

Kappaleen tunnistus cobotin voima-anturoinnin avulla

Konetekniikan tutkinto-ohjelma / Turun yliopisto

Kandidaatintutkielma

Toni Savonen

1.6.2026

Turku

Turun yliopiston laatu järjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu

Turnitin OriginalityCheck -järjestelmällä.

Kandidaatintutkielma

Koulutusohjelma, oppiaine: Konetekniikan tutkinto-ohjelma, Konetekniikka

Tekijä(t): Toni Savonen

Otsikko: Kappaleen tunnistus cobotin voima-anturoinnin avulla

Ohjaaja(t): Prof. Wallace Moreira Bessa, Jarno Salminen

Sivumäärä: 27

Päivämäärä: 1.6.2026

Tässä kandidaatintutkielmassa tutkittiin kappaleiden tunnistamista yhteistyörobotin (cobotin) voima-anturoinnin avulla ilman konenäköjärjestelmää. Työn tavoitteena oli selvittää, voidaanko kappaleen muotoa ja orientaatiota tunnistaa hyödyntämällä cobotin voima-antureiden mittaamaa dataa sekä arvioida menetelmän toimivuutta erilaisissa testausolosuhteissa.

Työn teoriaosuudessa käsiteltiin yhteistyörobottien toimintaperiaatteita, voima- ja momenttiantureiden rakennetta sekä mittaussignaalin käsittelyä. Lisäksi tarkasteltiin yhteistyörobottien turvallisuuteen liittyviä vaatimuksia erityisesti ISO/TS 15066 -standardin näkökulmasta.

Kokeellisessa osuudessa suoritettiin kolme testimenetelmää Rokae-cobotilla. Ensimmäisessä testissä havaittiin, että irtonaisten ja kevyiden kappaleiden tunnistaminen ei ollut mahdollista, koska cobotin pieninkin havaittava voima aiheutti kappaleen liikkumisen ennen kontaktin tunnistamista. Toisessa testissä kappaleet kiinnitettiin mekaanisesti alustaan, jolloin cobotti pystyi tunnistamaan kappaleiden muodon ja orientaation luotettavasti kontaktipisteiden perusteella. Kolmannessa testissä tunnistusta laajennettiin z-akselin suuntaan sekä tutkittiin alustan materiaalin ja kappaleen jäykkyyden vaikutusta mittaustuloksiin.

Tulokset osoittivat, että voima-anturointiin perustuva kappaleentunnistus on mahdollista silloin, kun kappale on riittävän vakaa eikä pääse liikkumaan mittauksen aikana. Menetelmän tarkkuuteen vaikuttavat erityisesti kappaleen kiinnitys, alustan kitka, cobotin nopeus sekä materiaalien jäykkyys. Työn perusteella voidaan todeta, että voimaohjattu tunnistus tarjoaa käyttökelpoisen vaihtoehdon tilanteissa, joissa konenäön käyttö on haastavaa, mutta tarkempia sovelluksia varten tarvittaisiin erillinen ranteeseen asennettava voima- ja momenttianturi.

Sisällysluettelo

1	Johdanto	4
2	Yhteistyörobotit ja voima-/momenttianturit	5
2.1	Cobotin toiminnan perusteet	5
2.2	Voima- ja momenttiantureiden toiminta	5
2.3	Signaalin käsittely ja kohinan vaimennus	7
3	Yhteistyörobotin turvallinen käyttö	9
3.1	Syventyminen ISO/TS 15066 standardiin	9
3.2	Turvallisuus Rokae xCore -ohjaimessa	11
4	Käytännön testaus	13
4.1	Johdanto testaukseen	13
4.2	Ohjelman toiminta	14
4.3	Testimenetelmät ja tulokset	16
4.3.1	Ensimmäinen testimenetelmä	16
4.3.2	Toinen testimenetelmä	17
4.3.3	Kolmas testimenetelmä	19
5	Johtopäätökset	21
	Lähteet	22
	Liitteet	23
	Liite 1. Rokae-cobotin kappaleentunnistuksen koodi (Ohjelmoinut: Ganesh Goutham)	23

1 Johdanto

Yhteistyörobotit eli cobotit on suunniteltu toimimaan turvallisesti samassa työtilassa ihmisen kanssa. Toisin kuin perinteiset teollisuusrobotit, jotka vaativat usein eristetyn työskentelyalueen, cobotit pystyvät havaitsemaan ympäristöään ja reagoimaan ulkoisiin voimiin. Tämä kyky perustuu monissa coboteissa moottorin virtamittaukseen, mutta tässä työssä käsitellään cobotteja, jotka havaitsevat ympäristöään voima- ja anturoinnin avulla. Antureiden ensisijainen tarkoitus on taata turvallisuus, mutta samaa mittausdataa voidaan hyödyntää myös muihin tehtäviin. Esimerkiksi kappaleiden muotoa, sijaintia ja asentoa voidaan arvioida pelkän kosketuksen perusteella ilman erillistä konenäköä.

Tässä työssä tutkitaan kappaleiden tunnistamista cobotin voima-anturoinnin avulla. Tavoitteena on selvittää, miten antureita käytetään kappaleen muodon ja orientaation tunnistamiseen ja miten erilaiset testausolosuhteet vaikuttavat mittaustarkkuuteen. Työn teoriaosuudessa käydään läpi cobottien perustoiminta, antureiden rakenne ja signaalinkäsittely. Lisäksi käsitellään turvallisuusvaatimuksia erityisesti ISO/TS 15066 -standardin asettamien voimarakojen pohjalta.

Tutkielman kokeellisessa osuudessa tunnistusta testataan käytännössä Rokae-cobotilla. Testeissä selvitetään, miten esimerkiksi kappaleen kiinnitys, materiaalit ja cobotin asetukset vaikuttavat kerättyyn voimadataan ja tunnistuksen onnistumiseen.

2 Yhteistyörobotit ja voima-/momenttianturit

2.1 Cobotin toiminnan perusteet

Cobotti määritellään ISO/TS 15066 standardin mukaan robotiksi, joka on suunniteltu suoraan vuorovaikutukseen ihmisen kanssa [1]. Cobotit voivat olla teollisuus-, ammattilais- tai palvelurobotteja, mutta teollisessa käytössä ne toimivat yhteisessä työtilassa ihmisten kanssa tuotannon aikana. Cobottien tavoitteena on yhdistää ihmisen joustavuus ja ongelmanratkaisukyky robotin voimaan, tarkkuuteen ja kestävyYTEEN. Cobotin työskentely ihmisen kanssa vapauttaa ihmisen kyvyn suorittaa muita haastavampia tehtäviä samanaikaisesti [2].

Cobotin ja ympäristön välinen vuorovaikutus perustuu kykyyn mitata ja tulkita ulkoisia voimia ja momenteja. Voima- ja momenttianturoinnin avulla cobotti pystyy havaitsemaan käsiteltäviin kappaleisiin ja ympäristöön kohdistuvat vuorovaikutusvoimat reaaliaikaisesti. Näiden mittausten perusteella cobotti voi mukauttaa toimintaansa ja parantaa käsittelyn tarkkuutta. Voima- ja momenttianturoinnin merkitys ei rajoitu pelkästään kosketustilanteiden havaitsemiseen, vaan sen avulla cobotti voi myös päätellä käsiteltävien kappaleiden ominaisuuksia, kuten massaa, jäykkyyttä ja kontaktipisteitä. Tämä mahdollistaa cobotin toiminnan monimutkaisemmissa tehtävissä, joissa pelkkä paikkatietoon perustuva ohjaus ei ole riittävä [3].

2.2 Voima- ja momenttiantureiden toiminta

Cobottiin voidaan asentaa erilaisia antureita. Anturit voivat olla nivelkohtaisia tai ranteeseen asennettuja. Nivelkohtaisten anturien etuna on kyky mitata voimia epäsuorasti nivelten momenttien perusteella, jolloin erillisiä ulkoisia antureita ei

tarvita. Toisaalta nivelkohtainen mittaus ei aina tarjoa yhtä tarkkaa tietoa kontaktivoimien suunnasta ja sijainnista kuin suoraan ranteeseen asennettu anturi. [3]

Voimaperusteisissa tehtävissä tarvitaan tyypillisesti kuusi voiman komponenttia, jotka koostuvat kolmesta lineaarisesta voimasta ja kolmesta momentista. Tämä mahdollistaa täydellisen kontaktivoiman määrittämisen cobotin ja ympäristön välillä. Tämän vuoksi voima- ja momenttianturit sijoitetaan usein cobotin ranteeseen, jolloin ne mittaavat suoraan työkalun ja ympäristön välisiä vuorovaikutusvoimia. Vaihtoehtoisesti voimia voidaan arvioida myös nivelmomenttien perusteella, mutta tällöin mittaus on epätarkempaa. [3]

Osien tunnistamisen kannalta anturin sijainti ja mittaustapa määrittelevät sen, millaisia fysikaalisia ominaisuuksia kappaleesta voidaan havaita. Ranteeseen asennettava kuusiakselinen voima- ja momenttianturi perustuu useimmiten venymäliuska-antureihin, jotka muuttavat mekaanisen muutoksen suoraan sähköiseksi signaaliksi. Anturi koostuu tyypillisesti elastisesta rakenteesta, johon kohdistuvat voimat ja momentit aiheuttavat pieniä muutoksia. Näitä muodonmuutoksia mitataan rakenteeseen sijoitettujen venymäliuskojen avulla, joiden sähköinen resistanssi muuttuu suhteessa venymään. [3], [4]

Mitatut signaalit ovat yleensä hyvin pieniä, minkä vuoksi niitä täytyy käsitellä ennen jatkokäyttöä. Lisäksi anturi vaatii kalibroinnin, jonka avulla mitatut signaalit voidaan muuttaa fysikaalisiksi suureiksi, kuten voimiksi ja momenteiksi. Koska venymäliuskaan vaikuttaa samanaikaisesti useita voimia ja momenteja, mittaustulokset erotellaan toisistaan kalibrintimatriisin avulla, mikä mahdollistaa yksittäisten voimien ja momenttien määrittämisen. [4]

Voima- ja momenttiantureissa voi esiintyä myös virheitä, joissa yhden voima- tai momenttikomponentin vaikutus näkyy useassa mittauskanavassa. Tämän vuoksi anturin rakenne ja venymäliuskojen sijoittelu suunnitellaan siten, että eri komponentit voidaan erotella mahdollisimman tarkasti. [4]

Tällainen rakenne mahdollistaa kuusiakselisen voimien ja momenttien mittaamisen samanaikaisesti, mikä on edellytys tarkalle kontaktivoiman mittaamiselle.

Mittaustarkkuuteen vaikuttavat myös lämpötilan vaihtelu, materiaalin muoto, hystereesi sekä mittauskohina, jotka on huomioitava anturin suunnittelussa sekä mittaustulosten käsittelyssä. [3], [4]

2.3 Signaalin käsittely ja kohinan vaimennus

Voima- ja momenttiantureiden tuottama data on harvoin suoraan käyttökelpoista cobotin ohjauksessa tai kappaleentunnistuksessa, sillä mittaussignaali sisältää tyypillisesti merkittävän määrän kohinaa. Kohina voi johtua useista lähteistä, kuten anturin sähköisestä häiriöstä, mekaanisesta värähtelystä tai cobotin rakenteesta. Tämän vuoksi mitattua signaalia on käsiteltävä ennen sen hyödyntämistä. [3], [4]

Yksi keskeisimmistä menetelmistä kohinan vaimentamiseen on alipäästösuodatus, jossa signaalista poistetaan korkean taajuuden komponentteja. Tämä perustuu oletukseen, että varsinainen hyödyllinen voimasignaali muuttuu hitaammin kuin kohina. Käytännössä alipäästösuodatus voidaan toteuttaa esimerkiksi digitaalisella suodattimella, joka säilyttää signaalin muodon ilman vääristymiä. Yksinkertaisempi vaihtoehto on liukuva keskiarvo, jossa useiden peräkkäisten mittausarvojen keskiarvo tasoittaa vaihtelua. [5]

Pelkkä suodatus ei kuitenkaan aina riitä. Tällöin voidaan hyödyntää Kalman-suodattimen kaltaisia tilanestimointimenetelmiä, jotka yhdistävät mittausdatan ja järjestelmän mallin tuottaakseen hyvän arvion todellisesta voimasta. Kalman-suodatin soveltuu erityisen hyvin robotiikkaan, koska se pystyy käsittelemään sekä mittauskohinaa että epävarmuutta samanaikaisesti. [6]

Signaalinkäsittelyllä on keskeinen rooli myös kappaleentunnistuksessa. Suodatetusta voimasignaalista voidaan erottaa piirteitä, kuten maksimiarvot, kontaktin alkamiskohta sekä signaalin muutosnopeus. Näiden piirteiden avulla voidaan muodostaa kappaleelle voimaprofiili, joka riippuu kappaleen geometriasta, orientaatiosta ja materiaalista. Esimerkiksi terävä kulma aiheuttaa tyypillisesti äkillisen muutoksen voimasignaalissa, kun taas pyöristetty pinta tuottaa pehmeämmän ja tasaisemman signaalin. [7]

3 Yhteistyörobotin turvallinen käyttö

3.1 Syventyminen ISO/TS 15066 standardiin

Cobotti on suunniteltu toimimaan samassa työtilassa ihmisen kanssa. Tämä tarkoittaa sitä, että työskentelyn on oltava turvallista ja sallittujen voimien on pysyttävä määritetyjen standardien rajoissa. Sallitut cobotin voimarajat ovat kriittinen näkökohta, kun puhutaan ihmisen turvallisuudesta. ISO/TS 15066 määrittelee suurimmat sallitut voima- ja painearvot eri kehon osille siten, että kontaktit eivät aiheuta kipua tai vammoja. Esimerkiksi herkemmille osille, kuten sormille tai käsille, sallitut voimat ovat pienempiä kuin vähemmän herkille osille, kuten selälle tai reisille. [1]

Cobotin ja ihmisen välinen kontakti voidaan jakaa kvaasi-staattiseen (engl. Quasi-Static) ja dynaamiseen (engl. Transient) tilanteeseen. Staattisessa tilanteessa cobotti voi puristaa ihmistä jotakin pintaa vasten, jolloin voima kohdistuu pidemmällä aikavälillä. Dynaaminen tilanne on lyhytaikainen törmäys, jossa vaikutus riippuu esimerkiksi cobotin nopeudesta tai massasta.

Cobotteihin liittyvässä turvallisuussuunnittelussa määritellään suurimmat sallitut kontaktivoimat ihmisen eri kehon osille. Nämä arvot perustuvat biomekaanisiin raja-arvoihin, joiden ylittäminen voi aiheuttaa kipuja tai vammoja. Taulukossa 1 esitetään esimerkkejä näistä raja-arvoista. [1]

Taulukko 1. Biomekaaniset raja-arvot kvaasi-staattiselle ja dynaamisille kontakteille (mukaillen ISO/TS 15066)

Body region	Specific body area		Quasi-static contact		Transient contact	
			Maximum permissible pressure ^a P_s N/cm ²	Maximum permissible force ^b N	Maximum permissible pressure multiplier ^c P_T	Maximum permissible force multiplier ^c F_T
Skull and forehead ^d	1	Middle of forehead	130	130	not applicable	not applicable
	2	Temple	110		not applicable	not applicable
Face ^d	3	Masticatory muscle	110	65	not applicable	not applicable
Neck	4	Neck muscle	140	150	2	2
	5	Seventh neck muscle	210		2	
Back and shoulders	6	Shoulder joint	160	210	2	2
	7	Fifth lumbar vertebra	210		2	2

Taulukosta 1 havaitaan, että sallitut voimat ja paineet vaihtelevat huomattavasti kehon osasta riippuen. Esimerkiksi herkillä alueilla, kuten kasvoissa (engl. face) suurin sallittu staattinen voima on vain 65 N, kun taas selän ja hartioiden (engl. Back and shoulders) alueella sallitaan jopa 210 N voima. Tämä asettaa korkeita vaatimuksia cobotin voimaanturoinnille ja ohjelmiston herkkyydelle, jotta kontakti pystytään havaitsemaan ja pysäyttämään ennen näiden raja-arvojen ylittymistä. Dynaamisen tilanteen voimakertoimista voidaan myös päätellä, että hetkellinen kontakti on huomattavasti turvallisempi ihmisen kehon osille kuin kvaasi-staattinen tilanne.

Taulukossa on myös arvoja, jotka eivät ole sovellettavissa voimamittaukseen. Tämä johtuu siitä, että tilanteet joissa cobotti osuu kriittiseen ruumiinosaan on pyritty välttämään esimerkiksi cobotin liikeradan ohjauksella. Teollisuudessa cobotin ja ihmisen liikeratoja arvioidaan ja voimarajat asetetaan pienimmän mahdollisen arvon mukaan.

Standardi ISO/TS 15066 ei ainoastaan aseta rajoja sallituille voimille, vaan se määrittelee myös tekniset periaatteet, joiden mukaan robottijärjestelmän on valvottava näitä voimia. Keskeinen turvallisuusmenetelmä on voiman ja tehontason rajoittaminen, joka vaatii cobotilta jatkuvaa ja reaaliaikaista kommunikointia sen omien voima-anturien kanssa. Järjestelmän on pystyttävä tunnistamaan poikkeamat suunnitelluista liikkeistä ja reagoitava niihin välittömästi, jotta kontaktista koituvat vauriot pysyvät sallituissa rajoissa. [1]

Turvallisuusasetuksissa määritetty kynnysvoima on kuitenkin aina kompromissi järjestelmän herkkyden ja prosessin sujuvuuden välillä. On huomioitava, että anturin ilmoittama voima ei ole suoraan verrannollinen vammautumisriskiin, sillä vamma riippuu usein paikallisesta paineesta. Paineen suuruus määräytyy kontaktipinnan pinta-alan mukaan, mikä korostaa cobotin työkalun muotoilun merkitystä voimanmittauksen rinnalla. Lisäksi dynaamisissa kontaktitilanteissa järjestelmän vasteaika on kriittinen tekijä. Tämä tarkoittaa sitä, että esimerkiksi signaalinkäsittelyssä käytettävät suodatusmenetelmät eivät saa aiheuttaa sellaista viivettä, joka hidastaisi cobotin reaktiota hätätilanteessa.

3.2 Turvallisuus Rokae xCore -ohjaimessa

Käytännön sovelluksissa ISO/TS 15066 -standardin vaatimukset toteutetaan cobotin ohjausjärjestelmän sisäänrakennetuilla turvatoiminnoilla. Työssä käytetty Rokae-cobotin xCore-ohjausjärjestelmä hyödyntää dynaamista voiman ja momentin valvontaa, joka mahdollistaa kymmenportaisen törmäyksentunnistuksen (engl. Collision Detection) herkkyden säädön. Järjestelmä valvoo jatkuvasti kunkin nivelen moottorin vääntömomenttia ja vertaa sitä cobotin dynaamiseen malliin, mikä mahdollistaa ulkoisten kontaktivoimien erottamisen cobotin omasta liikkeestä johtuvista voimista. [8]

Manuaalin mukaan järjestelmä tukee useita hallittuja pysäytyskategorioita, joista kappaleentunnistuksen kaltaisissa prosesseissa keskeinen on Katgoria 2 -pysäytys. Tässä tilassa cobotti pysäyttää liikkeensä välittömästi kontaktin havaitsemisen jälkeen, mutta säilyttää virrat moottoreissa ja pitää paikkansa, mikä mahdollistaa tarkan paikkatietojen lukemisen ja prosessin jatkamisen ilman järjestelmän uudelleenkäynnistystä. Cobotin kyky määrittää tämän kontaktipisteen sijainti peruskoordinaatistossa perustuu muunnosmatriiseihin. Pisteen sijainti lasketaan cobotin nivelkulmien ja työkalun mittojen perusteella matriisitulona:

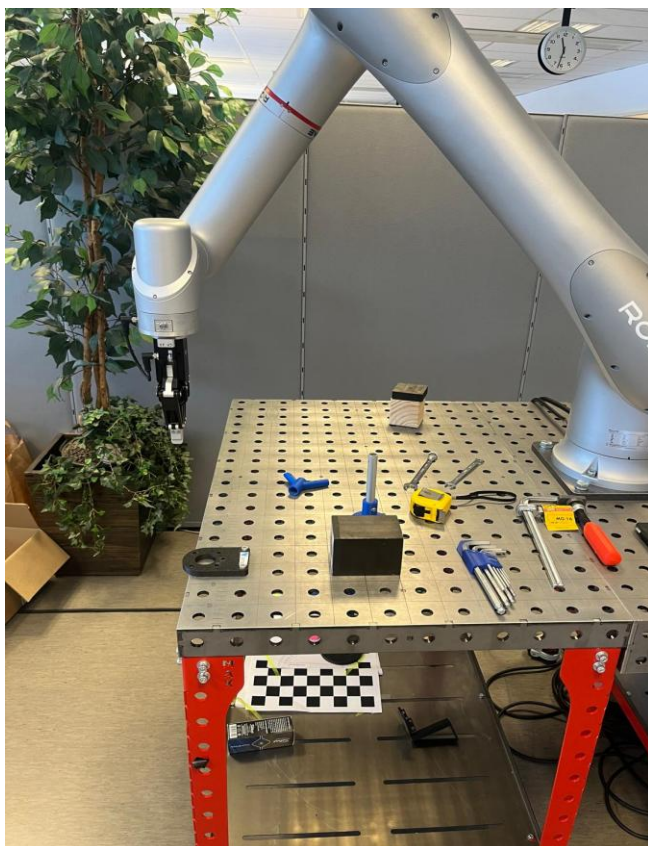
$$P_{base} = T_{base}^{tool} P_{tool} \quad (1)$$

, missä muunnosmatriisi T kuvaa laipan asentoa ja P on manuaalin mukaisesti määritetty työkalun keskipiste (engl. Tool Center Point (TCP)). Muunnosmatriisin hyödyntäminen on kriittinen osa robottiturvallisuutta, sillä se on se matemaattinen kieli, jolla cobotti ymmärtää oman sijaintinsa suhteessa ihmiseen ja ympäristöön. Muunnosmatriisin avulla cobotti laskee välittömästi, mihin suuntaan sen on liikutettava niveliään vähentääkseen kontaktipainetta. Jos matriisilaskenta on virheellistä, cobotti saattaisi hätätilanteessa liikkua vahingossa ihmistä kohti sen sijaan, että se vetäytyisi pois. [8], [9]

4 Käytännön testaus

4.1 Johdanto testaukseen

Tässä työssä testattiin cobotin kykyä tunnistaa erilaisia kappaleita ja niiden orientaatiota voima-anturien mittaaman datan perusteella. Voimien mittausdatan kerääminen onnistui cobotin jokaiseen niveleen integroitujen voima-antureiden avulla. Testauksen tavoitteena oli selvittää voiko cobotti tunnistaa tietyn ohjelman avulla kappaleen muodon sekä sen orientaation ilman visuaalista avustusta. Kokeessa hyödynnettiin eri menetelmiä liittyen kohdekappaleen kiinnitykseen sekä cobotin ranteen tarttujan asentoon sekä muotoon. Testeissä tarttujan tilalla käytettiin myös erilaisia yksinkertaisia kosketinkärkiä kuten pyöreää metallitankoa. Lisäksi suoritettiin testejä, joissa kohdekappale oli irtonainen eikä kiinnitettynä alustaan, jotta voitiin arvioida miten kappaleen mahdollinen liike vaikuttaa tuloksiin. Testaus suoritettiin Uudessakaupungissa Vertekin toimistotiloissa, jossa Rokae-cobotti oli pultattuna metallipöytään kiinni.



Kuva 1. Havainnollistava kuva testausympäristöstä. Kuva: Toni Savonen

Työssä tarkasteltiin erityisesti sitä, miten kappaleen kosketuksesta syntyvät voimamuutokset vaikuttavat cobotin mittaamaan dataan. Cobotti suoritti ohjelman mukaisia ennalta määritettyjä liikkeitä, joissa tarttuja kosketti kappaletta eri kohdista ja suunnista x-, y- ja z-akselilla. Näiden kosketusten aikana syntyneitä voimia hyödynnettiin kappaleen muodon ja orientaation arvioimisessa. Mittausdata tallennettiin analysointia varten, ja sitä hyödynnettiin piirto-ohjelman avulla, josta pystyttiin pisteiden avulla muodostamaan kappaleen geometriaa vastaava kuvaaja. Testeissä käytettiin muutamia eri muotoisia kappaleita, joiden tarkoituksena oli tuottaa toisistaan poikkeavaa mittausdataa, ja arvioida ohjelman toimivuutta eri muotoisilla kappaleilla.

Edellä kuvattu testausympäristö ja mittausperiaate muodostivat perustan kokeiden toteutukselle. Testeissä pyrittiin havainnoimaan, miten erilaiset kosketustilanteet sekä kappaleen kiinnitys ja orientaatio vaikuttavat cobotin mittaamaan voimadataan. Seuraavissa alaluvuissa kuvataan tarkemmin käytetyt testimenetelmät, mittausjärjestelyt sekä kokeiden suorittamiseen liittyvät vaiheet.

4.2 Ohjelman toiminta

Tässä työssä cobotin ohjelmana käytettiin (Ganesh Goutham) kehittämää RL-kielistä ohjelmaa. Kokeissa käytetty ohjelma käytti voimaohjausta törmäyksen, kappaleen muodon ja orientaation tunnistamiseen. Ohjelma perustui cobotin ohjattuun liikkeeseen ja kosketushavainnointiin, jossa cobotti liikkui tietyssä suunnassa kunnes

voima-anturi havaitsi kontaktin kappaleen pinnan kanssa. Havaitut kontaktipisteet tallennettiin cobotin koordinaatistossa jatkoanalyysiä varten.

Ennen mittauksen aloittamista cobotti siirtyy ennalta määritettyyn paikkaan. Tämän jälkeen voima-anturi alustetaan ja kalibroidaan, jotta mittaustuloksissa ei tule offsettiä. Kalibrointi suoritetaan käynnistämällä voimaohjaus ja suorittamalla anturin virhekalibrointi, pysäyttämällä ohjaus ennen varsinaista mittausta. Varsinainen mittaus suoritetaan iteroimalla cobotin liikettä x-, y ja z-tasossa siten, että jokaisella syklillä cobotti siirtyy ennalta määritetyn matkan verran eteenpäin. Cobotti liikkuu määritetyn matkan verran eteenpäin koordinaatiston x-tasossa kunnes voima-anturi havaitsee ennalta määritetyn voimarajan ylittymisen. Tämä voimarajan ylittyminen tulkitaan kontaktiksi. Kun kosketus havaitaan, cobotin sijainti luetaan ja tallennetaan. Tämä sijainti vastaa kappaleen pintaa kyseisessä mittauslinjassa. Tallennettu piste lisätään taulukkoon, johon kerätään kaikki mittauksen aikana havaitut kosketuspisteet x- ja y-akselilla.

x-suunnan kosketuksen jälkeen suoritetaan tarkentava mittaus y-suunnassa. Tämä mittaus käyttää samanlaista menetelmää kuin x-suunnassa, mutta tällä kertaa ohjelma sallii liikkeen vain y-suunnassa. Mittausprosessin tuloksena saadaan joukko pisteitä, jotka kuvaavat kappaleen reunaa tai pintaa robotin työkordinaatistossa. Lopuksi cobotti käy kaikki tallentetut pisteet läpi ja siirtyy niihin eri z-tasossa. Tätä vaihetta voidaan käyttää esimerkiksi myöhempiin työstö- tai hitsausoperaatioihin.

Voimaohjauksen aikana cobotin jäykkyys ja sallittu nopeus määritellään kullekin akselille erikseen. Tämä mahdollistaa cobotin pienen taipumisen kosketustilanteessa, lisätäkseen toiminnan turvallisuutta ja vähentääkseen törmäysriskiä. Samalla liikkeen maksiminopeutta rajoitetaan, mikä parantaa mittauksen tarkkuutta ja turvallisuutta. Tätä maksimiarvoa testattiin 2 mm/s ja 10 mm/s välillä ja todettiin, että hitaammalla liikkeellä saadaan selvästi tarkempia tuloksia.

Menetelmän etuna on, että cobotti pystyy havaitsemaan kontaktin luotettavasti tilanteissa, jossa kappaleen sijainti tai geometria ei ole etukäteen tiedossa. Tämä tekee menetelmästä käyttökelpoisen esimerkiksi kohdistuksessa, kappaleiden paikannuksessa sekä robottipohjaisessa mittaus- ja kalibrointitehtävissä.

4.3 Testimenetelmät ja tulokset

4.3.1 Ensimmäinen testimenetelmä

Työssä käytettiin kolmea eri testimenetelmää mittamaan kappaleen tunnistusta voima-anturien avulla. Kaikissa menetelmissä hyödynnettiin RL-ohjelmaa ja Rokae-cobotin omaa datapaneelia. Ensimmäisessä testimenetelmässä kokeita tehtiin irtonaiseen pöydällä olevaan kuution muotoiseen puukappaleeseen. Menetelmässä cobotti lähestyi kappaletta ohjelman ohjaamalla tavalla tunnistuen ensin x-tason ja tämän jälkeen y-tason. Kokeen alussa huomattiin keskeinen ongelma cobotin tarttujan käyttämän voiman suuruudessa suhteessa kappaleen tukivoimaan. Kun cobotti lähestyi palikkaa tietyllä voimalla, se aiheutti kappaleen liikkumisen, jolloin ohjelma ei toiminut halutulla tavalla. Kokeita ja ohjelmamuutoksia tekemällä päädyttiin siihen tulokseen, että cobotin pienin mahdollinen tuottama voima oli 13 N. Kokeen tavoitteena oli, että kohdekappale pysyy paikallaan cobotin lähestyessä sitä, jolloin ohjelma toimii halutulla tavalla ja koordinaattipisteet voidaan havaita. Tämä edellyttää, että cobotin kohdistama voima ei ylitä kappaleen ja alustan välistä suurinta lepokitkavoimaa. Tämä voidaan esittää ehdolla

$$F_{cobotti} \leq \mu_s mg, \quad F_{cobotti} \geq 13 \text{ N} \quad (2)$$

missä $F_{cobotti}$ on cobotin kohdistama voima, μ_s lepokitkakerroin, m massa ja g putoamiskiihtyvyys. Mikäli ehto (2) ei täyty, kappale alkaa liukua pinnalla eteenpäin

aiheuttaen tilanteen, jossa cobotin tarttuja seuraa kappaletta eikä tunnista sitä vaan työntää sitä voimalla $F_{cobotti}$ eteenpäin.

Ensimmäisen testimenetelmän avulla voidaan todeta, että kyseisellä cobotilla irtonaisten kevyiden kappaleiden tunnistaminen ei ole mahdollista. Tämä johtuu siitä, että cobotin tuottama pieninkin voima ylittää kappaleen paikallaan pysymisen kannalta sallitun raja-arvon eikä ohjelma ehdi reagoida ennen kuin kappaleen lepokitka ylittyy. Näin ollen ohjelma ei pysty käsittelemään tilanteita, joissa kohdekappaletta ei ole tuettu tai kiinnitetty riittävän tukevasti. Tulosten perusteella voidaan päätellä, että luotettava kappaleentunnistus kyseisillä laitteilla ja ohjelmalla edellyttää joko suurempaa kitkaa kappaleen ja alustan välillä tai kappaleen mekaanista kiinnittämistä testin ajaksi. [10]

4.3.2 Toinen testimenetelmä

Toisessa testimenetelmässä kohdekappaleina käytettiin samaa kuutionmuotoista puukappaletta sekä muovista kappaletta, joka muodostuu neliöstä, jonka yläreunassa oli puoliympyrä. Tässä testimenetelmässä kappaleet kiinnitettiin ruuvipuristimella metallipöytään, jotta ehto (2) toteutuu eikä kappale pääse liukumaan pöydän pinnalla. Menetelmässä cobotin kosketuspintana hyödynnettiin tarttujan sijasta myös pyöreää metallitankoa. Menetelmän tavoitteena oli saada selkeämpi ymmärrys cobotin kyvystä tunnistaa kohdekappaleen muoto, koko ja orientaatio annetun ohjelman avulla.

Kappaleen mekaaninen kiinnitys aiheutti testien onnistumisen ja cobotin kyvyn tunnistaa kappaleiden muoto. Testissä cobotti lähestyi kappaletta ohjelman määräämällä tavalla ja mittasi havaituista kontakteista koordinaattipisteet (x, y) . Näistä pisteistä voidaan muodostaa suoran yhtälöt, joiden leikkauspisteiden rajaama alue antaa kuvan kappaleen pinnan muodosta. Suoran yhtälö voidaan esittää muodossa:

$$y = kx + b \quad (3)$$

missä k on suoran kulmakerroin ja b suoran vakiotermi. Kulmakerroin k voidaan laskea kahden pisteen (x_1, y_1) ja (x_2, y_2) avulla seuraavasti:

$$k = \frac{\Delta y}{\Delta x} \quad (4)$$

Kun kulmakerroin k on tiedossa, voidaan ratkaista vakiotermi b käyttämällä jompaakumpaa pistettä:

$$b = y_1 - kx_1 \quad (5)$$

Näin muodostetut suorat kuvaavat kappaleen eri pinnan osia. Kahden tai useamman suoran leikkauspisteiden avulla voidaan määrittää alue, joka kuvastaa kappaleen muotoa. Leikkauspisteiden avulla voidaan myös laskea kappaleen sivujen pituudet, kulmat sekä kaarevien osien säteet. [9]

Mittausdataa analysoitiin GeoGebra-piirto-ohjelmaa sijoittamalla cobotin mitaamat pisteet koordinaatistoon. Suorien yhtälöt määritettiin ohjelman avulla visuaalisesti, mikä vähensi merkittävästi käsin laskemista ja helpotti tulosten havainnointia. Graafisen ohjelman etuna oli myös kokonaisuuden hahmottaminen ja tulosten vertailu todelliseen kappaleeseen. Havainnoinnin perusteella mitatut koordinaattipisteet ja niistä muodostetut suorien yhtälöt vastasivat kappaleen todellista muotoa. Tästä voidaan johtopäätöksenä todeta, että cobotin kyky tunnistaa kiinnitetty kappale oli hyvä.

4.3.3 Kolmas testimenetelmä

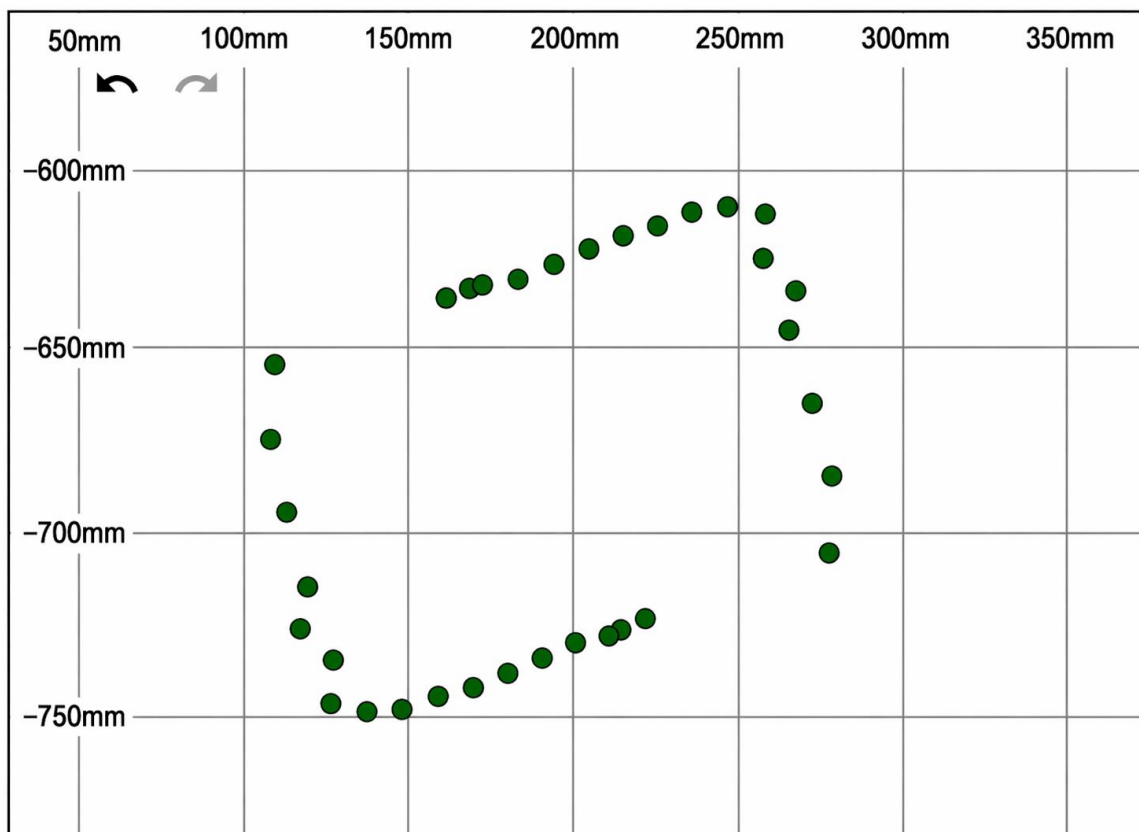
Kolmannessa testimenetelmässä lisättiin z-akselin suuntainen tunnistaminen ja muutettiin testialustan materiaalia. Tämän vaiheen tavoitteena oli muuttaa mittaus kaksiulotteisesta reunojen mittauksesta kolmiulotteiseksi kappaleen hahmottamiseksi ja arvioida, miten alustan ominaisuudet vaikuttavat mittauksen vakauteen. Z-akselin tunnistaminen tehtiin muuttamalla ohjelmaa tekemään myös z-akselin haku osana mittausprosessia. Muuten tunnistaminen toimi edellä mainitulla tavalla.

Voimaohjatussa liikkeessä cobotin ei tule olla täysin jäykkä, jotta kontakti voidaan havaita turvallisesti. Järjestelmän dynaaminen vaste noudattaa Hooken lakia:

$$F = k\Delta x \quad (6)$$

, missä k on ohjelmassa määritetty jäykkyyserroin (engl. stiffness) ja Δx on kontaktista aiheutuva poikkeama. Testeissä havaittiin, että optimoimalla jäykkyyservot sekä cobotin nopeus voitiin minimoida cobotin aiheuttama iskuvoima. Tämän muutoksen lisäksi testialustan muuttaminen metallialustasta kumialustaan auttoi kappaletta pysymään paremmin paikoillaan. Kumialusta lisäsi alustan kitkakerrointa riittävästi, jotta alustan muuttaminen yhdistettynä voiman minimointiin mahdollisti ehdon (2) täyttymisen. [8], [9]

Kolmannessa testausmenetelmässä hyödynnettiin myös pehmeämpiä materiaaleja kuten pahvia havainnoidakseen materiaalin jäykkyyden vaikutusta mitattuihin arvoihin. Testeissä huomattiin, että materiaalin vaikutus näkyi selvästi mittausdatassa ja havainnoitessa tuloksia, muoto ei ollut yhtä särnä kuin jäykemmällä kappaleella.



Kuva 2. GeoGebralla tehty kuva pahlilaatikon voimadatasta.

5 Johtopäätökset

Työn tavoitteena oli testata, miten cobotin keräämää voimadataa voidaan käyttää kappaleen tunnistamiseen ilman konenäköä. Teoriaosuudessa käsiteltiin voima- ja momenttianturoinnin roolia turvallisuudessa ja kappaleiden ominaisuuksien arvioinnissa, sekä ISO/TS 15066 -standardin asettamia voimarojoja ihmisen ja robotin välisissä kontakteissa.

Käytännön testeissä voimadata kerättiin Rokae-cobotin nivelkohtaisilla antureilla. Testit osoittivat, että nivelmomenteihin perustuvalla mittauksella pystytään havaitsemaan kontakteja ja toteuttamaan yksinkertainen kappaleentunnistus, mutta menetelmässä on selviä rajoitteita. Koska anturointi mittaa voimia epäsuorasti nivelistä, tuloksiin vaikuttavat robotin oma kitka, nivelten jäykkyys ja liikkeen aiheuttamat dynaamiset voimat.

Eryisesti irtonaisten ja kevyiden kappaleiden tunnistaminen todettiin nykyisellä laitteistolla haastavaksi. Cobotin vaatima minimivoima kontaktin havaitsemiseksi sai kappaleen liukumaan ennen kuin ohjelma ehti reagoida. Tunnistus toimi parhaiten silloin, kun kappale oli kiinnitetty tukevasti alustaan ja kontaktit tehtiin hitaalla nopeudella. Mittaustarkkuutta paransivat myös alustan suuri lepokitka ja jäykemmät materiaalit.

Nivelanturointi riittää yksinkertaisten kontaktien havaitsemiseen, mutta tarkempiin robotiikkasovelluksiin vaadittaisiin erillinen ranteeseen asennettava kuusiakselinen voima- ja momenttianturi. Ranneanturi mittaa voimat suoraan työkalun kärjestä, jolloin robotin rakenteesta johtuvat virheet jäävät pois. Tästä huolimatta voimaohjattu menetelmä on käyttökelpoinen ratkaisu sellaisissa teollisuuden prosesseissa, kuten hitsauksessa tai hionnassa, joissa konenäköä on vaikea käyttää esimerkiksi huonon valaistuksen, heijastuksen tai lian vuoksi.

Lähteet

- [1] ”ISO/TS 15066:2016”, ISO. Viitattu: 1. huhtikuuta 2026. [Verkossa]. Saatavissa: <https://www.iso.org/standard/62996.html>
- [2] M. Fernandez-Vega, D. Alfaro-Viquez, M. Zamora-Hernandez, J. Garcia-Rodriguez, ja J. Azorin-Lopez, ”Transforming Robots into Cobots: A Sustainable Approach to Industrial Automation”, *Electron. Basel*, vol. 14, nro 11, s. 2275–, 2025, doi: 10.3390/electronics14112275.
- [3] B. Siciliano ja O. Khatib, *Springer Handbook of Robotics*. Cham, SWITZERLAND: Springer International Publishing AG, 2016. Viitattu: 1. huhtikuuta 2026. [Verkossa]. Saatavissa: <http://ebookcentral.proquest.com/lib/kutu/detail.action?docID=6212326>
- [4] M. Y. Cao, S. Laws, ja F. R. y Baena, ”Six-Axis Force/Torque Sensors for Robotics Applications: A Review”, *IEEE Sens. J.*, vol. 21, nro 24, s. 27238–27251, joulu 2021, doi: 10.1109/JSEN.2021.3123638.
- [5] A. V. Oppenheim, R. W. Schafer, ja T. G. Stockham, ”Nonlinear filtering of multiplied and convolved signals”, *Proc. IEEE*, vol. 56, nro 8, s. 1264–1291, elo 1968, doi: 10.1109/PROC.1968.6570.
- [6] A. Valibeygi, M. H. Balaghi I., ja K. Vijayaraghavan, ”A comparative study of Extended Kalman Filter and an optimal nonlinear observer for state estimation”, teoksessa *2017 American Control Conference (ACC)*, touko 2017, s. 5211–5216. doi: 10.23919/ACC.2017.7963764.
- [7] M. W. Spong, S. Hutchinson, ja M. Vidyasagar, *Robot Modeling and Control (2nd Edition)*, Second edition. Hoboken, N.J: John Wiley & Sons, 2020.
- [8] Rokae, ”xCore User Manual V3.1_B_EN”, Rokae, 2023. [Verkossa]. Saatavissa: https://static.rokae.com/Downloads/Manual/xCore_UserManual_V3.1_B_EN.pdf
- [9] Siciliano Bruno, Sciavicco Lorenzo, Villani Luigi, ja Oriolo Giuseppe, *Robotics: Modelling, Planning and Control*. Springer Science & Business Media, 2010.
- [10] R. Volpe ja P. Khosla, ”A theoretical and experimental investigation of explicit force control strategies for manipulators”, *IEEE Trans. Autom. Control*, vol. 38, nro 11, s. 1634–1650, marras 1993, doi: 10.1109/9.262033.

Liitteet

Liite 1. Rokae-cobotin kappaleentunnistuksen koodi (Ohjelmoinut: Ganesh Goutham)

```

VAR robtarget p_collision;
VAR robtarget pointss[25];
VAR robtarget pcc;
VAR bool contact = FALSE;
VAR bool gg;
VAR int hh = 0;
VAR robtarget p_new;

GLOBAL PROC main()
  MoveL(p1,v300,z50,tool0,wobj0);
  FcInit (tool0, wobj0, 2);
  SetControlType (1);
  SetCartCtrlStiffVec (2000,5000,5000,1000,1000,1000);
  FcStart;

  // ---- Zero torque sensor ----

  For(int index from 0 to 100 step 1)
    p_new = p1;
    p_new.trans.y = p1.trans.y - (20*index);
    probex(p_new);
  Endfor

ENDPROC

FUNC int probex(robtarget p3)

  VAR INT cc = 0;
  VAR INT xx = 10;
  While(cc < 200 )
    // Move 5 mm in +X tool direction
    MoveL(RelTool(p3,xx,0,0,0,0),v50,z50,tool0,wobj0);

```

```

pcc = CRobT(tool0,wobj0);
if(pcc.trans.x < 338)
    p3.trans.y = -834;
    p3.trans.x = 345;
    MoveL(p3,v1000,z50,tool0,wobj0);
    wait(1);
    SetCartCtrlStiffVec (5000,2000,5000,1000,1000,1000);
    For(int index from 0 to 3 step 1)
        p_new = p3;
        p_new.trans.y = p3.trans.y - (3*index);
        p_new.trans.x = p3.trans.x - (3*index);
        probey(p_new);
    Endfor
    FcStop();
    probez();

    FcStop;

    MoveJ(p_z,v1000,z50,tool0,wobj0);
    For(int indexx from 1 to 25 step 1)
        if(pointss[indexx].trans.x!=0)
            print(indexx, " = ",pointss[indexx]);
        Endif
    Endfor
    MoveJ(p2,v1000,z50,tool0,wobj0);
    For(int inde from 1 to 25 step 1)
        cc=200;
        if(pointss[inde].trans.x!=0)
            pointss[inde].trans.z = 302;
            MoveL(pointss[inde],v1000,z50,tool0,wobj0);
        Endif
    Endfor
Endif
xx = xx+10;
if(CheckForcex())<-13)
    print("ob detected");
    cc = 200;

```

```

    Endif
    cc = cc+1;

Endwhile

// Store contact position
p_collision = CRobT(tool0,wobj0);

print("Contact detected:");

print(p_collision.trans.x,
p_collision.trans.y);
hh = hh+1;
pointss[hh] = p_collision;
return p_collision.trans.x;
ENDFUNC

FUNC int probey(robtarg p4)
    VAR INT c_x = p_collision.trans.x;

    VAR INT ccc = 0;
    VAR INT xxx = 20;

    While(ccc < 200 )
        MoveL(RelTool(p4,0,xxx,0,0,0,0),v50,z50,tool0,wobj0);
        xxx = xxx+20;
        if(CheckForcey())<-13)
            print("ob detected");
            ccc = 200;
        Endif
        ccc = ccc+1;

    Endwhile

    p_collision = CRobT(tool0,wobj0);
    VAR INT c_y = p_collision.trans.y;
    print("Contact detected:");
    hh = hh+1;

```

```

pointss[hh] = p_collision;
print(p_collision.trans.x,
p_collision.trans.y);

```

```
ENDFUNC
```

```
FUNC int probez()
```

```

MoveJ(p3,v1000,z50,tool0,wobj0);
wait(1);
FcInit (tool0, wobj0, 2);
SetControlType (1);
SetCartCtrlStiffVec(5000,5000,2000,1000,1000,1000);
FcStart;

```

```
VAR INT ccc = 0;
```

```
VAR INT xxx = 20;
```

```
MoveJ(p_z,v1000,z50,tool0,wobj0);
```

```
While(ccc < 200 )
```

```
MoveL(RelTool(p_z,0,0,xxx,0,0,0),v50,z50,tool0,wobj0);
```

```
xxx = xxx+20;
```

```
if(CheckForcez(<-13)
```

```
print("ob detected");
```

```
ccc = 200;
```

```
Endif
```

```
ccc = ccc+1;
```

```
Endwhile
```

```
p_collision = CRobT(tool0,wobj0);
```

```
print("Contact detected:");
```

```
print("z coords of the point is = ",p_collision.trans.z);
```

```
ENDFUNC
```

```
FUNC int CheckForcex()
```

```
VAR TorqueInfo tmp_info;
```

```
tmp_info = GetEndToolTorque(tool0, wobj0, 2);
```

```
return tmp_info.cart_torque.m_force[1];
```

```
ENDFUNC
```

```
FUNC int CheckForcey()
```

```
VAR TorqueInfo tmp_info;
```

```
tmp_info = GetEndToolTorque(tool0, wobj0, 2);
```

```
return tmp_info.cart_torque.m_force[2];
```

```
ENDFUNC
```

```
FUNC int CheckForcez()
```

```
VAR TorqueInfo tmp_info;
```

```
tmp_info = GetEndToolTorque(tool0, wobj0, 2);
```

```
print("z axis force value = ", tmp_info.cart_torque.m_force[3]);
```

```
return tmp_info.cart_torque.m_force[3];
```