Examensarbete

HÖGSKOLAN I SKÖVDE

AUTOMATISK KALIBRERING AV TOOL CENTER POINT VIA VISIONSYSTEM

AUTOMATIC CALIBRATION OF TOOL CENTER POINT VIA VISIONSYSTEM

Examensarbete inom huvudområdet Produktionsteknik.
Grundnivå G2E, 30 Högskolepoäng
Vårtermin 2022

Karl-Filip Boman Flodin
Tobias Jonsson

Handledare på Högskolan i Skövde: Martin Birtic
Handledare på Elektroautomatik: Anton Stensson & Fredrik Wahlström

Examinator: Stefan Ericson

Sammanfattning

Elektroautomatik är ett företag som erbjuder helhetslösningar gällande automation. De arbetar med kunder och designar, tillverkar, implementerar samt erbjuder service av automationslösningar. Många av dessa projekt innehåller robotar som guidas av visionsystem och för att samarbete mellan visionsystem och robot skall fungera måste en samkalibrering utföras. En del i samkalibreringsprocessen är kalibrering av tool center point vilket idag utförs manuellt. Detta medför risken att noggrannhet blir bristande och beror på hur erfaren operatören som utför kalibreringen är. Därför presenteras en lösning för en automatiserad kalibreringsprocess som nyttjar tvådimensionella visionsystem.

Projektet delades upp i tre delmål. Dessa var kartläggning av nuvarande manuell kalibreringsprocess, utveckla en automatiserad kalibreringsprocess och att identifiera noggrannheten samt jämföra den med den manuella processens noggrannhet. De lösningsförslag och resultat som presenteras bygger på data som har identifierats utifrån metoder som förknippas med forskningsmetodiken design research methodology.

Under projektet nyttjades en mjukvara vid namnet Robotstudio för att utveckla och validera kalibreringsmetoden, vilket möjliggör snabbare testning av programmet och utvärdering av det. Arbetet resulterade i en artefakt som har använts som stöd för att utveckla den kalibreringsmetod som presenteras. Kalibreringsmetoden som presenteras är en metod som iterativt utvärderar och förflyttar tool center point tills önskat läge är uppnått.

Därefter presenteras utvärdering av den metod som har utvecklats. Utvärderingen tyder på att den automatiserade processen var mer noggrann än den manuella process som tidigare nämnts. Metoden och artefaktens fördelar och nackdelar diskuteras för att identifiera arbete som i framtiden kan utföras för att förbättra processen.

Framtida arbete kommer vara att undersöka undantag som medför att kalibreringen blir felaktig eller omöjlig att utföra samt att ytterligare arbete kommer att behöva göras för att utvärdera processens tillämpbarhet och användbarhet.

Abstract

Elektroautomatik is a company which offers complete solutions for automation. They collaborate with their customers to design, manufacture, implement and offer service of automation solutions. Many of these projects contain solutions where robots are guided by vision systems. To make the cooperation between vision system and robot possible, both systems need to be co-calibrated. One part of the co-calibration process is the calibration of the robot's tool center point. Today the calibration of tool center point is done manually. This can come with risks of insufficient accuracy and the results depend on the amount of experience the operator has. Which is why a solution for automized calibration process which utilizes two-dimensional vision systems is presented.

The project is divided into three sub-goals. These where mapping the current manual calibration process, develop an automized calibration process, identify the accuracy of the process and compare the results with that of the current calibration process. The solutions which are presented are based on data generated by methods which are associated with the design research methodology.

The calibration method was developed using a software called Robotstudio, which allows for faster testing and validation of the method. The project has produced and artifact which has been used as support for developing and evaluating the calibration method which is presented. The calibration method which is presented is a method that iteratively evaluates and corrects the position of tool center point until the desired result is achieved.

Thereafter an evaluation of the developed method is presented. The evaluation suggests that the automized solution is more accurate than the manual process. The pros and cons of the developed method is discussed to identify what future work could be done to improve the process.

Future work will be to examine exceptions where the method cannot accurately calibrate the tool center point. More work needs to be done to evaluate the applicability and utility of the process.

Intyg

Denna uppsats har lämnats in av Karl-Filip Boman Flodin och Tobias Jonsson till Högskolan i Skövde som uppsats för erhållande av betyg för på grundnivå G2E inom huvudområdet Produktionsteknik.

Undertecknande intygar härmed att allt material i denna uppsats som inte är resultatet av eget arbete har redovisats med källangivelse. Uppsatsen innehåller inte heller material som undertecknande redan tidigare fått tillgodoräknat sig inom sina akademiska studier.

Karl-Filip Boman Flodin *Tobias Jonsson*

Skövde 2022-06-02
Institutionen för Ingenjörsvetenskap

Förord

Vi vill tacka alla som har hjälpt oss under projektets genomförande. Vi tackar Elektroautomatik för ett intressant och utmanande uppdrag. Vi vill tacka alla våra handledare Martin Birtic (HiS), Anton Stensson (EA) och Fredrick Wahlström (EA). Vi vill till sist tacka Stefan Karlsson på Högskolan i Skövde för all hjälp med linjär algebra. Vi skulle också vilja tacka Martin Skoog på Elektroautomatik som har hjälpt oss med utrustning och andra problem under projektet.

Innehållsförteckning

1 Introduktion 1
1.1 Bakgrund 1
1.2 Problemformulering 1
1.3 Syfte och Mål 2
1.4 Avgränsningar 2
1.5 Hållbar utveckling 2
1.5.1 Ekologisk hållbarhet 2
1.5.2 Ekonomisk hållbarhet 3
1.5.3 Social hållbarhet 3
1.6 Rapportens disposition 4
2 Teoretisk referensram 5
2.1 Industri 4.0 5
2.1.1 Simulering 5
2.1.2 Industrial internet of things 5
2.2 Automation 5
2.3 Industriella robotar 7
2.3.1 ABB specifika data 9
2.4 Visionsystem 9
2.5 Kalibrering 10
2.6 Linjär algebra 10
3 Litteraturgenomgång 13
3.1 Robot och vision samarbete 13
3.2 Definiering av TCP med vision 14
3.3 Analys av litteraturgenomgång 15
4 Metod 17
4.1 Forskningsmetodik 17
4.1.1 Design research methodology (DRM) 17
4.2 MoSCoW 19
4.3 Litteraturgenomgång 19
5 Genomförande 21
5.1 Forskningsförtydligande 21

5.2 Datainsamling 25
5.3 Utveckling av kalibreringsmetod 25
5.3.1 Uppbyggnad av robotcell 25
5.3.2 Utvärdering av TCP 28
5.3.3 Orientering av koordinatsystem 32
5.3.4 Faktor beräkning 34
5.4 Utvärdering 34
5.4.1 Utvärdering av stöd 35
5.4.2 Utvärdering av tillämpbarhet 36
5.4.3 Framgångsutvärdering 39
6 Resultat 41
6.1 Metod för kalibrering av TCP 41
6.2 Artefakt 41
6.3 Resultat från utvärdering 42
7 Diskussion 43
7.1 Diskussion av projektets resultat 43
7.2 Diskussion om artefakten 43
7.3 Diskussion om utvärdering 43
8 Slutsatser och framtida arbete 45
Litteraturförteckning 46
Bilaga A Preliminär och aktuell tidsplan för projekt A.1
Bilaga B Data för Experiment med Kalibreringsmetoderna B.1
Bilaga C Artefakt C.1

Figurförteckning

Figur 2.1. Illustration av roboten Unimate från Unimation. (EBattleP, 2019) CC BY-SA 4.0. 7
Figur 2.2. Sexaxlig Robotarm (Eshah Budin Shah, et al., 2020). 8
Figur 4.1. De fyra stegen i DRM. 17
Figur 4.2. Anpassade DRM steg. 19
Figur 5.1. Processkarta över nuläge 21
Figur 5.2. Relationsdiagram för nuläget 22
Figur 5.3. Processkarta över önskat läge 23
Figur 5.4. Relationsdiagram av önskat läge med stöd 24
Figur 5.5. 3D-modell av fäste och kalibreringsplatta. 26
Figur 5.6. Robotcell & volymsensor med koordinatsystem. 26
Figur 5.7. Logik för simulerat visionsystem. 27
Figur 5.8. Plattans mittpunkt i relation till TCP. 28
Figur 5.9. Punkter från cirkulär rörelse. 29
Figur 5.10. Bildning av likbent triangel. 29
Figur 5.11. Bilda rätvinkliga trianglar av likbent triangel. 30
Figur 5.12. Resultat av experiment för manuell kalibrering för position för TCP i X-led från tool0. 37
Figur 5.13. Resultat av experiment för manuell kalibrering för position för TCP i Y-led från tool0. 38
Figur 5.14. Resultat av experiment för automatiserad kalibrering för position för TCP i X-led från tool0. 38
Figur 5.15. Resultat av experiment för automatiserad kalibrering för position för TCP i Y-led från tool0. 39
Figur 6.1. Diagram med kalibreringssteg 41

1 Introduktion

Introduktionen innehåller de underlag som krävs för att skapa förståelse av arbetet och dess genomförande. Följande kapitel beskriver företaget som arbetet utförs i samarbete med samt arbetets bakgrund, vilket är kopplat till problembeskrivningen.

1.1 Bakgrund

Examensarbetet utförs på företaget Elektroautomatik (EA) som i huvudsak är en helhetsleverantör inom automation. EA erbjuder design, uppbyggnad och service av automatiserade produktionslinor. När ett företag fattar ett beslut om att automatisera någon del i sitt system, exempelvis en robotcell, kontaktar de EA. Därefter designar EA en automatiserad lösning på problemet samtidigt som de har nära kontakt med kunden. När båda parterna är överens om utförandet av robotcellen kan systemering och programmering av de automatiserade delarna inledas. Det görs av en projektgrupp som ansvarar för olika delar av projektet.

I ett antal av projekten EA arbetar med används visionsystem för att guida processer. Hur systemen ser ut kan variera. I de fall vision används i samarbete med robotar krävs det att dessa samkalibrerats med visionsystemen. Det innebär att parametrar i systemen har jämförts för att skapa en gemensam bild över olika datapunkter. En position i kamerans koordinatsystem ska därefter motsvara en punkt i robotens koordinatsystem som hamnar på samma plats i den fysiska världen. Delar av den processen utförs idag manuellt, vilket leder till att noggrannheten av slutresultatet riskerar att vara otillräcklig.

1.2 Problemformulering

Elektroautomatik arbetar huvudsakligen med robotar som ingår i automationslösningar. I fallen där roboten har som uppgift att utföra arbete på ospecificerade positioner används visionsystem som vägledning. För att roboten ska förstå vilken position i dess koordinatsystem (i mm) som kameran menar med sitt koordinatsystem (i pixlar) behöver dessa två koordinatsystem samkalibreras. I nuläget görs delar av samkalibreringen automatiskt, och andra manuellt. En kalibreringsplatta fästs på roboten som kameran ska ta bilder på i olika positioner. Därefter jämförs positionerna i kamerans bilder med de positioner roboten anser att plattans mittpunkt har varit, för att sedan skapa ett relativt förhållande mellan de två systemen. För att kunna utföra kalibreringen behöver roboten känna till positionen för kalibreringsplattans mittpunkt, dess Tool Center Point (TCP). Inledningsvis skapas en virtuell cell där en TCP definieras utifrån CAD-geometri. När arbetet sedan går vidare till den fysiska världen behöver TCP kalibreras då det fysiska verktyget kan skilja sig från den virtuella miljön. Det utförs i dagsläget manuellt vilket tar tid och utgör en risk för noggrannheten vid kalibrering av TCP vilket resulterar i lägre noggrannhet i samkalibreringen. Om kalibrering av TCP är bristfällig upptäcks det inte alltid direkt och då kanske upptäckten görs efter flera timmar eller dagars arbete. Vilket leder till att

en ny samkalibrering måste göras om tillsammans med allt annat arbete som utfördes efter den bristfälliga samkalibreringen. Därför ska en generell metod för automatisk bestämning av kalibreringsplattans TCP tas fram.

1.3 Syfte och Mål

Syftet med projektet är att vid samkalibrering av robot och kamera skapa en automatiserad kalibreringsmetod för TCP som sedan utvärderas. Arbetet skall resultera i en artefakt i form av kod som skall användas för att verifiera kalibreringsmetoden. Projektet är uppdelat i tre delmål. Första delmålet innefattar en kartläggning av hur den manuella processen ser ut i nuläget. Andra delmålet är att utveckla en automatiserad kalibreringsprocess. Tredje är att få fram en noggrannhet av den aktuella kalibreringen. Den automatiserade processen ska utvärderas genom att jämföra dess noggrannhet med dagens manuella kalibreringsprocess av TCP.

1.4 Avgränsningar

Kalibrering av z-led utesluts från kalibreringsrutinen, men med mån av tid kan den bestämmas. Implementation och testning i produktion anses falla ur tidsramen för arbetet så det utesluts från utvärdering av artefakten.

1.5 Hållbar utveckling

Hållbar utveckling blev välkänt efter Brundtlandkommissionen i FN-rapporten med namnet vår gemensamma framtid, där begreppet definieras som följande: *”Sustainable development is development that meets the needs of the present without compromising the ability of future generations to meet their own needs.”* (World commission of environment and development 1978, s.41). Holmgren och Gulliksson (2018) beskriver tre förhållningssätt till hållbar utveckling, ekologisk, ekonomisk och social hållbarhet, varav ekologiska aspekten är den mest omtalade. Att uppnå hållbarhet i alla tre aspekter är ett uppdrag framtida ingenjörer får ta hänsyn till, att överväga vilka avtryck en produkt eller tjänst kan ge för det sociala, miljömässiga och ekonomiska perspektiv. Exempelvis om det är möjligt att designa en produkt med demonterbarhet i åtanke, påverkan produkten har på samhället eller vilka ekonomiska konsekvenser den kan orsaka. För att uppnå hållbarhet för detta projekt är det viktigt att ha samtliga perspektiven i åtanke vid skapandet av artefakten.

1.5.1 Ekologisk hållbarhet

För att kunna arbeta med ekologisk hållbarhet är kännedomen av ekologi ett krav. Ekologi är kort förklarat relationen mellan organismer och deras närmiljö menar Gulliksson och Holmgren (2018). Alla organismerna lever i ekosystem vilket är ett avgränsat område som är betraktat som ett system. Ekosystemet består av en biotisk del som innefattar djur, växter och svampar, en abiotisk del som består av mineraler och vatten. För att arbeta med hållbar utveckling är det viktigt att överväga hur materialet utvinns och vilket material som utvinns.

Artefakten som kommer skapas består endast av kod är dess påverkan på miljön och ekosystem är minimal, dock kan arbetet betraktas som en del av industri 4.0 vilket kan ha inverkan på ekologisk hållbarhet. Skorobogatova (2019) nämner att i samband med industri 4.0 har maskiner blivit mer resurseffektiva och kan skapa produkter av högre kvalité. Detta kan medföra att mindre materialutvinning krävs för produktion och produkter inte behöver ersättas i samma takt på grund av att produkten är tåligare.

1.5.2 Ekonomisk hållbarhet

Ekonomisk hållbarhet innefattar flertal områden, Gulliksson och Holmgren (2018) menar att ifrån företagsperspektiv kan ekonomisk hållbarhet innebära strategier för lönsamhet över tid i samverkan med sund naturresurshantering och acceptabelt förhållningssätt till personals rättigheter och löneutveckling. Ekonomisk hållbarhet ur ett globalt perspektiv handlar om hur fattigdom kan utrotas med hjälp av ekonomisk utveckling utan att ge negativa avtryck för den sociala och ekologiska hållbarheten. För att uppskatta möjligheten för ett land att uppnå hållbar utveckling används enligt Persson och Persson (2011) ett socio-ekonomiskt utvecklings index av FN som kallas Human Development Index (HDI). Länder poängsätts från 0–1 utifrån levnadsstandard, utbildning och hälsa. Länder med låg HDI har befolkning drabbad utav fattigdom och människorna har färre möjligheter att kunna leva sina liv och arbeta hur de vill samt att de ekonomiska klyftorna bland befolkningen är större. Land med lågt HDI näringslivsstruktur kännetecknas av att en större andel av befolkningen arbetar inom jordbruket. Länder med högre HDI har en större del av befolkningen arbetar inom IT och service.

Projektet kommer påverka den ekonomiska hållbarheten i den utsträckningen att processen bör spara tid för företaget när kalibrering av TCP utförs. Företaget kan därmed allokera anställda till andra uppgifter vilket också spara pengar för företaget då företaget inte behöver anställa flera människor.

1.5.3 Social hållbarhet

Att arbeta med social hållbarhet innebär enligt Gulliksson och Holmgren (2018) att sträva efter att uppnå och minska samhällets sociala klyftor i form av jämlikhet, jämställdhet och demokrati. Alla människor ska ha lika förutsättningar och rättigheter att utbilda sig och få anställning. Hur korruption ska bekämpas och motverkas inom företag, stat och skolor är även en viktig del för social hållbarhet.

Vid skapandet av artefakten ska social hållbarhet tas i beaktning kring hur artefakten kan komma att påverka människor och samhället. Skapandet av den automatiserade kalibreringsprocess av TCP anses kunna bidra till att bibehålla samt öka användandet av automation. Genom att nyttja automation kan tunga och repetitiva arbetsuppgifter undvikas, då det i stället utförs av de maskiner som tillsätts arbetsuppgifterna. Sådana uppgifter tenderar att orsaka förslitningsskador, vilket gör det fördelaktigt att minska sådant manuellt arbete (Axelsson & Tangen, 2008).

1.6 Rapportens disposition

Rapporten inleddes med ett introduktionskapitel där bakgrund och problemformulering nämndes följt av syfte, mål och projektets avgränsningar. Avslutningsvis beskrivs hur hållbar utveckling berör projektet.

En teoretisk referensram som beskriver relevanta begrepp som används i arbetet som är till för att ge läsaren en förståelse av teorin inom projektet.

Litteraturgenomgången är till för att ge en fördjupad blick inom forskningsområdet och dess begrepp. Avslutningsvis finns en analys av materialet från litteraturgenomgången.

Ett metodkapitel som beskriver forskningsmetodiken i arbetet och hur den kommer tillämpas samt vilka metoder som används.

Genomförandekapitlet beskriver hur arbetet utförts. Kapitlet är presenterat som en linjär process för att förenkla för läsaren men alla de steg som beskrivs har utförts i flytande ordning då vissa steg behöver repeteras utefter behov.

Resultatkapitlet presenterar kortfattat vad som har skapats under arbetet.

Ett diskussions kapitel som tar upp vilka för- och nackdelar som resultatet ger samt diskussion kring hur det kan förbättras.

Avslutningsvis avslutats arbetet med en slutsats om arbetet och vilka framtida arbeten som kan genomföras.

2 Teoretisk referensram

Följande kapitel är till för att underlätta för läsaren. Genom att presentera koncept samt teorier som är relevanta för arbetet bör läsaren få en djupare förståelse för projektet och dess resultat.

2.1 Industri 4.0

Den fjärde industriella revolutionen, även känd som industri 4.0, myntades i Tyskland 2011 (Gunal, 2019). Att skapa ”smarta fabriker” och Cyber-physical systems (CPS) är det slutgiltiga målet med industri 4.0. CPS uttrycker sig i form av fysiska maskiner, produkter som kan kommunicera med varandra trådlöst i fabriken med hjälp av digitala komponenter (Boyes, Hallaq, Cunningham, Watson. 2018). I tidigare revolutioner har mänskligt ingripande för beslutstagande varit ett krav för att anpassa produktion, men med utveckling av artificiell intelligens (AI) kommer maskiner och robotar kunna dra egna slutsatser och agera efter det menar Boyes et al (2018). Som tidigare industriella revolutioner ökas förmågan att kunna producera flera produkter till lägre kostnad och mer flexibel produktion. För att företag ska kunna ta sig in i industri 4.0 krävs flertal nya tekniker bli implementerade som beskrivs nedan.

2.1.1 Simulering

Simulering kan uttrycka sig som en digital kopia av ett verkligt system. Simulering används enligt Robinson (2019) för att kunna arbeta med optimering av produktion eller flöden utan behov av att störa det verkliga systemet, även kunna testa nya produkter eller lösningar i en digital värld innan implementering. Det finns flera olika metoder för simulering, discrete-event simulation (DES) där entiteter strömmar genom ett system. Entiteter kan representera människor, produkter och andra fysiska objekt, därför har DES ett brett användningsområde och kan användas av olika slags organisationer som sjukhus eller restauranger. Robinson (2019) uttrycker att fördelar med att simulera är kostnad som krävs för att avbryta den dagliga verksamheten för att testa nya lösningar, särskilt om experimenten är tidskonsumerande. Nackdelar med simulering är att det krävs en viss expertis för att kunna bygga en realistisk modell, validera modellen och dra korrekta slutsatser från experimenten.

2.1.2 Industrial internet of things

Ett system där maskiner och industriella applikationer är sammankopplade kallas för Industrial internet of things (IIoT). Sensorer och instrument används enligt Boyes, et. al. (2018) för att samla in data om maskinen till molnlager som mjukvara har tillgång till. Därifrån kan mjukvaran hantera informationen med syftet att exempelvis möjliggöra effektivare underhåll på maskiner och förbättra produktiviteten i fabriken.

2.2 Automation

Automatiserad produktion är ett alternativ till manuell produktion. Det definierar Groover (2008) till den teknologi som är dedikerad till applikationen av mekaniska, elektroniska samt

datoriserade system för att styra och kontrollera produktionen. Anledningen till att dessa system kallas för automatiserade kommer från det grekiska ordet *automatos* (Ollson och Rosen, 2005), vilket betyder självverkande. Det är något som kännetecknar automatiserade system då de oftast utför dess dedikerade uppgifter med begränsat mänskligt deltagande. Ett produktionssystem kan vara automatiserade i olika grader. En hög automationsgrad betyder att mänskligt involverande är nästan obefintlig. En låg automationsgrad betyder i stället att en större mängd arbete görs manuellt. Några exempel på arbete som kan göras med automation är skärande bearbetning, målning, montering, materialhantering, kvalitetskontroll och nu kan även hus byggas delvis automatiskt med 3D-skrivning (Sakin och Kiroglu, 2017).

Groover (2008) menar att automation kan delas upp i tre enkla typer: bestämd automation, programmerbar automation och flexibel automation, enligt följande:

Bestämd Automation: Vad ett system med bestämd automation kan göra är begränsat av dess fysiska beståndsdelar. Ett sådant system består oftast av en samling simpla operationer som tillsammans bildar ett komplext system. Det som kännetecknar dessa system är höga initiala kostnader, höga produktionshastigheter samt låg flexibilitet för flera produktvarianter. Därför passar dessa system vid produktion som planerar att producera i höga volymer med ett fåtal produktvarianter. Ett exempel för detta är monteringsmaskiner som är specialtillverkade för en typ av produkt.

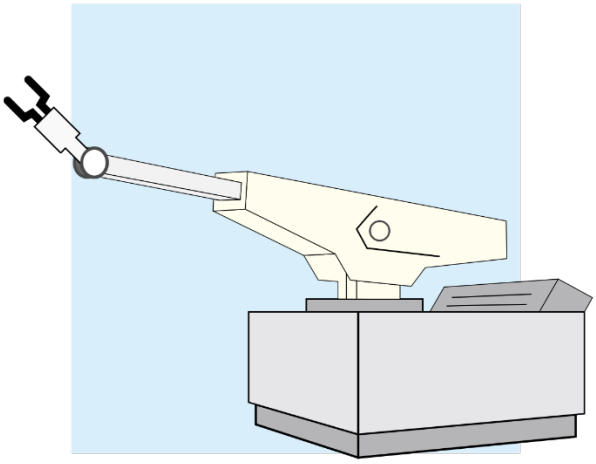
Programmerbar Automation: I ett system med programmerbar automation är det möjligt att påverka sekvensen genom att ändra i det kodade programmet samt den fysiska uppsättningen. Några ändringar som kan göras är: verktygsbyten, fästa fixturer och ändra maskinens inställningar. En sådan omställning kräver tid. Det som främst skiljer denna typ av automation från den bestämda automationen är att den lämpar sig för mindre produktionsvolymer samt större variation på produktvarianter vilket gör den lämpligare för batchproduktion. Några exempel på maskiner som kan tillämpa denna typ av automation är Numerical Control (NC) maskiner, industriella robotar samt Programmable Logic Controller (PLC) styrda system.

Flexibel Automation: Den sista typen av automation är flexibel automation, vilket liknar den programmerbar automation. Det som skiljer är att omställningstiden vid produktbyten är nästan obefintliga, ingen produktionstid förloras. Dessa system är lämpliga att använda vid styckvis produktion av olika produkttyper eller små batchstorlekar. Ett krav på att det ska fungera är att skillnaderna mellan produktvarianterna är minimal, för att undvika tidskrävande omställningar i maskinerna.

Introduktion av automation i ett produktionssystem har höga inledande kostnader. Därför krävs det att automation medför vissa fördelar för att företag ska använda sig av det. Enligt Groover (2008) kan några anledningar till att automatisera vara att öka produktivitet, minska arbetskostnader, mildra effekten av arbetskostnader, minska ledtider samt att vissa uppgifter inte kan utföras manuellt.

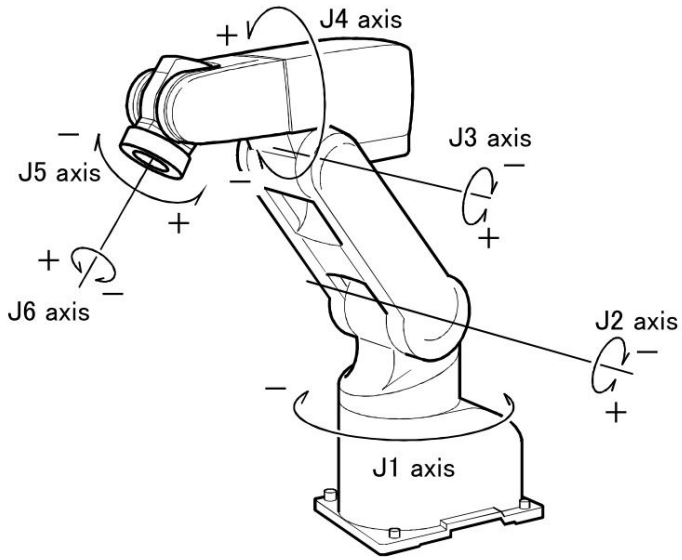
2.3 Industriella robotar

Den första industriella roboten togs i drift 1961 (Jordan, 2016). Roboten som kallades Unimate togs fram av grundarna till dett första företaget med fokus på robotik, George Devol och Joseph Engelberger. Roboten användes för att flytta objekt korta distanser och var utformad som en arm med ett gripverktyg, ses i figur 2.1.



Figur 2.1. *Illustration av roboten Unimate från Unimation. (EBattleP, 2019) CC BY-SA 4.0.*

Definitionen av vad som är en robot är inte klarställd då den accepterade versionen förändras utefter vilket område ”roboten” ska verka i. Men en robot är ofta en automatiskt opererad maskin som utför ett arbete som en människa annars hade genomfört. Ofta kallas den fysiska rörliga delen av en robot för manipulator, men i följande text kommer ordet robot användas synonymt för manipulator. Hur en robot kan se ut finns i flera utföranden. En kategori av robotar är sexaxliga robotarmar. De består av sex leder som roterar runt sex axlar, vilket innebär att den har sex frihetsgrader. Lederna benämns som led 1, led 2… led 6. Led 1 är den närmast basen och led 6 är närmast sluteffektorn eller verktyget som interagerar med omgivningen. Sluteffektorn fästs på det som kallas för verktygsfläns, vilket är änden på led 6, se figur 2.2 (Budin Shah, Osman, Zainuddin, 2020).



Figur 2.2. *Sexaxlig Robotarm (Eshah Budin Shah, et al., 2020).*

Dessa robotar har en sfärisk arbetsvolym runt roboten. Volymen påverkas av geometrin på den externa sluteffektorn som fästs på handleden som är den platta i änden närmast axel 6. Arbetsvolymen är det område roboten han manipulera änden på den sista leden till önskad orientering (Snyder, 1985).

För att en industriell robot ska utföra arbete måste de programmeras, vilket Rembold (1990) nämner flera metoder för. En metod för detta går ut på att leda roboten manuellt till önskade positioner, det kallas att jogga roboten. När roboten har nått önskad position tilldelas denna en rörelseinstruktion som inkluderar olika parametrar för rörelsen till denna positionen. Det är parametrar som bestämmer vilken hastighet roboten skall röra sig till positionen, hur precis roboten är vid rörelsen och om den skall röra sig linjärt till punkten eller inte. Proceduren upprepas tills att alla önskade positioner har tilldelats in rörelseinstruktion. Vid vissa positioner där roboten samt dess sluteffektor ska interagera med omvärlden kan ytterligare instruktioner som styr detta läggas till. Denna typ av programmering måste göras vid den fysiska roboten på plats. Detta kallas för online-programmering. En fördel med att programmera online är att det går att prova olika lösningar direkt när problem uppstår. En nackdel blir i stället att produktionen kan behöva stoppas om roboten står i en aktiv produktion. Alternativet till online-programmering är offline-programmering (ibid.). Då skrivs programmet i stället på någon typ av arbetsstation ej kopplad till roboten i en mjukvara som är kompatibel med robotkontrollern.

Vid skapande av rörelseinstruktioner för en robotarm menar Rembold (1990) att de kan skapas på två olika sätt, beroende på syftet med rörelseinstruktionen. Det första är att bestämma positionen på enstaka leder. Exempelvis kan led 2 köras till två grader medsols från dess nollpunkt. Det andra sättet görs genom att bestämma en position i robotens koordinatsystem som den ska köra sitt TCP till. Därefter räknar robotkontrollern ut positionerna på alla leder som krävs för att nå punkten. TCP är en punkt vilket används som referens för positionen på robotens sluteffektor. Denna punkt är något som kan omdefinieras utefter utformningen på

sluteffektorn för att underlätta programmering av rörelseinstruktioner. Ett TCP bör innehålla data för dess position samt orientering relaterat till robotens verktygsfläns.

2.3.1 ABB specifika data

ABB har skapat ett högnivåprogrammeringsspråk kallat RAPID för att programmera deras robotar. I manualen av ABB Robotics (2010) beskrivs de instruktioner, funktioner och datatyper som används i ABB:s IRC5 system. Tre datatyper som kan användas i robotprogram från ABB är Wobjdata (Wobj), Tooldata och RobTarget (RT).

wobjdata – Datatypen används för att beskriva de arbetsobjekt roboten arbetar mot. Om ett arbetsobjekt defineras i en rörelseinstruktion, så kommer kordinaterna för rörelsemålet att vara definierat inom arbetsobjektet koordinater. I datan för ett arbetsobjekt inkluderas koordinaterna för dess ursprung. Det är definierat som det X- Y- och Z-koordinater som ursprunget för arbetsobjektet befinner sig på utifrån roboten koordinatsystem, som har sin ursprungspunkt i botten av roboten basplatta.

tooldata – Tooldata används för att beskriva egenskaperna för den sluteffektor datan tillhör. Det innehåller koordinater för TCP i förhållande till den verktygsfläns sluteffektorn är fäst på.

robtarget – RT innehåller data för ett rörelsemål, alltså dess koordinater i det arbetsobjekt målet är definierat i samt orientation för de TCP som skall föras dit.

2.4 Visionsystem

Visionsystem används inom industrin för att kunna utöka effektiviteten av produktionen i en fabrik genom att reducera tid av mänskligt inblandande i produktionen, då maskinen kan utföra ett arbete med högre hastighet (Amini och Shanbehzadeh, 2009). Tillämpningsområden för kamera är exempelvis vid kvalitetskontroller där maskinen med hjälp av kamera kan identifiera om det finns avvikelser på produkt och därefter godkänna eller kassera produkten. Ett annat användningsområde är för industrirobotar kameran samspelar med roboten för att hitta positionen för ett objekt inom ett arbetsområde. Med hjälp av en kamerabild kan industrirobot eller maskin tolka och förstå bilden och därefter ta eget beslut om vart roboten ska färdas för att utföra sin uppgift. Teknologier som krävs för att åstadkomma ett fungerande visionsystem med industrirobot innefattar enligt Cheng och Denman (2005) datorsyn, bildbehandling kalibrering av kamera och robot. För att behandla bilder läser programvara för kameran objektets position eller intresseområde i koordinater i form av pixlar i två dimensioner. En tvådimensionell (2D) kamera kan dock inte läsa av djupet av ett tredimensionell (3D) objekt, vilket kan försämra noggrannheten på positionen av ett 3D objekt. Ett monokamerasystem innehåller endast en kamera och kan därför endast fånga bilder i ett 2D perspektiv. Ett stereokamerasystem innehåller i stället två kameror. Detta möjliggör att systemet kan uppfatta djup.

2.5 Kalibrering

Enligt Swedac (2022) kalibreras verktyg genom att jämföra det med en normal. De menar även att en mätosäkerhetssiffra alltid bör medfölja resultatet av en kalibrering för att visa hur precist verktyget är efter senaste kalibreringen. I fallet av en industriell robot behöver TCP kalibreras. Första gången är vid uppstart när det definieras, men det kan även behövas vid flera tillfällen beroende på vad robotens sluteffektor används till samt hur noggrant arbete det är som utförs. Om en sluteffektor utsätts för höga krafter kan enligt Swedac (2022) den deformeras vilket gör att det TCP som är definierat i robotkontrollern inte längre stämmer. När ett TCP kalibreras används den fysiska sluteffektorn som den normal resultatet jämförs med.

2.6 Linjär algebra

Linjär algebra är ett brett ämne med en mängd olika tillämpningsområden. Det kan användas för att underlätta geometriska beräkningar i rummet och planet. Det möjliggörs av beräkningar med vektorer och matriser som kan representera förflyttning och rotation (Lemurell, 2010).

Lemurell (2010) definierar en geometrisk vektor i rummet som ett objekt med egenskaperna längd (magnitud) samt riktning. Det finns ett undantag som kallas nollvektorn som har magnituden noll och saknar riktning. Ett annat verktyg för beräkning med vektorer är enhetsvektorer, vilket är en vektor med magnituden 1. Magnituden av en vektor ett reelt tal som är lika med eller större än noll. En vektor som startar i punkt A och slutar i punkt B betecknas som $\overline{AB}$. Beroende på om vektor är i rummet eller planet kan de representeras med talpar eller taltripplar. En vektor $\overline{AB}$ i rummet med tre koordinataxlar vilka representeras av tre enhetsvektorer e_x, e_y och e_z kan alltså beskrivas:

$$\overline{AB} = xe_x + ye_y + ze_z \quad (2.1)$$

Där xe_x, ye_y och ze_z representerar ortogonala projektioner av $\overline{AB}$ på respektive koordinataxel. Om e_x, e_y och e_z är tre parvis ortogonala enhetsvektorer och x, y och z är reella tal kan x, y och z kallas för vektorns koordinater. Därför kan en vektor skrivas ut som:

$$\overline{AB} = (x \quad x \quad z)^T = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.2)$$

För att beräkna magnituden på en vektor utnyttjas Pythagoras sats för att framställa ekvationen:

$$|\overline{AB}| = \sqrt{x^2 + y^2 + z^2} \quad (2.3)$$

Om det är riktningen av en vektor som är intressant och magnituden kan ignoreras så kan vektorns enhetsvektor beräknas. Det möjliggör och förenklar diverse operationer. Magnituden beräknas med ekvationen:

$$\widehat{\overline{AB}} = \frac{\overline{AB}}{|\overline{AB}|} \quad (2.4)$$

En vektor kan även multipliceras med ett reellt tal för att för att förändra magnituden. För att förlänga en vektor till dess dubbla magnitud multipliceras den med två. Riktningen för en vektor kan förändras till motsatt riktning genom att multiplicera med ett negativt tal. En vektor som multipliceras med ett reellt tal k skrivs ut:

$$\overline{AB} * k = \begin{pmatrix} x * k \\ y * k \\ z * k \end{pmatrix} \qquad (2.5)$$

Lemurell (2010) menar att vektorer i planet och rummet går att förflytta och rotera med hjälp av matriser. En matris skrivs ut som ett rutnät med siffror. En 3x3 matris skrivs ut:

$$\boldsymbol{X} = \begin{pmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{pmatrix} \qquad (2.6)$$

För att rotera en vektor multipliceras en vektor med en rotationsmatris. Positiv rotation, medurs, runt axlarna för x, y och z i vinklarna $\alpha,\ \beta$ och γ om sinus betecknas S och cosinus betecknas C skrivs som:

$$\boldsymbol{R_x}(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C(\alpha) & -S(\alpha) \\ 0 & S(\alpha) & C(\alpha) \end{pmatrix} \qquad (2.7)$$

$$\boldsymbol{R_y}(\beta) = \begin{pmatrix} C(\beta) & 0 & S(\beta) \\ 0 & 1 & 0 \\ -S(\beta) & 0 & C(\beta) \end{pmatrix} \qquad (2.8)$$

$$\boldsymbol{R_z}(\gamma) = \begin{pmatrix} C(\gamma) & -S(\gamma) & 0 \\ S(\gamma) & C(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (2.9)$$

För att skapa en rotationsmatris som inkluderar rotation kring de tre axlarna krävs de att matriserna (2.7), (2.8) och (2.9) multipliceras ihop. Matrismultiplikation är inte kommutativ. Det resulterar i att ordningen som matriser multipliceras på är viktig. Multiplikationen görs från höger och resulterar i följande matris:

$$\boldsymbol{R} = \boldsymbol{R_z}(\gamma) * \boldsymbol{R_y}(\beta) * \boldsymbol{R_x}(\alpha) =$$
$$\begin{pmatrix} C(\gamma) & -S(\gamma) & 0 \\ S(\gamma) & C(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C(\beta) & 0 & S(\beta) \\ 0 & 1 & 0 \\ -S(\beta) & 0 & C(\beta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C(\alpha) & -S(\alpha) \\ 0 & S(\alpha) & C(\alpha) \end{pmatrix} =$$
$$\begin{pmatrix} C(\beta)C(\gamma) & S(\alpha)S(\beta)C(\gamma) - C(\alpha)S(\gamma) & C(\alpha)S(\beta)C(\gamma) + S(\alpha)S(\gamma) \\ C(\beta)S(\gamma) & S(\alpha)S(\beta)S(\gamma) + C(\alpha)C(\gamma) & C(\alpha)S(\beta)S(\gamma) - S(\alpha)C(\gamma) \\ -S(\beta) & S(\alpha)C(\beta) & C(\alpha)C(\beta) \end{pmatrix} \qquad (2.10)$$

En operation som inte är möjlig gällande matriser är division. I stället används matrisens invers som en faktor i en multiplikation. En invers till en matris $\boldsymbol{A}$ är en matris $\boldsymbol{B}$ som resulterar i matrisens identitet $\boldsymbol{I}$ likt följande:

$$\boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I} \qquad (2.11)$$

Om identiteten $\boldsymbol{I}$ till en matris $\boldsymbol{A}$ multipliceras med matrisen $\boldsymbol{A}$, så skall matrisen vara oförändrad likt:

$$\boldsymbol{AI} = \boldsymbol{IA} = \boldsymbol{A} \qquad (2.12)$$

För 2x2 matriser är identiteten:

$$\boldsymbol{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (2.13)$$

För att beräkna inversen $\boldsymbol{A}^{-1}$ till matris $\boldsymbol{A}$ som är en 2x2 matris likt:

$$\boldsymbol{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad (2.14)$$

Genomförs det på följande sätt om och endast om determinanten det($\boldsymbol{A}$) inte är lika med noll:

$$\boldsymbol{A}^{-1} = \frac{1}{\det(\boldsymbol{A})} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \qquad (2.15)$$

Determinant till en 2x2 matris är direkt relaterad till area och riktning av vektorer i en matris. Determinanten till matris $\boldsymbol{A}$ i ekvation (2.14) beräknas som följande:

$$\det(\boldsymbol{A}) = ad - bc \qquad (2.16)$$

Ett koncept som kan användas för att representera orienteringen av koordinatsystem är enligt Hanson (2006) kvaternioner. Kvaternioner är fyrdimensionella vektorer som kan användas för att representera rotationsförhållanden mellan olika koordinatsystem. En kvaternion betecknas:

$$q = (q_0, q_1, q_2, q_3) \qquad (2.17)$$

För att kunna använda en kvaternion i beräkningar i samband med vektorer och matriser behöver de omvandlas till en rotationsmatris. Den allmänna rotationsmatrisen beräknas med följande ekvation:

$$\boldsymbol{R}(q) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \qquad (2.18)$$

Då en kvaternion kan representera en rotation från en position till en annan så kan kvaternionens konjugat användas för att representera rotationen tillbaka till den första positionen. Den betecknas som följande:

$$q^{-1} = (q_0, -q_1, -q_2, -q_3) \qquad (2.19)$$

3 Litteraturgenomgång

För att ta lärdom av liknande projekt inom visionsystem och kalibrering av industrirobotar utförs en litteraturgenomgång. Syftet är att sammanfatta den existerande kunskapen inom området och granska vilka metoder forskare har använt. Hur litteraturgenomgången utfördes beskrivs i kapitel 5.2.

3.1 Robot och vision samarbete

För att arbeta mot industri 4.0 där produktionen är helt automatiserad i den utsträckningen att utrustningen kan fatta egna beslut. Detta kan göras med hjälp av visionsystem som kommunicerar med maskin eller robot (Gunal, 2019).

Författarna Tan, Ayob, Hassrizal, Ismail, Muhamad, Hashim, Othman, Shahriman och Low. (2021) beskriver hur de har löst upprepade problem med otydliga bilder och färger på fotograferade objekt på grund av misslyckad bildbehandling. Författarna hade för avsikt att förbättra ett sorteringsarbete för en kartesisk robot. Författarna skapade ett koordinatsystem för intresseobjektet, vilket inkluderade flera steg. Först måste objektet fotograferas för att sedan bli kalibrerat av mjukvaran LABView, med syfte att motverka fel som uppkom vid olika djup, då kameran som användes endast kunde urskilja två dimensioner. En algoritm vid namn Canny Edge Detection användes för att urskilja konturer av objektet på bilden för att sedan jämföra bilden med en mall av objektet. Om bilden matchade mallen gick programmet till nästa steg där bildens koordinater i pixlar konverterades till robotens världskoordinater. Därefter fanns möjligheten för roboten att plocka och placera objektet. För att utvärdera metoden tog Tan, et. al. (2021) fram en matchningsnoggrannhet som ska vara på åtminstone 80%. Matchnings noggrannheten beror på en matchningspoäng som erhålls vid utvärdering av bilden. Författarna påvisar att användandet av en sämre kamera jämfört med en ideell industriell kamera var det svårt att åstadkomma en nära 100% noggrannhet, därför nöjde de sig med 80% noggrannhet. På vilket sätt matchningsnoggrannheten beror på matchningspoängen förklaras inte av författarna, de nämnde endast att deras slutliga matchningspoäng gav en nära 100% noggrannhet.

Författaren Cheng (2017) beskrev att i en industri med flexibel automation blir arbete med att programmera in nya mål för robotar vid implementering av nya produktvarianter vara en tidskrävande process. Med hjälp av en 2D kamera fäst på en robotarm kan position på ett intresseobjekt erhållas. För att kalibrera kameran använde sig författaren av ett rutnätsblad bestående av små och stora cirklar som vision mjukvaran använde för att få position på rutnätsbladet och dess pixelvärden i x- och y-led. Med en funktion från mjukvaran FANUC VisLOC kunde specifika geometriska former identifieras på en 2D bild byggd på två lokaliserare, i detta fall sökte funktionen efter det cirkulära huvudet av en golfpeg. Den första lokaliseraren kunde hitta flertal golf peggar i ett område och den andre lokaliseraren var till för att identifiera en specifik golfpeg. När lokaliserare 2 upptäckte en specifik golfpeg kunde lokaliserare 1 mäta dess djup genom att ta bilder från flera perspektiv. När programmet kördes hämtades data av en fördefinierad TCP och bilder togs på rutnätsblad för kalibrering. Kameran

förflyttades till olika positioner på arbetsytan och lokaliseraren identifierade golfpeggarna, sedan kunde rörelseinstruktioner köras. Metoden som används i deras projekt kräver ett visionsystem som har alla nämnda funktioner för att lösningen ska fungera.

Schmidt och Wang (2014) presenterade en lösning till kalibrering av arbetsobjekt på en slumpmässig plats och orientering med hjälp av kameror som identifierar och hämtar data från referensmarkörer. Referensmarkörerna användes för att identifiera markerings ID, dimensioner och transformationsmatris med dess position med avseende på arbetsobjektet. Arbetet är lämpat i sammanhang med montering och svetsning. Kamera för ett globalt perspektiv över arbetsytan användes för att söka efter flertalet referensmarkörer, desto fler referensmarkörer kameran identifierade vid kalibrering av arbetsobjektet gjorde kalibreringen noggrannare. En kamera som var närmare arbetsytan för att ge ett lokalt perspektiv som tillåter en noggrannare lokalisering av arbetsobjektet som dessutom var monterad på robotarmen. Schmidt och Wang (2014) nämner att referensmarkörerna är från ArUco bibliotek som är AR verktyg. Referensmarkörerna bestod av kvadrater i ett rutnät av storlek 5x5 som ger logikvärde ”1” eller ”0”. För att utvärdera lösningen testade författarna med en kalibrerad kamera för att ta 30 ögonblicksbilder på referensmarkörerna från olika vinklar för att få noggrannheten. Resultatet tydde på att lösningen lämpas bäst för arbeten som tillåter en noggrannhet på 1 mm.

3.2 Definiering av TCP med vision

Trots att visionsystem används för att skapa ett självständigt automatiserat system finns det fortfarande vissa processer som brukar göras manuellt. En sådan process är definiering av TCP.

Hallenberg (2007) skapade en metod som med hjälp av datorseende kalibrerar TCP för en robots sluteffektor. Målet var att skapa en metod som var minst lika noggrann som den tidigare manuella metoden, vilket var ±1 mm. Författaren var medveten om att det redan fanns metoder för att automatisera denna process. Dessa krävde specialiserad utrustning och fungerade endast för specifika verktyg. Författaren avsåg att eliminera den begränsningen med skapandet av en ny metod. Med den nya metoden utförs kalibrering av TCP med en kamera som är kopplad till en dator med USB-kabel och datorn är även kopplad till roboten med en TCP-IP-kabel. Roboten hade sex frihetsgrader. För att bestämma TCP utgick Hallenberg (2007) från fyra olika transformationer: X_1, X_2, T_1 och T_2. X_1 är transformationen från sluteffektorns TCP till TCP för verktygsflänsen på roboten, därför kan TCP bestämmas om X_1 är känd. T_1 är transformationen från robotens bas till TCP för verktygsflänsen, vilken är känd från start. T_2 är transformationen från kamera till sluteffektorns TCP och X_2 är från kamera till robotens bas, dessa två var okända. För att kunna bestämma X_1ställde författaren upp en ekvation: $T_2X_1 = X_2T_1$. Vid bestämning av TCP användes fyra positioner som kameran tog bild på. Mest korrekta resultat uppnåddes då alla sex frihetsgrader nyttjades vid förflyttning av verktyget. För att uppnå önskad noggrannhet använde sig författaren av en iterativ metod vilket jämförde den beräknade TCP med verktygets verkliga TCP i flera positioner. Det eliminerade påverkan av felaktig kalibrering av roboten samt kameran. För att utvärdera noggrannheten av metoden behövdes först kamerans noggrannhet beräknas och därefter skillnaden mellan robotens rörelse mot den rörelse som uppfattades av kameran. Därefter kunde noggrannhet samt

upprepningsnoggrannheten på metoden testas. Den begränsande faktorn blev kamerans noggrannhet på ±0,03 mm vilket i metoden resulterade i en noggrannhet på ±0,1 mm.

Yang, Gong och Liu (2021) utvecklade en universiell visuell metod för kalibrering av TCP vilket först visuellt kalibrerar ett målobjekt som roboten ska verka mot, för att därefter kalibrera TCP. Detta eliminerar behovet av kalibrering av TCP som en egen process och minskar den totala tiden som läggs på kalibrering av visionsystem. Det görs genom att använda inneboende begränsningar i de målpunkter som används vid kalibreringen av målobjektet. De använde sig av referensmarkörer för att identifiera positioner på objekt samt robot. Vid kalibrering av målobjektet körs robotens verktyg manuellt till olika målpunkter för att sedan beräkna punkternas positioner relaterat till tillhörande referensmarkör. Utefter dessa positioner samt beräkningar kan TCP beräknas. Sedan används liealgebra och gauss-newton metoden för att minska reprojektionsfelet. Metoden prövades i två olika experiment där den jämfördes med den konventionella metoden att kalibrera systemet för hand. De testade noggranheten, men även hur tidseffektiv den var. Tidseffektiviteten testades genom att först dela upp båda metoder i mindre steg för att sedan mäta tiden det tog att utföra samtliga steg. Den föreslagna metoden bevisades vara mer noggrann samt 37,5% mer tidseffektiv än den konventionella metoden.

3.3 Analys av litteraturgenomgång

Utifrån litteraturen har några allmänna problem som uppstår vid samarbete mellan robot och visionsystem identifierats. För att säkerställa en effektiv produktion bör dessa beaktas och hanteras. Det första problemet Tan, et. al. (2021) och Cheng (2017) tog upp var att någon typ av mjukvara för igenkänning av intresseobjekt måste användas. Utan mjukvaran utvinns ingen data som är användbar för robotarbete. Ett tillvägagångssätt för att utföra detta är att med hjälp av flertalet bilder på intresseobjektet träna in mjukvaran för att känna igen de särdrag som kännetecknar objektet.

Nästa problem som identifierades är att koordinatsystem skiljer sig mellan en robot och ett 2D kamerasystem (ibid.). Koordinatsystemen i dessa visionsystem består av pixlar i två dimensioner. Robotens koordinatsystem motsvarar i stället millimeter i tre dimensioner. En förutsättning för att robot och visionsystem ska kunna samarbeta krävs en samkalibrering av de två koordinatsystem där visionsystemets pixlar omvandlas till millimeter i robotens koordinater. Schmidt och Wang (2014) använder i stället referensmarkörer för att få positionen av arbetsobjektet relativt till roboten.

Författarna Hallenberg (2007) och Tan, et. al. (2021) presenterar olika metoder för att hantera förvrängningar av bild som beror på kameralinsens egenskaper och vinkeln mellan kameras bildplan, lins och objektet som fotograferas.

Utifrån litteraturgenomgång är det klart att det finns en mängd olika studier och metoder för att kalibrera TCP via visionsystem. Det som särskiljer dessa artiklar med förutsättningarna som finns för detta projekt är att de kalibrerar TCP från grunden, medan i detta projekt kommer ett ungefärligt TCP redan att finnas för att sedan finjusteras till korrekt position. Yang, Gong och Liu (2021) använder sig av kalibrering av målobjektet för att kalibrera TCP. Det är inte möjligt

i projektet då kalibreringen av kalibreringsplattans TCP ska ske innan samkalibreringen. Istället är det deras metod för att utvärdera kalibreringsmetoden som kan vara relevant. Även deras metod för jämförelse av tidseffektivitet är relevant. Detsamma gäller metoden Hallenberg (2007) använder för utvärdering av metoden. Utifrån den metoden kommer noggrannheten av kalibrering att beräknas.

4 Metod

I detta kapitel kommer metoderna som används i projektet beskrivas om hur de kommer användas och varför metoderna är lämpade utifrån syftet och målet.

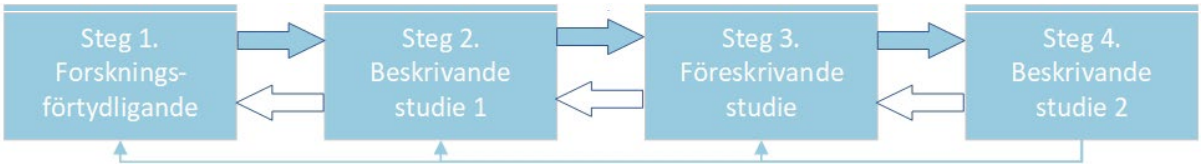
4.1 Forskningsmetodik

Forskningsmetodik är ett tillvägagångssätt för att lösa ett problem menar Säfsten och Gustavsson (2019). Att genomföra vetenskapliga undersökningar bidrar till att utöka kunskapen inom ett särskilt kunskapsområde. Data kan kategoriseras till kvantitativa och kvalitativa data. Kvantitativa data kopplas till numeriska data som mängd och kan tilldelas numeriska värden medan kvalitativa data kan beskrivas som egenskap och beskaffenhet och kan ej tilldelas ett numeriskt värde. Detsamma gäller även för kvantitativa och kvalitativa forskningsmetoder som behandlar hur och vilken data som behövs hämtas under ett projekt (ibid.). Nedan förklaras design research methodology (DRM) som är detta projekts huvudsakliga strategi.

4.1.1 Design research methodology (DRM)

Säfsten och Gustavsson (2019) skriver om DRM som de på svenska kallar produktutvecklingsforskning, då de använder definitionen för design som de aktiviteter som ”genererar och utvecklar en produkt, baserat på ett behov, en produktidé eller teknologi, till den nivå som behövs för att kunna realisera produkten och tillfredställa de upplevda behoven hos användare och andra intressenter” (Säfsten och Gustavsson, 2019, s. 112). För en forskare är syftet med att bedriva ett arbete med DRM att skapa en artefakt som är värdefull för en specifik målgrupp. Därefter skall artefaktens värde för målgruppen vetenskapligt bedömas. Designforskning används för att bygga kunskap om utvecklingsprocessen och förbättra produkt- och processutvecklingsprocessen (Säfsten & Gustavsson, 2019).

DRM omfattar hela forskningsprocessen som tillhör design research. Det kan beskrivas som ett upplägg som innefattar olika metoder och tekniker för datasamling i forskningssammanhang. Enligt Blessing och Chakrabarti (2009) består DRM av fyra steg som görs itterativt och i vilken ordning de görs kan variera, se figur 4.1. Samtliga steg behöver inte alltid utföras. Vilka steg som utförs och i vilken utsträckning beror på faktorer som syftet med undersökningen, tillgänglig tid samt vilken förkunskap forskargruppen besitter.



Figur 4.1. *De fyra stegen i DRM.*

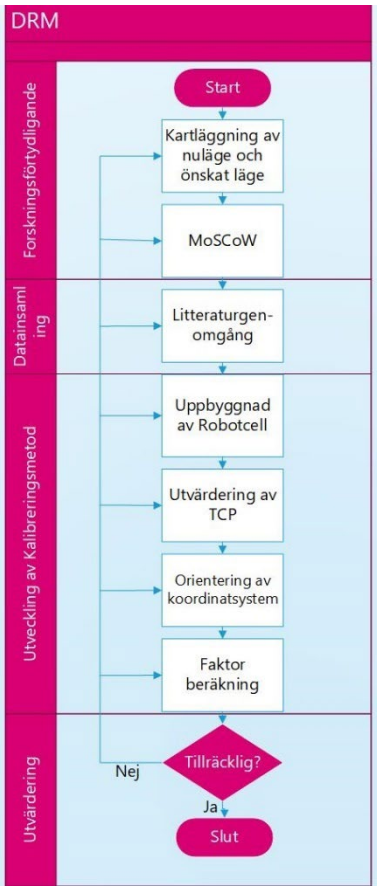
Steg 1. Forskningsförtydligande: Utifrån de antaganden som är anledningen till att undersökningen görs, kan dess syfte och mål bestämmas. En nulägesbeskrivning skapas samt en beskrivning för det önskade framtida läget. Kriterier för framgång ska definieras så att arbetet senare kan utvärderas för att se om undersökningen var framgångsrik. Dessa bör vara formulerade på ett vis att de är möjliga att uppfylla inom undersökningens tidsplan.

Steg 2. Beskrivande studie 1: Nästa steg är att forskargruppen fördjupar sig inom ämnet genom att läsa relevant litteratur samt komplettera eventuella kunskapsbrister med empiriska studier. Detta görs för att skapa ett förstående för vilka forskningsbehov som finns. Resultatet av studien bör vara en komplett nulägesbild samt tydligare mål för arbetet.

Steg 3. Föreskrivande studie: I det tredje steget används den nya kunskapen som införskaffades i föregående steg för att utveckla eller korrigera det framtida läget ytterligare än vad som gjordes i första steget. Denna studie ska innehålla beskrivningar kring förändringar om nuläget för att lättare kunna nå det önskade läget med hjälp av en eller flera idéer som tagits fram. Detta är till stöd för att påbörja arbete på det önskade läget och kan bestå av empiriska data och från erfarenhet av processen. Resultatet av den föreskrivande studien ger stöd som är redo för utvärderingen.

Steg 4. Beskrivande studie 2: Det sista steget utförs för att kontrollera hur arbetet har uppnått de utsatta målen. Det resulterar i kunskap om arbetets användbarhet.

DRM måste anpassas efter projektets behov innan det kan nyttjas för arbetet. Första steget **forskningsförtydligande** kommer i projektet innehålla kartläggning av nuläge samt det önskade läge som identifieras och formuleras i samband med möten. För att identifiera framgångskriterier för det önskade läget används MoSCoW-metoden. Nästa steg, beskrivande studie 1 kommer i projektet att kallas för **datainsamling**. Steget kommer att innehålla insamling av data genom en litteraturgenomgång där relevanta artiklar kommer att undersökas. Informationen som insamlas kommer att användas för att skapa en tydligare bild över forskningens nuläge samt en tydligare bild över projektets önskade läge. Därefter kan en plan för nästa steg formuleras. Utefter planen utförs **utveckling av kalibreringsmetod** och informationen som tillhandhölls i föregående steg, samt empiriska data som framkommer kontinuerligt används för att utveckla en metod för kalibrering av TCP. Utvecklingen av metoden kommer att ske i flera olika steg. Först behöver den virtuella robotcellen byggas upp. Därefter behöver det nuvarande nominella TCP utvärderas. Sedan måste kamerans orientering relaterat till robotens koordinataxlar identifieras och faktorer som noggrant beskriver förhållandet mellan kamerans pixlar och robotens koordinater skall beräknas. Till sist kan TCP förflyttas utifrån de utförda beräkningarna. Därefter kommer TCP att utvärderas ännu en gång och hela processen skall utföras iterativt till önskat resultat är uppnått. Nästa steg i forskningsprocessen är **utvärdering** där arbetets resultat kommer att utvärderas. Frågor som kommer besvaras är: Vilka framgångskriterier är uppfyllda? Hur tillämpbar är metoden? Hur har arbetet bidragit till ny kunskap? För att verifiera att den nya automatiserade metoden är användbar skall noggrannheten som uppnås jämföras med noggrannheten för dagens manuella kalibreringsmetod. Visualisering av hur DRM:s steg anpassas till projektet kan ses på figur 4.2.



Figur 4.2. *Anpassade DRM steg.*

4.2 MoSCoW

MoSCoW metoden är ett verktyg för att kunna säkerställa att rätta prioriteringar görs i ett projekt (Tonnquist, 2021). Metoden är vanlig inom mjukvaruutveckling för att visualisera vilka krav ska prioriteras i rätt ordning menar Projektledning (2021). Metoden baseras på tre nivåer av prioritering i projektet och en uteslutande nivå. De fyra nivåerna är baserat på ordet MoSCoW som är en akronym och står för:

- **M**ust have. Krav som är obligatoriska, annars kan ej resultatet levereras.
- **S**hould have. Dessa krav bör uppfyllas om möjligt.
- **C**ould have. Ej nödvändiga krav men de ökar värdet av resultatet.
- **W**on't have. Krav som inte är nödvändiga för tillfället.

För att få fram kraven kommer möten och workshops utföras tillsammans med företaget, därefter kan kraven och önskemålen rangordnas utefter metoden.

4.3 Litteraturgenomgång

För att kunna tillföra kunskap inom ett existerande område är det viktigt att veta vilken kunskap som redan existerar. Därför kan det vara viktigt att införskaffa den informationen vid starten av en ny undersökning eller projekt, den kan även utvecklas under projektets gång. För att

införskaffa den nödvändiga informationen kan en litteraturgenomgång utföras. Enligt Säften och Gustavsson (2019) innebär det att relevant litteratur som exempelvis studier identifieras, sammanställs och analyseras. En litteraturgenomgång kan ha flera syften beroende på när den utförs. Tidigt i arbetet kan det handla om att skaffa sig kunskap om nuläget för forskningen i området vilket kräver en översiktlig genomgång av litteraturen. Senare i arbetet kan en djupare kunskap inom ämnet vara nödvändig, vilket då leder till att en mer detaljerad genomgång kan krävas. Innan en litteraturgenomgång genomförs bör en plan formuleras, vilket bör innehålla de 10 steg som beskrivs nedan (ibid.).

Steg 1. Precisera syfte med litteraturgenomgången: Inför litteraturgenomgången ska syftet med den formuleras. Det är grunden för att planera samt genomföra litteraturgenomgången.

Steg 2. Identifiera lämpliga sökord: Sökorden bör utformas utifrån de syfte som formulerades i föregående steg.

Steg 3. Formulera kriterier för inkludering och exkludering av litteratur: Exempel på kriterier kan vara vilka typer av publikationer som ska vara med samt under vilket tidsspann de publicerades.

Steg 4. Välja sökverktyg och databas: Det finns flera verktyg och databaser som är lämpliga för användning. Några exempel är IEEE Xplore, Scopus och Google Scholar.

Steg 5. Formulera sökstrategi och genomför sökningar: Två sökstrategier som är lämpliga är databassökningar samt snöbollstekniken. Den första innebär att de identifierade nyckelorden används för att formulera söksträngar för att söka igenom olika databaser. Snöbollstekniken innebär att identifiera relevant litteratur för att sedan spåra deras referenser.

Steg 6. Översiktlig genomgång av identifierad litteratur: Först utvärderas resultaten av sökningar för att identifiera om de är relevanta till syftet. Annars kan det finnas behov att justera nyckelord samt sökstrategier. Vid godtagbara resultat ska litteratur inkluderas och exkluderas utifrån de formulerade kriterierna. För att vara tidseffektiv är det lämplig att först läsa titlar, för att sedan gå vidare till nyckelord samt sammanfattning.

Steg 7. Fördjupad genomgång av relevant litteratur: Nu kan resterande innehåll i de inkluderade verken läsas. I detta steg kan litteratur exkluderas beroende på innehåll och kvalitet.

Steg 8. Extrahera data och skapa översikter: Ur de kvarvarande verken ska nu relevant data extraheras.

Steg 9. Analysera innehåll: Vid mer omfattande litteraturgenomgång kan både en beskrivande- och en innehållsanalys genomföras. Den beskrivande analysen kan innehålla uppgifter om vilka tidskrifter artiklar har publicerades i, forskningsmetoder som har använts samt publicerings år. Innehållsanalysen ska innehålla en genomgång av innehållet som analyseras utifrån litteraturgenomgångens syfte.

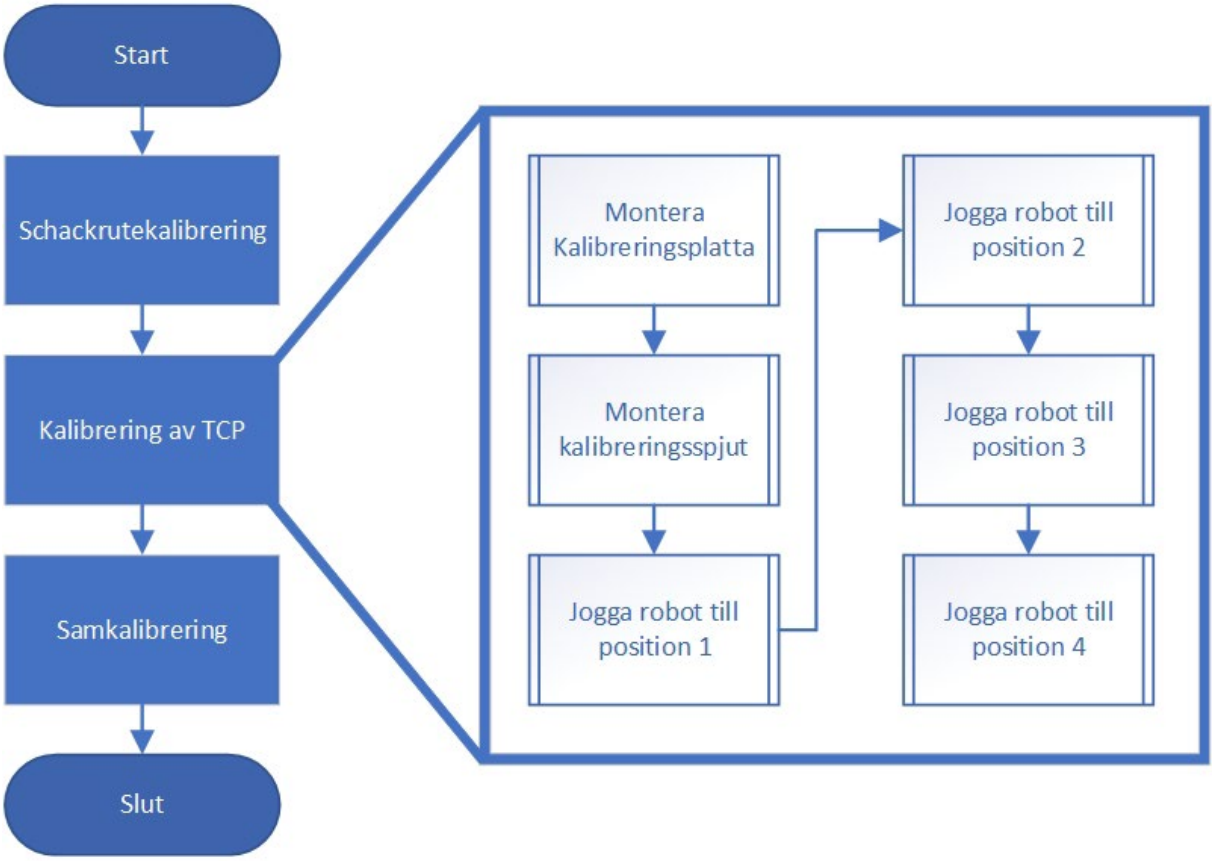
Steg 10. Presentera resultat: Resultatet som presenteras ska vara innehållsfokuserad, den ska inte bara återge vad författarna har skrivit. En kritisk granskning av innehållet är nödvändig.

5 Genomförande

I detta kapitel beskrivs projektets tillvägagångssätt linjärt trots att det i verkligheten är strukturerat enligt de arbetssättet som beskrivs i kapitel 4.1.1. om DRM. Tidsplanen för projektets genomförande ses i Bilaga A.

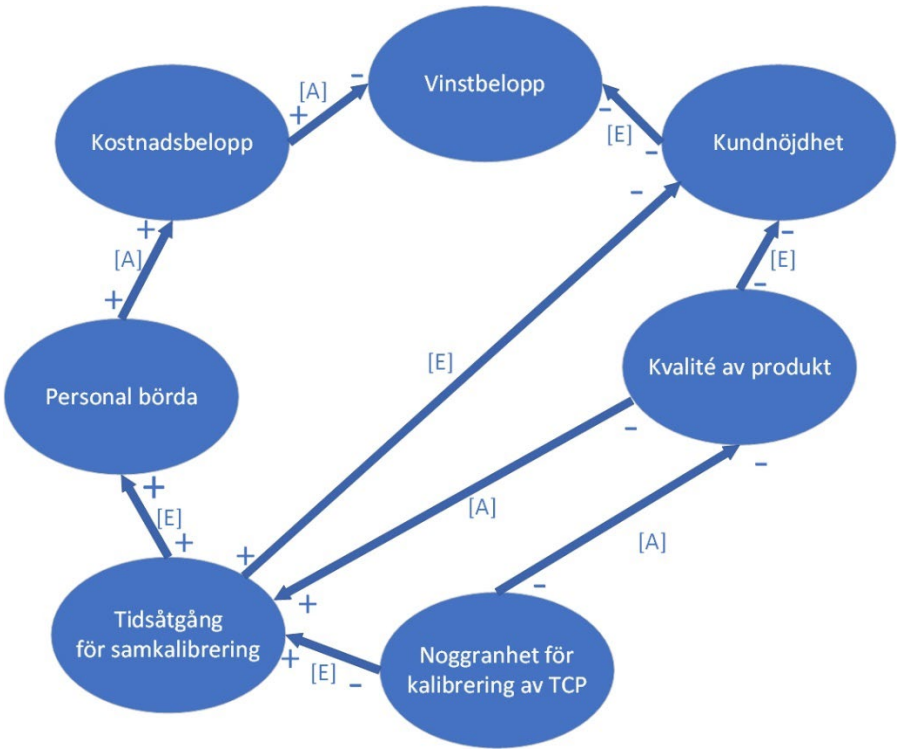
5.1 Forskningsförtydligande

Genom flertalet möten kunde en kartläggning av den nuvarande manuella processen skapas. När en kamera tar bilder sker en viss förvrängning beroende på olika faktorer. Därför inleds kalibreringsprocessen med en schackrutekalibrering. Där ett svartvitt schackbräde används som referens för att med mjukvara motverka förvrängningarna. Därefter kan kalibrering av TCP påbörjas. För att kalibrera TCP i dagsläget används en manuell metod. Roboten joggas med kalibreringsplattan fäst på lämplig plats nära eller på verktyget, till en position där mittpunkten på kalibreringsplattan förts till änden av ett spetsigt kalibreringsspjut. Positionen sparas för att därefter flytta verktyget till en ny position där kalibreringsplattan igen nuddar toppen på spjutet från en annan vinkel. Proceduren upprepas fyra gånger och därefter beräknas positionen på plattans TCP automatiskt utifrån de fyra punkterna. Därefter avlutas processen med samkalibrering av robot och visionsystem. En processkarta över nuläget kan ses i figur 5.1.



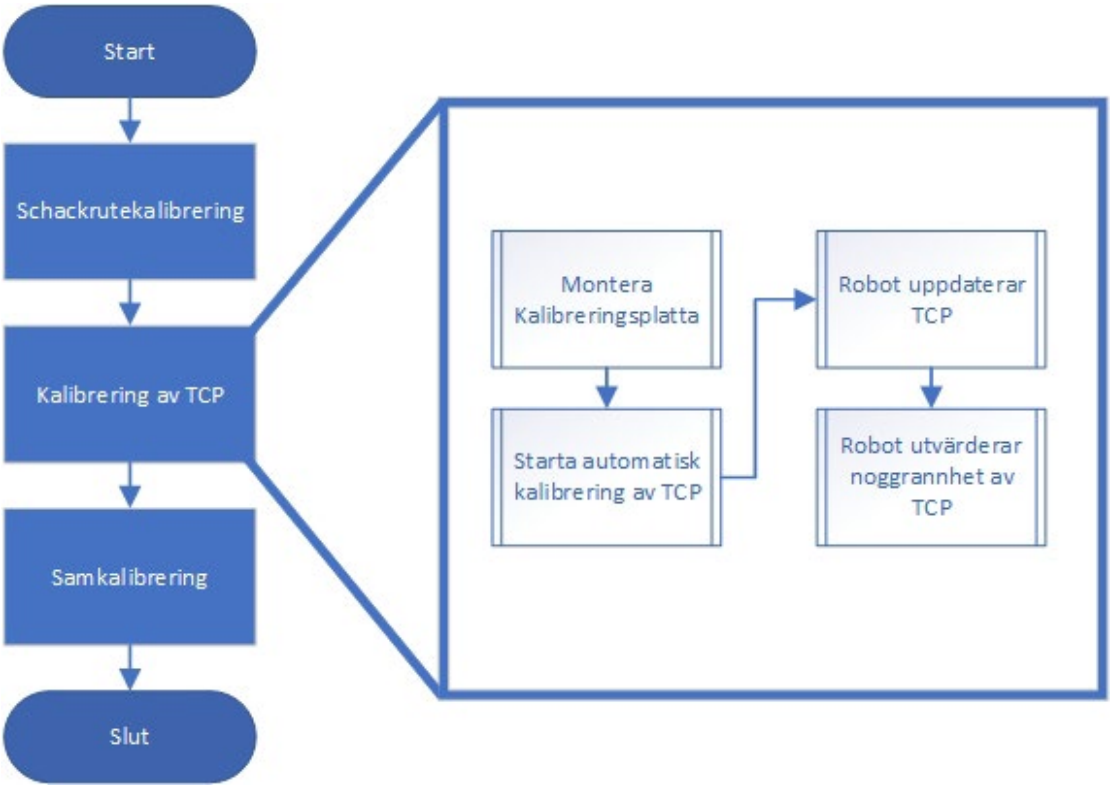
Figur 5.1. *Processkarta över nuläge*

Fördelen med den manuella metoden är att den är universell för sexaxliga robotar, därför kan personer med grundläggande robotkunskap kalibrera TCP. Nackdelen med denna metod är processen kan vara tidskrävande, särskilt om kalibrering behöver upprepas. Noggrannheten av TCP kan även variera beroende på hur noggrann operatören är. En person utan grundläggande kunskap kan inte kalibrera TCP med den manuella metoden. Vid de tillfällen det inte finns personal med tillräcklig kunskap tillgänglig kan därför ingen kalibrering utföras förrän personal med rätt kompetens har tillkallats. För att få djupare förståelse av nuläget skapades en modell som jämför vilken inverkan olika faktorer har på varandra vilket kan ses på figur 5.2. Om länken mellan påståendena innehåller ett [A] innebär det ett antagande och om [E] mellan länkarna är det grundat på erfarenhet från intressenter. I detta fall är intressenterna personer på företaget som projektet görs i samarbete med. Modellen ska tolkas att en låg (-) noggrannhet vid kalibrering av TCP innebär en hög (+) tidsåtgång för samkalibrering. Hög tidsåtgång för samkalibrering leder till hög personalbörda.



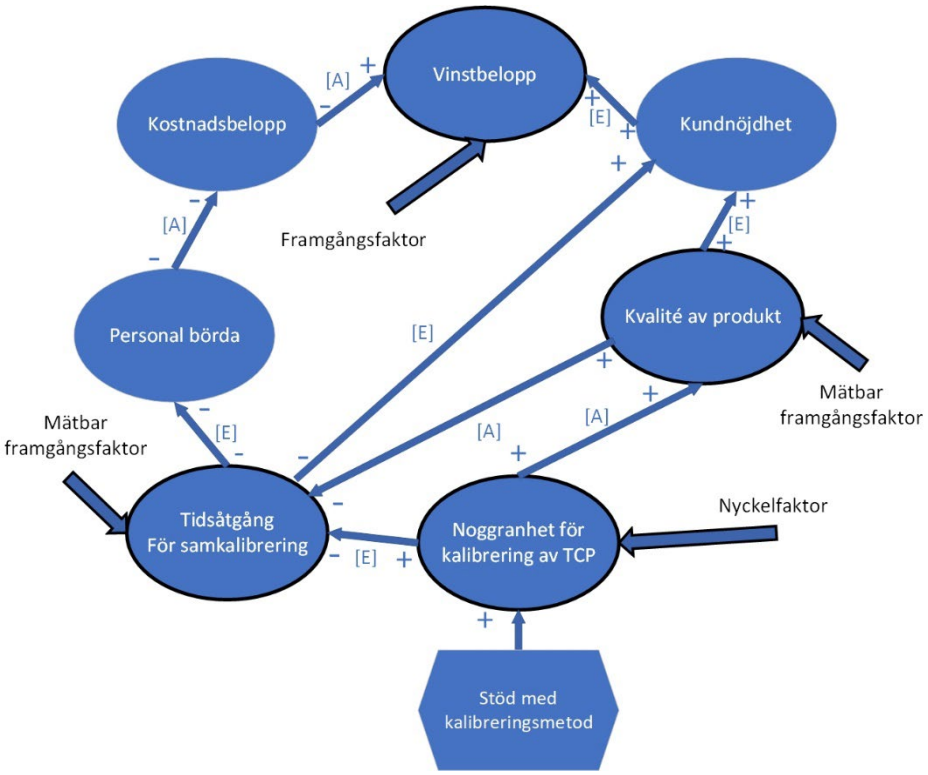
Figur 5.2. *Relationsdiagram för nuläget*

Med kunskap över dagsläget identifierades behovet av att utföra kalibreringen av TCP automatiskt som ger god noggrannhet. Därefter kunde projektets översiktliga syfte och mål formuleras. Flera olika krav och önskemål identifierades i samband med möten med handledare på företaget som möjliggjorde skapande av det önskade läget som kan ses i figur 5.3.



Figur 5.3. *Processkarta över önskat läge*

En modell med stöd av kalibreringsmetod har skapats över önskat läge och vilken påverkan den förväntas medföra illustreras i figur 5.4. Modellen innehåller nyckelfaktorer, framgångsfaktor och mätbar framgångsfaktor. Nyckelfaktor innebär att den faktorn medför störst påverkan över nuläget och påverkas direkt med stöd av kalibreringsmetod. Framgångsfaktor är det slutgiltiga målet för uppdragsgivaren som forskningen ska bidra med och är ofta placerade överst i nätverket i modeller, se vinstbelopp i figur 5.4. Eftersom framgångsfaktorn ofta är svår att mäta under ett projekt, exempelvis för att det är omöjligt att utföra inom tidsramen för projektet, bestäms en mätbar framgångsfaktor för att möjliggöra mätning över hur stor påverkan stödet ger. En mätbar framgångsfaktor skall bedömas vara möjlig att utvärderas inom projektets tidsram. Noggrannhet för kalibrering av TCP anses vara nyckelfaktorn. I figur 5.4 ses stödet direkt under nyckelfaktorn som det är kopplat till, i projektet är stödet definierat som den artefakt som nyttjar kalibreringsmetoden. Vinstbeloppet är framgångsfaktorn som är omöjlig att mäta inom projektets tidsram. För att kunna mäta den används tidsåtgång vid samkalibrering som mätbar framgångsfaktor och kvalité av produkten. Kvalitén i detta fall syftar till kvalitén på samkalibreringen, som påverkar hur väl och repeterbart roboten som har samkalibrerats utför arbetet som den tilldelats. En kalibrering som resulterar i att en robot misslyckas med att plocka detaljer flera gånger för att plockpunkten är dålig, anses vara av låg kvalitét. Tidsåtgång för samkalibrering syftar till all tid som går åt för att få en fullständig samkalibrering. Det innebär att även tid för schackrutekalibrering, kalibrering av TCP, samkalibreringen och den tid som går åt för att göra om stegen i processen.



Figur 5.4. *Relationsdiagram av önskat läge med stöd*

Med fastställda krav och önskemål behövdes en metod för att kunna rangordna dem för att säkerställa att den viktigaste funktionaliteten prioriterades. MoSCoW-metoden användes för att visualisera rangordningen av kraven som fastställts. Dessa krav och önskemål finns i tabell 5.1.

Tabell 5.1. *MoSCoW-tabell*

Must have	• Kalibrering X- och Y-koordinaterna av TCP • Artefakten skall vara kompatibel med ABB:s programmeringsspråk RAPID • Utvärdera TCP efter kalibrering • Metoden skall hantera att kamerans koordinataxlar har annan orientering runt Z axeln än robotens koordinatsystem
Should have	• Anpassningsbara rörelseinstruktioner i roboten för att öka tillämpbarheten där utrymme för rörelser är begränsat
Could have	• Kalibrering av Z-koordinaterna av TCP • Hantering av att kamerans och robotens koordinataxlar för Z ej är parallella
Won't have	• Kalibrering utan existerande TCP • Produktionsklar artefakt

5.2 Datainsamling

Datainsamlingssteget genomförs för att få en djupare förståelse inom ämnet och identifiera kunskapsluckor som arbetet kan fylla. För att göra detta utfördes en litteraturgenomgång som baseras på Säfstens och Gustavssons (2019) metod. Syftet med litteraturgenomgång var att identifiera de arbeten som gjorts inom robot och vision samarbete samt kalibrering av TCP via visionsystem. Sökningar gjordes i databaserna IEEE Xplore, DiVa samt i Högskolan i Skövdes verktyg som används för samsökning i olika databaser vid namnet LibSearch. Fraser och termer som har använts i olika kombinationer vid sökning är ”vision system”, TCP OR ”Tool Center Point”, ”Industrial robot” och ”calibration”. Publikationer som används ska vara: forskningsartiklar, konferenspublikationer eller examensarbete, som är publicerat senaste 15 åren. Utöver databassökningar användes även snöbollstekniken för att finna litteratur. För att avgöra hur relevanta publikationerna var lästes titeln, nyckelord samt sammanfattningen av dem för att sedan bestämma om de skulle inkluderas eller ej. Den litteratur som ansågs irrelevant exkluderades ur studien. Därefter granskades den kvarvarande litteraturen i sin helhet. I slutändan inkluderades fem verk i studien. Innehållet i de litterära verken analyserades för att identifiera metoder och arbetssätt som kan vara till nytta i projektet.

Från empiriska data samt den genomgångna litteraturen är det klart att Kalibrering av TCP kräver data från flertalet bilder där kalibreringsplattan har förflyttats till olika positioner. För att möjliggöra korrigering av det nominella TCP, behövs inledningsvis att relationen mellan axlarna i robotens och kamerans koordinatsystem identifieras. Utan det går det inte att veta hur en förflyttning i robotens koordinater skall ske utifrån bilder i kameran. Därefter ska det utredas hur den nominella TCP:n står i relation till den verkliga mittpunkten på kalibreringsplattan för att sedan korrigera TCP. Utifrån litteraturen konstateras det att en iterativ kalibreringsprocess ger en högre noggrannhet. Därför bör resultatet av utvärderingen och korrigeringen av TCP baseras på om önskad noggrannhet uppnås

Att avancera från nuläge till önskat läge kräver uppbyggnad av virtuell robotcell samt programmering av robot för att sköta kommunikation mellan robot och vision samt styra robot. Lämpliga beräkningar för samspel mellan robot och vision krävs.

5.3 Utveckling av kalibreringsmetod

Utifrån den informationen som har samlats i tidigare steg vidareutvecklas det önskade läget. Då den samlade informationen inte är tillräcklig för att utforma artefakten utförs empiriska studier i form av experiment samt möten och diskussioner.

5.3.1 Uppbyggnad av robotcell

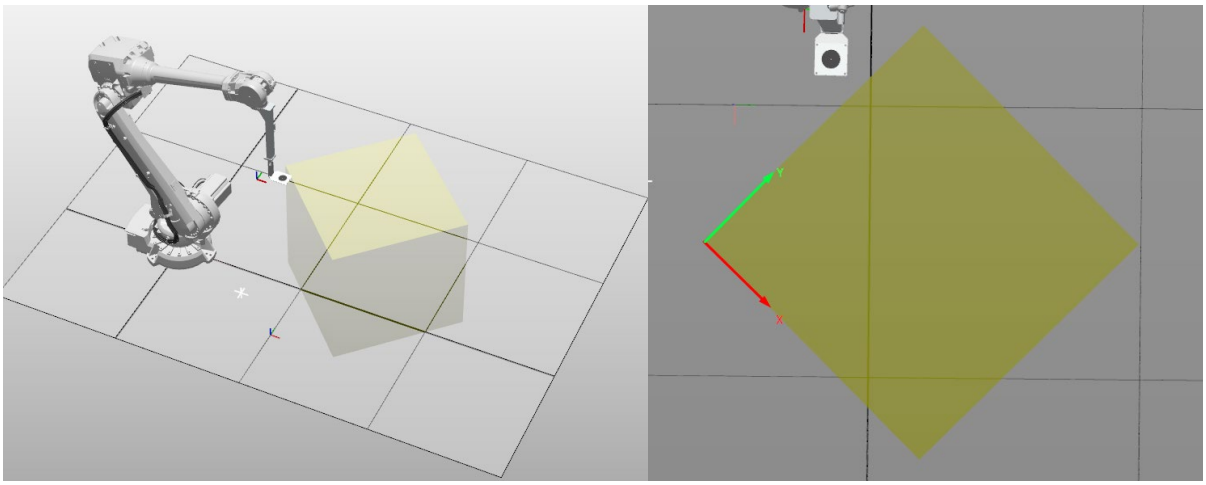
För att kunna utveckla och utvärdera metoden krävs en robot, kalibreringsplatta samt ett visionsystem. Det ansågs lämpligt att börja utvecklingen av kalibreringsmetoden i en simulerad miljö för att bevisa att metoden som skapas fungerar matematiskt. Mjukvaran som ansågs lämplig för att utveckla metoden var RS eftersom den innehåller verktyg för att simulera en verklig robotcell som kan programmeras och valideras. En sexaxlig robot valdes för

utvecklingen av rutinen då den har sex frihetsgrader, vilket innebär att det är möjligt att köra den i flera positioner och orienteringar än en robot med färre frihetsgrader. En 3D modell tillhandahölls som innehåller kalibreringsplatta och medföljande fäste som fästs på robotens 3D modell, se figur 5.5.

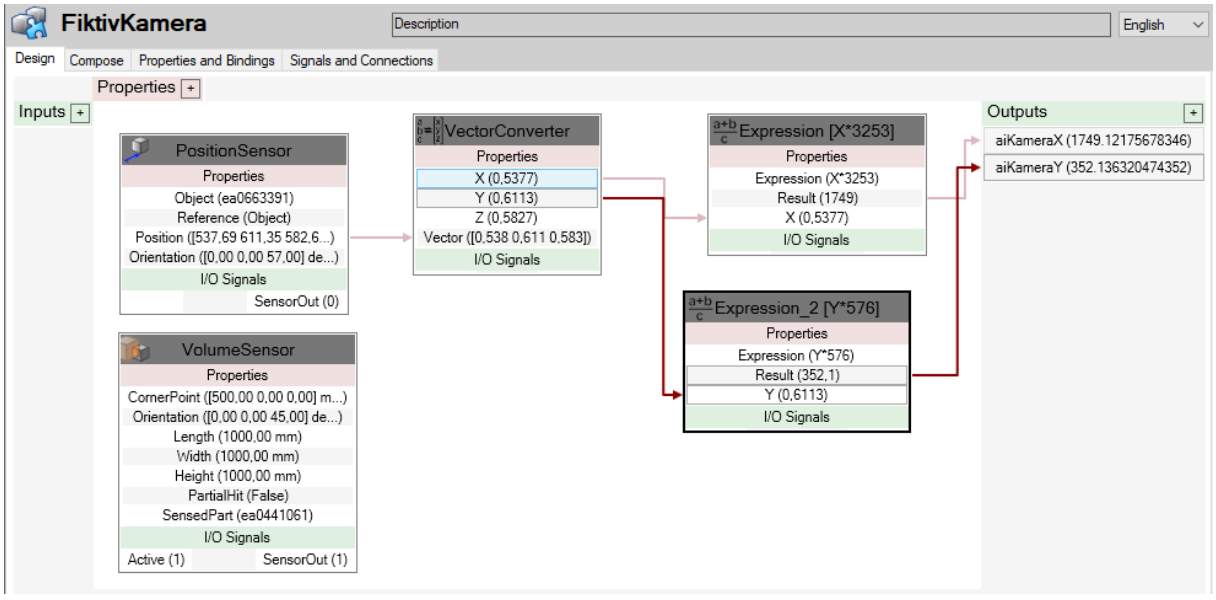
Figur 5.5. *3D-modell av fäste och kalibreringsplatta.*

Slutligen behövdes ett visionsystem som kan skicka koordinater på kalibreringsplattans mittpunkt. En undersökning gjordes av vilka funktioner som finns tillgängliga i RS för att emulera ett visionsystem samt tillägg till RS. Resultatet av undersökningen gav att ingen funktion för emulerad kamera som tar bilder fanns tillgänglig, i stället skapades ett system för att simulera visionmjukvara och kamera. Detta gjordes med hjälp av RS:s smart components (SC), vilket möjliggör skapandet och användande bland annat av givare och logik i den virtuella miljön. För att simulera kamerans koordinatsystem används en positionssensor som återger positionen av mittpunkten av kalibreringsplattan. Koordinaterna som positionssensorn skickar refererar till ett hörn av en volymsensor, se figur 5.6.

Figur 5.6. *Robotcell & volymsensor med koordinatsystem.*



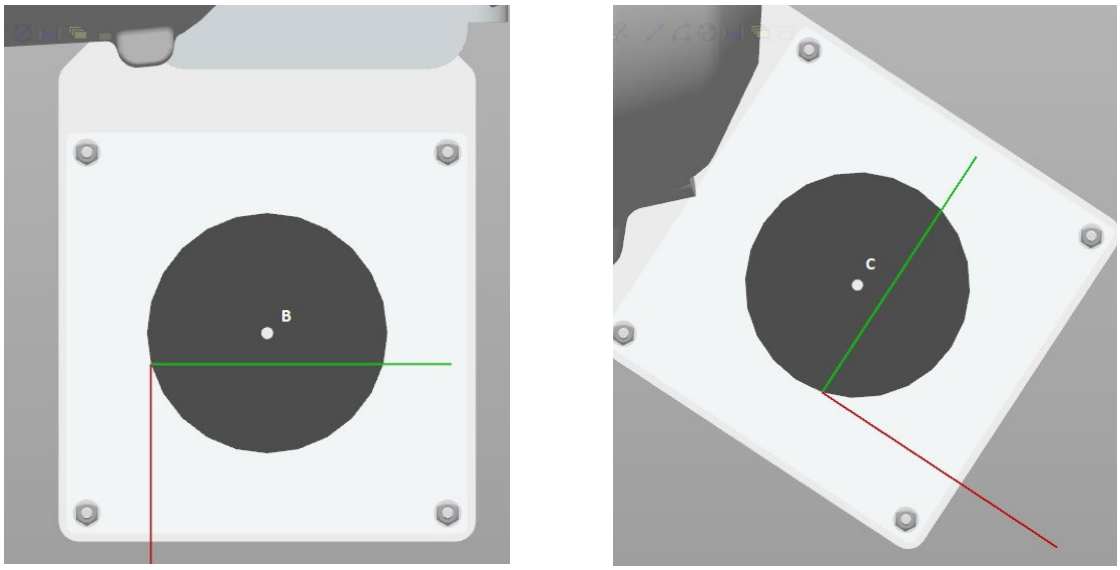
För att åstadkomma
en representation lik verkligheten om hur ett
visionsystem skulle skicka koordinater till roboten, där kamerans pixlar och robotens koordinatsystem i millimeter är förhållandet mellan dem inte 1:1. Robotens koordinatsystem roteras även så att deras X och Y axlar ej är parallella, då båda koordinatsystemen inte nödvändigtvis är justerade till varandra i verkligheten. Positionssensor skickar koordinaterna för kalibreringsplattan i form av en vektor från hörnet av volymsensor till kalibreringsplattans mittpunkt. Vektorn konverteras sedan till tre variabler x, y och z. Endast x och y extraheras från vektorn eftersom ett 2D visionsystem kan endast skicka koordinater i två dimensioner. De extraherade koordinaterna multipliceras med olika faktorer för att åstadkomma olika förhållande till varandra och robotens världskoordinater. Dessa skickas som analoga signaler till robotkontrollern. I robotkontrollern benämns visionsystemets **x** och **y** koordinater sedan som **i** respektive **j** för att urskilja dessa från robotens koordinater. Logiken för kameran framgår i figur 5.7.



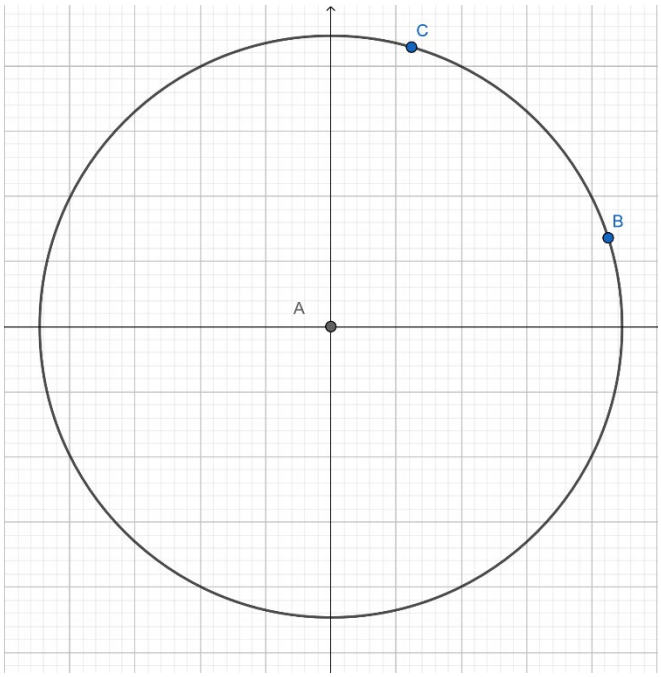
Figur 5.7. *Logik för simulerat visionsystem.*

5.3.2 Utvärdering av TCP

För att korrigera TCP till mittpunkten av kalibreringsplattan behövdes en utvärdering av nuvarande position av TCP. För att avgöra vart TCP:s position relaterat till kalibreringplattans mittpunkt krävs data från robot och kamera. Data införskaffades genom att förflytta robotverktyget till olika punkter och spara positionerna för dessa. Då roboten stannar vid de olika punkterna fotograferar kameran och återger ett värde för kalibreringsplattans position, som paras ihop med tillhörande robotposition och sparas i robotkontrollern. Tiden för rörelse av robot och beräkningar ökar i samband med antalet punkter som används, därför undersöks lösningar med minsta möjliga antal punkter. Vid lösningar med endast en punkt är det inte möjligt att kunna beräkna en relation mellan kamerans och robotens koordinatsystem. Om en förflyttning skulle ske i ett koordinatsystem kan det inte översättas till det andra koordinatsystemet. Användning av två punkter möjliggör utvärdering av hur en rörelse i roboten koordinatsystem påverkar kamerans koordinatsystem. Vid tillfället då TCP är i rätt position i kalibreringsplattans mittpunkt, förändras inte punkten erhållen av kameran vid vridning av TCP om rotationen sker kring en axel som är ortogonal mot planet av 2D-kamerans koordinatsystem. Då position av TCP är inkorrekt, förflyttas plattans mittpunkt i en cirkulär rörelse runt TCP vilket möjliggör att geometriska samband kan utnyttjas. Vid rotering kring TCP kommer de sparade punkterna vara samma avstånd ifrån TCP. Se figurer 5.8 och 5.9.

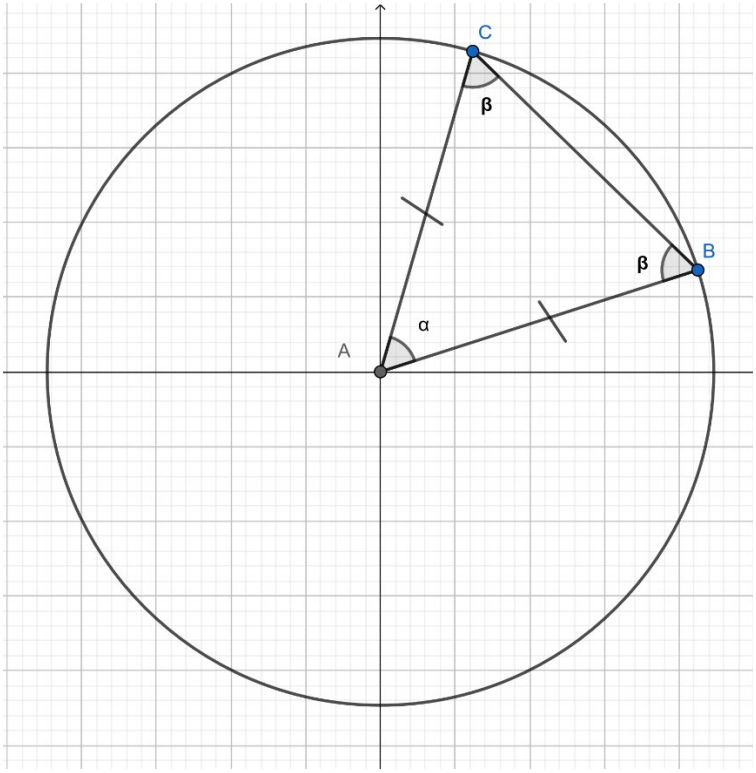


Figur 5.8. *Plattans mittpunkt i relation till TCP.*



Figur 5.9. *Punkter från cirkulär rörelse.*

Detta kan utnyttjas genom att bilda en likbent triangel mellan två punkter och TCP. Den likbenta triangeln kan delas till två rätvinkliga trianglar. Se figur 5.10. Målet med de geometriska sambanden är att få distansen och riktningen från en av punkterna till TCP för att veta vart TCP ska förflytta sig för att komma till rätt plats.



Figur 5.10. *Bildning av likbent triangel.*

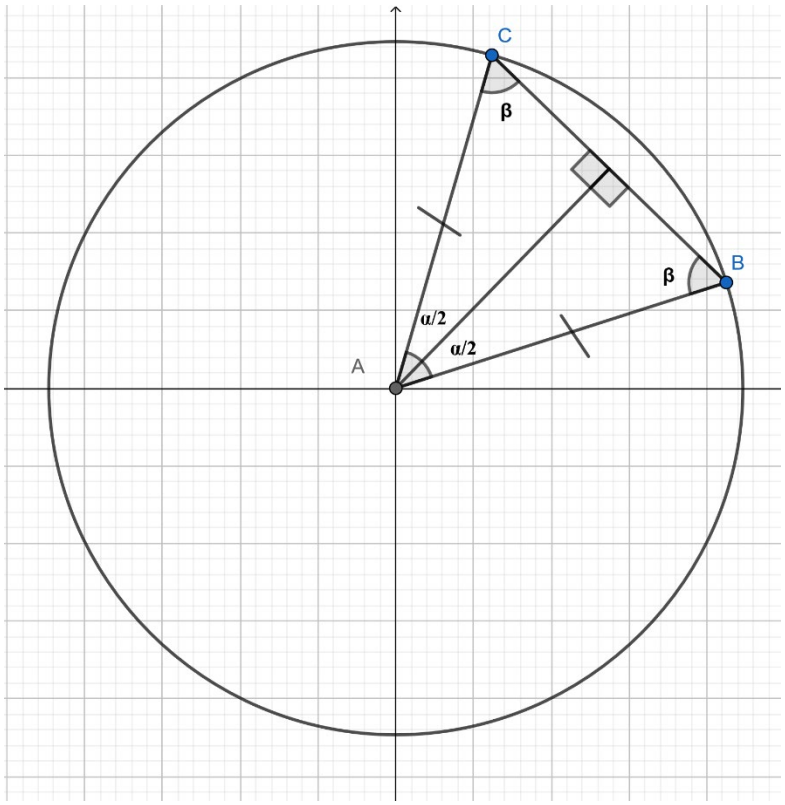
För att förflytta TCP i rätt riktning och distans krävs en vektor $\overline{AB}$ som går från punkt A till B. Detta görs med hjälp av en vektor $\overline{BC}$ som skapas mellan punkt B och C. $\overline{BC}$ roteras med vinkeln β för att rikta den mot punkt A. Vinkeln β räknas ut med hjälp av att vinkeln α är känd och sambanden mellan vinklar i likbenta trianglar i ekvation 5.1.

$$\beta = \left(\frac{180-\alpha}{2}\right)° \qquad (5.1)$$

För att rotera $\overline{BC}$ används en rotationsmatris R_z som vrider vektorn kring dess Z-axel. Rotations matrisen uttrycks som följande:

$$R_z = \begin{pmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (5.2)$$

Distansen mellan punkt A och B är okänd. För att beräkna den utnyttjas de geometriska sammanhangen för att skapa två identiska rätvinkliga trianglar i den likbenta triangeln som kan ses i figur 5.11.



Figur 5.11. *Bilda rätvinkliga trianglar av likbent triangel.*

För att identifiera distansen TCP behöver förflyttas behövs avståndet mellan punkt A och B beräknas, vilket görs med följande formel där i och j betecknar vektorns komponenter.

$$|\overline{BA}| = \frac{|\overline{BC}|/2}{\sin\beta} \qquad (5.3)$$

Magnituden av $\overline{BC}$ ifrån ekvation 5.3 kan beräknas med följande formel.

$$|\overline{BC}| = \sqrt{i^2 + j^2} \qquad (5.4)$$

För att kunna multiplicera skalären med den roterade vektorn $\overline{BC}$ behövs dess enhetsvektor beräknas, enhetsvektorer kommer att betecknas med ”^”.

$$\widehat{\overline{BC}} = \frac{\overline{BC}}{|\overline{BC}|} \qquad (5.5)$$

När enhetsvektorn $\widehat{\overline{BC}}$ multipliceras med$|\overline{BA}|$ fås vektor $\overline{BA}$.

$$\overline{BA} = |\overline{BA}| \left(R_Z * \widehat{\overline{BC}}\right) \qquad (5.6)$$

Negera sedan ekvation 5.6 för att få $-\overline{BA} = \overline{AB}$ vilket ger en förflyttning från TCP till plattans mittpunkt.

Detta ger endast förflyttningen som krävs i kamerans koordinatsystem vilket inte räcker för att flytta den till korrekt position. Vid de tillfällen där kalibreringplattans TCP har en annan orientering än tool0s TCP behövs relationen identifieras och implementeras i förflyttning av kalibreringsplattans TCP. Denna rotering åstadkoms genom att använda förhållandet mellan tool0:s TCP och kalibreringsplattans TCP som uttrycks i kvaternioner till en rotationsmatris. Förhållandet är uttryckt som en rotation från tool0:s TCP till kalibreringsplattans TCP. Därför behövs kvaternionens invers för att få rotation från Kalibreringsplattan till tool0.

$$Q_{tool0}^{platta} = Q_{Platta}^{Tool0\ -1} = (q_w, -q_x, -q_y, -q_z) \qquad (5.7)$$

Därefter kan rotationsmatrisen skapas.

$$R_{tool0}^{platta} = \begin{pmatrix} 1 - 2q_y^2 - 2q_z^2 & 2q_xq_y - 2q_zq_w & 2q_xq_z + 2q_yq_w \\ 2q_xq_y + 2q_zq_w & 1 - 2q_x^2 - 2q_z^2 & 2q_yq_z - 2q_xq_w \\ 2q_xq_z - 2q_yq_w & 2q_yq_z + 2q_xq_w & 1 - 2q_x^2 - 2q_y^2 \end{pmatrix} \qquad (5.8)$$

För att flytta TCP med hänsyn till tool0 multipliceras R_{tool0}^{platta} med $\overline{AB}$.

$$T_{TCP,tool0} = R_{tool0}^{platta} * \overline{AB} \qquad (5.9)$$

RT:s orientering i förhållande till arbetsobjektet behöver beaktas för att säkerställa en korrekt förflyttning av TCP. Detta åstadkoms genom att ta inversen av kvaternionen för RT vilket ger en rotation från RT till arbetsobjektet, vilket bestäms likadant som ekvation 5.7 och 5.8 vilket ger $Q_{Wobj}^{Robtarget} = Q_{Robtarget}^{Wobj\ -1}$ och $R_{Wobj}^{Robtarget}$. Genom att utöka ekvation 5.9 med $R_{Wobj}^{Robtarget}$ ger en ekvation som tar hänsyn till både RT orientering i relation till arbetsobjektet samt kalibreringsplattans TCP orientering i relation till tool0:s TCP.

$$T_{TCP,tool0,wobj} = R_{tool0}^{platta} * R_{Wobj}^{Robtarget} * \overline{AB} \qquad (5.10)$$

För att utvärdera resultatet av den kalibrerade TCP:n utförs en iterativ process där magnituden för T_{TCP} kontrolleras och jämförs med bestämt krav av noggrannhet av TCP. Om magnituden är större än kravet förflyttas TCP och utvärdering upprepas med nya värden på punkt B och C, annars avslutas kalibreringen av TCP. TCP förflyttas genom att addera vektorn $T_{TCP,tool0,wobj}$ med koordinaterna för TCP som kan ses i ekvation 5.11.

$$TCP = (TCP_x + T_{(TCP,tool0,wobj)x}, TCP_y + T_{(TCP,tool0,wobj)y}, TCP_z + T_{(TCP,tool0,wobj)z}) \quad (5.11)$$

5.3.3 Orientering av koordinatsystem

Metoden som utvecklas är en generell metod som skall vara användbar i flertalet robotceller. Därav kan förhållandet mellan robotens och visionsystemets koordinataxlar variera. Därför inleds kalibreringen med att beräkna detta förhållande vilket resulterar i en faktor som kommer att användas för att skapa två rotationsmatriser till och från kamerans koordinatsystem och arbetsobjektets koordinatsystem. Faktorn är hur många millimeter i robotens koordinatsystem är pixlar i kamerans koordinatsystem. Inledningsvis joggas roboten till en önskad position som används som utgångspunkt för TCP-kalibrering, den benämns som $\boldsymbol{t_0}$. Punkten sparas i robotens program, resterande rörelseinstruktioner i programmet utgår från denna punkt. Beräkningen inleds med att kameran fotograferar den nuvarande positionen på kalibreringsplattans mittpunkt och kameran skickar dess koordinater $\boldsymbol{k_0}$ till programmet där de sparas i en variabel, därefter förflyttas kalibreringplattan i arbetsobjektets X-led till $\boldsymbol{t_x}$ där dess nya positionen fotograferas och skickas till programmet $\boldsymbol{k_x}$, denna process upprepas för arbetsobjektets Y-led där $\boldsymbol{t_y}$ och $\boldsymbol{k_y}$ sparas. Därefter kan faktorerna för förhållandet mellan robotens och visionsystemets koordinatsystem beräknas med följande ekvationer.

$$f_i = \frac{1}{\frac{k_x - k_0}{t_x - t_0}} \quad (5.12)$$

$$f_j = \frac{1}{\frac{k_y - k_0}{t_y - t_0}} \quad (5.13)$$

f_i och f_j ger en enhet på millimeter per pixel. Den beräknade faktorn blir inte exakta faktorn eftersom RS avrundar, vilket ger ett osäkrare resultat om ekvationerna 5.12 och 5.13 används som en faktor vid slutgiltig beräkning av förhållande, men faktorn är tillräckligt precis för att skapa rotationsmatriserna. Vid tillfällen där kamerans koordinatsystem har en annan orientering än arbetsobjektet krävs en rotering från arbetsobjekt till kamera, utan att ta hänsyn utav kamerans koordinatsystem i relation till arbetsobjektet resulterar i felaktig kalibrering. En rotationsmatris behöver beräknas som vrider kamerans koordinater till arbetsobjektet (R_{Wobj}^{Kamera}) och en rotationsmatris som roterar robotens koordinater från arbetsobjekt till kamera

(R_{Kamera}^{Wobj}). För att skapa matriserna krävs två vektorer ifrån båda koordinatsystemen där vektorerna har en motsvarighet ifrån det andra koordinatsystemet. Vektorerna bestäms med de tidigare sparade punkterna $\boldsymbol{k_0}$, $\boldsymbol{k_x}$ och $\boldsymbol{k_y}$ samt $\boldsymbol{t_0}$, $\boldsymbol{t_x}$ och $\boldsymbol{t_y}$ och multipliceras med ekvationerna 5.12och 5.13 för att ta hänsyn om en pixel är större eller mindre än en millimeter vilket blir följande vektorer.

$$\bar{r}_1 = f_i * (t_x - t_0) \qquad \bar{r}_2 = f_j * (t_y - t_0)$$

$$\bar{k}_1 = f_i * (k_x - k_0) \qquad \bar{k}_2 = f_j * (k_y - k_0)$$

Där $\bar{r}_1$ motsvarar $\bar{k}_1$ och $\bar{r}_2$ motsvarar $\bar{k}_2$. Matrisen som krävs kan beräknas genom att lösa ut rotationsmatrisen ifrån två ekvationer. Nedsänkta x och y betecknar förflyttning x respektive y-led i vektorerna $\bar{r}_1$, $\bar{r}_2$, $\bar{k}_1$ och $\bar{k}_2$.

$$R_{Kamera}^{Wobj} * \begin{pmatrix} k_{1x} \\ k_{1y} \end{pmatrix} = \begin{pmatrix} r_{1x} \\ r_{1y} \end{pmatrix} \tag{5.14}$$

$$R_{Kamera}^{Wobj} * \begin{pmatrix} k_{2x} \\ k_{2y} \end{pmatrix} = \begin{pmatrix} r_{2x} \\ r_{2y} \end{pmatrix} \tag{5.15}$$

Eftersom både ekvation 5.14 och 5.15 använder sig av samma rotationsmatris är det möjligt att kombinera dessa till en ekvation.

$$R_{Kamera}^{Wobj} * \begin{pmatrix} k_{1x} & k_{2x} \\ k_{1y} & k_{2y} \end{pmatrix} = \begin{pmatrix} r_{1x} & r_{2x} \\ r_{1y} & r_{2y} \end{pmatrix} \tag{5.16}$$

En multiplicering av inversen av kameramatrisen i ekvation 5.16 gör att rotationsmatrisen blir ensam i vänsterledet och rotationsmatrisen kan beräknas med kända värden.

$$R_{Kamera}^{Wobj} = \begin{pmatrix} r_{1x} & r_{2x} \\ r_{1y} & r_{2y} \end{pmatrix} * \begin{pmatrix} k_{1x} & k_{2x} \\ k_{1y} & k_{2y} \end{pmatrix}^{-1} \tag{5.17}$$

Alla vektorer omvandlades till enhetsvektorer vid beräkningen av rotationsmatrisen, då determinanten av kameramatrisen inte behöver beräknas med inversmatrisen samt att strukturen på koden blir tydligare i RS.

$$R_{Kamera}^{Wobj} = \begin{pmatrix} \frac{r_{1x}}{|r_1|} & \frac{r_{2x}}{|r_2|} \\ \frac{r_{1y}}{|r_1|} & \frac{r_{2y}}{|r_2|} \end{pmatrix} * \begin{pmatrix} \frac{k_{1x}}{|k_1|} & \frac{k_{2x}}{|k_2|} \\ \frac{k_{1y}}{|k_1|} & \frac{k_{2y}}{|k_2|} \end{pmatrix}^{-1} \tag{5.18}$$

Enhetsvektorerna i ekvation 5.18 betecknas nu som $\widehat{\bar{r}_1}$, $\widehat{\bar{r}_2}$, $\widehat{\bar{k}_1}$ och $\widehat{\bar{k}_2}$. Innan matrismultiplikationen beräknas inversmatrisen med enhetsvektorer från 5.18 vilket blir följande.

$$\begin{pmatrix} \widehat{\bar{k}_{1x}} & \widehat{\bar{k}_{2x}} \\ \widehat{\bar{k}_{1y}} & \widehat{\bar{k}_{2y}} \end{pmatrix}^{-1} = \begin{pmatrix} \widehat{\bar{k}_{2y}} & -\widehat{\bar{k}_{2x}} \\ -\widehat{\bar{k}_{1y}} & \widehat{\bar{k}_{1x}} \end{pmatrix} \tag{5.19}$$

Nu kan R_{Kamera}^{Wobj} beräknas med en matrismultiplikation.

$$R_{Kamera}^{Wobj} = \begin{pmatrix} \widehat{r_{1x}} & \widehat{r_{2x}} \\ \widehat{r_{1y}} & \widehat{r_{2y}} \end{pmatrix} * \begin{pmatrix} \widehat{k_{2y}} & \widehat{-k_{2x}} \\ \widehat{-k_{1y}} & \widehat{k_{1x}} \end{pmatrix} =$$
$$\begin{pmatrix} \widehat{r_{1x}} * \widehat{k_{2y}} + \widehat{r_{2x}} * \widehat{-k_{1y}} & \widehat{r_{1x}} * \widehat{-k_{2x}} + \widehat{r_{2x}} * \widehat{k_{1x}} \\ \widehat{r_{1y}} * \widehat{k_{2y}} + \widehat{r_{2y}} * \widehat{-k_{1y}} & \widehat{r_{1y}} * \widehat{-k_{2x}} + \widehat{r_{2y}} * \widehat{k_{1x}} \end{pmatrix} \quad (5.20)$$

R_{Wobj}^{Kamera} skapas med samma princip som R_{Kamera}^{Wobj} men med inversen på robotens vektormatris vilket blir följande.

$$R_{Wobj}^{Kamera} = \begin{pmatrix} \widehat{k_{1x}} * \widehat{r_{2y}} + \widehat{k_{2x}} * \widehat{-r_{1y}} & \widehat{k_{1x}} * \widehat{-r_{2x}} + \widehat{k_{2x}} * \widehat{r_{1x}} \\ \widehat{k_{1y}} * \widehat{r_{2y}} + \widehat{k_{2y}} * \widehat{-r_{1y}} & \widehat{k_{1y}} * \widehat{-r_{2x}} + \widehat{k_{2y}} * \widehat{r_{1x}} \end{pmatrix} \quad (5.21)$$

5.3.4 Faktor beräkning

R_{Kamera}^{Wobj} utnyttjas för att kunna skapa faktorer som är närmare den verkliga faktorn än vad ekvationerna 5.11 och 5.12 gör. Detta åstadkoms genom att med två vektorer $\overline{rX}$ och $\overline{rY}$ som gör en förflyttning av RT:s i kamerans x och y-led när de multipliceras med R_{Kamera}^{Wobj} som följande.

$$R_{Kamera}^{Wobj} * \overline{rX} = \overline{kX}$$
$$R_{Kamera}^{Wobj} * \overline{rY} = \overline{kY}$$

Roboten kör till ursprungspunkten och sparar kamerans koordinater och robotens koordinater i k_0 respektive t_0. Sedan rör sig roboten i kamerans x-led och t_x och k_x sparas, detta repeteras för kamerans y-led där t_y och t_y sparas. Därefter kan följande ekvationer skapas.

$$f_i = \frac{t_x - t_0}{k_x - k_0} \quad (5.22)$$

$$f_j = \frac{t_y - t_0}{k_y - k_0} \quad (5.23)$$

Faktorerna f_i och f_j multipliceras med x respektive y koordinaterna av punkterna B och C vid skapandet av vektorn $\overline{BC}$.

Rotationsmatrisen R_{Wobj}^{Kamera} multipliceras med vektor $\overline{BC}$ för att rotera vektorn från kamerans koordinatsystem till arbetsobjektet koordinatsystem.

5.4 Utvärdering

Det sista steget i DRM innebär att arbetet utvärderades och verifierades. För att utföra verifieringen delar Blessing & Chakrabarti (2009) upp den i tre steg: Utvärdering av stöd, utvärdering av tillämpbarhet och utvärdering av framgång. Utvärdering av stöd skedde

kontinuerligt under utvecklingen av kalibreringsmetoden och det är först när det är färdigt som utvärderingen av tillämpbarheten kan utföras. Utvärdering av tillämpbarhet undersökte hur stödet påverkar nyckelfaktorerna i projektet. Till sist sker utvärdering av framgång som skall besvara hur väl stödet når upp till de utsatta målen i form av positiv påverkan på de mätbara framgångsfaktorerna. Utvärderingen av framgång skall göras i praktiken och kan kräva långsiktigt arbete, vilket leder till att det inte är möjligt att utföra under projektets utsatta tidsram.

5.4.1 Utvärdering av stöd

Enligt Blessing & Chakrabarti (2009) bör utvärdering av stöd utföras innan resterande utvärdering genomförs. Det innebär att stödet genomgår fortlöpande tester under utvecklingsfasen. Stödet utvärderades med hjälp av de krav som framställts med MoSCoW metoden, se tabell 5.2. Status för kravet markeras med färg, grön färg betyder att kravet är uppfyllt medans röd färg betyder att kravet ej är uppfyllt.

Tabell 5.2. *MoSCoW-tabell med resultat*

Typ	Krav	Status
Must have	• Kalibrering X- och Y-koordinaterna av TCP	Metoden av TCP:s X- och Y-koordinater utförs Utvärderingen utförs iterativt då kalibrering upprepas tills att önskat resultat är uppnått
	• Artefakten skall vara kompatibel med ABB:s programmeringsspråk RAPID	Artefakten är utvecklad och skriven i RAPID
	• Utvärdera TCP efter kalibrering	Metoden utvärderar TCP innan och efter kalibrering Utvärderingen utförs iterativt då kalibrering upprepas tills att önskat resultat är uppnått
	• Metoden skall hantera att kamerans koordinataxlar har annan orientering runt Z-axeln än robotens koordinatsystem	Metoden kan kalibrera oavsett orientering på kamerans koordinatsystem runt dess Z-axel
Should have	• Anpassningsbara rörelseinstruktioner i roboten för att öka tillämpbarheten där utrymme för rörelser är begränsat	Rörelselängden för rörelser i robotens X- och Y-koordinater är anpassningsbara Antalet grader som roteras i rotationsrörelsen för utvärdering av TCP är anpassningsbar mellan 1° och 90° Rotationer över 90° har ej testats och därför finns ingen data om vilka övriga rotationer som är möjliga
Could have	• Kalibrering av Z koordinaterna av TCP	Ej uppfylld
	• Hantering av att kamerans och robotens koordinataxlar för Z ej är parallella	Ej uppfylld

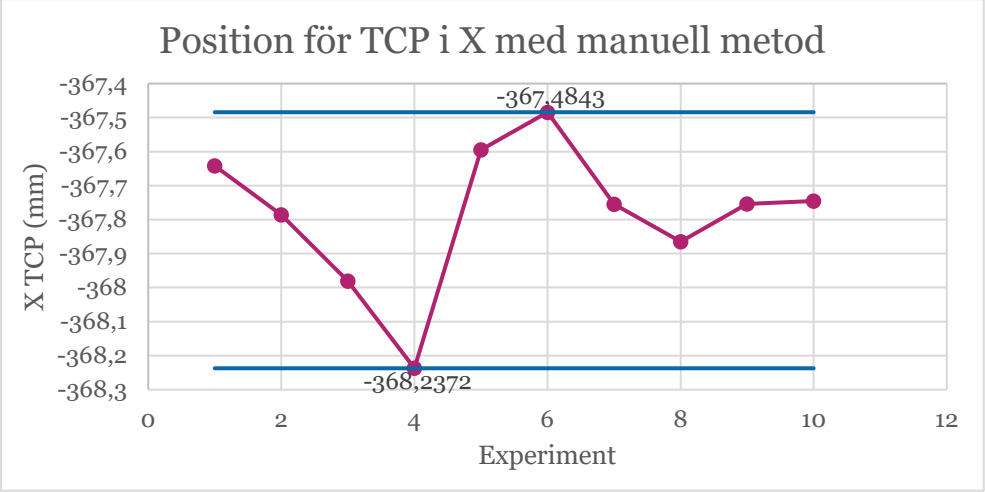
5.4.2 Utvärdering av tillämpbarhet

Utvärdering av tillämpbarhet kallas den del av DRM som utvärderar hur nyckelfaktorn/faktorerna påverkas av stödet (Blessing & Chakrabarti, 2009). Experiment med artefakten som nyttjar metoden för att automatiskt kalibrera ett nominellt TCP har påvisats fungera under specifika förutsättningar. Den första av dessa förutsättningar är att det skall finnas ett nominellt TCP som kan användas som utgångspunkt för rörelseinstruktioner under

kalibreringsprocessen. Den andra förutsättningen är att Z-axlarna i kamerans samt robotens koordinatsystem skall vara parallella.

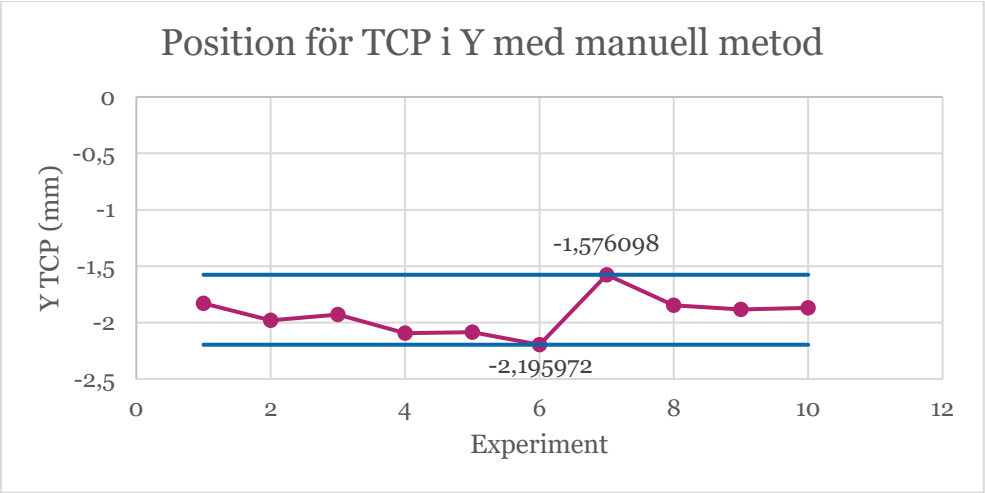
För att utvärdera hur stödet påverkar nyckelfaktorn som är noggrannhet för kalibrering av TCP så har experiment utförts där TCP kalibreras med de båda metoderna. Först användes en fyrapunkts manuell kalibreringsmetod i en fysisk robotcell. Centrum på en kalibreringsplatta fördes till spetsen på ett metallspjut fyra gånger så att roboten (ABB IRB4600) stod i olika konfigurationer vid varje tillfälle. Kalibreringen utfördes tio gånger och resultatet noterades för samtliga experiment. Därefter utfördes kalibrering med den utvecklade metoden i den virtuella stationen tio gånger. Till sist utfördes tio kalibreringar med den utvecklade metoden igen, fast i ett fysiskt system. För att utvärdera resultatet från kalibreringen dokumenterades de kalibrerade positionerna för TCP i X-led och Y-led vid alla experiment för att sedan få ut intervallen som TCP varierade inom, sammanställning av datan ses i bilaga B.

Vid experiment med den manuella metoden varierade positionen för TCP i X-led från tool0 mellan -367,4843 mm och -368,2372 mm. Det ger ett intervall med storleken 0,7529 mm, se figur 5.12.



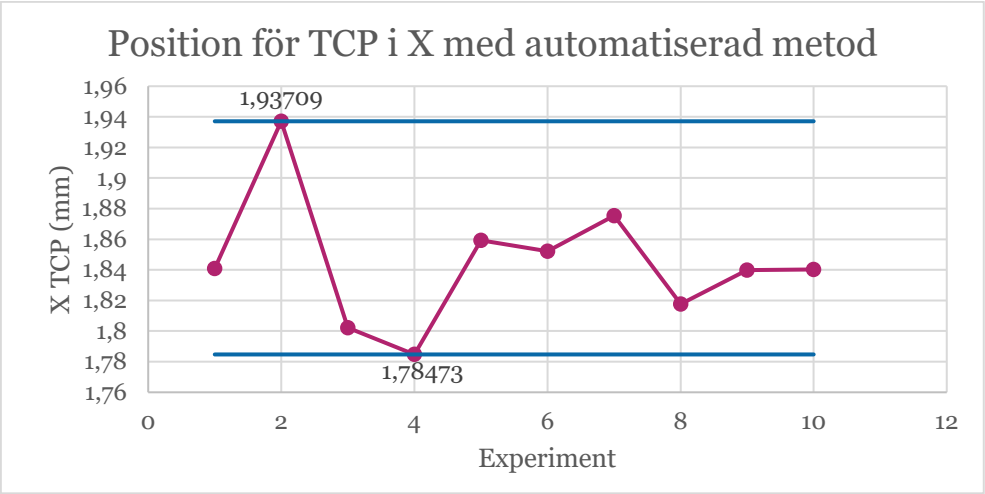
***Figur 5.12**. Resultat av experiment för manuell kalibrering för position för TCP i X-led från tool0.*

Detsamma gjordes för Y-led. Positionen varierade mellan -2,195972 och -1,576098, vilket ger ett intervall på 0,619873 mm, se figur 5.13.



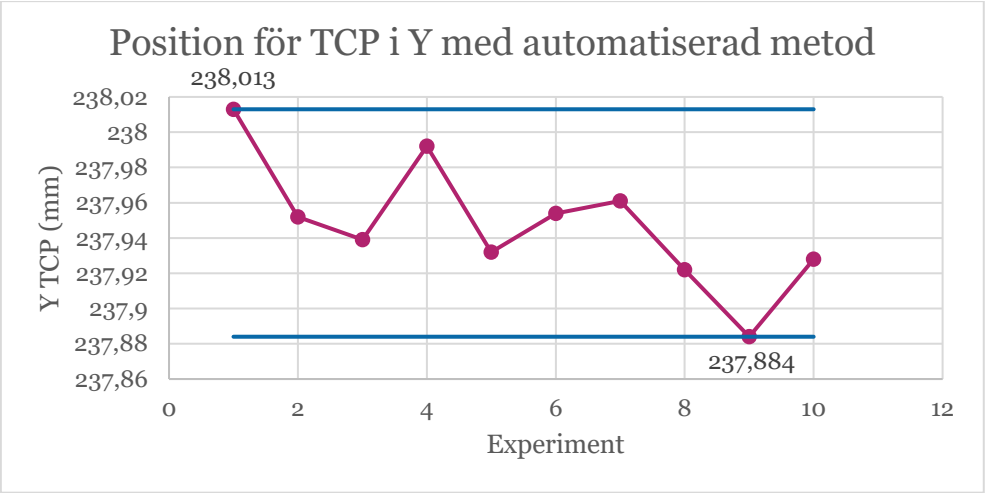
Figur 5.13. *Resultat av experiment för manuell kalibrering för position för TCP i Y-led från tool0.*

Experimenten som utfördes med den automatiserade metoden i en virtuell miljö gav ingen variation för position på TCP. I den virtuella miljön är det möjligt att mäta exakt hur långt TCP hamnar från mittenpunkten i plattan, och kalibreringen resulterade i att TCP hamnade perfekt i mittpunkten under alla tio experiment. Det bevisar att den matematiska modellen som ställts upp är korrekt, men det visar inte hur metoden hanterar de störningar som uppkommer i ett fysiskt system. För att utvärdera detta utfördes tio experiment med ett fysiskt system som består av en ABB IRC5 IRB1200 (robot) och en Cognex Insight 5100–01 (kamera) då det var dessa som fanns tillgängliga vid tillfället för experimenten. För att utvärdera hur värdet som visionsystemet återger kan variera vid centrum i kamerans bild på en utsatt höjd togs 20 bilder på kalibreringsplattan i samma position. Värdet varierade med ett intervall på ca 0,08 pixlar i X-led och 0,06 pixlar i Y-led. Det är sedan i den punkten de sista utvärderingarna av TCP utfördes i under varje experiment. Experimentet resulterade i att positionen i X-led varierade mellan 1,78473 och 1,93709, vilket innebär ett intervall på 0,15236 mm, se figur 5.14.



Figur 5.14. *Resultat av experiment för automatiserad kalibrering för position för TCP i X-led från tool0.*

I Y-led varierade resultatet mellan 237,884 och 238,013, ett intervall på 0,129 mm, se figur 5.15.



Figur 5.15. *Resultat av experiment för automatiserad kalibrering för position för TCP i Y-led från tool0.*

För en sammanställning av resultateten, se tabell 5.3.

Tabell 5.3*. Sammanställning av resultatet från experiment av noggrannhet med den manuella metoden och den automatiserade metoden i ett fysiskt system.*

Metod	**Manuell kalibreringsmetod**		**Automatiserad kalibreringsmetod**	
Axel	**X**	**Y**	**X**	**Y**
Max	-367,4743	-1,576098	1,93709	238,013
Min	-368,2372	-2,195972	1,78473	237,884
Intervall	0,7529	0,619873	0,15236	0,129

Då intervallet för resultatet av den manuella kalibreringsmetoden är cirka 494% större i X-led och 480% större i Y-led än för den automatiserade metoden tyder det på att noggrannheten är bättre efter implementation av stödet.

5.4.3 Framgångsutvärdering

Vid framgångsutvärderingen undersöks hur användbart det skapade stödet är på att uppnå de uppsatta framgångskriterierna. För att mäta tidsåtgång vid samkalibrering krävs att artefakten är produktionsredo, vilket inte är möjligt i artefaktens nuvarande stadie. Med en artefakt som är parkopplad med ett verkligt visionsystem kan kvantitativa data samlas in i form av åtgången tid för hela samkalibreringsprocessen. Den datan kan sedan jämföras med nuvarande samkalibreringsprocessen efter datainsamling, vilket ger en indikation på hur personalbördan påverkas. Då flera steg i en manuell process kan utföras automatisk med framtagen metod anses det att personalbördan kommer minska. Den andra mätbara framgångsfaktorn är kvalité av produkt vilket är utom tidsramen för projektet, eftersom den kan mätas först när produkten är levererad till kund. Data kan samlas in genom kvalitativ undersökning med kund, exempelvis genom enkäter eller intervjuer där frågor ställs hur kunden upplevde produkten och dess positiva och negativa egenskaper.

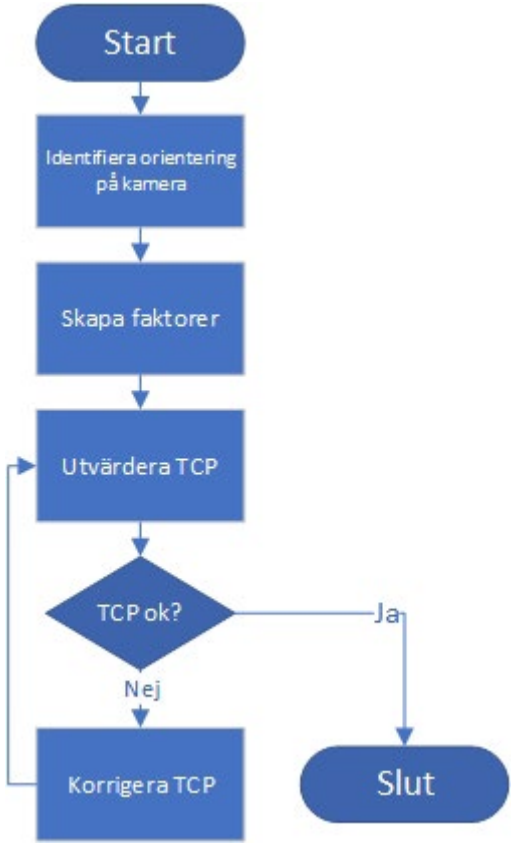
Då utvärdering av båda mätbara framgångsfaktorer faller utanför projektets tidsram är det omöjligt att identifiera arbetets påverkan på framgångsfaktorn om vinstbelopp. Därför kan inte modellen för framtida önskat läge valideras.

6 Resultat

I följande kapitel presenteras projektets resultat översiktligt.

6.1 Metod för kalibrering av TCP

Arbetet resulterade i en metod för kalibrering av nominell TCP med assistans av ett 2D visionsystem som kan identifiera ett verktygs önskade TCP position. Metoden beräknar hur fel nuvarande TCP är placerad gentemot vart TCP ska vara placerad i form av en vektor. För att riktningen av förflyttningen ska stämma i robotens koordinatsystem multipliceras vektorn med en rotationsmatris som vrider vektorn. Det gör att förflyttningen stämmer med tool0:s (verktygflänsens) koordinatsystem. Därefter multipliceras en annan rotationsmatris med vektorn far att ta hänsyn till orientering på RT i förhållande till arbetsobjektet. Metoden kan hantera kameror som har annan orientering än robotens koordinatsystem och olika längder på pixlar gentemot robotens millimeter. Stegen för kalibreringsmetoden ses i figur 6.1.



Figur 6.1. *Diagram med kalibreringssteg*

6.2 Artefakt

Artefakten är skapad med programvaran RS i formen av kod i programmeringsspråket Rapid. Artefakten testats i en robotcell som innehåller en robot med en monterad kalibreringsplatta. För att emulera ett visionsystem användes SC:s som kan identifiera mittpunkten av kalibreringsplattan och ge ut dess X och Y koordinater. Koden kan hittas i bilaga C.

6.3 Resultat från utvärdering

Efter utvärdering påvisas metoden tillräcklig för att uppnå de målen som skapades med MoSCoW-matrisen i samband med forskningsförtydligandet.

Stödet som utvecklades påverkade arbetets nyckelfaktor, noggrannhet vid kalibrering av TCP. Den automatiserade kalibreringen uppmättes i RS och på ett fysiskt visionsystem att vara mer noggrann än den tidigare manuella kalibreringsmetoden som utfördes fysiskt. Den manuella kalibreringens TCP varierade inom ett intervall av 0,75 mm i X-led och 0,62 mm i Y-led ifrån tool0 position på tio experiment. Kalibrering med den automatiserade metoden i en virtuell miljö återgav den exakta positionen för ett korrekt TCP. Fysiska experiment utfördes med ett riktigt visionsystem och robot som gav ett resultat där TCP varierade inom ett intervall av 0,15 mm i X-led och 0,13 mm i Y-led ifrån tool0 position på tio experiment.

En framgångsutvärdering för projektet har ännu inte utförts då artefakten inte är redo för produktion samt att den måste göras under en längre tid som inte är möjlig under projektets tidsram.

7 Diskussion

I följande kapitel diskuteras projektets resultat, problem och potential för vidareutveckling.

7.1 Diskussion av projektets resultat

Projektet resulterade i en metod för kalibrering av robot TCP utifrån data från ett tvådimensionellt visionsystem. Innan en kalibrering skall utföras måste en robotoperatör skapa ett RT som utgångspunkt för kalibreringen. Orientering på detta RT kan variera på grund av olika förutsättningar som skiljer sig från robotcell till robotcell. Därför har orientering på RT tagits med i beräkningarna för att möjliggöra valfri orientering med kravet att dess Z-axel är parallellt med kamerans teoretiska Z-axel. Exempel på dessa förutsättningar kan vara att utrymmet i cellen är begränsat och möjligheter för rörelse i cellen är därför begränsad. Kamerans placering kan göra att kravet på konfiguration på roboten varierar. Om fel konfiguration riskerar delar av roboten att skymma verktyget för kameran. Metoden kan hantera att orienteringen på kamerans koordinatsystem inte stämmer överens med orienteringen på robotens koordinatsystem, med förutsättningen att Z-axeln på båda koordinatsystem är parallella. I de fall detta inte stämmer kan ett arbetsobjekt skapas i robotens kontroller där dess Z-axel är parallell med kamerans. Detta kan vara svårt och därför hade ett önskemål varit att endast behöva skapa kalibreringens RT med Z-axlarna parallella för det är något enklare, tyvärr faller det ur tidsramen för projektet. Metoden är anpassad för att rörelseinstruktionerna skall kunna anpassas utefter behov men det finns vissa fall som metoden för tillfället inte kan hantera, exempelvis vissa vridningar vid utvärdering om TCP. Dessa fall skulle behöva undersökas för att identifiera lösningar.

7.2 Diskussion om artefakten

Projektet har levererat en artefakt i form av kod som utför den kalibrering. Koden är skapad i ABB:s språk rapid och är därmed kompatibel med de flesta ABB robotsystem. För tillfället är koden inte anpassad för att användas i produktion utan den är mest ett bevis för att metoden fungerar. Artefakten är inte anpassad för att ta emot signaler från alla visionsystem utan endast virtuellt ifrån SC i RS samt manuell inmatning av koordinater från fysiskt visionsystem. Artefakten har inga anpassningsbara rörelseinstruktioner ännu. Om operatören vill minska vridningen av verktyget vid utvärdering av TCP är detta ännu inte möjligt. Då denna funktion är testad genom olika ändringar i artefakten har möjligheterna för anpassningsbara rörelseinstruktioner identifierats, men det är utom projektets tidsram att implementera dessa funktioner i artefakten. Då artefakten skall levereras till EA måste den anpassas för att uppfylla vissa standarder som EA arbetar med.

7.3 Diskussion om utvärdering

Vid testning av artefaktens noggrannhet i en virtuell värld uppnåddes alltid ett perfekt resultat, där TCP felar med 0,00 mm i samtliga koordinataxlar. Resultatet är dock inte helt pålitligt, eftersom testningen av artefakt utfördes i en simulerad värld där inga störningar sker och alla

uträkningar ger exakta värden blir noggrannheten perfekt, vilket endast innebär att matematiken kalibreringsmetoden utnyttjar fungerar. Kalibreringsmetoden testades därefter i ett fysiskt visionsystem för att bevisa att kalibreringsmetoden fungerar i verkligheten. Resultatet tyder på att den automatiska kalibreringsmetoden ger ett noggrannare resultat än den manuella metoden trots olika störningar som kan påverka kameran som ljusskillnader och kamerans noggrannhet. För att få ett pålitligare resultat skulle den sanna positionen av vart TCP ska vara i verkligheten finnas med för att jämföra med positionen som metoderna gav. I stället fås intervall på hur utspritt alla positionerna på TCP utifrån tool0 efter att kalibreringsmetoden bestämde att positionen för TCP var tillräckligt korrekt. Experimenten för den manuella och automatiska kalibreringsmetoden använde olika robotar för kalibrering av TCP. Detta ger ett opålitligare resultat då olika robotars precision på rörelser kan variera. Kamerans noggrannhet i millimeter hade varit fördelaktigt att ha information kring, eftersom dess noggrannhet är en begränsning på hur noggrann kalibreringsmetod kan vara. Det som kan antas ifrån kameranoggrannheten är att kameran som användes för kalibreringen av TCP var tillräcklig för att få ett bättre resultat än den manuella kalibreringsmetoden. Om fler antal experiment hade utförts med den automatiska och manuella kalibreringsmetoden skulle pålitligheten av resultaten stärkts ytterligare, på grund av brist av tid under projektet gång. Då utvärdering av framgång föll utanför projektets tidsram kan inte heller påverkan på framgångsfaktorn om vinstbelopp utvärderas. Detta leder till att det är omöjligt att validera modellen över det framtida önskade läget utvärderas.

8 Slutsatser och framtida arbete

Syftet med arbetet var att ta fram en automatiserad kalibreringsmetod för ett nominellt TCP i X och Y led med hjälp av ett 2D visionsystem för EA:s samkalibreringsprocess. En noggrannhet ska erhållas av TCP:s nya placering för att sedan jämföras med den manuella kalibreringsmetoden. Projektets mål delades upp i tre delmål som blev att kartlägga den nuvarande processen, utveckla en automatiserad kalibrerings process för TCP och till sist ta fram en noggrannhet av den automatiserade kalibreringsprocessen för att sedan jämföra den med den nuvarande kalibreringsprocessen. Arbetet resulterade i en generell metod för kalibrering av TCP samt en artefakt i form av kod i programvaran RS som användes för att utvärdera metoden. Metoden räknar ut förhållanden mellan kamerans koordinatsystem och robotens koordinatsystem för att sedan utnyttja geometriska samband för att beräkna hur mycket TCP ska förflyttas i robotens koordinatsystem samt vilken riktning.

Under arbetet har metoden utvärderats genom de metoder som vanligtvis används med DRM. Under utvärdering av stödet uppfylldes fyra av sju krav varav två av kraven som ej uppfylldes var i kategorin could have av en MoSCoW-tabell. Ett should have krav blev delvis uppfyllt, men då inget fullständigt test hade genomförts kan det inte anses vara helt uppfyllt. Vid utvärdering av tillämpbarhet anses stödet påverka den utsatta nyckelfaktorn positivt. Ytterligare undersökningar krävs för att säkerställa resultatet av utvärderingen. Utvärdering av framgång är utom projektets tidsram och kräver vidare arbete i form kvantitativa undersökningar om tidsåtgång för samkalibrering och kvalitativa undersökningar om kundnöjdhet kring produkten kvalité.

Framtida arbete kan innefatta vidare utveckling och utvärdering av metoden, exempelvis undersöka möjligheter för att kalibrera TCP:s Z-axel eller möjliggöra kalibrering av TCP där arbetsobjektet Z-axel inte är parallellt med kamerans Z-axel. Ytterligare kan metoden bli bättre på att hantera matematiska undantag för anpassningsbara rörelseinstruktioner. Artefakten behöver anpassas för att arbeta med riktiga visionsystem och den är ej fullständigt anpassad för att hantera anpassningsbara rörelseinstruktioner. För att erhålla mer realistiska data behöver mer experiment utföras för att testa skillnaden i noggrannhet för den manuella och automatiserade metoden. Efter att artefakten är klar för att användas i produktion bör en långvarig studie där de påverkan på de mätbara framgångsfaktorerna utvärderas. För att sedan utvärdera hur framgångsfaktorn vinstbelopp påverkas som möjliggör validering av modellen för önskat läget.

Litteraturförteckning

ABB Robotics, 2010. *Technical reference manual: RAPID Instructions, Functions and Data types.* Revision J ed. Västerås: ABB Robotics.

Amini, M. & Shanbehzadeh, J., 2009. An Experimental Machine Visionsystem for Quality Control of Industrial Colour Printer. *Internation Conference on Machine Vision,* Volume 2, pp. 225-228.

Axelsson, D. & Tangen, S., 2008. *Strategi och produktionsutveckling: handbok för utformning av produktionsstrategi och det framtida produktionsystemet.* Stockholm: KTH Industrill produktion.

Blessing, L. & Chakrabarti, A., 2009. *DRM, a Design Research Methodology.* London: Springer.

Boyes, H., Hallaq, B., Cunningham, J. & Watson, T., 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry,* Volym 101, pp. 1-12.

Budin Shah, N. S. B. E., Osman, K. & Zainuddin, N., 2020. 5-Axis of Mitsubishi RV-2AJ Robotics Arm Modelling Using Solidworks. *IEEE International Colloquium on Signal Processing & Its Applications (CSPA),* Volume 16.

Cheng, F., 2017. Robot manipulation of 3D cylindrical objects with a robot-mounted 2D vision camera. *Computing conference.*

Cheng, F. S. & Denman, A., 2005. A study of using 2D visionsystem for enchanced industrial robot intelligence. *IEEE International Conference Mechatronics and Automation,* Volym 3, pp. 1185-1189.

Development World Comission on Environment and Development, 1987. *Our Common Future,* Oxford: Oxford University Press.

Eshah Budin Shah, N., Osman, K. & Zainuddin, N., 2020. 5-Axis of Mitsubishi RV-2AJ Robotics Arm Modelling Using Solidworks. *2020 16th IEEE International Colloquium on Signal Processing & its Applications,* 16(1), pp. 207-212.

Groover, M., 2008. *Automation, Productions Systems, and Computer-Integrated Manufacturing.* 3:e red. Upper Saddle River, New Jersey: Prentice Hall, Inc.

Gulliksson, H. & Holmgren, U., 2018. *Hållbar utveckling: Teknik, samhälle och livskvalitet.* 3 ed. Lund: Studentlitteratur AB.

Gunal, M. M., 2019. *Simulation for Industry 4.0 : Past, Present, and Future.* Cham: Springer.

Hallenberg, J., 2007. *Robot Tool Center Point Calibration using Computer Vision,* Linköping: Linköping University.

Hanson, A. J., 2006. *Visualizing Quaternions.* San Francisco, CA.: Morgan Kaufmann.

Jordan, J. M., 2016. *Robots.* Cambridge, MA.: The MIT Press.

Lemurell, S., 2010. *Linjär algebra - Från en geometrisk utgångspunkt.* 1:6 ed. Lunds: Studentlitteratur AB.

Ollson, G. & Rosen, C., 2005. *Industrial automation : applications, structures and systems.* 3:e ed. Lund: Department of Industrial Electrical Engineering and Automation, Lund University.

Persson, C. & torsten, P., 2011. *Hållbar utveckling - människa, miljö och samhälle.* 2:a ed. Lund: Studentlitteratur.

Projektledning, 2021. *Projektledning.* [Online] Available at: https://projektledning.se/moscow-metoden/ [Accessed 23 Mars 2022].

Rembold, U., 1990. *Robot technology and applications.* New York: Marcel Dekker.

Robinson, S., 2019. *Simulation The practise of model development and use.* 2:a ed. London: Red Globe Press.

Sakin, M. & Kiroglu, Y., 2017. 3D Printing of Buildings: Construction of the Sustainable Houses of the Future by BIM. *Energy Procedia*, 23 Oktober, pp. 702-711.

Schmidt, B. & Wang, L., 2014. Automatic work objects calibration via a global–local camera system. *Robotics and Computer-Integrated Manufacturing,* 30(6), pp. 678-683.

Skorobogatova, N., 2019. Sustainable Development of an Enterprise Under Industry 4.0 Conditions. *2019 International Conference on Creative Business for Smart and Sustainable Growth,* pp. 1-5.

Snyder, W. E., 1985. *Industrial robots: Computer interfacing and control.* Engelwood Cliffs, N.j.: Prentice Hall.

Swedac, 2022. *Ämnesområden: Kalibreringslaborationer.* [Online] Available at: https://www.swedac.se/amnesomraden/kalibrering/ [Accessed 22 Februari 2022].

Säfsten, K. & Gustavsson, M., 2019. *Forskningsmetodik för ingenjörer och andra problemlösare.* Lund: Studentlitteratur.

Tan, K. S. et al., 2021. Integrating Vision System to a Pick and Place Cartesian Robot. *Journal of Physics,* 2107(1).

Tonnquist, B., 2021. *projektledning.* 8:e ed. Stockholm: Sanoma utbildning AB.

Wang, J. & Olson, E., 2016. *AprilTag 2: Efficient and robust fiducial detection.* s.l., s.n., pp. 4193-4198.

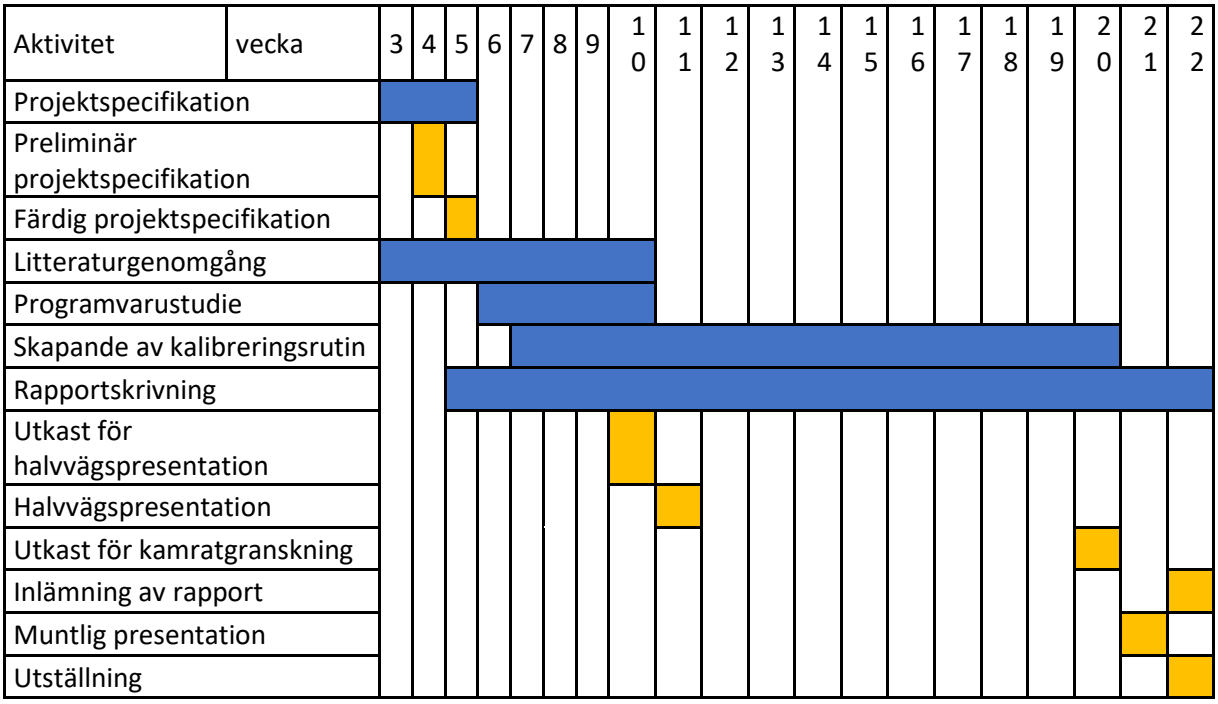
Yang, Z., Gong, L. & Liu, C., 2021. Efficient TCP Calibration Method for Vision Guided Robots Based on Inherent Constraints of Target Object. *IEEE Access,* Volume 9, pp. 8902-8911.

Bilaga A Preliminär och aktuell tidsplan för projekt

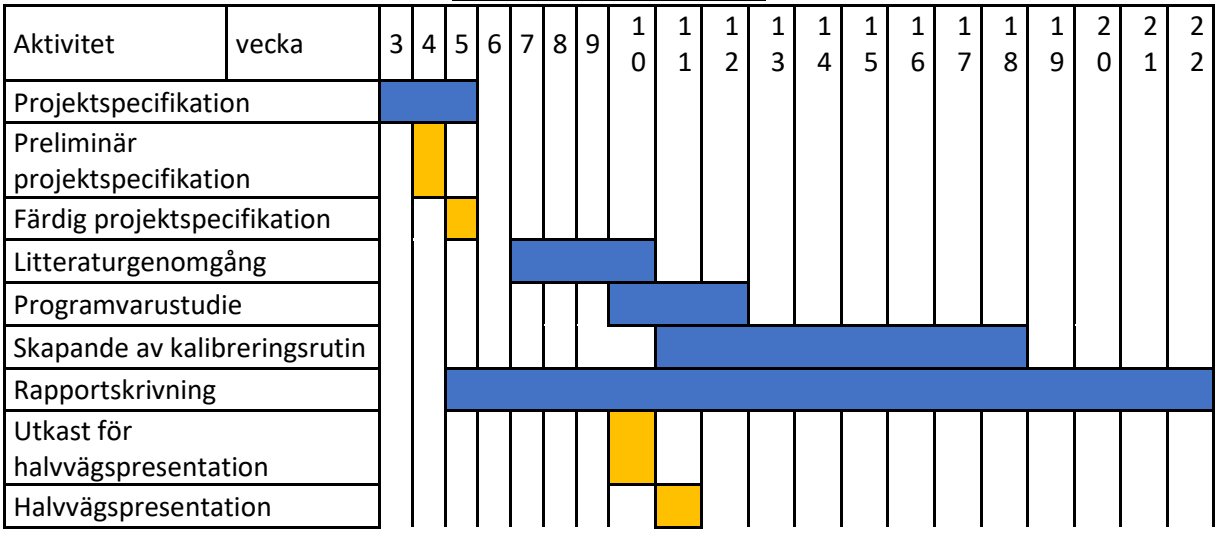
Den preliminära tidsplanen skiljer sig inte drastiskt med den faktiska tidsplanen. Det skedde en överskattning på hur tidskrävande skapandet av kalibreringsrutinen var samt att litteraturgenomgången utfördes snabbare än planerat.

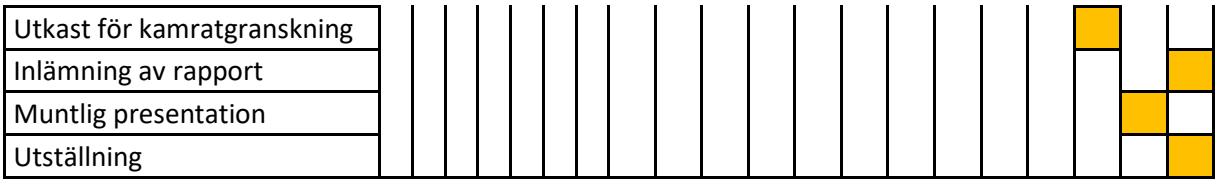
Tabell A.1. *Preliminär tidsplan som skapades inför projektets start och aktuell tidsplan som har uppdaterats under projektets genomförande.*

Preliminär tidsplan



Aktuell tidsplan




Utkast för kamratgranskning
Inlämning av rapport
Muntlig presentation
Utställning

Bilaga B Data för Experiment med Kalibreringsmetoderna

Tabell B.1. *Noggrannhet för manuell kalibrering av TCP.*

Test för manuell kalibreringsmetod				
Experiment	**X (mm)**	**Y (mm)**	**Z (mm)**	**Fel (mm)**
1	-367,6417	-1,829061	235,6583	0,6567893
2	-367,7855	-1,97878	235,6791	0,7866702
3	-367,9803	-1,928476	237,0759	0,5612689
4	-368,2372	-2,092419	237,95556	0,4013789
5	-367,5952	-2,086348	236,3874	0,9043201
6	-367,4843	-2,195972	237,0361	0,6281746
7	-367,7547	-1,576098	235,6704	0,9484702
8	-367,8645	-1,847215	236,2802	0,592815
9	-367,7541	-1,883408	236,60835	0,7420308
10	-367,7455	-1,870433	236,63956	0,75470745
Medel				0,697663
Median				0,699410
Max	-368,2372	-2,195972		0,9484702
Min	-367,4843	-1,576098		0,4013789
Max-Min	0,7529	0,619873		0,5470913

Tabell B.2. *Noggrannhet för automatiserad kalibrering av TCP i virtuell miljö.*

Test för automatiserad kalibreringsmetod i virtuell miljö				
Experiment	**X (mm)**	**Y (mm)**	**Z (mm)**	**Fel (mm)**
1	-175,167	-10,0012	624,2	0
2	-175,167	-10,0012	624,2	0
3	-175,167	-10,0012	624,2	0
4	-175,167	-10,0012	624,2	0
5	-175,167	-10,0012	624,2	0
6	-175,167	-10,0012	624,2	0
7	-175,167	-10,0012	624,2	0
8	-175,167	-10,0012	624,2	0
9	-175,167	-10,0012	624,2	0
10	-175,167	-10,0012	624,2	0
Medel				0
Median				0
Max	-175,167	-10,0012		0

Min	-175,167	-10,0012		0
Max-Min	0	0		0

Tabell B.3. *Noggrannhet för automatiserad kalibrering av TCP i fysiskt system.*

Test för Automatisk kalibreringsmetod i fysiskt system		
Experiment	**X (mm)**	**Y (mm)**
1	1,84089	238,013
2	1,93709	237,952
3	1,80224	237,939
4	1,78473	237,992
5	1,85932	237,932
6	1,85225	237,954
7	1,87531	237,961
8	1,81771	237,922
9	1,8398	237,884
10	1,84031	237,928
Medel	1,844965	237,9477
Median	1,8406	237,9455
Max	1,93709	238,013
Min	1,78473	237,884
Max-Min	0,15236	0,129

Tabell B.4. *Resultat av experiment för kamerans noggrannhet.*

Experiment för kamerans noggrannhet		
Experiment	x	y
1	293,632	336,511
2	293,589	336,485
3	293,587	336,492
4	293,639	336,486
5	293,619	336,491
6	293,6	336,475
7	293,621	336,507
8	293,623	336,496
9	293,604	336,519
10	293,595	336,518
11	293,648	336,487
12	293,608	336,459
13	293,59	336,514
14	293,594	336,493
15	293,571	336,487
16	293,637	336,457
17	293,619	336,509
18	293,579	336,496
19	293,647	336,478

20	293,587	336,509
Max	293,648	336,519
Min	293,571	336,457
Intervall	0,077	0,062

Bilaga C Artefakt

```
MODULE mMainModule
  TASK PERS wobjdata Wobj:=[FALSE,TRUE,"",[[1007.507101727,-
484.383353677,0],[0.991444862,0,0,-0.130526191]],[[-395.816,-1994.361,0],[1,0,0,0]]];
  TASK PERS tooldata KalibPlatta:=[TRUE,[[-175.169,-10.001,625.5],[0,-5.78E-
7,1,0]],[1,[0,0,1],[1,0,0,0],0,0,0]];

  CONST robtarget
rtTCP:=[[763.099944481,2655.761293796,582.66],[0.4539905,0,0,0.891006524],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget
rtKoordinatsystem1:=[[358.98,611.21,582.66],[0.4539905,0,0,0.891006524],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  CONST robtarget
rtKoordinatsystem2:=[[358.98,611.21,582.66],[0.4539905,0,0,0.891006524],[0,-
1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
  PERS robtarget rtTCPX:=[[811.396,2668.7,582.66],[0.45399,0,0,0.891007],[0,-
1,0,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
  PERS robtarget rtTCPY:=[[750.16,2704.06,582.66],[0.45399,0,0,0.891007],[0,-
1,0,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
  PERS robtarget rtTCPTwist:=[[763.1,2655.76,582.66],[0.309017,0,0,-0.951057],[0,-1,-
1,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];

  VAR num Xoffset:=50;
  VAR num Yoffset:=50;
  VAR num Rotationdegrees:=90;
  VAR num KameraPosX;
  VAR num KameraPosY;
  VAR num XAxelKameraPosX;
  VAR num XAxelKameraPosY;
  VAR num YAxelKameraPosX;
  VAR num YAxelKameraPosY;
  VAR num KameraPosTwistX;
  VAR num KameraPosTwistY;



  VAR num VinkelV;
  VAR num MagnitudA;
  VAR num SkalarC;

  VAR pos VektorA;
  VAR pos EnhetA;

  VAR pos RoteradEnhet;
```

```
!Skapa matriser utav kvaternioner

VAR num Invkvart1;
VAR num Invkvart2;
VAR num Invkvart3;
VAR num Invkvart4;

VAR pos kvartRotMatrisRad1;
VAR pos kvartRotMatrisRad2;
VAR pos kvartRotMatrisRad3;


VAR pos kvartRotMatrisVektorARad1;
VAR pos kvartRotMatrisVektorARad2;
VAR pos kvartRotMatrisVektorARad3;

VAR num TCPInverKvat1;
VAR num TCPInverKvat2;
VAR num TCPInverKvat3;
VAR num TCPInverKvat4;

VAR pos TempEnhetVektor;

!Vridkoordinatsystem
VAR pos VektorKamera1;
VAR pos VektorKamera2;
VAR pos VektorRobot1;
VAR pos VektorRobot2;

VAR num MagnKamera1;
VAR num MagnKamera2;
VAR num MagnRobot1;
VAR num MagnRobot2;

VAR pos K1;
VAR pos K2;
VAR pos R1;
VAR pos R2;

VAR pos Kameramatris1;
VAR pos Kameramatris2;

!Tillfällig
Var pos TempvektorA;

!För beräkning av skalfaktor
VAR pos Faktormatris1;
VAR pos Faktormatris2;
VAR pos VektorX;
VAR pos VektorY;
```

```
VAR pos SkalfaktorX;
VAR pos SkalfaktorY;

VAR num XkameraX;
VAR num XkameraY;
VAR num YkameraX;
VAR num YkameraY;

!Itterativ kontroll
VAR num RoteradEnhetMagn;


PROC main()

  rtTCPX:=Offset(rtTCP,Xoffset,0,0);
  rtTCPY:=Offset(rtTCP,0,Yoffset,0);
  rtTCPTwist:=RelTool(rtTCP,0,0,0\Rz:=Rotationdegrees);
  rtTCPTwist.robconf.cf6:=rtTCPTwist.robconf.cf6-1;

  !Beräknar skalfaktor från förflyttning i robot till förflyttning i kamera
  MoveJ rtTCP,v200,fine,KalibPlatta\Wobj:=Wobj;
  WaitRob\InPos;
  waittime(1);
  kameraPosX:=AInput(AICamPosi);
  kameraPosY:=AInput(AICamPosj);
  MoveJ rtTCPX,v200,fine,KalibPlatta\WObj:=Wobj;
  WaitRob\InPos;
  waittime(1);
  XAxelKameraPosX:=AInput(AICamPosi);
  XAxelKameraPosY:=AInput(AICamPosj);
  VektorRobot1:=rtTCPX.trans-rtTCP.trans;

  MoveJ rtTCPY,v200,fine,KalibPlatta\WObj:=Wobj;
  WaitRob\InPos;
  waittime(1);
  YAxelKameraPosX:=AInput(AICamPosi);
  YAxelKameraPosY:=AInput(AICamPosj);
  VektorRobot2:=rtTCPY.trans-rtTCP.trans;


  IF (rtTCPx.trans.x-rtTCP.trans.x>1 OR rtTCPx.trans.x-rtTCP.trans.x<(-1)) AND
(XAxelKameraPosX-kameraPosX>1 or XAxelKameraPosX-kameraPosX<(-1)) THEN
    SkalfaktorX.x:=1/((XAxelKameraPosX-kameraPosX)/(rtTCPx.trans.x-rtTCP.trans.x));
  ELSE
    SkalfaktorX.x:=0;
  ENDIF
  IF (rtTCPx.trans.y-rtTCP.trans.y>1 OR rtTCPx.trans.y-rtTCP.trans.y<(-1)) AND
(XAxelKameraPosY-kameraPosY>1 or XAxelKameraPosY-kameraPosY<(-1)) THEN
    SkalfaktorX.y:=1/((XAxelKameraPosY-kameraPosY)/(rtTCPx.trans.y-rtTCP.trans.y));
  ELSE
```

```
    SkalfaktorX.y:=0;
  ENDIF
  IF (rtTCPy.trans.x-rtTCP.trans.x>1 OR rtTCPy.trans.x-rtTCP.trans.x<(-1)) AND
(YAxelKameraPosX-kameraPosX>1 or YAxelKameraPosX-kameraPosX<(-1)) THEN
    SkalfaktorY.x:=1/((YAxelKameraPosX-kameraPosX)/(rtTCPy.trans.x-rtTCP.trans.x));
  ELSE
    SkalfaktorY.x:=0;
  ENDIF
  IF (rtTCPy.trans.y-rtTCP.trans.y>1 OR rtTCPy.trans.y-rtTCP.trans.y<(-1)) AND
(YAxelKameraPosY-kameraPosY>1 or YAxelKameraPosY-kameraPosY<(-1)) THEN
    SkalfaktorY.y:=1/((YAxelKameraPosY-kameraPosY)/(rtTCPy.trans.y-rtTCP.trans.y));
  ELSE
    SkalfaktorY.y:=0;
  ENDIF
  VektorKamera1.x:=(XAxelKameraPosX-kameraPosX)*SkalfaktorX.x;
  VektorKamera1.y:=(XAxelKameraPosY-kameraPosY)*SkalfaktorY.y;
  VektorKamera2.x:=(YAxelKameraPosX-kameraPosX)*SkalfaktorX.x;
  VektorKamera2.y:=(YAxelKameraPosY-kameraPosY)*SkalfaktorY.y;

  !Omvandlar koordinatvektorerna till enhetsvektorer

  !K1
  MagnKamera1:=VectMagn(VektorKamera1);
  K1:=[VektorKamera1.x/MagnKamera1,VektorKamera1.y/MagnKamera1,0];

  !K2
  MagnKamera2:=VectMagn(VektorKamera2);
  K2:=[VektorKamera2.x/MagnKamera2,VektorKamera2.y/MagnKamera2,0];

  !R1
  MagnRobot1:=VectMagn(VektorRobot1);
  R1:=[VektorRobot1.x/MagnRobot1,VektorRobot1.y/MagnRobot1,0];

  !R2
  MagnRobot2:=VectMagn(VektorRobot2);
  R2:=[VektorRobot2.x/MagnRobot2,VektorRobot2.y/MagnRobot2,0];

  !Vrider koordinatsystem från workobject till kamera
  Faktormatris1:=[(R1.x*k2.y+r2.x*(-k1.y)),(r1.x*(-k2.x)+r2.x*k1.x),0];
  Faktormatris2:=[(r1.y*k2.y+r2.y*(-k1.y)),(r1.y*(-k2.x)+r2.y*k1.x),0];
  !Vrider koordinatsystem från kamera till workobject
  Kameramatris1:=[K1.x*R2.y+K2.x*R2.x,(K1.x*R1.y+K2.x*R1.x),0];
  Kameramatris2:=[(K1.y*R2.y+K2.y*R2.x),K1.y*R1.y+K2.y*R1.x,0];

  !Vektorer för förflyttning i kamerans X- samt Y-led bill
  VektorX:=[Xoffset*Faktormatris1.x,Xoffset*Faktormatris2.x,0];
  VektorY:=[Yoffset*Faktormatris1.y,Yoffset*Faktormatris2.y,0];

  !       !offsets target i kamerans riktrningar bill
  rtTCPX:=Offset(rtTCP,VektorX.x,VektorX.y,0);
```

```
rtTCPY:=Offset(rtTCP,VektorY.x,VektorY.y,0);


!Nya skalfaktorer
MoveJ rtTCP,v200,fine,KalibPlatta\WObj:=Wobj;
MoveJ rtTCPX,v200,fine,KalibPlatta\WObj:=Wobj;
WaitRob\InPos;
waittime(1);
XkameraX:=AInput(AICamPosi);
XkameraY:=AInput(AICamPosj);
MoveJ rtTCP,v200,fine,KalibPlatta\WObj:=Wobj;
MoveJ rtTCPY,v200,fine,KalibPlatta\WObj:=Wobj;
WaitRob\InPos;
waittime(1);
YkameraX:=AInput(AICamPosi);
YkameraY:=AInput(AICamPosj);


IF XKameraX-KameraPosX>1 OR XKameraX-KameraPosX<(-1) THEN
  SkalfaktorX.x:=Xoffset/(XkameraX-KameraPosX);
ELSE
  SkalfaktorX.x:=0;
ENDIF
IF XKameraY-KameraPosY>1 OR XKameraY-KameraPosY<(-1) THEN
  SkalfaktorX.y:=Xoffset/(XkameraY-KameraPosY);
ELSE
  SkalfaktorX.y:=0;
ENDIF
IF YKameraX-KameraPosX>1 OR YKameraX-KameraPosX<(-1) THEN
  SkalfaktorY.x:=Yoffset/(YkameraX-KameraPosX);
ELSE
  SkalfaktorY.x:=0;
ENDIF
IF YKameraY-KameraPosY>1 OR YKameraY-KameraPosY<(-1) THEN
  SkalfaktorY.y:=Yoffset/(YkameraY-KameraPosY);
ELSE
  SkalfaktorY.y:=0;
ENDIF


WHILE TRUE DO


  !utvärdering av TCP
  MoveJ rtTCP,v200,fine,KalibPlatta\WObj:=Wobj;
  WaitRob\InPos;
  WaitTime(1);
  kameraPosX:=AInput(AICamPosi);
```

```
kameraPosY:=AInput(AICamPosj);

MoveJ rtTCPTwist,v200,fine,KalibPlatta\Wobj:=Wobj;
WaitRob\InPos;
WaitTime(1);
kameraPosTwistX:=AInput(AICamPosi);
kameraPosTwistY:=AInput(AICamPosj);

!Beräknar vektor A

VektorA.x:=(kameraPosTwistX-kameraPosX)*SkalfaktorX.x;
VektorA.y:=(kameraPosTwistY-kameraPosY)*SkalfaktorY.y;


TempvektorA:=VektorA;

VektorA.x:=kameramatris1.x*TempvektorA.x+Kameramatris2.x*TempvektorA.y;
VektorA.y:=kameramatris1.y*TempvektorA.x+Kameramatris2.y*TempvektorA.y;




MagnitudA:=VectMagn(VektorA);

IF MagnitudA<1 THEN
  GOTO slut;
ENDIF

EnhetA.x:=VektorA.x/MagnitudA;
EnhetA.y:=VektorA.y/MagnitudA;

SkalarC:=(MagnitudA/2)/Sin(Rotationdegrees/2);

VinkelV:=-((180-Rotationdegrees)/2);


RoteradEnhet.x:=-((EnhetA.x*(Cos(-VinkelV))-EnhetA.y*(Sin(-VinkelV)))*SkalarC);

RoteradEnhet.y:=-((EnhetA.x*(sin(-VinkelV))+EnhetA.y*(Cos(-VinkelV)))*SkalarC);


!Tar hänsyn till att RobTarget är annan orientering än Wobj
TCPInverKvat1:=rtTCP.rot.q1;
TCPInverKvat2:=-rtTCP.rot.q2;
TCPInverKvat3:=-rtTCP.rot.q3;
TCPInverKvat4:=-rtTCP.rot.q4;

kvartRotMatrisVektorARad1:=[1-2*(TCPInverKvat3*TCPInverKvat3)-
2*(TCPInverKvat4*TCPInverKvat4),2*TCPInverKvat2*TCPInverKvat3-
```

```
2*TCPInverKvat4*TCPInverKvat1,2*TCPInverKvat2*TCPInverKvat4+2*TCPInverKvat3*
TCPInverKvat1];

kvartRotMatrisVektorARad2:=[2*TCPInverKvat2*TCPInverKvat3+2*TCPInverKvat4*TCPI
nverKvat1,1-2*(TCPInverKvat2*TCPInverKvat2)-
2*(TCPInverKvat4*TCPInverKvat4),2*TCPInverKvat3*TCPInverKvat4-
2*TCPInverKvat2*TCPInverKvat1];
    kvartRotMatrisVektorARad3:=[2*TCPInverKvat2*TCPInverKvat4-
2*TCPInverKvat3*TCPInverKvat1,2*TCPInverKvat3*TCPInverKvat4+2*TCPInverKvat2*
TCPInverKvat1,1-2*(TCPInverKvat2*TCPInverKvat2)-
2*(TCPInverKvat3*TCPInverKvat3)];



    TempEnhetVektor:=RoteradEnhet;


RoteradEnhet.x:=kvartRotMatrisVektorARad1.x*TempEnhetVektor.x+kvartRotMatrisVektor
ARad1.y*TempEnhetVektor.y+kvartRotMatrisVektorARad1.z*TempEnhetVektor.z;

RoteradEnhet.Y:=kvartRotMatrisVektorARad2.x*TempEnhetVektor.x+kvartRotMatrisVekto
rARad2.y*TempEnhetVektor.y+kvartRotMatrisVektorARad2.z*TempEnhetVektor.z;

RoteradEnhet.Z:=kvartRotMatrisVektorARad3.x*TempEnhetVektor.x+kvartRotMatrisVekto
rARad3.y*TempEnhetVektor.y+kvartRotMatrisVektorARad3.z*TempEnhetVektor.z;


    !kvaternioninvers
    Invkvart1:=KalibPlatta.tframe.rot.q1;
    Invkvart2:=-KalibPlatta.tframe.rot.q2;
    Invkvart3:=-KalibPlatta.tframe.rot.q3;
    Invkvart4:=-KalibPlatta.tframe.rot.q4;



    !Tar hänsyn till tool0
    kvartRotMatrisRad1:=[(Invkvart1*Invkvart1)+(Invkvart2*Invkvart2)-
(Invkvart3*Invkvart3)-(Invkvart4*Invkvart4),2*Invkvart2*Invkvart3-
2*Invkvart4*Invkvart1,2*Invkvart2*Invkvart4+2*Invkvart3*Invkvart1];

kvartRotMatrisRad2:=[2*Invkvart2*Invkvart3+2*Invkvart4*Invkvart1,(Invkvart1*Invkvart1)
-(Invkvart2*Invkvart2)+(Invkvart3*Invkvart3)-
(Invkvart4*Invkvart4),2*Invkvart3*Invkvart4-2*Invkvart2*Invkvart1];
    kvartRotMatrisRad3:=[2*Invkvart2*Invkvart4-
2*Invkvart3*Invkvart1,2*Invkvart3*Invkvart4+2*Invkvart2*Invkvart1,(Invkvart1*Invkvart1
)-(Invkvart2*Invkvart2)-(Invkvart3*Invkvart3)+(Invkvart4*Invkvart4)];

    TempEnhetVektor:=RoteradEnhet;
```

```
RoteradEnhet.x:=kvartrotmatrisrad1.x*TempEnhetVektor.x+kvartrotmatrisrad1.y*TempEnhe
tVektor.y+kvartrotmatrisrad1.z*TempEnhetVektor.z;

RoteradEnhet.Y:=kvartrotmatrisrad2.x*TempEnhetVektor.x+kvartrotmatrisrad2.y*TempEnh
etVektor.y+kvartrotmatrisrad2.z*TempEnhetVektor.z;

RoteradEnhet.Z:=kvartrotmatrisrad3.x*TempEnhetVektor.x+kvartrotmatrisrad3.y*TempEnhe
tVektor.y+kvartrotmatrisrad3.z*TempEnhetVektor.z;

    RoteradEnhetMagn:=VectMagn(RoteradEnhet);
    IF RoteradEnhetMagn<1 THEN
      GOTO slut;
    ENDIF



KalibPlatta.tframe.trans:=[KalibPlatta.tframe.trans.x+RoteradEnhet.x,KalibPlatta.tframe.trans
.y+RoteradEnhet.y,KalibPlatta.tframe.trans.z+RoteradEnhet.z];
   ENDWHILE

   slut:
 ENDPROC
ENDMODULE
```