



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Juho Haikka

# Konenäön hyödyntäminen Delta-robotin liikkeenohjauksessa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Konetekniikka

Insinöörityö

17.5.2019

Tekijä Otsikko	Juho Haikka Konenäön hyödyntäminen Delta-robotin liikkeenohjauksessa
Sivumäärä Aika	39 sivua 17.5.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Konetekniikka
Ammatillinen pääaine	Koneautomaatio
Ohjaaja	Lehtori Antti Liljaniemi
<p>Insinööriyön tilaajana oli Metropolia Ammattikorkeakoulu, ja työn tavoitteena oli kehittää Metropolian koneautomaatiolaboratoriossa oleva Delta-robotti toimimaan vastustajana ilmakiekkopelissä. Delta-robotti on ollut Metropolian tiloissa vähäisellä käytöllä, ja insinööriyön on tarkoitus tehdä uuden sovelluksen myötä robotista houkuttelevampi kohde uusille laboratorioharjoituksille.</p> <p>Työn alussa tutustuttiin Sickin konenäkökameraan ja SOPAS Engineering Tool -ohjelmistolla tehtävään kuvankäsittelyyn. Ohjelmiston avulla kameran ottamasta kuvasta määriteltiin halutut parametrit ja tiedonsiirtoprotokolla. Kiekon liikeradan ennakoimiseksi kehiteltiin matemaattiset kaavat, jotka laskevat robotille koordinaatistopisteen, jossa se torjuu lähestyvän kiekon. Ohjelmakoodin kirjoittamisen jälkeen sovellusta testattiin ensin virtuaaliakseleilla sekä myöhemmin Delta-robotilla.</p> <p>Robotin tehtävänä oli torjua pelipöydällä lähestyvä kiekko, josta otetaan konenäkökameralla kuvia ja näiden kuvien avulla lasketaan kiekon liikerata ja annetaan oikeat koordinaatit robotille kiekon torjumiseksi. Kameran ja robotin välisen kommunikoinnin toteuttaminen TCP/IP:n välityksellä saatiin toimimaan kunnolla vasta työn loppusuoralla, ja näin ollen ilmakiekkosovelluksen ohjelmakoodin testaamiseen jäi aikaa vähemmän kuin oli suunniteltu.</p> <p>Lopputuloksena oli robottisolu, jossa kameralta saatavaa koordinaattidataa vastaanotetaan TwinCATissa ja tätä dataa käytetään osana robotin ohjelmakoodia. Robottia voidaan ajaa manuaalisesti virtuaalisen käyttöpaneelin avulla. Pää tavoitteena ollut automaattinen kiekon torjuminen jäi kesken robotin ohjelmoinnissa ilmenneiden haasteiden takia. Robotti lähtee torjumaan lähestyvää kiekkoa, mutta ajoittain robotille lasketut koordinaatit eivät vastaa kiekon todellisia koordinaatteja ja robotti lähtee torjumaan kiekkoa väärään suuntaan.</p>	
Avainsanat	Konenäkö, liikkeenohjaus, robotiikka, logiikkaohjelmointi

Author Title	Juho Haikka Use of Machine Vision in the Motion Control of the Delta Robot
Number of Pages Date	39 pages 17 May 2019
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Professional Major	Machine Automation
Instructor	Antti Liljaniemi, Senior Lecturer
<p>The objective of this Bachelor's thesis was to improve a Delta robot to act as a player in air hockey games. The Delta robot has been used very little in Metropolia Myyrmäki campus and with the help of this project, the Delta robot will be a more interesting subject for future exercises. This thesis was commissioned by Metropolia University of Applied Sciences.</p> <p>The thesis was started with an introduction to the Sick machine vision sensor and the software SOPAS Engineering Tool which is used in image processing. The required parameters of the pictures taken and the communication protocol are defined with this software. The essential formulas for estimating the air hockey puck's trajectory had to be solved, before it was possible to calculate the exact coordinates for the Delta robot to be able to hit the approaching puck. After the programming was done, it was tested first with virtual axes and then with the Delta robot.</p> <p>In this thesis, a new application was made for the Delta robot, and the application starts when the machine vision sensor takes photos of the air hockey table and detects the moving puck. The coordinates of the puck are transferred via TCP/IP to TwinCAT where the programming is done. Consequently, in TwinCAT it was possible to calculate the right coordinates for the robot to hit the puck. The communication between the machine vision sensor and TwinCAT was a challenging task and it was completed at the end of the project. Because of this, there was not time to test the actual code as much as it was planned.</p> <p>As a result, TwinCAT was able to receive the right data from the machine vision sensor and use that data as a part of a robot's motion control. The Delta robot can be driven manually via the virtual operation panel unit. However, the main objective to develop an automatic air hockey robot was not accomplished because of the programming challenges. While the air hockey application was tested, it was discovered that the Delta robot usually received the right coordinates to hit the puck but sometimes the coordinates did not match with the puck's trajectory. That's why the Delta robot went in the wrong direction.</p>	
Keywords	Machine vision, motion control, robotics, PLC programming

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Robotiikka	2
2.1	Robotiikka teollisuudessa	2
2.2	Robottityypit	3
2.2.1	Suorakulmainen robotti	4
2.2.2	Sylinterirobotti	5
2.2.3	Napakoordinaatistorobotti	5
2.2.4	SCARA-robotti	5
2.2.5	Nivelvarsirobotti	5
2.2.6	Rinnakkaisrakenteinen robotti	6
2.2.7	Yhteistoimintarobotti	6
2.3	Unimate – ensimmäinen teollisuusrobotti	7
2.4	Kinematiikka	8
3	Delta-robotti	9
3.1	Rakenne	9
3.2	Delta-robotti markkinoilla	10
3.3	Delta-robotin konfigurointi Beckhoffin TwinCAT-ohjelmointiympäristössä	11
4	Konenäkö	12
4.1	Valaistus	12
4.2	Optiikka	15
4.3	Kuva-anturityypit	15
4.4	Kuvankäsittely	16
4.5	Viivakoodit ja tekstin lukeminen	17
4.6	Tiedonsiirto	18
4.7	Älykamerat	18

5	Työn toteutus	19
5.1	Ohjelmistot ja laitteet	20
5.1.1	Beckhoff ja TwinCAT	20
5.1.2	SICK ja SOPAS Engineering Tool	21
5.1.3	Konenäkökamera	21
5.1.4	Ilmakiekkopeli	22
5.2	Kameran datan määrittely ja lähettäminen	23
5.3	Kameran datan vastaanottaminen ja käsittely	25
5.3.1	Koordinaattidatan käsittely	27
5.3.2	Automaattiajo	28
5.3.3	Manuaaliajo	29
5.4	Ilmakiekko ja kiekon liikeradan ennakointi	30
5.5	Kiekon ennakointiin tarvittavat kaavat	31
6	Yhteenveto ja kehitysideat	34
	Lähteet	37

## Lyhenteet

CCD	<i>Charge-Coupled Device</i> . Valoherkkä kennotyyppi.
CMOS	<i>Complementary Metal Oxide Semiconductor</i> . Valoherkkä kennotyyppi.
FBD	<i>Function Block Diagram</i> . Toimilohkokaavio.
FPS	<i>Frames per second</i> . Kuvausnopeus sekunnissa.
GVL	<i>Global variable list</i> . Globaali muuttujalista.
I/O	<i>Input / Output</i> . Tulo ja lähtö.
IL	<i>Instruction List</i> . Käskylista.
LD	<i>Ladder</i> . Tikapuukaavio.
NC/CNC	<i>Numerical Control / Computerized Numerical Control</i> . Numeerinen ohjaus / tietokoneistettu numeerinen ohjaus.
OCR	<i>Optical Character Recognition</i> . Optinen merkin luku.
OCV	<i>Optical Character Verification</i> . Optinen merkin varmistus.
PLC	<i>Programmable Logic Controller</i> . Ohjelmoitava logiikka.
SFC	<i>Sequential Function Chart</i> . Sekvenssikaavio.
ST	<i>Structured Text</i> . Strukturoitu teksti.
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i> . Internetliikennöintiin käytettävä usean tiedonsiirtoprotokollan yhdistelmä.

## 1 Johdanto

Insinööriyön tarkoituksena on suunnitella ja toteuttaa alun perin piirtopöytärobotiksi ohjelmoitu Delta-robotti toimimaan täysin uudenaikaisessa sovelluksessa, jossa robotin liikkeenohjaus perustuu kameran saatavaan tietoon. Työn tilaajana on Metropolia Ammattikorkeakoulu. Kyseessä on toiminnallinen työ, joka toteutetaan projektityyppisenä.

Työn päätavoitteena on suunnitella ja toteuttaa automaattinen ilmakiekkopeli, jossa vastustajana toimii omassa robottisolussaan oleva Delta-robotti. Tavoitteen saavuttamiseksi on ratkaistava robotin ja kameran välinen kommunikointi sekä luotava robotin liikkeenohjaukseen tarvittavat matemaattiset kaavat lähestyvän kiekon torjumiseksi.

Lisäksi tavoitteena on ohjelmoida robotille virtuaalinen käyttöliittymä, joka mahdollistaa virtuaaliakseleiden suuntaisen manuaalijonon. Robottisolu on ollut vähäisellä käytöllä Metropolian koneautomaatiolaboratoriossa, ja tämän insinööriyön yhtenä tavoitteena on saada robottisolusta kiinnostavampi aihe tulevaisuuden laboratorioharjoituksia ajatellen. Mahdollisia aiheita tulevaisuuden harjoituksiin löytyy insinööriyön lopusta.

Insinööriyössä perehdytään syvemmin Sickin konenäkötekniikkaan ja kameran saatavan datan käsittelyyn TwinCAT-ohjelmointiympäristössä. Tavoitteena on myös selvittää eri laitevalmistajien välinen kommunikointi ja tarkastella valitun konenäkökameran soveltuvuutta liikkuvan kohteen paikantamiseen.

## 2 Robotiikka

Robotti on kappaleiden tai työkalujen käsittelyyn suunniteltu vähintään kolmiakselinen sekä uudelleenohjelmoitavissa oleva mekaaninen laite. Robotti voidaan ohjelmoida tekemään yksinkertaista työtä, joka toistaa samoja liikeratoja tai se voidaan suunnitella suoriutumaan teollisuuden työtehtävistä, joille ei aina pysty suunnittelemaan liikeratoja etukäteen. Tätä varten robottien ohjelmoinnissa käytetään antureilta saatua tietoa, jonka avulla robotti on ohjelmoitu suoriutumaan työtehtävistään vaihtelevilla liikeradoilla. (Kui-  
vanen 1999: 13.)

### 2.1 Robotiikka teollisuudessa

Teollisuusroboteilla tavoitellaan automatisoituja tuotantolinjoja, joissa robotit tekevät pitkästyttävän ja ihmiselle vaarallisen työn säästäten samalla yrityksen resursseja ja pitäen työntekijät terveempinä ja motivoituneempina (RobotWorks 2018a). Teollisuusrobottien eksponentiaalista kasvua on kutsuttu jopa teollisuuden neljänneksi vallankumoukseksi, mikä kuvaa hyvin robotiikan laajentumista aina vain suuremmaksi osaksi teollistunutta maailmaa. Ihmisten kanssa ilman fyysisiä raja-aitoja työskentelevien yhteistoimintarobottien liittäminen osaksi tuotantolinjaa auttaa pääsemään aina vain tuottoisampiin ratkaisuihin. Kansainvälisen teollisuusrobotiikan markkina-arvon on arvioitu olevan 32 miljardia dollaria (US) vuonna 2018. Pelkästään autoteollisuus kasvatti robottien lukumäärää 43 % vuosien 2013 ja 2014 välisenä aikana. (RobotWorks 2018b.)


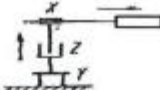


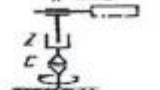




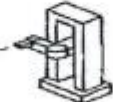
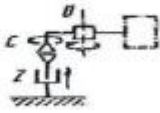







Robotiikkaan investoiminen on yrityksille kallista, mutta siitä saadut hyödyt maksavat itsensä nopeasti takaisin. Teollisuudessa työskenteleviin ihmisiin verrattuna robotit voivat työskennellä lähes tauotta kellon ympäri, puhumattakaan vuosilomista tai kahvitauoista. Oikein ohjelmoitujen robottien avulla saadaan poistettua tuotannosta inhimilliset virheet samalla, kun tuotannon laatu ja luotettavuus paranevat. Teollisuuden robotisointi vähentää yrityksen työvoiman tarvetta, jolloin yritys pystyy valmistamaan tuotteitaan kustannustehokkaammin. Teollisuuden työvoimantarve pienenee automatisoinnin seurauksena, ja työntekijöitä voidaan kouluttaa tuotantolinjoilta robottien valvonta- ja huoltotehtäviin, mikä vähentää työntekijöiden onnettomuusriskiä. (RobotWorks 2018b.)



Yritykset tarvitsevat osaavaa työvoimaa myös tulevaisuudessa, mikä ei tule katoamaan robottien valloittaessa teollisuuden työtehtäviä. Yritykset ovat jo pitkään investoineet tuotantolaitoksiin Kaukoidässä ja etenkin Kiinassa, missä on halvempaa tuottaa manuaalisesti valmistettavia tuotteita. Lincoln Electric Automation on Yhdysvalloissa tehdyn kehitystyön tuloksena onnistunut kehittämään hitsausrobotteja, joiden avulla tuotannon laatu ja kannattavuus paranevat. Investoidessaan Lincoln Electric Automationin kehittämään robottitekologiaan saavat amerikkalaiset yritykset luotua uusia työpaikkoja tuotannon pysyessä Yhdysvalloissa. Amerikkalaiset yritykset ovat jo investoineet edellä mainittuihin robotteihin ja saavuttaneet tuottavuuden kasvua, jonka ansiosta on pystytty perustamaan uusia tehtaita ja palkkaamaan satoja uusia työntekijöitä työskentelemään Yhdysvalloissa. (IndustryWeek 2015.)

## 2.2 Robottityypit

Perinteiset teollisuusrobotit jaotellaan usein niiden mekaanisten eroavaisuuksien perusteella, kuten vapausasteiden, käyttötarkoituksen, valmistajan tai sisäisen rakenteen perusteella. Uusimpana robottien jaotteluperusteena on vielä robotin yhteistoiminnallisuus. (Bouchard 2014.) Kuvassa 1 näkyvät yleisimmät teollisuusrobottimallit standardin ISO 8373 mukaisesti jaoteltuna sekä niiden rakenne, kinemaattinen kaavio ja työalue (Kuijanen 1999: 12–13).

Nimitys pääakseleiden mukaan	Rakenne	Kinemaattinen kaavio	Työalue
Suorakulmainen robotti			
Sylinterirobotti			
Napa-koordinaatisto-robotti			
Scara-robotti			
Kiertyvänivelinen robotti			
Rinnakkaisrakenteinen robotti			

Kuva 1. Robottityypit (Kuivanen 1999: 12).

### 2.2.1 Suorakulmainen robotti

Suorakulmaisen robotin (*cartesian robot*) vapausasteista kolme ensimmäistä ovat lineaarisia (Kuivanen 1999: 16). Suorakulmaista robottia käytetään usein nosta ja aseta -työssä sen erinomaisen tarkkuuden ja toistettavuuden ansiosta, mutta suoraviivaisten liikeratojensa takia se ei kykene kiertyviin liikkeisiin. Robotti ei sovellu kosteaan tai likaiseen ympäristöön sen vaikeasti suojattavien lineaaristen nivelten takia. (BARA 2017.)

### 2.2.2 Sylinterirobotti

Sylinterirobotilla (*cylindrical robot*) on kolme akselia, joista kaksi on lineaarisia ja yksi kiertyvä. Robotin nimi tulee sylinterikoordinaatistosta. Sylinterirobotit ovat suorakulmaisten robottien tapaan hyviä suoraviivaisia liikkeitä vaativiin toimintoihin. Jämäkän rakenteensa ja suoraviivaisen liikeratansa ansiosta sylinterirobotit pystyvät ulottumaan piekeen tilaan, kuten asettamaan kappaleen koneistuskeskukseen ja kappaleen pois ottamiseen. (BARA 2017.)

### 2.2.3 Napakoordinaatistorobotti

Napakoordinaatistorobotti (*spherical robot*) on samankaltainen kuin sylinterirobotti, mutta sillä on kaksi kiertyvää ja yksi lineaarinen akseli. Robotin työskentelyalue on pallomainen, mistä se on myös saanut nimensä. Modernien nivelvarsirobottien kehittäminen niin mekaniikka- kuin ohjelmistotasolla on ajanut napakoordinaatistorobottien yli, eikä napakoordinaatistorobotteja enää käytetä teollisuudessa. Ne ovat kuitenkin historiallisesti merkittäviä, koska teollisuusrobotiikka alkoi menestyä napakoordinaatistorobotti Unimaten myötä 1960-luvulla. (All On Robots 2013.)

### 2.2.4 SCARA-robotti

SCARA-robotissa (*Selective Compliance Assembly Robot Arm*) on kolme kiertyvää niveltä sekä lineaarinen pystyjohde (Kuivanen 1999: 16). SCARA-robotteja käytetään laajasti elektroniikkateollisuudessa niiden tarkan paikoituksen ja nopeuden ansiosta. Robotin käyttökohteita ovat esimerkiksi kokoonpano, kuormalavoille lastaus sekä konetäyttö. (BARA 2017.)

### 2.2.5 Nivelvarsirobotti

Nivelvarsiroboteilla (*articulated robot*) on enemmän vapausasteita kuin millään muulla markkinoilla olevalla robottityypillä, ja näin ollen ne ovat muunneltavissa moniin eri käyttötarkoituksiin. Nivelvarsiroboteilla on yleensä neljä tai kuusi vapausastetta, mutta niitä voi olla jopa kymmenen, minkä ansiosta niiden ulottuma ja ketteryys ovat erinomaiset.

Käyttöikä ja suosio ovat nostaneet nivelvarsirobottien määrää teollisuudessa, minkä ansiosta niiden hinta on myös laskenut. Tämä takaa sen, että yhä useampi yritys kykenee sijoittamaan teollisuusrobotiikkaan nostaakseen tuotantoaan. (RobotWorks 2018c.)

#### 2.2.6 Rinnakkaisrakenteinen robotti

Rinnakkaisrakenteisessa robotissa (*parallel robot*) on muista robottityypeistä poiketen suljettu kinemaattinen rakenne. Rinnakkaisrobotti koostuu nivelöidyistä tukivarsista, joilla liikutetaan työkalutasoa. Rinnan kytketyillä tukivarsilla saavutetaan jämäkkä rakenne sekä nopeat ja tarkat liikeradat. Näiden ominaisuuksien ansiosta rinnakkaisrakenteisia robotteja käytetään paljon lääke- ja ruokateollisuudessa kappaleensiirtotehtävissä. Rinnan kytketyillä tukivarsilla saadaan aikaan kupolimainen työalue, jonka rajallisuus on kuitenkin rinnakkaisrakenteisen robotin heikkous. (Gonzales 2016.)

#### 2.2.7 Yhteistoimintarobotti

Yhteistoimintarobotit ovat robotteja, jotka voivat tehdä yhteistyötä ihmisten kanssa ilman perinteisen teollisuusrobotin vaatimia turva-aitoja. Standardien ISO 10218-1 ja ISO 10218-2 mukaan roboteille on neljä eri yhteistoimintatyyppiä. (Bélanger-Barrette 2015.)

Ensimmäinen tapa on määrittää robotille omat turvarajat, joiden sisäpuolella havaittu liike pysäyttää robotin ennen mahdollista yhteentörmäystä. Robotin turvarajat mahdollistavat työskentelyn sen lähellä. Turvarajojen aktivoituessa on robotille annettava uusi lähtökäsky. (Bélanger-Barrette 2015.)

Toinen yhteistoimintatyyppi on käsiohjaus, jolloin robotin akselikohtaiset moottorit vapautetaan, minkä jälkeen robotin käsivartta liikutetaan haluttuun asentoon ja näin opetetaan nopeasti uusia reittejä robotille. Tämä vaatii robottiin asennetun voiman tuntevan anturin, joka sallii ihmisen liikuttaa robottia halutulla tavalla. Käsiohjaus ei ominaisuutena poista tarvetta robotin turvalliseen aitaamiseen. (Bélanger-Barrette 2015.)

Kolmas tyyppi on monitoroida robotin ympärillä olevaa liikettä lasereilla tai konenäöllä, jolloin robotin työskentelyn nopeus hidastuu kohteen lähestyessä sitä ja pysähtyy koko-

naan kohteen tullessa tarpeeksi lähelle. Robotti palaa automaattisesti normaaliin työskentelynopeuteensa työntekijän poistuessa sen valvomasta tilasta. (Bélanger-Barrette 2015.)

Tunnetuin yhteistoimintarobottiin tyyppi on voiman tunteva robotti. Tämä robotti pystyy työskentelemään ihmisten kanssa turvallisesti, ja se mittaa kosketuksesta koituvan voiman ja pysähtyy kohdatessaan esteen. Turvallisuuden lisäämiseksi näiden robottien muodot on pyöristetty, eikä niissä ole ulkonevia moottoreita. Standardissa ISO/TS 15066 on määritetty maksimivoima ja -energia, jotka ihminen voi kokea turvallisesti osuessaan robottiin. (Bélanger-Barrette 2015.)

### 2.3 Unimate – ensimmäinen teollisuusrobotti

Modernien teollisuusrobottien kehityksen voidaan sanoa alkaneen keksijä George Devolin esitettyä 1954 patentoimansa idean ohjelmoitavasta kappaleenkäsittelijästä Joseph Engelbergerille vuonna 1956. Engelberger oli jo pitkään ollut kiinnostunut Isaac Asimovin robotteja käsittelevästä tieteiskirjallisuudesta, eikä kestänyt kauaa, että Engelberger sai päällikkönsä, Condecin toimitusjohtajan, rahoittamaan Devolin keksintöä. Vajaa kaksi vuotta tästä ensimmäinen prototyyppi Unimate #001 valmistui. Prototyyppi sijoitettiin työskentelemään General Motorsin painevalulinjastolle Trentoniin, New Jerseyyn vuonna 1959. Vuonna 1966 robotti esiteltiin suurelle yleisölle television välityksellä, jolloin robotti hauskuutti yleisöä pelaamalla golfia ja kaatamalla olutta. Samana vuonna tuotantoa päätettiin laajentaa Yhdysvaltojen ulkopuolelle ja Euroopan robottivalmistukseen lisensioitiin suomalainen Nokia, ja vuonna 1969 lisenssisopimuksen allekirjoitti Kawasaki Heavy Industries Aasian markkinoille. (Robotic Industries Association 2018.)

## 2.4 Kinematiikka

Robotin kinematiikalla tarkoitetaan sen kaikkia mahdollisia liikkeitä huomioiden sen käsisivarsien pituudet sekä nivelten rajoitteet. Kinematiikka ei ota kantaa siihen, mikä voima aiheuttaa robotin liikkeitä, vaan ovatko liikkeet mahdollisia vai ei. (Owen-Hill 2015.)

Teollisuusrobotin toiminnan kannalta on tärkeää, että sen liike ja työkalun asema ovat hallittavissa. Robottia voidaan hallita joko suoralla tai käänteisellä kinematiikalla. Suoralla kinematiikalla tarkoitetaan tilannetta, jossa halutaan selvittää työkalun asema robotin nivelten paikka-arvojen perusteella. Suora kinematiikka on yksiselitteisempää, koska sen avulla saadaan vain yksi robotin asento, joka toteuttaa halutun työkalun aseman. Käänteisen kinematiikan tehtävä on laskea robotin nivelten paikka-arvot työkalun aseman perusteella. Moniakselisella robotilla on useampi variaatio nivelten paikka-arvoiksi tietyssä työkalun asemassa, mikä luo haasteita robotin ohjelmoimiseen. Kuusiakselisella robotilla on esimerkiksi 16 mahdollista vaihtoehtoa tietyn työkalun aseman saavuttamiseksi. (Kuivanen 1999: 20; Owen-Hill 2015.)

### 3 Delta-robotti

Delta-robotin kehittäminen sai alkunsa 1980-luvun alussa Raymond Clavelin loistokkaasta ideasta käyttää suunnikkaiden tarjoamaa muotoa rinnakkaisrobotin suunnittelussa. Aikalaisjulkaisujen vastaisesti idea tuli Clavelilta itseltään eikä Willard L. Pollardin vuonna 1942 patentoimasta rinnakkaisrobotista. Clavel kutsui robottiaan Delta-robotiksi ja loi keksinnöllään menestyneimmän mallin rinnakkaisrobottien keskuudessa. Delta-robotteja oli vuosituhanen vaihteeseen tultaessa satoja ympäri maailmaa. Raymond Clavel voitti jopa ABB Flexible Automationin sponsoroiman Golden Robot Awardin ansioituneesta työstään rinnakkaisrobottien kehittäjänä. (Parallemic 2001.)

Delta-robotti sisältää 36 patenttia, joista tärkein, WIPO-patentti, julkaistiin 18.6.1987 (WO 87/03528). Yhdysvalloissa patentti julkaistiin 11.12.1990 (US 4,976,582). Euroopassa patentti julkaistiin 17.7.1991 (EP 0 250 470). Patentit suojaavat Delta-robottia Yhdysvalloissa, Kanadassa, Japanissa sekä useimmissa Länsi-Euroopan maissa. (Parallemic 2001.)

#### 3.1 Rakenne

Tyypillisesti Delta-robotti saa liikevoimansa kolmelta servomoottorilta, jotka on kytketty robotin kiinteään runkorakenteeseen. Delta-robotissa on kaksiosaiset käsivarret, joiden ylempi varsi on liitetty servomoottorille joko suorakäytöllä tai vaihteiston välityksellä. Alempi varsi on kytketty työkalutasoon ja on molemmista päistä pallonivelöity vapaamman liikkuvuuden saavuttamiseksi. Delta-robotissa voi myös olla työkalua pyörittävä akseli, joka nostaa vapausasteiden määrän neljään. (Maw 2018.)

Delta-robotin liikuteltava rakenteellinen massa on saatu alhaiseksi, koska käsivarsia liikuttelevat moottorit eivät liiku käsivarsien mukana vaan luovat suljetun kinematiikan toisin kuin esimerkiksi nivelvarsirobotissa, jossa avoimen kinematiikan takia jokaisessa kääntyvässä nivelessä on oma moottorinsa. Kevyempään rakenteeseen päästään myös valmistamalla robotin käsivarret hiilikuidusta. (Maw 2018.) Delta-robotin nopeus ja ketteryys tekevät siitä loistavan valinnan nosta ja aseta -työhön (Parallemic 2001).

### 3.2 Delta-robotti markkinoilla

Vuonna 1983 Delta-robotin markkinoinnin aloittivat sveitsiläiset veljekset Marc-Olivier ja Pascal Demaurex perustaessaan Demaurex-nimisen yrityksensä Sveitsin Romanel-sur-Lausannessa. He ostivat lisenssin Delta-robottiin vuonna 1987 päätavoitteenaan rinnakkaisrobottien kaupallistaminen pakkausteollisuuden käyttöön. Tämä lisenssi kuitenkin kattoi vain Delta-robotit, joiden koko oli alle 800 mm. Muutamassa vuodessa Demaurexistä oli tullut merkittävä tekijä uusilla ja haasteellisilla markkinoilla. Yrityksen tehtyä lukuisia muutoksia robotteihinsa neljä niistä kaupallistettiin nimillä Pack-Placer, Line-Placer, Top-Placer sekä Presto. (Parallemic 2001.)

Demaurex osti kattavamman lisenssin vuonna 1996, mutta ennen kuin kauppa oli varmistunut, oli lisenssi myyty jo kahdelle yritykselle. Demaurex sai yksinoikeuden valmistaa alle 800 mm:n kokoisia Delta-robotteja, ja suurempien robottien valmistamista varten oleva lisenssi myytiin AID-nimiselle yritykselle. DeeMed-niminen yritys osti lisenssin AID:ltä ja seuraavaksi lisenssin sai ruotsalainen Elekta ostaessaan DeeMedin. Elekta valmistaa Delta-robotteja, jotka pystyvät kantamaan 20 kg:n painoisia SurgiScopemikroskooppeja. Medtronic osti Delta-robottitekniikan Elektalta loppuvuodesta 1999. (Parallemic 2001.)

Elekta neuvotteli ennen SurgiScopen myyntiä myös ABB-konsernin kanssa Delta-robotin lisensseistä, minkä seurauksena ABB hankki itselleen lisenssin suurempien Delta-robottien valmistukseen. Tässä vaiheessa myös Demaurex olisi halunnut laajentaa Delta-robotteja valmistuksensa kattamaan myös suurempien robottien valmistusta, mutta heidän ostamansa lisenssi alle 800 mm:n Delta-roboteille rajasi heidät ulos isompien Delta-robottien valmistuksesta. (Parallemic 2001.)

Vuonna 1999 ABB Flexible Automation julkaisi Delta-robottinsa IRB 340 FlexPicker. Tämä robotti oli suunnattu lääke-, ruoka- sekä elektroniikkateollisuuteen. FlexPickerissä on sisäänrakennettu imujärjestelmä, joka kykenee nopeisiin kappaleensiirtotehtäviin jopa kg:n painoisilla kappaleilla. Robotissa on Cognexin konenäkökamera ja ABB S4C -hallintalaite. IRB 340 FlexPicker saavuttaa jopa 10 m/s ja 3,6 deg/s nopeuden sekä 100 m/s<sup>2</sup> ja 1,2 rad/s<sup>2</sup> kiihtyvyyden. (Parallemic 2001.)



Demaurex hallitsi Delta-robottimarkkinoita yli 15 vuotta, mutta ABB:n tuodessa omat robottinsa taloudellisesti tuottoisille markkinoille oli Demaurexin keksittävä uusi myyntivaltti. Se siirtyi pelkkien Delta-robottien suunnittelusta kokonaisten robottisolujen valmistajaksi, sekä päästäkseen maailman markkinoille yhtiö myytiin sveitsiläiselle SIG Groupille vuoden 1999 lopussa. (Parallemic 2001.) Vuodesta 2004 lähtien Demaurex on ollut osa Bosch Groupin pakkausteknologiaosastoa, kun Bosch Group osti SIG Groupin pakkausosaston (Saltmarsh 2016).

### 3.3 Delta-robotin konfigurointi Beckhoffin TwinCAT-ohjelmointiympäristössä

Suljetun kinemaattisen rakenteensa takia Delta-robotin ohjelmointi tapahtuu rinnakkaiskinematiikan mukaisesti, jolloin kaikkia robotin moottoreita on ajettava samanaikaisesti työkalutason liikuttamiseksi. Sarjakinematiikkaa noudattavia robotteja on mahdollista ohjata moottori kerrallaan, ja näitä robotteja ovat avoimen kinematiikan robotit kuten SCARA ja nivelvarsirobotti. (Beckhoff 2009.)

TwinCATissa tehtävä robotin kinematiikan konfigurointi tapahtuu lisäämällä kirjasto Kinematic Transformation. Kirjasto sisältää robotin ohjaukseen tarvittavat matemaattiset kaavat sekä valmiit ohjelmointilohkot. Kirjaston lisäämisen jälkeen voidaan luoda NC-konfiguraatio kinematiikka- tai interpolaatioryhmälle TwinCAT System Manageriin. (Beckhoff 2019.) Kinematiikkaryhmään lisätään virtuaaliakselit sekä moottoreille akselit. Delta-robottia ajetaan käänteisellä kinematiikalla, jolloin työkalutason liikuttaessa kinematiikkakirjaston avulla lasketaan moottoreille tarvittavat kääntökulmat halutun työkalutason liikkeen toteuttamiseksi. Interpolaatiolla saadaan aikaan kaarevia liikeratoja kinematiikkakirjaston avulla, kun moottoreiden nopeutta säädetään suhteessa toisiinsa robotin tekemän liikkeen aikana.

## 4 Konenäkö

Automated Imaging Association (AIA) määrittelee konenäön koostuvan sovellutuksista, joissa laitteiston ja ohjelmiston yhdistelmällä saadaan otettua kuva ja prosessoitua siitä toimintakäskeä ohjattavalle laitteelle (Cognex 2016). Konenäön avulla saadaan nopeutettua tuotantoa ja parannettua tuotannon laatua joko korvaamalla tarkastustyö kokonaan konenäöllä tai täydentäen manuaalisesti tehtävää tarkastustyötä konenäön ottamien kuvien avulla. Konenäköä sovelletaan teollisuudessa kappaleen paikan määrittämiseen, mittojen ottamiseen, visuaaliseen tarkastamiseen sekä tuotteiden identifioimiseen koodien perusteella. (SICK 2006.)

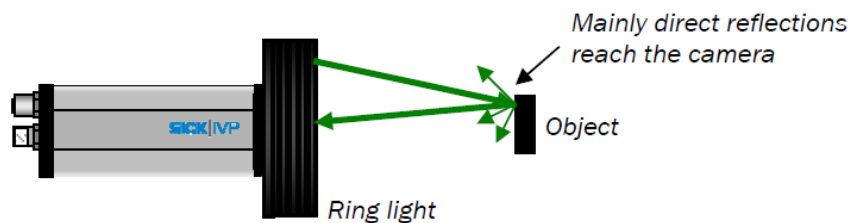
Konenäköjärjestelmä sisältää useita komponentteja, joiden tehtävänä on huolehtia oikeanlaisen kuvan ottamisesta ja sen käsittelystä. Komponentit sisältävät valaistuksen, objektiivit, kuva-anturin, kuvankäsittelyn ja tiedonsiirron. Valaistuksen avulla saadaan kappaleesta etsittävät muodot paremmin esille. Kuva-anturi muokkaa objektiivin läpi tulevan valon digitaalseksi kuvaksi, joka lähetetään analysoitavaksi. (Cognex 2016.) Kuvien analysointiin käytetään yleisimmin järjestelmään liitettävää tietokonetta, jolla mahdollistetaan erilaisten ohjelmointisovellusten hyödyntäminen niin kuvankäsittelyssä kuin muussakin järjestelmään liitettävässä automatiikassa (AIA 2016). Konenäköjärjestelmän toimintaa hallitaan käyttöliittymän avulla, mikä voi olla tietokone tai logiikkaan kytketty näyttöpäätte. Käyttöliittymän avulla järjestelmän toimintoja seurataan, säädetään ja raportoidaan. (Voutilainen 2004.)

### 4.1 Valaistus

Valaistus on ratkaiseva tekijä kuvan onnistumisen kannalta, koska konenäköjärjestelmä luo kuvan analysoimalla kappaleesta heijastuvaa valoa. Valaistuksen kannalta merkittäviä tekijöitä ovat valotusaika sekä aukkoarvo. Aukkoarvolla säädetään kuva-anturille kerralla tulevan valon määrää, ja valotusajalla, nimensä mukaisesti, määritellään valotuksen aika. Nopeasti liikkuvia kappaleita kuvatessa tulee valotusajan olla tarpeeksi lyhyt, jotta vältetään liike-epäterävyys. (SICK 2006.)

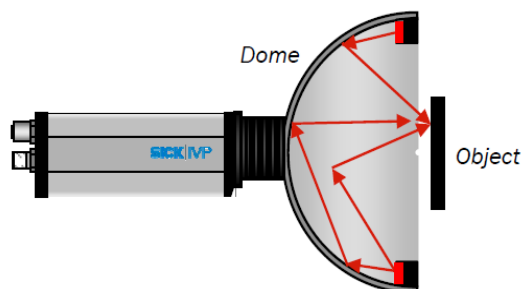
Konenäköjärjestelmien valaistuksessa on käytetty halogeenivaloja ja loisteputkia etenkin suurten kappaleiden valaisemiseen ja LED-valoa pienillä kappaleilla. LED-valon kehittyneen valaistusvoimakkuuden ansiosta siitä on tullut eniten käytetty valaisintyyppi myös suurille kappaleille. LED-valon hyötyjä ovat myös tasainen valo, pitkä elinkaari sekä alhainen energiankulutus. (Vision Doctor 2018a.)

Valaistustekniikan valinta riippuu siitä, millaisia havaintoja kameralta odotetaan. Suoraa valaistusta, kuten kuvassa 2 nähtävää rengassalamaa, käytetään sen kirkkaan valotehon ja alhaisen säteilykulman ansiosta. Heikkoutena suoralla valaistuksella on ylivalotumista aiheuttavat kirkaspisteet heijastuvalla pinnalla. (SICK 2006; Vision Doctor 2018b.) Rengassalaman kirkaspisteitä voidaan välttää asettamalla kamera viistosti tasoon nähden, mikäli se ei vääristä kuvattavaa kohdetta.



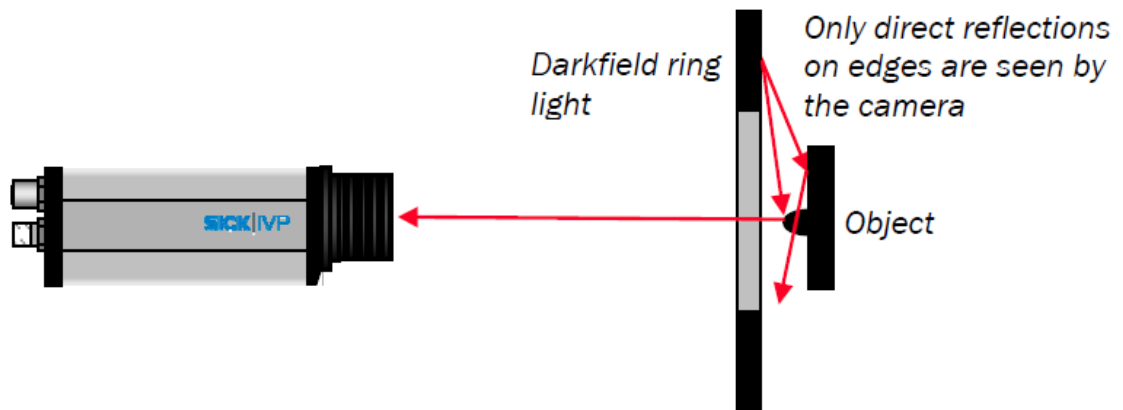
Kuva 2. Rengassalama ja valon kulku (SICK 2006).

Hajavalon avulla saadaan kuvattua kirkkaita pintoja ilman kirkaspisteitä ja varjoja. Eräs valaistustekniikka, joka käyttää hajavaloa, on kuvan 3 mukainen kaarivalo, jonka avulla saadaan tasalaatuinen pintavalaisuus valon heijastuessa kaaren sisäreunan kautta kappaleeseen. Kaarivalo tulee asettaa todella lähelle kappaletta, koska etäisyyden kasvaessa myös ei-toivottu suoran valaistuksen määrä kasvaa.



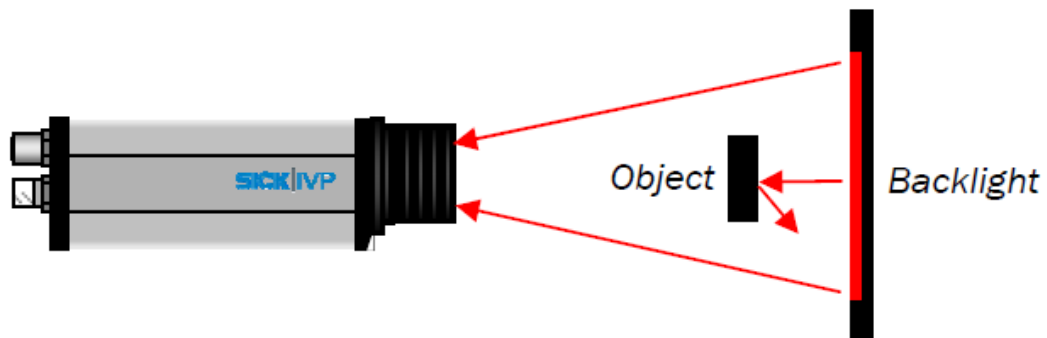
Kuva 3. Kaarivalo ja valon kulku (SICK 2006).

Kuvassa 4 näkyvän epäsuoran valaistuksen avulla saadaan kappaleen särmät hyvin esiin ja etenkin naarmujen ja epäpuhtauksien etsiminen kiiltäviltäkin pinnoilta onnistuu tällä valaistustekniikalla erinomaisesti. Epäsuora valo on hyvä pienille ja tasaisille pinnoille. Aivan kuten kaarivalossa, tulee epäsuoraa valoa käyttäessä pitää kohteen ja kameran välinen etäisyys pienenä. (SICK 2006.)



Kuva 4. Epäsuoran valaistuksen valon kulku (SICK 2006).

Kuvan 5 mukaisen taustavalon avulla saadaan näkymään läpinäkyvämmän kappaleen ääriviivat, jolloin niiden mittaaminen voidaan suorittaa tarkasti ilman kuvassa olevaa ylimääräistä informaatiota. Taustavalo läpäisee läpinäkyvät materiaalit, joten sitä voidaan käyttää myös lampun hehkulangan tarkastamiseen. (Vision Doctor 2018c.)



Kuva 5. Taustavalo ja valon kulku (SICK 2006).

## 4.2 Optiikka

Objektiivi on komponentti, jonka kautta kuvauskohteesta heijastuvat valonsäteet kulkeutuvat kuva-anturille. Objektiivi valitaan usein kuvauskohteen mukaan, ja tähän vaikuttavat niin kuvattavan kohteen ala kuin objektiivin linssin etäisyys kohteesta eli kuvausetäisyys. Objektiivin valinta tarkkojen yksityiskohtien saamiseksi tulisi tehdä siten, että kuvausala olisi mahdollisimman vähän kuvattavaa kappaletta suurempi. (Fales 2003.)

Optiikan laskemisessa voidaan käyttää apuna kaavan 1 linssiyhtälöä, jonka avulla saadaan laskettua esimerkiksi kuvauskulman määräävä polttoväli. Polttoväli ilmoitetaan yleensä millimetreinä ja sen tunnus on  $f$ . Linssiyhtälöä voidaan käyttää polttovälin laskemiseen, kun tiedetään kuvan etäisyys linssistä ( $a$ ) sekä kohteen etäisyys linssistä ( $b$ ).

$$\frac{1}{f} = \frac{1}{a} - \frac{1}{b} \quad (1)$$

(Jain, Kasturi & Schunck 1995: 249–250.)

## 4.3 Kuva-anturityypit

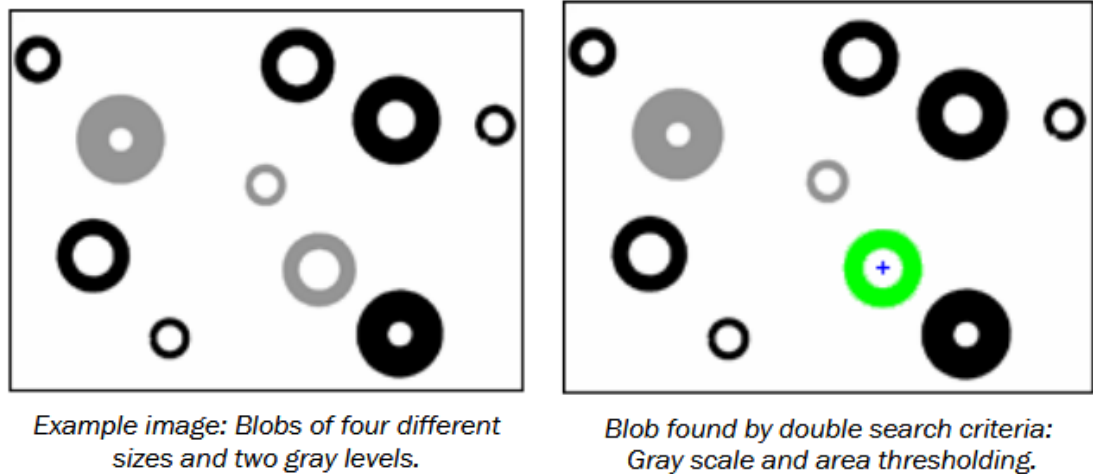
Konenäkökamerat käyttävät kuva-anturityyppiä CCD (*Charge-Coupled Device*) tai CMOS (*Complementary Metal Oxide Semiconductor*), joiden avulla valo muutetaan digitaaliseksi kuvaksi (Cognex 2016). Perinteisesti teollisuudessa on ollut käytössä CCD kuva-antureita niiden todella tarkan kuvanottokyvyn sekä valoherkkyyden ansiosta. CCD tuottaa analogisia signaaleja, jotka muutetaan digitaaliseksi kuvaksi, kun taas CMOS käyttää digitaalisia signaaleja kuvan muodostamiseen. Molempia tyyppejä käytetään teollisuudessa, mutta CMOS-kuva-antureita kehitetään paljon laajemmin ja niiden tarkkuus on saatu vastaamaan jo jopa CCD:n tasoa. CMOS-kuva-antureiden parempi kuvanopeus, alhaisempi energiankulutus ja hinta houkuttelevat investointeja suunnittelevia yrityksiä. (AIA 2017; UniverseOptics 2018.)

#### 4.4 Kuvankäsittely

Konenäkökameran liipaisu voidaan toteuttaa antureilla tai kamera voidaan asettaa tilaan, jossa se ottaa jatkuvasti uusia kuvia. Liipaisusignaalin toteuttaminen anturilla voidaan toteuttaa kytkemällä anturilta saatava tulosisignaali kameran liipaisusignaaliin. Kuvanoton jälkeen kuvan prosessoiminen voidaan aloittaa ja sitä voidaan tarvittaessa esikäsitellä tarkasteltavien muotojen korostamiseksi. (Cognex 2016.)

Kuvankäsittely alkaa siitä, kun kuva siirtyy kuva-anturilta digitaalisessa muodossa kuvankäsittelyohjelmistolle. Kuvankäsittelyllä kuvasta saadaan erotettua haluttuja piirteitä tai mittoja. Kuvankäsittelyn ansiosta kuvasta saadaan rajattua pois epäoleellinen informaatio, minkä ansiosta siirrettävä informaatio pienenee ja tiedonsiirto nopeutuu. (Cognex 2016; Voutilainen 2004.)

Kuvankäsittelyn avulla kappaleen paikka ja asento voidaan määrittää ja näitä tietoja voidaan käyttää sovelluksissa, joissa robotti käsittelee kuvattavaa kappaletta. Kuvankäsittelyn kannalta on tärkeää, millaista informaatiota halutaan. Tämän takia on tärkeää määritellä alue, jota tarkastellaan. Samassa kuvassa voi olla useampi tarkasteltava kohde, joita tulee tarkastella erikseen. Tarkasteltavalle kohteelle voidaan tehdä haluttuja toimenpiteitä, kuten kuvassa esiintyvien kontrastierojen perusteella tehtävä reunanetsintä, jota käytetään kappaleen mittaamiseen, tai kuvan vertaaminen verrokkikuvaan laaduntarkkailutehtävissä. BLOB-analyysillä haetaan kuvasta halutun geometrisen muodon ja värin omaavia kohteita, kuten kuvassa 6 on haluttu löytää juuri oikean kokoinen ja oikean värinen kappale. BLOB-analyysillä voidaan selvittää kohteen painopiste, pikselimäärä tai asema. (SICK 2006.)



Kuva 6. BLOB-analyysi (SICK 2006).

#### 4.5 Viivakoodit ja tekstin lukeminen

Kaupallisia tuotteita identifioidaan erilaisilla koodeilla nopeuttamaan niiden automaattista tunnistusta tuotantolaitoksissa sekä mahdollistetaan tuotteiden seuraaminen toimitusketjun läpi. Konenäkökameraa voidaan soveltaa näiden koodien lukemiseen. Tuotteiden identifioimiseen on käytetty pitkään yksiulotteisia viivakodeja sekä enenevässä määrin kaksiulotteisia Data Matrix -koodoja niiden suuremman tietokapasiteetin ansiosta. (Cognex 2016.)

Tekstin lukemiseen on käytössä OCV (*Optical Character Verification*) sekä OCR (*Optical Character Recognition*). OCV lukee tekstin ja vertaa sitä ennalta määrättyyn tekstiin tarkistaen, onko se oikein. Tällä voidaan tarkistaa esimerkiksi, onko tuotteessa oleva päivämäärä oikein, jolloin OCV hyväksyy tekstin. OCV hylkää tekstin, mikäli tuotteessa oleva päivämäärä ei vastaa ennalta määrättyä tai teksti on muuten viallinen. OCR on algoritmi, joka pystyy lukemaan ennestään tuntematonta tekstiä. OCR antaa tuloksena lukemansa tekstin, mutta hylkää tekstin, mikäli se ei sitä pysty sitä lukemaan. Toiminnan varmistamiseksi on kehitetty fontteja, joiden lukeminen OCR:n avulla on helpompaa. Näissä fontteissa kaikki kirjaimen ovat selkeästi eri näköisiä ja kirjaimilla on tasaiset välit. (SICK 2006.)

#### 4.6 Tiedonsiirto

Konenäköjärjestelmät ovat tärkeä osa nykyaikaista automatisoituvaa teollisuutta. Kerätäkseen haluttua tietoa tai toimiakseen muuten halutulla tavalla, tulee konenäköjärjestelmän kommunikoida muiden järjestelmässä olevien laitteiden kanssa. Tiedonsiirtoprotokolla valitaan käytettävien laitteiden perusteella. Ohjelmoitavan logiikan välityksellä tiedonsiirtotapa valitaan laitevalmistajan mukaan. Useimmiten käytetään yhtä seuraavista protokollista:

- EtherNet/IP
- MODbus
- DeviceNet
- Profibus, ProfiNET
- FINS
- IDA.

(SICK 2006.)

#### 4.7 Älykamerat

Teollisuudessa on perinteisesti käytetty konenäköjärjestelmiä, joissa kuvankäsittelyprosessi tapahtuu järjestelmään liitettyssä tietokoneessa. Ilman erillistä tietokonetta toimiva älykamera on kompakti kokonaisuus, jossa yhdistyy kamera, kuvankäsittely ja konenäkösovellukset. Älykamerat pystyvät prosessoimaan itsenäisesti otetut kuvat ja lähettämään tarvittavat tiedot ohjelmoitavalla logiikalle, sekä kuvanottodataa voidaan kerätä haluttuun osoitteeseen. Erillistä tietokonetta ei näin ollen tarvita ohjausyksiköksi, mikä vähentää tiedonsiirtoon kuluvaa aikaa ja järjestelmän tilantarvetta. (AIA nd; UKIVA 2016.)



## 5 Työn toteutus

Insinööriyön tarkoituksena oli muokata Metropolian koneautomaatiolaboratoriossa sijaitseva Delta-robotti pelaamaan ilmakiekkoa ihmistä vastaan. Työn mekaanisen kokoonpanon suunnittelu oli pääpiirteittäin melko yksinkertainen. Robotin alapuolelle sijoitettiin ilmakiekkopöytä ja robottisolun kattoon kiinnitettiin konenäkökamera. Konenäkökamera ottaa kuvia liikkuvasta kiekosta ja lähettää kiekon koordinaatit TwinCATiin, jossa niitä käytetään kiekon liikeradan ennakoimiseen.

Aluksi tutustuttiin käytettävään Sickin konenäkökameraan ja tarkasteltiin SOPAS Engineering Toolin avulla kameran ottamaa kuvaa sekä sitä, minkälaisia työkaluja ohjelmisto tarjoaa. Työkaluksi valikoitui melko nopeasti Object locator, jonka avulla kuvasta voidaan etsiä haluttu muoto. Kameran ottamaa kuvaa tarkasteltaessa saatiin varmistus uuden linssin välttämättömyydelle.

Työn alussa kameran liipaisu suunniteltiin tapahtuvan anturilla ja tähän tarkoitukseen testattiin Sickin WL170-P132 valokennoanturia. Anturin toiminta vaikutti lupaavalta, mutta työn edetessä todettiin, että on yksinkertaisempaa pitää kameraa jatkuvan kuvanoton tilassa ja hallita siltä saatavaa dataa TwinCATin ohjelmakoodin avulla. Kiekon ennakointiin tarvittavat kaavat olivat ohjelmakoodin kirjoittamisen perusta, ja näiden kaavojen keksimisen jälkeen ohjelman luominen oli mahdollista aloittaa. Robotin turvahäkin mittojen pohjalta hankittiin ilmakiekkopöytä ja pöydän mittoja käytettiin osana ohjelmakoodia.

Robotin alkuperäiseen turva-aitaukseen saatiin yksi seinä poistamalla tarpeeksi tilaa ilmakiekkopöydän asentamiselle. Tämän jälkeen robottisolun katossa olevaan alumiini-profiiliin kiinnitetty konenäkökamera asetoitiin kohtisuoraan ilmakiekkopöytää vasten ja kalibroitiin, jotta konenäkökameran tuottama data saatiin muutettua pikseleistä millimetreiksi. Tähän asti TwinCATin ohjelmakoodia oli pystytty testaamaan vain virtuaaliakseleilla, mutta loppukokoonpanon valmistuttua ohjelmakoodin soveltaminen työn kannalta oikeassa ympäristössä voitiin aloittaa.

## 5.1 Ohjelmistot ja laitteet

Insinööriyössä haasteita aiheutti eri laitevalmistajien välinen kommunikointi, kun käytössä oli Sickin konenäkökamera ja Beckhoffin TwinCAT-ohjelmointiympäristö. Sickin kameran ottamia kuvia analysoitiin SOPAS Engineering Tool -ohjelmistolla. Robotin ohjaamiseen tehtävä ohjelmakoodi kirjoitettiin strukturoituna tekstinä Beckhoffin TwinCAT-ohjelmointiympäristössä, johon kameralta saatavat ilmakiekkopelin kiekon koordinaattitiedot tuotiin SOPAS ET -ohjelmiston kautta.

### 5.1.1 Beckhoff ja TwinCAT

Beckhoff on teollisuusautomaatioon keskittynyt yritys, joka on erikoistunut PC-pohjaisen ohjaustekniikan kehittämiseen. Yrityksen pääkonttori sijaitsee Verlissä, Saksassa ja sillä on toimintaa 75 maassa. Beckhoff työllisti 3 900 henkilöä vuonna 2018 ja yrityksen liikevaihto oli 810 miljoonaa euroa vuonna 2017. (Beckhoff nd.) Beckhoffilla on Suomen ja Viron kattava toiminta-alue ja tämän yhtiön pääkonttori ja varasto sijaitsevat Hyvinkäällä. Pääkonttorin lisäksi Beckhoffilla on Suomessa useita toimipisteitä sekä toimipiste Tallinnassa. Kaikki toimipisteet kattavat myynnin, teknisen tuen, tuotekehityksen, sovellukset sekä huollon. (Beckhoff 2018.)

Beckhoffin PC-pohjainen ohjaustekniikka on perusta heidän avoimelle automaatorakenteelleen. Beckhoffin tuotteita ovat teollisuus-PC:t, I/O- ja kenttäväyläkomponentit, liikkeenohjaustuotteet sekä automaatio-ohjelmistot. Teollisuuskäyttöön tarkoitettuja komponentteja voidaan lisätä osaksi olemassa olevaa järjestelmää tai niistä voidaan kasata saumaton ohjausjärjestelmä. Beckhoffin laitteet ovat käytössä maailmanlaajuisesti ja niitä voidaan soveltaa monipuolisesti eri alojen automaatoratkaisujen toteuttamiseen. Beckhoffin sivuilta löytyy oma ongelmanratkaisukanava Beckhoff Information System, Infosys, joka tarjoaa tietoa eri komponenteista ja ohjelmointiin liittyvistä kysymyksistä, kuten muuttujista ja vikakoodeista. (Beckhoff nd.)

Työssä käytettiin Beckhoffin luomaa automaatio-ohjelmistoa TwinCATia, jolla voidaan korvata perinteiset PLC- ja NC/CNC-hallintalaitteet. TwinCAT tukee useimpia tietokoneeseen liitettäviä kenttäväyläkortteja, mikä mahdollistaa ohjelmiston konfiguroimisen useimpien laitevalmistajien laitteille. Ohjelmistolla voidaan rakentaa ohjelmakoodia sekä seurata reaaliajassa ajettavaa koodia. TwinCAT tukee standardin IEC 61131-3 mukaisia

ohjelmointikieliä, joita ovat käskylista (*Instruction List*, IL), tikapuukaavio (*Ladder Diagram*, LD), toimilohkokaavio (*Function Block Diagram*, FBD), sekvenssikaavio (*Sequential Function Chart*, SFC) ja strukturoitu teksti (*Structured Text*, ST). (Beckhoff 2014.)

### 5.1.2 SICK ja SOPAS Engineering Tool

Sick on maailmanlaajuisesti yksi merkittävimmistä anturien ja anturisovellusten valmistajista automaatioteollisuudessa, ja yrityksen anturit luovat perustan luotettavalle ja turvalliseen tuotantoon tehokkuutta unohtamatta. Sickin pääkonttori sijaitsee Waldkirchissä, Saksassa, ja sillä on toimipisteitä ympäri maailman sekä yli 50 tytäryhtiötä. Yrityksen palveluksessa työskentelee noin 9 000 henkilöä ja liikevaihto oli 1,5 miljardia euroa vuonna 2017. (Sick 2019a.) Suomessa Sick Oy työllistää 31 henkilöä, jotka toimivat asiakaspalvelutehtävissä tarjoten ammattitaitoista sovellusosaamista läpi tuotteen elinkaaren (Sick 2019b).

SOPAS Engineering Tool on Sickin kehittämä konfigurointityökalu, joka näyttää heti ohjelman avautuessa samassa verkossa olevat Sickin anturit. Näistä antureista käyttäjä voi konfiguroida haluamansa. SOPAS ET ottaa käyttöön anturille sopivat ajurit tai lataa ne tarvittaessa laitteelta tai internetistä. Konfigurointityökalun avulla voidaan tarkastella anturilta saatavaa dataa ja esimerkiksi konenäköantureiden kohdalla voidaan määrittää kuvasta etsittävät parametrit kuten kuvankäsittelykappaleessa 4.4 kerrotaan.

### 5.1.3 Konenäkökamera

Työssä käytettävä konenäkökamera on kuvan 7 mukainen Sick Inspector PIM 60 harmaasävykamera CMOS-matriisianturilla. Kamera on 2D-kamera, jonka tarkkuus on 640 x 480 pikseliä ja kuvataajuus 40 kuvaa sekunnissa. (SICK 2019c.) Konenäkökamera tarvitsi laajemman kuvausalan omaavan linssin, koska kameran mukana tuleva vakio-linssi 10 mm:n polttovälillä osoittautui aivan liian suureksi näyttäen vain 210 x 180 mm:n kokoisen alueen toimintaetäisyyden ollessa 600 mm. Suuremman alueen kuvaamiseksi tilattiin Sickiltä 2,5 mm:n polttovälillä oleva linssi, jonka kuva-alue 860 x 645 mm on työn kannalta riittävä.



Kuva 7. Sick Inspector PIM 60 (Sick 2019c).

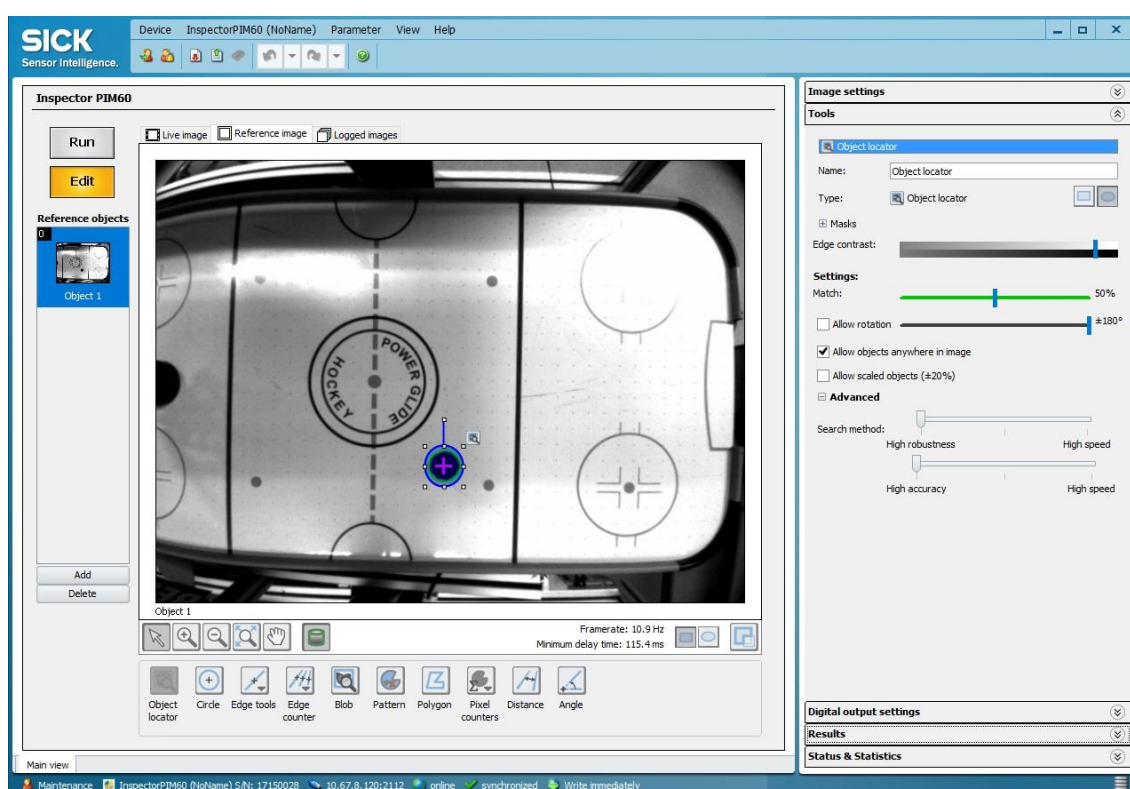
Työssä käytetty konenäkökamera soveltui liikkuvan kiekon paikantamiseen. Testausvaiheessa kiekon liikenoisuus pidettiin todella alhaisena ja tarkoitus oli vain testata, lähteekö robotti torjumaan lähestyvän kiekon. Kamera ei kuitenkaan pysynyt perässä, jos kiekolle antoi pelinomaisen nopeuden.

#### 5.1.4 Ilmakiekkopeli

Ilmakiekko on peli, jossa pelaajat yrittävät saada kiekon toisen pelaajan maaliin käyttäen ilmakiekkoon tarkoitettuja mailoja. Ilmakiekkopöydässä on säännöllisin välein reikiä, joista pöydän alapuolella oleva tuuletin puhalttaa ilmaa pelipöydän läpi. Tuulettimelta tuleva ilma saa kiekon leijumaan pelipöydän päällä, mikä antaa kiekolle kitkattoman ja nopeamman liikkeen. Ilmakiekkopöytiä löytyy eri kokoisia, ja tätä insinööritöitä varten kriteereinä oli sen mahtuminen robotin jo olemassa olevaan turvakehikkoon. Työtä varten hankittiin ilmakiekkopöytä, jonka ulkomitat ovat 122 x 61 cm.

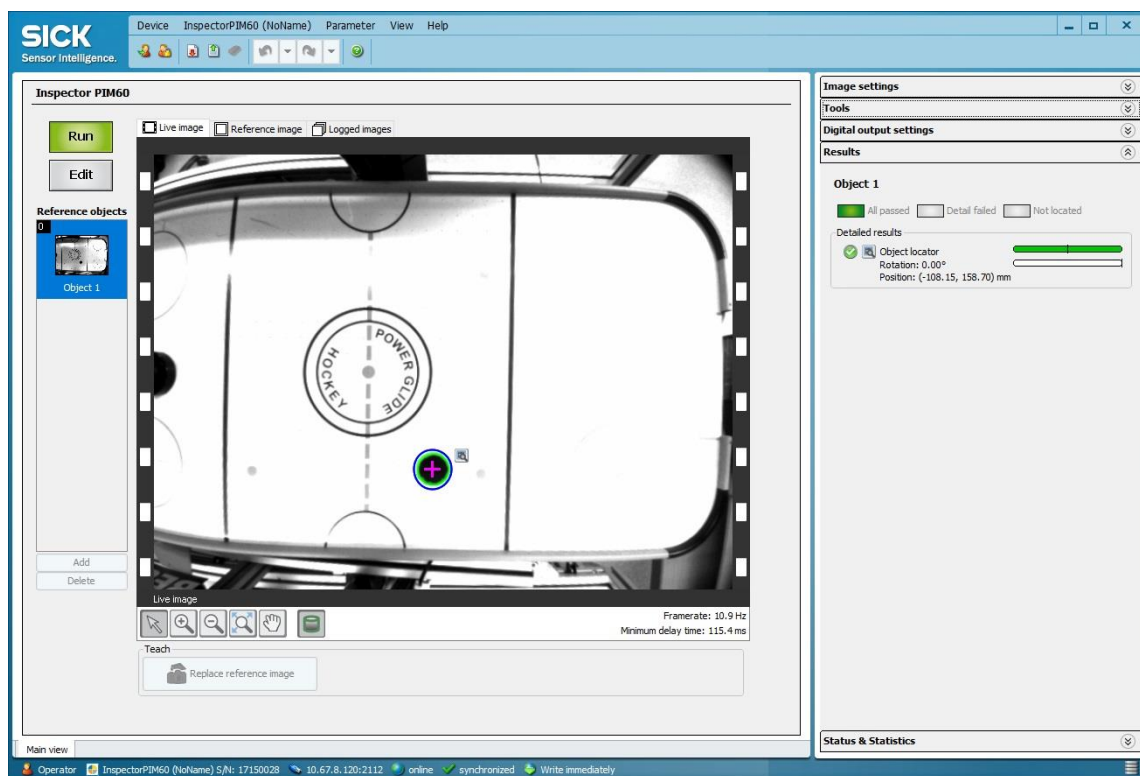
## 5.2 Kameran datan määrittely ja lähettäminen

SOPAS ET -ohjelmiston työkaluista valittiin Object locator -työkalu kiekon havaitsemiseen. Työkalulla rajataan hieman kiekkoa suurempi pyöreä alue ja määritellään Tools-valikosta kiekon havaitsemiseen tarvittavat parametrit kuvan 8 mukaisesti. Object locatorin ilmoittamat koordinaattitiedot haluttiin millimetreinä, minkä takia kamera oli kalibroitava. Kalibrointi on toteutettu kalibrointiin tarkoitetulla A3-kokoisella mustavalkoruudukolla siten, että kameran koordinaatiston origo on ilmakiekkopöydän keskipiste. Työkalu sekä kameran kalibrointi tallennetaan kameran Flash-muistiin, kun kamera asetetaan Run-tilaan.



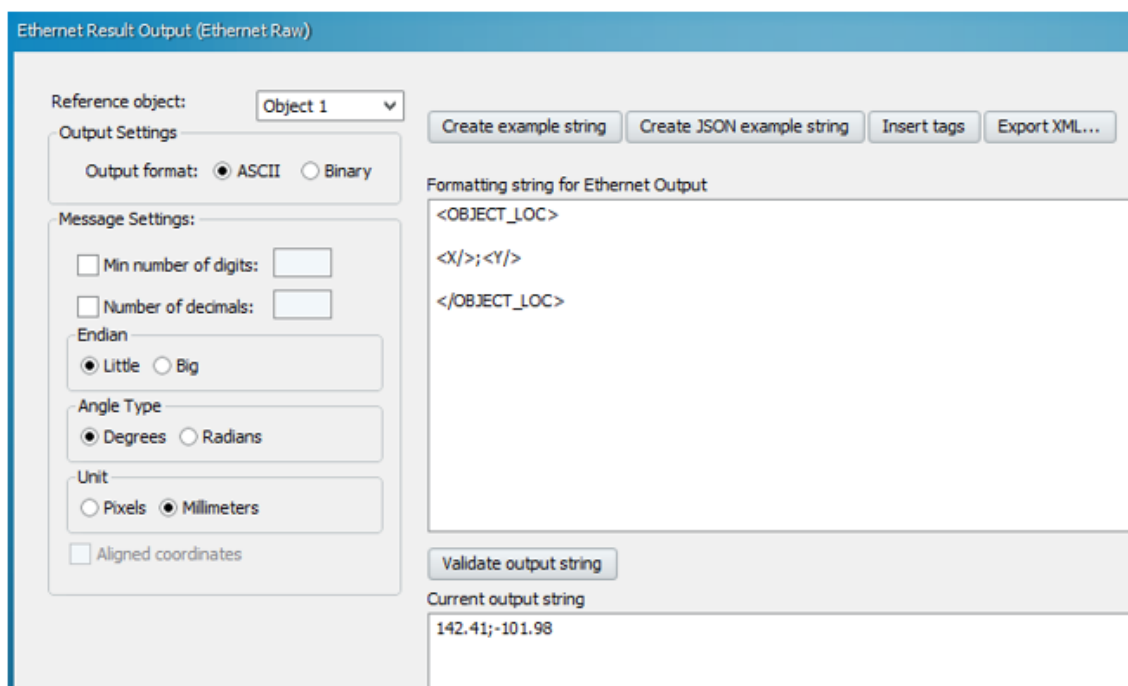
Kuva 8. Referenssikuvan määrittäminen.

Kuvassa 9 on havainnointi Object locator -työkalun toimintaa sen havaitessa sille määritellyn kappaleen kameran ollessa jatkuvasti kuvanotossa. Kuvan oikeassa reunassa Results-välilehdellä näkyy kiekon koordinaatit millimetreinä.



Kuva 9. Ilmakiikkopöytä, jossa Object locator -työkalu on havainnut kiekon.

Kuva 10 on ruutukaappaus tilanteesta, jossa kiekko on asetettu ilmakiikkopöydän päälle kuvan 9 mukaisesti ja kohdassa Current Output String näkyy kiekon koordinaatit millimetreissä. Näitä koordinaattitietoja käytetään ohjelmakoodin luomiseen TwinCATissa, kun ennustetaan kiekon liikeradan perusteella robotille haluttu piste koordinaatistossa. Kohdassa Formatting string for Ethernet Output, voidaan kameralta tulevasta tiedosta ottaa pois kaikki tarpeeton. Tässä tapauksessa halutaan vain Object locator -työkalun X- ja Y-koordinaatit, joten kaikki muu tieto on poistettu.



Kuva 10. SOPAS Engineering Tool, Ethernetin kautta tuleva data räätälöitynä.

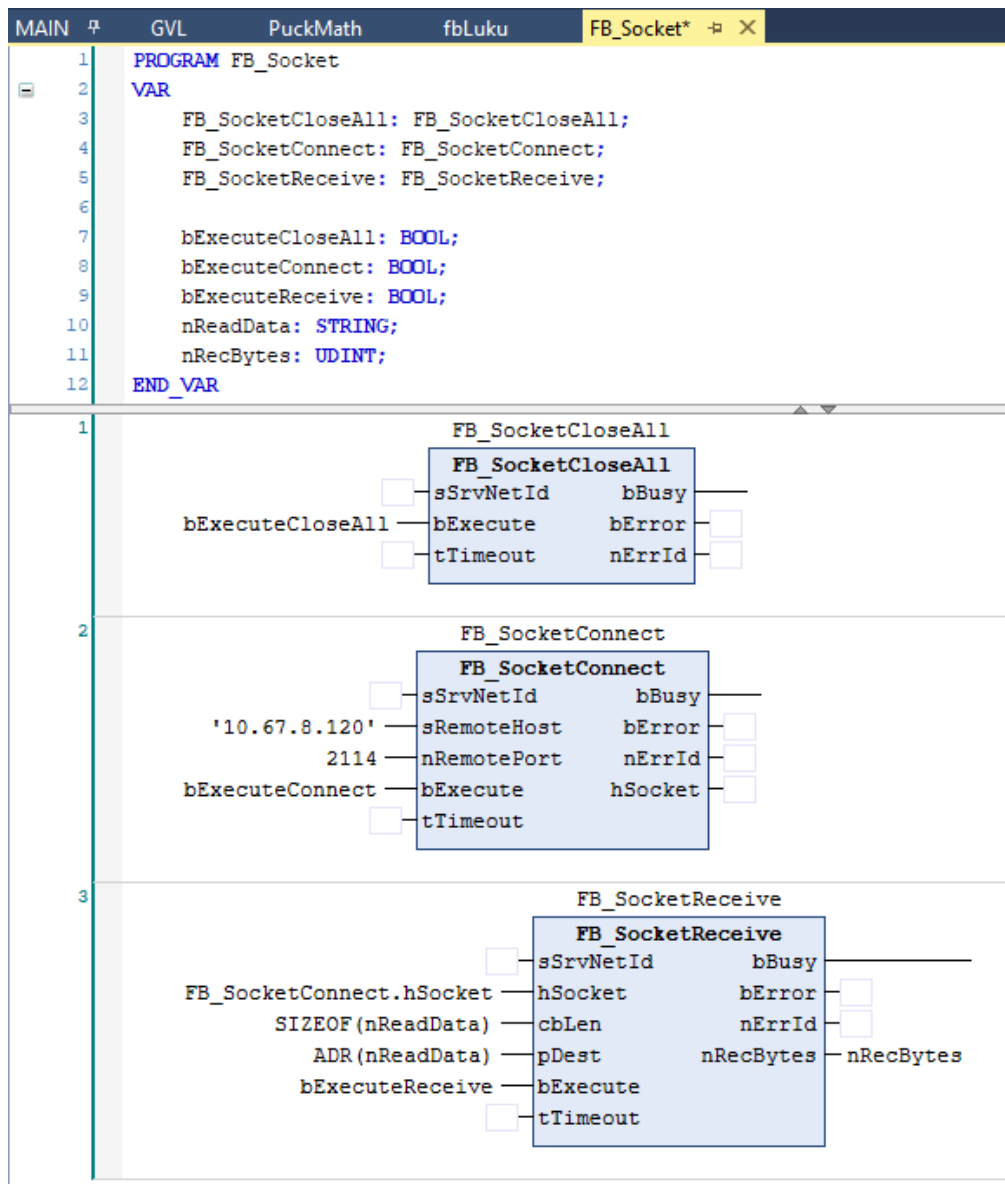
SOPAS ET:lla hallinnoidaan kameran lähettämää dataa muuttamalla tiedonsiirtoasetuksia anturikohtaisesta valikosta. Valikko on työssä käytetyn anturin mukaisesti nimeltään InspectorPIM60(NoName). Valitsemalla valikosta Interface and I/O Settings päästään valitsemaan oikea tiedonsiirtokanava, joka tässä työssä on Ethernet Raw. Samasta valikosta päästään Ethernet Raw -välilehdelle, josta valitaan tietoliikenneprotokollaksi TCP. Tältä välilehdeltä tarkistettiin myös kameran lähtöportti TwinCATissa tehtävää yhteydenottoa varten.

### 5.3 Kameran datan vastaanottaminen ja käsittely

Kameralta Ethernetin kautta lähtevän datan määrittämisen jälkeen oli vuorossa lähetetyn datan vastaanottaminen TwinCAT-ohjelmointiympäristössä. TCP/IP:n välityksellä tehtävä tiedonsiirto edellyttää kirjaston Tcplp aktivoimista TwinCATissa, mikä mahdollistaa tarvittavien ohjelmalohkojen käyttämisen. Logiikalle oli myös erikseen asennettava Beckhoffin sivuilta saatava TF6310-TCP-IP-tiedosto, joka mahdollistaa TCP/IP:n välityksellä tapahtuvan tiedonsiirron.

TwinCATiin tehtiin FB\_Socket-niminen ohjelma, johon tehtiin kuvan 11 mukainen toimilohkokaavio, jolla otettiin yhteys kameraan, vastaanotettiin dataa sekä suljettiin yhteys. Toimilohkolla FB\_SocketConnect otettiin yhteys kameraan kameran IP-osoitteen välityksellä ja lähtöportin avulla. Datan vastaanotto toteutettiin toimilohkolla FB\_SocketReceive, jonka hSocket-tulo linkitettiin FB\_SocketConnectin hSocket-lähtöön. Muuttujatyyppiltään T\_HSOCKET oleva hSocket toimii edellä mainittujen toimilohkojen yhteydenpitolinkkinä. FB\_SocketReceive-lohkoon tehtiin myös nReadData-muuttuja datan lukemista varten. Tämä muuttuja on tyyppiä string, joka vastaa kameralta tulevan datan muuttujatyyppiä. FB\_SocketCloseAll-toimilohkoa käytettiin yhteyden sulkemiseen. Ohjelmakoodin monipuolisuuden takia lopullinen koodi kirjoitettiin kuvasta 11 poiketen strukturoidulla tekstillä.





Kuva 11. Toimilohkokaavio yhteyden muodostamiseen.

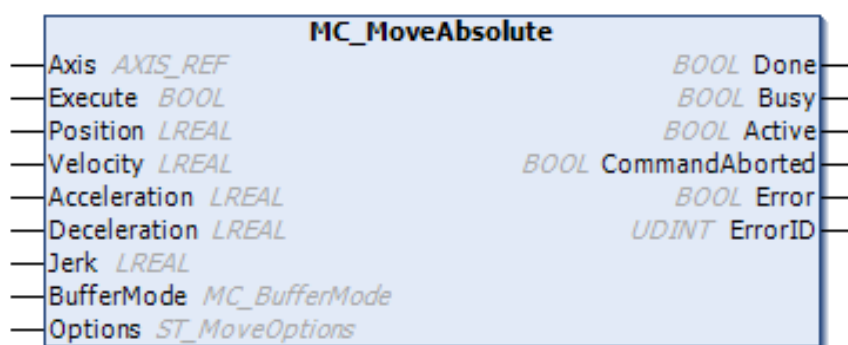
### 5.3.1 Koordinaattidatan käsittely

Kiekon koordinaattitiedot saatiin FB\_Socket ohjelmassa näkyviin nReadData-muuttujaan kuvan 10 mukaisena merkkijonona. Muuttujan saamasta merkkijonosta oli tämän jälkeen eroteltava X- ja Y-koordinaatit, joista tehtiin omat reaalitylukumuuttujat globaaliin muuttujalistaan (*Global Variable List*, GVL).

TwinCATiin luotiin laskentaohjelma puck\_math, joka hakee GVL:sta koordinaattien arvot ja laskee kiekon liikeradan avulla robotille pisteen, jossa se törmää kiekkoon. Laskentaohjelmassa lasketaan myös, osuuko kiekko pelipöydän laitoihin ennen sen päätymistä robotille ja mikä on robotille annettava koordinaatti mahdollisen laitaa osumisen jälkeen. Lasketusta Y-koordinaatista tehdään muuttuja GVL:iin, josta se haetaan robotin ohjaamiseen. Laskentakaavat on esitetty tarkemmin kappaleessa 5.5.

### 5.3.2 Automaattiajo

Ilmakiekon pelaamiseen robotille luotiin ohjelma nimeltä AutoMove, johon lisättiin kuvan 12 mukaiset MoveAbsolute-toimilohkot, jotka linkitettiin robotin virtuaaliakseleihin. Näillä toimilohkoilla robottia voidaan ajaa pisteestä pisteeseen robotin omassa koordinaatistossa. Robotin Z-akselin arvo pidetään vakiona, jotta robotin työkalutasoon asetettu ilmakiekkomaila ei irtoa pelipöydästä. Automaattiajossa robottia ajetaan vain XY-koordinaatistossa.



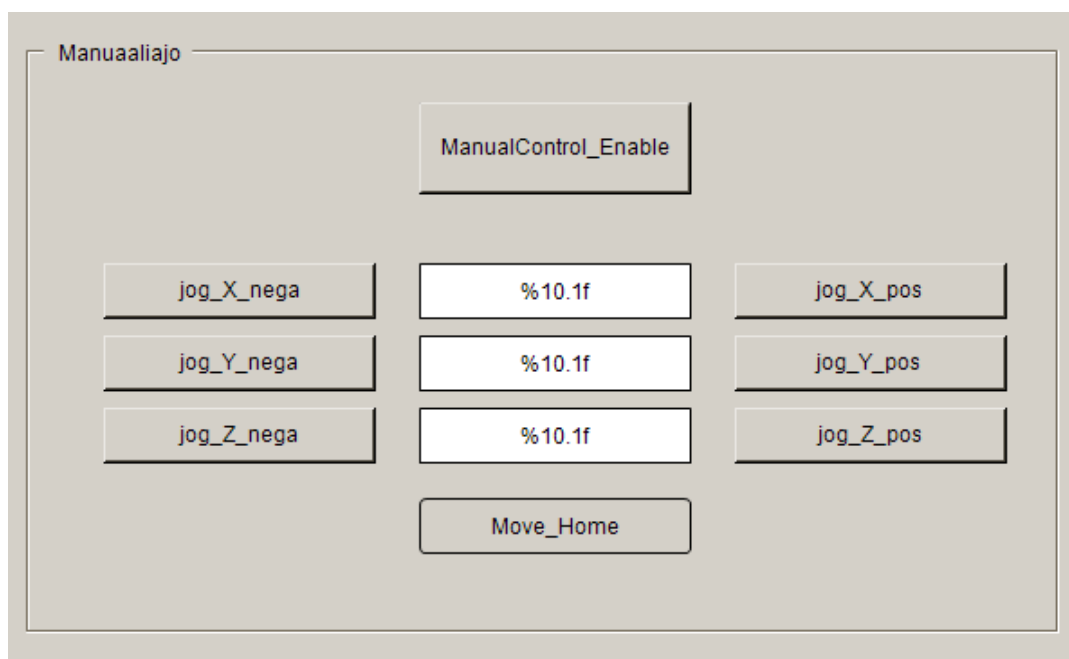
Kuva 12. MoveAbsolute-ohjelmointilohko.

Automaattiajon sisältämän AutoMoven ohjelmakierron alussa robotti ajaa kotipositioon (-100, 0, -40). Robotin moottorien liikkeen pysähtyttyä se on valmis vastaanottamaan uudet koordinaatit ja aktivoi globaalissa muuttujalistassa olevan muuttujan. Tämä muuttuja käynnistää ohjelman puck\_math ohjelmakierron. Robotti pysyy kotipositiossa, kunnes puck\_math on laskenut sille uuden koordinaatistopisteen lähestyvän kiekon torjumiseksi, minkä jälkeen robotti liikkuu puck\_mathin antamaan pisteeseen ja palaa takai-

sin kotipositioon. FB\_Socket-ohjelma päivittää GVL:iin kiekon koordinaattitietoja reaaliajassa, ja puck\_math ottaa aina päivitettyt koordinaatit robotin palatessa kotipositioon ja aktivoi puck\_mathin ohjelmakierron aloittavan muuttujan.

### 5.3.3 Manuaaliajo

Manuaaliajo toteutettiin tekemällä ohjelma ManualControl, joka sisältää MC\_Jog-toimilohkot, jotka linkitettiin robotin virtuaaliakseleihin. Manuaaliajoa varten TwinCATiin tehtiin kuvan 13 mukainen visuaalinen käyttöliittymä, jossa on painonapit jokaiselle akselille positiiviseen ja negatiiviseen suuntaan. Käyttöliittymässä on myös painonappi ManualControl\_Enable, jolla aktivoidaan robotin manuaaliajo. Kuvassa keskellä oleviin tekstimuuttujiin on linkitetty virtuaaliakseleiden paikka-arvot, jotka muuttuvat reaaliajassa, kun robottia ajetaan manuaali- tai automaattijolla.



Kuva 13. Manuaaliajon visuaalinen käyttöliittymä.

Painonapilla Move\_Home aktivoidaan samanniminen ohjelma, joka sisältää MoveAbsolute-toimilohkot kaikille virtuaaliakseleille. Painonapin aktivoituessa robotti siirtyy aiemmin esitettyyn kotipositioon. Manuaaliajon painonappien tulokonfiguraatiot ovat muotoa TAP, jolloin painonappi on aktiivinen vain sen aikaa, kun sitä painetaan, minkä jälkeen robotin liike pysähtyy. Kinematiikkakirjaston sisältämien laskentakaavojen ansiosta

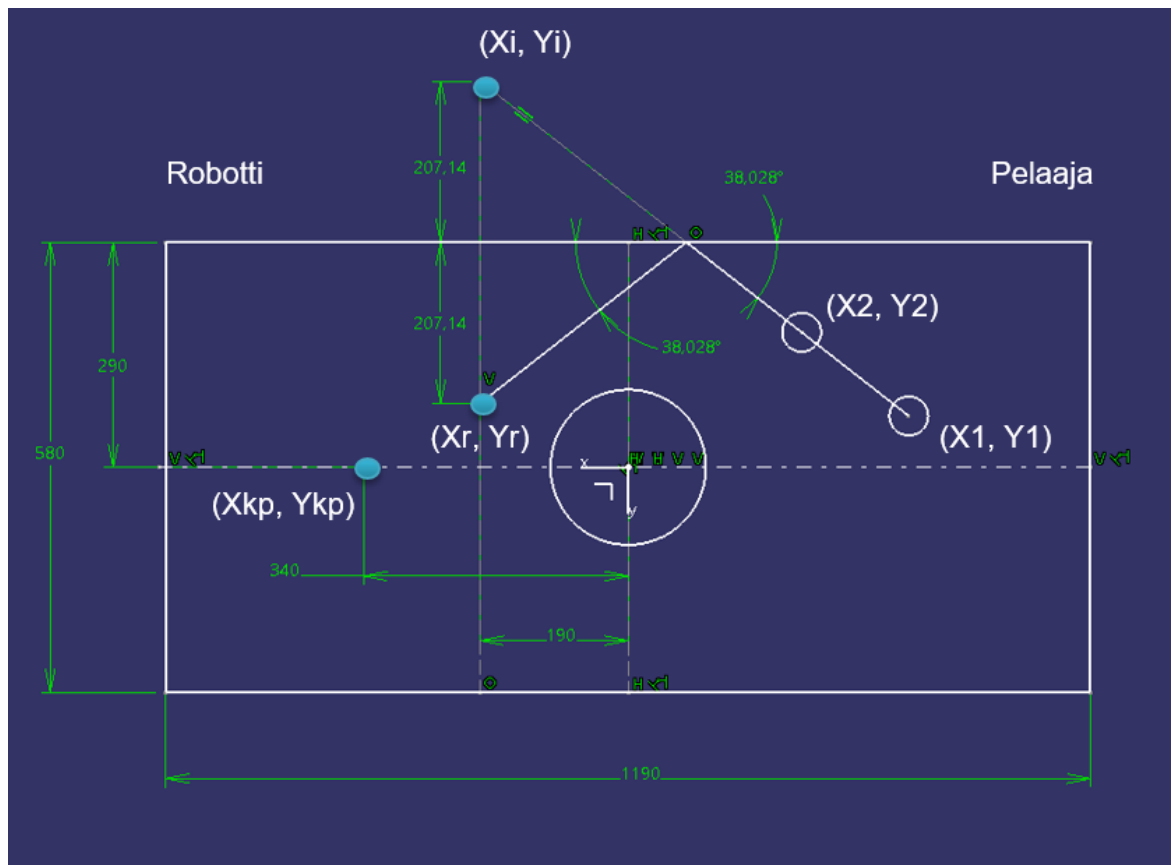
TwinCAT osaa ottaa huomioon robotin kiihtyvyyden sekä hidastuvuuden eikä robotti voi ylittää sille asetettuja päätyrajoja, vaikka manuaaliajon painonappia pitäisi aktiivisena.

#### 5.4 Ilmakiekko ja kiekon liikeradan ennakointi

Ilmakiekkosovelluksessa robotin lähtiessä kotiposiitiosta torjumaan saapuvaa kiekkoa saapuu se aina saman X-koordinaatin omaavaan pisteeseen, Robot\_X\_positioniin, jolla myös kuvassa 14 pisteet  $X_i$  ja  $X_r$  sijaitsevat. Y-koordinaatti saadaan laskemalla kiekon liikeradan suoran ja edellä mainitun Robot\_X\_positionin leikkauspiste. Kameralta saatujen koordinaattitietojen avulla lasketaan kiekon liikeradan kulmakerroin, jota käytetään koordinaattiarvojen kanssa kiekon liikeradan suoran yhtälön muodostamiseen. Muodostamalla yhtälöpari edellä mainitusta suorasta ja Robot\_X\_positionista saadaan selville haluttu Y-koordinaatti.

Kuva 14 havainnollistaa kiekon liikerataa pisteestä  $(X_1, Y_1)$  pisteeseen  $(X_r, Y_r)$  ja sen ennakointiin tarvittavia muuttujia. Muuttujat  $X_1$ ,  $Y_1$ ,  $X_2$  ja  $Y_2$  kuvaavat kameran ottamia kiekon koordinaattiarvoja näissä pisteissä. Kuvassa hahmotetaan kappaleessa 5.5 käytettyjen kaavojen paikkansapitävyys toteamalla, että kiekon osuessa laitaa sen tulo- kulma ja lähtökulma suhteessa pelipöydän laitaa vastaavat toisiaan. Näin ollen voidaan myös todeta, että laskukaavoissa käytetty mielikuvituspiste  $(X_i, Y_i)$  on yhtä kaukana pelipöydän laidasta, kuin kiekon todellinen kimpoamisen jälkeen oleva piste  $(X_r, Y_r)$ . Kuvassa oleva piste  $(X_{kp}, Y_{kp})$  kuvaa robotin kotiposition pistettä kameran koordinaatistossa, mikä vastaa robotin XY-koordinaatistossa pistettä  $(-100, 0)$ .

Kuvassa olevat koordinaattiakselit ovat kameran koordinaatiston suuntaiset. Robotin koordinaatiston X-akseli on vastakkaiseen suuntaan sekä robotin koordinaatiston origo on X-akselin suuntaisesti -240 mm suhteessa kameran koordinaatistoon. Nämä on korjattu ohjelmakoodissa vähentämällä ensin kameralta saadusta X-koordinaatin arvosta 240 sekä muuttamalla vähennyksen jälkeen X-koordinaatin etumerkki.



Kuva 14. Kiekon liikkeen havainnointi.

### 5.5 Kiekon ennakointiin tarvittavat kaavat

Tässä työssä käsitellään kiekon liikettä tilanteessa, jossa sen liikkeen kiihtyvyyttä tai hidastuvuutta ei huomioida. Tämän takia oletetaan, että kiekon nopeus pysyy alitusnopeuden laskemisen jälkeen vakiona. Kiekon alitusnopeuden laskemiseen tarvitaan lii-  
kematka kameran ottamien kuvien välissä, mikä saadaan laskettua kaavan 2 avulla.

$$puck\_camTravel = \sqrt{(rObjLocX2 - rObjLocX1)^2 + (rObjLocY2 - rObjLocY1)^2} \quad (2)$$

Kaavassa olevat rObjLoc-muuttujat ovat Object locator -työkalulta saatuja kiekon koor-  
dinaattitietoja ensimmäisestä ja toisesta havainnosta. Muuttujat on havainnollistettu ku-  
vassa 14. Kiekon nopeus saadaan kaavalla 3.

$$puck\_startVelocity = \frac{puck\_camTravel}{1000} / camera\_delay. \quad (3)$$

Puck\_camTravel antaa kiekon liikematkan millimetreissä, joten se on jaettava tuhannella, jotta nopeus saadaan muotoon m/s. Camera\_delay kuvaa kahden peräkkäisen kuvan aikaeroa. Kameran ilmoitettu kuvataajuus on 40 kuvaa/s ja tällä arvolla Camera\_delay olisi 0,025 s. Ottaen huomioon kiekon havaitsemiseen käytetyn Object locator -työkalun toiminta-ajan, realistisempi arvo muuttujalle Camera\_delay on SOPAS ET:sta tarkistettuna 0,1 s.

Kiekon Y-koordinaatin laskeminen aloitetaan laskemalla kiekon liikeradan kulmakerroin. Tälle muuttujalle annetaan nimi  $k\_puck$  ja se saadaan laskettua kaavalla 4 käyttäen apuna kameralta saatuja kiekon koordinaattitietoja.

$$k\_puck = \frac{rObjLocY2 - rObjLocY1}{rObjLocX2 - rObjLocX1} \quad (4)$$

Kiekon Y-koordinaatti saadaan selville sijoittamalla kameralta saadut kiekon toisen pisteen koordinaattiarvot suoran yhtälöön yhdessä kiekon liikeradan kulmakertoimen  $k\_puck$  sekä Robot\_X\_positionin kanssa. Kiekon Y-koordinaatti saadaan selville kaavalla 5.

$$puck\_Y\_position = (k\_puck * (Robot\_X\_position - rObjLocX2) + rObjLocY2) \quad (5)$$

Ratkaisemalla kiekon Y-koordinaatti saadaan myös selville, kimpoaako kiekko laidan kautta vai kulkeeko se kimpoamatta suoralle Robot\_X\_position. Tämä tarkistetaan kaavoilla 6 ja 7, joissa Y\_Limit on ilmakiekkopöydän Y-suuntainen leveys origosta pelipöydän reunaan. Kaavoilla selvitetään, pysyykö robottia kohti lyöty kiekko negatiivisen ja positiivisen Y\_Limit-arvon välissä robotille asti. Kiekon Y-koordinaatille lasketun arvon toteuttaessa nämä kaavat, on se valmis annettavaksi robotille. Muussa tapauksessa kiekolle on laskettava uusi arvo käyttäen kaavoja 8 ja 9.

$$puck\_Y\_position > Y\_Limit * (-1) \quad (6)$$

$$puck\_Y\_position < Y\_Limit \quad (7)$$

Kiekon Y-koordinaatin ollessa suurempi kuin  $Y\_Limit$ , kiekko kimpoaa pelipöydän positiiviselta reunalta ja uusi Y-koordinaatti saadaan kaavalla 8.

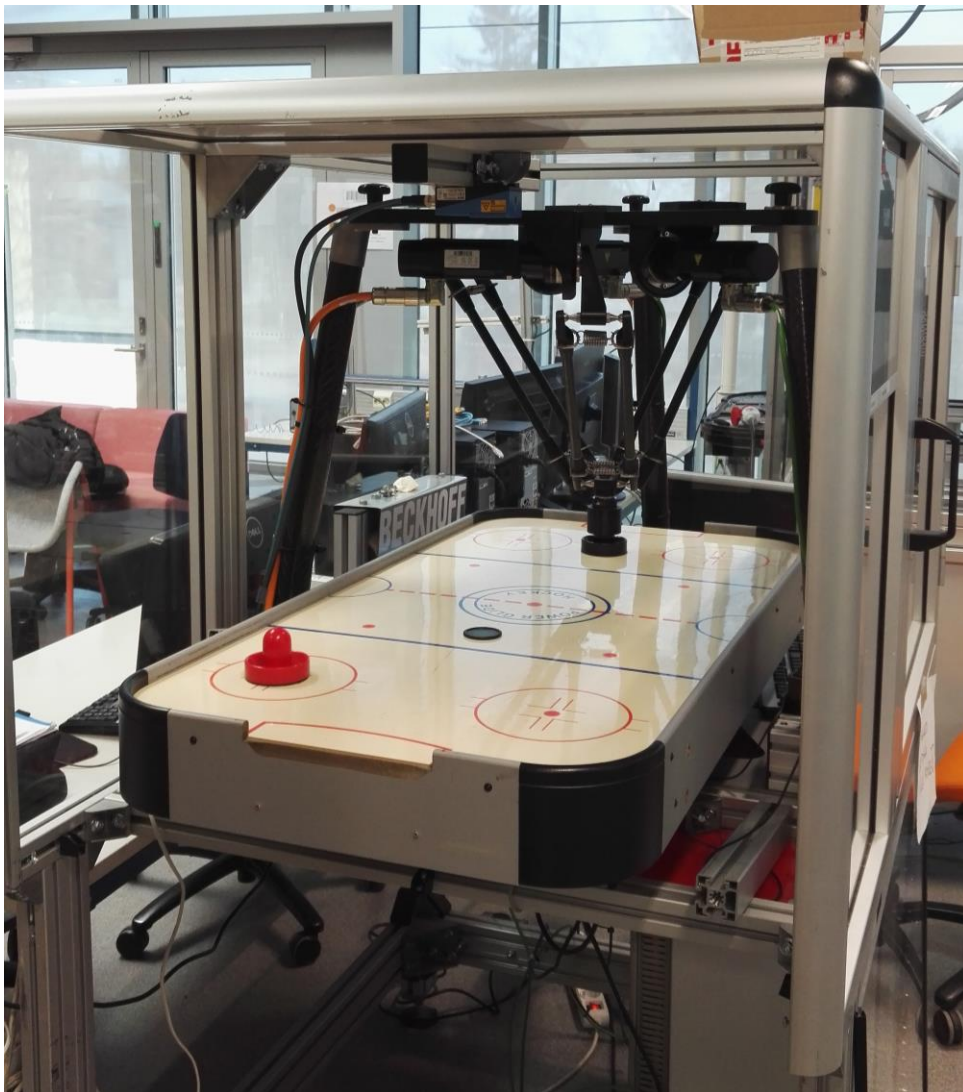
$$puck\_Y\_position = Y\_Limit - (puck\_Y\_position - Y\_Limit) \quad (8)$$

Kiekon kimmotessa pelipöydän negatiiviselta puolelta, eli Y-koordinaatin ollessa pienempi kuin negatiivinen  $Y\_Limit$ , saadaan uusi Y-koordinaatti kaavalla 9.

$$puck\_Y\_position = (Y\_Limit * (-1)) - (puck\_Y\_position + Y\_Limit) \quad (9)$$

## 6 Yhteenveto ja kehitysideat

Insinööriyön tavoitteena oli suunnitella ja toteuttaa sovellus, jossa Delta-robotti toimii vastustajana ilmakiekkopelissä. Kuvassa 15 on esitetty työn lopputulos, jossa näkyy robottisolussa oleva Delta-robotti ja työssä esitellyt komponentit. Tavoite oli yksinkertainen, mutta konenäkökameran käyttäminen koordinaattipisteiden hakemiseen liikkuvasta kohteesta ja näiden pisteiden käyttäminen osana robotin liikkeenohjausta antoivat omat haasteensa.



Kuva 15. Robottisolun loppukokoonpano.



Insinööriyössä haasteelliseksi osoittautunut kommunikoinnin muodostaminen eri valmistajien laitteiden välillä saavutettiin. Kameran lähettämää koordinaattidataa onnistuttiin vastaanottamaan TwinCATissa, sekä koordinaattidatan käyttö osana robotin ohjelmakoodia onnistui.

Robotin manuaaliajon toteuttaminen onnistui suunnitelmien mukaisesti. Robottia pystytään ajamaan visuaaliseen käyttöliittymään sijoitetuilla painonapeilla sekä seuraamaan virtuaaliakselien koordinaatistoarvojen muutosta tekstimuuttujien avulla. Ilmakiekon pelaamista varten robotille luodun automaattiohjelman testausvaiheessa robotti lähti liikkeelle, kun kiekkoa liikutti ilmakiekkopöydällä robottia kohti. Yleensä robotti lähti torjumaan kiekkoa aivan oikeaan suuntaan, mutta toisinaan robotille lasketut koordinaatit eivät vastanneet kiekon todellista liikerataa. Täsmälleen oikeiden koordinaattien laskeminen robotille on monen muuttujan summa eikä kiekko välttämättä liiku aina täysin oletettua liikerataa pitkin. Robotille annettavat koordinaatit suhteessa kiekon liikerataan ovat kuitenkin niin lähellä toisiaan, että eroavaisuudet johtuvat todennäköisesti koordinaattitojen välisistä eroista sekä kameran epätarkkuudesta.

Työn edetessä havaittiin useita kehitysideoita, mutta työn valmiiksi saattamiseksi oli joistakin ideoista luovuttava. Näitä ideoita voidaan soveltaa tulevissa Metropolian koneautomaation laboratorioharjoituksissa. Merkittävä turvallisuuteen liittyvä parannus olisi valoverhoanturin asentaminen robotin turvahäkin sisälle ilmakiekkopöydän keskiviivan tuntumaan, jotta robotti saadaan pysähtymään välittömästi ylimääräisen esteen saapuessa sen vaara-alueelle. Valoverhoanturi myös mahdollistaisi turvallisen kiekon hakemisen robotin vaara-alueelta, mikäli robotti ei sitä torjunut.

Pelipöydän kalibrointi ei tässä työssä vastaa täysin haluttua ja sen huomaa esimerkiksi kuvasta 8, jossa näkyy pelipöydän laitojen kaarevuus. Tämä olisi mahdollista korjata käyttämällä ulkoista koordinaatistoa tai suurempaa kalibrointipaperia. Pöydän tarkempi kalibrointi parantaisi kameran antamia kiekon koordinaatteja vastaamaan todellisuutta. Samalla kameran koordinaatiston origon voisi määrittää samaksi kuin robotilla, jolloin välttyttäisiin ohjelmakoodissa tehdyiltä koordinaatistomuutoksilta.

Laitteiston välisen kommunikoinnin toteuttaminen koulun verkon kautta onnistui, mutta yhteydenpito oli ajoittain epävakaa. Kenttäväylällä muodostettu yhteys olisi vakaampaa ja tätä kannattaisikin lähteä suunnittelemaan.

Tässä työssä Delta-robotin oli tarkoitus torjua lähestyvä kiekko, mutta huomioon ei otettu muita pelitilanteita. Jatkoprojekteissa voisi perehtyä syvällisemmin konenäöllä toteutettavaan kohteen tunnistukseen ja uusien ohjelmakoodien tekemiseen erilaisissa ilmakiekopeliin liittyvissä pelitilanteissa. Robotin nopeus oli tarkoitus sovittaa kiekon liikenopeuteen ja tätä varten laskettiin kiekon nopeuden määrittämiseen tarvittavat kaavat. Nopeuden säätö ei ollut prioriteettilistalla kovinkaan korkealla, mutta jatkossa tätä ideaa voisi jalostaa.

Robotin manuaaliajosta saisi mielenkiintoisemman liittämällä järjestelmään ulkoisen käsiohjaimen, jonka lähdöt linkitettäisiin virtuaalisen käyttöliittymän tapaan virtuaaliakseleille. Näin ollen robottia voisi ohjata erillisellä ohjaimella virtuaalisten painonappien sijaan.

## Lähteet

AIA. 2016. Components of Machine Vision Systems. <https://www.visiononline.org/blog-article.cfm/Components-of-Machine-Vision-System/28>. Luettu: 8.1.2019.

AIA. 2017. CCD vs CMOS Image Sensors: Which are Better? <https://www.visiononline.org/blog-article.cfm/CCD-vs-CMOS-Image-Sensors-Which-are-Better/82>. Luettu: 27.12.2018.

AIA. nd. Smart Cameras in Life Science Applications. <https://www.visiononline.org/smart-cameras-in-life-science-applications>. Luettu: 9.1.2019.

All On Robots. 2013. Spherical robots. <http://www.allonrobots.com/spherical-robots.html>. Luettu: 11.12.2018.

BARA. 2017. Robot Structures. <https://www.ppma.co.uk/bara/automation-explained/robots/robot-structures.html>. Luettu: 11.12.2018.

Beckhoff. 2009. Merger of Motion Control, PLC and robotics on one CPU. [https://download.beckhoff.com/download/document/Application\\_Notes/DK9322-1109-0009.pdf](https://download.beckhoff.com/download/document/Application_Notes/DK9322-1109-0009.pdf). Luettu: 24.4.2019.

Beckhoff. 2014. PLC and Motion Control on the PC. <https://www.beckhoff.com/twincat/>. Luettu: 8.4.2019.

Beckhoff. 2018. New Automation Technology. <https://www.beckhoff.fi/>. Luettu: 5.4.2019.

Beckhoff. 2019. TS511x | TwinCAT Kinematic Transformation. [https://www.beckhoff.com/english.asp?twincat/twincat\\_kinematic\\_transformation.htm](https://www.beckhoff.com/english.asp?twincat/twincat_kinematic_transformation.htm). Luettu: 24.4.2019.

Beckhoff. nd. Beckhoff short profile (PDF). [http://download.beckhoff.com/download/press/Short\\_profile\\_Beckhoff.pdf](http://download.beckhoff.com/download/press/Short_profile_Beckhoff.pdf). Luettu: 5.4.2019.

Bélanger-Barrette, Mathieu. 2015. What Does Collaborative Robot Mean? <https://blog.robotiq.com/what-does-collaborative-robot-mean>. Luettu: 11.12.2018.

Bouchard, Samuel. 2014. Industrial robots: What are the different types? <https://blog.robotiq.com/bid/63528/what-are-the-different-types-of-industrial-robots>. Luettu: 10.12.2018.

Cognex. 2016. Introduction to machine vision. [https://www.assemblymag.com/ext/resources/White\\_Papers/Sep16/Introduction-to-Machine-Vision.pdf](https://www.assemblymag.com/ext/resources/White_Papers/Sep16/Introduction-to-Machine-Vision.pdf). Luettu: 18.12.2018.

Fales, Gregg. 2003. Ten lens specifications you must know for machine-vision optics. <https://www.edn.com/electronics-news/4382728/Ten-lens-specifications-you-must-know-for-machine-vision-optics>. Luettu: 10.1.2019.

Gonzales, Carlos M. 2016. What's the Difference Between Industrial Robots? <https://www.machinedesign.com/datasheet/what-s-difference-between-industrial-robots-pdf-download>. Luettu: 3.1.2019.

IndustryWeek. 2015. How Robots Can Save and Create Manufacturing Jobs. <https://www.industryweek.com/robotics/how-robots-can-save-and-create-manufacturing-jobs>. Luettu: 7.12.2018.

Jain, Ramesh, Kasturi, Rangachar & Schunck, Brian G. 1995. Machine Vision. Singapore: McGraw-Hill Book Co.

Kuivanen, Risto. 1999. Robotiikka. Suomen robotiikkayhdistys ry. Vantaa: TummaVuoren kirjapaino Oy.

Maw, Isaac. 2018. The What, Why and How of Delta Robots. <https://www.engineering.com/AdvancedManufacturing/ArticleID/16651/The-What-Why-and-How-of-Delta-Robots.aspx>. Luettu: 10.12.2018.

Owen-Hill, Alex. 2015. Kinematics: Why Robots Move Like They Do. <https://blog.robotiq.com/kinematics-why-robots-move-like-they-do>. Luettu: 14.12.2018.

Parallelic. 2001. Delta Parallel Robot – the Story of Success. <https://www.parallelic.org/Reviews/Review002.html>. Luettu: 7.12.2018.

Robotic Industries Association. 2018. UNIMATE. <https://www.robotics.org/joseph-engelberger/unimate.cfm>. Luettu: 7.12.2018.

RobotWorks. 2018a. Industrial Robot Applications. <https://www.robots.com/applications>. Luettu: 7.12.2018.

RobotWorks. 2018b. Advantages and Disadvantages of Automating with Industrial Robots. <https://www.robots.com/blogs/advantages-and-disadvantages-of-automating-with-industrial-robots>. Luettu: 7.12.2018.

RobotWorks. 2018c. Articulated Robots - More Axes Equal More Possibilities. <https://www.robots.com/articles/articulated-robots-more-axes-equal-more-possibilities>. Luettu: 11.12.2018.

Saltmarsh, Abigail. 2016. Flexible in automation. <http://uniquelyaway.co.uk/wp-content/uploads/2016/11/demaurex.pdf>. Luettu: 15.1.2019.

SICK. 2006. Machine Vision Introduction. <https://www.sick.com/medias/Machine-Vision-Introduction2-2-web.pdf?context=bWFzdGVyfHJvb3R8NzIxNDY0MXxhcHBsaWNhdGlvbi9wZGZ8aDBmL2g2MC84ODMyMDk0MjA4MDMwLnBkZnw1ZTZjNDViMTQ2MDA4NmJjNTA2NTM1ZjE4MGY0YjEyY2M3MjM0OW-FiMWI3ZjYyMWNmMGFmZml5ZDA0YzRkYWE1>. Luettu: 19.12.2018.

SICK. 2019a. SICK AG:n yritysesittely. <https://www.sick.com/fi/fi/sick-agn-yritysesittely/w/about/>. Luettu 8.4.2019.

SICK. 2019b. SICK Suomessa. <https://www.sick.com/fi/fi/tietoa-meistae/w/sick/>. Luettu: 8.4.2019.

SICK. 2019c. 2D-konenäkö Inspector / Inspector PIM-series. <https://www.sick.com/fi/fi/konenaekoe/2d-konenaekoe/inspector/vspm-6f2113/p/p317945>. Luettu: 17.4.2019.

UKIVA. 2016. Smart cameras and vision sensors. <http://www.ukiva.org/embedded-vision/smart-cameras-and-vision-sensors.html>. Luettu: 9.1.2019.

UniverseOptics. 2018. Machine Vision CCD and CMOS Technologies. <http://www.universeoptics.com/machine-vision-ccd-and-cmos-technologies/>. Luettu: 27.12.2018.

Vision Doctor. 2018a. Selecting the perfect lighting. <https://www.vision-doctor.com/en/selecting-the-perfect-lighting.html>. Luettu 21.12.2018.

Vision Doctor. 2018b. Lighting for industrial machine vision. <https://www.vision-doctor.com/en/lighting-for-machine-vision.html>. Luettu 21.12.2018.

Vision Doctor. 2018c. Inspecting with backlight illumination. <https://www.vision-doctor.com/en/illumination-techniques/backlight-illumination.html>. Luettu: 21.12.2018.

Voutilainen, Päivi. 2004. Konenäkö. <http://www03.edu.fi/oppimateriaalit/puutuoteteollisuus/automaatio/konenako/index.html>. Luettu: 9.1.2019