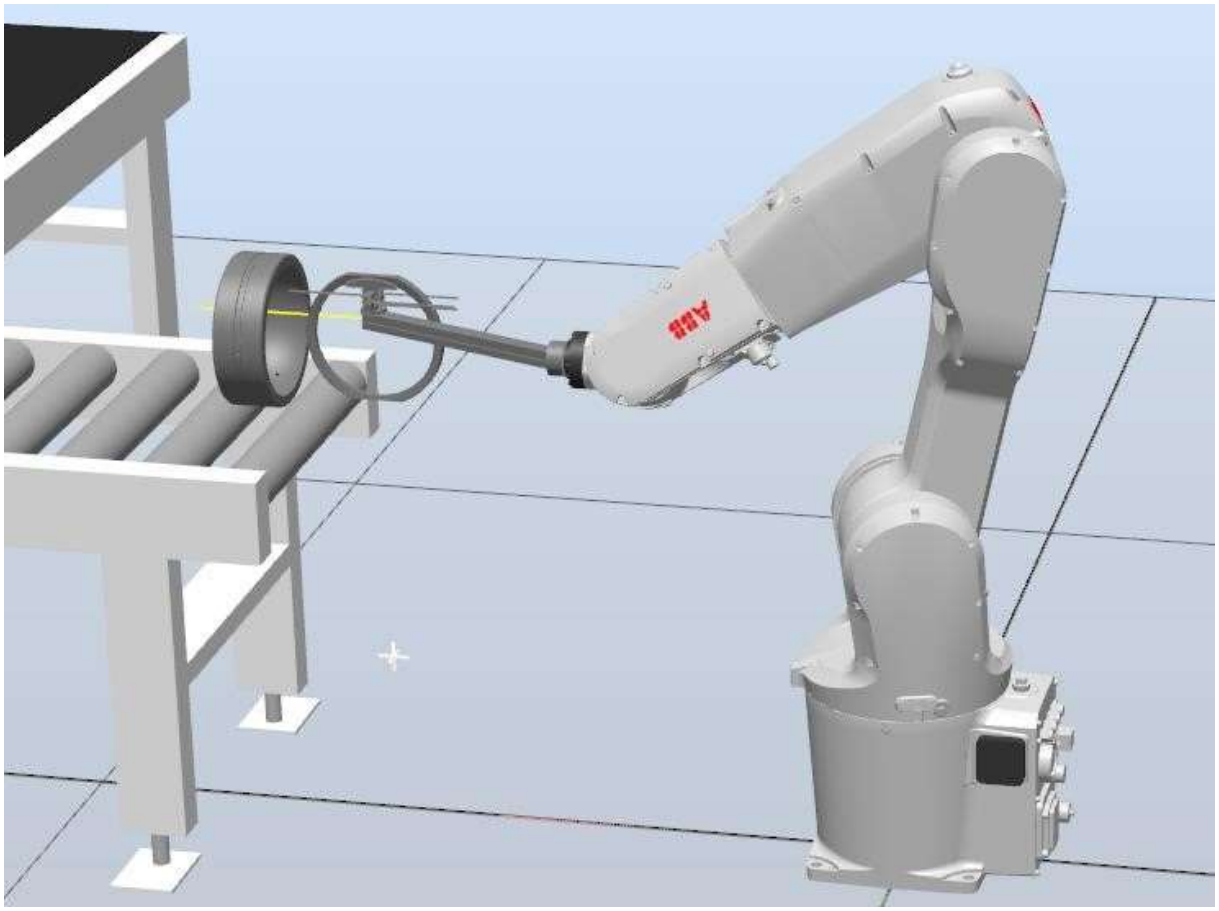
CHALMERS



Design och programmering av robotlösning för montering av rullager

Examensarbete inom högskoleingenjörsprogrammet Elektroteknik.

Oskar Manfredi

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2022

www.chalmers.se

Sammanfattning

Rapporten är en förstudie för SKF Sverige AB och syftar till att utveckla en automationslösning med mål att ersätta ett manuellt tillverkningsmoment. Lösningen innefattar både utvärdering och programmering av industrirobot. Roboten som används i simuleringen är framtagen med hjälp av en elimineringsmatris, där olika robotars egenskaper utvärderades för att hitta en lämplig modell. För att optimera roboten mot behovet är ett verktyg framtaget. Verktyget som används på roboten är utformat för att gripa och rotera en styrring i en ytterring. Robotcellen är programmerad och simulerad i ABBs robotsimulerings mjukvara RobotStudio.

Abstract

The report is a preliminary study for SKF Sverige AB and aims to develop an automation solution with the goal of replacing a manual manufacturing step. The solution includes both evaluation and programming of an industrial robot. The robot used in the simulation was chosen using an elimination matrix, where different robots' characteristics were evaluated to find a suitable model. To optimize the robot for the need, a tool was developed. The tool used on the robot is designed to grip and rotate a guide ring in an outer ring. The robot cell is programmed and simulated in ABB's robot simulation software RobotStudio.

Innehåll

1 Inledning

Den fjärde industriella revolutionen skapar nya möjligheter och utmaningar för producerande företag. Artificiell intelligens, internet och robotteknik är delar i den fjärde industriella revolutionen och syftar till att öka produktiviteten i industrin. Det är både fysiska processer och cyberfysiska system som utvecklas för att skapa smarta fabriker [1]. Den fjärde industriella revolutionen innebär stora förändringar i produktionskedjan för företag och nya lösningar måste undersökas.

Ett företag som ställs inför denna revolution är SKF Sverige AB. Genom investeringar i SKF:s fabriker har ett helt automatiserat tillverkningsflöde utvecklats där digital teknik driver nya processer. Målet med investeringarna för SKF är att öka produktiviteten och begränsa resursslöseri. Genom att flödet sköts av robotar tillverkas rullager med hög precision och i slutet av produktens livslängd kan de rekonditioneras, vilket minskar energianvändning med 80% jämfört med nytillverkade. Denna investering bidrar till mindre miljöpåverkan och sänker även kundens kostnader [2].

1.1 Bakgrund

SKF är ett globalt företag som arbetar med utveckling, tillverkning av lager, tätningar och smörjsystem. Enligt SKF:s årsredovisning år 2020 består företaget av 40 963 medarbetare och har 91 tillverkningsenheter internationellt [2]. År 2015 satsade SKF Sverige på att modernisera sina produktionskanaler för sfäriska rullager i fabriken i Göteborg [3]. Dessa kanaler är fullt automatiserade.

Rullager används i många olika applikationer och processer. De kan användas i gruvdrift, metallbearbetning, i vindkraftverk och fordon [2]. Rullager består av ytterring, innerring, två hållare, rullar och en styrring. Det finns olika varianter av rullager, men alla syftar till att minska friktionen och främja rörelse i sitt sammanhang.

Fabriken i Göteborg tillverkar ett speciellt lager som ska tåla extra kraftiga vibrationer. Vid montering av nämnda lager används en speciell styrring. Vid monteringen ska styrringen vila mot ytterringen till skillnad från ett vanligt lager där styrringen vilar mot hållare och rullar. I den tillverkningsprocess som existerar vid utförandet av examensarbetet så måste en operatör manuellt föra in styrringen i ytterringen. Detta är ett arbetsmoment som SKF vill automatisera.

SKF har samarbetspartners för utveckling av tillverkningsprocesser. För att identifiera nya möjligheter med hjälp av automation samarbetar SKF med teknikbolaget ABB. SKF och ABB ska tillsammans identifiera och åtgärda problem för att förbättra de brister som finns i tillverkningskapaciteten [4].

1.2 Syfte

På uppdrag av SKF syftar denna förstudie till att undersöka hur en robotarm kan föra in en styrring i en ytterring som en del i tillverkningskedjan. Det undersökta resultatet och simuleringar av robotarmen ska vara ett hjälpmedel för SKF att fatta ett beslut om inköp av robot.

1.3 Frågeställning

Följande frågeställningar besvaras i denna förstudie:

- Hur kan en robotarm användas i SKF:s produktion för att montera en styrring i en ytterring?
- Vilka tekniska förutsättningar krävs av roboten för att utföra monteringen?
- Hur ska ett verktyg konstrueras för att hantera ingående material och tillsammans med robotarmen få in styrringen i ytterringen?

1.4 Avgränsningar

Denna förstudie undersöker hur en robotarm kan tillämpas för att montera en styrring i en ytterring. Med hänsyn till uppdragets omfattning kommer inte en robot monteras, utan detta arbete avgränsas till en simulering. Undersökning av eventuell installation är inget som ska genomföras i denna rapport för SKF.

2 Teknisk bakgrund

Detta kapitel utgörs av en fördjupad beskrivning av de teoretiska och tekniska termerna som detta projekt består av.

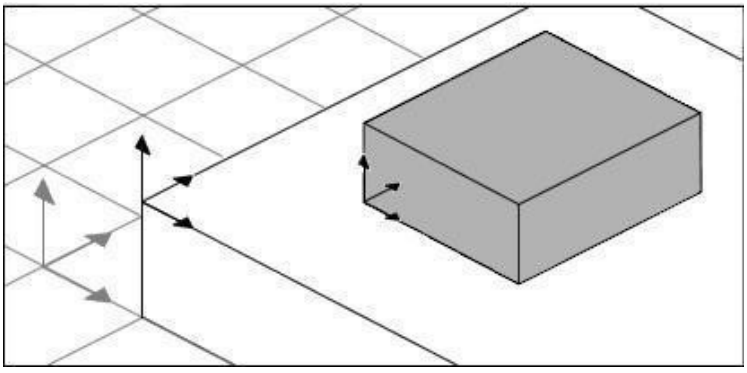
2.1 PTC Creo

PTC Creo är en mjukvara för datorstödd konstruktion (CAD) som används för att konstruera och designa komponenter och ritningar. I CAD-program designas modeller i en tredimensionell miljö. Genom användningen av CAD-program så får ritningarna en exakt noggrannhet. PTC Creo är ett av de äldsta CAD-programmen och lanserades år 1988 [5].

2.2 RobotStudio

RobotStudio är ett verktyg för offline-programmering av 3D-robotar. RobotStudio är en mjukvara från ABB för att simulera och programmera robotar. Offline-programmering är programmering av en simulerad robot i stället för att programmera en verklig robot. De robotar som simuleras är en digitaltvilling av den roboten som sedan kan införskaffas. Med en digitaltvilling kan effekter ses direkt vid förändringar i programmeringen och stör inte den verkliga driften. Den förändring som gjorts på den simulerade roboten kan sedan appliceras på den verkliga roboten och syftar till att minimera tiden som den verkliga roboten behöver stoppas [6].

Vid simulering av en robot i RobotStudio så förflyttar robotarmen sig i tredimensionella koordinatsystem. Koordinatsystemen har en hierarkisk uppbyggnad, där origo för ett koordinatsystem är definierat som en punkt i ett tidigare system. Huvudkoordinatsystemet har sin origo vid basen av roboten. De verktyg som fästs på roboten har sitt eget koordinatsystem som definieras i huvudkoordinatsystet. Origo för verktygets koordinatsystem kallas för Tool Center Point (TCP). Arbetsområden där roboten ska utföra en uppgift kallas work-objekt, som också är ett koordinatsystem. I detta koordinatsystem sätts de punkter som robotarmen ska förflytta sig till. Det är bra att använda sig av ett work-objekt för om den simulerade omgivningen inte stämmer överens med den verkliga, så behöver man endast ändra work-objektets ursprungsposition i stället för att ändra varje punkt i systemet [7]. I figur 1 illustreras hur de olika koordinatsystemen bygger på varandra. Det gråa stora koordinatsystemet är huvudsystemet, det svarta koordinatsystemet illustrerar work-objektet. Det svarta koordinatsystemet vid det grå blocket är en punkt i work-objektet.



Figur 1. Illustration över koordinatsystemuppbyggnad i RobotStudio

2.2.1 Rapid

Rapid är programmeringsspråket som används vid programmering av ABB:s robotar [8]. I RobotStudio används Rapid för att programmera robotens rörelser och hantering av signaler. Rapid är ett högnivåspråk. De klassiska funktionerna för ett högnivåspråk finns som IF-satser och FOR-loopar.

2.2.2 Smart Component

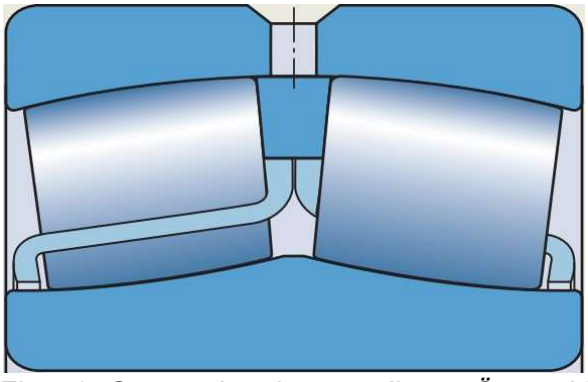
Smart Components är objekt i RobotStudio med beteende implementerat genom anpassad kod. Beteendena kan variera från att fästa objekt till varandra, att få objekt att röra på sig och att skicka signaler till och från roboten [9]. Med Smart Components kan logik för robotsimuleringen skapas, som i verkliga fall skulle komma från en PLC som hanterar signaler mellan roboten och dess omgivning.

2.2.3 SafeMove

SafeMove är en säkerhetsfunktion i RobotStudio som begränsar valda områden där roboten inte får röra sig inom [10]. SafeMove används för att skydda människor och annan utrustning. SafeMove är en del i säkerheten för en cell och kan användas med annan säkerhetsutrustning som skyddsdörrar och ljusridå.

2.3 Lager för vibrerande inbyggnader

Robotens uppgift är att montera styrring i ytterring vid produktion av lager för vibrerande inbyggnader. Lagerna används främst i skaksiktar och järnvägshjul, där stötbelastningar förekommer [11]. Vibrerande inbyggnader kan leda till kraftig acceleration på lagrets komponenter. Lager för inbyggda vibrationer är konstruerade så att lagret ska klara av höga accelerationer. Lager för vibrerande inbyggnader har två ythärdade hållare och en styrring som är centrerad på ytterringens löpbana. I figur 2 så illustreras en genomskärning av ett lager för vibrerande inbyggningar [11].



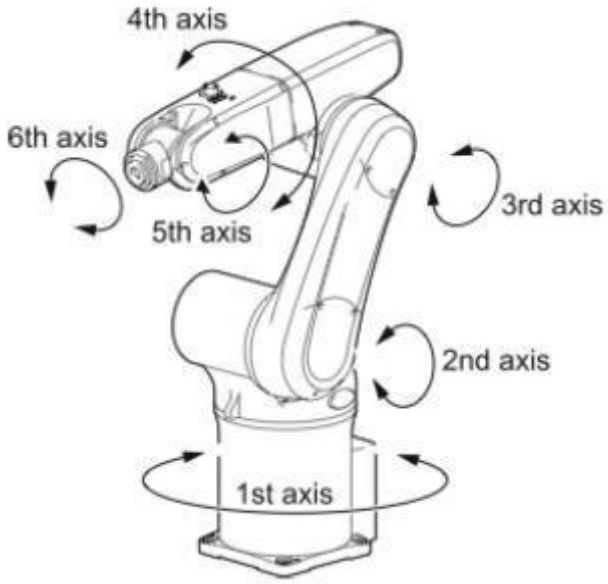
Figur 2. Genomskärning av rullager, Överst i figuren är den sfäriska ytterringen.

2.4 Styrsystem IRC5

Styrsystemet är hjärnan i stationen där all signalhantering sker. Styrsystemet som används i detta projekt är en simulerad version av ABB:s IRC5 styrsystem. Detta system används för att exekvera all Rapidkod, även signaler för verktyg och stationen [12].

2.5 Industrirobot

En industrirobot är en robotarm som är skapad för att utföra en mängd olika uppgifter med en mycket hög precision. Dess uppgifter kan innebära tunga lyft och arbete i obekväma ställningar som svetsning och målning. Användningen av en industrirobot kan leda till en minskning av utslitningsskador [13]. I figur 3 så illustreras en industrirobot med 6 axlar. Den första axeln är basen av roboten, denna axel kan rotera åt vänster och höger. Den andra axeln är robotens undre arm som flyttar hela armen fram eller tillbaka, som en armbåge. Den tredje axeln styr robotens vertikala rörelse. Den fjärde axeln agerar som robotens handled som roterar axel 5 och 6. Den femte axeln är den axel som styr positioneringen av det eventuellt monterade verktyget. Den sista axeln, axel 6, är den axel som styr hur verktyget roteras i fästet [14].



Figur 3. Illustration av axelkonfigurationen för en industrirobot. [15]. Återgiven med tillstånd.

3 Metod

Denna del av rapporten behandlar tillvägagångssättet vid val av robot och beskrivning av den iterativa processen

3.1 Val av robot

Att identifiera vilka tekniska förutsättningar som krävs är kritiskt för att välja robot. För att välja korrekt robot för denna förstudie så gjordes en elimineringsmatris. I elimineringsmatrisen utvärderades ett antal industrirobotar från ABB som potentiellt skulle kunna utföra uppgiften. Robotarna utvärderades på följande punkter:

- Räckvidd
- Hanteringsvikt
- SafeMove
- Kollaborativ
- Styrenhet
- RP

Punkterna valdes från automationsföretaget Robotiqs guide för att välja rätt robot för sin applikation [15]. Där hanteringsvikt och räckvidden är de mest kritiska punkterna. I hanteringsvikten ingår tyngden på komponenter som roboten lyfter men även de verktyg som monteras på roboten. Räckvidden är det avstånd som roboten kan arbeta på. Ytan där roboten ska installeras är begränsad med ett avstånd på 1.5 meters mellan befintliga celler, därför kan inte roboten vara för stor. RP är hur nära en robot kan komma en exakt punkt vid flera repetitioner. Lägre värde på RP indikerar på en träffsäkerrobot som kan utföra samma moment flera gånger med samma resultat. Den elimineringsmatrisen som användes finns i tabell 1.

Tabell 1 Elimineringsmatris

ROBOT	Räckvidd(mm)	VIKT(KG)	SAFEMOVE	KOLLABORATIV	Styrenhet	RP(mm)	Kompatibel

I elimineringsmatrisen så utvärderas flera robotars fysiska egenskaper för att fastställa vilken typ av robot som ska simuleras i förstudien.

3.2 Iterativ process

Den arbetsmetodik som användes i projektet är en iterativ process. Denna metod används vid utveckling av produkter och programvara. En iterativ process är en metod där upprepade test och modifieringar av en grund idé eller koncept utförs till ett tillfredställande resultat uppnåtts [16]. Varje iterations slut är början på varje ny iteration där vidare utveckling av positiva förändringar görs och avlägsna förändringar som var negativa för att åstadkomma ett bra resultat.

4 Genomförande

I denna del av rapporten presenteras resultatet. Genomförandet är uppdelad i 4 delar: Elimineringsmatrisen, verktyget, simuleringen av stationen och programmeringen av roboten.

4.1 Elimineringsmatrisen

För att välja rätt robot så användes elimineringsmatrisen i tabell 1. Här fylldes de olika robotarnas egenskaper och begränsningar in. Informationen utvärderades för att välja roboten som är bäst lämpad för uppgiften.

Tabell 2 Elimineringsmatrisen för att välja robot

ROBOT	Räckvidd(mm)	VIKT(KG)	SAFEMOVE	KOLLABORATIV	Styrenhet	RP(mm)	Kompatibel
IRB 1100	580	4	JA	NEJ	OmniCore	0.01	NEJ
IRB 120	580	3	NEJ	NEJ	IRC5	0.01	NEJ
IRB 1200 TYPE A	700/900	5 eller 7	NEJ	NEJ	OmniCore, IRC5	0.025	NEJ
IRB 1200 TYPE B	700/900	5 eller 7	JA	NEJ	OmniCore, IRC5	0.02	JA
IRB 1300	1400	7	NEJ	NEJ	OmniCore, IRC5	0.02	NEJ
IRB 140	810	6	NEJ	NEJ	IRC5	0.03	NEJ
IRB 1410	1440	5	NEJ	NEJ	IRC5	0.025	NEJ
CRB 15000 GOFA	950	5	JA	JA	OmniCore	0.05	JA
SWIFTI CRB 1100	580	5	JA	JA	OmniCore	0.01	NEJ
IRB 14050 YuMi	559	0,5	JA	JA	OmniCore	0.02	NEJ

Efter undersökning av nio olika industrirobotar så eliminerades sju robotar på grund av att de inte uppfyllde kravet för hanteringsvikt. Den tyngsta styrringen i kanalen som roboten ska lyfta är tre kilo. De två robotarna som valdes för vidare utvärdering var IRB 1200 och CRB 15000 GOFA. CRB 15000 GOFA är en kollaborativ robot och IRB 1200 är en vanlig industrirobot. Efter diskussion med operatörer och handledare på SKF så togs beslutet om att välja IRB 1200. Syftet med den nya robotcellen är att operatörerna inte ska låsas till en uppgift utan att roboten ska vara helt självgående.

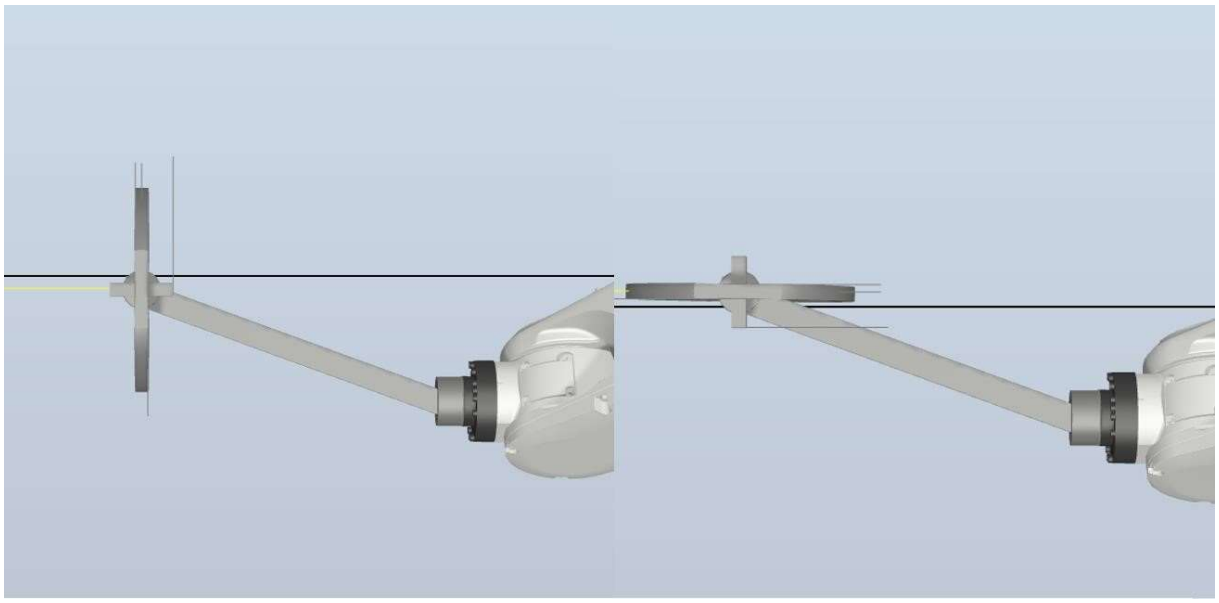
Roboten som valdes fram i elimineringsmatrisen är IRB 1200. Denna robot är tredje minsta industriroboten som ABB har i sitt sortiment. IRB 1200 har en hanteringsvikt på fem kilo. I hanteringsvikten så räknas även in vikten på de verktyg som monteras på roboten samt de som grips med verktyget. Räckvidden på IRB 1200 är 900 millimeter, ytan som roboten behöver för att klara uppgiften är inom radien 900 millimeter. IRB 1200 är en 6-axlad robot, där de olika axlarna har olika vridningsgrader och armrörelser. Den första axeln är basen på roboten som har en vridningsgrad mellan -170° och 170°. Robotens andra axel har en armrörelse på +130° till -100°, robotens tredje axel har en armrörelse på +70° till -200°, den fjärde axel har en vridningsgrad på +270° till -270°, robotens femte axel har en armrörelse på +130° till -130° och robotens sjätte och sista axeln har en vridningsgrad på +400° till -400°. IRB 1200 är kompatibel att använda sig utav rörelsebegränsning verktyget SafeMove [17].

4.2 Verktyget

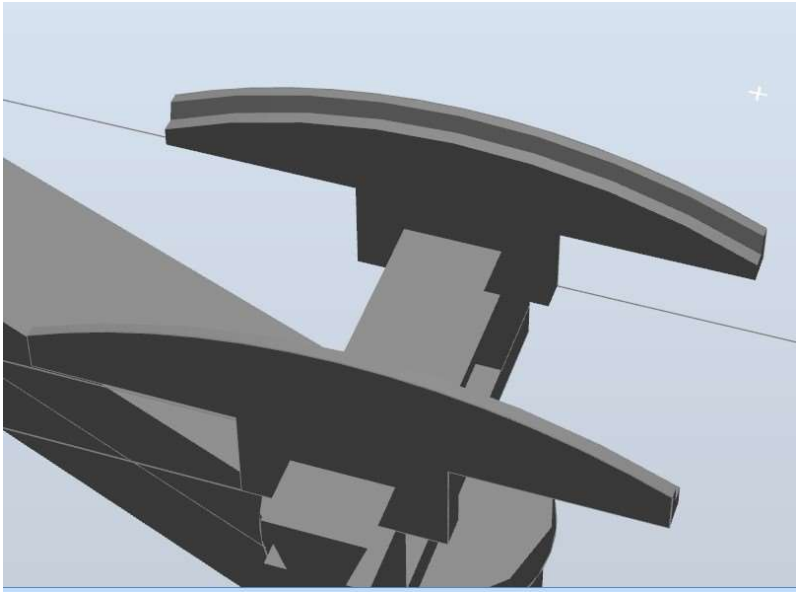
Verktyget som ska monteras på roboten konstruerades i PTC Creo och består av tre olika delar. De tre delarna är basen, den roterande delen och gripfingrarna. Verktygets syfte är att kunna gripa och rotera styrringen. För att designa ett verktyg som skulle kunna uppfylla syftet så var externa faktorer väsentliga att ta i beaktning. De externa faktorerna är vart verktygets ska fästas och vad verktyget ska greppa. Verktyget designades med den iterativa processen, där de olika delarna konstruerades i flera olika steg och vars funktion testades fram i RobotStudios kontinuerligt under hela designfasen. Designen är ett koncept utifrån hur en hand lyfter, griper och roterar styrringen i ytterringen

Verktyget designades i kronologiskordningen, där basen var den första delen som konstruerades. Basen ska fästas på roboten och sedan ska de övriga delarna av verktyget fästas på basen. Basen designades för att passa roboten sjätte axel, där ett cirkulärt hål för att fästa verktyg finns. För att kunna rotera styrringen inne i ytterringen konstruerades basen av en lång balk för att ringen inte skulle krocka med roboten. Balken konstruerades om då den först var rak men efter hopsättning av hela verktyget så upptäcktes att styrringen krockade med balken vid rotationen. En lösning på detta problem var att konstruera balken med en vinkel från fästet i roboten så att styrringen aldrig kan komma i kontakt med balken. Sista delen av basen är ett fäste för den roterande delen och gripfingrarna. Fästet består av två delar en cirkulär platta och en cylindrisk kropp. Den cylindriska kroppen är monterad på plattan. Syftet med fästet är att den roterarande delen ska kunna fästas och rotera på fästet. Den roterande delen konstruerades för att gripfingrarna skulle ha en axel att fästas och färdas på och för att skapa rotation för verktyget. Axeln där gripfingrarna ska färdas på är ett t-spår. Gripfingrarna är designade för att kunna gripa styrringen. Gripfingrarna är konstruerade för att styrringen ska passa perfekt mellan de. De har exakt samma radie och vinkel som styrringen har i sin innerbana.

Verktyget som designades klarar av de krav som ställ på det från de externa faktorerna i form av styrringen och roboten sjätte axel. I simuleringen så uppvisar verktyget på en funktion som klarar att gripa, lyfta och rotera styrringen i ytterringen. I figur 4 uppvisas hur styrringen kan roteras i gripfingrarna på verktyget. Denna rotation utgör fastlåsningen av styrringen i ytterringen. Figur 5 är en mer detaljerad bild över hur gripfingrarna är konstruerade utifrån styrringen vinkel och radie. Verktyget fungerar endast i en simulerad miljö men funktionen av verktyget skulle även vara den samma i verkligheten.



Figur 4: Bild över rotationen i verktyget



Figur 5: Gripfingrarna på verktyget.

4.3 Stationssimulering

Simuleringsmiljön uppbyggnad skapades efter verktyget var designat och roboten var vald. Det första som placerades var den valda roboten i mitten av simuleringens koordinatsystem i RobotStudios. Efter att roboten var på plats i stationen sattes banorna på plats, vars längd och bredd är samma som de verkliga banor som finns i den nuvarande kanalen. Banorna består av en rullbana och en bandtransportörsbana. Alla lager som tillverkas i kanalen åker på bandtransportbanan men endast de ringar som ska användas till lager för inbyggda vibrationer ska åka på rullbanan mot roboten. I verkligheten dras ringarna över till rullbanan med hjälp av en cylinder. Denna cylinder är inte simulerad eller konstruerad i detta projekt utan ringen dras över till rullbanan utan någon visuell hjälp. I verkligheten finns även en tilt som vrider ringarna upp mot stationen för införing av styrring. Denna tilt är inte heller simulerad utan ringen vrids utan visuell hjälp.

För att enkelt kunna applicera roboten i verkligheten är det viktigt att simuleringen görs på ett så noggrant sätt som möjligt. De simulerade delarna i robotstationen är banor, ytterring, styrring, robot med verktyg och givare. Stationssimuleringen är ett visuellt resultat från de övriga delarna i projektet. Här kan man se roboten, verktyget och alla övriga komponenter röra sig efter programkoden i styrenheten. Alla komponenter har inte en programmerad funktion. De komponenter som har en programmerad funktion är ytterringen, styrringen och roboten med verktyget. Ringarnas funktion är att färdas på rullbanorna samt att vrida sig upp mot roboten. Ringarnas rörelse i simuleringen sker med hjälp av Smart Components. Smart Components styr rörelsegivare och ringarnas förflyttningar. Simuleringens logik med tillhörande Smart Components bygger på att styrsignalen läser av signaler från stationen och att stationen läser av signaler från styrenheten. Logiken för rörelsen av styrring och ytterring har varsin Smart Component med ingångar för givare och signaler från styrenheten.

4.3.1 Stationslogik

Stationslogiken är huvudprogrammet för robotstationen, här finns all logik som behövs för att simulera stationen. I logiken för stationen så utförs ett flertal funktioner som hanterar kommunikation mellan styrenheten och den simulerade cellen. Logiken består av alla in och utgångar till styrenheten, alla givare och två block som hanterar rörelsen för ringarna. Givarna består av block som kallas LineSensor. Ett LineSensor-block har en start och en slut koordinat, mellan start och slut koordinaterna så bildas en linje. När ett objekt rör denna linje

så aktiveras blockets utsignal. Givarna används för att stoppa ringarnas rörelse. En sensor används som avståndsgivare på roboten. Avståndsgivarens utgång går till styrenheten. I stationslogiken finns även block som heter Attacher och Detacher, dessa block används för att fästa komponenter i varandra. Attacher används för att fästa styrringen i verktyget. Så när roboten griper styrringen i verktyget så följer det med verktyget i simulationen.

4.3.2 Logik för styrringen

Rörelsen för styrringen sker i stationslogiken i ett Smart Component-block. Logiken i detta block styr all rörelse för styrringen. Signalen för att starta styrringens rörelse kommer från styrenheten, startsignalen blir aktiv när simulationen startas. Signalen från styrenheten aktiverar ett rörelseblock som flyttar ringen i simuleringen. Rörelseblocket heter LinearMover och flyttar styrringen en bestämd riktning och med en bestämd hastighet. Styrringen transporteras på en rullbana fram till en bandtransportörsbana. Framme vid bandtransportbanan påverkar ringen en givare och transporteras med ett annat rörelseblock mot roboten. När ringen åkt fram till roboten så påverkas en givare som skickar en signal till att börja vrida ringen för att roboten ska plocka den. När ringen är i läge för att hämtas av roboten, skickas en signal tillbaka till styrenheten som skickar en signal till roboten för att hämta styrringen. När roboten hämtat styrringen så fäster styrringen mot robotens verktyg. Att fästa styrringen mot görs av ett Attacher block. Därefter väntar roboten med styrringen i verktyget på att ytterringen är i läge. Blocket för styringens logik får en signal från styrenheten när ytterringen har kommit fram till roboten. Denna signal aktiverar en rotation av styrringen samtidigt som verktyget roterar. När styrringen har roterats vinkelrätt mot ytterringen så för roboten in styrringen i ytterringen. När roboten har fört in styrringen i ytterringen, så skickas en signal från styrenheten till blocket och verktyget som roterar tillbaka styrringen till sitt ursprungliga läge. När roboten roterat tillbaka styrringen till sitt ursprungliga tillstånd så skickar styrenheten en signal från styrenheten. Signalen är kopplad till ett Detacher block som släpper styrringen från verktyget.

4.3.3 Logik för ytterringen

Likt styrringen så har ytterringen egen logik som styr ytterringens rörelse. Logiken består av en Smart Component som innehåller flera block. Ytterringen går på samma banor som styrringen. Det som startar ytterringens transport kommer från styrenheten och sätts av att roboten har plockat upp styrringen. Signalen som startar transporten gör även ytterringen synlig i simuleringen. På samma sätt som styrringen så transporteras ytterringen fram till bandtransportbanan, där transporten avbryts när ytterringen påverkar en givare. Givaren startar transporten mot roboten. Framme vid roboten så roteras ytterringen 90° för att roboten ska kunna föra in styrringen i ytterringen.

4.4 Programmering av roboten

Vid programmeringen av roboten så användes ABB:s egna programmeringsspråk Rapid. I Rapid skrivs kod för robotstyrning i olika moduler. Koden som exekveras programmeras i huvudmodulen. För att koden ska vara så enkel att följa, utfördes koden i enlighet med SKF:s riktlinjer. Riktlinjerna framgick muntligt och i exempel från andra program. För att säkerställa att riktlinjerna följts så upprättes en dialog med en driftsäkerhetsingenjör på SKF. Riktlinjerna innehöll vilka typer av funktioner som skulle vara i det olika modulerna, hur de olika variablerna ska namnges och hur man kommenterar kod. Det finns 6 moduler för detta projekt, där de olika moduler innehåller olika typer av funktioner och variabler. De moduler som behövs för projektet är:

- DataModule
- MoveModule
- ToolFunction
- ToolWobj
- FunctionModule
- MainModule

Genom att dela upp koden i olika moduler är det enkelt att korrigera koden vid behov. Till exempel om den verkliga platsen där roboten ska utföra uppgiften skiljer sig ifrån den simulerade platsen, så kan man gå in i den korrekta modulen som i detta fall är ToolWobj och redigera positionen för workobjectet. Följande del kommer beskriva vilken funktion de olika modulerna har för projektet. All programkod som skrivits går att läsa i appendix A.

4.4.1 DataModule

I DataModule lagras alla positioner för de olika platser som roboten behöver röra sig till för att kunna utföra uppgiften. Funktionen som används för att skapa robotpositioner i Rapid kallas för robtarget. Positionerna kan skapas på två olika sätt. Första sättet är genom att manuellt köra roboten till en önskvärd plats, och sedan spara positionen som en robtarget. Detta sätt är bra att använda sig av när man vill skapa ett hemmaläge eller röra sig till en position som inte behöver preciseras till 100 %, då detta sätt är snabbt och smidigt. Det andra sättet är att manuellt fylla i de indatavariabler som funktionen kräver. Robtarget kräver fyra indatavariabler i from av en lista, de fyra olika indatavariablerna är positionen, rotationen, robotens konfiguration och externa axlar [18]. Positionen ges i tre koordinater. Den första koordinaten är x-koordinaten, den andra är y-koordinaten och den tredje är zkoordinaten. Koordinaterna är i förhållande till det aktiva work objectet i millimeter, detta är bra för om roboten ska göra samma rörelse på flera olika ställen i cellen så kan man använda samma robtarget fast ändra work objectet. Rotationen för funktionen anges i kvaternioner, rotation med kvaternioner har inte behövts för att roboten ska kunna utföra uppgiften i detta projekt, då robotens verktyg har haft samma rotation som work objectet. Tredje indatavariabeln är robotens konfiguration där listan består av fyra nummer, dessa nummer är kopplade till hur robotens axlar är roterade. Genom att tilldela de olika axlarna ett nummer så kan ändras deras konfiguration.

Tabell 4 Konfigurationsnummer till förhållande för robotaxelns rotation.

Nummer	Rotation
0	0 ° till 90°
1	90° till 180°
2	180° till 270°
3	270° till 360°
-1	0° till -90°

Vissa axlar kan rotera mer än andra. Det första numret bestämmer hur den första robotaxeln, det vill säga den roterade basen, är roterad. I detta projekt så är roboten riktad mot platsen där uppgiften ska utföras, så värdet på denna indatavariabel kommer alltid vara noll. Det andra numret är för att konfigurerarobotens fjärde axel. Tredje numret konfigurerar robotens sjätte axel. Sista numret används för att konfigurera någon extern axel. Detta används ej i detta projekt så detta nummer kommer förbli noll.
Sista indatavariabeln är externa axlar, där numret 9E9 betyder att ingen extern axel är kopplad till roboten. Man kan koppla sex externa axlar till roboten. Axlar kan vara roterande eller linjära. De linjära axlarna definieras i millimeter ifrån en kalibrerad punkt och de roterande axlarna definieras i grader ifrån samma punkt. Vilken av dessa nummer som korrelerar till vilken axel väljer man i systemets parametrar. Detta projekt kommer inte använda sig av några externa axlar.

Exempel på data av typen robtarget

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

I exemplet är namnet på robtargeten är p15 och det är en konstant, värdet på indatavariablerna kan inte ändras under tiden som roboten exekverar koden. Alla indatavariabler kan utläsas från exemplet. Här syns det att positionen för robtarget är 600 mm i positivt x-led, 500 mm i positivt y-led och 225,3 mm i positivt z-led. Verktygets rotation tolkas från den andra indatavariabeln, det går att utläsa att robotens verktyg är i samma vinkel som koordinatsystem. Den tredje indatavariabeln ger robotens konfiguration, där syns det att robotens bas och robotens fjärde axel har roterat 90 grader. Användningen av någon extern axel har inte implanterats.

DataModule innehåller ett flertal olika robtargets för de olika platserna som roboten behöver köra till för att kunna utföra uppgiften. Enligt SKF:s riktlinjer ska alla robtargets börja med ett litet p, till exempel pGrabRJ är namnet på robtargeten vars position är där roboten ska gripa styrringen. Alla värden på robtargetsen är samma förutom position delen, då de olika robtargets använder samma rotation, samma externa axlar och samma robotkonfiguration. Positionerna framtogs genom att manuellt köra roboten till närliggande positioner och simulera körningen. Genom att kontinuerligt korrigera positionerna på de olika robtargetsen så framkom bra värden. Vissa koordinater kunde räknas fram med vetskap om ringarnas förhållande till work objectet och dimensionen på verktyget. De sista positionerna i DataModule kommer skrivas över under exekveringen av koden, därför är de av typen PERS, som är en persistent variabel. De behåller sitt värde från föregående körning om de inte ändrats under tiden som koden exekveras [18].

4.4.2 MoveModule

Denna modul innehåller alla förflyttningar som roboten utför. Förflyttningarna utgår från de robtargets som är angivna i modulen DataModule. Förflyttningarna är processer som innehåller Move-funktioner. Detta projekt använder sig av två olika Move-funktioner, MoveL och MoveJ. MoveL för flyttar robotens aktiva verktyg linjärt i koordinatsystemet och MoveJ förflyttar verktyget den lättaste vägen för roboten, det vill säga den vägen där minst axlar behöver röra sig. Move-funktionerna har samma typ av indatavariabler som i DataModule[18]. I detta arbete har användandet av offset funktionen i Move-funktionerna används flitigt. Offset-funktionen gör att man kan röra sig till positioner förhållande till de robtargets som är angivna i DataModule, genom användandet av offset-funktionen kan man närma sig robtargeten med olika hastighet.

Exempel på en Move-funktion med Offset-funktion:
MoveL offs(pGrabRJ,0,0,-50),v50,z50,tGripper\WObj:=wBana;

MoveL offs(pGrabRJ,0,0,-10),v10,z50,tGripper\WObj:=wBana; MoveL pGrabRJ,v10,z50,tGripper\WObj:=wBana;

Exemplet ovan visar hur användning av MoveL för att roboten ska närma sig robtargeten. Indatavariablerna för MoveL är robtarget, hastighet, zon, verktyg och workobject. Vid användandet av offset-funktionen så ersätts robtarget med funktionen. I offset-funktionen så är indatavariablerna: en robtarget, en x-koordinat, en y-koordinat och en z-koordinat. Koordinaterna är offset punkter så robtarget blir förflyttad med de koordinaterna man väljer. I exemplet så närmar roboten sig robtargeten underifrån, genom att gå från ett negativt z värde. Även hastigheten ändrar sig och minskar när roboten närmar sig robtargeten. Hastighet fås från numret efter v:et, till exempel v50 är en hastighet på 50 mm/s. Zonen bestämmer hur stor radie roboten kan ta ut en sväng, det vill säga om z0 är angiven som zon så kommer roboten svänga exakt enligt de programmerade punkterna. Detta är inte alltid önskvärt och genom att ge roboten en radie att svänga på, så kan roboten köra en smidigare väg. Variabeln för verktyget väljer vilket verktyg som ska användas vid körning. Då en robot kan ha flera verktyg så är det viktigt att korrekt verktyg väljs för rätt Move-funktion. Workobjectet är platser i koordinatsystemet som rörelsen ska utföras. Detta projekt använder sig enbart av ett workobject. En process i MoveModule får MM_ som prefix när den namngivs, processerna har oftast flera Move-funktioner, genom att ha hela robotvägen vald i processen så blir koden som exekveras i Main modulen mycket mindre.

4.4.3 ToolWobj

I modulen ToolWobj finns datan för workobjectet och för de olika verktygen. Wobjdata funktionen skapar ett nytt koordinatsystem för en vald yta. I detta fall så genererades datan av RobotStudio, genom att skapa en frame. En frame blir ett nytt koordinatsystem i huvudkoordinatsystemet. De workobject som används i detta projekt är rullbanan där ringarna transporteras. De position som framen har är högra kanten på sista rullen på rullbanan. När framen var skapt så konverteras det över till ett work object och sparas i ToolWobj modulen. ToolData skapar ett liknade koordinatsystem fast för verktygen. ToolDatan skapades genom att importera komponenterna från Creo till RobotStudio, när komponenterna från Creo var inlagda i RobotStudio så skapades en Mechanism. En mechanismen skapar sammankopplingar mellan de olika komponenterna för verktyget det vill säga hur de olika komponenterna som är fästa mot varandra ska röra sig. Exempel på detta är hur verktyget ska kunna rotera och hur gripfingrarna rör sig. Denna Mechanism sparas i Rapid som Tooldata.

4.4.4 ToolModule

I denna modul innehåller alla funktioner som påverkar verktygen, i detta fall är det roteringen och gripfingrarnas rörelse som styrs. När verktyget skapades så fick det fasta positioner som de kunde röra sig mellan. Rotationen var antingen vinkelrät mot ringen eller samma riktning. Gripfingrarna kan vara i 3 lägen, öppna, stängda eller hålla fast ringen. Funktionerna används för att verktyget ska röra på sig men även som utgångar till styrenheten så att den simulerade cellen vet om när roboten griper ringen.

4.4.5 FunctionModule

Koden i FunctionModule är funktioner som används i huvudmodulen, anledning till att koden delas upp är att huvudfunktionen ska vara enklare att följa. Den funktionen som finns i FunctionModule är till för att hitta centrum av ringen med hjälp av en lasersensor. Den använder sig av en tre punkts mätning för att räkna ut centrumpunkten av en ring. Koden använder sig av en funktion som heter SearchL. Där roboten kör i en linjär bana ifrån en approximerad mittpunkt, med lasern aktiverad. När lasersensorn indikerar på en träffad punkt så lagras den punkten till en robtarget. De tre punkterna är jämnt fördelande över ytterringen med 120° mellan varandra. För att räkna ut vart varje punkt skulle vara i koordinatsystemet så användes funktionen offset ifrån den approximerade mittpunkten. De uträknade punkterna är approximerade där ringen borde vara fördelad jämnt, i verkligheten kommer inte

alla ringar vara exakt placerade som i simuleringen. Då extern påverkan kan ändra ringens position. Det är därför det är viktigt att göra en tre punkts mätning och hitta ringens exakta mittposition för montering av styrring i ytterring. De tre punkterna är uträknade medhjälp av enhetscirkeln, där radien av enhetscirkeln är den samma som ytterringen. Då punkt 1 är 90° rakt upp, det vill säga enbart hela radien på z-axeln, så har de andra punkterna en 120° förflyttning.

Tabell 5 Uträkning av punkter för tre punkts mätning.

	Punkt 1 (90°)	Punkt 2 (210°)	Punkt 3 (330°)
Z axel	85 mm	$\sin 210\ 85 = -42.5$	$\sin 330\ 85 = -42.5$
Y axel	0 mm	$\cos 210\ 85 = -73.61$	$\cos 330\ 85 = 73.61$

Den punkten där lasern indikerar på att ringen är sparas, när lasern hittat ringen i de tre punkterna räknar roboten ut en ny mer exakt mittpunkt. Denna punkt används i modulen MoveModule, där verktygets parametrar läggs till på mitt punkten för att komma till läget för att föra in styrringen.

4.4.6 MainModule

MainModule är huvudmodulen. I huvudmodulen bestäms ordningen på hur koden ska exekveras. Huvudkoden har ganska få kodrader och ska vara enkel att förstå. Koden i huvudmodulen kallar på funktioner från de andra modulerna. Huvudmodulen hämtar och skickar signaler till styrenheten som i sin tur skickar signaler vidare till de egentillverkade blocken och stationslogiken. De signaler som skickas mellan huvudmodulen och styrenheten består av handskakssignaler. Handskakssignalerna mellan styrenheten och huvudmodulen används för givarfunktionen när roboten väntar på att komponenterna transporteras på banan för att plockas.

Funktionen i huvudmodulen är att roboten väntar på att styrringen ska vara i läge för att roboten ska gripa ringen i verktyget. När ringen är i läge för att bli hämtad så griper roboten ringen i verktyget och skickar en startsignal till styrenheten för att skicka fram ytterringen på transportbanan. Därefter väntar roboten på att ringen ska transporteras fram till läget för att roboten ska kunna föra in styrringen in i ytterringen. När ytterringen är framme vid roboten så utför roboten trepunktsmätningen och för sedan in styrringen i ytterringen.

5. Resultat

Resultatet för förstudien är en fungerande robotsimulering över hur en lösning på att automatisera ett manuellt arbetsmoment. Robotsimulering består av programmoduler för alla robotens olika funktioner, stationslogik som agerar som en simulerad PLC och ett verktyg som är designat för ändamålet. I förstudien så ingick även att välja en robot som var lämplig för att utföra uppgiften. Programmodulerna är skriven i RAPID och är skriven enligt SKFs standard. Stationslogiken är gjord så att den simulerade roboten kan kommunicera med sin omgivning. Verktyget är designat för att kunna greppa en styrring och sedan rotera styrringen. Resultatet är visualiserat i RobotStudio där roboten utför uppgiften att plocka upp en styrring och föra in den i en ytterring. Där en laser används för att lokalisera centrum på ytterringen.

6. Diskussion

Genom denna förstudie har det visats en potentiell lösning på hur SKF skulle kunna automatisera momentet att föra in en styrring i en ytterring. Detta visas upp med en fungerande robotkod där en trepunktmätning används för att hitta centrum av ringen. Dock behövs det en vidare undersökning av vilken lasersensor som är bäst lämpad för uppgiften.

Verktyget som är designat för att gripa och rotera styrringen är även ett resultat som visar hur en potentiell lösning skulle kunna utformas. Detta verktyg är ett koncept på dess fysiska funktion, men inte exakt hur de olika delarna ska röra sig. För att få ett fungerande verktyg att appliceras på en verklig robot så krävs det en vidare undersökning på hur de rörliga delarna påverkas, om det ska styras av tryckluft eller en elektrisk motor.

En annan del av resultatet som behöver undersökas vidare för SKF är vilka marginaler som roboten klarar av vid införing av styrringen. All rörelse i simulationen för ringarna sker i SmartComponents i stället för att använda sig av fysiken som finns i programmet. Att använda sig av fysiken i RobotStudio var tyvärr inget alternativ för detta projekt då programmet huvudsakliga syfte är robotsimuleringen. Att testa om en robot kan föra in styrringar i ytterringar på hela sortimentet behövs utföras för att få ett mer empiriskt resultat. Testet bör utföras i en fysikmotor samt även veta med hur stor felmarginal som roboten kan ha vid utförande av detta moment.

Kommunikation till övrig utrustning är även en intressant punkt för vidare undersökning. Vilka signaler är relevanta från övrig utrustning som är kopplad till robotcellen. Alla komponenter som ska hanteras av roboten till kommer från en tidigare cell, och att allt komponenter ut från robotcellen ska direkt till en annan, så måste en kommunikation mellan befintliga cellerna och den nya robotcellen upprättas. Handskakssignaler och gemensamma nödstopp är viktigt att lyfta vid en vidare undersökning.

Ett etiskt dilemma är hur utveckling av robotar och automatisering av produktion påverkar arbetstillfällen. Digitaliseringen och en ökad automation har medfört att färre arbetare behövs i tillverkningsprocesser. I artikeln *”Allt mer ensamt när robotar ersätter arbetskamrater”* från Dagens Arbete [19] så lyfter arbetarna på SKF sina tankar och åsikter gällande automatiseringen i fabriken. Arbetarna förklarar att det är betydligt mer ensamt nu och att det är mer övervakning än fysiskt arbete. Gällande det fysiska arbetet så lyfter arbetarna att de

inte är lika krävande som det tidigare varit, innan digitaliseringen så kunde var och en av arbetarna lyfta 4 till 5 ton per skift. I artikeln uppger gruppordförande för verkstadsarbetarna att ingen behövde lämna SKF på grund av automatiseringen. Robotsimuleringen som presenterats i detta projekt är just en simulering och hur SKF väljer att gå vidare med den och om det kommer påverka arbetstillfällena får SKF besluta. Men min egen åsikt är att inga arbetstillfällen kommer påverkas, eftersom roboten kommer bli en del av tillverkningsprocessen som redan övervakas av arbetare. En ytterligare robot skulle även kräva samma övervakning. SKF skriver i sin årsredovisning [2] att robotar och automatiseringen har bidragit till en mer hållbarutveckling då SKF kan producera superprecisionslager som minskar resursslöseri hos deras kunder. Med hållbarhetsaspekten och det minskade monotona och fysiska arbetet för arbetarna så tror jag att det är rätt väg att fortsätta att automatisera tillverkningsprocessen.

7 Slutsats

Den simulerade roboten kan gripa styrringen och föra in den i ytterringen. Detta gör roboten med hjälp av ett verktyg som är framtaget för ändamålet. Roboten söker även efter ytterringens exakta mittpunkt för att säkerställa att styrringen monteras på ett bra och säkert sätt. Robotcellen styrs av två system i robotsimuleringsmjukvaran RobotStudio, De två systemen är stationslogiken, och en simulerad robotkontroller som exekverar Rapid kod som styr roboten. Tillsammans så styr systemen all rörelse i simuleringen. Frågeställningen som ställts går att besvara. Den simulerade robotcellen som består av en IRB 1200 klarar av uppgiften. Dock så måste ytterligare undersökningar utföras för att säkerställa vilka felmarginaler roboten har för att föra in styrringen i ytterringen.

Källförteckning

[1] C. Lindholm, *"fjärde industriella revolutionen"*, www.ne.se. [Online]. Tillgänglig: https://www.ne.se/uppslagsverk/encyklopedi/l%C3%A5ng/fj%C3%A4rde-industriellarevolutione. [Hämtad: 19-Juni-2022].

[2] SKF, 2020." *Årsredovisning 2020".* [online] investor.skf.com. Tillgänglig https://investors.skf.com/sites/skf-ir/files/pr/202103032689-1.pdf [Hämtad 7-Juli-2022]

[3] SKF, 2015. "*SKF investerar 190 miljoner kroner i produktionskanalerna i Göteborg".*[online] investor.skf.com https://investors.skf.com/sites/default/files/pr/201504176583-1.pdf [Hämtad 9-Juli 2022]

[4] SKF, 2022. "*SKF och ABB ökar sammarbete inom industriell automation".* [online] investor.skf.com. Tillgänglig https://www.skf.com/se/news-and-events/news/2022/2022-jul-07-skf-och-abb-utokarsamarbetet-inom-industriell-automation-4306314 [Hämtad 9-Juli-2022]

[5] M. Brunelli, "*A Quick History of PTC and PTC Creo".* www.ptc.com, 2014 [Online]. Tillgänglig: https://www.ptc.com/en/blogs/cad/a-quick-history-of-ptc-and-ptc-creo [Hämtad 9Juli-2022]

[6] *"Robotsimulering av 3D robot i vår RobotStudio"*, new.ABB.com, 2022. [Online]. Tillgänglig: https://new.abb.com/products/robotics/sv/robotstudio. [Hämtad: 15-Juni-2022].

[7] *"Coordinate Systems"*, Developercenter.robotstudio.com, 2021. [Online]. Tillgänglig: https://developercenter.robotstudio.com/api/robotstudio/articles/Concepts/Math/CoordinateSystems.html. [Hämtad: 13-Juni-2022].

[8] *"Introduction"*, Developercenter.robotstudio.com, 2021. [Online]. Tillgänglig: https://developercenter.robotstudio.com/api/robotstudio/articles/Concepts/Rapid/What-isRapid.html. [Hämtad: 13-Juni-2022].

[9] *"Introduction"*, Developercenter.robotstudio.com, 2021. [Online]. Tillgänglig: https://developercenter.robotstudio.com/api/robotstudio/articles/Concepts/SmartComponentTopics/SmartComponentIntroduction.html [Hämtad: 13-Juni-2022].

[10] *"SafeMove"*, new.abb.com, 2022. [Online]. Tillgänglig: https://new.abb.com/products/robotics/controllers/safemove. [Hämtad: 20-Juni-2022].

[11] SKF, 2019. *Rullningslager.* [online] skf.com. Tillgänglig: https://www.skf.com/binaries/pub19/Images/0901d1968096b7e7-Rolling-bearings--17000_1-SV_tcm_19-121486.pdf [Hämtad 16-Juni-2022]. pp 778-779

[12] "IRC5 Controller overview", new.abb.com, 2022. [Online]. Tillgänglig: https://new.abb.com/products/robotics/controllers/irc5-overview. [Hämtad: 20-Juni-2022].

[13] "Industri Robotar", new.abb.com, 2022. [Online]. Tillgänglig: https://new.abb.com/se/omabb/teknik/sa-funkar-det/industrirobotar. [Hämtad: 20-Juni-2022].

[14] "What Are Six Axis Robots?", RobotWorx, 2011. [Online]. Tillgänglig: https://www.robots.com/faq/what-are-six-axis-robots. [Hämtad: 20-Juni-2022].

[15] [M. Bélanger-Barrette, "How to Choose the Right Industrial Robot?", Blog.robotiq.com, 2021. [Online]. Tillgänglig: https://blog.robotiq.com/how-to-choose-the-right-industrial-robot. [Hämtad: 12- Maj- 2022].

[16] Martins, J. 2024 Förstå iterativa processer, med exempel, Asana. Tillgänglig: https://asana.com/sv/resources/iterative-process [Hämtad: 01 September 2024].

[17] ABB, 2019. *Product specification: IRB 1200* [online] ABB.com Tillgänglig: https://library.e.abb.com/public/5545a515e1c7454594132dfd715368d0/3HAC046982%20PS%20IRB%201200-en.pdf?x-sign=YaHf7XOLWpmATGlyPOWZYsvo2ynpdapDrLoWshVWoGepf2OD/2e+pnucBmuWILs9 [Hämtad 16 juli 2022]

[18] Technical reference manual RAPID Instructions, Functions and Data types. ABB, 2010, p. 1176, 235. Tillgänglig: https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf [Hämtad 16 juni 2022]

[19] Garu, H. 2020. *Allt mer ensamt när robotar ersätter arbetskamrater.* Dagens Arbete Tillgänglig: https://da.se/2020/03/allt-mer-ensamt-nar-robotar-ersatter-arbetskamrater/ [Hämtad 4 Januari 2025]

Appendix A

Alla moduler för roboten

```
MODULE DataModule
    !*********************************************************
**
    !This module contains the positions for the robot
different robtargets.
    !*********************************************************
    !Robot targets
CONST robtarget
pGrabRJ:=[[22.855,192.34,154.67],[1,0,0,0],[0,2,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget
pRJtoOR:=[[59.3,178.11,173.69],[1,0,0,0],[0,2,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget pHOME:=[[-100,182.5,100],[1,0,0,0],[0,2,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
!robtarget for the LaserSearch function.
    !*********************************************************
    !Approximate center of the OR
CONST robtarget
pMiddle:=[[129.29,178.5,100.58],[1,0,0,0],[0,2,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    !*********************************************************
!dummy values, will change when the program executes.
    PERS robtarget
pEdge1:=[[129.291,178.495,177.09],[1,1.77624E-6,2.4998E-6,-
5.18964E-6],[0,2,-
1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
    PERS robtarget
pEdge2:=[[129.293,112.509,62.4854],[1,3.20873E-6,-2.20355E-6,-
4.68971E-6],[0,2,-
1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
PERS robtarget
pEdge3:=[[129.292,245.151,62.1021],[1,2.16905E-
6,1.94685E7,3.88189E-6],[0,2,-
1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];    PERS robtarget
pRealCenter:=[[129.292,178.941,100.535],[1,-1.77624E-
6,2.4998E-6,-5.18964E-
6],[0,2,1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
    PERS robtarget
pRJtoORnew:=[[129.292,178.941,100.535],[1,1.77624E-6,2.4998E-
6,-5.18964E-6],[0,2,-
1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
    !*********************************************************
**
```

```
ENDMODULE

MODULE MoveModule
!*********************************************************
**
    ! This module contains the different movements that the
robot will do in a run.
    !*********************************************************
**
    !Move to the RJ
    PROC MM_GetRJ()
        MoveL offs(pGrabRJ,-
100,0,50),v500,z50,tGripper\WObj:=wBana;
        MoveL offs(pGrabRJ,-
50,0,50),v50,z50,tGripper\WObj:=wBana;
        MoveL
offs(pGrabRJ,0,0,50),v50,z50,tGripper\WObj:=wBana;
        MoveL
offs(pGrabRJ,0,0,10),v10,z50,tGripper\WObj:=wBana;
        MoveL pGrabRJ,v10,z50,tGripper\WObj:=wBana;
    ENDPROC

    !*********************************************************
**
    !Move to a temporary position
    PROC MM_Temp()
        MoveJ offs(pGrabRJ,-200,0,-
20),v500,z50,tGripper\WObj:=wBana;
    ENDPROC

    !*********************************************************
**
    !
    PROC MM_RJtoOR()
        MoveL Offs(pRJtoOR,-
200,0,0),v50,z50,tGripper\WObj:=wBana;
```

```
        MoveL
Offs(pRJtoOR,50,0,0),v50,z50,tGripper\WObj:=wBana;
        MoveL Offs(pRJtoOR,10,0,0),v5,z50,tGripper\WObj:=wBana;
        MoveL pRJtoOR,v5,z50,tGripper\WObj:=wBana;
    ENDPROC

    !*********************************************************
**
    ! Move out of the OR
    PROC MM_LeaveOR()
        MoveL
Offs(pRJtoOR,0,0,50),v10,z50,tGripper\WObj:=wBana;
        MoveL Offs(pRJtoOR,-
50,0,50),v10,z50,tGripper\WObj:=wBana;
        MoveL Offs(pRJtoOR,-50,0,-
50),v50,z50,tGripper\WObj:=wBana;
    ENDPROC

    !*********************************************************
**
    ! Move home
    PROC MM_Home()
        MoveL pHOME,v500,z50,tGripper\WObj:=wBana;
    ENDPROC

    PROC MM_RealCenter()
        MoveJ
Offs(pRealCenter,200,0,0),v500,z50,tGripper\WObj:=wBana;
        MoveL
Offs(pRealCenter,50,0,0),v100,z50,tGripper\WObj:=wbana;
        MoveL
Offs(pRealCenter,25,0,0),v50,z50,tGripper\WObj:=wbana;
        MoveL pRealCenter,v5,z50,tGripper\WObj:=wbana;
    ENDPROC
!moves to the position where RJ is inserted to OR, based on
the calculated center    PROC MM_RJtoORnew()        var
pos toolparameters;
```

```
! Parameters of the tool (the distance of the TCP of the
gripper from tool0         toolparameters.x :=-70;
        toolparameters.y:=0.6;
toolparameters.z:=73.6;
pRJtoORnew:=pRealCenter;
        MoveL offs(pRJtoORnew,toolparameters.x-
200,toolparameters.y,toolparameters.z),v100,z50,tgripper\WObj:
=wBana;
        MoveL offs(pRJtoORnew,toolparameters.x-
50,toolparameters.y,toolparameters.z),v50,z50,tgripper\WObj:=w
Bana;
        MoveL offs(pRJtoORnew,toolparameters.x-
10,toolparameters.y,toolparameters.z),v10,z50,tgripper\WObj:=w
Bana;
        MoveL
offs(pRJtoORnew,toolparameters.x,toolparameters.y,toolparamete
rs.z),v5,z50,tgripper\WObj:=wBana;
    ENDPROC

    !*********************************************************
**
ENDMODULE
MODULE
ToolFuncti
ons
    !*********************************************************
*****************

    !Functions for the gripper

    !*********************************************************
*****************



    !*********************************************************
*****************
    !Open gripper, without any rotation, Also used when you
want to rotate back to the orignal postition
    !*********************************************************
*****************
    PROC TF_OpenGripper()
IF RobOS() THEN
waittime 0.1;
        ELSE
```

```
        SetDO oOpenNoRotation,1;
SetDO oOpenNoRotation,0;
        endif
ENDPROC

    !*********************************************************
*****************
    !Grip the RJ with the gripper, Also used when you want to
rotate back to the orignal postition
    !*********************************************************
*****************
    PROC TF_GripRJ()
        SetDO oRingInGripper,1;
        SetDO oRingInGripper,0;
    ENDPROC

    !*********************************************************
*****************
    !Open gripper with a 90 deg rotation
    !*********************************************************
*****************
    PROC TF_OpenWithRotation()
        SetDO oOpenWithRotation,1;
        SetDO oOpenWithRotation,0;
    ENDPROC
!*********************************************************
*****************
    !closed gripper, without rotation. Also used when you want
to rotate back to the orignal postition
    !*********************************************************
*****************
    PROC TF_CloseGripper()
        SetDO oClosedNoRotation,1;
        SetDO oClosedNoRotation,0;
    ENDPROC

    !*********************************************************
*****************
    !Close gripper with a 90 deg rotation
    !*********************************************************
*****************
    PROC TF_CloseWithRotation()
        SetDO oClosedWithRotation,1;
        SetDO oClosedWithRotation,0;
    ENDPROC

    !*********************************************************
*****************
```

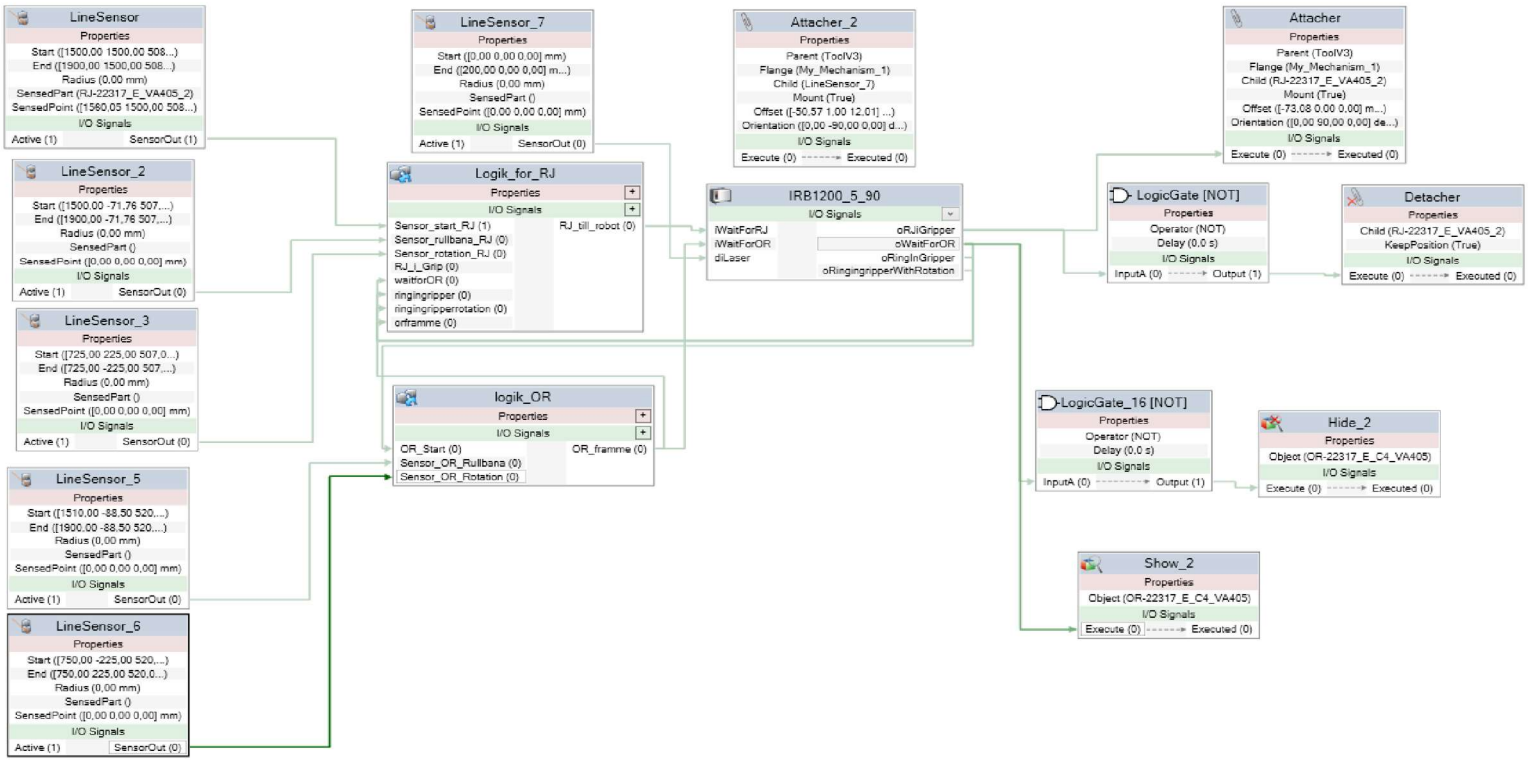
```
    !Rotate with the ring in the gripper
    !*********************************************************
*****************     PROC TF_RotateRJ()
setdo oRingingripperWithRotation,1;
SetDO oRingingripperWithRotation,0;
    ENDPROC ENDMODULE
MODULE ToolWobj
    !*********************************************************
**
    !Workobject for the roller conveyor
    !*********************************************************
**
    PERS wobjdata wBana:=[FALSE,TRUE,"",[[700,-
182.5,500],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
    !*********************************************************
**
    !ToolData for the gripper
PERS tooldata
tGripper:=[TRUE,[[50.57,89,255.27],[0,0.707107,0,0.707107]],[2
,[10,-50.1,130.16],[1,0,0,0],0,0,0]];
    PERS tooldata
My_Mechanism_1:=[TRUE,[[50.57,89,255.27],[0,0.707107,0,0.70710
7]],[2,[10,-50.1,130.16],[1,0,0,0],0,0,0]];
    !*********************************************************
**
    !ToolData for the laser
PERS tooldata
toLaser:=[TRUE,[[0,90,467.27],[0,0.707107,0,0.707107]],[0.1,[1
50,0,0],[1,0,0,0],0,0,0]]; ENDMODULE



MODULE MainModule
    !*********************************************************
**     !
    ! Module:  MainModule
!
    ! Description:
    !   This module is the main module of the program, here is
the code that will be executed by the controller and the robot.
    !
    ! Author: Oskar
!
    ! Version: 1.0
!
    !*********************************************************
**
```

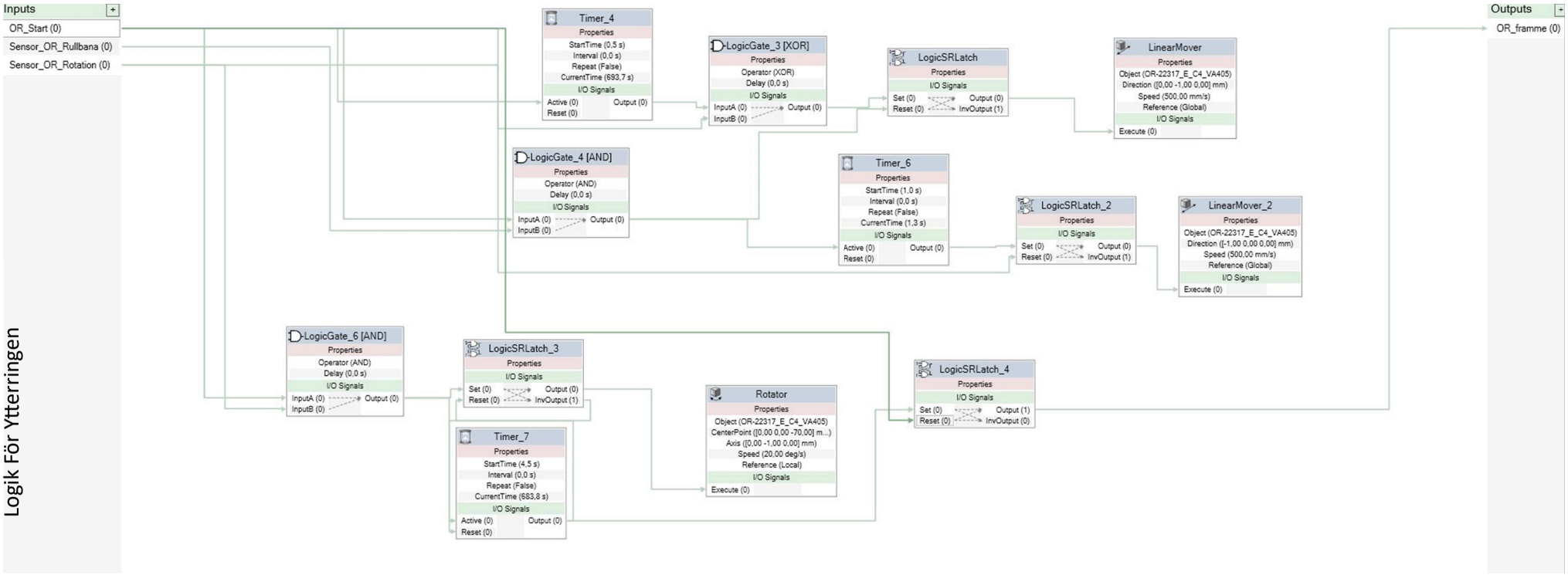
```
    !*****************************************************
**     !
    ! Procedure main
!
    !   This is the entry point of your program
!
    !*****************************************************
**



    PROC main()
        !moves to the start position.
    MM_Temp;
        !*****************************************************
******
        !set the signal for the robot to get OR to 0
setdo oWaitForOR,0;
        !*****************************************************
******
        !Wait until RJ is in position on the convoyer, signal
from sensor                 waituntil iWaitForRJ=1;
IF iWaitForRJ=1 THEN



            !sätter signaler för att öppna gripen och sätta en
signal att styrringen inte är i verktyget.            setdo
oRJiGripper,0;            TF_OpenGripper;

            waittime 2;
            !Roboten närmar sig ringen
            MM_GetRJ;
            WaitTime 2;
            !Grip ringen i verktyget
            TF_GripRJ;
WaitTime 3;             setdo
oRJiGripper,1;             !flyttar
ringen till templäge             MM_Temp;
            setdo oWaitForOR,1;
        ENDIF
        WaitUntil iWaitForOR=1;
IF iWaitForOR=1 THEN

            waittime 1;
            !Finding the real center of the OR
            FM_LaserSearch;
MM_Temp;             waittime
1;            !Rotera ringen
```

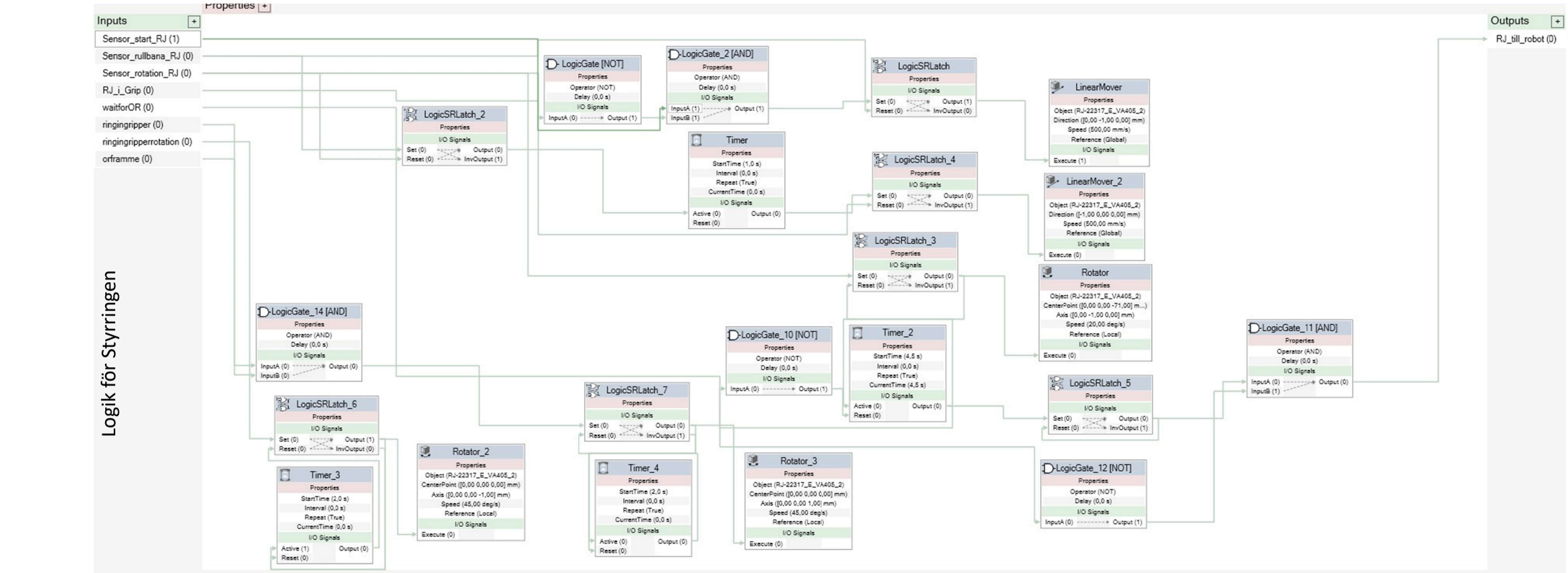
```
            TF_RotateRJ;
            !Sätter signalen för att vänta på OR till sann,
väntar på OR
            SetDO oWaitForOR,1;
waittime 1;              !går
mot ringen
            MM_RJtoORnew;
            WaitTime 2;
            !vrider ringen
TF_GripRJ;              waittime
2;            !släpper ringen
TF_OpenGripper;
setdo oRJiGripper,0;
waittime 2;
MM_LeaveOR;
            MM_Home;
            SetDO oRJinOR,1;
        ENDIF


    ENDPROC ENDMODULE
```

Appendix B

Logiken för Stationen

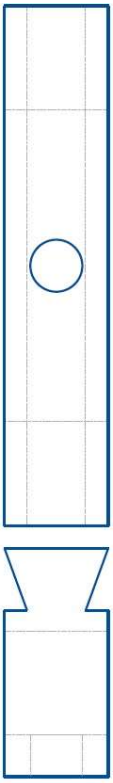


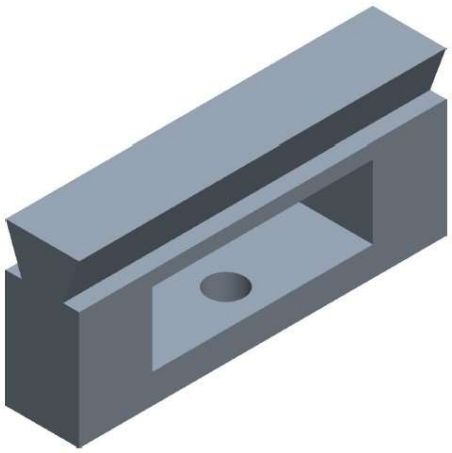

Logik För Ytterringen
Inputs
OR_Start (0)
Sensor_OR_Rullbana (0)
Sensor_OR_Rotation (0)
Outputs
OR_framme (0)
Timer_4
Properties
StartTime (0,5 s)
Interval (0,0 s)
Repeat (False)
CurrentTime (693,7 s)
I/O Signals
Active (0)
Reset (0)
Output (0)
LogicGate_3 [XOR]
Properties
Operator (XOR)
Delay (0,0 s)
I/O Signals
InputA (0)
InputB (0)
Output (0)
LogicSRLatch
Properties
I/O Signals
Set (0)
Reset (0)
Output (0)
InvOutput (1)
LinearMover
Properties
Object (OR-22317_E_C4_VA405)
Direction ([0,00 -1,00 0,00] mm)
Speed (500,00 mm/s)
Reference (Global)
I/O Signals
Execute (0)
LogicGate_4 [AND]
Properties
Operator (AND)
Delay (0,0 s)
I/O Signals
Timer_6
Properties
StartTime (1,0 s)
Interval (0,0 s)
Repeat (False)
CurrentTime (1,3 s)
LogicSRLatch_2
LinearMover_2
LogicGate_6 [AND]
LogicSRLatch_3
Rotator
Properties
Object (OR-22317_E_C4_VA405)
CenterPoint ([0,00 0,00 -70,00] m...)
Axis ([0,00 -1,00 0,00] mm)
Speed (20,00 deg/s)
Reference (Local)
LogicSRLatch_4
Timer_7
StartTime (4,5 s)
Interval (0,0 s)
Repeat (False)
CurrentTime (683,8 s)

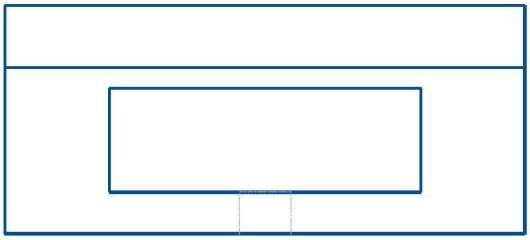

Logik för Styrringen
Inputs
Sensor_start_RJ (1)
Sensor_rullbana_RJ (0)
Sensor_rotation_RJ (0)
RJ_i_Grip (0)
waitforOR (0)
ringingripper (0)
ringingripperrotation (0)
orframme (0)
Outputs
RJ_till_robot (0)
LogicGate [NOT]
Properties
Operator (NOT)
Delay (0,0 s)
I/O Signals
InputA (0)
Output (1)
LogicGate_2 [AND]
Properties
Operator (AND)
Delay (0,0 s)
I/O Signals
InputA (1)
InputB (1)
Output (1)
LogicSRLatch
Properties
I/O Signals
Set (0)
Reset (0)
Output (1)
InvOutput (0)
LinearMover
Properties
Object (RJ-22317_E_VA405_2)
Direction ([0,00 -1,00 0,00] mm)
Speed (500,00 mm/s)
Reference (Global)
I/O Signals
Execute (1)
LogicSRLatch_2
Properties
I/O Signals
Set (0)
Reset (0)
Output (0)
InvOutput (1)
Timer
Properties
StartTime (1,0 s)
Interval (0,0 s)
Repeat (True)
CurrentTime (0,0 s)
I/O Signals
Active (0)
Reset (0)
Output (0)
LogicSRLatch_4
LinearMover_2
Execute (0)
LogicSRLatch_3
Rotator
Properties
Object (RJ-22317_E_VA405_2)
CenterPoint ([0,00 0,00 -71,00] m...)
Axis ([0,00 -1,00 0,00] mm)
Speed (20,00 deg/s)
Reference (Local)
I/O Signals
Execute (0)
LogicGate_14 [AND]
Properties
Operator (AND)
Delay (0,0 s)
I/O Signals
InputA (0)
InputB (0)
Output (0)
LogicGate_10 [NOT]
Properties
Operator (NOT)
Delay (0,0 s)
I/O Signals
InputA (0)
Output (1)
Timer_2
Properties
StartTime (4,5 s)
Interval (0,0 s)
Repeat (True)
CurrentTime (4,5 s)
I/O Signals
Active (0)
Reset (0)
Output (0)
LogicGate_11 [AND]
Properties
Operator (AND)
Delay (0,0 s)
I/O Signals
InputA (0)
InputB (1)
Output (0)
LogicSRLatch_6
Properties
I/O Signals
Set (0)
Reset (0)
Output (1)
InvOutput (0)
LogicSRLatch_7
Properties
I/O Signals
Set (0)
Reset (0)
Output (0)
InvOutput (1)
LogicSRLatch_5
Properties
I/O Signals
Set (0)
Reset (0)
Output (0)
InvOutput (1)
Timer_3
Properties
StartTime (2,0 s)
Interval (0,0 s)
Repeat (True)
CurrentTime (0,0 s)
I/O Signals
Active (1)
Reset (0)
Output (0)
Rotator_2
Properties
Object (RJ-22317_E_VA405_2)
CenterPoint ([0,00 0,00 0,00] mm)
Axis ([0,00 0,00 -1,00] mm)
Speed (45,00 deg/s)
Reference (Local)
I/O Signals
Execute (0)
Timer_4
Properties
StartTime (2,0 s)
Interval (0,0 s)
Repeat (True)
CurrentTime (0,0 s)
I/O Signals
Active (0)
Reset (0)
Output (0)
Rotator_3
Properties
Object (RJ-22317_E_VA405_2)
CenterPoint ([0,00 0,00 0,00] mm)
Axis ([0,00 0,00 1,00] mm)
Speed (45,00 deg/s)
Reference (Local)
I/O Signals
Execute (0)
LogicGate_12 [NOT]
Properties
Operator (NOT)
Delay (0,0 s)
I/O Signals
InputA (0)
Output (1)

Appendix C

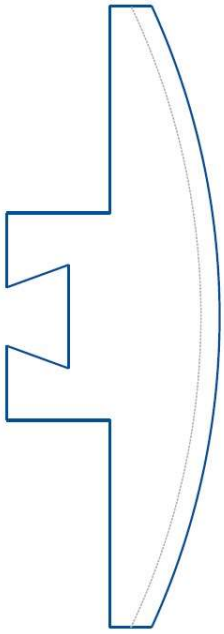
Ritningar för gripverktyget

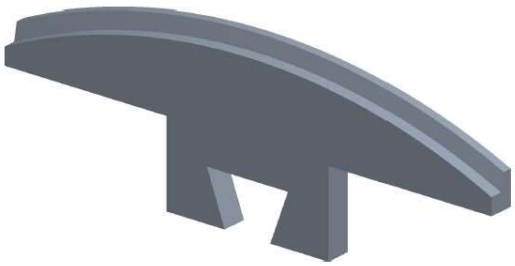




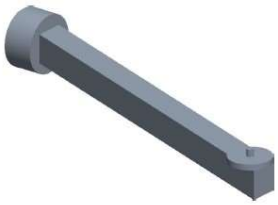


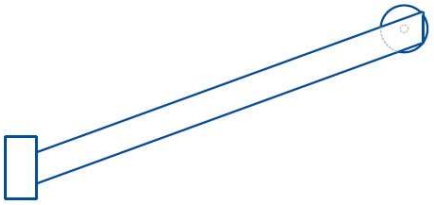
Part Name	Owner	Date
Rotator	Oskar Manfredi	2022-04-18





Part Name	Owner	Date
Grip	Oskar Manfredi	2022-04-28





Part Name	Owner	Date
Mount	Oskar Manfredi	2022-04-14

INSTITUTIONEN FÖR ELEKTROTEKNIK
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2022
www.chalmers.se