyypillisen pinnankarheuden alueelle. Tietyntylaisia pinnankarheuksia voidaan saavuttaa myös sellaisella valmistusmenetelmällä, jotka eivät ole tarkoitettu kyseiseen tarkkuuteen, mutta johon voidaan päästä olosuhteita ja työstötavan parametreja muokkaamalla. Tämän johdosta onkin tärkeää valita soveltuvimmat menetelmät tapauskohtaisesti, jotta valmistusprosessista saadaan mahdollisimman tehokas. Ennen kuin tiedetään minkä suuruinen on haluttu pinnankarheus, on syytä selvittää, mistä osista pinnankarheus koostuu ja miten se määritetään. Kuva 20 esittää pinnankarheuden laskemisessa käytettäviä suureita.



Kuva 20. Pinnankarheuden määrittämisen suureet, perustuen lähteeseen (Narayana 2006, s. 242)

Keskiviivajärjestelmä käyttää pinnankarheuden määrittämisessä perusviivana kappaleen keskiviivaa (Mean Profile, M_f). Tämän pohjalta voidaan muodostaa ylläolevan kuvan horisontaalinen akseli. Vertikaalinen akseli puolestaan kertoo pinnan poikkeaman keskiviivasta. Poikkeamilla on huippuja ja laaksoja riippuen siitä, kumpaan suuntaan poikkeama on. Huippu tarkoittaa keskiviivasta ulospäin, eli ympäröivää ainetta kohti, suuntautunutta osaa, joka rajoittuu keskiviivan ylittävään yhtenäiseen alueeseen. Kuvassa on tunnistettu profiilin maksimihuipunkorkeus (Reference Profile, R_f) pisteessä H. Laaksolla tarkoitetaan puolestaan keskiviivasta sisään päin suuntautunutta osaa, joka rajataan keskiviivan alittavalla yhtenäisellä alueella. Kuvassa näkyy maksimilaaksonsyyvyys (Datum Profile, D_f) pisteessä B. (Pere 2009, s. 21-4 - 21-6)

Profiilinsyyvyys (Peak-to-valley Height, R_t) kertoo peräkkäisen huippu- ja laaksoviivan välisen etäisyyden mittauspituudella. Mittauspituus (Sampling Length, L) tarkoittaa tarkasteltavaa pituutta, jonka alueelta mitataan pinnankarheuden suureita. Kuvassa näkyy

myös kappaleen perusviiva (Actual Profile, Af), joka kuvaa kappaleen pinnan muotoa. Tämän viivan pohjalta suoritetaan profiilin parametrien mittaukset. Pinnan korkeuksien absoluuttisia arvoja merkataan kuvassa tunnuksella h . Näiden tasaisella etäisyydellä toisistaan olevien suureiden kautta saadaan pinnankarheuden laskentaa varten tarvittava tieto. (Narayana 2006, s. 242-243)

Profiilin aritmeettinen keskipoikkeama (Mean Roughness Index, R_a) tarkoittaa profiilipoikkeamien absoluuttisten arvojen aritmeettista keskiarvoa L :n suuruisella pituudella ja se voidaan laskea käyttämällä kaavoja

$$R_a = \frac{1}{L} \int_{x=0}^{x=L} |h_i| dx \quad (1)$$

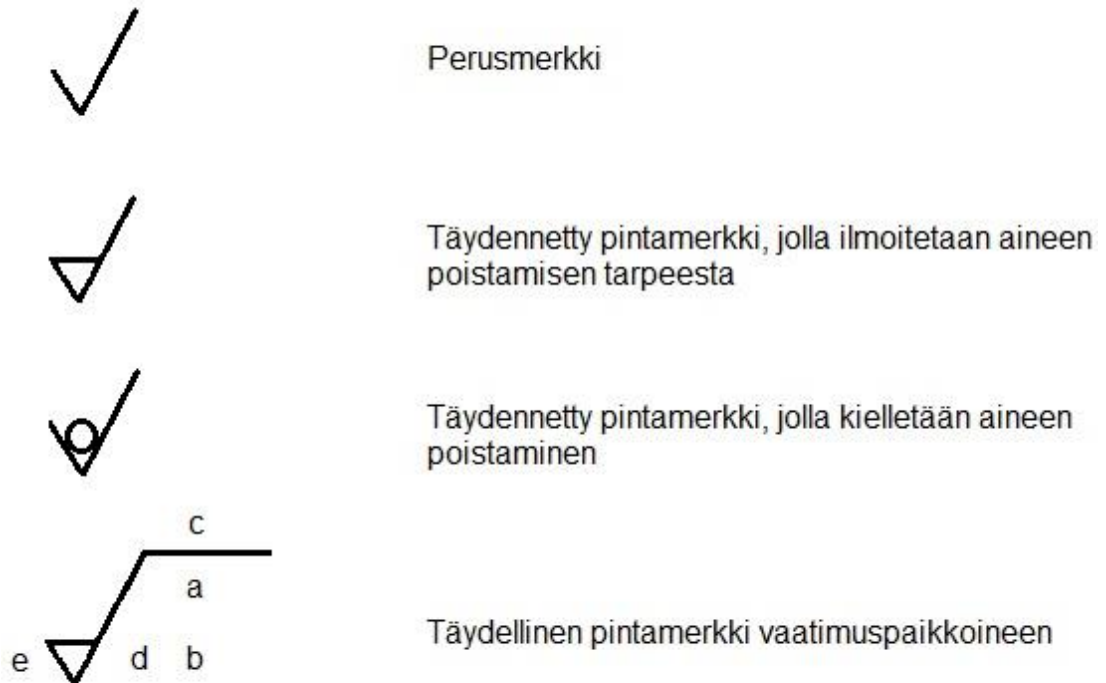
$$R_a \approx \frac{|h_1|+|h_2|+|h_3|+\dots+|h_n|}{n} \quad (2)$$

jossa L on mittauspituus, h_i vastaa pinnan karheuden absoluuttisia arvoja ja n on absoluuttisten arvojen mittauspisteiden lukumäärä. Profiilin aritmeettista keskipoikkeamaa voidaan havainnollistaa tarkasti kaavan 1 avulla ja likiarvoisesti käyttäen kaavaa 2. Keskipoikkeama lasketaan siis kääntämällä mitatun profiilin laaksot huipuiksi ottamalla niistä itseisarvo ja näin ollen eliminoidaan merkkivirheiden mahdollisuus. Keskiviivajärjestelmän heikkoutena on pintojen toimintaominaisuuksien kuvaaminen. Kaikista selvimmin ero näkyy tarkasteltaessa pintaa ja sen peilikuvapintaa. Keskipoikkeamasta ja profiilinsyvyydestä saadaan samat, mutta toiminnalliset eroavaisuudet voivat olla hyvinkin suuret. (Pere 2009, s. 21-7)

3.2.1 Pintamerkit

Pohanish (2016, s. 272) kuvaa pintamerkkien (englanniksi Surface Texture Symbols) tarkoittavan pinnan epäsäännöllisyyksien hallitsemista eri tavoin. Pintamerkkien avulla saadaan tarvittaessa kerrottua kaikki tarvittavat toimenpiteet halutun pinnan aikaansaamiseksi. Pintamerkkejä on kolmenlaisia: Perusmerkki ilmoittaa, että vaatimuksia on esitetty pinnan ominaisuuksille. Täydennetty pintamerkki kertoo, että pinnan ominaisuuksien saavuttamiseksi ainetta joko saa tai ei saa poistaa. Täydellinen pintamerkki on pe-

rusmerkki tai täydennetty pintamerkki, joka on jatkettu siten, että täydentävät pinnan ominaisuuksia koskevat vaatimukset voidaan merkitä siihen. Kuva 21 esittää käytettävät pintamerkkien merkkityypit.

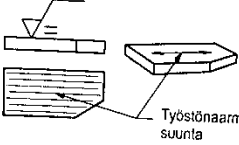
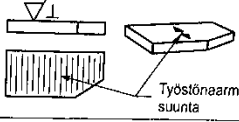
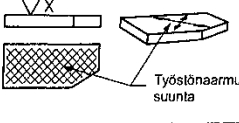


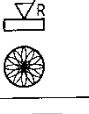
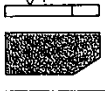


Kuva 21. Pintamerkit, mukailten lähteitä (Pere 2009; Pohanish 2016)

Täydellistä pintamerkkiä käytetään tapauksissa, joissa pintasuureen ja sen numeroarvon lisäksi on tarpeen määritellä lisävaatimuksia pinnan ominaisuuksille. Ylläolevassa kuvassa on esitetty erilaisten lisävaatimusten merkitsemispaikat pintamerkkiin. Nämä vaatimukset koostuvat pintasuureista, lukuarvoista sekä päästökaistasta/mittausjaksosta. Pintasuureella tarkoitetaan pinnan mikrogeometrisiä ominaisuuksia, päästökais-talla profiilikäyrän aallonpituusaluetta kahden määritellyn suodattimen välillä ja mittaus-jaksolla mittauspituuden muodostavia jaksoja. Paikkaan a tulee yksittäinen pinnan omi-naisuuksiin koskeva vaatimus. Jos näitä vaatimuksia on kaksi, toinen sijoitetaan paik-kaan b. Useampien vaatimusten tapauksessa pintamerkkiä jatketaan pystysuunnassa ja lisättävät vaatimukset kirjataan lisäriveille b:n alle. Paikkaan c kirjataan käytettävät vaatimukset koskien valmistusmenetelmää, käsittelyä, pinnoitusta tai muuta valmistus-prosessia. Paikkaan d merkataan pintakuvio ja työstönaarmujen suunta, joista kerrotaan

seuraavasta kappaleesta alkaen. Paikkaan e ilmoitetaan mahdollinen vaadittu työvara. (SFS-EN ISO 1302 2003)

Pintakuviot (englanniksi Surface Lay Symbols) tarkoittavat kappaleen pinnassa olevia naarmuja. Näiden havainnollistamisessa käytetään tunnuksia, jotka kuvaavat naarmujen suuntaa ja muotoa. Kuva 22 esittää pintakuvion ilmoittamisessa käytettävät tunnukset ja niiden tulkinat. Mikäli kyseessä on kuvassa esittämätön tapaus, tästä tarvitsee lisätä mitoitettavaan dokumenttiin huomautus.

Tunnus	Tulkinta ja esimerkki	
=	Naarmut yhdensuuntaisia piirustuksen sen viivan kanssa, johon pintamerkki on piirretty	
⊥	Naarmut kohtisuorassa piirustuksen sitä viivaa vastaan, johon pintamerkki on piirretty	
X	Naarmut ristikkäisesti vinossa piirustuksen siihen viivaan nähden, johon pintamerkki on piirretty	
M	Naarmut monisuuntaisesti	
C	Naarmut suunnilleen ympyrämaisesti pinnan keskipisteen suhteen	
R	Naarmut suunnilleen säteittäisesti pinnan keskipisteen suhteen	
P	Naarmut oval pistemäisiä, suunta ei ole määriteltävissä tai ovat kohoutumia	

Jos on tarpeen määritellä taulukossa esiintymätön pintakuviointi, tämä on piirustuksessa selvästi ilmoitettava lisäämällä sopiva huomautus.

Kuva 22. Pintakuvion ilmoittaminen, perustuen lähteeseen (Pere 2009, s. 21-30)

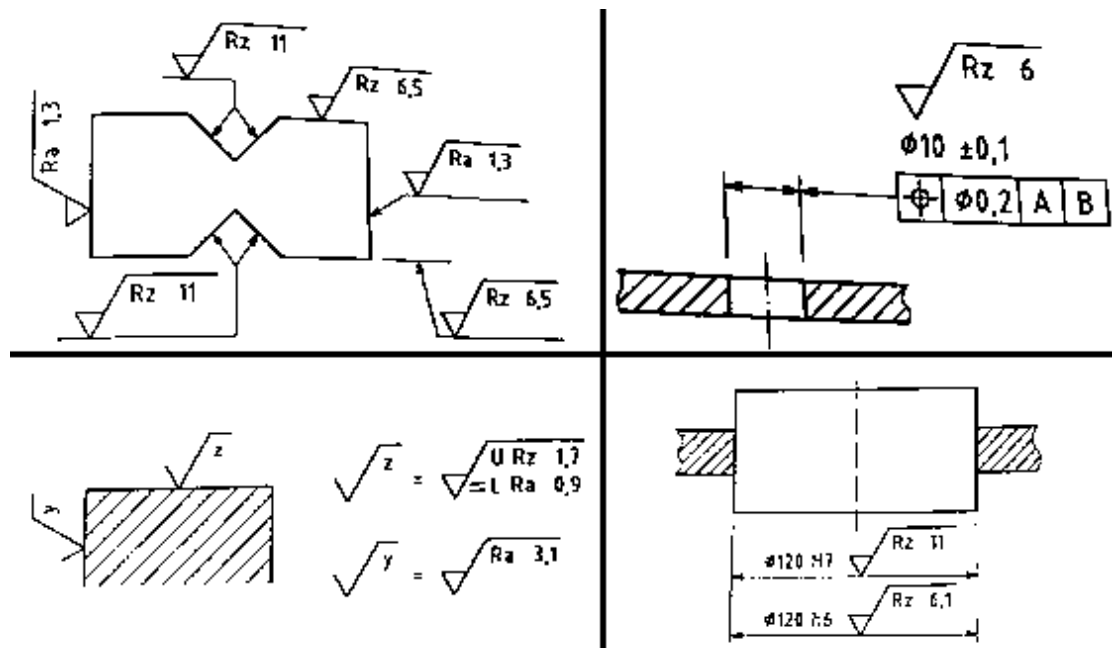
Pintakuvioiden lisäksi pintasuureet ovat määrittelyä tarvitseva käsite. Pintasuureiden ilmoittamisessa on vastattava neljään vaatimukseen, jotka mahdollistavat tulkinan. Vaatimukset ovat seuraavan laisia: Mikä kolmesta pintaprofiilista on kyseessä? Mikä profiilin

tunnusluku on. Kuinka monesta mittausjaksosta mittauspituus muodostuu? Miten tulki-taan määrittelyrajaa? Pintaprofiileja ovat R-profiili eli karheusprofiili, W-profiili eli aalto-maisuusprofiili ja P-profiili eli rakenneprofiili. Profiilin tunnusluvulla tarkoitetaan tarkkailu-tapaa valittuun pintaprofiiliin kohdistuen (esimerkiksi Ra, jossa a on tunnus). Mittausjak-sojen lukumäärän valinta riippuu siitä, kuinka yksityiskohtaisesti tarkastellaan ja kuinka pitkä mittauspituus on. Määrittelyrajan tulkinnessa on kaksi vaihtoehtoa: 16 %:n sääntö ja maksimisääntö. (Pere 2009, s. 21-24 - 21-47)

16 %:n säännöllä tarkoitetaan sitä, että pintasuureen ylä-/alarajan määrittämiä vaatimuk-sia varten pinta on hyväksyttävä, jos ei enempää kuin 16 % kaikista arvioitavaan pituu-teen perustuvista pintasuureen mitatuista arvoista ylitä/alita piirustuksissa tai teknisissä tuotedokumenteissa määriteltyä arvoa. Jos viitataan ylä- ja alarajoihin niin pintasuureen symbolia käytetään ilman "max" -merkintää. Maksimisäännöllä puolestaan tarkoitetaan seuraavaa: Tutkittaessa pintasuureen maksimi-arvojen vaatimuksia, yksikään koko tut-kittavan alueen pintasuureen mitatuista arvoista ei saa ylittää piirustuksissa tai teknisissä tuotedokumenteissa määriteltyä arvoa. Maksimaaliseen sallittavaan pintasuureen ar-voon viitattaessa pintasuureen symboliin tulee lisätä "max" -merkintä. (Pere 2009, s. 21-26)

Piirustuksiin ja muihin teknisiin dokumentteihin on olemassa monia erilaisia keinoja mer-kitä pinnankarheuksia. Yleissääntönä voidaan pitää sitä, että pintamerkki lisämerkintöi-neen sijoitetaan siten, että se on luettavissa piirustuksesta alhaalta tai oikealta. Pinta-merkin voi joko lisätä suoraan muotoviivalle, tai liitettävä siihen nuoleen päättyvällä vii-teviivalla. Tämän lisäksi pintamerkin tulee liittyä työkappaleen pintaan aineen ulkopuo-lelta. Pintamerkki voi myös sijaita mittaviivalla mittamerkinnän yhteydessä, jolloin luetta-vuus helpottuu. Lisäksi, pintamerkin voi sijoittaa geometrisia toleransseja kuvaavan to-leranssikehyksen päälle. Ahtaassa piirustustilassa pintamerkit voidaan yksinkertaistaa käyttämällä esimerkiksi kirjaintunnusta. Tässä tapauksessa merkitys tarvitsee selittää

asianomaisen työkappaleen vieressä. Kuva 23 esittää esimerkkejä kerrotuista merkitsemistavoista. (SFS-EN ISO 1302 2003)



Kuva 23. Pintamerkkien merkitsemistapoja, perustuen lähteeseen (Pere 2009)

Esitettyssä kuvassa vasemmassa yläreunassa pintamerkkejä on esitetty suoraan muotoviivalla ja viiteviivalla liitettynä kappaleeseen. Oikeassa yläreunassa pintamerkki on liitetty toleranssikehykseen. Vasemmassa alareunassa pintamerkit on esitetty yksinkertaistettuna ja niiden merkitykset on selitetty. Oikeassa alareunassa pintamerkit on puolestaan liitetty mittaviivoille. Pintamerkkien esittämistapoja on siis monia ja tämän johdosta ei ole vain yhtä oikeaa tapaa merkitä niitä piirustuksiin. Tässä aluvussa esitetyt kuvat ja määritelmät perustuvat ISO-standardien mukaisiin esittämistapoihin, joten on syytä tarkastella, onko eri standardityyppien välillä eroavaisuuksia.

3.2.2 Pinnankarheudet eri standardeissa

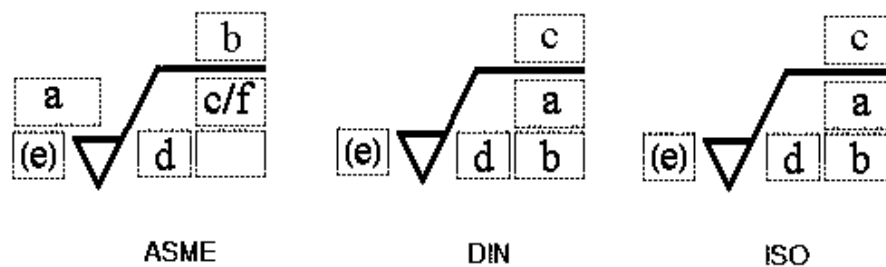
Pinnankarheuksia määriteltäessä noudatetaan mahdollisimman yhdenmukaisia merkin-tätapoja yksiselitteisyyden ja ymmärrettävyyden varmistamiseksi. Yrityksien välillä on käytössään erilaisia standardeja, joiden eroavaisuuksia havainnollistetaan tässä alu-vussa. Tutkittavia standardityyppejä ovat jo aluvussa 3.1.6 esitellyt SFS, ISO, DIN ja ASME. Tutkimusta vaikeuttaa se ettei itse DIN ja ASME standardeihin päästä käsiksi, mutta tutkimus suoritetaan kirjallisuudesta ja NX:stä löytyvästä tiedosta. Pere (2009, s.

21-64) kertoo, että SFS:n ja ISO:n pintamerkkien merkintätavoissa pintamerkit ovat lähestulkoon yhteneviä muilta osin paitsi karheusarvon tunnuksen osalta. Kuva 24 Havainnollistaa eroavaisuuden.

SFS 2039	SFS-ISO 1302	SFS-EN ISO 1302:2002
6,3/ ▽	Ra6,3/ ▽	▽ Ra 6,3

Kuva 24. Pintamerkin eroavaisuus SFS:n ja ISO:n kesken, perustuen lähteeseen (Pere 2009, s. 21-64)

Pintamerkkien rakenteiden eroavaisuuksia eri standardien välillä pystyy tutkimaan myös NX:n avulla. Pintamerkkiä lisättäessä voi asetuksista valita standardityypin, jonka seurauksena ruudulla näkyy määriteltävä pintamerkki, siihen syötettävien arvojen nimet ja niiden sijainnit. NX:n avulla voidaan tutkia ASME:n, DIN:n ja ISO:n eroavaisuuksia, jotka havainnollistaa Kuva 25.



Kuva 25. ASME:n, DIN:n ja ISO:n täydellisten pintamerkkien esittäminen NX:ssä

Aluvussa 3.2.1 on määritelty ISO:n täydellisen pintamerkin kirjaimien merkitykset ja sijainnit. Ylläolevan kuvan perusteella ISO ja DIN ovat yhtenevät, joten voidaan keskittyä ISO:n ja ASME:n eroavaisuuksiin. ISO:ssa pinnan ominaisuuksia koskevat vaatimukset a ja b sijaitsevat ASME:n merkintätavassa kirjaimissa a ja f. ASME:n kirjain b vastaa valmistusmenetelmää, joka on merkattu ISO:ssa kirjaimella c. Kirjaimet d ja e ovat yhtenevät näiden standardien välillä. ASME:n kirjain c (cutoff) tarkoittaa mittausjakson pituutta, jota ei esitetä ISO:ssa erikseen, vaan se on osa kirjaimia a ja b.

Pinnankarheuden tarkkuuden voi merkitä kolmella erilaisella tavalla. Lukuarvolla merkitäessä jokainen pystyy lukemaan arvon, mutta pinnantarkkuusasteen numerointia ja tarkkuuksia kuvaavia symboleja käyttäessä lukijan pitää tietää tai selvittää, millaisesta

tarkkuudesta on kyse. Taulukko 3 esittää erilaiset merkintätavat pinnankarheuden tarkkuuteen.

Taulukko 3. *Pinnankarheuden tarkkuuden merkintä (Narayana 2006, s. 246)*

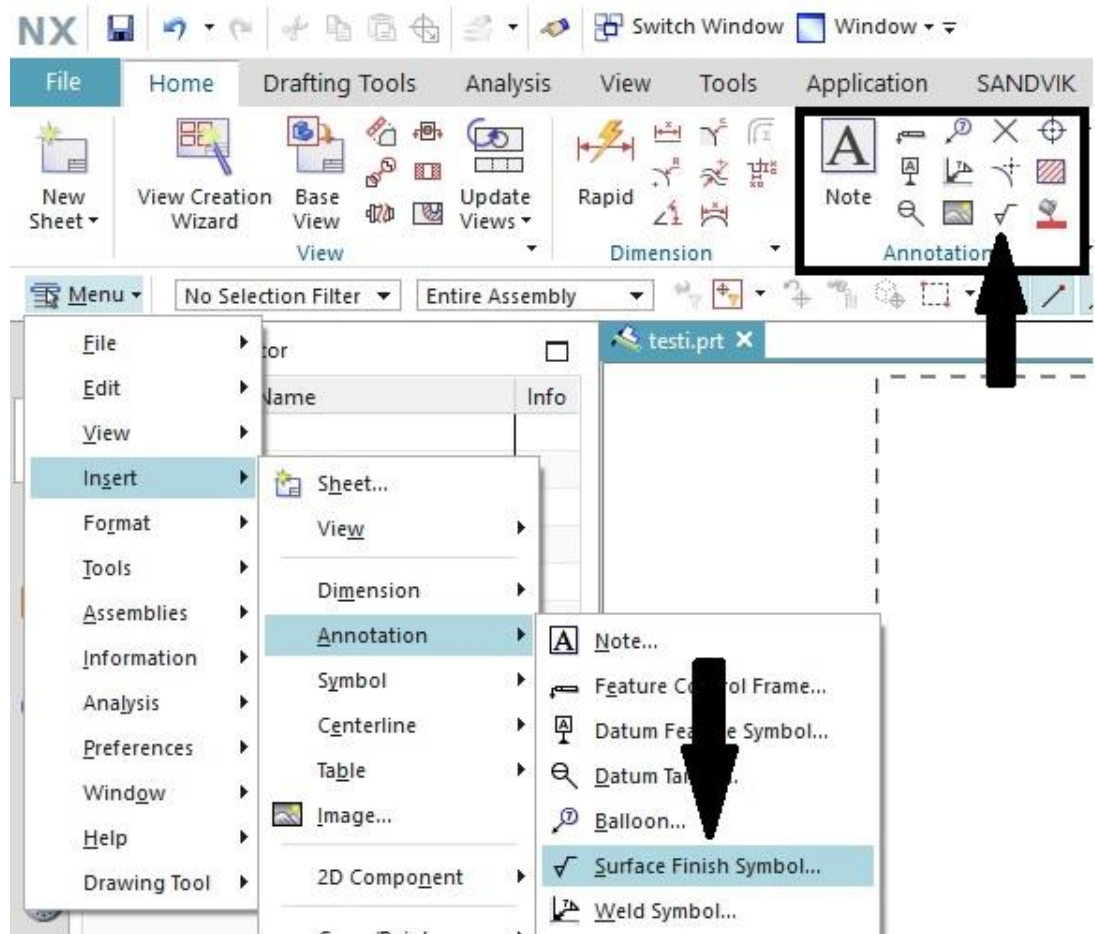
<i>Roughness values</i> $R_a \mu m$	<i>Roughness grade number</i>	<i>Roughness grade symbol</i>
50	N12	~
25	N11	▽
12.5	N10	
6.3	N9	▽▽
3.2	N8	
1.6	N7	
0.8	N6	▽▽▽
0.4	N5	
0.2	N4	
0.1	N3	▽▽▽▽
0.05	N2	
0.025	N1	

Ennen ISO:n mukaan määrittelemistä, SFS:n standardissa pintamerkit määriteltiin vain pinnankarheuden symbolin ja lukuarvon avulla (Pere 2009, s. 21-64). ASME:n ja DIN:n standardeissa pinnankarheus esitettiin lukuarvon avulla, osana pintamerkkiä. ISO-standardit poikkeavat siten, että pinnankarheuden esittämisessä voidaan käyttää myös skaalattua numeroarvoa ilmoittamaan pinnankarheus (englanniksi Roughness Grade Number). Tämän avulla saadaan selville pinnankarheuden suuruusluokka, eikä tarvitse miettiä mittayksikköjen arvoja (Narayana 2006, s. 246). Huonona puolena tässä vaikuttaa olevan se, että kun työstötarkkuudet kehittyvät niin skaalaus muuttuu. Tämän seurauksena vanhempien dokumenttien merkintöjen kuvauksissa on eroavaisuuksia uusien dokumenttien merkintöihin nähden tarkkuuksien skaalausten osilta.

3.2.3 Pinnankarheuksien merkitseminen Siemens NX:ssä

Kun kappaleeseen on tarkoitus lisätä pinnankarheuden informaatio, NX:n käyttäjä voi tehdä sen toleranssien tapaan joko työpiirustukseen tai suoraan 3D-malliin. Piirustuksissa pinnankarheuksia pystyy lisäämään joko klikkaamalla ruudun Home-välilehden Annotation-osuudesta kuvaketta Surface Finish Symbol tai valitsemalla sen polkua Menu-

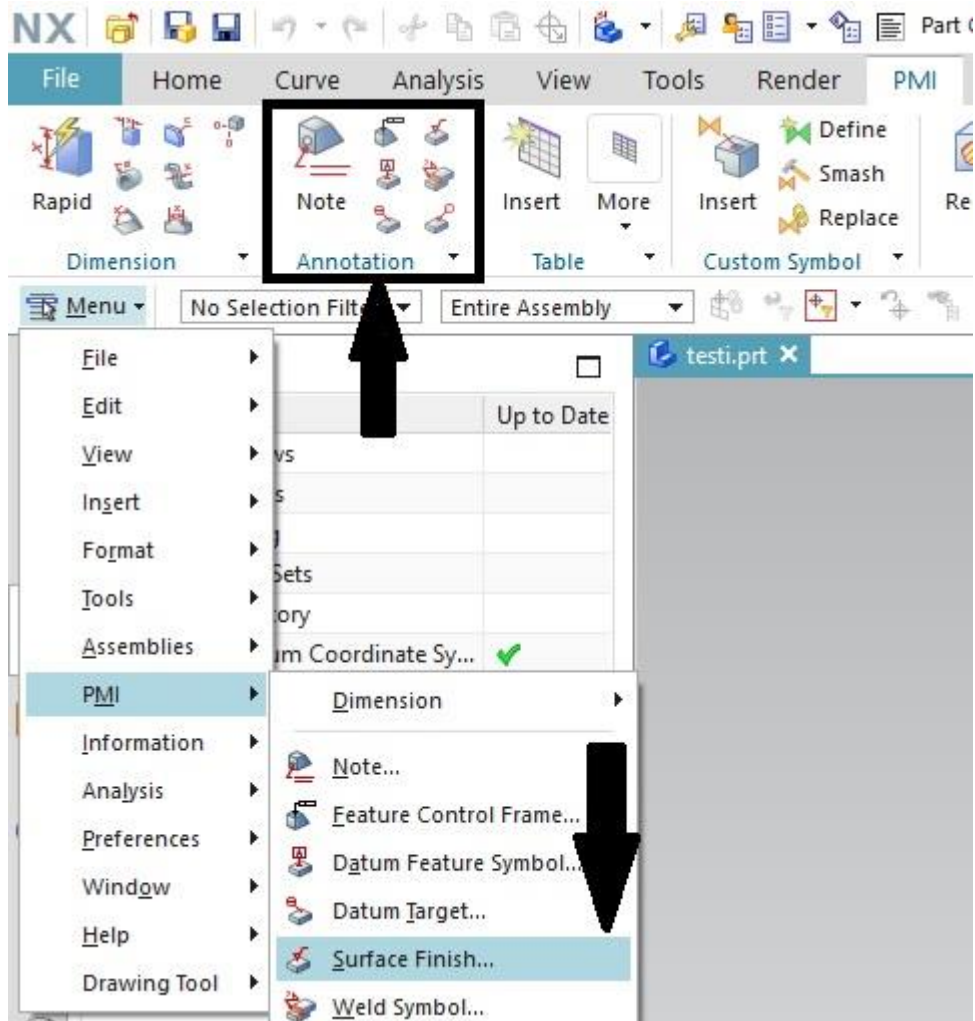
Insert-Annotation-Surface Finish Symbol pitkin. Kuva 26 havainnollistaa kerrotut lisäämistavat NX:ssä (Siemens 2019)



Kuva 26. Pinnankarheuden lisääminen NX-piirustukseen

3D-mallin tapauksessa käyttäjä pääsee lisäämään pinnankarheuden PMI-välilehdeltä kohdasta Annotation. Sieltä löytyy symboli nimeltä Surface Finish, jonka avulla pinnan-

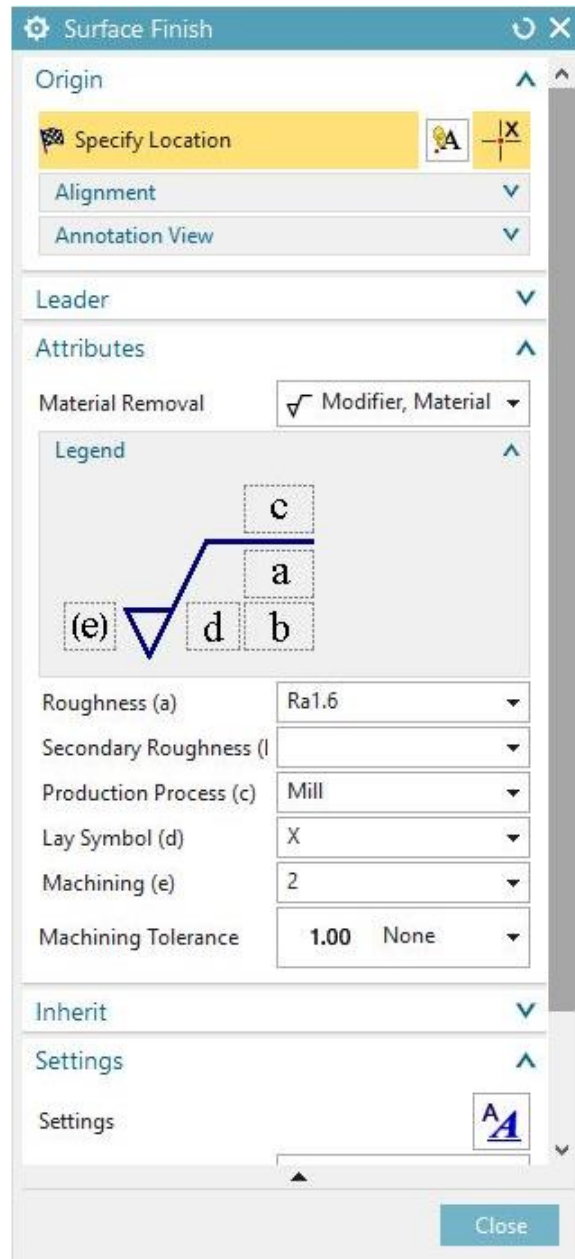
karheuden voi määrittää. Toinen vaihtoehto on seurata polkua Menu-PMI-Surface Finish. Kuva 27 havainnollistaa näiden kahden tavan noudattamisen pinnankarheuden lisäämisessä 3D-malliin.



Kuva 27. Pinnankarheuden lisääminen NX-malliin

Kun pinnankarheuden symboli on valittu, ruudulle tulee ikkuna, jossa asetetaan halutun lainen esittämistapa. Ensimmäisenä askeleena määritetään paikka, johon pinnankarheudella viitataan. Seuraavana askeleena on syytä valita oikeanlainen standardityyppi. 3D-malliin lisääessä standardin voi valita suoraan kohdasta Standard, mutta piirustukseen lisääessä kohdan Settings avaamalla käyttäjä voi valita käytettävän standardityypin, joita ovat muun muassa ISO, DIN ja ASME. Standardityypin valinnan jälkeen valitaan käytettävä pintamerkki ja lisätään vaatimuspaikkoihin tulevat ehdot. Ehtoja ovat pinnan ominaisuuden vaatimukset (Roughness), valmistusmenetelmän vaatimukset (Pro-

duction Process), pintakuviot ja naarmujen suunta (Lay Symbol) sekä työvara (Machining) tarvittaessa toleransseineen. Kuva 28 esittää pinnankarheuden lisäämisen NX:n käyttöympäristössä.



Kuva 28. Pinnankarheuden määrittäminen NX-piirustuksessa

Esitetyssä kuvassa on valittuna standardiksi ISO ja pintamerkiksi valittu täydellinen pintamerkki, joka ilmoittaa materiaalin poistamisen tarpeesta. Pinnan ominaisuuden ai-noana vaatimuksena on 1.6 µm:n Ra-arvo. Valmistusmenetelmänä on jyrsintä, jonka

työvarana on kaksi millimetriä. Pintakuvion naarmut ovat ristikkäisesti vinossa pintamerkin kohdistuvaan pintaan nähden. Toleranssialuetta ei määritetä pinnankarheuden merkinnässä. Kuvan oikeassa reunassa on havainnollistettu määriteltävä pinnankarheus.

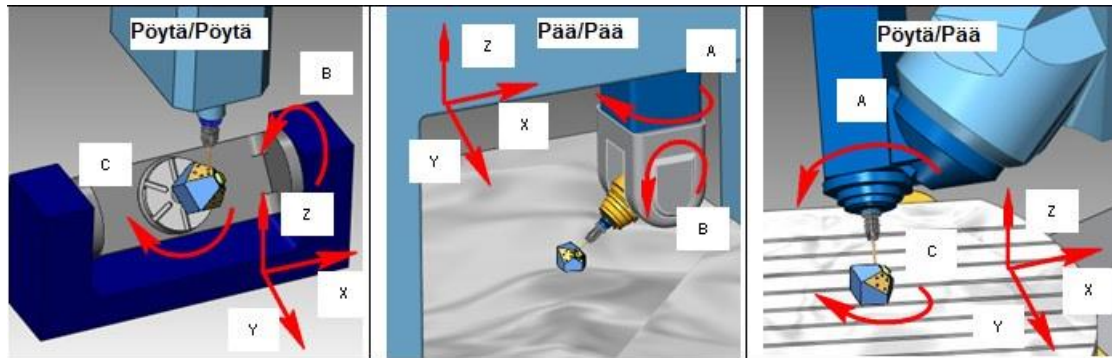
3.3 5-akselinen koneistuskeskus

Suunniteltuja kappaleita koneistettaessa käytetään soveltuvia työstökoneita halutun laisen lopputuloksen aikaansaamiseksi. Kun koneistuksen avulla tehtävä kappale on monimutkainen ja prismaattinen, on tarvetta käyttää useita eri työstökoneita tai sitten 5-akselista koneistuskeskusta, jonka avulla voi hyödyntää monia erilaisia työstötapoja. Tässä alaluvussa käydään läpi peruseriaatteita 5-akselisen koneistuskeskuksen rakenteesta, verrataan sitä 3-akseliseen koneistuskeskukseen sekä kerrotaan mahdollisista tarkkuuksiin liittyvistä ongelmakohtista ja niiden ennaltaehkäisemisestä. Alaluvun jälkeen lukijalla on selkeä käsitys 5-akselisesta koneistuskeskuksesta ja hän pystyy sisäistämään paremmin työn esimerkkitapauksen, jonka työstäminen toteutetaan 5-akselisen koneistuskeskuksen avulla.

Dimitrov ja Saxer (2012, s. 278) kertovat artikkelissaan, että 5-akselinen koneistuskeskus voidaan jakaa kolmeen eri pääosaan. Nämä pääosat ovat koneen fyysiset ominaisuudet, CNC-ohjausjärjestelmä sekä CNC-ohjaimen kyvykkyys. Koneen fyysisillä ominaisuuksilla tarkoitetaan akselien määrittelyä, koneen ulottuvuutta, vääntöä, karan maksimipyörimisnopeutta ja komponenttien laatua. CNC-ohjausjärjestelmällä tarkoitetaan komponentteja, joiden avulla koneen karat ja luistit liikkuvat. Komponentteja ovat muun muassa servomootorit, palloruuvit, sijainnin ohjain sekä syötön hallinta. CNC-ohjaimen kyvykkyyttä voidaan pitää koneen aivoina sillä sen funktiona on kontrolloida datankäsittelyä, käytettävissä olevaa muistin määrää sekä dynaamisen pyörimisen synkronisointia. Kerrottujen pääosien määrittelyä voidaan käyttää hyväksi 5-akselisen koneistuskeskuksen rakennuksessa ja valinnassa.

Koska 5-akselisia koneistuskeskuksia on monenlaisia, on olennaista määrittää käytettävä konfiguraatio. Lähteessä Shinde, Bhole (2015, s. 2) kerrotaan kolme yleisintä konfiguraatiota, jotka ovat Pöytä/Pöytä, Pää/Pää ja Pöytä/Pää (englanniksi Table/Table, Head/Head, Table/Head). Pöytä/Pöytä-konfiguraatiossa käytetään pöydän liikkumisessa standardiakselien X-Y-Z lisäksi myös kahta pyörimisakselia, jotka mahdollistavat 5-akselisen operoinnin. Työstöpää pysyy siis koko ajan paikallaan ja liikuttelun hoitaa kokonaan pöytä Tämä konfiguraatio sopii parhaiten kevyen kappaleen koneistamiseen. Pää/Pää-konfiguraatiossa puolestaan pöytä ei liiku ja kaikki liikkeet hoitaa työstöpää. Tämä konfiguraatio sopii parhaiten tilanteeseen, missä koneistettava kappale on liian

painava liikuteltavaksi pöydän avulla. Pöytä/Pää-konfiguraatiossa osa akseleista syntyy pöydän ja osa työstöpään avulla esimerkiksi siten, että työstöpää pyörii kahdella akselilla ja pöytä hoitaa lineaarisen liikuttelun kolmen akselin avulla. Kyseinen konfiguraatio sopii kahden aiemman toteutusmallin väliselle alueelle. Kuva 29 havainnollistaa esitetyt konfiguraatiot.



Kuva 29. 5-akselisten koneistuskeskusten yleisimmät konfiguraatiot, perustuu lähteeseen (Shinde, Bhole 2015)

Verrattaessa 5- ja 3-akselisia koneistuskeskuksia, voidaan jo niiden nimistä huomata eroavaisuus työstettävien akselien lukumäärässä. Zębala ja Plaza (2014, s. 345-348) kirjoittavat artikkelissaan 5- ja 3-akselisten koneistuskeskusten eroavaisuuksista. Ensimmäisenä huomiona kerrotaan olevan se, kuinka hyvin työkalut on aseteltu koneistettavaan pintaan nähden. Kaltevuuskulmalla on suuri vaikutus leikkausvoimiin sekä koneistuksella saavutettavaan pinnanlaatuun, jonka johdosta koneistuskeskuksen pitää olla soveltuva työstämään valmistukseen tulevaa kappaletta tarkasti. 3-akselisella koneistuskeskuksella on rajoitetummat liikealueet, jonka johdosta sen avulla ei pysty koneistamaan yhtä monipuolisesti kuin 5-akselisella. Kun vertaillaan erityyppisiä koneistuskeskuksia keskenään, avainasemaan nousevat erilaiset hyödyt, joiden seurauksena lopulta valittava konsepti on tuottavampi hankinta yritykselle.

Eräs merkittävä hyöty 5-akselisessa on se, että kyseisen laitteen käyttäminen nopeuttaa kiertoaikaa (englanniksi Cycle Time), jolla tarkoitetaan prosessin alusta loppuun viemiseen kuluvaa aikaa. Kiertoaikaa aikaa nopeuttavat koneistusvaiheessa esimerkiksi työkalujen paremman asettelun mahdollistamat suuremmat syöttönopeudet. Työkalujen optimoidun käyttämisen johdosta niiden käyttöänsä aikana saadaan enemmän aikaiseksi kun epätoivottua kulumista ei synny työkalun puutteellisen asettelun seurauksena. Useamman akselin käyttäminen mahdollistaa ideaalitapauksessa koneistamisen vain yhden kiinnityksen avulla, jonka johdosta aikaa ei kulu hukkaan kappaleen uudelleenasetteluun. Koneistuksen suunnittelua mietittäessä voi tulla mieleen, että mitä useampaa akselia käytetään, sitä monimutkaisempaa ja hitaampaa suunnittelu on. Kun kyseessä on

monimutkainen kappale, suunnittelu on itse asiassa yksinkertaisempaa viittä akselia käytettäessä. Kolmiakselisen työstön suunnittelussa tarvitsee ottaa huomioon kaikki laitteen rajoitteet ja suunnitella valmistus niitä silmälläpitäen kun taas viisiakselisessa operaatioiden luominen on yksinkertaisempaa ja vapaampaa kun koneistuskeskuksella voi koneistaa monipuolisemmin. (Zębala, Plaza 2014)

Monipuolisuutensa seurauksena 5-akselinen koneistuskeskus on kalliimpi hankkia ja ylläpitää kuin 3-akselinen. Valmistettavien kappaleiden geometria ja volyymit vaikuttavat koneistuskeskuksen tyyppin valintaan. Zębalan ja Plazan (2014) artikkelissa käsiteltiin kompleksista turbiinia, jonka valmistuskustannuksia vertailtiin 5- ja 3- akselisten koneistuskeskusten kesken. Tuloksista kävi ilmi että valmistettaessa pieniä määriä tuotetta, 5-akselinen tulee edullisemmaksi mutta kun valmistusmäärä nousee vähintään kahdeksaan erään samoilla asetuksilla niin 3-akselisen kustannukset ovatkin alhaisemmat. Kun kyseessä on pyörähdyssymmetrisiä tai geometrisesti yksinkertaisia tuotteita niin koneistuskeskukselta ei vaadita suurta monipuolisuutta. Näissä tapauksissa 3-akseliset työstökoneet tulevat kannattavimmaksi, sillä hankinta- ja ylläpitokustannukset ovat suuremmassa roolissa kuin koneistuksessa syntyvät kustannukset. Huomionarvoista valinnassa ovat siis, millaisia kappaleita koneistuskeskuksella tehdään ja kuinka suuria volyymeja tuotteita valmistetaan.

Kun 5-akselisia koneistuskeskuksia käytetään osana valmistusprosesseja, on syytä olla kartalla yleisimmistä ongelmakohtista. Lähteessä (Lin, Lin 2010, s. 136-137) kerrotaan koneistamisesta johtuvien muotovirheiden olevan yleinen huolenaihe 5-akselisessa valmistuksessa. Näillä virheillä tarkoitetaan approksimaatioita, jotka aiheuttavat liian suurta ja pientä materiaalin poistamista koneistuksessa. Jos liikeratojen mallintamisessa käytetään lineaarista orientaatiota niin liikeratojen muuttaminen ei onnistu tarkasti tapauksissa, joissa orientaatio ei ole kokonaan lineaarista. Perinteinen lähestymistapa tähän ongelmaan on määrittää työstöradalle tarvittava määrä segmenttejä, jotta virheen mahdollisuus olisin sallituissa rajoissa. Tämä tapa voi kuitenkin aiheuttaa koneistuspöydän liiallista värähtelyä ja kasvattaa koneistamiseen kuluvaan aikaan, jonka johdosta lähestymistapa ei ole optimaalinen.

Edellisessä kappaleessa kuvattuun ongelmaan löytyy kirjallisuudesta (Shinde, Bhole 2015; Sungchul, Taehoon 2003) kaksi ratkaisutapaa. Ensimmäinen on tarpeeksi tarkan CAD-mallin luominen, jonka käsittely ei aiheuta sallittujen rajojen ulkopuolella olevia muotovirheitä. Toinen ratkaisutapa on nimeltään NURBS (Non-Uniform Rational B-Spline), jonka avulla saavutetaan tarkkoja liikeratoja muodostamatta kuitenkaan suurta

määrää NC-dataa. NURBS-käyriä pystytään vapaasti muokkaamaan siten, etteivät niiden laadulliset ominaisuudet muutu. Käyrät muodostetaan erilaisten yhtälöiden avulla ja niitä muokattaessa ei tarvitse jälkikäteen simuloida kuin muuttunutta aluetta. Vertailtaessa lineaarista orientaatiota ja NURBS-tapaa erilaisissa tapauksissa huomattiin, että mitä monimutkaisempi muotoviiva kappaleen pinnalla on, sitä suurempi hyöty NURBS:n avulla saadaan.

Tarkkuuteen vaikuttavana ongelmakohtana voidaan pitää myös koneistuskeskusten liikkuvien osien kulumista. Näitä ovat esimerkiksi pinnat, joiden avulla hallitaan aksiaalista liikuttelua. Perinteinen tapa lineaarisen liikkumapinnan toteuttamiseen on käyttää kierteellä olevaa laakeria, jonka kuulat koskettavat kierrettä. Koska pinnat ovat koko ajan kosketuksissa, ajan kuluessa kuoliin muodostuu kierteen muotoinen, tarkkuutta heikentävä kuluma. Ratkaisuna tähän ongelmaan on hydrostaattinen toteutus, jossa metallipinnat eivät kosketa toisiaan. Hoonattujen liukumapintojen välissä on suuttimen avulla syötettyä paineistettua öljyä, jota on varastoituna liukuvassa osassa oleviin öljykammioihin. Öljyn paine säädetään siten, että se jaksaa pitää liukuvan osan irrallaan liukumapinnasta. Hydrostaattisessa ratkaisussa tarvitaan siis painekammio ja öljykammio, mutta perinteiseen tapaan verrattuna käyttöikä on pidempi. Kuva 30 havainnollistaa hydrostaattista ratkaisua. (Shinde, Bhole 2015)



Kuva 30. Hydrostaattinen liikuttaminen, perustuu lähteeseen (Shinde, Bhole 2015)

Kappaletta koneistettaessa syntyy välttämättä kulumista työkaluihin, jonka seurauksena pinnanlaatu ja tarkkuus heikentyvät. Kyseinen ongelma voidaan minimoida käyttämällä hyväksi kylmäfysiikkaa, jonka tarkoituksena on laskea työstölämpötilaa ja näin ollen hidastaa työkalun kulumista. Menetelmässä käytetään useimmiten jäähdytykseen nestemäistä tyyppiä tai hiilidioksidia, joka laskee lämpötilan noin miinus seitsemääkymmenneenviiteen celsiusasteeseen sekoitessaan ympäröivään ilmaan. Kun koneistettavan alueen lämpötila lasketaan matalaksi, työkalun terä ei kuumene niin nopeasti ja tämän

johdosta tarkkuus paranee. Jäähdyttäminen voidaan toteuttaa käyttämällä työkaluja, joissa on jäähdytyskanavia aineen oikeaan paikkaan kohdistamiseen. Vertailtaessa koneistamista ilman jäähdytystä ja jäähdytyksen kanssa, huomattiin selkeästi parempia tuloksia: Työstöterän sivun kuluminen laski 63 prosenttia arvosta 0.16 mm arvoon 0.06 mm ja materiaalin poisto lisääntyi 72 prosenttia syötön lisäämisen seurauksena. Näiden seurauksena menetelmän käyttö lisää koneistamisen tehokkuutta ja näin ollen myös tuottavuus kasvaa. (Cordes, Hübner et al. 2014)

Viimeisenä tässä työssä käsiteltävänä ongelmana on karan lämpeneminen, jonka seurauksena työstökappaletta ei pystytä käsittelemään optimaalisesti ja näin ollen lopputuloksen tarkkuus kärsii. Ratkaisuna tähän ongelmaan on suunnitella systeemi, joka alentaa lämpenemistä ja poistaa odottamattomat vaikutukset koneistamisessa syntyvästä lämpenemisestä. Systeemi voidaan suunnitella symmetriseksi karan akselin suhteen, jolloin lämpeneminen laajentaa elementtejä tasaisesti. Lämpöä lisääviä elementtejä, kuten sähkökeskus ja jäähdytyslaitteisto, voidaan eristää muusta laitteesta. Lämpötilaystävällinen ratkaisu koostuu kahdesta funktiosta: Karaan keskittyvässä funktiossa jaotellaan sensoreita tasaisesti karan ylä- ja alaosaan. Tämä mahdollistaa muun muassa pyörimisnopeuden muuttumisen ja pysähtymisen seuraamisen. Palautteen pohjalta systeemi pystyy kompensoimaan muodonmuutoksia. Toinen funktio keskittyy valuihin ja siinä levitetään sensoreita tasaisesti ympäri konetta. Tämän seurauksena pystytään seuraamaan lämpötilaa reaaliaikaisesti valetuissa osissa. Jos jokin lämpötila ylittää sallitut rajat, systeemi ilmoittaa siitä ja koneistus jatkuu taas rajojen alittuessa. (Shinde, Bhole 2015)

3.4 Yhteenveto

Tarkkuusvalmistuksen ymmärtäminen on eräs olennainen osa tätä opinnäytetyötä, joten käsitteen määrittäminen ja siihen liittyvistä asioista kertominen on välttämätöntä työssä esitettävän tiedon sisäistämiseksi. Luvun alussa olevan tarkkuusvalmistuksen määrittelyn jälkeen käytiin läpi toleranssien kokonaisuutta. Kyseisessä alaluvussa esitellään jokaisen toleranssimerkin merkitys, niiden piirtämistapa sekä niihin liittyviä sääntöjä ja rajoitteita. Huomionarvoisia asioita perusteorian lisäksi olivat toleranssien merkitseminen suunnitteluohjelmiston NX avulla sekä toleranssien merkitsemisen eroavaisuudet erilaisissa standardeissa. Kun toleransseihin liittyvä tarpeellinen tieto on omaksuttu, siirrytään pinnankarheudesta kertomiseen.

Pinnankarheuteen liittyvissä alaluvuissa noudatettiin samankaltaista rakennetta kuin toleranssien kanssa. Ensimmäisenä määriteltiin pinnankarheus käsitteenä, jonka jälkeen

kerrottiin teoriaa pinnankarheuden määrittämiseen liittyen. Seuraavana keskityttiin pintamerkkeihin ja niiden rakenteeseen. Alaluvussa huomattiin, että pintamerkkeihin pystytään lisäämään kaikki tarvittava tieto valmistuksen suunnitteluun liittyen ja niiden esittäminen on yksiselitteinen. Kun pintamerkkejä rakennetaan ja merkitään, on huomionarvoista tietää käytettävä standardi, sillä eri standardien välillä on eroavaisuuksia. Luvussa käydään läpi kaikki tietämisen arvoiset eroavaisuudet ja yhteneväisyydet eri standardien välillä. Tämän lisäksi luvussa havainnollistettiin pinnankarheuden merkitseminen malleihin ja piirustuksiin NX:ää hyväksikäyttäen.

Kappaleen valmistuksen ymmärtämiseksi on olennaista tutustua käytettävään valmistuslaitteistoon joka on tässä työssä 5-akselinen koneistuskeskus. Ensimmäisenä käydään läpi yleistä teoriaa 5-akselisista koneistuskeskuksista ja niihin liittyvistä erilaisista konfiguraatioista. Kun peruskäsitys on saatu, siirrytään vertailemaan 5- ja 3-akselisia koneistuskeskuksia keskenään kertoen useamman työstöakselin käyttämisen hyödyistä. Tämän lisäksi esitellään erilaisia tekijöitä joiden pohjalta koneistuskeskusten hankkimisiin kohdistuvia valintoja pystytään tekemään. Yhtenä tärkeänä kokonaisuutena on erilaisien ongelmakohtien sisäistäminen. Luvussa esitettiin erilaisia ongelmia aiheuttavia tekijöitä ja kerrottiin keinoja niiden ennalta ehkäisemiseksi ja vaikutuksen minimoimiseksi. Kaikki esitellyt ongelmakohdat liittyvät koneistuksen tarkkuuteen ja näin ollen myös tämä osa-alue on hyödyllistä sisällyttää tarkkuusvalmistus-lukuun. Kun tarpeellinen tieto aihepiiristä on sisäistetty, voidaan työssä siirtyä eteenpäin käsittelemään esimerkkitapausta.

4. ESIMERKKITAPPAUS

Sandvikin erästä ydinosaamista ovat porakoneet ja kyseisestä syystä johtuen porakoneisiin liittyvät osat on luokiteltu salaisiksi. Tässä luvussa kerrotaan mahdollisimman tarkalla tasolla esimerkkinä olevasta kappaleesta kuitenkin siten, ettei salaisiksi luokiteltuja tietoja tule ilmi. Aihepiirin kehittämisen toteuttaminen kerrotaan myös tämän luvun yhteydessä. Opinnäytetyötä tehdessä päästiin tutkimaan yksityiskohtaisesti esimerkkitapaukseen liittyviä tietoja. Näin ollen saatiin tarkka nykykuva tilanteesta ja voitiin tunnistaa kehittämistä tarvitsevia kohteita. Diplomityöstä saatava paras mahdollinen hyöty pääteltiin olevan prismaattisissa kappaleissa, sillä monimutkaiset geometriset muodot asettavat enemmän vaatimuksia tarkkuusvalmistuksen suhteen verrattaessa niitä pyörähdysymmetrisiin kappaleisiin. Tästä syystä suuremman määrän vaatimuksia sisältävän tarkkuusvalmistusprosessin optimoimisella on potentiaalia päästä parempiin lopputuloksiin.

Esimerkkitapauksena käytetään porakoneen osaa, jonka monimutkaisuuden johdosta 5-akselinen koneistaminen on paras vaihtoehto kappaleen realisointiin. Kappaleen käsittelemiseen riittää yksi koneistuskerta, jonka aikana raaka-aineesta työstetään tarkkuusvalmistuksella valmis porakoneeseen liitettävä lopputulos. Kyseessä on tarkkuusvalmistusta vaativa uusi komponentti, jonka vuoksi CAM-prosessi alkaa kappaleen geometrian suunnittelusta ja päättyy valmiin kappaleen laatutarkistukseen. Sandvikin toimintaan kuuluu myös valmistettujen tuotteiden korjaus, huolto sekä parantelu mutta tässä esimerkkitapauksessa lähtökohtana ei ole jo aikaisemmin tuotettua versiota kappaleesta, vaan liikkeelle lähdetään puhtaalta pöydältä. Esimerkkitapauksessa käytettävästä tarkkuusalueesta voidaan kertoa, että se sijoittuu toleranssien ja pinnankarheuksien osilta luvussa 3 määriteltyjen perinteisen ja tarkan koneistuksen välimaastoon.

Diplomityön alkaessa tarkkuusvalmistusta vaativan kappaleen geometriaa ja kiinnittämistä ei oltu suunniteltu loppuun, joten itse valmistamiseen liittyvä CAM-suunnittelu oli vasta alkutekijöissään. Valitun esimerkkitapauksen tilanteen tekee mielenkiintoiseksi se, ettei työssä voi käyttää hyväksi jo aikaisemmin toteutunutta, valmista prosessisuunnitelmaa. Tämän johdosta esimerkkitapaukseen liittyvän prosessin kulku tarvitsee muodostaa haastatteleamalla eri rooleissa olleita työntekijöitä. Luvussa 5 käydään läpi esimerkkitapaukseen liittyvä prosessi ja havainnollistetaan kokonaisuus selkeyttävien kuvien

avulla. Tarkempaa huomiota kiinnitetään tässä työssä digitaaliseen osuuteen suunnitteluvaiheessa sekä CMM ohjelman luomisessa.

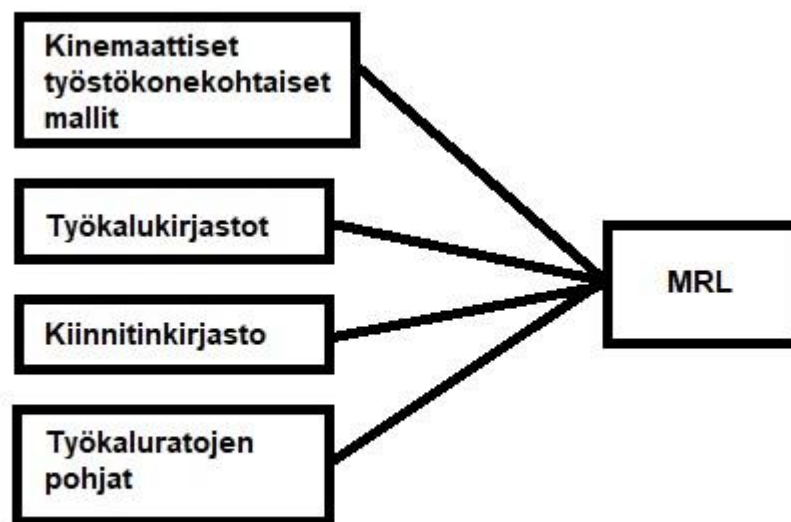
Koska tarkkuusvalmistukseen liittyvää esimerkkitapausta ei voida yksityiskohtaisesti näyttää julkisessa diplomityössä, tarvitsee käyttää yleisen tason esimerkkejä kehitysehdotusten tutkimisessa. Alaluvussa 6.2 paneudutaan aluksi NX:n CAD-puolella avulla luotaviin erilaisiin muotoihin, joita luodaan hyväksikäyttämällä NX Openin ohjelmointirajapintojen kirjastossa olevia komentoja. NX Openia käytetään puhtaasti Pythonin ohjelmointikielen ominaisuuksia hyödyntäen, mutta myös muissa ohjelmointikielissä toimintaperiaate on hyvin pitkälti samanlainen. Komentojen nimissä on eroavaisuuksia mutta suoritettavat operaatiot ovat yhtenevät. Tarkkuusvalmistukseen liittyen ohjelmointirajapintojen käyttämisessä ei paneuduta syvällisesti, mutta myös toleransseja ja pinnankarheuksia voidaan vapaasti määrittää NX Openin avulla.

CAM-puolen tutkimisessa esimerkkinä käytetään pyörähdyssymmetristä kappaletta koska pääasiallisena tavoitteena on pitää alaluvussa 6.2 NX Openin avulla luotu koodi mahdollisimman selkeänä ja yksinkertaisena. Prismaattisten, tarkkuusvalmistusta vaativien kappaleiden kohdalla NX Openin käyttämisen toimintaperiaate on sama mutta lukijan kannalta liitteissä olevien koodien tulkitseminen olisi vaikeaselkoisempaa monimutkaisemman rakenteen vuoksi. Tarkkuusvalmistusta silmälläpitäen myös CAM-puolella voidaan vaikuttaa toleransseihin ja pinnankarheuteen eri komentojen avulla. Tämän lisäksi valitulla koneistuskeskuksella sekä työkaluilla on vaikutusta kappaleen tarkkuuteen.

CMM-puolella käytetään samaa kappaletta kuin CAM-puolella, jotta lukija pystyy havaitsemaan, kuinka sulavaa NX Openin hyödyntäminen on saman tarkkuusvalmistukseen liittyvän kappaleen eri vaiheiden aikana. Alaluvussa 6.3 esitettyjä toimintaperiaatteita voidaan hyödyntää suoraan yrityksessä olevaan tarkkuusvalmistuksen avulla tehtävään kappaleeseen, sillä havainnollistetut asiat toimivat ratkaisukeinoina yleisellä tasolla. Diplomityössä pyritään myös tässä osa-alueessa mahdollisimman selkeään ja yksinkertaiseen esittämiseen lukijaa silmälläpitäen. Alaluvun alkupuolella käsitellään Zeissin Calypsoa mutta tutkimisen ja lisenssien käytön kannalta käydään läpi NX:n puolella olevaa kokonaisuutta CMM ohjelmien tekemiseen. Tästä johtuen NX Openia voidaan hyödyntää myös tässä osa-alueessa. Tarkkuusvalmistuksen kannalta CMM ohjelmien avulla voidaan tarkistaa myös tiukimpien toleranssien täytyminen, mikäli käytössä on tarpeeksi laadukas mittauskone ja ohjelmisto. Esimerkkitapausta käsitellään siis pääasiallisesti luvuissa 5 ja 6.

5. CAM-PROSESSIN METODOLOGIA

Tässä luvussa käydään läpi Sandvikin käytössä oleva rakenne tarkkuusvalmistukseen liittyvälle CAM-prosessille sekä siihen liittyviä ongelmia esimerkkitapauksessa olevan prismaattisen porakoneen osan CAM-prosessin tutkimisen avulla. Tietoa kerätään haastatteleamalla eri työtehtävissä toimivia henkilöitä ja tutkimalla yrityksessä käytettäviä ohjelmia. Näiden pohjalta pystytään luomaan havainnollistava malli prosessin rakenteesta. CAM-prosessin ensimmäisenä kokonaisuutena voidaan pitää Siemensin tuote- ja suunnittelutiedon hallintaohjelmistoa Teamcenter, jonka avulla pystytään muun muassa hallitsemaan nimikkeitä, niiden rakenteita ja niissä käytettäviä resursseja. Sandvikin Teamcenterin avaintekijänä on MRL. Kuva 31 esittää Teamcenteristä hyödynnettävien MRL:n osa-alueet.



Kuva 31. MRL:n osa-alueet

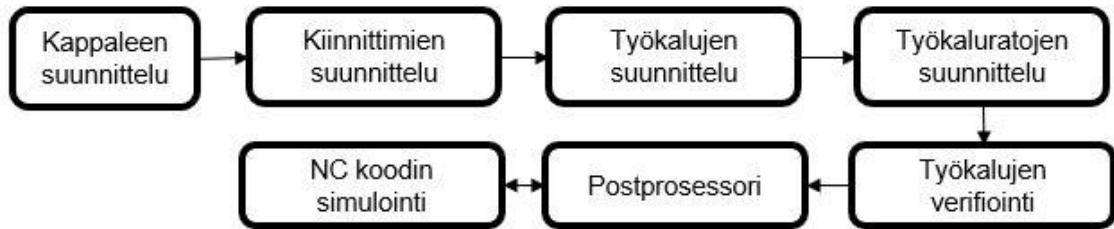
Ensimmäisenä osa-alueena ovat tuotannon tiloissa olevien työstökoneiden mallit, jotka tarkoittavat virtuaalisia, identtisiä malleja käytössä olevien työstökoneiden rakenteista. Mallit on rakennettu siten, että niiden avulla pystyy tutkimaan muun muassa kinemaattisia liikkeitä, joita voidaan käyttää esimerkiksi törmäystarkasteluissa ja koneistuksen simuloinnissa. Työkalukirjastot pitävät sisällään työkalumyyjien katalogeja, koneistuksiin valittuja työkaluja sekä kokoonpanoja, joissa työkalu on kiinnitettyä työkalupitimeen. Tämänhetkisenä ongelmana on joidenkin työkalumyyjien sähköisten katalogien puuttu-

minen, joissa olisi jo valmiina työkaluun liittyviä parametreja. Kun parametreja ei ole valmiina, Teamcenterin käyttäjän tarvitsee lisätä ne manuaalisesti järjestelmään, mikä vie paljon aikaa. Työkalun ja sen pitimen kokoonpano tarvitsee luoda, mutta tämä mahdollistaa niiden hyödyntämisen prosessin myöhemmissä vaiheissa. MRL:n ansiosta prosessissa käytettävät työkalut löytyvät kaikki samasta paikasta, jotta suunnittelussa ja datankäsittelyssä menisi minimimäärä aikaa.

Kiinnitinkirjasto pitää sisällään aikaisemmin suunniteltujen kiinnityksien dokumentaation, joiden kokonaisuuksia voidaan optimaalisessa tapauksessa hyödyntää sellaisenaan suunnitteluvaiheessa. Tämän lisäksi kiinnitinkirjastosta saa käsityksen siitä, millaisia kiinnittimiä on mahdollista käyttää milläkin työstökoneella. Aikaa voidaan säästää myös silloin kun muokataan valmiita malleja sopivaksi tapausta varten, verrattuna kokonaan uuden kiinnittimen suunnitteluun. Työkaluratojen pohjilla tarkoitetaan kiinnitinkirjaston tapaan aikaisemmissa CAM-prosesseissa suunniteltuja ratkaisuja, joissa olevaa tietoa pystytään hyödyntämään. Työkaluratojen pohjista näkee muun muassa valittuja työkaluja, syötön ja nopeuksien arvoja sekä työkalujen liikeratoja aikaisempien kappaleiden valmistuksissa. MRL:n avulla saatavaa tietoa voidaan hyödyntää kappaleen, kiinnittimien, työkalujen ja työkaluratojen suunnittelussa.

Ennen kuin tarkkuusvalmistuksen suunnitteluprosessi voi alkaa, tarvitsee tietää siihen liittyvät rajoitukset. Rajoituksia asettavat eri komponenttien, esimerkiksi tiivisteiden, valmistajat. Tämän lisäksi valittavat standardit, koordinaattimittauskoneiden tarkkuudet, työstökoneiden ominaisuudet, suunnittelijan aikaisempi tietämys ja lujuuslaskennasta saatava tieto asettavat rajoituksia. Kun rajoitukset ovat selvillä, voidaan siirtyä suunnitteluvaiheeseen joka tapahtuu Siemensin CAD/CAM ohjelmistossa NX. Ohjelmiston

avulla suunnitellaan, verifioidaan ja simuloidaan kappaleen työstössä käytettäviä resursseja. Kuva 32 esittää NX:n sisällä suoritettavat vaiheet.



Kuva 32. NX:n sisällä suoritettavat vaiheet

Esitetyn kuvan mukaan ensimmäisenä vaiheena tapahtuu kappaleen suunnittelu. Suunnittelijan tehtävänä on mallintaa geometria, joka täyttää kaikki tapauskohtaiset rajoitteet. Suunniteltaessa geometriaa, suunnittelija on yhteydessä lujuuslaskijaan, joka varmistaa geometrian sopivuuden. Käytettäviä ohjelmia lujuuslaskentaan ovat Ansys ja Matlab. Staattinen tarkastelu suoritetaan Ansyskella ja dynaaminen tarkastelu puolestaan Matlabiin koodatulla ohjelmalla. Suunnittelijalla on myös käytössään MRL, josta näkee rajoitteita ja käytössä olevia resursseja. Suunnittelun tuloksena syntyy kappaleen geometrinen malli sekä mitat ja tarkkuudet havainnollistava työpiirustus. Kappaleen suunnittelu toimii lähtökohtana lopuille valmistuksen suunnittelun vaiheille, joten se on CAM-prosessin ensimmäinen suoritettava vaihe.

Kiinnittimien suunnittelu tarkoittaa koneistettavan kappaleen geometrian perusteella tehtävää suunnittelua kiinnityksen suhteen. Menetelmäsuunnittelija voi tehdä kiinnitinsuunnittelun itse, yhteistyössä toisen osapuolen kanssa tai sitten ulkoistaa koko suunnittelun toiselle osapuolelle. Joka tapauksessa kiinnittimien suunnittelussa hyödynnetään MRL:n tietoja ja tarvittaessa luodaan uusia resursseja kiinnittimiin liittyen. Suunnitellut kiinnittimet tarkastetaan yrityksen toimesta, jonka jälkeen voidaan aloittaa työkalujen suunnittelu. Työkalujen suunnittelussa käytetään niin ikään hyväksi MRL:n tietoja, jonka johdosta kappaleen työstämiseen pystytään valitsemaan käytettävissä olevia työkaluja. Kappaleen ja kiinnittimien suunnittelun ansiosta työkalujen suunnittelussa voidaan ennakoida valittavien työkalujen rakennetta siten, ettei niitä tarvitsisi CAM-prosessin seuraavissa vaiheissa.

Työkaluratojen suunnittelussa tarkoituksena on suunnitella kappaleen koneistaminen. Suunnittelutietoa ja MRL:n sisältöä käyttämällä luodaan operaatioita, jotka muuttavat työstökoneelle tulevan raaka-aineen mallinnetuksi kappaleeksi. Suunnittelija määrittää

käytettäville operaatioille parametrit ja resurssit, jonka yhteydessä syntyy myös operaatiojärjestys. Työkalujen valinnassa ei tarvitse noudattaa työkalujen suunnittelussa valittuja työkaluja, vaan suunnittelijalla on mahdollisuus tehdä muutoksia ja valita erilaisia komponentteja MRL:n sisällöstä. Valittujen työkalujen avulla hahmotellaan koneistamisessa käytettävät työkaluradat. Suunnittelun jälkeen voidaan arvioida, kuinka kauan koneistaminen kestää ja kuinka monimutkainen koneistusprosessi on. Suunnittelijalla on tavoitteena luoda mahdollisimman nopea ja vähin operaatioin suoritettava ohjelma, joka täyttää tarkkuusvalmistukselle annetut vaatimukset.

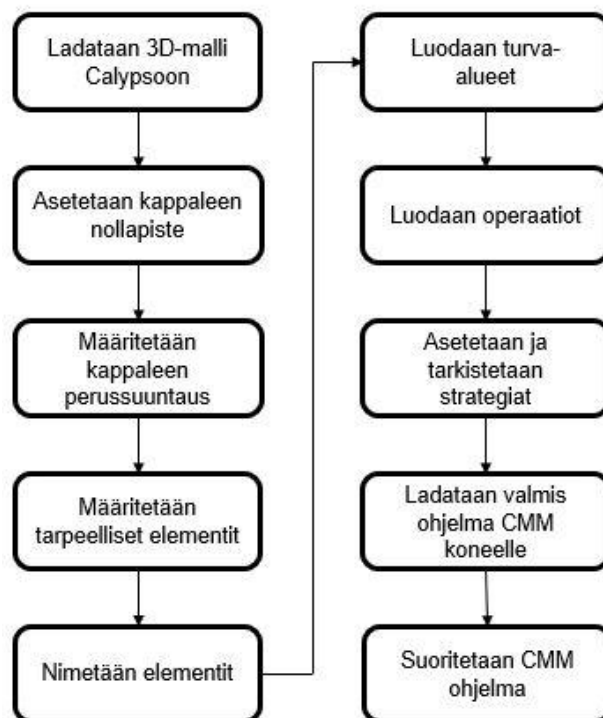
Suunniteltujen työkaluratojen jälkeen tarvitsee varmistaa, että niiden avulla saadaan koneistettua kappale ilman törmäyksiä tai työkalujen hajoamisia. Työkalujen verifiointissa käydään läpi työkalujen ja työkaluratojen muodostamia operaatioita, varmistaen niiden soveltuvuuden CAM-prosessin loppuvaiheessa tapahtuvaan koneistukseen. Verifiointin jälkeen saatava tieto tarvitsee muuttaa soveltuvaksi CNC-koneelle, jonka johdosta käytetään postprosessoria. Postproessorista saatua NC-koodia tarvitsee simuloida, jotta sen toimivuus koneistuksessa voidaan varmistaa. Mikäli postprosessoitu tieto ei ole oikeanlaista, työkaluratoihin tehdään muutoksia jotka simuloidaan uudelleen postproessorin muuntamisen jälkeen. NC-koodin simuloimisessa ja työkalujen verifiointissa on se ero, että NC-koodin simulointi tutkii tarkasti CNC-koneelle menevää ohjelmaa kun taas verifiointissa pyritään nopeuttamaan postprosessoria ennalta ehkäisemällä mahdollisia virheilmoituksia.

Kun simuloitu NC-koodi on hyväksytty, lähetetään se tuotannonohjausjärjestelmään, joka on Sandvikin tapauksessa Fastemsin ohjelma MMS (lyhenne sanoista Manufacturing Management Software). NC-koodi ei siirry ohjelmaan automaattisesti, vaan käyttäjän tulee ladata se manuaalisesti kansioista, johon se on tallennettu simuloinnin jälkeen. Tässä ohjelmassa voidaan valita koneistuksessa käytettävä työstökone ja siirtää sille koneistusta varten tuotettu NC-koodi. Kaikki NC-koodit ovat tallessa MMS-ohjelmassa, sillä työstökoneilla on rajallinen muisti ja niissä on optimaalista olla vain ajankohtaiset koneistusohjelmat. Kun tarkkuusvalmistuksen työstökone ja koneistuksen ajankohta on valittu, siirtyy NC-koodi koneistusta varten DNC:n kautta työstökoneelle.

Ennen kappaleen koneistusta työstökoneelle ladattu NC-koodi validoidaan vielä viimeisen kerran koneistajan toimesta. Validoinnissa käytetään Siemensin käyttöliittymää SINUMERIK. Tämän käyttöliittymän avulla koneistaja pystyy varmistumaan NC-koodin toiminnallisuudesta ja jos joitain pieniä korjauksia tarvitsee tehdä, ne voi suorittaa käyttöliittymän avulla. Lopullinen NC-koodi tallennetaan dokumentointina MMS-ohjelmaan.

muiden koodien joukkoon. Kun koneistuksen ajankohta koittaa, CNC-koneelle on toimitettu koneistusohjelman lisäksi raaka-aineet ja käytettävät työkalut. Näitä kutsutaan koneistuksessa käytettäviksi resursseiksi. Koneistuksen aikana työstökone suorittaa operaatiot parametreineen järjestyksen mukaisesti ja lopputuloksena on koneistettu kappale. Kun kappale on valmis, lopullinen suunnittelun ja koneistuksen tieto tallennetaan Teamcenteriin, jossa sitä voi tarkastella MRL:n kautta.

Viimeisenä vaiheena CAM-prosessissa on koneistetun kappaleen tarkistus, jossa varmistetaan kappaleen kelvollisuus ennen kuin se lähetetään eteenpäin. Tässä vaiheessa käytetään koordinaattimittauskoneita tutkimaan kappaleen tärkeimpiä dimensioita, toleransseja ja pinnankarheuksia. Koordinaattimittauskoneiden operaatiot suoritetaan CMM ohjelman avulla. CMM ohjelma voidaan laatia jo mallinnetun kappaleen geometrian ja työpiirustuksen teon jälkeen mutta luominen ja käyttäminen tapahtuu vasta prosessin loppuvaiheessa kun ollaan saatu valmistettua ensimmäisiä prototyyppejä. CMM ohjelma luodaan mittauskoneen yhteydessä tulevalla omalla ohjelmistolla, joka pystyy lukemaan NX:n suunnasta tulevia tiedostoja. Kuva 33 esittää Sandvikin CMM ohjelman tekemiseen liittyvät vaiheet.



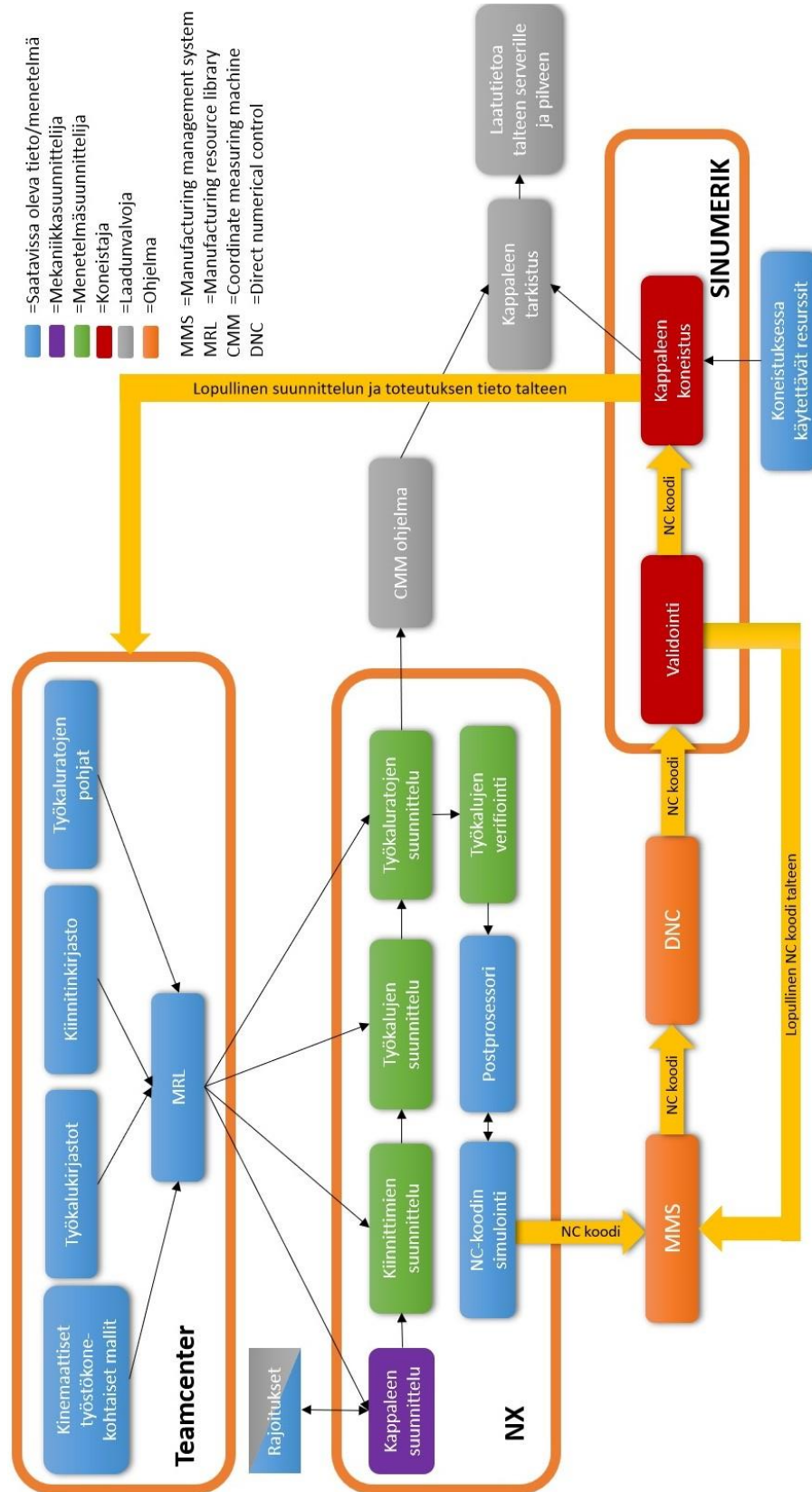
Kuva 33. CMM ohjelman vaiheet

Kun kappaleen malli ladataan käytettävään Zeissin Calypso-nimiseen ohjelmaan, alkaa CMM ohjelman luominen asettamalla tarkasteltavalle kappaleelle nollapiste sekä määrittämällä kappaleen asento CMM mittauksessa. Tätä kutsutaan perussuuntaukseksi ja

siinä lukitaan tarvittavilta osin koordinaatisto sekä pyöriminen akselien ympäri. Tämän seurauksena CMM kone ymmärtää tutkittavan kappaleen sijainnin ja geometrian mittauksen aikana. Seuraavana askeleena käyttäjä määrittää, yhdistelee ja nimeää mittauksissa käytettäviä elementtejä. Kun elementit on määritetty, on tarpeellista luoda turva-alue. Turva-alue rajoittaa muun muassa mittapään pikaliikkeitä ja poistaa törmäysriskit. Seuraavana voidaan luoda mittausoperaatiot, joissa hyödynnetään aikaisemmin määritettyjä elementtejä. Ennen lopullisten operaatioiden tekemistä käyttäjä keskustelee työpiirustuksen tehneen suunnittelijan kanssa tarpeellisista mittauksista. Operaatioiden avulla suoritetaan kaikki tarpeelliset mittaukset valmiissa ohjelmassa.

Viimeisenä vaiheena ennen valmista CMM ohjelmaa on operaatioiden käyttämien mittausstrategioiden tarkistaminen. Strategioilla tarkoitetaan esimerkiksi pinnalta mitattavien pisteiden lukumäärää ja sitä, onko mittapää koko ajan kiinni pinnassa vai tapahtuuko kosketus vain mitattavan pisteen kohdalla. Calypsoon on asetettu oletukseksi yleisimmin käytettyjä strategioita mutta ne eivät ole yleispäteviä. Kun kaikkien operaatioiden käyttämät strategiat on asetettu sopiviksi, CMM ohjelma on valmis. Tämän jälkeen valmis ohjelma ladetaan CMM koneelle ja se suorittaa määritetyn mittauksen. Kun CMM ohjelmalla tehty laaduntarkastus on suoritettu, tieto tallennetaan serverille ja pilveen,

josta sitä voidaan tarkastella myöhemmin. Kuva 34 esittää tässä luvussa selitetyn tarkkuusvalmistukseen liittyvän CAM-prosessin kokonaisuudessaan.



Kuva 34. CAM-prosessi kokonaisuudessaan

Ylläolevassa kuvassa on havainnollistamisen helpottamiseksi merkitty väreillä eri vaiheissa vastuussa olevat työntekijät, käytettävät ohjelmat sekä jos kyseessä on saatavissa oleva tieto tai menetelmä. Kohdassa rajoitukset tietoa saadaan monelta eri taholta, joten se on havainnollistettu usean eri värin käytöllä. Kuvassa käytettävät lyhenteet on kirjoitettu auki englanniksi. Jos lyhenteiden merkitykset eivät ole vielä selvillä, ne voi lukea luvun 2.1 alaluvuista. Ohjelmissa Teamcenter, NX ja SINUMERIK tapahtuu useita eri vaiheita, joten sisältyvät vaiheet on havainnollistettu käyttämällä oranssia kehystä niiden ympärillä. Ohjelmissa MMS ja DNC ei tapahdu erikseen havainnollistettavia vaiheita, joten niiden asettelu on jätetty samanlaiseksi kuin yksittäisissä vaiheissa. Kuvassa harmaalla värillä tarkoitettuihin laadunvalvojiin on sisällytetty lujuuslaskijat sekä CMM ohjelman tekijät yksinkertaistamiseksi. Kuvassa näkyvien mustien ja keltaisten nuolien avulla havainnollistetaan vaiheiden järjestystä ja prosessin kulkua. Rajoituksien ja kappaleen suunnittelun sekä NC-koodin simuloinnin ja postprossessorin välillä on kaksisuuntainen nuoli. Tämä tarkoittaa vuorovaikutusta ja iterointia CAM-prosessin vaiheessa syntyvän oikeanlaisen lopputuloksen saamiseksi.

Edellisen kuvan perusteella voidaan huomata tarkkuusvalmistukseen liittyvässä CAM-prosessissa olevan lukuisia vaiheita ja että tietoa liikkuu paljon eri vaiheiden välillä. Prosessin entistä syvemmän ymmärtämisen mahdollistamiseksi on tarpeen kertoa prosessin vaiheiden välillä siirtyvistä tietomuodoista. Tietomuodoilla tarkoitetaan tässä yhteydessä vaiheisiin tulevia ja niistä lähteviä tiedostotyyppisiä. Taulukko 4 havainnollistaa jokaisen CAM-prosessin vaiheen kohdalla tulevan ja lähtevän tiedostotyyppin sekä kertoo, miten vaiheen tuoma lisäarvo näkyy. Vaiheiden tuomat yksityiskohtaiset lisäarvot on kerrottu aikaisemmin tässä luvussa olevassa vaihekohtaisessa selostuksessa.

Taulukko 4. CAM-prosessin vaihekohtaiset tiedostotyyppit ja lisäarvon näkyminen

Vaiheen nimi	Tulevat tiedostotyyppit	Lähtevät tiedostotyyppit	Miten lisäarvo näkyy
Kinemaattinen työstökonekohtainen malli	-	.asm	Tiedostona
Työkalukirjastot	.stp .xml	.stp .xml	Tiedostona
Kiinnitinkirjasto	.pdf .stp	.pdf .stp	Tiedostona
Työkaluratojen pohjat	.prt	.prt	Tiedostona
MRL	.stp .xml .asm	.stp .xml .asm	Kirjastona
Rajoitukset	-	.pdf .dwg .mat .wbpz	Tiedostona, Hyväksyntänä
Kappaleen suunnittelu	.pdf .mat .wbpz	.prt .dwg .jt	Tiedostona
Kiinnittimien suunnittelu	-	.stp	Tiedostona
Työkalujen suunnittelu	.stp .xml	.stp	Tiedostona
Työkaluratojen suunnittelu	.prt .dwg .stp	.prt	Tiedostona
Työkalujen verifiointi	.prt	.prt	Tiedostona, Hyväksyntänä

CMM ohjelma	.prt .dwg	.xml	Tiedostona
Postproessori	.prt	.mpf	Koodina
NC koodin simulointi	.mpf	.mpf	Koodina, Hyväksyntänä
MMS	.mpf	.mpf	Datan siirtona
DNC	.mpf	.mpf	Datan siirtona
Validointi	.mpf	.mpf	Hyväksyntänä
Koneistuksessa käytettävät resurssit	.stp .pdf	.stp .pdf	Materiaalina
Kappaleen koneistus	.mpf	.pdf	Kappaleena
Kappaleen tarkistus	.xml .pdf	.pdf	Hyväksyntänä
Laatutietoa talteen serverille ja pilveen	.pdf	.pdf	Tiedostona

Ylläolevassa taulukossa käytettäviä tiedostotyyppisiä ovat .stp .xml .pdf .prt .asm .dwg .jt .mat .wbpz ja .mpf. Kappaleiden, työkalujen, kokoonpanojen ja piirustusten sisältäviä tiedostomuotoja ovat .stp .prt .asm .jt ja .dwg. Excelillä tehdyt tiedostot ovat .xml tyyppissä. Lujuuslaskennassa käytettävien Ansysin ja Matlabin tuottamat tiedostomuodot ovat .wbpz ja .mat. Luettavia tiedostoja ja raportteja tallennetaan .pdf muotoon. Työstökoneelle siirtyvä NC-koodi tallennetaan ja liikutetaan vaiheiden välillä .mpf muodossa. Kohdissa joissa tiedostomuodon tyyppinä on viiva, tarkoitetaan ettei vaiheessa ole pohjatietoa. Vaiheesta lähtevä tiedostotyyppi siis luodaan tyhjästä.

Lisäarvon tuottamisen havainnoimiseen on erilaisia tunnistamistapoja. Näiden pohjalta voidaan päätellä, minkä tyyppisestä vaiheesta on kyse. Suurin osa lisäarvon näkymisestä ilmenee tiedostona. Tällä tarkoitetaan sitä että kyseisissä vaiheissa on joko luotu dataa tyhjästä tai sitten jalostettu vaiheisiin tulevaa dataa. Hyväksyntä lisäarvon näkymistyyppinä tarkoittaa vaiheeseen tulevan datan tarkistamista ja verifiointia. Näissä vaiheissa varmistetaan tiedon oikeellisuus ja näin ollen ennalta ehkäistään myöhemmissä vaiheissa mahdollisesti syntyviä ongelmia. Lisäarvo voi näkyä myös koodina, jolla tarkoitetaan työstökoneelle kulkeutuvaa NC-koodia. Koodin sijaan voisi käyttää ilmaisua ”tiedostona”, mutta tällä halutaan korostaa kyseessä olevan NC-koodi.

Lisäarvon näkyminen kirjastona tarkoittaa paikkaa, jossa erilaisiin tiedostoihin pääsee käsiksi. Jos vaiheissa ei suoriteta muokkausta tulevan ja lähtevän tiedon välillä, on lisäarvon tyyppinä mainittu datan siirto. Lisäarvon näkyminen ei ole pelkästään digitaalista, sillä lopputuloksena on kuitenkin fyysinen tuote. Kappaleen koneistuksen ollessa ajan-kohtainen, koneelle tuodaan käytettävät resurssit. Tässä tapauksessa lisäarvo näkyy

materiaalina. Koneistuksen jälkeen syntyy valmis lopputulos, jolloin lisäarvon näkymistapana on käytetty ilmaisua kappale.

Käytettävän CAM-prosessin ongelmakohtia esiintyi osittain jo prosessikuvausta tehdessä mutta selkeän käsityksen saamiseksi ongelmakohdat potentiaalisine ratkaisuehdotukseen on syytä käydä läpi. Ensimmäisenä puutteena tarkkuusvalmistukseen liittyvän prosessin suunnittelussa on selkeän linjauksen puuttuminen, jonka johdosta sitä ei voitu laittaa omaksi laatikokseen CAM-prosessin kuvaan. Tällä hetkellä ei ole yksittäisiä vastuuhenkilöitä prosessisuunnitelmien tekemiseen. Suunnittelu voisi onnistua Teamcenterin Manufacturing Process Plannerin avulla jos sinne saataisiin ensin luotua tehdasta ja sen resursseja vastaava ympäristö. Kyseinen Teamcenterin kokonaisuus on tarkoitettu erilaisten prosessien havainnoimiseen sekä läpiviennin auttamiseen. Manufacturing Process Plannerin avulla saisi yksiselitteisen ja todennäköisesti tehokkaamman toimintatavan sekä voisi seurata prosessien kulkua reaaliaikaisesti.

Toisena ongelmakohtana on uusien työkalujen lisääminen järjestelmään. Kaikilla työkalumyyjillä ei ole sähköisiä katalogeja, joista saisi suoraan lisättyä tarvittavat parametrit MRL:ään. Näissä tapauksissa käyttäjän tulee syöttää sinne kaikki tiedot manuaalisesti ja tähän kuluu paljon aikaa. Työkalumyyjiä on informoitu asiasta ja he tekevät työtä sähköistämisen parantamiseksi, jotta olemassa oleva ongelma saataisiin ratkaistua. Ongelmaksi voidaan myös tunnistaa tiedon tallentaminen useaan eri paikkaan: Muun muassa lujuuslaskennasta saadut raportit ja NC-koodit voitaisiin tallentaa Teamcenteriin niihin liittyvän nimikkeen alle. Mitä enemmän yhdessä paikassa on tietoa, sitä vähemmän tietoa tarvitsee etsiä.

Toleransseihin liittyvä ongelmakohta voidaan myös luokitella tiedon jakamiseksi. Kappaleen suunnittelun jälkeen toleranssit saattavat muuttua työkaluratojen mallinnusvaiheessa ja postprosessoidun NC-koodin ennen koneistusta olevassa validoinnissa. Suunnittelija ei välttämättä saa tietoa muutoksista ja tämä voi aiheuttaa systemaattisen muutoksen tarpeen synnyn tulevaisuudessa suoritettavissa muissa prosesseissa. Joka tapauksessa kappaleen laatutarkistus suoritetaan kappaleen suunnittelun mukaan tehdyllä CMM ohjelmalla mutta toleranssien muuttumattomuus prosessin aikana olisi toivottavaa. Ongelma saadaan ratkaistua luomalla sääntö kaikkien tehtyjen muutoksien informoisesta prosessin aikaisemmissa vaiheissa oleville työntekijöille, jotka ovat tekemisissä muutoksia vaatineen tiedon parissa. Tiedon muokkaamiseen liittyen lisäksi NC-

koodiin tehtävät muutokset olisi syytä tehdä suoraan työkalurataohjelmaan, jotta Teamcenterissä olisi varmasti toimivia työkaluratoja myöhempää hyödyntämistä varten.

Tehostamista vaativana kokonaisuutena on kappaleen geometrian ja työkaluratojen mallintaminen: Kyseisissä kohdissa kuluu eniten aikaa CAM-prosesseissa jonka johdosta niiden nopeuttamisella saataisiin eniten hyötyä. Tähän haetaan toimivaa tapaa työn myöhemmässä vaiheessa tehtävällä ohjelmointirajapintojen hyödyntämisellä, jonka seurauksena voitaisiin hyväksikäyttää aikaisempaa dataa sekä automatisoida toistettavia operaatioita yksinkertaista ja tehokasta apuvälinettä hyödyntäen. Lisäksi tehostusta vaatii NC-koodin siirtäminen tuotannonohjausjärjestelmään: Tällä hetkellä käyttäjä joutuu siirtämään tarkastetun koodin väliaikaiseen paikkaan, josta se ladataan manuaalisesti järjestelmään. Rajapinnan avulla saatavalla automatisoinnilla voitaisiin varmistaa tiedon siirtyminen oikeaan paikkaan ja nopeuttaa prosessia.

Viimeisenä ongelmakohtana on tarkkuusvalmistukseen liittyvän kappaleen CMM ohjelman tekeminen: Tällä hetkellä ohjelma luodaan manuaalisesti Zeissin Calypsolla ilman tuotteen ja valmistuksen informaatiota (englanniksi Product and Manufacturing Information, PMI) sisältävää PMI-mitoitusta, mutta prosessia voitaisiin nopeuttaa automaattisella generoinnilla suoraan mallinnusvaiheessa muodostetusta datasta. Automatisointiin liittyviä ongelmakohtia ja vaatimuksia käsitellään luvuissa 6.3 ja 7. Esimerkitapaukseen liittyvän CAM-prosessin kartoittamisen jälkeen voidaan alkaa selvittämään potentiaalisia keinoja, joiden avulla saadaan vastattua työssä käsiteltäviin tutkimuskysymyksiin.

6. TOTEUTUS

Diplomityön eräänä tavoitteena on tutkia ja soveltaa erilaisia keinoja, joiden avulla voidaan ratkaista olemassa olevia ongelmia. Kun työhön liittyvä kirjallisuudesta löytyvä teoria ja tarkkuusvalmistukseen liittyvän esimerkkitapauksen kautta määritelty yrityksen nykytila ovat selvillä, voidaan ruveta keräämään laajempaa kuvaa CAM-prosessien ongelmakohdista ja kehittämään ratkaisumalleja myös niihin liittyen. Ensimmäisessä kokonaisuudessa kartoitetaan tilannetta eri yrityksissä ja sitä kautta laajennetaan tietoisuutta olemassa olevista toimintatavoista ja tunnistetuista kehityskohteista CAM-prosesseihin liittyen. Yrityksiä lähestyttäessä ei käytetty erillistä lomaketta tiedonkeruuta varten vaan kyseessä oli vapaamuotoisia tapaamisia, joissa kirjoittaja tiedusteli yrityksen tilanteesta ja vastavuoroisesti kertoi työn teettävän yrityksen tilanteesta. Näin ollen molemmat osapuolet saivat lisätietoa CAM-prosesseihin liittyen ja tämä lisäsi diplomityön potentiaalia tarjota ratkaisumalleja laajempaan käyttöön. Yritykset valikoitiin siten, että heidän toimintansa olisi jossain määrin samanlaista esimerkkitapaukseen nähden.

Useamman yrityksen tilanteen selvityksen sekä kirjallisuudesta kerätyn tiedon jälkeen voidaan perehtyä ensimmäiseen potentiaaliseen ratkaisutapaan, joka on ohjelmointirajapintojen hyödyntäminen. Tämän avulla pyritään nopeuttamaan CAD ja CAM suunnittelun parissa työskentelevien henkilöiden työtä ja mahdollistamaan tehokkaampaa tiedonsiirtoa CAM-prosessien eri vaiheiden välillä. Ohjelmointirajapintojen avulla tutkitaan myös sitä, onko CMM ohjelma mahdollista luoda automaattisesti CAD suunnittelussa tuotettua dataa hyväksikäyttäen. Työssä tuotetut ratkaisumallit on luotu siten, ettei niiden käyttämisen aloittamiseen vaadita syvällistä ymmärtämistä ohjelmoinnista vaan apuvälineiden luomisessa pystyy hyödyntämään suunnitteluvaiheissa käytettäviä ohjelmia, joiden ominaisuuksiin kuuluu ohjelmointikielien hyödyntäminen.

Ohjelmointirajapintojen hyödyntämisen yhteydessä tutkitaan koordinaattimittausohjelman tekemisessä olevia puutteita sekä kehitetään potentiaalisia ratkaisumalleja niiden automaattiseen generoimiseen. Työlle asetettuna tavoitteena oli selvittää, millaisia vaatimuksia automaattisella mittausohjelman luomisella on ja alaluvussa 6.3 kerrotaan yleiskuva tilanteesta. Ratkaisumallit käyvät sekä tarkkuusvalmistukseen että yleisen tason

valmistukseen. Kolmanteen tutkimuskysymykseen tehdään yhteenveto tarvittavista minimivaatimuksista luvussa 7.

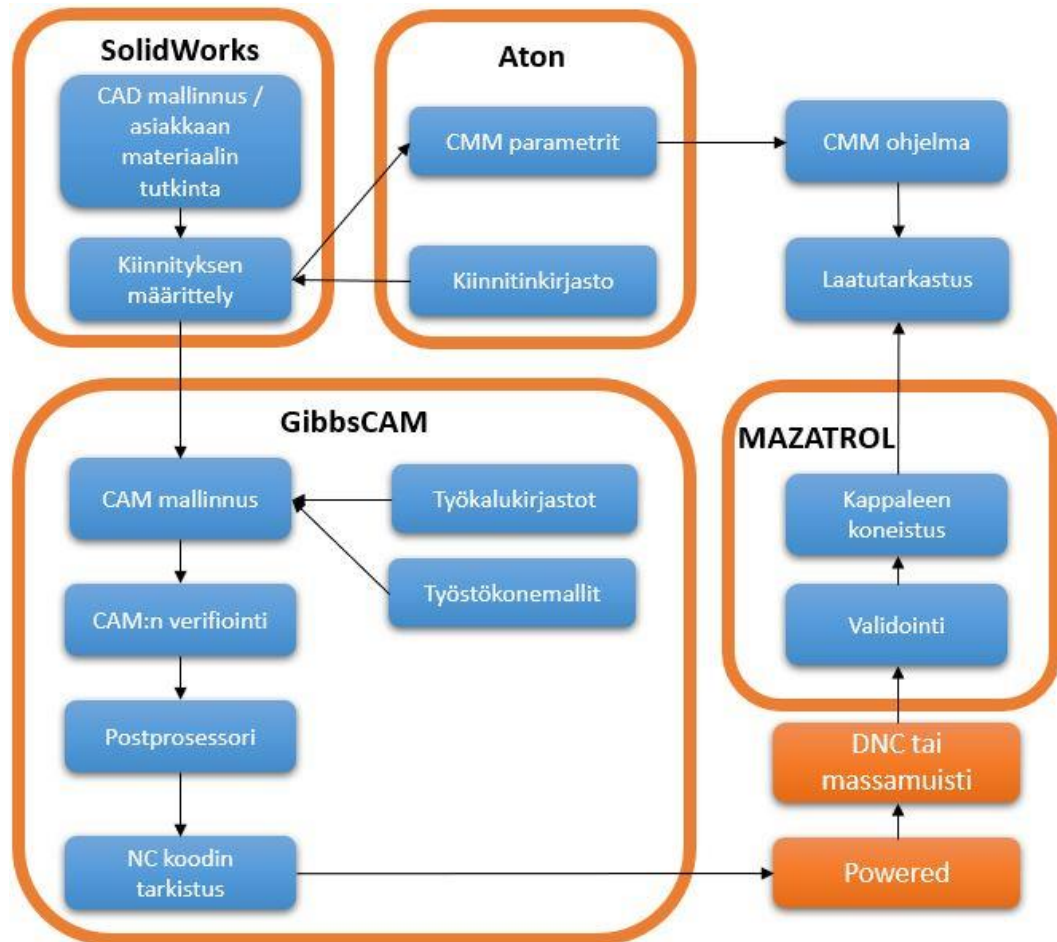
6.1 CAM-prosesseja ja kehityskohteita muissa yrityksissä

Tasowheel Systems Oy on Tampereella toimiva teknologiayritys, jonka pääasiallisena toimialueena on valmistaa paperikoneisiin liittyviä toimilaitteita sekä järjestelmiä. Koneistettavat tarkkuusvalmistusta vaativat kappaleet ovat sekä pyörähdyssymmetrisiä että prismaattisia, jonka johdosta Sandvikin tapaukseen verrattaessa voidaan havaita yhteneväisyyksiä. Itse kappaleiden käyttötarkoitukset tai toimiala eivät ole samankaltaisia Sandvikin kanssa mutta CAM-prosessia tutkittaessa päämääränä on niin ikään valmis lopputuote, joka täyttää asetetut vaatimukset. Tasowheelin prosessi alkaa CAD mallinnuksesta, jonka aikana luodaan kappaleen geometria. Käytettävänä ohjelmassa on tässä vaiheessa SolidWorks. Geometriaa mallinnettaessa ollaan yhteydessä muun muassa koneistajaan, jotta oikeanlaiset parametrit saataisiin mallinnettua kerralla oikein. Mikäli kyseessä on alihankintana tehtävä kappale asiakkaalle, vaiheen nimi on asiakkaan materiaalin tutkinta. SolidWorks on käytössä myös mallinnetun kappaleen kiinnityksen määrittelyssä.

Tasowheelin kappaleiden kiinnityksen määrittelyssä käytetään hyväksi kiinnitinkirjastoa, joka löytyy tuotetiedon hallintaohjelmasta Aton. Tämän lisäksi ohjelmaa käytetään säilyttämään CMM parametreja, jotka voidaan luoda kappaleen geometrian ja kiinnityksen ollessa määriteltynä. Kyseiset CMM parametrit sisältävät geometrian ja kiinnityksen lisäksi kappaleen nollapisteen määrittelyn, perussuuntauksen sekä turva-alueet. CMM parametrien avulla suoritetaan koordinaattimittaus käyttäen Zeissin mittauskonetta. CAM mallinnuksessa Tasowheel käyttää ohjelmaa GibbsCAM, jonka lähtötietona on SolidWorks-ohjelmasta tulevat geometrian ja kiinnityksen määrittelyt. GibbsCAM:n avulla luodaan ensimmäisenä työkaluradat. Työkaluratojen määrittelyssä käytetään ohjelman sisällä olevia työkalukirjastoja sekä työstökonemalleja, jotta kappale saadaan valmistettua oikeiden resurssien avulla.

Tasowheelin CAM mallinnuksen ollessa valmis, työkaluradat verifioidaan, muutetaan NC-koodiksi postproessorin avulla sekä tarkistetaan tuotettu NC-koodi simuloimalla sitä GibbsCAM:n avulla. Kun valmistusratoja vastaava NC-koodi on valmis, lähetetään se toiminnanohjausjärjestelmään nimeltä Powered. Kyseisestä järjestelmästä tieto siirtyy koneistuskeskukselle joko DNC:n kautta tai toisinaan massamuistina, esimerkiksi USB-tikun, avulla. Mazak:n valmistavalle koneistuskeskukselle saapuessa NC-koodi validoi-

daan, jonka jälkeen kappale koneistetaan mallinnettuun muotoon. Validoinnissa käytetään Mazak:n käyttöliittymää MAZATROL. Kun koneistettu kappale on valmis, suoritetaan laatu tarkastus CMM ohjelman avulla. CMM ohjelman teossa hyödynnetään Atonissa olevia CMM parametreja ja ohjelmointi hoidetaan Zeissin Calypson avulla. Laatu tarkastuksen jälkeen tuote on valmis. Kuva 35 esittää Tasowheelin CAM-prosessia. Oranssilla värillä tarkoitetaan käytettäviä ohjelmistoja ja sinisellä puolestaan prosessin vaiheita.



Kuva 35. Tasowheelin prosessi

Tasowheelin CAM-prosessiin liittyviä ongelmia ja kehityskohteita kysyttäessä ensimmäisenä ja todennäköisesti tärkeimpänä asiana on käytettävien ohjelmien yhteensovittaminen. SolidWorks, Aton, GibbsCAM ja loput ohjelmat ovat kaikki eri palveluntarjoajilta ja tästä johtuen esimerkiksi uuden SolidWorks version tullessa Aton ei välttämättä tue sitä ja syntyy ongelmia. Tämä voitaisiin ratkaista joko siirtymällä käyttämään useaa saman tarjoajan ohjelmaa tai sitten varmistamalla sujuva toimiminen käyttämällä sellaisia tarjoajia, jotka pystyvät reagoimaan toistensa toimintaan nopeallakin varoajalla. Tämän lisäksi tietoa on monessa eri paikassa, jonka johdosta sitä on vaikeampi ylläpitää ja löytää

verrattuna siihen, että tieto olisi vain yhdessä paikassa, esimerkiksi Atonissa. Ongelmaksi tunnistettiin myös työkaluhallinta työstökoneilla, joissa tarvitsee lisätä ja poistaa paletin työkaluja kappalekohtaisesti. Kaikilla työkalutoimittajilla ei ole sähköisiä malleja, joten työkalut pitää määritellä järjestelmään toisinaan käsin.

Tasowheelin työkaluhallinnan kanssa on myös kehitettävää GibbsCAM:n käytön osin, sillä ohjelmassa ei pysty tilaamaan työkohtaisia työkaluja. Tämä tarkoittaa ilmeisesti sitä, että kun käyttäjä on asettanut työstössä käytettävät työkalut, niiden tilaaminen koneistusta varten ei onnistu GibbsCAM:n avulla. Jos ohjelman avulla saataisiin informoitua työkalutoimittajaa, ei tulisi sellaisia tilanteita, ettei työkalua löydy yrityksestä koneistuksen ollessa ajankohtainen. Tunnistettuna kehityskohteena on myös CMM ohjelmien tekeminen. Nykyisessä tilanteessa yritys ei hyödynnä PMI tietoa, jonka avulla säästettäisiin aikaa mittausohjelman teossa. Yrityksellä on halukkuutta siirtyä PMI:n käyttämiseen mutta SolidWorks-toimittajalta ei saa juurikaan tukea aihepiiriin liittyen. Tämä saataisiin korjattua joko opettelemalla PMI:n käyttö itse, osallistumalla sopivaan koulutukseen tai käyttämällä sellaista toimittajaa, joka tarjoaa tukea PMI:n käyttämiseen.

Tasowheelin työntekijät tunnistivat myös yrityksen ulkopuolella olevia yleisen tason ongelmia: CAM ohjelman käyttäjältä vaaditaan korkean tason osaamista, jotta ohjelmien tekeminen ja korjaaminen olisi sujuvaa. Tällaisia henkilöitä ei ole koulutettu kovin paljoa ja pullonkaulana olevan perehtymättömän työntekijän opettaminen kestää noin vuoden, jonka johdosta hyötyjä voi joutua odottamaan suhteellisen kauan. CAM käyttäjällä tulisi myös olla tuoretta tietoa työkalujen ja kiinnityksen suunnittelemisesta, jotta pystytään hyödyntämään olemassa olevia resursseja parhaalla mahdollisella tavalla. Viimeisenä ongelmakohtana on kiinnityksen ja koneen tukeva asettaminen. Jos ei käytetä tarpeeksi tukevia komponentteja, työstöterät eivät kestä alun perin suunniteltua aikaa sekä työstöprosessia ei voida pitää vakaana. Tasowheel tunnistaa siis lukuisia tärkeitä kehityskohteita ja pyrkii tulevaisuudessa poistamaan ne hyödyntäen saatavilla olevia resursseja.

Salon konepaja on sopimusvalmistuskonepaja, jossa työskentelee noin 40 työntekijää. Yrityksen tuotannosta yli puolet menee ulkomaalaisille asiakkaille ja valmistettavia tuotteita ovat sekä prismaattiset että pyörähdyssymmetriset kappaleet. Salon Konepajan ydinosaamista ovat suuret ja vaativan geometrian omaavat osat, joissa käytetään tarkkoja toleransseja. Osien koneistaminen, hitsaaminen ja kokoonpaneminen kuuluvat yrityksen palveluihin. Sandvikiin verrattuna yrityksen toimintaperiaate on erilainen mutta tarkasteltaessa puhtaasti CAM-prosessia, voidaan huomata samankaltaisuuksia. Sand-

vikilla on yhteyksiä Salon Konepajaan asiakkaana olemisen kautta. Tästä johtuen tarkkuusvalmistus on vaatimuksena osalle kappaleista. Salon Konepajan konekanta on monipuolinen ja valmistusta voidaan tehdä viisiakseliseen koneistukseen saakka.

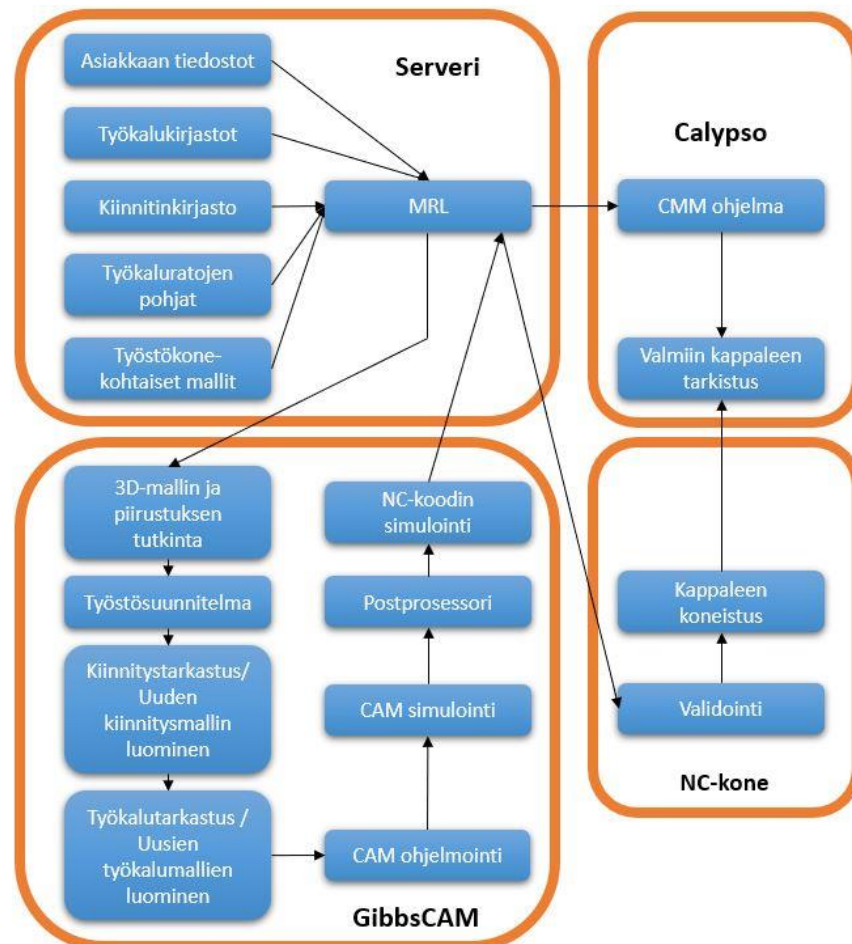
Salon Konepajan CAM-prosessi alkaa asiakkaan lähettämän materiaalin tutkimisesta. Materiaaliin kuuluu työpiirustus ja 3D-malli. Materiaalin tutkimisen yhteydessä tarkastellaan konepajan omien resurssien tilannetta, jotka löytyvät serverillä olevasta MRL:stä. Resurssikirjastosta MRL löytyy kaikki aikaisemmin tallennettu materiaali asiakkaiden tiedostoihin ja tehtaan omiin resursseihin liittyen. Kun tarkastelu on saatu päätökseen, aletaan suunnittelemaan kappaleen valmistusta. Tässä vaiheessa tehdään alustava työstösuunnitelma, jonka pohjalta tarvittavat resurssit voidaan tunnistaa ja varmistaa niiden saatavuus työstön ollessa ajankohtainen. Salon Konepajan alihankkijastatuksesta johtuen yrityksellä on paljon dokumentoitua materiaalia aikaisemmin toteutetuista tilauksista. Tätä voidaan käyttää hyväksi prosessin seuraavissa vaiheissa.

Salon Konepajan CAM-prosessi jatkuu kiinnityksen ja työkalujen varmistamisella ja luomisella. Kiinnityksen luomisessa tutkitaan ensin, löytyykö yritykseltä jo valmiiksi toimivaa ratkaisua tilanteeseen. Jos näin ei ole niin kiinnitys suunnitellaan itse esimerkiksi GibbsCAM ohjelman avulla. Sama periaate on käytössä myös työkalumallien luomisessa. Kun asiakkaan materiaali, kiinnitys ja työkalumallit on käyty läpi, CAM ohjelmointi voidaan suorittaa. Tässä yhteydessä tehdään lopulliset määrytykset käytettäville työkaluille ja suunnitellaan lopputulokseksi valmistettavan kappaleen työstö. CAM ohjelmoinnin jälkeen työstöradat simuloidaan ja suoritetaan törmäystarkastelu. Postprosessoinnin jälkeen NC-koodi vielä tarkastetaan ennen koodin hyväksymistä eteenpäin lähetettäväksi.

Simuloitu ja hyväksytty NC-koodi sekä työkaluradat tallennetaan serverillä olevaan MRL:ään. Kappaleen koneistuksen ollessa ajankohtainen koneistaja lataa NC-koneelle MRL:ssä olevan NC-koodin. Tässä kohtaa koneistaja tekee viimeisen tarkastelun ja validoi koneistuksessa käytettävän lopullisen NC-koodin. Validoinnin valmistuttua NC-koodi tallennetaan koneelle ja viimeisin versio myös MRL:ään tulevaisuuden tarkastelua ja mahdollista hyödyntämistä varten. Koneistuksen seurauksena saadaan lopputulok-

sena oleva valmis kappale, jolle tarvitsee tehdä laatutarkastus ennen asiakkaalle lähettämistä. Laatutarkastusta varten kappaleesta on tehty kappaleen geometrian tietoa hyödyntäen CMM ohjelma, jonka avulla laatutarkastus suoritetaan

Salon Konepajan CMM ohjelman kaikki lähtötiedot löytyvät MRL:stä. CMM ohjelmointi tehdään 3D-mallin ja työpiirustuksen avulla Zeissin ohjelmassa Calypso. Jos kappaleesta on käytettävissä asiakkaalta tullut 3D-malli, CMM ohjelma voidaan luoda ennen koneistamista. Jos alkutilanteessa on vain työpiirustus, CMM ohjelma luodaan siinä vaiheessa kun tuotannossa saadaan koneistettua valmiita kappaleita. Kun laatutarkastus on suoritettu Zeissin mittauskoneella niin Salon Konepajan CAM-prosessin viimeisenä asiana on tallentaa laatutarkastuksesta saatu tieto MRL:ään. Kuva 36 esittää Salon Konepajan CAM-prosessia. Kuvassa tarkoitetaan oranssilla värillä käytettäviä ohjelmistoja ja sinisellä prosessin vaiheita.



Kuva 36. Salon Konepajan prosessi

Salon Konepajan tavoitteena on kehittää jatkuvasti omaa osaamistaan ja ratkaista olemassa olevia ongelmakohtia. Aikaisempien haastatteluiden tapaan asiakailta saatava

data ei ole aina automaattisesti riittävää. Tämän johdosta syntyy lisäkustannuksia ja virheen mahdollisuus kasvaa. Työkaluvalmistajilta osittain puuttuva sähköinen työkaludata on myös Salon Konepajan eräänä ongelmakohtana. Sandvikin tapaan yrityksessä tehtävässä postprosessoidun NC-koodin muokkaamisessa uusi tieto tallentuu vain koneella olevaan ohjelmaan sen sijaan että muokkaukset näkyisivät myös mallinnusohjelmalla tehdyissä työkaluradoissa. Tämän johdosta mallinnusohjelmalla avattavassa, aikaisemmin tehdyssä ohjelmassa virheet ovat vielä olemassa ja ohjelma ei todennäköisesti toimi halutulla tavalla myöskään uusia työkaluratoja mallinnettaessa. Edellä mainitusta syystä johtuen korjaukset on syytä tehdä suoraan CAM ohjelmalla ja postprosessoida uudelleen. Jos muutokset on tehty NC-koodiin, täytyy ne päivittää myös CAM ohjelmaan.

Salon Konepajalla erilaisten ohjelmien käyttämisen osaamisessa halutaan päästä entistä korkeammalle tasolle. Tavoitellulla osaamisella tarkoitetaan tarpeeksi monen työntekijän valmiuksia käyttää tarvittavia ohjelmia monipuolisesti. Tällä ennaltaehkäistään sitä, että jos yksi työntekijä vaihtaa työpaikkaa tai sairastuu niin osaaminen ei katoa yrityksestä vaan joku toinen työntekijä pystyy suorittamaan poissaolevan henkilön tehtäviä. Kouluttaminen vie paljon aikaa mutta pitkällä aikavälillä se on kannattavaa. Oikeanlainen koulutus nähdään tarpeelliseksi tehdä yrityksessä valmistettavien tuotteiden avulla, jotta koulutuksessa olevat henkilöt oppisivat kaikki tarvittavat toimenpiteet niiden käyttämiseen. Tämän lisäksi huomionarvoisena asiana pidetään koulutuksessa olevan materiaalin harjoitteluun ja sisäistämiseen kuluva aika opetuksen päätyttyä. Jos koulutusmateriaalia ei pääse harjoittelemaan heti koulutuksen jälkeen niin koulutuksesta ei saada toivottua hyötyä irti asioiden unohtamisen seurauksena. Työkaluratojen mallinnuksessa olevat kappaleet ovat järjestelmällisesti haastavia ja tarkkuutta vaativia, joten GibbsCAM:n käyttöaste tarvitsee olla monipuolisella tasolla.

Salon Konepajalla ollaan tietoisia PMI:n käyttämisen mahdollisuuksista ja yritys on panostanut PMI tietoa hyödyntäviin ohjelmiin ja niiden uusimpiin versioihin. Asiakkailta tulevaan dataan ei ole vielä sisällytetty PMI tietoa mutta erilaisia testejä on käynnissä. Toleranssialueiden tulkitseminen PMI tiedon avulla saattaa tuoda haasteita: Esimerkkinä tulkitsemisen ongelmasta on esikoneistuksen suunnittelu, jossa tavoitteena on pysyä mahdollisimman lähellä toleranssialueen ylärajaa. Kysymyksenä on se, onnistuuko CAM ohjelmointi siten ettei koneistusstrategioissa pyritä aina liikkumaan toleranssialueen keskikohdassa. GibbsCAM:n toimittajalta saa kattavasti tietoa ohjelman toiminnoista, joten

mahdollisesti PMI tietoa voidaan käyttää tulevaisuudessa niin korkealla tasolla, että kappaleen geometriaan aletaan sisällyttämään PMI.

Salon Konepajassa CMM ohjelmat tehdään Zeissin Calypson avulla ja tämänhetkinen tilanne on sama kuin Sandvikilla. Potentiaali automaattisesti generoitavaan ohjelmaan on havaittu ja Calypsolle on hankittu uusi versiopäivitys, jonka avulla markkinoijien mukaan saadaan luotua mittausohjelma 3D-mallista napin painalluksella PMI tietoa hyödyntäen. Salon Konepajalla ei ole vielä omakohtaista kokemusta tämän ominaisuuden käyttämisestä sillä versiopäivityksen käyttöönotto on marraskuussa 2019. Yrityksessä tehdään jatkuvasti erilaisia toimenpiteitä oman toimintansa tehostamiseen ja ongelmat sekä kehityskohteet on tunnistettu hyvällä tasolla.

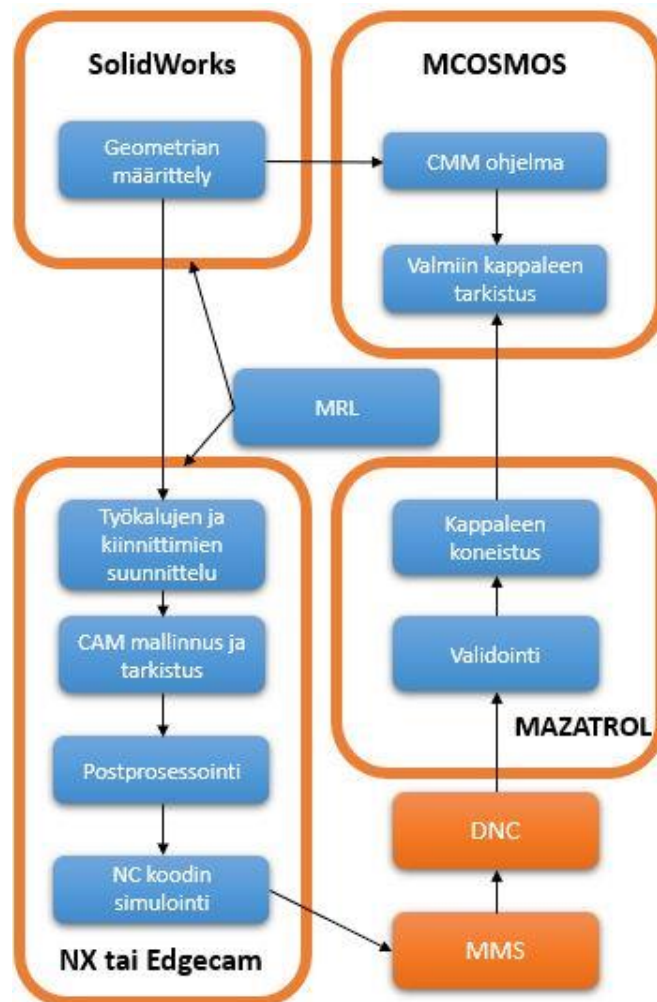
Nomet on Pirkanmaalla toimiva konepaja, joka valmistaa alihankkijana moninaisia tuotteita useaan suomalaiseen yritykseen. Osa tuotteista vaatii tarkkuusvalmistusta, joten tarkkuusalueissa on yhteneväisyyksiä Sandvikiin. Alihankkijastatuksen roolista johtuen yrityksen toimintamalli eroaa työssä käytettävästä Sandvikin esimerkkitapauksesta. CAM-prosessissa on kuitenkin sama päämäärä esimerkkitapauksen kanssa, jolloin saadaan käsitys samankaltaisesta prosessista hieman erilaisesta perspektiivistä. Päämääränä on siis valmistaa vaatimukset täyttävä lopputulos käyttäen yrityksessä olevaa toimintamallia CAM-prosessin toteuttamisessa.

Nometin CAM-prosessi alkaa asiakkaalta tulevan työpiirustuksen ja mahdollisen 3D-mallin tulkinnasta, jota voidaan kuvata geometrian määrittelyksi. Tässä vaiheessa vastaanotettu materiaali avataan ohjelmalla SolidWorks ja varmistetaan että kyseinen tuote voidaan valmistaa olemassa olevilla resursseilla. Tätä varten Käytetään serverillä olevaa resurssikirjastoa MRL, josta löytyy tiedot kaikista yrityksen käytössä olevista resursseista. Mikäli kappaleesta on saatu vain työpiirustus, 3D-malli luodaan tässä vaiheessa. MRL:n resursseja hyväksikäytetään myös kiinnityksen suunnittelussa. Geometrian määrittelyn jälkeen siirrytään käyttämään CAM ohjelmaa. Jos aikaisemmin ei ole määritelty kyseiseen tapaukseen soveltuvaa kiinnitystä niin sellainen suunnitellaan joko itse tai käyttämällä alihankkijaa. Yrityksessä on käytössä sekä NX että Edgecam.

Toisena CAM ohjelmassa tehtävänä operaationa on työkalujen määrittelemine, jossa hyödynnetään MRL:n sisältämiä resursseja. MRL on myös osana CAM mallinnusta, sillä joissain tapauksissa on jo olemassa soveltuvat työkaluradat. Tämän seurauksena prosessin läpivieminen nopeutuu sillä ei tarvitse suunnitella uutta tietoa. CAM mallinnuksen jälkeen tuotos tarkastetaan ja lähetetään postprosessoitavaksi. Ennen kuin NC-koodi

voidaan lähettää eteenpäin, on tarvetta tehdä postprosessoidun koodin simulointi. Kuvatut vaiheet ovat yhtenevät käytettävästä CAM ohjelmasta riippumatta. Seuraavana vaiheena NC-koodi tuotannonohjausjärjestelmästä DNC:n kautta Mazakin koneistuskeskukselle. Koneistuskeskuksen käyttöliittymänä on MAZATROL ja sen avulla koneistaja pystyy validoimaan NC-koodin lopullisesti ennen koneistamista.

Koneistamisen jälkeen on vuorossa laatutarkastus. Kappaleesta tehdään mittausohjelma siinä vaiheessa kun ensimmäisiä kappaleita saadaan koneistettua. Mittausohjelman teko vaatii kappaleen geometrian ja käytettävänä ohjelmistona on Mitutoyon MCOSMOS. PMI tietoa ei hyödynnetä yrityksessä mutta sen potentiaali on tiedostettu. Mittausohjelman toteuttamisen jälkeen CAM-prosessi on suoritettu, jolloin materiaalia voidaan tallentaa serverille ja lähettää asiakkaalle. Kuva 37 Esittää Nometin prosessia. Muiden kuvien tapaan oranssi väri tarkoittaa ohjelmistoa ja sininen prosessin vaiheita.



Kuva 37. Nometin prosessi

Nomet pyrkii myös jatkuvasti kehittämään omaa toimintaansa. Tämän seurauksena voidaan tunnistaa tämänhetkisiä ongelmakohtia ja kehityskohteita prosessin toteuttamiseen

liittyen. Ensimmäisenä huomiona on asiakkailta saatava data tilattuun tuotteeseen liittyen: Tällä hetkellä vastaanotetaan joko pelkästään työpiirustuksia tai 3D-malleja työpiirustuksineen. Kehitettävänä asiana olisi vastaanottaa PMI tiedot sisältäviä malleja, joiden avulla prosessi saataisiin vietyä tehokkaammin läpi. Tietoisuus PMI:n sisältävien mallien tietojen hyödyntämisestä työkaluratoja ja operaatioita luodessa on myös kehittämistä vaativa kokonaisuus. Koska yrityksessä käytetään CAM sovelluksia NX ja Edgecam, olisi luotava sellaiset postprocessorit jotka muodostavat varmasti oikeanlaisen NC-koodin CAM sovelluksesta riippumatta.

Yrityksessä on tunnistettu tiedon tallentamiseen ja siirtämiseen liittyvä ongelma on tällä hetkellä se, ettei ennen koneistamista muokattua NC-koodia tallenneta kuin koneistuskeskukselle. Ideaalitapauksessa muokattu NC-koodi tallentuisi automaattisesti koneistuskeskuksen sijaan kirjastoon, jossa sijaitsevat kaikki NC-koodit. Toisena tiedon siirtämiseen yhdistettynä huomiona on asiakkaille lähetettävä data. Tällä hetkellä osalle asiakkaista välitettävä tieto hoituu sähköpostitse vain yhteyshenkilön kautta, jolloin asiakkaan puolelta voi tulla vaikeuksia käsitellä dataa tehokkaasti. Toimivampi malli olisi käyttää rajapintaa, jonka avulla data saataisiin lähetettyä suoraan asiakkaan järjestelmiin. Tässä tapauksessa asiakkaan puolelta dataa voidaan tutkia kaikkien käyttöoikeuksien omaavien työntekijöiden tahoilta ja jokainen pystyy tarkastelemaan ajan tasalla olevaa dataa.

Nometilla on tällä hetkellä CAD-puolen suunnittelussa SolidWorks mutta tulevaisuutta silmälläpitäen koetaan hyödylliseksi opetella myös NX:n CAD-puolen käyttämistä. NX:n CAD ja CAM osa-alueiden käyttö varmistaisi integroituvuuden prosessin edetessä sekä edesauttaisi vuorovaikutusta asiakkaiden kanssa, joilla on käytössä NX. Alihankkijana toimiminen tarkoittaa monien eri työkalujen käyttämistä: tästä syystä aikaa säästettäisiin paljon jos työkalutoimittajien tarjoamaa tietoa voitaisiin käyttää suoraan. Tällä hetkellä joitain parametreja ja geometrioita joudutaan luomaan omaan järjestelmään ja tähän kuuluu paljon aikaa. CMM ohjelmien luomisessa ei tällä hetkellä käytetä hyväksi PMI tietoa, joten ohjelmointi on hidasta. Jos PMI tieto saataisiin onnistuneesti käyttöön, mittausohjelmien teossa säästettäisiin aikaa.

Nometilla tunnistettuna kehityskohteena on myös laaduntarkastuksessa käytettävät välineet. CMM koneen ja ohjelman tekijänä toimii Mitutoyo. Kun CMM koneen ja ohjelman parissa työskentelevältä henkilöltä kysyttiin Zeissin ja Mitutoyon eroavaisuuksia, saatiin seuraavia vastauksia: Mitutoyon ohjelman tekeminen MCOSMOS ohjelmalla on käytännössä koodaamista rivi riviltä kun taas Zeissin Calypsossa ohjelman tekeminen on sujuvaa eri työkalujen avulla. Tämän johdosta Mitutoyon käyttäminen vaatii käyttäjältä paljon

enemmän lähtötietoja. Mitutoyon avulla tehdyn ohjelman muokkaaminen on hidasta ja muokattava ohjelma tarvitsee generoida kokonaan uudelleen kun taas Calypsossa muokkaaminen on paljon nopeampaa ja generoinnin voi tehdä vain halutulle osalle ohjelmaa.

Graafinen puoli on haastatellun henkilön mukaan Mitutoyon heikkous sillä virheellisen tiedon paikantaminen ja esittäminen koodista on vaikeaa. CMM koneita verrattaessa Mitutoyon kalibroinnissa kestää kauemmin mutta niitä ei tarvitse huoltaa yhtä usein kuin Zeissin koneita. Yhteenvetona aihepiiristä tuli se, että Zeissin konsepti toimisi paremmin mutta Mitutoyo on käytössä koska sillä saadaan suoritettua operaatiot huomattavasti halvemmalla hinnalla. Tämän eri yrityksiä käsittelevän alaluvun pohjalta voitiin tunnistaa paljon samankaltaisia ongelmia Sandvikin kanssa ja lisäksi saatiin tietoisuuteen lukuisia muita olemassa olevia ongelmia CAM-prosesseihin liittyen. Luvussa 7 kootaan tämän alaluvun asioita ja luodaan näin ollen yhteenveto Sandvikin ja muiden yritysten ongelmakohteista sekä kehityskohteista.

6.2 Ohjelmointirajapintojen hyödyntäminen

Ohjelmointirajapintojen, tässä työssä NX Openin, hyödyntämisen tutkimisessa käytetään ohjelmointikieltä Python. Kirjoittajalla ei ole juurikaan aikaisempaa kokemusta ohjelmoinnista ja Pythonin käyttämisen aloittamiselle ei ole korkeaa lähtötasoa, joten valinta on sopiva tähän työhön liittyen. Ohjelmointirajapinnat soveltuvat sekä tarkkuusvalmistukseen että muun tyyppiseen valmistukseen. NX Openin käyttäminen on mahdollista ilman aikaisempaa ohjelmointikokemusta, sillä NX:n avulla voi luoda koodia Journal-työkalun avulla. Journalin hyödyntäminen on käytännössä käyttäjän tekemien operaatioiden tallentamista, jonka jälkeen niitä voi toistaa suorittamalla tallennettuja tiedostoja. Journaliin pääsee käsiksi joko ylävalikosta Tools-Play Journal tai jos näkyvissä on Developer-ylävalikko niin sen aktivoimalla saa näkyviin kaikki Journalin työkalut. Journaliin pääsee myös käsiksi seuraamalla polkua **Menu-Tools-Journal**. Makrojen ja Journalien toimintaperiaatteet ovat samankaltaiset mutta tässä työssä käytetään Journalia, sillä makrojen muodostama koodi on vaikeaselkoista, eikä ole tarvetta tallentaa jokaista hiiren ja näppäimistön painallusta.

Kun käyttäjä haluaa luoda uuden Journalin, on ensiksi tarpeellista määritellä tallennettavan tiedoston tyyppi ja käytettävä ohjelmointikieli. Tämä onnistuu seuraamalla polkua Menu-Preferences-User Interface ja avautuvasta ikkunasta navigoimalla ikkunaan Tools-Journal-Journal Language. Siellä näkyvät kaikki tallennuskelpoiset ohjelmointikie-

let, joista tässä tapauksessa valitaan Python. Kun oikea tiedostotyyppi on valittu niin voidaan aloittaa tallentaminen navigoimalla Journalin työkaluihin ja klikkaamalla kohtaa "Record". Avautuvasta ikkunasta valitaan tallennuspaikka ja määritellään ohjelman nimi myöhempää käyttämistä varten. Tallentaessa ohjelmaa käyttäjä voi pysäyttää komentojen tallentamisen napista "Pause Recording" siksi aikaa kun esimerkiksi kokeilee seuraavien operaatioiden tekemistä. Pysäyttämisen avulla lopullista koodia saadaan lyhyemmäksi kun turhia kokeiluita ei kirjata ylös. Haluttujen operaatioiden tallentamisen jälkeen käyttäjä voi painaa nappia "Stop Recording", joka lopettaa tallentamisen ja tallentaa koodin asetettuun tiedostoon.

Tallennettua Journalia pääsee katsomaan ja muokkaamaan tekstieditorin avulla. Tässä työssä käytettävä tekstieditori on Notepad++ mutta Journalit saa avattua myös esimerkiksi ohjelmilla Notepad ja Visual Studio. Käyttäjä voi tiedoston avattuaan tutkia ohjelman rakennetta ja halutessaan poistaa, lisätä ja muokata koodirivejä ohjelman optimoimiseksi. Käyttäjä voi tarkastella erilaisten komentojen ominaisuuksia ja saada näin ollen käsityksen niiden toiminnoista: Luvussa 2.3 kerrotaan, mistä ja miten erilaisten komentojen ominaisuuksia voidaan tutkia kirjallisuudessa. Journalin käytön havainnollistamiseksi esitetään pallon muodostamiseen käytettävä koodi, jonka esittää kuva 38. Kuvan Journalia voi käyttää silloin kuin NX:ssä on avattuna mallinnustila.

```

import math
import NXOpen
import NXOpen.Features
import NXOpen.GeometricUtilities
def main() :

    theSession = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work

    sphereBuilder1 = workPart.Features.CreateSphereBuilder(NXOpen.Features.Sphere.Null)

    piste = NXOpen.Point3d(60.0, 250.0, 0.0)
    origo = workPart.Points.CreatePoint(piste)
    sphereBuilder1.CenterPoint = origo
    sphereBuilder1.Diameter.RightHandSide = "80"
    sphereBuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Create

    sphereBuilder1.Commit()
    sphereBuilder1.Destroy()

if __name__ == '__main__':
    main()

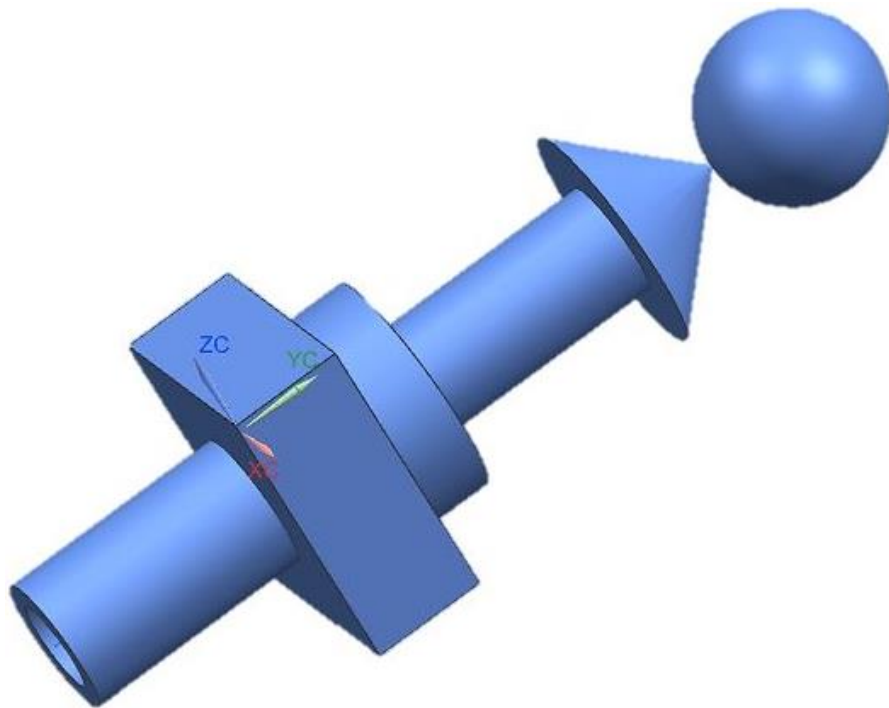
```

Kuva 38. Pallon muodostamiseen tarvittava koodi

Ohjelman aluksi tarvitsee tuoda käytettävät kirjastot, jotka näkyvät sanojen "import" jälkeen. Näiden avulla saadaan käyttöön kaikki tarvittavat työkalut ja rajoitetaan tuominen vain käytettäviin kirjastoihin. Kirjastojen tuomisen jälkeen alkaa itse komentojen suorittaminen pääohjelmassa "main". Ensimmäisenä luodaan istunto ja työkappale, jolla luodaan lähtökohdat ohjelman myöhemmässä vaiheessa tapahtuvalle muokkaamiselle.

Seuraavaksi kutsutaan pallon muodostavaa komentoa, joka on merkattu lyhyemmäksi nimellä sphereBuilder1. Kyseinen komento tarvitsee ensimmäiseksi pisteen, joka määrittellään pallon origoksi. Tämän jälkeen pallolle asetetaan halkaisija, joka on 80 millimetriä. Millimetri on määritelty yksiköksi NX:n mallinnustilaan oletuksena ja tämä ei näy koodissa. Viimeisenä määrittämisnä tässä koodissa on se, että pallo asetetaan Boolean tyyppillä "Create" olevan oma, uusi kokonaisuutensa.

Kun kaikki tarvittavat parametrit on määritetty, pallo voidaan muodostaa komennolla "sphereBuilder1.Commit". Muistin vapauttamiseksi pallon muodostava kokonaisuus tuhoaan. Tuhoaminen ei vaikuta luotuun palloon, sillä se tuhoaa pelkästään pallon muodostamisessa käytetyn komentosarjan. Ylläolevan kuvan kaksi viimeistä riviä tarkoittavat sitä, että ohjelma pysyy aktiivisessa istunnossa ja näyttää suoritettujen komentojen. Ylläolevassa kuvassa on esitetty pelkästään ohjelman tarvitsevia komentoja pallon määrittämiseen liittyen. Koodista on siis poistettu kaikki selkeyttämistä heikentävät käyttämättömät parametrit. Journalin avulla on tehty onnistuneesti myös erityyppisiä muotoja, jotka esittää kuva 39.



Kuva 39. Journalin avulla tehtyjä muotoja

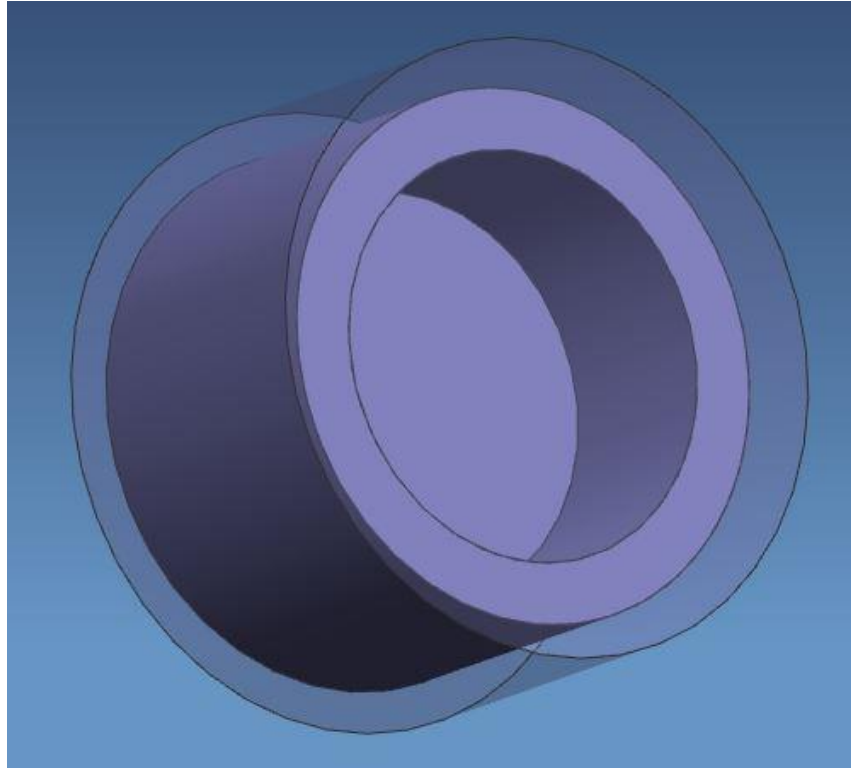
Ylläolevassa kuvassa on mallinnettu erilaisia muotoja Journalia hyväksikäyttäen. Mallinnettavia kappaleita ovat kaksi sylinteriä, kartio, kuutio, pallo ja putki. Nämä muodot on kaikki tehty riippumattomiksi toisistaan ja jokainen on tehty omilla komennollaan. Muodot määrittelevät komennot ovat jokseenkin samanlaisia kuin pallon tapauksessa, mutta

jokaisen komennon tarkat vaatimukset löytyvät Siemensin sivuilla olevasta referenssioppaasta. Linkki referenssioppaaseen löytyy luvusta 2.3. Ainoastaan putkea mallinnettaessa oli tarpeellista tehdä myös luonnos (englanniksi Sketch), jolla sijaitisi putken muotoviiva. Näiden kokeilujen seurauksena voidaan päätellä, että Journalia hyväksikäyttäen pystytään tekemään paljon samoja operaatioita kuin tyypillisessä mallintamisessa. Journalin operaatiot soveltuvat sekä tarkkuusvalmistukseen että muihin valmistusluokkiin.

Ylläoleva kuva on luotu tyhjään ympäristöön mutta potentiaalisesti esitetyn periaatteen avulla voidaan luoda mallinnettavan kappaleen rakenteen lähtökohta koodin avulla. Tässä tapauksessa ohjelmointirajapinnasta ei ole merkittävää hyötyä koska saman asian pystyy tekemään lataamalla aikaisemmin tallennetun mallin kirjastosta. Pääasiallinen hyöty tulee mallinnetun kappaleen piirteiden lisäämiseen. Tässä tapauksessa voidaan potentiaalisesti aktivoida piirteen pinta, jolle lisättävät muodot generoidaan. Ohjelmointikielen hyödyntämisellä saadaan siis lukuisia piirteitä lisättyä haluttuun paikkaan muutaman klikkauksen avulla, jolloin käyttäjän ei tarvitse mallintaa käsin samanlaisia ominaisuuksia kuin joskus aikaisemmin. **Liite A** havainnollistaa ylläolevan kuvan muodostavan koodin, joka on Python-muodossa.

NX Openin käyttämistä testattiin myös valmistuksen suunnittelun puolella. Tutkittavana kappaleena on Sylinterin muotoisesta raaka-aineesta valmistettava kappale, jossa tarvitaan erilaisia operaatioita. Ensimmäiseksi tutkittavana oleva kappale mallinnettiin CAD-puolella, jonka yhteydessä mallinnettiin myös valmistukseen tuleva aihio (englanniksi Blank). Tämän jälkeen siirrytään CAM-puolelle ylävalikoista Application-Manufacturing, missä määritellään koordinaatisto, työstettävä kappale ja aihio sekä välttäminen (englanniksi Avoidance). Tämän jälkeen voidaan aloittaa koneistuksen suunnittelu. Kuva 40 esittää mallinnetun kappaleen, joka viedään CAM-puolelle. Osittain läpinäkyvä osa esit-

tää kappaleen ympärillä olevaa aihiota, jota aletaan koneistamaan. Kappaletta ei rajoiteta tarkkuusvalmistusta vaativaksi mutta se olisi mahdollista asettamalla tarkkuusvalmistuksen alueeseen liittyvät toleranssit ja pinnankarheudet.

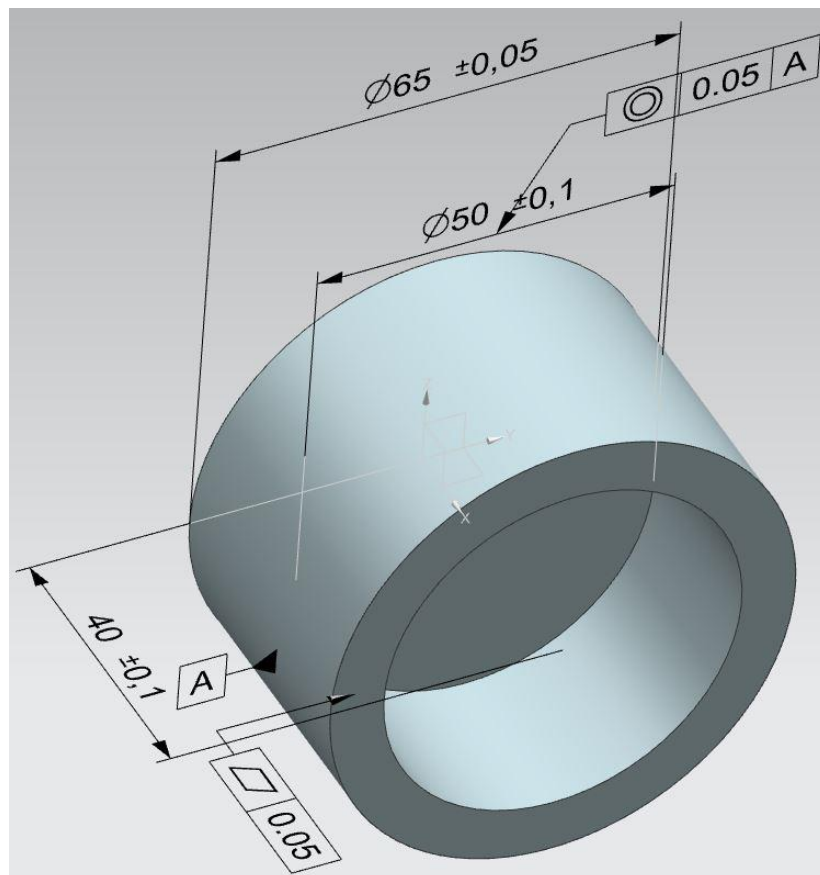


Kuva 40. CAM-puolen kappale

Koneistamiseen tarvitaan kolme eri operaatiota ja niille omat työkalunsa. Koodin yksinkertaistamiseksi käytettävät työkalut on määritelty etukäteen, jonka johdosta ne pystytään valitsemaan suoraan operaatioita luodessa. Suoritettavat operaatiot ovat pinnan jyrshintä (NX:ssä Face Milling), tasomainen jyrshintä (NX:ssä Planar Mill) sekä karkea ulkopuolisen halkaisijan sorvaus (NX:ssä Rough Turn OD). Operaatioiden saaminen Python-muotoon onnistuu Journalia käyttämällä, jonka tallennus käynnistetään ennalta määriteltyjen asioiden ollessa valmiita. Journalia tallennettaessa pyritään käyttämään mahdollisimman vähän toimintoja koodin ymmärtämisen selkeyttämiseksi ja pituuden minimoimiseksi. Journalin tallentaminen lopetetaan kun suoritettavat operaatiot on saatu tehtyä. **Liite B** havainnollistaa ylläolevan kuvan kappaleen muodostavien koneistusoperaatioiden koodin.

Viimeisenä kokonaisuutena NX Openin käyttämiseen on CMM ohjelmien luominen Journalin avulla. Kappaleen geometrian ja mitoituksen ollessa valmis, laatuinsinööri tai suunnittelija voi alkaa muodostamaan CMM ohjelmaa. Mittausohjelman teon kokonaisuuden tutkimisessa käytetään hyväksi tämän alaluvun aikaisemmassa vaiheessa mallinnetun

kappaleen geometriaa. Kuva 40 toimii siis lähtökohtana kappaleelle, jolle CMM ohjelma muodostetaan NX Openin Journal ominaisuuden avulla. Kappaleen tiedostoa muutetaan kahdella tavalla, jotta se olisi soveltuva siirrettäväksi suoraan mittausohjelman tekopuolelle, joka on nimeltään Inspection. Tähän kokonaisuuteen pääsee ylävalikoista polkua Application-Inspection pitkin. Jos halutaan luoda erillinen tiedosto Inspection puolelle niin se tehdään polkua File-New-Inspection pitkin, jonka jälkeen valitaan sopiva mallipohja. Kappaleeseen tehdyt muutokset ovat aihion poistaminen sekä tarvittavan PMI tiedon lisääminen suoraan geometriaan. Kuva 41 esittää CMM ohjelman tekemisessä hyödynnettävän kappaleen geometrian.



Kuva 41. Kappale CMM ohjelmaa varten

Kuten ylläolevasta kuvasta huomataan, kappaleen PMI on laadittu vain tutkimista vaativien piirteiden suhteen. CMM ohjelmassa tarkastetaan ulkopuolisen sylinteripinnan halkaisija sekä muoto. Tämän lisäksi sisäpuolisessa sylinteripinnassa tutkitaan halkaisija ja samakeskisyys peruselementin A suhteen. Viimeisenä tutkittavana piirteenä on ylläolevan kuvan etupuolella oleva ympyrämäinen pinta, josta tarkastetaan tasomaisuus. Kun Inspection työkalu on aktiivinen, oikea mallipohja valittu ja kappaleen PMI tiedon sisältävä geometria tuotu, voidaan mittausohjelma tehdä hyväksikäyttämällä etukäteen luotua Journalia. **Liite C** havainnollistaa Journalin avulla tehtävän CMM ohjelman koodin.

Muodostettavaa CMM ohjelmaa sekä Inspection kokonaisuutta käsitellään seuraavassa alaluvussa.

CAD , CAM ja CMM-puolen käyttämisen liittämällä NX Openiin on potentiaalia laskea merkittävästi suunnitteluun kuluva aikaa. Kun CAD-suunnittelija mallintaa kappaletta, hän voi ideaalitapauksessa hyödyntää kaikkea aikaisemmin määriteltyä informaatiota uuden tuotteen suunnittelussa. Esimerkiksi jos suunnittelija on tekemässä mallia tarkkuusvalmistusta vaativasta monimutkaisesta prismaattisesta kappaleesta, hän voi hyödyntää aikaisemmissa kappaleissa mallinnettuja muotoja mikäli ne ovat yhteneviä. Muotojen lisääminen onnistuisi suorittamalla esimerkiksi Python-muodossa olevan Journalin, joka toteuttaa tallennetun ohjelman ja luo halutun laisen lisäyksen. Toinen esimerkki voisi olla pintaan tehtävien reikien mallintaminen: Sen sijaan että käyttäjä määrittäisi reikien koot, sijainnit ja operaation, hän voisi suorittaa Journalin joka tekee nämä asiat automaattisesti jo aikaisemmin tehdyn määrittelyn mukaan.

CAM-puolella suunnittelija pystyisi ideaalitapauksessa lisäämään mallinnettuun kappaleeseen kaikki operaatiot toimivilla parametreilla Journalia hyödyntäen. Valmistuksen suunnittelu kokonaisuudessaan vain yhtä Journalia käyttämällä vaatii paljon, mutta jo yksittäisten operaatioiden lisäämisellä säästettäisiin aikaa. CAM-prosesseissa valmistusratojen mallintamiseen kuluu eniten aikaa, joten sitä tehostamalla saataisiin suurimmat hyödyt läpimenoaikaan silmälläpitäen. Jos valmistuksen suunnittelussa saadaan automatisoitua toistamista vaativat vaiheet, menetelmäsuunnittelija voi keskittyä täysin uuden tiedon mallintamiseen ja siihen liittyviin parametreihin. Journalin avulla mallinnettujen toistettavien operaatioiden tarkistaminen tulee tehdä mutta aikaa säästyy mallintamiseen kuluvan ajan minimoinnilla. CMM-puolesta kertoo seuraava alaluku.

NX Openin onnistuneessa hyödyntämisessä on kuitenkin tiettyjä ongelmia, jotka vaativat syvällistä paneutumista ohjelmointiin. Kun koodin avulla luodaan muotoja ja operaatioita, on ensisijaisen tärkeää että ne tulevat oikeille pinnoille ja oikeisiin paikkoihin. Suunnittelijan mallintaessa uudenlaisia kappaleita käy usein niin, että Journalissa olevat parametrit ovat erilaisia kuin käyttäjä toivoisi. Tämä vaatii Journalilta kykyä ymmärtää määritellyt parametrit. Määrittelyn voi hoitaa esimerkiksi aktivoimalla käytettävän pinnan ennen Journalin suorittamista tai sitten käyttämällä yhtenevää nimeämistä tarpeellisista pin-

noista vanhan ja uuden kappaleen välillä, jolloin aikaisemmin muodostettua koodia voidaan hyödyntää suoraan. Kaikista suurimpana huomioitavana asiana on siis vanhan tiedon onnistunut hyödyntäminen uudessa kappaleessa.

Kun uutta kappaletta suunnitellaan, pääasiallisena tavoitteena on halutun operaation luominen eikä oikeanlaisen nimeämisen käyttäminen. Suunnittelija voi käyttää erilaista järjestystä uusien muotojen kanssa ja tämä tuottaa eroavaisuuksia parametrien nimeämisissä. Ratkaisuna tähän ongelmaan olisi viittaamista vaativien parametrien nimeämisen muuttaminen halutuksi ennen kuin Journalia aletaan käyttämään. Valmistuksen suunnittelussa uuden kappaleen tapauksessa käytettävät resurssit ja parametrit, kuten syöttö ja pyörimisnopeus, voivat olla erilaisia kuin tallennetussa Journalissa. Tässä tapauksessa vanha Journal voidaan suorittaa ja sen jälkeen muokata sopivaksi tai sitten luoda uusi operaatio, jonka aikana se tallennetaan uudeksi Journaliksi.

Edellä mainitut ratkaisumallit pohjautuvat aikaisemmin määritellyn Journalin ominaisuuksien hyödyntämiseen mutta ohjelmoinnista paljon tietävä henkilö voi potentiaalisesti luoda Journaleita, jotka pystyvät mukautumaan käyttäjän asettamiin vaatimuksiin. Tässä tapauksessa ratkaisuna voisi olla Journalin suorittamisen alussa avautuva ikkuna, johon käyttäjä voisi syöttää esimerkiksi viitattavat pinnat, suoritettavien operaatioiden lukumäärän, koneistamisessa käytettäviä parametreja tai muita tarpeellisia tietoja. Suoritettavien operaatioiden lukumäärällä voidaan esimerkiksi määrittää CAD suunnittelussa ympyrän kaarelle tulevien reikien lukumäärä tai CAM suunnittelussa toistettavien vaiheiden lukumäärä. Kun Journal on interaktiivinen käyttäjän kanssa, tallennetun ohjelman tarvitsee vain lukea käyttäjän syöttämiä arvoja ja näin ollen nimeämisestä tai mukautumattomuudesta syntyviä hankaluuksia ei synny.

6.3 CMM ohjelmien automaattinen generointi

Luvussa 5 käytiin läpi nykyinen menetelmä CMM ohjelmien tekemiseen. Tällä hetkellä tarkkuusvalmistuksen avulla koneistettujen kappaleiden mittausohjelmien teko tapahtuu Sandvikilla manuaalisesti Zeissin Calypso-nimisellä ohjelmalla kappaleen 3D-mallia sekä työpiirustusta hyödyntäen. Calypso pystyy tunnistamaan 3D-mallissa olevan geometrian mutta työpiirustusten tulkitsemisessä ohjelmasta ei ole apua. Ennen nykyistä toimintatapaa CMM ohjelmat luotiin opettamalla koneita operaatio kerrallaan. CMM ohjelmien parissa työskentelevän henkilön mukaan nykyinen toimintatapa on noin 80% nopeampi verrattuna vanhaan toimintatapaan. Tutustuessa Calypson toimintaan huomataan sen tukevan malleja, joihin on sisällytetty tuotteen ja valmistuksen tieto (englanniksi Product and Manufacturing Information, PMI). PMI tiedon avulla käyttäjä säästää aikaa,

sillä ohjelma osaa hyödyntää automaattisesti PMI:n sisältämiä parametreja. Diplomityön toteutusvaiheessa PMI:n avulla saadaan automatisoitua operaatioiden ja elementtien määrittely Sandvikilla tapahtuneen testauksen avulla. Koko ohjelman teon automatisointia varten on syytä paneutua puutteisiin sekä käydä läpi kehityskohteita.

Suurimpana puutteena on mitattavan kappaleen nollapisteen, perussuuntauksen ja turva-alueen määrittely. Käyttäjä joutuu nykyisessä toimintamallissa asettamaan nämä joka kerta manuaalisesti. Lisäksi manuaalista työtä vaatii käytettävien mittausstrategioiden tarkastaminen. Strategioiden tapauksessa ohjelmassa on oletuksena tietynlaiset strategiat mutta niitä ei pysty soveltamaan kaikkiin kappaleisiin. Yksittäisten elementtien automaattinen muodostaminen onnistuu mutta niiden järkevä nimeäminen ja samanlaisten elementtien yhdistäminen puolestaan ei. Puutteena on myös se, ettei Calypso jostain syystä toimi automaattisesti muoto- ja sijaintitoleranssien mittauksien määrittelyssä. Viimeisenä tunnistettuna puutteena on se, ettei Calypso osaa automaattisesti tunnistaa mittaukseen haluttuja piirteitä vaan tulkitsee kaikki PMI tiedossa olevat piirteet mitattaviksi. Kompleksisissa tarkkuusvalmistetuissa prismaattisissa kappaleissa on niin paljon piirteitä, ettei kaikkien piirteiden mittaamisessa ole järkeä.

Näiden tunnistettujen puutteiden avulla voidaan laatia erilaisia kehityskohteita. Kehityskohteiden onnistuneen toteuttamisen avulla päästään ideaalitapauksessa tilanteeseen, jossa CMM ohjelmien automaattinen generoiminen olisi mahdollista. Ensimmäisenä kehityskohteena on käytettävien elementtien nimeäminen ymmärrettävään muotoon. Esimerkiksi jos kappaleessa on kaksi yhtä suurta halkaisijaa, voitaisiin toinen niistä nimetä merkinnällä "D50 oikea". Tällaisella nimeämisellä voidaan helposti identifioida siihen liittyvä elementti jos ohjelmaa tarvitsee tulkita. Nimeämisen saisi ratkaistua joko Zeissin puolelta tulevalla nimeämisspäivityksellä tai jos Zeiss asettaa automaattisen nimeämisen oletukset käyttäjän muokattaviksi. Kehityskohteena voidaan pitää myös samanlaisten elementtien ryhmittelemistä, jonka avulla myöhemmin luotava operaatio saataisiin linkattua suoraan yhtenevät elementit sisältävään ryhmään.

Eräänä kehityskohteena voidaan pitää nollapisteen, perussuuntauksen ja turva-alueen määrittelyn sisällyttämistä kappaleen tiedostoon. Tämän onnistuneella toteutuksella CMM ohjelman tekovaiheessa niitä ei tarvitsisi määrittellä manuaalisesti. Lisäksi Calypso pitäisi saada ymmärtämään operaatiot kaikissa toleranssityypeissä. Tämä vaatii tarkkuusvalmistuksen toleranssien määrittelyä erilaisin keinoin PMI mallia luodessa. Kehityskohteena on myös oikeanlaisten strategioiden määrittelyminen kappaleesta riippuen. Tähän ratkaisuna voisi olla mallipohjien luominen erilaisille kappaleryh-

mille. Yhdessä kappaleryhmässä olisi samankaltaisia kappaleita, joiden mittaaminen onnistuisi samojen strategioiden avulla. Ryhmäkohtainen mallipohja tulisi aktivoida Calypsoon kappaleesta riippuen, jolloin oikeat mittausstrategiat muuttuisivat oletuksiksi.

Viimeisenä kehityskohtena on vain tarvittavien piirteiden sisältävän PMI mallin tekeminen. Calypson luoma mittausohjelma ei tarvitse täydellistä PMI mitoitusta kappaleesta vaan riittävänä tietona on ainoastaan mitattavien suureiden informaation sisällyttäminen PMI malliin. Tämä varmistaa vain haluttujen piirteiden mittaamisen ja säästää näin ollen aikaa. Toisaalta epätäydellinen PMI malli ei poista tarvetta luoda työpiirustusta kappaleen mallinnusvaiheessa, jonka johdosta suunnittelijoiden työmäärä ei pienene. Potentiaalisenä ratkaisuna olisi myös se, että jos kyseessä on täydellinen PMI mitoitus esimerkiksi tarkkuusvalmistetusta kappaleesta, niin siihen pitäisi määrittellä erikseen vain CMM ohjelmassa tutkittavat piirteet.

CMM ohjelman automaattiseen generointiin voidaan perehtyä tutkimalla NX:n Inspection kokonaisuutta, johon viitattiin jo edellisessä alaluvussa. Tunnistettujen kehityskohteiden sekä lähtötietona olevan kappaleen määrittelyn seurauksena mittausohjelma voidaan määrittellä Journalin avulla myöhempää käyttöä varten. Ensimmäisenä tehtävänä on valita mittauksessa käytettävä kappale, CMM kone, mittapään pidike sekä mittapää. Sandvikin tarkkuusvalmistuksen esimerkkitapauksen laatutarkastuksessa käytetään mittauskoneita Zeiss Prismo, jonka malli löytyi myös kirjastosta. Pidike ja mittapää valitaan satunnaisesti, sillä kirjastossa ei ole tarkkuusvalmistuksen esimerkkitapauksen laatutarkastuksessa käytettäviä resursseja. Tämän jälkeen tarkasteltavalle kappaleelle tehdään

nollapisteen määrittely ja perussuuntaus, jonka seurauksena mittaukset voidaan toteuttaa kaikille halutuille pinnoille. Kuva 42 esittää tilanteen kappaleen sekä mittauskoneen resurssien määrittelyn jälkeen.



Kuva 42. Tilanne kappaleen ja mittauskoneen resurssien määrittelyn jälkeen

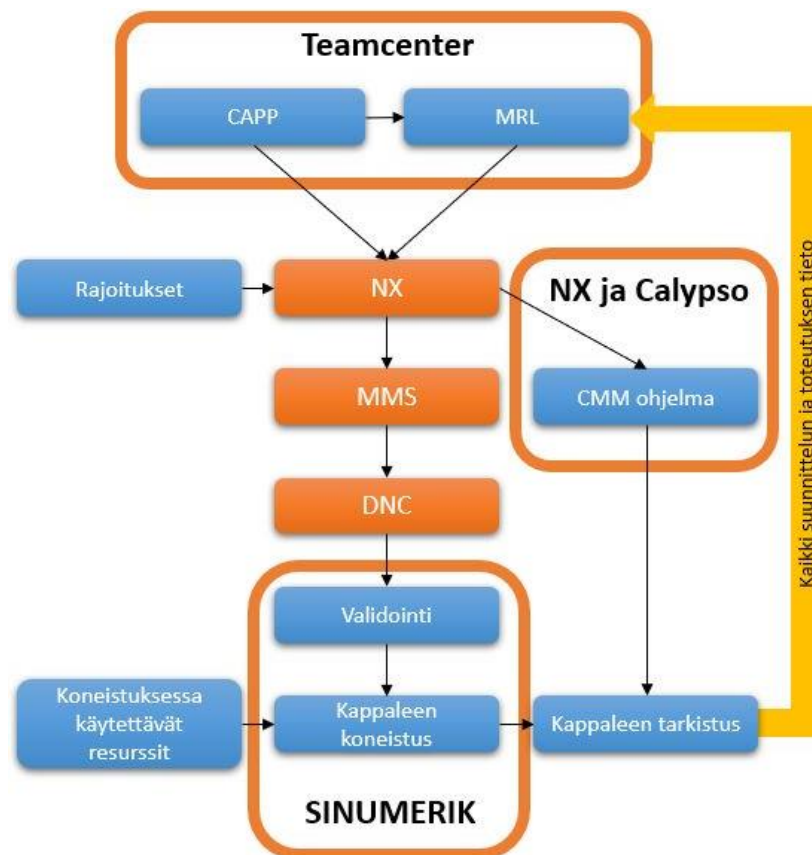
Ylläolevassa kuvassa on zoomattu tarpeeksi lähelle, jotta nähdään kappaleen perussuuntaus, pidike ja käytettävä mittapää. Kuvan ulkopuolella on CMM kone, jonka pinnalla tutkittava kappale on. Tämän jälkeen kappaleeseen linkitettyä PMI tietoa voidaan hyväksikäyttää. PMI tiedon saa lisättyä Inspection kokonaisuuteen ylävalikoista Home-Link to PMI polkua pitkin tai seuraamalla polkua Menu-Tools-Inspection Navigator-Operation-Link to PMI. Käyttäjä joutuisi manuaalisesti lisäämään kaikki tarvittavat suureet jos PMI tietoa ei olisi saatavilla, joten käytettävä tapa on paljon nopeampi ja se pystytään myös automatisoimaan Journalin avulla. Kun PMI tietoa lisätään, ohjelma kysyy käytettävää työkalua, mittapään tyyppiä sekä mittaratojen luomista. Tässä tapauksessa radat halutaan luoda ja työkalu valitaan automaattisesti NX:n toimesta. PMI tiedon onnistuneen linkityksen jälkeen NX luo valmiin mittausohjelman automaattisesti.

Kun verrataan luotua Journalia ja minimivaatimuksia toisiinsa, huomataan että turva-alueita ja mittausstrategioita ei määritellä. Kyseiset osa-alueet voidaan määritellä NX:n avulla esimerkiksi mallipohjaan. Realistisena nopean aikavälin ratkaisuna automaattisen NX:n avulla luotavaan mittausohjelmaan pääsemiseksi olisi luoda mallipohjia erilaisille

kappaleille ja eri tilanteisiin soveltuvia Journaleita. Tarkkuusvalmistus voidaan ottaa mallipohjien luomisessa huomioon. Tämän seurauksena valmiin mittausohjelman luomiseen riittäisi oikealla mallipohjalla luotu tiedosto, johon saadaan resurssit ja valmis mittausohjelma sopivaa Journalia hyväksikäyttäen. Huomioitavaa on se, että tässä tapauksessa NX:n avulla kokonaan generoitu CMM ohjelma tarvitsee muuttaa sopivaksi Zeissin mittauskonetta varten. Tähän Zeiss tarjoaa ratkaisuksi NX:ssä tehdyn CMM ohjelman DMIS (mittausstandardi, englanniksi Dimensional Measuring Interface Standard) standardin mukaisen postprosessoidun tiedon hyväksikäytön. Tässä tiedostotyyppissä Calypso kykenee potentiaalisesti hyödyntämään kaikkia NX:n avulla määriteltyjä suureita ja muodostamaan toimivan CMM ohjelman. Tätä ei ole vielä todistetusti saatu toimimaan mutta Zeiss tekee tutkimus- ja kehitystyötä aiheeseen liittyen. Diplomityössä olevaan kolmannen tutkimuskysymyksen koordinaattimittausohjelman automaattisesta generoinnista vastataan luvussa 7.

7. TULOKSET JA ANALYYSI

Ensimmäisenä tutkimuskysymyksenä pyydetään esittämään esimerkitapauksen pohjalta laadittu ideaalinen CAM-prosessi tarkkuusvalmistuksessa ja siihen liittyvät vaiheet. Suurinta osaa tutkimushetkellä olevan prosessin vaiheista ei tarvitse muuttaa mutta selkeimpänä lisäyksenä olisi kirjallisuudessa sekä alaluvussa 2.1.3 selitetyn CAPP:n lisääminen prosessikuvaukseen. Ideaalitapauksessa tarkkuusvalmistuksen prosessi olisi täysin suunniteltu alusta loppuun jo ennen sen aloittamista. CAPP linkittyy periaatteessa prosessin kaikkiin kohtiin mutta asian selkeyttämiseksi se on linkitetty kuvassa vain laatikkoihin MRL ja NX. Kuva 43 havainnollistaa esimerkitapauksen ideaalisen CAM-prosessin. Oranssilla värillä kuvataan käytettäviä ohjelmia ja sinisellä prosessissa olevia vaiheita.



Kuva 43. Esimerkitapauksen ideaalinen CAM-prosessi

Ylläolevassa kuvassa on yksinkertaistettu visualisointia koska esimerkiksi NX:n sisällä tapahtuvat vaiheet ovat täysin samat kuin nykytilan tapauksessa. MRL:n rakenteeseen tulee lisäyksenä loputkin tallennettavat tiedot, jolloin kaikki tieto löytyy samasta paikasta. Yksinkertaistuksena MRL:n muodostavat kirjastot on jätetty pois kuvasta. Tämän lisäksi

yksinkertaistettuna asiana on kaiken tiedon tallentaminen vasta kappaleen tarkistuksen jälkeen. Todellisuudessa tietoa tallennetaan sitä mukaa kun tietoa syntyy tai muokataan. Tätä tapahtuu useassa kohdassa mutta jokaisen tallennuskerran havainnollistaminen kuvassa tekisi siitä epäselvän. Tarkkuusvalmistuksen prosessin kulku on siis pääpiirteissään saman näköinen kuin nykytilassa mutta prosessissa suoritettavien vaiheiden toimintatavoissa on muutoksia. Kuva 34 esittää ylläolevassa kuvassa olevat yksinkertaistetut vaiheet mutta siitä puuttuu ideaalisessa prosessissa oleva CAPP ja CMM ohjelman tekemisessä käytettävät ohjelmistot.

MRL:n muodostavien kirjastojen tieto olisi päivitettävissä automaattisesti sen jälkeen kun tarvittavat muutokset olisivat selvillä joko CAPP:n tai suoritettavan vaiheen kautta. NX:n käyttämisessä hyödynnettäisiin ohjelmointirajapintoja mallintamisessa ja sitä kautta saataisiin nopeutettua aikaa vievempiä vaiheita prosessissa. Tieto liikkuisi myös automaattisesti vaiheiden välillä sen jälkeen kun aikaisempi vaihe on saatu valmiiksi. NX:n tapauksessa tämän voi hoitaa NX Openin avulla ja NX:n ulkopuolella joko olemassa olevien ohjelmien omilla ohjelmointirajapinnoilla tai luomalla uuden, tiedon siirtämisen mahdollistavan ohjelman. CMM ohjelmien kokonaan automaattisessa teossa tarvitaan sekä NX:ää että Calypsoa. Kolmannen tutkimuskysymyksen vastauksessa tähän paneudutaan tarkemmin mutta ylläolevaan kuvaan on lisätty tarvittavat ohjelmistot.

Luodessa ideaalista kuvaa esimerkkitapauksena olevasta tarkkuusvalmistuksen CAM-prosessista on tarvetta miettiä, voidaanko erilaisista validoinneista ja tarkastuksista luopua. Tämän avulla saataisiin poistettua prosessissa olevia vaiheita mutta tuotetun tiedon ja koneistuksen tulisi olla sillä tasolla, ettei virheitä syntyisi koskaan. Prosessissa työskentelee joka tapauksessa ihmisiä, koneistuksessa on virheen mahdollistavia muuttujia ja käytettävien työkalujen versioita päivitetään säännöllisesti. Investointina kaikkien inhimillisten virheiden, koneistuksessa tapahtuvien virheiden sekä versiohallinnasta seuraavien virheiden poistaminen on liian suuri, mikäli se on nykypäivänä edes mahdollista.

Esimerkkitaupauksen ideaalisessa CAM-prosessissa koneistajan rooli muuttuu hyödynnettäessä 5-akselista koneistusta. Kyseisessä koneistustavassa työkaluratoja ei aina voida tehdä ilman CAM-ohjelmistoa jolloin koneistajan tuoma lisäarvo tulee parhaiten esille tarvittavien muutoksien tekemisessä suoraan CAM-ohjelmistoon. Tämä vaatii CAM-ohjelmiston käyttämisen osaamista mutta osaavat koneistajat eivät joudu kuormittamaan menetelmäsuunnittelijoita pienillä muutoksilla. Esimerkkitaupauksen tarkkuusvalmistuksen ideaalinen CAM-prosessi on siis käytännössä olemassa olevien työkalujen maksimaalinen hyödyntäminen sekä sellaisten lisäysten tekeminen, joihin realistisella tasolla investoimalla saadaan merkittävää hyötyä yritykselle. Huomioitavaa on se, että

kyseinen ratkaisutapa ei ole yleispätevä kaikille CAM-prosesseille ja näin ollen on tärkeää olla tietoinen olemassa olevasta tilanteesta ennen kuin alkaa miettimään ideaalista versiota prosessista.

Toisena tutkimuskysymyksenä on selvittää olemassa olevia ongelmakohtia CAM-prosesseihin sekä miettiä niihin erilaisia ratkaisutapoja. Luvussa 5 kerrottiin Sandvikin esimerkkitapauksen prosessiin liittyviä ongelmakohtia ja niiden ratkaisuja. Alaluvussa 6.1 käytiin puolestaan läpi muissa yrityksissä olevia CAM-prosesseja, niiden ongelmakohtia sekä pääteltiin potentiaalisia ratkaisutapoja ongelmakohtiin. Selkeimpänä esittämistapana kaikista esille tulleista ongelmista ratkaisutapoinen on luoda taulukko, johon ongelmat ja niiden ratkaisutavat on kerätty. Taulukko 5 esittää CAM-prosessien ongelmakohtia ja niihin liittyviä ratkaisutapoja.

Taulukko 5. CAM-prosessien ongelmakohtia ja niihin liittyviä ratkaisutapoja

Ongelmakohta	Ratkaisutapa
CAPP:n puuttuminen	Teamcenterin Manufacturing Process Plannerin käyttöönotto
Työkalujen lisääminen MRL:ään	Työkalutoimittajat muuttavat kaikki katalogit sähköisiksi
Tiedon tallentaminen useaan eri paikkaan	Kaikki tieto tallennetaan samaan ohjelmaan, esimerkiksi Teamcenteriin
Aikaisemmin luodun tiedon, esimerkiksi toleranssien tai NC-koodien, muuttuminen prosessin aikana	Muutoksista tiedotetaan tarvittaville henkilöille ja muutokset tehdään jokaiseen niitä käsittelevään ohjelmaan
Kappaleen geometrian ja koneistuksen hidas mallintaminen	API:en, kuten NX Openin, avulla aikaisemmin luodun tiedon hyödyntäminen
Manuaalinen tiedonsiirto vaiheiden välillä	API:en avulla tiedonsiirron automatisointi
CMM ohjelman hidas luominen	CMM ohjelman luominen automatisoinnin minimivaatimuksien avulla tai saamalla tarvittavaa tukea palveluntarjoajalta
Alihankkijan asiakkaalta saatavan tuotteen datan laatu	Vastaanotetaan työpiirustuksen lisäksi aina 3D-malli mahdollisine PMI tietoineen
Alihankkijan lähettämä data asiakkaalle	API:en avulla datan tallentamisen mahdollisuus suoraan asiakkaan järjestelmään
Eri palveluntarjoajien ohjelmien toimiminen keskenään esimerkiksi yhden ohjelman version muuttuessa	Siirrytään käyttämään joko saman palveluntarjoajan ohjelmia tai varmistetaan versioiden integroituvuus palveluntarjoajilta
Tarvittavien työkalujen tilaaminen heti tarpeen syntyessä	Käyttämällä ohjelmaa, jonka avulla voidaan tilata työkaluja heti tai luomalla sellaisen ohjelman itse API:en avulla
Käytettävien ohjelmien kaikkia ominaisuuksia ei tiedetä tai ohjelmia ei osata käyttää toivotulla tasolla riittävän monen työntekijän toimesta	Koulutukset palveluntarjoajan tai kolmannen osapuolen kautta. Opettelu itsenäisesti ja mahdollisuus käyttää ohjelmien kaikkia ominaisuuksia
PMI tiedon tulkitseminen työstörajojen suunnittelussa	Tarvitsee tehdä käsin tai odottaa palveluntarjoajien ohjelmien uusia versioita
CMM ohjelman tekemisen korkea käyttökynnys ja kankeus Mitutoyon MCOSMOS ohjelman avulla	Odottaa palveluntarjoajan ohjelman parantamista päivityksillä. Vaihtaa ohjelman palveluntarjoajaa siten, että olemassa oleva mittauskone toimii myös uuden ohjelman avulla tai investoida toisen palveluntarjoajan ohjelmaan ja mittauskoneeseen.

Ylläolevan taulukon tietoja kerättäessä merkittävänä ongelmakohtana tunnistettiin olevan kappaleen geometriaan ja työkaluratojen mallintamiseen kuluva aika. Ratkaisuna tähän esiteltiin NX Openia ja sen avulla luotuja apuvälineitä, jotka hyväksikäyttävät aikaisemmin luotua dataa. Tämän seurauksena voidaan siis automatisoida toistettavia operaatioita ja nopeuttaa mallintamista. Ratkaisua pidetään potentiaalisena Sandvikin, Tasowheelin, Salon Konepajan ja Nometin mallintamisen parissa työskentelevien ihmisten mielipiteitä kysyttäessä. Tätä voidaan pitää merkittävänä tuloksena sille, että työn kautta saadaan lisättyä lukijoiden tietoisuutta ongelmanratkaisuun täysin uudenlaista apuvälinettä hyödyntäen.

Kolmantena tutkimuskysymyksenä on tarkoitus määrittää, millaisia minimivaatimuksia automaattisen CMM ohjelman mahdollistamiseksi on. Alaluvun 6.3 pohjalta voidaan muodostaa vastaus tähän kysymykseen. Ensimmäisenä minimivaatimuksena on kappaleen 3D-malli, joka sisältää PMI:n mitattavista piirteistä. Tämän lisäksi automaattiseen generointiin vaaditaan nollapisteen, perussuuntauksen ja turva-alueen määrittely osana kappaleen tiedostoa, mallipohjaa tai aikaisemmin luotua Journalia. Onnistuneen automaattisen generoinnin vaatimuksena on myös se, että ohjelma osaa määrittää oikeat mittausstrategiat CMM koneelle tulevan kappaleen perusteella. Tämä voidaan hoitaa tekemällä mallipohjia erilaisille kappaleille. Minimivaatimuksien oikeanlaisen luomisen varmistamiseksi mittauskoneesta tarvitsee olla malli, jonka avulla vaatimukset saadaan asetettua käytettävälle koneelle. Mittauskoneen malli voi olla osana mallipohjaa tai mallin tuonti voidaan sisällyttää Journalin avulla tehtyyn ohjelmaan. Mikäli määrittely tehdään ilman mittauskoneen tietojen liittämistä, riskinä on ohjelman virheellinen toteuttaminen. Minimivaatimukset ovat yleisellä tasolla yhtenevät tarkkuusvalmistuksessa sekä muiden valmistuskategorioiden tapauksissa.

Automaattisen CMM ohjelman voi tehdä ainakin kahdella eri tavalla. Ensimmäinen tapa on määritellä kaikki minimivaatimukset sekä generoida CMM ohjelma kokonaisuudessaan suunnitteluohjelmiston, esimerkiksi NX:n, avulla. Tämän jälkeen ohjelma tarvitsee muuttaa sopivaksi mittauskoneen valmistajan laitteeseen. Zeissin tuotteisiin kuuluu työkalu joka osaa tulkita NX:ssä DMIS standardin mukaan postprosessoitua CMM ohjelmaa. Toinen tapa on määritellä minimivaatimukset suunnitteluohjelmistossa ja generoida CMM ohjelma mittauskoneen valmistajan tarjoamalla ohjelmalla mallipohjan avulla. Tässä tapauksessa määrittelyt tulee olla oikeanlaisessa muodossa, jotta ne tulkitaan onnistuneesti mittausohjelman generoivassa ohjelmassa. Tähän tarvitaan todennäköisesti CMM koneen kinemaattinen malli, jota voidaan hyödyntää suunnitteluohjelmistossa

tehtävissä minimivaatimuksissa. Erityisesti tarkkuusvalmistuksessa mittausohjelman generointi pitää onnistua suunnitellusti tiukkojen vaatimusten toteutumisen tarkastamiseksi. Parhaimman tavan löytämiseksi on syytä hankkia ilmaiset testilisenssit molempien vaihtoehtojen testaamiseen, jonka aikana pystytään tekemään päätöksiä.

Neljäntenä tutkimuskysymyksenä selvitetään, millä tavoin voidaan toteuttaa sulava tiedonsiirto eri osapuolten välillä. Osapuolilla tarkoitetaan yrityksen sisäisiä työntekijöitä sekä toisen osapuolen yrityksiä, kuten alihankkijoita. Luvussa 6 selvitetään erilaisia tilanteita ja tutkitaan potentiaalisia apuvälineitä myös tiedonsiirtoon liittyen. Tärkeimpänä tunnistettuna asiana ovat erilaiset ohjelmointirajapinnat, joiden avulla tietoa voisi siirtää paikasta toiseen automaattisesti yrityksen sisällä. Tämän johdosta tieto liikkuisi seuraavalle osapuolelle heti kun se on muodostettu ja päätyisi varmasti oikeaan paikkaan. Alihankkijan näkökulmasta on toivottavaa saada tarpeeksi laadukasta dataa asiakkaalta ja mahdollisuus siirtää omia tuotoksia suoraan asiakkaan tiedonhallintajärjestelmään.

Sulava tiedonsiirto eri yritysten kesken voidaan siis toteuttaa seuraavalla tavalla: tietoa lähettävä osapuoli pystyy siirtämään tietonsa suoraan vastaanottavan osapuolen järjestelmään kuitenkin siten, ettei lähettävä tai vastaanottava osapuoli pääse käsiksi muihin järjestelmissä oleviin luottamuksellisiin tietoihin. Tämä voidaan hoitaa joko asettamalla rajoitetut oikeudet tai luomalla ohjelman, joka osaa siirtää halutun tiedon oikeaan paikkaan. Jos tieto halutaan siirrettäväksi automaattisesti, tarvitsee käyttää jälkimmäisenä esitettyä tapaa eli ohjelmointirajapintoja. Opinnäytetyön laajuutta silmälläpitäen tässä työssä vain esitetään potentiaalisimmat apuvälineet saavuttaa tutkimuskysymyksessä esitetyt asiat. Työn pohjalta lukija voi havaita tarpeen ja hankkia tarvittavat resurssit esitettyjen apuvälineiden hyödyntämiseksi.

Analyysi voidaan tehdä tutkimuskysymysten ratkaisukeinojen relevanttiuteen liittyen. Ideaalinen CAM-prosessi esimerkkitapaukseen liittyen tarjoaa monenlaisia työkaluja esittelyineen mutta ei itsessään vie vielä nykytilaa kohti kuvattua prosessia. Diplomi-työssä on kyseessä uusi tuote ja siihen liittyvää prosessia ei olla vielä toteutettu, joten minkäänlaista konkreettista vertailukohtaa ei ole samanlaiseen aikaisempaan prosessiin nähden. Ideaalitapauksessa työssä oltaisiin verrattu aikaisempaa ja kehitettyä toiminta-

tapaa toisiinsa lukuarvojen saamisen kautta mutta tässä työssä tehty tutkimus antaa paljon valmiuksia tehostaa prosessien toimintatapoja ja sillä silmällä prosessien kehittämiseen liittyvää kokonaisuutta voidaan pitää onnistuneena.

Erilaisten ongelmakohtien ja ratkaisujen kohdalla tutkimus keskittyi yhteensä neljän eri yrityksen toimintaan, jota voidaan pitää sopivana työn laajuutta silmälläpitäen. Yrityksissä pähkäiltiin samankaltaisten ongelmakohtien parissa, jonka johdosta työssä esitetyt ratkaisukeinot voivat auttaa myös useita tutkimuksen ulkopuolella toimivia yrityksiä tunnistamaan potentiaalisia työkaluja olemassa olevien ongelmien ratkaisemiseksi. Haastatellut yritykset toimivat kaikki koneistamisen parissa ja tarkkuusvalmistus on vaatimuksena useille yrityksien tuotteille. Tämän johdosta ongelmakohtien ratkaisutavat on suunnattu konepajoille, jotka käyttävät materiaalia poistavia menetelmiä lopputulokseen pääsemiseksi. Ratkaisutavat saadaan kirjallisuudessa olevasta materiaalista, yrityksen eri työntekijöiden kommentteista, sekä työssä läpikäydyistä kirjoittajan löytämistä keinoista. Diplomityössä annetaan suuntaa ratkaisukeinojen käyttämiseen mutta niitä ei viedä siihen pisteeseen, että jokin ongelma olisi todistettusti ratkaistu. Joka tapauksessa esimerkiksi NX Openin käyttäminen koetaan potentiaaliseksi kaikissa haastatelluissa yrityksissä joten apuvälineiden kartoittamisessa onnistuttiin erittäin hyvin ainakin tältä osin.

Automaattisen mittausohjelman luomisen minimivaatimuksia määriteltäessä kirjallisuudessa ei ole yksiselitteistä mallia, jota voisi suoraan hyödyntää työssä. Tämän vuoksi mittausohjelman luomisen vaiheisiin ja eri vaihtoehtojen kartoittamiseen tarvitaan useiden työntekijöiden tapaamista. Kerätty tieto mahdollistaa tarkan kuvauksen kohdeyrityksen toimintatavasta sekä käytettävistä ohjelmista sekä tarkkuusvalmistuksessa että muussa valmistuksessa. Ohjelmien toiminnan tutkimista rajoittivat Calypson omakohtaisen käytön mahdottomuus lisenssin hinnan johdosta sekä se, ettei NX:n Inspection työkalulla tehtyä mittausohjelmaa pystytä muuttamaan sopivaksi Zeissin Prismo-nimiselle koordinaattimittauskoneelle toimivuuden varmistamiseksi. Tätä varten olisi tarvittu Zeissin tarjoama työkalu, jota yrityksessä ei ole. Työssä käytetään saatavilla olevia resursseja mahdollisimman hyvin, jonka johdosta työssä esitetyt tuotoksia voidaan pitää potentiaalisena CMM ohjelman luomisen automaattiseen generointiin pääsemiseksi tulevaisuudessa.

Useiden yritysten sekä kohdeyrityksen haastattelujen kautta saadaan tunnistettua tiedonsiirtoon liittyviä ongelmakohtia. Kirjallisuudesta ongelmakohtia ei etsitä, sillä työssä halutaan keskittyä erityisesti Suomessa toimivien konepajojen tämänhetkiseen toimintaan tiedonsiirrossa olevien puutteiden sekä kehityskohtien tutkimisen kautta. Jokaisella

yrityksellä on oma kokonaisuutensa käytettävien ohjelmien suhteen mutta työssä esitellään yleisellä tasolla soveltuvaa ratkaisutapa sekä Sandvikin sisäiseen että myös eri yritysten väliseen tiedonsiirtoon. Ohjelmointirajapinnan kautta tietoa lähettävällä yrityksellä ei tarvitse olla asennettuna vastaanottajan käyttämää ohjelmistoa sillä esimerkiksi verkkoselaimen kautta tehtävällä siirrolla ainoat vaatimukset ovat verkkoselain ja tunnistautuminen. Toinen vaihtoehto on luoda yksinkertainen ohjelma, johon kirjautumalla voi lähettää tietoa internetin välityksellä ennalta määriteltyn paikkaan. Esitettyjä ratkaisuehdotuksia ei toteuteta loppuun asti mutta ne esittävät keinon edistää tiedonsiirtoa eri osapuolten välillä. Kaiken kaikkiaan tuloksien perusteella saa varteenotettavan kokoelman tarkkuusvalmistukseen sekä muun tyyppiseen valmistukseen liittyen erilaisista ratkaisuvälineistä tämänhetkisten tilanteiden tehostamisiksi sekä Sandvikilla että eri konepajayrityksissä.

Diplomityön maksimaalisen hyödyn saamiseksi työssä tehtyjä havaintoja ja ehdotuksia ankkuroidaan sekä Sandvikille että muille yrityksille järjestämällä kaksi eri tilaisuutta. Ensimmäinen tilaisuus järjestetään Sandvikin eri rooleissa oleville työntekijöille ja tilaisuuden teemana on workshop. Tilaisuuden aikana eri rooleissa olevat työntekijät pääsevät kokeilemaan työkaluja ja antamaan oman näkemyksensä niiden käytöstä. Samassa yhteydessä eri rooleissa olevat työntekijät pääsevät kertomaan toisilleen CAM-prosessiin liittyvien asioiden hyödyistä ja tämän johdosta kaikki tilaisuuteen osallistuvat henkilöt oppivat uutta ja ymmärtävät eri osa-alueiden hyödyt vuorovaikutuksen ansioista. Toisessa tilaisuudessa osallisena on eri yritysten edustajia ja tavoitteena on keskustella tällä hetkellä olevasta tilanteesta sekä tunnistetuista kehityskohteista. Tilaisuudessa voidaan sopia yhteisiä pelisääntöjä tulevaisuudelle ja järjestää erilaisia pilottiprojekteja olemassa olevien ongelmien ratkaisukeinojen testaamiseksi.

8. KESKUSTELU

Diplomityön luotettavuutta silmälläpitäen on tarpeellista käydä läpi työn rakennetta ja selvittää, kuinka hyvin työn sisältö vastaa kirjallisuudessa olevaa tietoa. Työn teorian kannalta kolme tärkeintä asiaa ovat CAM-prosessit, tarkkuusvalmistus ja 5-akselinen koneistuskeskus. Kirjallisuustutkimuksessa, toisin sanottuna luvuissa 2 ja 3, esitetään kattava tietopaketti työn aihepiireihin liittyen ja annetaan lukijalle kaikki valmiudet ymmärtää työssä ratkaistavien tutkimuskysymysten kontekstit. Tärkeimpien asioiden lisäksi luvuissa esitellään myös teoriaa liittyen työn myöhemmissä vaiheissa käsiteltäviin työkaluihin ja niiden toimintaperiaatteisiin. Kirjallisuustutkimuksessa esille tullutta tietoa aletaan käyttämään hyväksi sen jälkeen kun työhön liittyvä esimerkkitapaus on esitelty niiltä osin kuin mahdollista. Esimerkkitapaukseen liittyvä tarkkuus ja koneistusmenetelmä voidaan yhdistää suoraan kirjallisuustutkimuksessa käytyyn teoriaan.

Kohdeyrityksen nykytila esitellään ja käydään yksityiskohtaisesti läpi luvussa 5. Kirjallisuudessa käytettäviä käsitteitä ja esitettyjä CAM-prosesseja verrattaessa kohdeyrityksen prosessiin huomataan lukuisia yhteneväisyyksiä. Sandvikin prosessikuvauksen kautta lukija pystyy huomaamaan, missä kohtaa CAM-prosessia kirjallisuudessa esiteltävät käsitteet sijaitsevat ja sen, millaisia eroavaisuuksia Sandvikin prosessissa on kirjallisuudessa oleviin vastaavanlaisiin prosesseihin verrattuna. Sandvikin prosessi ei ole täysin samanlainen kirjallisuudessa olevien mallien kanssa, sillä yrityksellä on omat toimintatavat ja kyseessä on puhtaasti esimerkkitapaukseen liittyvä prosessi. Kirjallisuudesta löytyvää tietoa täydennetään ja tarkennetaan muun muassa MRL:n koostumuksesta, NX:n sisällä tapahtuvista operaatioista, CMM ohjelman muodostamisesta sekä prosessin vaiheissa syntyvistä ja käytettävistä tietotyypeistä. Luvussa esitettyä prosessia voidaan pitää luotettavana, sillä se ei ole ristiriidassa kirjallisuuden kanssa. Tämän lisäksi kirjallisuudessa olevaa tietoa laajennetaan tarkennettujen kokonaisuuksien osilta.

Luvussa 6 laajennetaan käsitystä nykytilasta ja kirjallisuudesta käymällä ensiksi haastattelemassa eri yrityksiä. Saadun tiedon perusteella kirjallisuustutkimuksessa käytetty tieto sekä kohdeyrityksen tarkkuusvalmistukseen liittyvä CAM-prosessi ovat kaikki keskenään suurilta osin yhteneviä. Erilaisten ohjelmien ja laitteiden käyttö lisäsi tietoisuutta yleisen tason ongelmista, joita ei käydä läpi kirjallisuudessa. Tämän lisäksi Sandvikilla pystytään huomaamaan, kuinka yleisiä yrityksessä ovat ongelmat ovat ja minkälaisia potentiaalisia ongelmia voi ilmentyä tulevaisuudessa CAM-prosesseihin liittyen. Luvussa käytiin

läpi myös kaksi muuta tärkeää osa-aluetta, ohjelmointirajapinnat ja CMM ohjelman automaattinen generointi, joissa paneuduttiin työssä esitettyjen tutkimuskysymyksien sekä ilmenneiden ongelmakohtien ratkaisemiseen.

Kehitysehdotus ohjelmointirajapintojen hyödyntämisestä pohjautuu kirjallisuudessa olevaan tietoon. Tiedosta saa hyvät lähtökohdat aihepiirin opetteluun sekä oikeaoppiseen hyödyntämiseen. Kirjallisuudessa oleva tieto on siis yhdenmukaista käytettävän kehitystyön kanssa ja sitä laajennetaan soveltamalla ohjelmointirajapintoja työssä käytettävien yleisen tason esimerkkien operaatioiden tehostamisessa. Ohjelmointirajapintojen käyttö nähdään potentiaalisena ratkaisukeinona kohdeyrityksessä sekä kaikissa haastatteluisissa yrityksissä joten kaikki merkit viittaavat siihen, että ohjelmointirajapintojen avulla saadaan tehostettua CAM-prosessin vaiheita. Todistetuksi tätä ei voida esittää sillä esimerkitapauksen tarkkuusvalmistukseen liittyvää prosessia ei viedä työn kirjoittamisen aikana loppuun. Kuitenkin, käsiteltyä asiaa voidaan testata sopivilla malleilla liitteissä olevien koodien avulla ja huomata, että koodien käyttäminen on paljon nopeampaa kuin manuaalinen mallintaminen.

Kehitysehdotuksena on myös automaattisen koordinaattimittausohjelman generoiminen käyttämällä esitettyjä minimivaatimuksia. Kirjallisuuteen verrattuna käytettävä tieto on yhdenmukaista ja soveltamisen avulla kirjallisuudessa olevaa tietoa laajennetaan CMM ohjelmien automaattisen generoinnin osalta. Kirjoitushetkellä automaattiseen generointiin ei ollut yksiselitteistä toimintatapaa kirjallisuudessa, joten kirjoittajan omakohtaisen perehtymisen sekä CMM ohjelmien parissa työskentelevien henkilöiden haastattelemisen kautta saadaan muodostettua potentiaalisia ratkaisumalleja. Kehitysehdotus CAPP:n lisäämisestä pohjautuu yrityksen sisällä tapahtuviin haastatteluihin sekä kirjallisuudessa olevaan materiaaliin, jonka pohjalta kirjoittaja pystyi kokeilemaan ohjelman toimintaa. Toimintaa ei esitetä työssä mutta kirjoittaja pystyi tunnistamaan potentiaalisen omakohtaisen kokeilun pohjalta.

Kaiken kaikkiaan jokainen työssä esitetty ratkaisumalli sekä tutkimuskysymysten vastaukset on tehty yhdistelemällä kirjallisuudessa olevaa tietoa omakohtaisen soveltamisen ja haastattelujen kautta saatuun tietoon. Näin ollen työssä kehitettävät toimintatavat eivät ole ristiriidassa olemassa olevan tiedon kanssa ja kehitystyötä tapahtuu. Työn tuloksia voidaan siis pitää luotettavina ja käytössä olevia resursseja voidaan todeta hyödynnettäneen parhaalla mahdollisella tavalla. Tulokset eivät rajoitu pelkästään tarkkuusvalmistukseen vaan niitä pystytään ideaalitapauksessa hyödyntämään kaikkien eri tarkkuuksien omaavien kappaleiden CAM-prosesseissa.

9. YHTEENVETO JA KEHITYSKOhteET

Tässä opinnäytetyössä perehdytään olemassa olevien CAM-prosessien rakenteisiin kirjallisuudessa, Sandvikilla sekä kolmessa muussa yrityksessä. Kirjallisuudessa ja muissa yrityksissä tietoa kerätään yleisellä tasolla kirjallisuustutkimuksen sekä haastatteluiden avulla. Sandvikilla tarkkuusvalmistukseen liittyvän esimerkkitapauksen CAM-prosessin rakenteeseen kerätään tietoa omakohtaisen tutkimisen sekä eri rooleissa toimivien työntekijöiden haastatteluiden kautta. Esimerkkitapauksena olevan kappaleen muoto on prismaattinen, koneistus suoritetaan 5-akselisella koneistuskeskuksella ja valmistus voidaan luokitella tarkkuusvalmistukseksi. CAM-prosessien ongelmakohtia kerättiin Sandvikista ja muista yrityksistä haastatteluiden sekä omakohtaisen tutkimisen avulla. Kirjallisuudesta ongelmakohtia ei etsitä sillä työssä halutaan keskittyä tutkittavissa yrityksissä olevien tämänhetkisten ongelmakohtien määrittelemiseen.

Ongelmakohtien ratkaisemiseksi tietoa kerätään kirjallisuudesta, ongelmakohtiin liittyvien ohjelmien valmistajien tahoilta sekä saatavissa olevien ohjelmien omakohtaisella käyttämisellä. Tämän lisäksi ongelmakohtien ratkaisemiseen liittyviä ideoita syntyy ongelmakohtien kartoittamisen yhteydessä olevien haastatteluiden aikana. Automaattisen koordinaattimittausohjelman luomiseen selvitetään minimivaatimukset haastattelun ja ohjelmien omakohtaisen käyttämisen kautta. Minimivaatimuksien avulla voidaan tulevaisuudessa kokeilla automaattisten CMM ohjelmien luomista ja sen seurauksena tehdä päätöksiä käyttöönotosta. Sulavaan tiedonsiirtoon eri osapuolten välillä kerätään ideoita kirjallisuudesta, eri osapuolten haastatteluilla sekä rajapintojen tutkimisella. Kaikkien ongelmakohtien ratkaisuehdotukset kerätään ylös luvussa 7 olevaan taulukkoon ja joidenkin ratkaisuehdotusten toimintaa kuvaillaan työn aikana siten, että lukijalle syntyy alustava käsitys niiden toiminnasta. Diplomityön tavoitteena on siis antaa lukijalle työkaluja olemassa olevien ongelmien ratkaisemiseksi, mutta työn rajoitetun laajuuden seurauksena ongelmia ei ratkaista kokonaisuudessaan esitettyjen työkalujen avulla. Tämän johdosta kerätään ylös erilaisia kehityskohteita auttamaan tulevaisuudessa tapahtuvaa tutkimustyötä.

Ensimmäisenä kehityskohteena on CAPP:n käyttöönotto, jonka avulla prosessiin liittyvä tieto saadaan yksiselitteisesti esitettyä. Saatavilla olevan tiedon laatuun ja sitä kautta käytettävyyteen tarvitsee panostaa tarkkuusvalmistukseen liittyvän CAM-prosessin toteuttamisen optimoimiseksi. Tähän liittyvä osa-alue on esimerkiksi työkalujen lisääminen MRL:ään ja yleisemmin eri osapuolten välillä liikkuva tieto. Kehityskohteena on myös

CAM-prosessin vaihekohtainen nopeuttaminen, joka voidaan hoitaa hyödyntämällä ohjelmointirajapintoja joko suoraan ohjelmassa tai sitten luomalla ohjelman, joka nopeuttaa työskentelyä rajapintojen sisällä. Kaiken kaikkiaan kehityskohteena on työssä esitettyihin ratkaisuehdotuksien sisäistäminen sekä potentiaalisimpiin vaihtoehtoihin perehtyminen syvällisemmin joko olemassa olevien resurssien avulla tai hankkimalla tarpeelliset resurssit tutkimista varten.

Laajimpana kehityskohteena voidaan pitää yleisen tason ideaaliseen CAM-prosessiin pääsemistä. Käsite voidaan ymmärtää monella tapaa mutta diplomityötä tehdessä kirjoittajalle muodostuu käsitys kohdeyrityksen esimerkkitapauksen yhteydessä käytettävästä CAM-prosessista ja sitä kautta voidaan päätellä potentiaalisesti ideaalinen CAM-prosessi siihen liittyen. Ideaalisessa CAM-prosessissa prosessikuvauksessa otetaan huomioon jokainen toteutettava vaihe ja kokonaisuus suunnitellaan käyttämällä siihen sopivaa työkalua. Tällainen CAPP työkalu on esimerkiksi Teamcenterissä oleva Manufacturing Process Planner, jonka avulla voidaan luoda oikeaa tehdasta ja siinä käytettäviä resursseja vastaava digitaalinen kokonaisuus. Tämän avulla prosessien suunnittelu ja seuraaminen hoituu kokonaisuudessaan yhden työkalun avulla ja se integroituu parhaiten Sandvikin prosessin eri vaiheissa käytettävien ohjelmien kanssa. Manufacturing Process Plannerin toimintaa ei esitetä yksityiskohtaisesti työssä mutta kirjallisuudesta löytyy lukuisia artikkeleita sekä käyttöohjeita ohjelman toiminnan sisäistämiseksi.

Ideaalisessa CAM-prosessissa päästään täydellisen suunnittelun lisäksi optimaaliseen toteuttamiseen. Toteuttamisessa jokainen prosessin vaihe hoidetaan tehokkaimmin siten että saatavilla olevaa tietoa, oli se sitten toiselta osapuolelta saatua tai aikaisemmassa prosessissa luotua, pystytään löytämään ja hyödyntämään parhaalla mahdollisella tavalla. Tämän seurauksena työntekijöiden ei tarvitse luoda uudelleen jo aikaisemmin muodostettua tietoa vaan he voivat keskittyä täysin kyseisessä vaiheessa tarvittavaan uuden tiedon luomiseen. Toimiva vaihtoehto olemassa olevan tiedon löytämiseen on säilöä se yhteen paikkaan, esimerkiksi Teamcenteriin, josta sen löytää hakuparametrien avulla. Hakuparametrina voi olla esimerkiksi etsittävän tiedoston nimi tai siihen liittyvän nimikkeen tunnus.

Tiedon tallentamisessa yhteen paikkaan, johon kaikilla sitä tarvitsevilla työntekijöillä on pääsy, ja prosessin vaiheiden välisessä tiedonsiirrossa hyödyllinen tapa on käyttää ohjelmointirajapintoja. Rajapintojen avulla esimerkiksi alihankkija voi lähettää tuotoksensa suoraan yrityksen tiedonhallintajärjestelmässä olevaan ennalta määritellyyn paikkaan tai prosessin aikaisemmassa vaiheessa tuotettu tieto saadaan lähetettyä valmiiksi seuraa-

vassa vaiheessa operoivalle työntekijälle. Rajapintojen avulla saadaan minimoitua manuaalinen tiedonsiirto ja varmistettua tiedon nopein mahdollinen siirtyminen varmasti oikeaan paikkaan. Esimerkkeinä manuaalisesta tiedonsiirrosta on NC-koodin siirtäminen työstökoneelle massamuistin avulla tai alihankkijan tuottaman tiedon vastaanottaminen ensiksi yhteyshenkilön sähköpostiin ja vasta sen jälkeen siirtäminen tiedonhallintajärjestelmään. Rajapinnat mahdollistavat siis ideaalisessa CAM-prosessissa tiedon optimaalisen liikkumisen prosessin eri vaiheiden ja osapuolten välillä.

Uuden tiedon luomisessa käytettävien ohjelmien sekä työkalujen kyvykkyys ja tehokas käyttäminen ovat avainasemassa. Kyvykkyydellä tarkoitetaan kaikkien vaiheissa käytettävien ominaisuuksien löytymistä ja uusimpien keksittyjen menetelmien käyttöönoton mahdollisuutta tarvittaessa. Tehokkain mahdollinen käyttäminen tarkoittaa vaiheen suorittamista minimiajassa ilman virheitä. Tehokkuuden optimoinnissa esimerkiksi toistettavia ja aikaisemmin luotuja operaatioita voidaan automatisoida tai käytettävää ohjelmaa voidaan kustomoida yrityksen omiin tarkoituksiin sopivaksi työssä esiteltyjen ohjelmointirajapintojen avulla. Tehokkaassa käyttämisessä avainasemassa on myös käyttäjä. Ohjelman käyttäjällä tulee olla ajankohtainen käsitys ohjelman ominaisuuksista ja vahva rutiini ohjelman käyttämisestä. Käyttäjän työskentelyn tehostamista voidaan hoitaa säännöllisten koulutuksien avulla ja antamalla mahdollisuus ohjelman kaikkien ominaisuuksien hyödyntämiseen. Ideaalisen CAM-prosessin toteuttamiseksi yrityksen tarvitsee investoida omissa prosesseissaan olevien resurssien optimaaliseen käyttämiseen sekä ideaaliseen CAM-prosessiin tunnistettujen puuttuvien resurssien hankkimiseen.

LÄHTEET

Business Advantage (2018). Worldwide CAD Trends 2018-2019 Survey Report. Business Advantage Group. pp. 59. Saatavissa: https://www.business-advantage.com/landing_page_CAD_Trends_2018_March.php

Casas Fernández, J.S., Sordo, J., Henzold, G. (2006). Geometrical Dimensioning and Tolerancing for Design, Manufacturing and Inspection: A Handbook for Geometrical Product Specification Using ISO and ASME Standards. Oxford: Elsevier Science & Technology. pp. 7-39. Saatavissa: <http://ebookcentral.proquest.com/lib/tampere/detail.action?docID=274217>

Chang, T., Wysk, R.A., Wang, H. (1991). Computer-aided manufacturing. Englewood Cliffs (NJ): Prentice-Hall. pp. 674.

Chang, T., Wysk, R.A., Wang, H. (2006). Computer-aided manufacturing. 3. edition. Upper Saddle River (NJ): Pearson Prentice Hall. pp. 54-96. Saatavissa: <https://www.worldcat.org/title/computer-aided-manufacturing/oclc/607680690>

Cheypoca, T., Lekthamrong, C., Koomgaev, C., Wuti, V., Riewruja, V. (2010). High efficiency postprocessor for generic NC machine. IEEE: International Conference on Control Automation and Systems. pp. 511-514. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=5669956&tag=1>

Chiabert, P., Lombardi, F., Orlando, M. (1998). Benefits of geometric dimensioning and tolerancing. Journal of Materials Processing Technology. volume 78, issue 1. ScienceDirect. pp. 29-35. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0924013697004597>

Cordes, S., Hübner, F., Schaarschmidt, T. (2014). Next Generation High Performance Cutting by Use of Carbon Dioxide as Cryogenics. Procedia CIRP. pp. 401-405. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S2212827114002340>

Dimitrov, D., Saxer, M. (2012). Productivity Improvement in Tooling Manufacture through High Speed 5 Axis Machining. Procedia CIRP. pp. 277-282, Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S2212827112000510>

Dornfeld, D., Lee, D. (2008). Precision Manufacturing. Springer Verlag. pp. 775. Saatavissa: <https://link-springer-com.libproxy.tuni.fi/content/pdf/10.1007%2F978-0-387-68208-2.pdf>

Ismail, N., Lim, K.S., Tan, C.F. (2003). Modelling and machining of complex mechanical parts using CAD/CAM systems. IEEE: Student Conference on Research and Development. pp. 183-187. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=1459689&tag=1>

Jiang, G. (2011). Direct Numerical Control Techniques applications in local area network. IEEE: International Conference on Electronics & Mechanical Engineering and Information Technology. pp. 3388-3390. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=6023813>

Kurowski, P.M. (2017). *Finite Element Analysis for Design Engineers*. 2. edition. Warrendale, PA: SAE International. pp. 209-233. Saatavissa: <http://web.a.ebsco-host.com/ehost/detail/detail?vid=0&sid=5152604f-fb29-4ae2-88f5-0c6022743e4e%40sdc-v-sessmgr02&bdata=JkF1dGhUeXBIP-WNvb2tpZSxpcCx1aWQmc2l0ZT1laG9zdC1saXZlJnNjb3BIPXNpdGU%3d##AN=1879562&db=nlebk>

Lai, X. (2014). A design of general compiler for NC code in embedded NC system. *IEEE: Conference on Industrial Electronics and Applications*. pp. 1515-1519. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=6931409>

Lin, R., Lin, T. (2010). Trajectory analyses for five-axis machine tools. *IEEE: Conference on Robotics, Automation and Mechatronics*. pp. 136-141. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=5513198>

Meadows, J.D. (2009). *Geometric dimensioning and tolerancing: applications, analysis & measurement (per ASME Y14.5-2009)*. Hendersonville, TN: New York. James D. Meadows & Associates; ASME Press. pp. 574.

Meng, M., Steinhardt, S. and Schubert, A. (2018). Application Programming Interface Documentation: What Do Software Developers Want?. *Journal of Technical Writing and Communication*. volume 48. issue 3. Sage Publications Ltd. pp. 295-330. Saatavissa: <https://journals-sagepub-com.libproxy.tuni.fi/doi/pdf/10.1177/0047281617721853>

Mian, S.H., Al-Ahmari, A. (2014). Enhance performance of inspection process on Coordinate Measuring Machine. *Measurement*. volume 47. ScienceDirect. pp. 78-91. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0263224113004041>

Narayana, K.L. (2006). *Machine Drawing*. Daryaganj: New Age International Ltd. pp. 232-250. Saatavissa: <https://ebookcentral.proquest.com/lib/tampere/detail.action?docID=431526>

Ni, X.H., Gao, Z.Q. (2013). System Development in Integration of DNC-Based Manufacturing Information. *Applied Mechanics and Materials*. volume 312. Zurich, Switzerland: Trans Tech Publications Ltd. pp. 464. Saatavissa: <https://search-proquest-com.libproxy.tuni.fi/docview/1442495510?accountid=14242>

Novak-Marcincin, J., Barna, J., Torok, J., Novakova-Marcincinova, L. (2013). Definition of Computer Aided Manufacturing Engineering and its place in CA systems chain. *IEEE: International Symposium on Applied Machine Intelligence and Informatics*. pp. 263-266. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=6480989&tag=1>

Pere, A. (2009). *Koneenpiirustus 1 & 2*. 10. edition. Espoo: Kirpe. pp. 776

Pohanish, R.P. (2016). *Surface Texture. Machinery's Handbook Pocket Companion*. Revised 1. edition. Industrial Press. pp. 272-278. Saatavissa: <https://app.knovel.com/hotlink/pdf/id:kt010ZN0G1/machinerys-handbook-pocket/material-removal-required>

Rajurkar, K.P., Levy, G., Malshe, A., Sundaram, M.M., McGeough, J., Hu, X., Resnick, R. and DeSilva, A. (2006). Micro and Nano Machining by Electro-Physical and Chemical Processes. *Manufacturing Technology*. volume 55. issue 2. CIRP Annals. pp. 643-666. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S166027730600003X>

Rennels, K.E. (2003). Current methodologies for geometric dimensioning and tolerancing. Indianapolis, USA: IEEE. pp. 565-569. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/document/1247948>

Rufe, P.D. (2013). Geometric Dimensioning and Tolerancing. Fundamentals of Manufacturing. 3. edition. Society of Manufacturing Engineers (SME). pp. 149-164. Saatavissa: <https://app.knovel.com/hotlink/pdf/id:kt00BKEG52/fundamentals-manufacturing/gd-t-symbols-definitions>

Sandvik (2019). About us. Verkkosivu. Saatavissa: <https://www.home.sandvik/en/about-us/>

SFS-EN ISO 1101 (2017). Geometrinen tuotemäärittely (GPS). Geometriset toleranssit. Muodon, suunnan, sijainnin ja heiton toleranssit. Helsinki: Suomen standardisoimisliitto SFS ry. pp. 116.

SFS-EN ISO 1302 (2003). Geometrinen tuotemäärittely (GPS). Pinnan ominaisuuksien ilmoittaminen teknisissä tuoteasiakirjoissa. Helsinki: Suomen standardisoimisliitto SFS ry. pp. 91.

She, C., Lin, C., Yang, T. (2012). Development of the Integration of Postprocessor for Five-Axis Machine Tools. IEEE/ASME: International Conference on Advanced Intelligent Mechatronics (AIM). pp. 638-643. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=6265932&tag=1>

Shinde, S.M., Bhole, K.S. (2015). Review of accuracy improvement techniques in high speed 5 axis machining. IEEE: International Conference on Nascent Technologies in the Engineering Field. pp. 1-5. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=7029906&tag=1>

Siemens (2019). Documentation Center. Verkkosivu. Saatavissa: <https://www.plm.automation.siemens.com/global/en/support/docs.html>

Simmons, C.H., Phelps, N., Dennis, E., Maguire, C. (2012). Differences between American ASME Y 14.5M Geometric Dimensioning and Tolerancing (GD&T) and ISO/BS 8888 Geometrical Tolerancing Standards. Manual of Engineering Drawing. 4. edition. Oxford: Butterworth-Heinemann. pp. 247-252. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/B9780080966526000309>

Stojadinovic, S.M., Majstorovic, V.D., Durakbasa, N.M., Sibalija, T.V. (2016). Towards an intelligent approach for CMM inspection planning of prismatic parts. Measurement. volume 92. pp. 326-339. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0263224116303293>

Sungchul, J., Taehoon, K. (2003). Tool-path generation for NURBS surface machining. IEEE: Proceedings of the 2003 American Control Conference. pp. 2614-2619. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=1243471&tag=1>

Tang, Q., Yin, S., Chen, F., Huang, S., Luo, H., Geng, J. (2018). Development of a post-processor for head tilting-head rotation type five-axis machine tool with double limit rotation axis. The International Journal of Advanced Manufacturing Technology. volume 97. issue 9. London: Springer London. pp. 3523-3534. Saatavissa: <https://link-springer-com.libproxy.tuni.fi/content/pdf/10.1007%2Fs00170-018-2195-3.pdf>

Yang, T., Zhang, D., Chen, B., Li, S. (2008). Research on Plant Layout and Production Line Running Simulation in Digital Factory Environment. IEEE: Pacific-Asia Workshop on Computational Intelligence and Industrial Application. pp. 588-593. Saatavissa: <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=4756843>

Yusof, Y., Latif, K. (2014). Survey on computer-aided process planning. The International Journal of Advanced Manufacturing Technology. volume 75. issue 1. London: Springer London. pp. 77-89. Saatavissa: <https://link-springer-com.libproxy.tuni.fi/content/pdf/10.1007%2Fs00170-014-6073-3.pdf>

Zębala, W., Plaza, M. (2014). Comparative study of 3- and 5-axis CNC centers for free-form machining of difficult-to-cut material. International Journal of Production Economics. volume 158. ScienceDirect. pp. 345-358. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0925527314002643>

Zhang, S.J, To, S., Wang, S.J., Zhu, Z.W. (2015). A review of surface roughness generation in ultra-precision machining. International Journal of Machine Tools and Manufacture. volume 91. ScienceDirect. pp. 76-95. Saatavissa: <https://www-sciencedirect-com.libproxy.tuni.fi/science/article/pii/S0890695515300092>

LIITE A: LUVUN 6.2 CAD PUOLEN KOODI

```

import math
import NXOpen
import NXOpen.Annotations
import NXOpen.Features
import NXOpen.GeometricUtilities
import NXOpen.Preferences

def main() :

    theSession = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work

    #Sylinteri 1

    cylinderbuilder1 = workPart.Features.CreateCylinderBuilder(NXOpen.Features.Cylinder.Null)

    cylinderbuilder1.Diameter.RightHandSide = "100"
    cylinderbuilder1.Height.RightHandSide = "30.0"
    cylinderbuilder1.Origin = NXOpen.Point3d(60.0, 30.0, 0.0)
    cylinderbuilder1.Direction = NXOpen.Vector3d(0.0, 1.0, 0.0)
    cylinderbuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Create

    cylinderbuilder1.Commit()
    cylinderbuilder1.Destroy()

    #Sylinteri 2

    cylinderbuilder2 = workPart.Features.CreateCylinderBuilder(NXOpen.Features.Cylinder.Null)

    cylinderbuilder2.Diameter.RightHandSide = "50"
    cylinderbuilder2.Height.RightHandSide = "100"
    cylinderbuilder2.Origin = NXOpen.Point3d(60.0, 60.0, 0.0)
    cylinderbuilder2.Direction = NXOpen.Vector3d(0.0, 1.0, 0.0)
    cylinderbuilder2.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Create

    cylinderbuilder2.Commit()
    cylinderbuilder2.Destroy()

    #Kartio

    conebuilder = workPart.Features.CreateConeBuilder(NXOpen.Features.Cone.Null)

    conebuilder.BaseDiameter.RightHandSide = "100"
    conebuilder.Height.RightHandSide = "50"

    origin1 = NXOpen.Point3d(60.0, 160.0, 0.0)
    vector1 = NXOpen.Vector3d(0.0, 1.0, 0.0)
    direction1 = workPart.Directions.CreateDirection(origin1, vector1,
    NXOpen.SmartObject.UpdateOption.WithinModeling)

    axis1 = conebuilder.Axis

```

```

point1 = workPart.Points.CreatePoint(origin1)

axis1.Point = point1
axis1.Direction = direction1

conebuilder.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOperation.BooleanType.Create

conebuilder.Commit()
conebuilder.Destroy()

#Kuutio

blockfeaturebuilder1 = workPart.Features.CreateBlockFeature-
Builder(NXOpen.Features.Block.Null)
blockfeaturebuilder1.Type = NXOpen.Features.BlockFeature-
Builder.Types.OriginAndEdgeLengths

origo = NXOpen.Point3d(0.0, -20.0, -60.0)
blockfeaturebuilder1.SetOriginAndLengths(origo, "120", "50", "120")
blockfeaturebuilder1.BooleanOption.Type = NXOpen.GeometricUtili-
ties.BooleanOperation.BooleanType.Create

blockfeaturebuilder1.Commit()
blockfeaturebuilder1.Destroy()

#Pallo

sphereBuilder1 = workPart.Features.CreateSphereBuilder(NXOpen.Fea-
tures.Sphere.Null)

piste = NXOpen.Point3d(60.0, 250.0, 0.0)
origo = workPart.Points.CreatePoint(piste)
sphereBuilder1.CenterPoint = origo
sphereBuilder1.Diameter.RightHandSide = "80"
sphereBuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.Boo-
leanOperation.BooleanType.Create

sphereBuilder1.Commit()
sphereBuilder1.Destroy()

#Putki

sketchInPlaceBuilder1 = workPart.Sketches.CreateSketchInPlace-
Builder2(NXOpen.Sketch.Null)

origin1 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal1 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane1 = workPart.Planes.CreatePlane(origin1, normal1,
NXOpen.SmartObject.UpdateOption.WithinModeling)

sketchInPlaceBuilder1.PlaneReference = plane1

nXObject1 = sketchInPlaceBuilder1.Commit()

sketch1 = nXObject1
feature1 = sketch1.Feature

```

```

sketch1.Activate(NXOpen.Sketch.ViewReorient.TrueValue)

sketchInPlaceBuilder1.Destroy()
plane1.DestroyPlane()

#Putken muotoviivan sketchin määrittely

startPoint1 = NXOpen.Point3d(60.0, -120.0, 0.0)
endPoint1 = NXOpen.Point3d(60.0, -20.0, 0.0)
line1 = workPart.Curves.CreateLine(startPoint1, endPoint1)

theSession.ActiveSketch.AddGeometry(line1, NXOpen.Sketch.InferCon-
straintsOption.InferNoConstraints)

theSession.ActiveSketch.Update()

#Putken muotoviivan määrittely

tubeBuilder1 = workPart.Features.CreateTubeBuilder(NXOpen.Features.Fea-
ture.Null)

tubeBuilder1.Tolerance = 0.01

tubeBuilder1.OuterDiameter.RightHandSide = "60"
tubeBuilder1.InnerDiameter.RightHandSide = "40"

tubeBuilder1.BooleanOption.Type = NXOpen.GeometricUtilities.BooleanOp-
eration.BooleanType.Create

tubeBuilder1.PathSection.DistanceTolerance = 0.01
tubeBuilder1.PathSection.ChainingTolerance = 0.01

tubeBuilder1.PathSection.SetAllowedEntityTypes(NXOpen.Section.Allow-
Types.OnlyCurves)

features1 = [NXOpen.Features.Feature.Null] * 1
sketchFeature1 = feature1 #sketchin viiva
features1[0] = sketchFeature1
curveFeatureRule1 = workPart.ScRuleFactory.CreateRuleCurveFeature(fea-
tures1)

tubeBuilder1.PathSection.AllowSelfIntersection(True)

rules1 = [None] * 1
rules1[0] = curveFeatureRule1
helpPoint1 = NXOpen.Point3d(0.0, 0.0, 0.0)
tubeBuilder1.PathSection.AddToSection(rules1, NXOpen.NXObject.Null,
NXOpen.NXObject.Null, NXOpen.NXObject.Null, helpPoint1, NXOpen.Sec-
tion.Mode.Create, False)

#Putki määritely

nXObject2 = tubeBuilder1.Commit()

tubeBuilder1.Destroy()

if __name__ == '__main__':
    main()

```

LIITE B: LUVUN 6.2 CAM PUOLEN KOODI

```

import math
import NXOpen
import NXOpen.CAM
import NXOpen.Features
def main() :

    theSession = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work
    displayPart = theSession.Parts.Display

    #Luodaan pinnan jyrsintä halutulla työkalulla ja geometrialla

    nCGroup1 = workPart.CAMSetup.CAMGroupCollection.FindObject("NC_PRO-
GRAM")
    method1 = workPart.CAMSetup.CAMGroupCollection.FindObject("METHOD")
    tool1 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0212_002")
    orientGeometry1 = workPart.CAMSetup.CAMGroupCollection.FindOb-
ject("MCS_SPINDLE")
    operation1 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
method1, tool1, orientGeometry1, "mill_planar", "FACE_MILLING",
NXOpen.CAM.OperationCollection.UseDefaultName.TrueValue, "FACE_MILL-
ING")

    faceMilling1 = operation1
    faceMillingBuilder1 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling1)
    faceMillingBuilder1.PartGeometry.InitializeData(False)

    geometrySetList1 = faceMillingBuilder1.PartGeometry.GeometryList

    taggedObject1 = geometrySetList1.FindItem(0)
    geometrySet1 = taggedObject1

    #Määritellään kappale

    partLoadStatus1 = workPart.LoadThisPartFully()

    partLoadStatus1.Dispose()
    bodies1 = [NXOpen.Body.Null] * 1
    body1 = workPart.Bodies.FindObject("EXTRUDE(10)")
    bodies1[0] = body1
    bodyDumbRule1 = workPart.ScRuleFactory.CreateRuleBodyDumb(bodies1,
True)

    scCollector1 = geometrySet1.ScCollector

    rules1 = [None] * 1
    rules1[0] = bodyDumbRule1
    scCollector1.ReplaceRules(rules1, False)

    NXObject1 = faceMillingBuilder1.Commit()
    faceMillingBuilder1.Destroy()

    faceMilling2 = NXObject1
    faceMillingBuilder2 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling2)

```

```

#Määritellään aihio

origin1 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal1 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane1 = workPart.Planes.CreatePlane(origin1, normal1,
NXOpen.SmartObject.UpdateOption.AfterModeling)

unit1 = workPart.UnitCollection.FindObject("MilliMeter")

boundary1 = faceMillingBuilder2.BlankBoundary
boundarySetList1 = boundary1.BoundaryList

taggedObject2 = boundarySetList1.FindItem(0)

boundaryMillingSet1 = taggedObject2
origin2 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal2 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane2 = workPart.Planes.CreatePlane(origin2, normal2,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryMillingSet1.Plane = plane2

faces1 = [NXOpen.Face.Null] * 1
extrude1 = workPart.Features.FindObject("EXTRUDE(10)")
face1 = extrude1.FindObject("FACE 130 {(40,0,0) EXTRUDE(10)}")
faces1[0] = face1
faceDumbRule1 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces1)

scCollector2 = workPart.FindObject("ENTITY 113 1")
rules2 = [None] * 1
rules2[0] = faceDumbRule1
scCollector2.ReplaceRules(rules2, False)

bndsets1 = boundary1.AppendFaceBoundary(face1, True, False, False,
NXOpen.CAM.BoundarySet.ToolSideTypes.InsideOrLeft)

taggedObject3 = boundaryMillingSet1.BoundaryMemberList.FindItem(0)

boundaryMillingMemberSet1 = taggedObject3
postEventsCiBuilder1 = boundaryMillingMemberSet1.StartEventsBuilder
postEventsCiBuilder2 = boundaryMillingMemberSet1.EndEventsBuilder

plane1.DestroyPlane()
nXObject2 = faceMillingBuilder2.Commit()
faceMillingBuilder2.Destroy()

faceMilling3 = nXObject2
faceMillingBuilder3 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling3)

#Luodaan työstöradat

nXObject3 = faceMillingBuilder3.Commit()

objects1 = [NXOpen.CAM.CAMObject.Null] * 1
faceMilling4 = nXObject3
objects1[0] = faceMilling4
workPart.CAMSetup.GenerateToolPath(objects1)

```



```

faceMillingBuilder3.Destroy()
faceMillingBuilder4 = workPart.CAMSetup.CAMOperationCollection.Create-
FaceMillingBuilder(faceMilling4)
nXObject4 = faceMillingBuilder4.Commit()
faceMillingBuilder4.Destroy()
theSession.CleanUpFacetedFacesAndEdges()

#pinnan jyrsintä suoritettu
#Luodaan tason jyrsintä halutulla työkalulla ja geometrialla

tool2 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0201_004")
operation2 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
method1, tool2, orientGeometry1, "mill_planar", "PLANAR_MILL",
NXOpen.CAM.OperationCollection.UseDefaultValue.TrueValue, "PLANAR_MILL")

planarMilling1 = operation2
planarMillingBuilder1 = workPart.CAMSetup.CAMOperationCollection.Cre-
atePlanarMillingBuilder(planarMilling1)

#Määritellään kappale

origin3 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal3 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane3 = workPart.Planes.CreatePlane(origin3, normal3,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryPlanarMill1 = planarMillingBuilder1.PartBoundary
boundarySetList2 = boundaryPlanarMill1.BoundaryList

taggedObject4 = boundarySetList2.FindItem(0)
boundarySetPlanarMill1 = taggedObject4
origin4 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal4 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane4 = workPart.Planes.CreatePlane(origin4, normal4,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundarySetPlanarMill1.Plane = plane4

faces2 = [NXOpen.Face.Null] * 1
faces2[0] = face1
faceDumbRule2 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces2)

scCollector3 = workPart.FindObject("ENTITY 113 7")
rules3 = [None] * 1
rules3[0] = faceDumbRule2
scCollector3.ReplaceRules(rules3, False)

plane5 = plane4.CopyPlane()

bndsets2 = boundaryPlanarMill1.AppendFaceBoundary(face1, False, True,
False, NXOpen.CAM.BoundarySet.ToolSideTypes.OutsideOrRight, 1, 1)

boundarySetPlanarMill2 = bndsets2[1]
boundarySetPlanarMill2.Plane = plane5

taggedObject5 = boundarySetPlanarMill1.BoundaryMemberList.FindItem(0)

plane3.DestroyPlane()

```

```

nXObject5 = planarMillingBuilder1.Commit()

planarMillingBuilder1.Destroy()

planarMilling2 = nXObject5
planarMillingBuilder2 = workPart.CAMSetup.CAMOperationCollection.CreatePlanarMillingBuilder(planarMilling2)

#Määritellään aihio

origin5 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal5 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane6 = workPart.Planes.CreatePlane(origin5, normal5,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundaryPlanarMill2 = planarMillingBuilder2.BlankBoundary

boundarySetList3 = boundaryPlanarMill2.BoundaryList

taggedObject6 = boundarySetList3.FindItem(0)

boundarySetPlanarMill3 = taggedObject6
origin6 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal6 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane7 = workPart.Planes.CreatePlane(origin6, normal6,
NXOpen.SmartObject.UpdateOption.AfterModeling)

boundarySetPlanarMill3.Plane = plane7

faces3 = [NXOpen.Face.Null] * 1
faces3[0] = face1
faceDumbRule3 = workPart.ScRuleFactory.CreateRuleFaceDumb(faces3)

scCollector4 = workPart.FindObject("ENTITY 113 5")
rules4 = [None] * 1
rules4[0] = faceDumbRule3
scCollector4.ReplaceRules(rules4, False)

bndsets3 = boundaryPlanarMill2.AppendFaceBoundary(face1, True, False,
False, NXOpen.CAM.BoundarySet.ToolSideTypes.InsideOrLeft, 1, 1)
taggedObject7 = boundarySetPlanarMill3.BoundaryMemberList.FindItem(0)
plane6.DestroyPlane()
nXObject6 = planarMillingBuilder2.Commit()
planarMillingBuilder2.Destroy()

planarMilling3 = nXObject6
planarMillingBuilder3 = workPart.CAMSetup.CAMOperationCollection.CreatePlanarMillingBuilder(planarMilling3)

#Määritellään lattia

origin7 = NXOpen.Point3d(0.0, 0.0, 0.0)
normal7 = NXOpen.Vector3d(0.0, 0.0, 1.0)
plane8 = workPart.Planes.CreatePlane(origin7, normal7,
NXOpen.SmartObject.UpdateOption.AfterModeling)

plane8.SetUpdateOption(NXOpen.SmartObject.UpdateOption.AfterModeling)
plane8.SetMethod(NXOpen.PlaneTypes.MethodType.Distance)

```

```

geom1 = [NXOpen.NXObject.Null] * 1
extrude2 = workPart.Features.FindObject("EXTRUDE(12)")
face2 = extrude2.FindObject("FACE 130 {(15,0,0) EXTRUDE(10)}")
geom1[0] = face2

plane8.SetGeometry(geom1)
plane8.SetFlip(False)
plane8.SetReverseSide(False)
plane8.SetAlternate(NXOpen.PlaneTypes.AlternateType.One)
plane8.Evaluate()
plane8.RemoveOffsetData()

planarMillingBuilder3.Geometry.FloorPlane = plane8

nXObject7 = planarMillingBuilder3.Commit()

#Luodaan työstöradat

objects2 = [NXOpen.CAM.CAMObject.Null] * 1
planarMilling4 = nXObject7
objects2[0] = planarMilling4
workPart.CAMSetup.GenerateToolPath(objects2)

planarMillingBuilder3.Destroy()

theSession.CleanUpFacetedFacesAndEdges()

#Pinnan jyrsintä suoritettu
#Luodaan ulkopuolen halkaisijan sorvaus halutulla työkalulla ja geomet-
rialla

tool3 = workPart.CAMSetup.CAMGroupCollection.FindObject("UGT0101_011")
featureGeometry1 = workPart.CAMSetup.CAMGroupCollection.FindOb-
ject("TURNING_WORKPIECE")
operation3 = workPart.CAMSetup.CAMOperationCollection.Create(nCGroup1,
method1, tool3, featureGeometry1, "turning", "ROUGH_TURN_OD",
NXOpen.CAM.OperationCollection.UseDefaultName.TrueValue,
"ROUGH_TURN_OD")

roughTurning1 = operation3
roughTurningBuilder1 = workPart.CAMSetup.CAMOperationCollection.Cre-
ateRoughTurningBuilder(roughTurning1)

#Määritellään työstöalueen rajat

roughTurningBuilder1.TrimPoint1.Option = NXOpen.CAM.TrimPoint.Op-
tions.Point

scalar1 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.Dimensional-
ityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

extractFace1 = workPart.Features.FindObject("SpunOutline(59)")

edge1 = extractFace1.FindObject("EDGE * -2 * 1
{(0,32.4999999999986,0)(20,32.4999999999986,0)(40,32.4999999999986,0)
SpunOutline(59)}")
point1 = workPart.Points.CreatePoint(edge1, scalar1,
NXOpen.SmartObject.UpdateOption.AfterModeling)

```

```

xform1, nXObject9 = workPart.Xforms.CreateExtractXform(edge1,
NXOpen.SmartObject.UpdateOption.AfterModeling, False)

scalar2 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.DimensionalityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)
edge2 = nXObject9

point2 = workPart.Points.CreatePoint(edge2, scalar2,
NXOpen.SmartObject.UpdateOption.AfterModeling)
point2.RemoveViewDependency()
roughTurningBuilder1.TrimPoint1.Point = point2
point2.RemoveViewDependency()
roughTurningBuilder1.TrimPoint2.Option = NXOpen.CAM.TrimPoint.Options.Point

scalar3 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.DimensionalityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

edge3 = extractFace1.FindObject("EDGE * -1 * 1
{(0,0,0)(0,16.2499999999993,0)(0,32.4999999999986,0) SpunOutline(59)}")
point3 = workPart.Points.CreatePoint(edge3, scalar3,
NXOpen.SmartObject.UpdateOption.AfterModeling)

xform2, nXObject10 = workPart.Xforms.CreateExtractXform(edge3,
NXOpen.SmartObject.UpdateOption.AfterModeling, False)

scalar4 = workPart.Scalars.CreateScalar(1.0, NXOpen.Scalar.DimensionalityType.NotSet, NXOpen.SmartObject.UpdateOption.AfterModeling)

edge4 = nXObject10
point4 = workPart.Points.CreatePoint(edge4, scalar4,
NXOpen.SmartObject.UpdateOption.AfterModeling)
point4.RemoveViewDependency()
roughTurningBuilder1.TrimPoint2.Point = point4
point4.RemoveViewDependency()

workPart.Points.DeletePoint(point1)
workPart.Points.DeletePoint(point3)

nXObject11 = roughTurningBuilder1.Commit()

#Luodaan työstöradat

objects3 = [NXOpen.CAM.CAMObject.Null] * 1
roughTurning2 = nXObject11
objects3[0] = roughTurning2
workPart.CAMSetup.GenerateToolPath(objects3)

roughTurningBuilder1.Destroy()
theSession.CleanUpFacetedFacesAndEdges()

if __name__ == '__main__':
    main()

```

LIITE C: LUVUN 6.2 CMM PUOLEN KOODI

```

import math
import NXOpen
import NXOpen.Assemblies
import NXOpen.CAM
import NXOpen.Positioning
import NXOpen.Preferences
def main() :

    theSession = NXOpen.Session.GetSession()
    workPart = theSession.Parts.Work
    displayPart = theSession.Parts.Display

    inspectionGroup1 = workPart.InspectionSetup.CmmInspectionGroupCollection.FindObject("GENERIC_MACHINE")
    machineGroupBuilder1 = workPart.InspectionSetup.CmmInspectionGroupCollection.CreateMachineGroupBuilder(inspectionGroup1)
    ncmctPartMountingBuilder1 = workPart.InspectionSetup.CreateNcmctPartMountingBuilder("zeiss_prismo")
    ncmctPartMountingBuilder1.Positioning = NXOpen.CAM.NcmctPartMountingBuilder.PositioningTypes.UseAssemblyPositioning
    selectNXObjectList1 = ncmctPartMountingBuilder1.Geometry

    #Lisätään kappale

    selectNXObjectList2 = ncmctPartMountingBuilder1.Geometry
    component1 = workPart.ComponentAssembly.RootComponent.FindObject("COMPONENT cam_tau 1")
    body1 = component1.FindObject("PROTO#.Bodies|EXTRUDE(3)")
    added1 = selectNXObjectList2.Add(body1)
    NXObject1 = ncmctPartMountingBuilder1.Commit()
    machineGroupBuilder1.RemoveMachine()

    #Lisätään mittauskone

    basePoint1 = NXOpen.Point3d(0.0, 0.0, 0.0)
    orientation1 = NXOpen.Matrix3x3()
    orientation1.Xx = 1.0
    orientation1.Xy = 0.0
    orientation1.Xz = 0.0
    orientation1.Yx = 0.0
    orientation1.Yy = 1.0
    orientation1.Yz = 0.0
    orientation1.Zx = 0.0
    orientation1.Zy = 0.0
    orientation1.Zz = 1.0
    component2, partLoadStatus1 = workPart.ComponentAssembly.AddComponent("C:\\Apps\\Siemens\\NX12\\cmm_inspection\\resource\\library\\machine\\installed_machines\\zeiss_prismo\\graphics\\Zeiss_Prismo.prt", "None", "ZEISS_PRISMO", basePoint1, orientation1, -1, True)

    partLoadStatus1.Dispose()
    objects1 = []
    nErrs2 = theSession.UpdateManager.AddObjectsToDeleteList(objects1)
    machineGroupBuilder1.MachinePartOccurrence = component2
    machineGroupBuilder1.UpdateCamSetup(NXOpen.CAM.MachineGroupBuilder.RetrieveToolPocketInformation.Yes, ncmctPartMountingBuilder1)

```

```

ncmctPartMountingBuilder1.Destroy()
machineGroupBuilder1.Destroy()

#Lisätään mittapää ja sen pidike

machineGroupBuilder2 = workPart.InspectionSetup.CmmInspectionGroupCollection.CreateMachineGroupBuilder(inspectionGroup1)
inspectionGroup2 = workPart.InspectionSetup.RetrieveDevice("Zeiss_VAST")
inspectionTool1, success1 = workPart.InspectionSetup.RetrieveTool("TP20_STD_A-5003-0040-01-A")
NXObject2 = machineGroupBuilder2.Commit()
machineGroupBuilder2.Destroy()

#Perussuuntaus ja nollapiste

componentPositioner1 = workPart.ComponentAssembly.Positioner
componentPositioner1.ClearNetwork()

arrangement1 = workPart.ComponentAssembly.Arrangements.FindObject("Arrangement 1")
componentPositioner1.PrimaryArrangement = arrangement1
componentPositioner1.BeginMoveComponent()
allowInterpartPositioning1 = theSession.Preferences.Assemblies.InterpartPositioning
expression1 = workPart.Expressions.CreateSystemExpressionWithUnits("1.0", NXOpen.Unit.Null)
unit1 = workPart.UnitCollection.FindObject("MilliMeter")
unit2 = workPart.UnitCollection.FindObject("Degrees")
network1 = componentPositioner1.EstablishNetwork()
componentNetwork1 = network1
componentNetwork1.MoveObjectsState = True
componentNetwork1.DisplayComponent = NXOpen.Assemblies.Component.Null
componentNetwork1.NetworkArrangementsMode = NXOpen.Positioning.ComponentNetwork.ArrangementsMode.Existing
componentNetwork1.RemoveAllConstraints()

movableObjects1 = [NXOpen.NXObject.Null] * 1
movableObjects1[0] = component1
componentNetwork1.SetMovingGroup(movableObjects1)
componentNetwork1.Solve()
loaded1 = componentNetwork1.IsReferencedGeometryLoaded()
componentNetwork1.BeginDrag()

translation1 = NXOpen.Vector3d(0.0, 0.0, 0.0)

rotation1 = NXOpen.Matrix3x3()
rotation1.Xx = 0.0
rotation1.Xy = 0.0
rotation1.Xz = -1.0
rotation1.Yx = 0.0
rotation1.Yy = 1.0
rotation1.Yz = 0.0
rotation1.Zx = 1.0
rotation1.Zy = 0.0
rotation1.Zz = 0.0

componentNetwork1.DragByTransform(translation1, rotation1)

```

```

componentNetwork1.EndDrag()
componentNetwork1.ResetDisplay()
componentNetwork1.ApplyToModel()
componentNetwork1.Solve()
componentPositioner1.ClearNetwork()
nErrs3 = theSession.UpdateManager.AddToDeleteList(componentNetwork1)
componentPositioner1.DeleteNonPersistentConstraints()
componentPositioner1.EndMoveComponent()
componentPositioner1.PrimaryArrangement = NXOpen.Assemblies.Arrangement.Null

#Liitetään PMI tieto ja luodaan mittausohjelma

inspectionLinkPmiBuilder1 = workPart.InspectionSetup.CmmInspection-
OperationCollection.CreateInspectionLinkPmiBuilder(NXOpen.CAM.CAMOb-
ject.Null)

inspectionLinkPmiBuilder1.WorkpieceString = "Cam Workpiece"
inspectionLinkPmiBuilder1.ViewString = "All"
inspectionLinkPmiBuilder1.ToolString = "Auto"
inspectionLinkPmiBuilder1.TipString = "Auto"
inspectionLinkPmiBuilder1.AngleString = "Auto"
inspectionLinkPmiBuilder1.ProgramLocationString = "INSPECTION_PATHS"
inspectionLinkPmiBuilder1.ProbeTipType = NXOpen.CAM.InspectionMove-
Builder.ProbeTipTypes.Any
inspectionLinkPmiBuilder1.CreatePathsEnum = NXOpen.CAM.Inspec-
tionLinkPmiBuilder.CreatePathsOptions.Yes

inspectionLinkPmiBuilder1.LinkToPmi()
inspectionLinkPmiBuilder1.OutputResults(NXOpen.ListingWindow.Device-
Type.Window, "")
inspectionLinkPmiBuilder1.Destroy()

if __name__ == '__main__':
    main()

```

LIITE D: SANDVIKILLE TEHTY SUUNNITELMA

NX Openin hyödyntäminen suunnittelussa: Suunnittelijoilla voidaan luetuttaa tämä diplomityö tai sitten kouluttaa käyttö osaavan henkilön toimesta. Nykyinen lisenssi kattaa NX Openin käytön CAD ja CAM puolten osin, jolloin ei synny lisäkustannuksia. Jos Sandvikilla työskentelee valmiiksi osaava ihminen niin koulutuksesta ei aiheudu kustannuksia. Lukemiseen tai koulutukseen riittää muutama tunti. NX Openin testaamisen kautta jokainen suunnittelija pystyy luomaan omaa työskentelyä helpottavia Journaleja ja sitä kautta työtä pystytään tekemään tehokkaammin. Journalien luominen lisää suunnittelijoiden käyttämää aikaa alussa jonkin verran koska operaatioiden tallentaminen tarvitsee suunnitella ja toteuttaa huolella mutta onnistuneen käytön hyödyt kompensoivat alussa käytetyn ajan pidemmällä aikavälillä.

CAPP: Teamcenterin lisenssi kattaa Manufacturing Process Plannerin käytön, joten käyttämisen aloittamiseksi ei tarvitse maksaa ylimääräistä. Tehtaan resurssien tekeminen vie todennäköisesti aikaa monta kuukautta mutta mallinnuksen seurauksena jokaiselle valmistettavalle tuotteelle voidaan tehdä täydellinen prosessisuunnitelma ja seurata, missä kohtaa prosessia liikutaan. Tehtaan resurssien mallintamiseksi, ylläpitämiseksi sekä prosessisuunnitelmien tekemiseksi voidaan palkata yksi uusi kokoaikainen työntekijä, jonka kustannus on vuositasolla kymmeniä tuhansia euroja. Toinen vaihtoehto on hyödyntää olemassa olevaa työntekijää vaihtamalla hänen rooliaan CAPP vastaavaksi, jolloin kustannuksia ei synny. Prosessin tilanteen päivittämiseen voidaan kehittää jonkinlainen kuittautapa tai sitten selvittää olemassa olevien ohjelmien kyvykkyyttä tilanteen päivittämiseen. Tähän liittyviä kustannuksia syntyy muutaman tuhannen euron edestä.

CMM: Automaattisen CMM ohjelman toteuttamiseksi pitää selvittää, kannattaako mitausohjelma tehdä kokonaan NX:n Inspection kokonaisuuden avulla ja postprosessoida se sieltä Zeissin koneelle sopivaksi vai käyttää Calypsoa CMM ohjelman tekemiseen hyväksikäyttäen NX:ssä määriteltyjä asioita. Joka tapauksessa kappaleen geometriaan tulee liittää vähintään CMM ohjelmassa tutkittavien piirteiden PMI tieto. Vähimmäistiedon lisäämiseen riittää alle tunti kappaletta kohti, joka lisää suunnittelijoiden tai laatuinsinöörin työmäärää. Jos koko ohjelma tehdään NX:n avulla niin nykyiseen lisenssiin tarvitsee lisätä Inspection osuus, hankkia DMIS postprosessori ja Lisätä Calypson lisenssiin DMIS tulkittamisen mahdollisuus. DMIS postprosessorin hinta on noin 9 000€ erikseen ostettuna mutta sen pitäisi kuulua ilman lisäkustannuksia NX:n Inspection osuuden kattavan

lisenssin hintaan. Calypson DMIS ominaisuuden lisenssikohtaiset kustannukset ovat tuhansia euroja. Tämän lisäksi tarvitsee saada käyttöön mittauskoneiden kinemaattiset mallit sekä pitimien ja mittapäiden mallit. Näitä löytyy valmiina Calypson kirjastoista mutta tarvittaessa Zeiss pystyy lähettämään CAD/CAM ohjelmilla toimivia malleja asiakkaille ilman lisäkustannuksia. Hyvinä puolina NX:n hyödyntämisessä on NX Openin käytämisen mahdollisuus sekä sopivien mallipohjien tekeminen suoraan NX:ään.

Calypson avulla tehtävässä automaattisessa mittausohjelmassa voidaan hyödyntää ohjelman ominaisuutta luoda radat automaattisesti. Mallipohjien kyvykkyys pitää selvittää turva-alueiden ja resurssien suhteen mutta mallipohjia voi luoda Calypsoon ilman lisäkustannuksia. Joka tapauksessa nollapisteen ja perussuuntauksen automatisointia ei voi tehdä Calypsossa joten ne tarvitsee todennäköisesti tuoda DMIS muodossa NX:stä. Kokonaan automaattiseen ohjelmantekoon tarvitaan PMI, NX Openilla tehty Journal, mallipohja, Inspection osuus sekä DMIS ominaisuus. Lähes automaattiseen ohjelmantekoon riittää PMI ja Calypson mallipohja. Parhaimman tavan löytämiseksi on syytä hankkia ilmaiset testilisenssit molempien vaihtoehtojen testaamiseen, jonka aikana pystytään tekemään päätöksiä.

Suunnittelutiedon muutokset: Muutoksista tulee ilmoittaa kaikille muutoksia tarvitsevan datan tekijöille, jotta systemaattinen muutoksentarve ennaltaehkäistään. Ilmoittaminen voidaan hoitaa työntekijöiden kommunikoinnilla tai tekemällä työkalun, joka ilmoittaa syntyneestä muutoksesta aikaisemmin datan parissa mallintaneille työntekijöille automaattisesti. Työkalun luominen voi maksaa muutamia tuhansia euroja. Tämän lisäksi muutokset tarvitsee tehdä myös aikaisempiin ohjelmiin, jotta MRL olisi aina ajan tasalla. Muutoksien tekeminen hoituu ohjelmaa osaavan käyttäjän toimesta ja vie aikaa todennäköisesti kymmeniä minuutteja per kappale.

Tiedon tallentaminen sekä siirtäminen: Kaiken mahdollisen datan tallentaminen Teamcenteriin voi olla liian suuri lyhyen aikavälin investointi mutta esimerkiksi NC-koodin ja mittausraportin voisi sisällyttää siihen liittyvän nimikkeen tiedostoihin. NC-koodeja käsitellään MMS:ssä mutta niitä ei ole välttämätöntä säilyttää siellä. Esimerkkiratkaisuna MMS:n käyttäminen muutetaan siten, että muokattava tai tarkasteltava NC-koodi ladattaisiin Teamcenteristä. Automaattinen tiedonsiirto läpi prosessin vaatii ohjelmoinnista tietävän henkilön. Tätä voisi lähestyä palkkaamalla yhden ohjelmoijan vuodeksi ja tutkimalla siitä saatuja tuloksia. Kustannukset olisivat kymmeniä tuhansia euroja. Halvemalla pääsisi hyödyntämällä opiskelijaa ja teettämällä aiheeseen liittyen opinnäytetyön mutta tuloksia saataisiin opinnäytetyön rajaamisen ehdoilla.