CODEN:LUTEDX/(TEIE-3134)/1-39(2024)

Industrial Electrical Engineering and Automation

Robotsimulering av ett manuellt produktionssteg

Kevin Rost

Division of Industrial Electrical Engineering and Automation
Faculty of Engineering, Lund University

Robotsimulering av ett manuellt produktionssteg

- Hur mycket ökar takten vid simulering av en automationslösning?

LUNDS UNIVERSITET
Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Avdelningen för Industriell Elektroteknik och Automation

Examensarbete:
Kevin Rost

Handledare: Bernard Schmidt
Kontaktperson på företaget: Helen Forsberg
Examinator: Morten Hemmingsson

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2024

Sammanfattning

I dagsläget är automatisering något som intresserar företag världen över då det möjliggör en snabbare och mer effektiv produktion. Detta examensarbete har gjorts i samarbete med företaget Aimpoint, som tillhör en av de förenämnda företag som är intresserade av automation och hur det kan hjälpa deras produktion. Aimpoint är ett ledande företag inom rödpunktssiktes sektorn och deras grundare, Arne Ekstrand, var även han som uppfann rödpunkssiktet. I dagsläget sker en majoritet av Aimpoints produktion för hand. Företaget anser detta vara en av anledningarna till deras produktkvalitet, men företaget känner även ett behov av att effektivisera sin produktion. Aimpoint söker därför efter ett förslag på en automationslösning som ökar deras effektivitet samtidigt som deras höga produktkvalitet förblir densamma. Detta är det som har utgjort grunden för detta examensarbete. Arbetet började med att välja ut vilket sikte vars produktionsflöde skulle undersökas i mål om att hitta ett steg att automatisera. Denna undersökning resulterade i beslutet att rengöringen av siktets linser i operation 6 i siktet X:s produktionsflöde borde automatiseras. Detta beror på att denna uppgift visade den näst högsta cykeltiden i produktionsflödet efter en operation som bedömes olämplig för automatisering, samt att det är en ansträngande uppgift för montörerna. Med viljan att behålla deras montörer och den kvalité de tillför i åtanke, tog studenten i samråd med handledare, beslutet att en simulering i företaget ABB:s programvara RobotStudio borde tas fram. I denna simulering skulle en cobot programmeras till att genomföra rengöringen som ansågs borde automatiseras. Genom att ta fram en simulering kunde det undersökas hur en implementering skulle påverka produktionsflödet och operationens cykeltid. Simuleringen använde sig av ABB:s cobot SWIFTI CRB 1100. Innan arbetet i RobotStudio kunde påbörjas gjordes det en noggrann analys av hur rengöringen går till och vad som är viktigt att ha i åtanke vid framtagning av simuleringen. Efter detta togs det fram en simulering i RobotStudio där SWIFTI genomförde rengöringsstegen. För att undersöka hur en implementering av simuleringen skulle påverka operationens cykeltid gjordes det tidmätningar av två simuleringar, en tidmätning där ett sikte rengjordes och en simulering där tre sikten rengjordes. Tidmätningen från rengöring av ett sikte resulterade i en simulationstid på 33.36 sekunder och rengöring av tre sikten resulterade i en uppmätt tid på 103.632 sekunder. Genom att sätta in den simulerade tiden för rengöring av ett sikte i ett diagram som representerar tider uppmätta i operation 6 kunde det ses hur cykeltiden hade förändrats. Vid effektiv implementering av lösningen blev resultat en minskning av cykeltiden från 178 sekunder till 117 sekunder, vilket innebär en minskning på 61 sekunder. I det fall där operationen skulle ändras till att hantera tre sikten per omgång blir den resulterande tiden i nuläget 535 sekunder, respektive 362 sekunder vid effektiv implementering av simuleringen. Detta ger en differens på 172 sekunder. Genom att använda den nya cykeltiden och vid optimering av den operation som agerar flaskhals i nuläget ökade takten och minskade snittet för cykeltiden för den delen av produktionen som undersöktes från 214,3 sekunder till 157,6 sekunder. Examensarbetet uppnådde därmed sitt mål genom att det togs fram en automationslösning som minskar cykeltiden för operation samt ökar takten.

Nyckelord: Cobot, RobotStudio, automation, produktionsflöde, effektivisering.

Abstract

Currently, automation is of interest to companies worldwide as it enables faster and more efficient production. This thesis has been conducted in collaboration with the company Aimpoint, which is among the aforementioned companies interested in automation and how it can benefit their production. Aimpoints is a leading company in the red dot sight sector, and its founder, Arne Ekstrand, was also the inventor of the red dot sight. Currently, the majority of Aimpoint's production is done manually. The company considers this to be one of the reasons for their product quality, but they also feel the need to streamline their production. Therefore, Aimpoint is seeking a proposal for an automation solution that increases their efficiency while maintaining their high product quality. This has formed the basis for this thesis. The work began by selectin which sight's production flow would be investigated to find a step to automate. This investigation resulted in the decision that the cleaning of the sight's lenses in operation 6 in the production flow of the X sight should be automated. This is because this task showed the second highest cycle time in the production flow after an operation that was deemed unfit for automation, as well as being a strenuous task for the workers. With the desire to retain their workers and the quality they contribute in mind, the student, in consultation with the supervisor, decided to develop a simulation in the ABB company's software RobotStudio. In this simulation, a cobot would be programmed to perform the cleaning that was deemed to be automated. By creating a simulation, it could be examined how an implementation would affect the production flow and the cycle time of the operation. The simulation used ABB's cobot SWIFTI CRB 1100. Before the work in RobotStudio could being, a thorough analysis was mode of how the cleaning process works and what is important to consider when developing the simulation. After this, a simulation was developed in RobotStudio where SWIFTI performed the cleaning steps. To investigate how an implementation of the simulation would affect the cycle time of the operation, time measurements were made for two simulations: one time measurement where one sight was cleaned and one simulation where three sights were cleaned. The time measurement for cleaning one sight resulted in a simulation time of 33,36 seconds, and cleaning three sights resulted in a measured time of 103,632 seconds. By inserting the simulated time for cleaning one sight into a diagram representing times measured in the operation 6, it could be seen how the cycle time would change. With effective implementation of the solution, the result was a reduction in cycle time from 178 seconds to 117 seconds, which represents a decrease of 61 seconds. In the case where the operation would be changed to handle three sights per round, the resulting time currently is 535 seconds, and 362 seconds with the effective implementation of the simulation. This gives a difference of 172 seconds. By using the new cycle time and optimizing the operation currently acting as a bottleneck, the pace production was increased as the average cycletime for the part of the production that was examined was reduced from 214,3 seconds to 157,6 seconds. The thesis thus achieved its goal by developing an automation solution that reduces the cycle time of the operation and reduces the takt time.

Keywords: Cobot, RobotStudio, automation, production flow, streamlining.

Förord

Detta examensarbete har genomförts under våren 2024 och är det sista momentet i min utbildning till Högskoleingenjör inom elektro- & automationsteknik på Lunds Tekniska Högskola.

Jag skulle vilja tacka Aimpoint för möjligheten att genomföra detta spännande och givande arbete hos dem. Jag även vilja ge ett speciellt tack till min kontaktperson på företaget, Helen Forsberg som varit väldigt hjälpsam under detta arbete, samt min handledare Bernard Schmidt som varit hjälpsam och givit bra tips.

Terminologi

OP – Operation

TCP – Tool Center Point är ett begrepp inom RobotStudio som refererar till den punkt som det aktiva verktyget är centrerat kring och förhåller sig till.

PLM – Product Lifecycle Management är en strategi kring hantering av en produkt genom hela dess livscykel.

Cykeltid – Tiden för en behandling i en produktionsprocess.

Innehållsförteckning

1. Inledning

I inledningen kommer bakgrund, syfte, målformulering, problemformulering, motivering av examensarbete och avgränsningar för examensarbetet presenteras.

1.1 Bakgrund

Aimpoint är ett företag som producerar och säljer rödpunktssikten till både den privata marknaden, såsom jägare, och den professionella marknaden, såsom polis och militär. Aimpoint har sitt huvudkontor i Malmö och säljer deras sikten världen över. Aimpoint är även känt som skaparna av rödpunktsiktet, vilket är ett optiskt sikte som projicerar en röd punkt på målet för att underlätta snabb och precis siktning. Företagets grundare, Arne Ekstrand, är personen som kom på idén för rödpunktsiktet år 1974. Ekstrand ville förbättra sin träffsäkerhet när han jagade och framför allt få en bättre träffsäkerhet på just rörliga mål. Ekstrand kom då på rödpunktsiktet och tog idén vidare till Gunnar Sandberg från företaget ElektroSandberg. Sandberg tyckte om idén och investerade i konceptet för att fortsätta utvecklingen av idén [1].

I dagsläget har Aimpoint samma mål som grundaren Ekstrand hade när han tog fram konceptet, vilket är att fortsätta utveckla sikten och andra produkter för att underlätta skytte för företagets användare. Aimpoint arbetar utifrån sina tre kärnvärden: prestanda, engagemang och innovation. Målet är att skapa de mest exakta och högteknologiska produkter möjligt för att ge kunden en så bra träffsäkerhet som möjligt [2].

Idag är Aimpoints produktion huvudsakligen manuell, en strategi de anser bidrar till den höga kvalitet de levererar. Dock strävar Aimpoint efter att förbättra sin produktionsprocess och samtidigt minska den negativa påverkan på montörernas välbefinnande genom att underlätta moment som kan leda till slitskador. I sin strävan att hitta lösningar som förbättrar och effektiviserar produktionen vill Aimpoint samtidigt bevara produkternas kvalitet och hjälpa montörerna. Tidigare analyser och mätningar av produktionsflödet har hjälpt företaget identifiera flaskhalsar och områden för förbättring. Med dessa insikter som grund har Aimpoint satt upp ett mål att öka deras nuvarande takt, vilket skulle innebära kortare produktionscykler och ökad effektivitet vilket detta arbete syftar till att bidra till.

I sökandet på sätt att effektivisera och förbättra produktionen har Aimpoint på senare tid blivit mer intresserade av automationsteknik och hur det kan hjälpa dem att uppnå sitt mål med att öka produktionstakten. Aimpoint anser också att automationsteknik kan vara till fördel för montörerna genom att automatisera moment i produktionsprocessen som de uppfattar fysiskt krävande.

1.2 Syfte

Syftet med examensarbetet är att gå igenom tidigare material från undersökningar av produktionsflödet som har gjorts och identifiera ett moment i ett siktes produktion som har stor förbättringsmöjlighet och som skulle nyttjas av att automatiseras. Valet av vilket moment som ska automatiseras bör också göras med hänsyn till att en sådan automatisering kan bidra till att minska slitage för montörerna. Efter att ett moment valts ut ska det tas fram en lösning

för hur en automatisering av denna uppgift i produktionen kan se ut och hur mycket lösningen bidrar till att minska cykeltiden och öka takten.

1.3 Målformulering

Målet med examensarbetet är att automatisera en uppgift i produktionen för Aimpoints sikte X och på så sätt bidra till det pågående arbetet med att öka takten. Automatiseringen av denna uppgift ska även eliminera ett slitagemoment för montörerna.

1.4 Problemformulering

Under genomförandet av examensarbetet kommer följande att bli besvarat:

1. Vilken uppgift i siktets produktion borde automatiseras?
2. Vilken robot bör användas?
3. Vad blir förändringen av cykeltiden för momentet vid automatisering?
4. Vad blir förändringen av takten vid automatisering?
5. Vad är den ungefärliga kostnaden vid implementering av automationslösningen?

1.5 Motivering av examensarbete

Arbetet handlar om automationsteknik, mer specifikt automatisering av ett produktionssteg vilket är något som är direkt kopplad till studentens utbildning och möjliga tjänster i framtiden. Ett examensarbete inriktat på automation är något som även anses vara väldigt hjälpsamt för vidare utveckling av de kunskaper som ackumulerats under utbildningens gång. Aimpoints ambitioner och vilja att utveckla sig inom automationsteknik har också haft betydelse i beslutet att genomföra just detta examensarbete.

1.6 Avgränsningar

I detta arbete kommer det enbart tas fram förslag på automatisering för ett moment i ett siktes produktionsflöde. Arbetet kommer inte att ta fram simuleringar med fler olika robotar.

1.7 Resurser

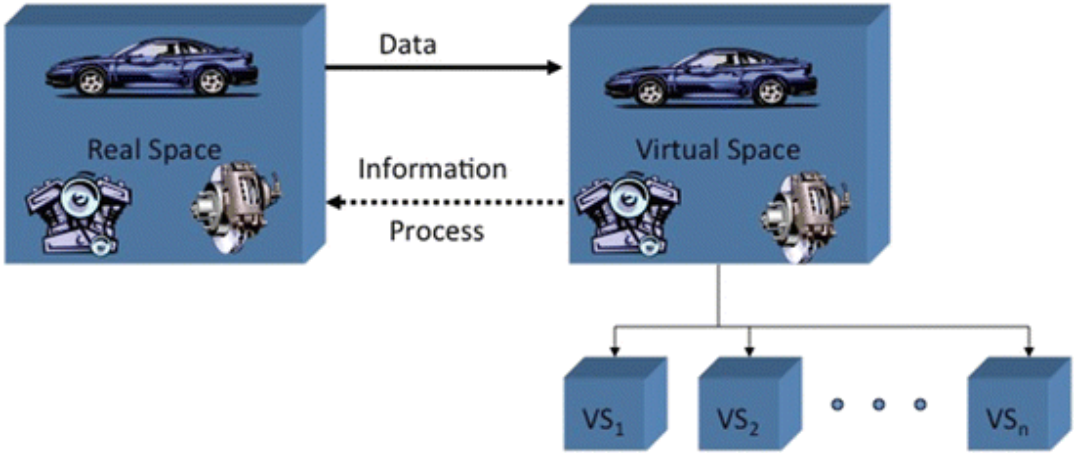
Acer Aspire A715-42G (Dator)
Microsoft Windows 11 (Operativsystem)
RobotStudio 2024 (Mjukvara)
Tillgång till Aimpoints produktion

2 Teknisk bakgrund

I detta kapitel kommer tekniken som har använts under examensarbete att presenteras för att få en bättre förståelse för resten av arbetet.

2.1 Digital Tvilling

Digital tvilling är ett koncept som introducerades år 2002 av Michael Grieves vid en presentation av hans arbete med produktlivscykelhantering. Grieves introducerade däremot konceptet först som ”Conceptual Ideal for PLM [3].” vilket innehöll alla beståndsdelar för vad som sedan skulle bli känt som Digital Tvilling. Dessa beståndsdelar är reell rymd, virtuell rymd, länken för datatrafik från reell rymd till virtuell rymd, länken för informationsflöde från virtuell rymd till reell rymd och virtuella delrymder, se figur 1 för visuell representation [3]. De virtuella delrymderna, VS i figur 1, är olika virtuella scenarier eller versioner av den virtuella miljön, vilket används för att göra olika tester och simuleringar. Det skulle inte komma att officiellt kallas för Digital Tvilling förrän år 2011 då begreppet, tidigare myntat av John Vickers användes av Grieves själv [4].



Figur 1. Bild som användes av Grieves vid första introduceringen av konceptet för att visa samspelet mellan beståndsdelarna [3].

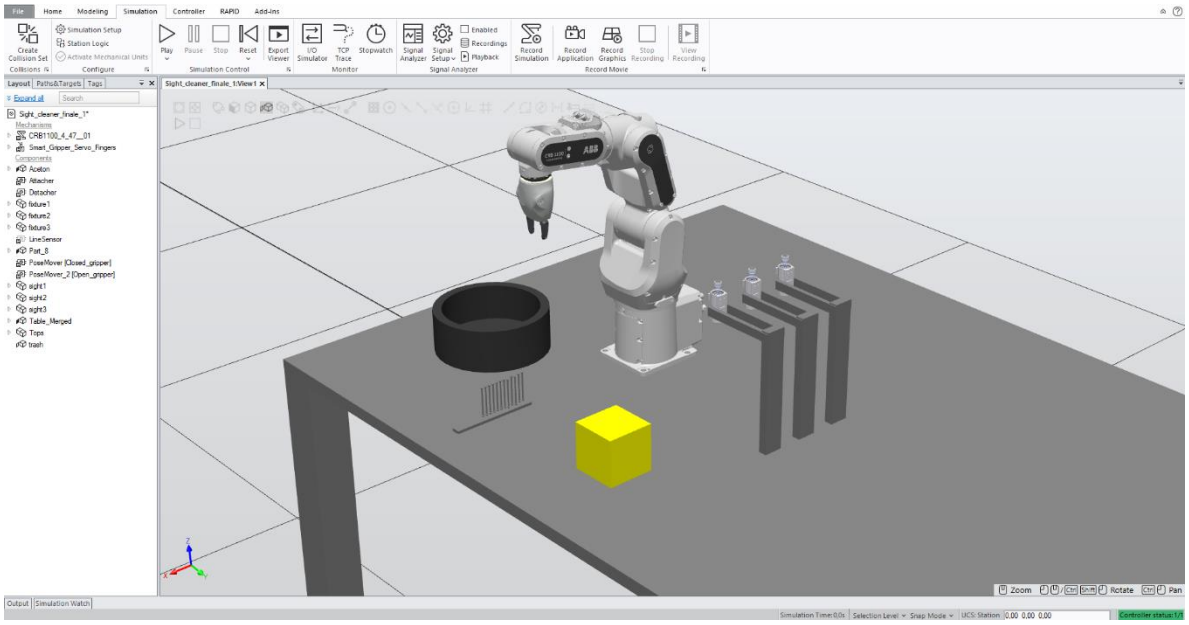
Grieves strävade efter att övergå från den traditionella, pappersbaserade och manuella hanteringen av produktdata till att istället skapa en digital modell av produkten, baserad på all tillgänglig produktinformation. Hans vision involverade ett övergripande system med två parallella system: ett digitalt system, som innehöll samma omfattande information som det fysiska systemet, samt det fysiska systemet själv. Genom denna struktur skapades en digital spegling av den verkliga världen, där all produktinformation kunde sammanföras och användas på ett effektivt sätt. Motiveringen bakom att PLM var med i konceptets första beskrivning var för att beskriva att länken mellan det digitala och det fysiska systemet inte var statiskt, utan snarare en aktiv länk som finns under hela systemets livscykel. Denna livscykel består av fyra olika faser: skapelse, tillverkning, drift och avskaffande [3].

En komplett Digital Tvilling är en kombination av virtuella informationskonstruktioner som speglar nästintill exakt hur det virtuella hade sett ut och betett sig i verkligheten, vare sig det är en process eller en produkt. Där finns olika typer av digitala tvillingar beroende på vilken fas det övergripande systemet befinner sig i. De olika typerna är Digital Twin Prototype (DTP), Digital Twin Instance (DTI) och Digital Twin Aggregate (DTA). DTP är en designversion av hur en Digital Tvilling av ett visst system kan se ut. En DTP används för

att demonstrera hur data från det fysiska objektet eller systemet kan sammanställas, analyseras och utnyttjas i den slutliga digitala tvillingen. DTI är den digitala tvillingen av varje enskilt producerad produkt eller del och används enbart där det är nödvändigt att inneha information om produkten eller delen under hela dess livscykel. Exempelvis, finns det DTIs av flygplansdelar och raketdelar men inte gem då det inte är en nödvändighet. DTAs är en sammansättning av flera DTIs, vilket är användbart eftersom att det ger möjligheten att exempelvis utsätta en DTA för olika stresstest och på så vis få reda på vilka DTIs, det vill säga komponenter, som slutar fungera vid olika simuleringar. Flygplansdelar kan exempelvis alla representeras av DTIs som sedan sätts samman i en virtuell yta för att bilda en DTA [3],[4].

2.2 RobotStudio

RobotStudio är en simulerings applikation utvecklad av ABB för modellering, programmering och simulering av deras robotar. Applikation har etablerat sig som ett kraftfullt verktyg för ingenjörer och forskare inom robotteknik och automatisering. Programmets kärnuppgift är att möjliggöra ett effektivt sätt att designa, simulera och testa robotapplikationer innan de implementeras i en verklig arbetsmiljö, se figur 2 för en exempelskärmbild av RobotStudios simuleringsmiljö [5]. Den version av RobotStudio som används vid detta arbete är RobotStudio 2024.1.



Figur 2. RobotStudios simuleringsmiljö.

En av de betydelsefulla funktioner som erbjuds i RobotStudio är möjligheten att arbeta med virtuella kontroller, ofta benämnda som ”VC”. En VC möjliggör så kallad offline programmering för programmets användare, vilket innebär att mjukvaran är ansluten till en virtuell kontroll som simulerar funktionen hos en verklig robotkontroller, men på en dator. Om roboten ansluts till en verklig och fysisk robotkontroller kallas detta i stället för online programmering. Därmed kan användaren finjustera och optimera sina robotprogram i en virtuell miljö med hjälp av en VC innan de överförs till en verklig produktionslinje [5].

FlexPendant är en handhållen operatörsenhet som används för att kontrollera roboten i verkligheten. En FlexPendant gör det möjligt att interagera med systemet för att exempelvis felsöka koden, manuellt ändra robotens leder eller ändra specifika instruktioner i programmet, exempelvis genom att uppdatera punkters koordinater och orientering. En punkt, så kallat *robtarget*, i RobotStudio är en punkt som roboten kan röra sig till där flera punkter tillsammans skapar en väg roboten rör sig igenom. I RobotStudio finns en virtuell version av en FlexPendant som går att använda för att programmera eller justera roboten i simuleringsmiljön på samma sätt som det går att göra i verkligheten. En FlexPendant gör det enkelt att köra roboten i en av två lägen, automatiskt eller manuellt läge. Vid automatiskt läge kör roboten igenom programmet utan behov av en operatör som styr. Läget används när programmet är i sitt slutskede och ska testas. Manuellt läge används för att köra igenom ett program steg för steg, detta läge kräver en operatör för att fungera och används främst vid kontroll av program och felsökning. [5]

RobotStudio använder sig av ett programmeringsspråk som kallas RAPID (Robot Application Programming Interface Data), som är ett språk utvecklat av ABB själva och används för att programmera deras robotar. RAPID använder sig av en modulär struktur, vilket innebär att RAPID-program är uppbyggda i *moduler* bestående av flera *processer*. RAPID är ett så kallat instruktionsbaserat språk där varje process och dess kodrader representerar en specifik instruktion för roboten att utföra [5].

Viktiga variabeltyper och instruktioner i RAPID som behöver förklaras för att förstå senare RAPID-kodexempel:

TCP: Tool Center Point refererar till den punkt som verktyget är centrerat kring och förhåller sig till.

tooldata: En variabel av datatypen tooldata används för att representera ett verktyg. Tooldata innehåller information om verktygets egenskaper, såsom position och orientering av TCP, samt de fysiska egenskaperna hos verktygsbelastningen.

robtarget: Ett mål för roboten, även kallat target, översätts till RAPID-datatypen robtarget under RAPID-synkroniseringen. Ett mål är en punkt i 3D-miljöns rymd som definierar den position och orientering som TCP ska nå.

speeddata: En variabel i RAPID som bestämmer hur snabbt roboten ska röra sig till en viss robtarget.

zonedata: En variabel i RAPID som definierar hur nära roboten ska vara en viss punkt för att anses ha nått sagd punkt.

MoveL: Används för att specificera en linjär rörelse för roboten till ett robtarget i miljön. Det innebär att roboten rör sig längs en rak linje mellan sin utgångspunkt och satt robtarget.

MoveJ: Används för att specificera en rörelse för robotens leder, även kallade joints, för att nå ett robtarget. Det innebär att varje led på roboten rör sig till en specifik position samtidigt för att nå målet. MoveJ är användbart när det krävs flexibilitet i robotens rörelsebana och när det är viktigt att undvika kollisioner eller oönskade hinder längs vägen.

MoveC: Används för att specificera en cirkulär rörelse för roboten genom att definiera en cirkelbåge i miljön. Det innebär att roboten följer en halvcirkel mellan två robtarget. Denna instruktion måste därför göras två gånger för att bilda en hel cirkel.

Offs: Returnerar ett robtarget förskjuten enligt angivna värden. Kan exempelvis användas för att förskjuta ett robtarget 3 mm i positiv x-riktning.

PulseDo: En instruktion som genererar en angiven typ av puls på en specificerad utgång. En puls kan exempelvis användas för att starta eller stoppa processer eller ändra enhetens tillstånd.

WaitTime: En instruktion som pausar programmet en angiven tid.

2.3 Cobots

Samarbetsrobotar, eller numera kända som ”Cobots” eller ”Collaborative robots”, är en typ av industriell robot som har blivit alltmer populär inom produktions- och tillverkningsmiljöer på senare tid. De skiljer sig från traditionella robotar genom sin förmåga att arbeta nära människor i samma arbetsutrymme i stället för att befinna sig i en designerad yta avskärmad med exempelvis galler. Denna närhet kräver att cobots är säkra att använda, vilket är något som har drivit utvecklingen av flera teknologier och säkerhetsfunktioner som är specifika för denna robotkategori [6].

En viktig faktor som gör cobots säkrare att arbeta nära människor är användningen av sensorer och avancerade säkerhetssystem. Dessa system är utformade för att upptäcka närvaron av människor i närheten av roboten och snabbt anpassa dess rörelse eller stoppa dem helt om det behövs, för att undvika kollisioner och skador. Till exempel kan cobots vara utrustade med trycksensorer som känner av när de kommer i kontakt med en människokropp och stoppar då omedelbart sin rörelse för att förhindra skador. Utöver sensorer kan cobots även använda sig av tekniker som ”force control” för att övervaka och anpassa kraften de applicerar i sina rörelser. Detta gör det möjligt för dem att upptäcka när de möter motstånd och justera sin kraft utifrån det [6].

Cobots är även designade för att vara användarvänliga och kunna kontrolleras av användare som inte har specialiserad utbildning i exempelvis robotik eller programmering. Detta uppnås genom att många modeller har intuitiva och enkla programmeringsgränssnitt med kontroller som är enkla att förstå och hantera. Då cobots är utformade för att samarbeta med människor och fungera sida vid sida i samma arbetsutrymme har denna typ av robot blivit väldigt attraktiv för företag som vill förstärka sin produktion men som inte vill ersätta sina arbetare [6].

2.3.1 SWIFTI CRB 1100

CRB 1100 SWIFTI är en snabb och noggrann sexaxlad cobot från ABB som erbjuder alla funktioner som väntas av en cobot och mer, se figur 3 för exempelbild på roboten. Genom sin avancerade mekaniska konstruktion och högpresterande dynamik erbjuder denna robot en imponerande kombination av hastighet och precision för att möta kraven i moderna produktionsmiljöer. SWIFTI är utrustad med sofistikerade kontrollsystem och intelligenta rörelsealgoritmer som gör det möjligt att utföra komplexa uppgifter.

Figur 3. ABB:s SWIFTI CRB 1100.

SWIFTI använder en kombination av ABB:s säkerhetsfunktion SafeMove med en säkerhets laserscanner. Detta låter SWIFTI detektera om en arbetare är innanför robotens arbetszon så att roboten saktar ner eller stannar helt för att förhindra en olycka. SWIFTI har även en interaktionslampa som signalerar om en arbetare rör sig innanför robotens arbetszon [7].

SWIFTI CRB 1100 använder OmniCore C30 som kontroller vilket tillsammans med en FlexPendant gör det enkelt att konfigurera roboten, även för utan någon erfarenhet i robotik. SWIFTI väger 21 kg och har en nyttolast på 4 kg med en räckvidd på antingen 47,5 cm eller 58 cm, beroende på vilket alternativ som väljs vid inköp, och har en max TCP hastighet på 5.05 meter/sekund [7].

2.4 Flaskhalsanalys

Flaskhals är ett begrepp som används inom produktion. Begreppet används för att beskriva steg i en produktionslina som har begränsad kapacitet i jämförelse med andra steg i processen. Flaskhalsar minskar det totala flödet i en produktion då de steg som agerar flaskhals inte kan hantera den inkommande arbetsmängden tillräckligt bra, vilket leder till uppbyggnad av objekt som ska vidare i linan vid steget som inte klarar trycket. Flaskhalsar kan även orsaka en större takttid än önskat då stegen efter flaskhalsen måste vänta innan de kan påbörjas, vilket orsakar dötid i produktionen och ineffektivt tidsbruk. Det är därför viktigt för en effektiv produktion att eliminera flaskhalsar. Detta kan göras genom att använda flaskhals analys som är en metod menad att hjälpa till med elimineringen av flaskhalsar i produktioner. Flaskhalsanalysens nyckelpunkter kan sammanfattas som:

- Identifiering: Första steget i att eliminera flaskhalsar är att identifiera var de befinner sig. Detta görs genom att analysera ett produktionsflöde med målet att identifiera var produktionen stannar upp och var vissa steg måste vänta för länge på ett annat.

- Mätning: När en flaskhals identifierats ska det sedan genomföras mätningar av produktionsflödet. Data från mätningarna ska omfatta sådant som cykeltid och kapacitet.

- Analys: Nästa steg är att analysera orsakerna till att flaskhalsen uppstår. Ta reda på vad i operationen det är som gör att steget tar den tid det gör.

- Lösningar och optimering: Med resultaten från analysen i åtanke ska det nu tas fram lösningsförslag som optimerar och effektiviserar operationen som agerar flaskhals. Dessa lösningar kan exempelvis vara en automationslösning, att ta fram ny design för produktionsgolvet, eller en ny design för processen.

- Implementering och övervakning: Efter att ha implementerat de lösningsförslag som tagits fram ska de övervakas noggrant för se till att lösningsförslagen gör den nytta de är tänkta att göra.

- Fortsatt utveckling: Flaskhalsanalys är en pågående process som måste genomföras kontinuerligt då det alltid finns en risk för bildandet av nya flaskhalsar som sedan måste återgäldas.

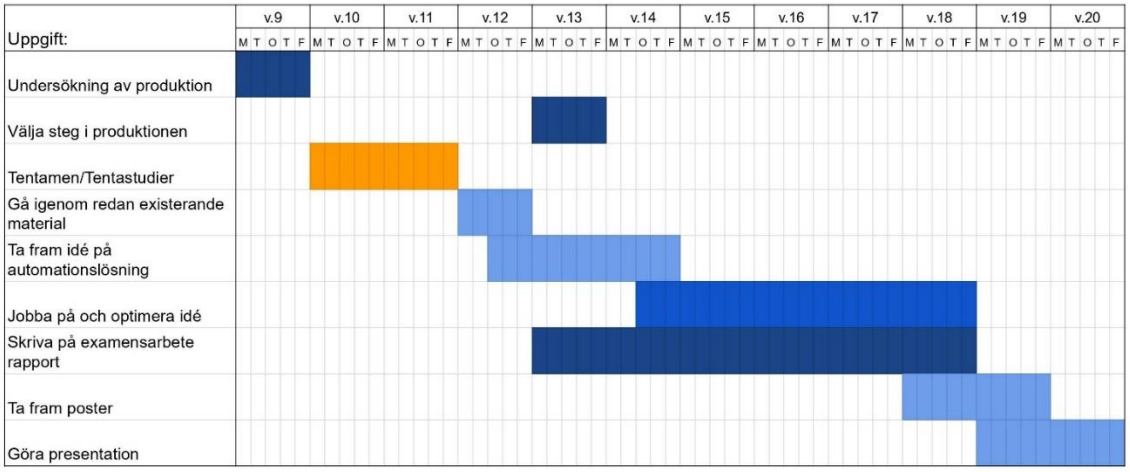
Målet med att genomföra flaskhalsanalys är att balansera produktionsflödet för att åstadkomma en så effektiv och optimal produktion som möjligt. Målet är produktionen ska flyta på utan några större stopp längst vägen [8].

3 Metod

I detta kapitel kommer metodiken och tillvägagångsättet som har använts under arbetet att presenteras.

3.1 Examensarbetets uppstart

Examensarbetet började med att tillsammans med en kontaktperson på företaget Aimpoint framställa en initial projektbeskrivning som skulle agera som grund för arbetet. Denna projektbeskrivning innehöll en koncis beskrivning av produktionens nuläge och målet med projektet. Målet var att ta fram ett förslag som gör att produktionstakten ökade. Det bestämdes att detta mål skulle uppnås med hjälp av en automatisering av en uppgift i ett siktes produktionsflöde. Efter intern diskussion på företaget beslutades det att siktet som skulle nyttjas mest av automatisering i sin produktion är Aimpoints sikte X och att studenten hade fria tyglar kring hur denna automationslösning skulle se ut. Efter en färdigställd projektbeskrivning som Aimpoint var nöjda med, skapades en initialbeskrivning med denna som grund som sedan godkändes av handledare och examinator. I samband med initialbeskrivningen skapades även ett Gantt-schema, se figur 4.



Figur 4. Gantt schema som togs fram i arbetets början.

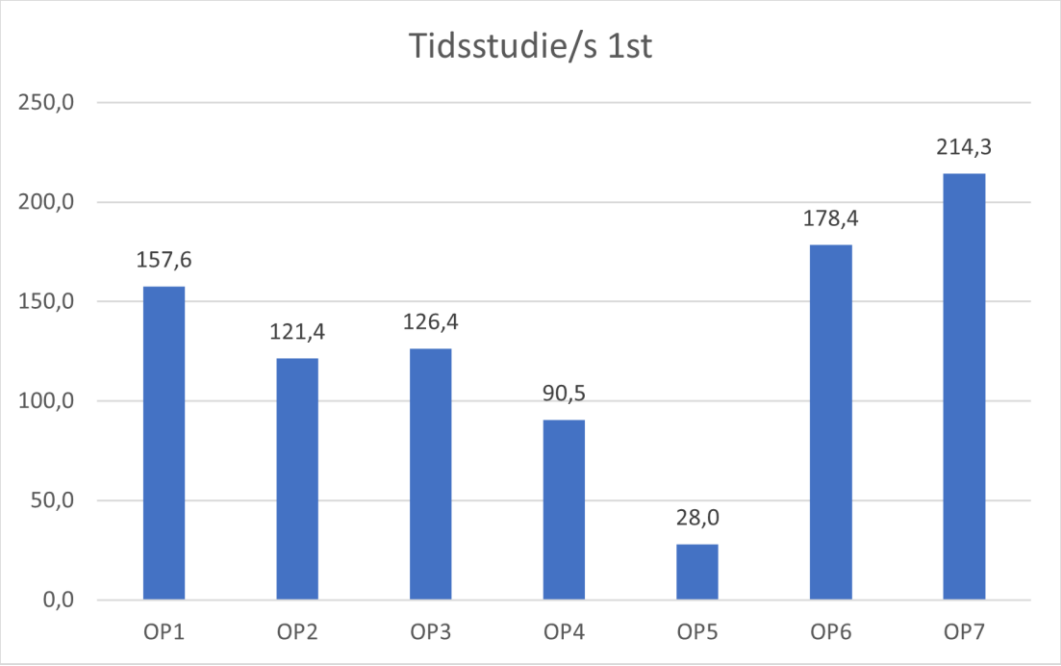
3.2 Examensarbetets faser

Arbetet började därefter med en genomgång av Aimpoints produktion. Detta för att bilda en förståelse för de termer som används för att beskriva olika steg i produktionen, samt för att få en övergripande förståelse innan en fördjupning i tidigare material inleddes. Efter detta gick arbetet in i olika faser som förhöll sig till Gantt-schemat.

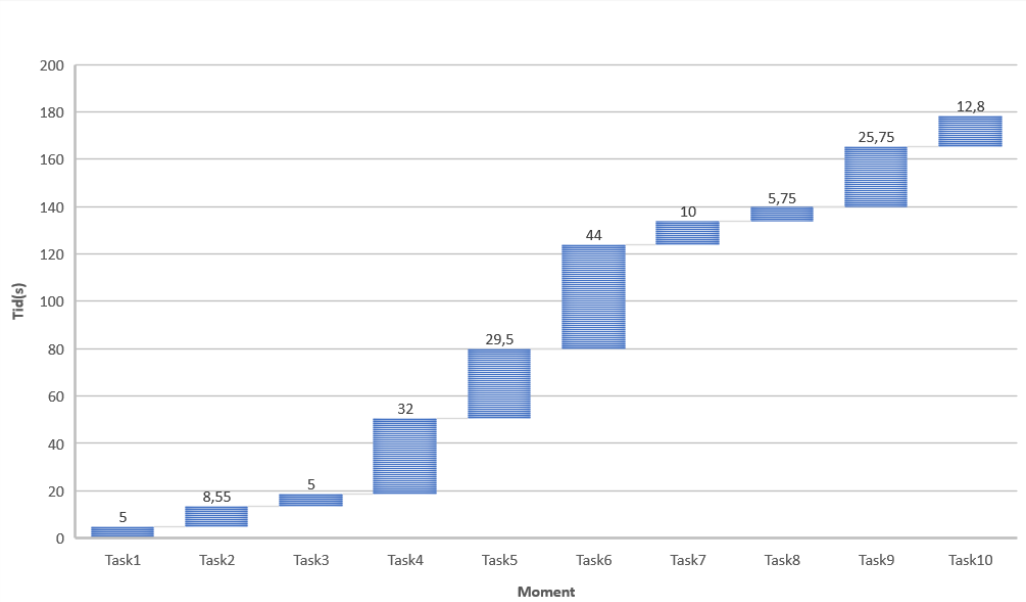
3.2.1 Granskning av tidigare material

Aimpoint har tidigare gjort omfattande undersökningar för att kartlägga sin produktion och identifiera flaskhalsar och områden där andra möjligheter till förbättring kan existera. Materialet varierade från Excelark där uppmätta tider var plottade i grafer, till anteckningar från brainstormingsessioner där olika förslag till förbättringar diskuterats. Det bestämdes att en översiktlig genomläsning av all data och information tillgänglig var nödvändig för att hitta den data som var mest lönsam att granska. Under den initiala genomläsningen antecknades det vilka dokument som kunde vara mest relevanta för arbetet. Efter den initiala filtreringen

av data påbörjades en mer noggrann granskning av de utvalda dokumenten, med målet att hitta de mest relevanta och nya dokumenten som presenterade tidmätningarna av siktet X:s produktion. Desto nyare ett dokument är desto bättre visar dokumentet de tider som gäller i produktionen i nuläget. Detta steg resulterade i upptäckten av två dokument som båda var de senaste och därmed mest relevanta tidmätningarna från produktionslinan. Ena dokumentet presenterade tidmätningar för varje enskild operation i den delen av produktionen som skulle undersökas, se figur 5. Det andra dokumentet presenterade tidmätningar från varje enskild uppgift inom varje operation, se figur 6 för utdrag. Operationerna och deras uppgifter har märkts om för att inte avslöja detaljer om produktionsprocessen. Med dessa dokument i åtanke kunde nästa fas påbörjas.



Figur 5. Dokument med uppmätta tider för varje operation i siktet X:s produktion



Figur 6. Uppmätta tider för enskilda steg i operation 6

3.2.2 Välja ut operation och uppgift i operationen för automatisering

Inför denna fas togs det fram punkter att utgå från i syfte att metodiskt komma fram till den operation och steget i den operationen som skulle nyttjas mest av automatisering. Dessa punkter var:

- Vilka operationer tar mycket längre tid än planerat?
- Vilka uppgifter finns i den operationen?
- Vilket/vilka uppgifter tar längst tid samt är mest benägen att ge montörerna slitskador?
- Vad är det som gör att uppgiften tar så lång tid?
- Vilka förbättringsförslag, om några, har tagits fram?
    - Implementerades de? Varför/varför inte?

Med dessa punkter som grund påbörjades analys av grafen från figur 5. Då OP 7 är den operation som tar längst tid började undersökningen med att kolla på denna. Det blev snabbt tydligt att denna operation inte är lämplig för automatisering då den innehåller finjusteringar som borde utföras av en människa. Det blev därför mest logiskt att undersöka och effektivisera den operation som tar näst längst tid, vilket är OP 6.

När operationen för arbetet hade valts var det aktuellt att närmare undersöka vilka uppgifter som ingår i operationen. Denna undersökning började med att gå igenom dokumentet som visas i figur 6, där en kort beskrivning och tiderna för varje uppgift är representerade. För att få en djupare inblick i stegen begärdes det tillgång till dokumentet med arbetsinstruktionerna som gavs av kontaktperson på Aimpoint. Detta dokument, i kombination med tiderna från figur 5, gav en överblick över vilka uppgifter som finns i operationen samt vilka uppgifter som tar längst tid. Arbetsinstruktionerna gav även en inblick i vilka uppgifter i operationen som var beroende av förgående uppgifter och vilka uppgifter som inte var det. Exempelvis är inte Task 6 beroende av något tidigare i just denna operation. Med detta som grund bokades det in en genomgång av produktionslinan, framför allt operation 6, med en skiftledare, för att även få en inblick i hur montörerna ställer sig till att automatisera en uppgift i operationen. Under denna rundvandring diskuterades och visades de olika uppgifterna i operationen upp. I dialog med skiftledaren framkom det att de uppgifter som tar längst tid även är de som är mest ergonomiskt utmanande vilket gör att montörerna ställer sig positiva till att automatisera just dem uppgifterna. Dessa uppgifter är rengöring av frontlins och insida hus, benämnda som Task 4 och Task 5 i figur 5. En analys av dessa uppgifter gav de steg som utgör uppgiften och som därför bör automatiseras. Dessa är:

1. Plockande av tops
2. Doppande av tops i aceton dispenser
3. Rengöring
4. Slängande av tops

Detta repeteras totalt fyra gånger, två gånger vid rengöring av insida hus och två gånger för frontlins. Den andra gången rengöringen genomfördes rengjordes enbart kanten på linserna. Efter en fråga om varför gavs svaret att det var på grund av att smutsuppsamlingen är större i kanterna än på resten av linserna.

Innan en automationslösning kunde börja utvecklas behövdes det utredas vad det är som gör att cykeltiden varierar och tar lång tid. Efter diskussion med skiftledaren visade det sig att anledningen till den stora variationen i cykeltid är oregelbunden rengöring och svårigheten att identifiera när ett sikte är tillräckligt rent. En bidragande faktor till att noggrannheten av

rengöringen blir oregelbunden är att mängden aceton som brukas vid varje omgång av rengöring varierar. Acetonflaskan som används för att dispensera aceton varierar mängden beroende på hur hårt och länge montören trycker ner topsen mot delen som dispenserar acetonen, se figur 7 för referensbild.

Figur 7. Referensbild för hur acetondispensern som används ser ut.

Efter att ha identifierat vilken operation och vilka uppgifter i operationen som skulle nyttjas av en automatisering, samt problemen med dessa uppgifter, undersöktes det om några tidigare förbättringsförslag för dessa uppgifter tagits fram. Detta undersöktes genom att gå igenom dokument med förbättringsförslag och genom att boka möte med en anställd på ME-(Manufacturing Engineer) avdelningen. I dokument med innehåll gällande förbättringsförslag hittades tidigare förslag rörande OP 6, däremot inga förslag rörande rengöring av insida hus och frontlins. Detta bekräftades efter möte med ME då det framkom att det inte diskuterats effektivisering gällande just rengöringsstegen i OP 6.

3.2.3 Framtagning av automationslösning

När operation samt uppgifter i denna operation hade valts ut och tidigare undersökningar tagits i beaktning började en automationslösning planeras utifrån den information som samlats. Ett möte bokades in med handledare för att diskutera vilka automationsalternativ som existerade. Mötet resulterade i att en automationslösning med en cobot från ABB borde tas fram i applikationen RobotStudio. Efter en granskning av tillgängliga cobots i ABB:s utbud bestämdes det att roboten CRB 1100 SWIFTI bör användas. Det togs sedan ett möte med kontaktpersonen på Aimpoint för att presentera idén. På detta möte uppkom det att vid automationsimplementeringar med robotteknik strävar Aimpoint efter att använda robotar från företaget ABB, vilket cementerade beslutet att använda ABB:s RobotStudio. Då studenten inte hade någon tidigare erfarenhet av RobotStudio behövdes det tid för att bli bekant med programmet och dess funktioner innan en automationslösning kunde framställas. Efter att bekantskap med programmet bildats togs det fram en automationslösning där roboten utförde de rengöringsuppgifter som valdes ut vid tidigare analys av OP 6. Denna automationslösning togs fram i RobotStudio och kan därför simuleras. Eftersom det går att simulera rengöringen går det även att mäta tiden det tar coboten att utföra uppgiften vilket gör det enkelt att se om lösning ger en förbättrad cykeltid.

3.3 Beräkning av kostnad vid implementering

ABB har inte kostnaden för sina robotar listade i webbutik utan begär en offertförfrågan vid köp. Det skickades därför in en offertförfrågan till ABB i syfte att kunna använda denna siffra vid beräkning av vad det skulle kosta Aimpoint att implementera automationslösningen. ABB svarade att de inte kunde tillhandahålla den typen av information eftersom roboten inte skulle köpas in. Sådan information ges endast till kunder. Då detta var enda sättet att få ett pris på vad ABB:s cobot SWIFTI CRB 1100 kostar togs det aldrig fram ett pris för roboten.

4 Analys

I denna del presenteras motiveringarna bakom de val som togs samt vilka problem som uppstod under arbetets process och hur dessa löstes.

4.1 Motivering av val

4.1.1 Val av sikte

Ett av de första valen som skulle göras i detta examensarbete var valet av vilket sikte som skulle undersökas och vars ena produktionssteg skulle automatiseras. Valet stod mellan Aimpoints sikte X och Y. Anledningen till att det stod mellan just siktet X och Y var att det finns ett flertal tidigare undersökningar av bådes sikte Y:s och X:s produktioner, Beslutet landade i att sikte X:s produktion skulle automatiseras och anledningen till sikte X valdes i stället för Y var att även om det finns mycket data på båda finns där ännu mer på X än Y. Anledningen till att det finns mer data på sikte X är att Aimpoint har gjort omfattande projekt med målen att effektivisera och förbättra siktets produktion. Att det finns mycket data från innan anses vara en fördel då det ger en bra grund för detta arbete och sparar studenten tid genom att denne inte behöver genomföra egna tidsstudier. Det var alltså en kombination av det fanns mer tidigare arbete på sikte X:s produktionsflöde än för Y och att Aimpoint länge arbetat med att få ned sikte X:s produktionstid till ett önskat värde som gjorde att sikte X valdes.

4.1.2 Val av ABB

Som tidigare nämnt i *3.2.3 Framtagning av automationslösning* strävar Aimpoint efter att använda företaget ABB:s robotar när det kommer till brukningen av robotik i sin produktionsmiljö. Motiveringen bakom detta beslut grundas på att de robotar som redan finns i produktionen, liksom de som planeras att anskaffas, är från ABB. Aimpoint anser att en enhetlig leverantör av robotar och kompatibilitet med samma mjukvara från ABB skulle underlätta driften och samordningen av automatiseringar.

4.1.3 Val av SWIFTI CRB 1100

I detta arbete var det upp till studenten att bestämma vilken av ABB:s cobots som skulle användas vid simulering. SWIFTI CRB 1100 valdes att användas i detta arbete då det både är en kompakt robot och snabbast av alla ABB:s cobots med en TCP hastighet på 5.05 m/sekund. Uppgiften som coboten har i detta produktionssteg hade också kunnat genomföras av ABB:s andra cobots med en lika bra precision och användarvänlighet som SWIFTI men då SWIFTI är snabbast och mest kompakt valdes denna. Det faktum att SWIFTI är en kompakt cobot är en fördel då arbetsytan ute i produktionen på Aimpoint inte är särskilt stor.

4.2 Arbetsrelaterade utmaningar

Under arbetet uppstod det olika utmaningar som behövde lösas för att arbetet skulle kunna fortsätta. Dessa utmaningar har varit väldigt givande i att de ofta gav en djupare förståelse för ämnet de berörde vilket förhindrade att samma problem uppstod vid ett senare tillfälle. Den första utmaningen var att välja ut vilket steg i produktionen där en automationslösning hade gjort nytta. Utmaningen i detta var att bearbeta den stora mängden data som fanns tillgänglig. Detta problem löstes genom att först ta fram en lista på vad det för typ av information som söktes. Det beslutades att det som söktes efter var framför allt två typer av

dokument. Ett dokument som presenterade aktuella tidmätningar från alla operationer i produktionen och ett dokument där tidmätningar från alla enskilda uppgifter i varje operation. Med detta i åtanke förenklades den första filtreringen då det gick snabbt att bedöma om ett dokument var användbart eller ej.

En annan utmaning som uppkom under arbetet var användningen av programmet RobotStudio. Utmaningen var att det inte fanns någon tidigare erfarenhet av RobotStudio. Detta löstes genom att det fick planeras in extra tid för upplärning av RobotStudio för att få en bättre förståelse för programmets olika funktioner och hur de kan nyttjas i syfte för detta arbete. Upplärning gick till igenom att börja med att göra enkla uppgifter i programmet för att få en djupare förståelse för hur roboten styrs. En sådan enkel uppgift var en så kallad ”pick and place” robot som greppar tag om ett objekt, flyttar det till en annan plats och sedan släpper objektet. Detta gav en förståelse för hur roboten programmeras, hur den flyttas från koordinat till koordinat och hur en greppfunktion kan genomföras.

När det kom till att programmera SWIFTI och skapa simuleringen användes en sekventiell utveckling där varje steg i processen utvecklades först enskilt innan de slogs samman. Exempelvis så utvecklades cirkelrörelsen som utförs vid rengöringen av linsen separat för att sedan sättas ihop med resten av programmet. Detta gjorde att det gick att experimentera med olika lösningar för ett visst steg utan risken att sabotera den kod och simulering som utvecklats dittills. Ett problem som ofta uppstod var när det kom till att lära roboten gå mellan två punkter. Först provades det att skapa punkterna och sedan låta programmet konfigurera vägen som SWIFTI skulle ta. Detta funkade inte alltid då programmet tog fram en väg som inte fungerade eller som krockade med ett något typ av objekt. Detta hinder överkoms genom användning av den virtuella FlexPendant som finns tillgänglig i RobotStudio. Genom att använda den gavs möjligheten att individuellt konfigurera rörelsevägen som roboten tar, i stället för att vägen konfigureras av programmet.

5 Resultat

I detta avsnitt presenteras tidmätningar från simulering som skapats och delar av koden som driver den.

5.1 Simuleringens kod

I detta delkapitel redovisas den kod som driver den framtagna simuleringen.

5.1.1 Huvudmetod i RAPID

Simuleringen som skapats i RobotStudio innefattar rengöring av tre sikten. Själva rengöringen omfattar rengöring av insida hus och frontlins. Simulering följer de steg observerade i *3.2.2 Välja ut operation och uppgift i operationen för automatisering*, vilket innebär att varje lins rengörs två gånger. Denna simulering är ett resultat av att köra den skrivna koden i RAPID:s huvudmetod, som kan ses i figur 8.

```
PROC main ()
    PulseDO Start;
    FOR Y FROM 0 TO 2 DO

    PulseDO Detach;
    WaitTime 0.5;

    pickUp_above;
    dip_above;

    round_one;
    throw;

    pickUp_above;
    dip_above;

    round_two;
    throw;

    pickUp_below;
    dip_below;

    round_three;
    throw_below;

    pickUp_below;
    dip_below;

    round_four;
    throw_below;


    target_offs := target_offs - 100;

    ENDFOR
```

Figur 8. Kodexempel på huvudmetod i RAPID.

Huvudmetoden börjar med en Start puls som användes vid tidmätning innan metoden påbörjar en for-loop. Denna for-loop körs lika många gånger som antalet sikten, vilket är tre. Metoden börjar med två rengöringar av insidans lins innan frontlinsen också rengörs två gånger. For-loopen börjar med *PulseDO Detach* som öppnar robotens grip för att släppa tag om objekt. SWIFTI plockar sedan upp en tops via kommandot *pickUp_above* innan den sedan doppar topsen i aceton genom kommandot *dip_above*.

När SWIFTI nu har en tops med aceton på sig börjar rengöringen genom exekveringen av processen *round_one*. Efter rengöringen så slänger roboten topsen via kommandot *throw;*. Samma process upprepas sedan en gång till fast denna gång byts *round_one* ut av *round_two* som är en process för andra rengöringen av samma lins.

Efter detta körs kod för att rengöra frontlinsen. Detta görs genom att roboten, precis som vid rengöring av första linsen, först plockar upp en tops och doppar den i aceton via kommandona *pickUp_below* och *dip_below*. Sedan exekveras processen *round three* som är första rengöringen av frontlinsen, varpå topsen sedan slängs via kommandot *throw*_below. Processen upprepar sig sedan genom att SWIFTI först plockar upp en ny tops och doppar den i aceton innan processen *round_four*, som är andra rengöringen av frontlinsen, körs. Slutligen slängs topsen från andra rengöringen innan variabeln *target_offs* ändras så att rengöringen av nästa sikte kan påbörjas. Vid testfall där enbart ett sikte ska rengöras så kommenteras for-loopen bort så att huvudmetoden enbart genomföring rengöring av första siktet, den resulterande koden blir då den representerad i figur 9.

```
PROC main ()
    PulseDO Start;
    !FOR Y FROM 0 TO 2 DO

    PulseDO Detach;
    WaitTime 0.5;

    pickUp_above;
    dip_above;

    round_one;
    throw;

    pickUp_above;
    dip_above;

    round_two;
    throw;

    pickUp_below;
    dip_below;

    round_three;
    throw_below;

    pickUp_below;
    dip_below;

    round_four;
    throw_below;


    target_offs := target_offs - 100;

    !ENDFOR


ENDPROC
```

Figur 9. RAPID kodexempel, ändrad huvudmetod för rengöring av enbart ett sikte.

5.1.2 Processerna för rengöring av linserna i RAPID

När linsen på insidan ska rengöras körs processerna *round_one* och *round_two*¸ vilka kan ses i figur 10. Vid rengöring av frontlinsen körs processerna *round_three* och *round_four*, som kan ses i figur 11.

```
PROC round_one()
    MoveL Offs(spin_setup,target_offs, 0, 0),v1000,fine,Servo\WObj:=wobj0;
    MoveL Offs(origo_begin, target_offs, 0, 0), v1000,fine,Servo;
    C1_above;
    C2_above;
    C3_above;
    C4_above;
    MoveL Offs(origo_begin, target_offs, 0, 0), v1000,fine,Servo;
    MoveL Offs(spin_setup,target_offs, 0, 0),v1000,fine,Servo\WObj:=wobj0;
ENDPROC

PROC round_two()
    MoveL Offs(spin_setup,target_offs, 0, 0),v1000,fine,Servo\WObj:=wobj0;
    MoveL Offs(origo_begin, target_offs, 0, 0),v1000,fine,Servo;
    MoveL Offs(origo_above, target_offs, 0, 0), v1000, fine, Servo;
    MoveL Offs(s3, target_offs, 0, 0),velocity,fine,Servo;
    C3_above;
    C4_above;
    MoveL Offs(origo_begin, target_offs, 0, 0),v1000,fine,Servo;
ENDPROC
```

Figur 10. Kodexempel på processerna *round_one* och *round_two*.

```
PROC round_three ()
    C1_below;
    C2_below;
    C3_below;
    C4_below;
ENDPROC
PROC round_four ()
    MoveJ Offs(under_1_3,target_offs, 0, 0),v1000,fine,Servo\WObj:=wobj0;
    MoveL Offs(under_south, target_offs, -3, 0), v500, fine, Servo_2;
    C3_below;
    C4_below;
ENDPROC
```

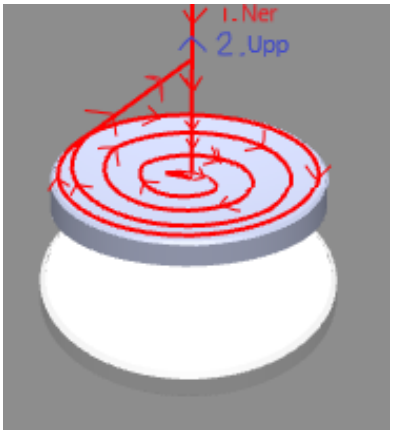
Figur 11. Kodexempel på processerna *round_three* och *round_four*.

Processerna *round_one* och *round_two* är strukturerade på liknande sätt. Processen *round_one* börjar med att roboten flyttar till en punkt ovanför siktet genom kommandot *MoveL Offs(spin_setup, target_offs, 0, 0), v1000, fine, Servo;* innan den sedan rör sig till en punkt på samma höjd, men som är rakt ovanför mittpunkten på linsen, via kommandot *MoveL Offs(origo_begin, target_offs, 0, 0), v1000, fine, Servo;*. Roboten rör sig sedan rakt nedåt till mittpunkten av linsen för att sedan börja rengöra genom att följa de kommandon som ges i processerna *C1_above*, *C2_above*, *C3_above* och *C4_above*. Dessa processer innehåller kommando som gör att topsen rör sig i fyra olika cirklar: C1, C2, C3 och C4, där C1 är den innersta cirkeln och C4 är den yttersta. Efter att ha rengjort linsen tar roboten ut topsen från sikten och gör sig redo att flytta till nästa punkt. På ett liknande sätt rör sig roboten till en startpunkt i processen *round_two* innan den sedan rör sig nedåt i Z-led i 3D-rumemt. Topsen förs sedan till punkten *s3*, som är en punkt på linsen, innan en ytterligare rengöring av den yttre kanten börjar genom exekvering av *C3_above* och *C4_above*.

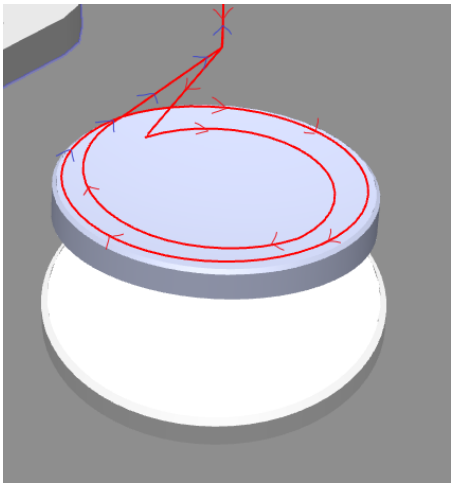
Vid rengöring av frontlinsen körs processerna *round_three* och *round_four*. Dessa processer har liknande struktur som *round_one* och *round_two* men är annorlunda då roboten har en annan ställning för att kunna nå frontlinsen med topsen. En annan skillnad är även att i stället för att startställningen för roboten befinner sig i *round_three* befinner den sig i *C1_below*. Men på samma sätt som vid rengöringen av insidans lins så drivs roboten till en startpunkt innan den sedan påbörjar rengöringen genom att följa cirklarna som bildas av processerna *C1_below*, *C2_below*, *C3_below* och *C4_below*. Processen *round_three* gör en omfattande rengöringen av hela linsen medan *round_four* genomför en ytterligare rengöring av specifikt linsens kanter.

5.2 Simuleringens rörelse

Genom att spåra spetsen av topsen som rengör siktet kunde de mönster som bildades vid rengöring av linserna visas. Mönstret vid de två rengöringarna av linsen på insidan kan ses i figur 12 och figur 13. Dessa steg genomförs när processerna *round_one* och *round_two* exekveras i huvudmetoden från figur 8.

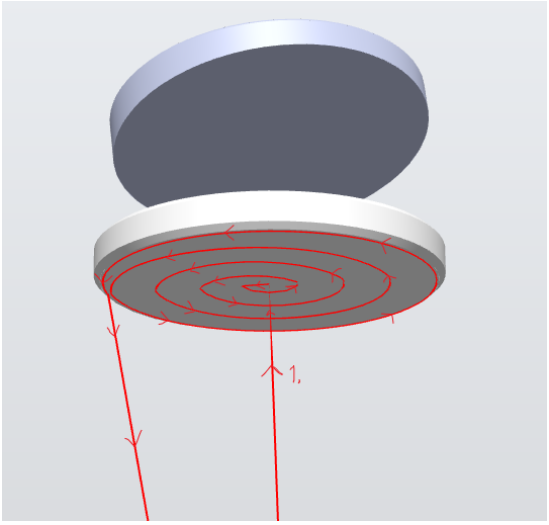


Figur 12. Mönstret vid första rengöringen av insidans lins.

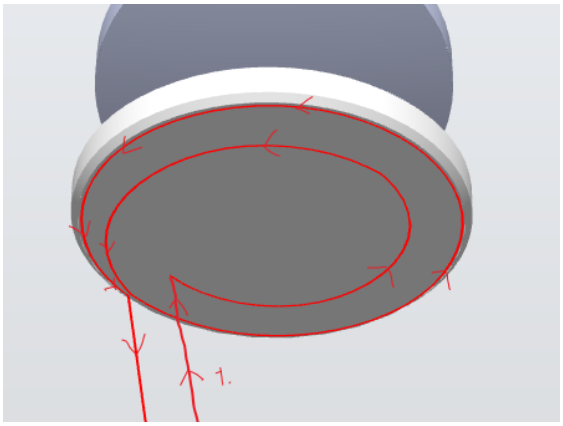


Figur 13. Mönstret vid andra rengöringen av insidans lins.

Mönstret som uppstår vid två stegs rengöringen av frontlinsen kan ses i figur 14 och figur 15 och uppstår när processerna *round_three* och *round_four* körs i huvudmetoden från figur 8.



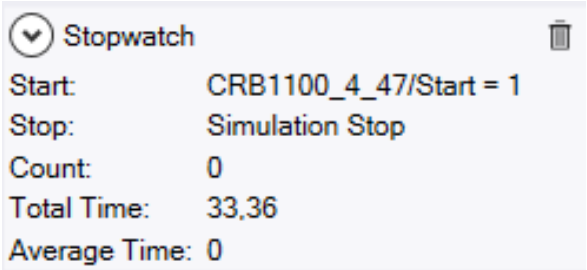
Figur 14. Mönstret vid första rengöringen av frontlins.



Figur 15. Mönstret vid andra rengöringen av frontlinsen.

5.3 Simuleringens tidmätning

Det gjordes en tidmätning för rengöring av enbart ett sikte. Detta gjordes genom att köra huvudmetoden från figur 9. Tidmätningen från simuleringen där endast ett sikte rengjordes resulterade i 33,36 sekunder, vilket kan ses i figur 16.



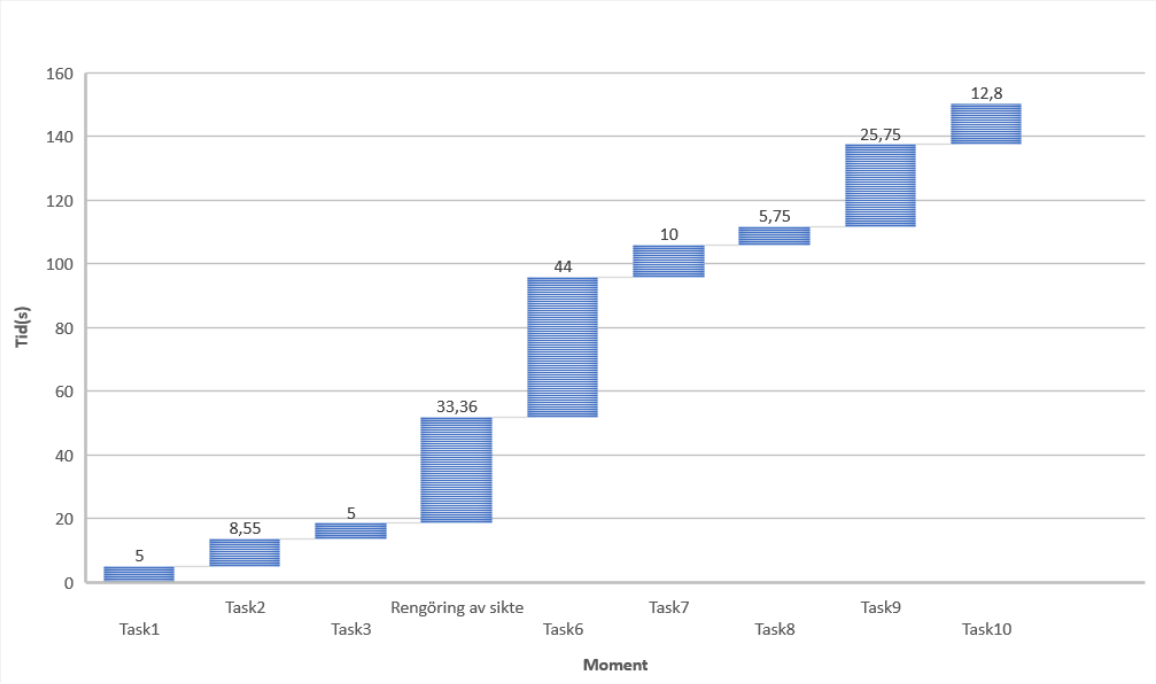
Figur 16. Tidmätning av simuleringen där ett sikte rengjorts i RobotStudio.

Det gjordes även en tidmätning av en simulering där huvudmetoden från figur 8 kördes. Den uppmätta tiden blev då 103,632 sekunder, vilket framgår av figur 17.

Figur 17. Tidmätning av simuleringen där tre sikten rengjorts i RobotStudio.

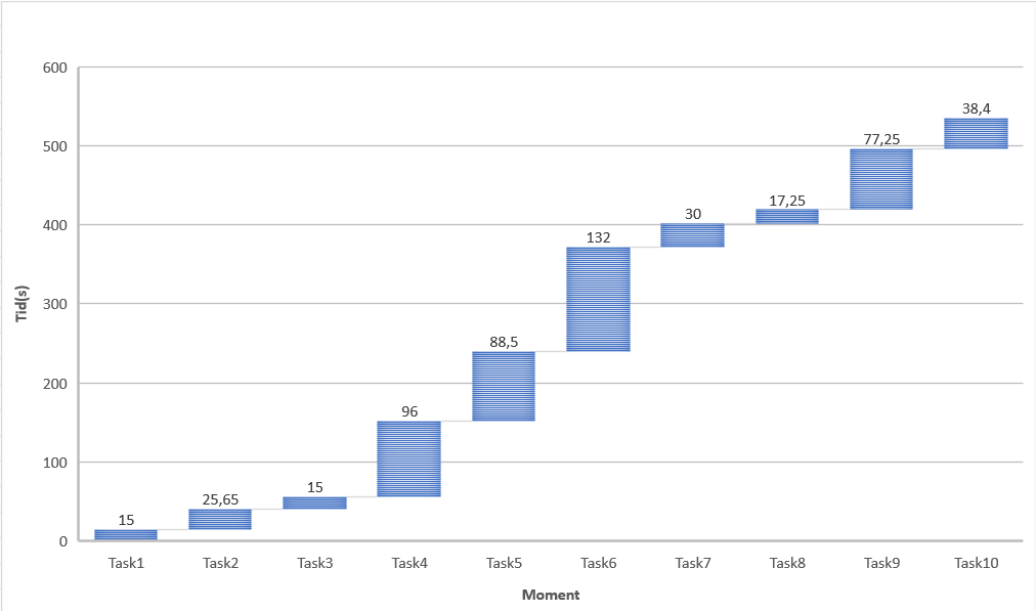
5.4 Uppdaterad cykeltid vid insättning av uppmätta tider

För att undersöka cykeltidens förändring används grafen från figur 5, där en summering av tiderna för varje uppgift blir 178 sekunder. Förändringen av cykeltiden kan undersökas genom att ersätta de angiva tiderna som plottats i diagrammet, med tiderna som uppmättes i figur 16. Uppställning av tiderna behövde ändras då simuleringen genomför två av stegen i ett, nämlingen Task 4 och Task 5. Det nya diagrammet har därför ersatt dessa två uppgifter med endast en uppgift: *rengöring av sikte*. Detta resulterar i att operationens cykeltid minskar från 178 sekunder till 150 sekunder, vilket kan ses genom summering av tiderna i figur 18. Detta ger en skillnad på 28 sekunder.

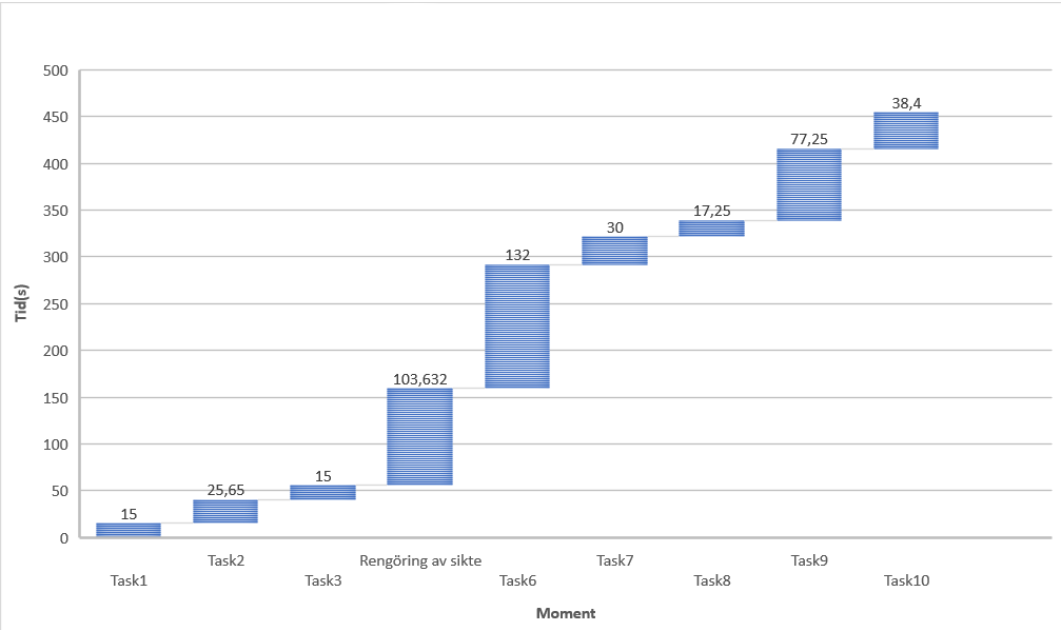


Figur 18. Tiderna i OP 6 uppdaterade med uppmätt tid från simulering.

Alternativt kan roboten köras så att den rengör tre sikten åt gången vilket innebär att diagrammets tid ersätts med tiden från figur 17 i stället för figur 16. Med denna implementering skulle montören behöva gör alla uppgifter, innan och efter rengöringen, tre gånger per genomkörning av operationen. Hade montören gjort tre sikten per genomkörning i dagsläget hade detta resulterat i en cykeltid på 535 sekunder, vilket kan ses i figur 19. Om tiden från figur 16 används skulle denna cykeltid i stället vara 454 sekunder, vilket kan ses i figur 20. Detta ger en skillnad på 81 sekunder.



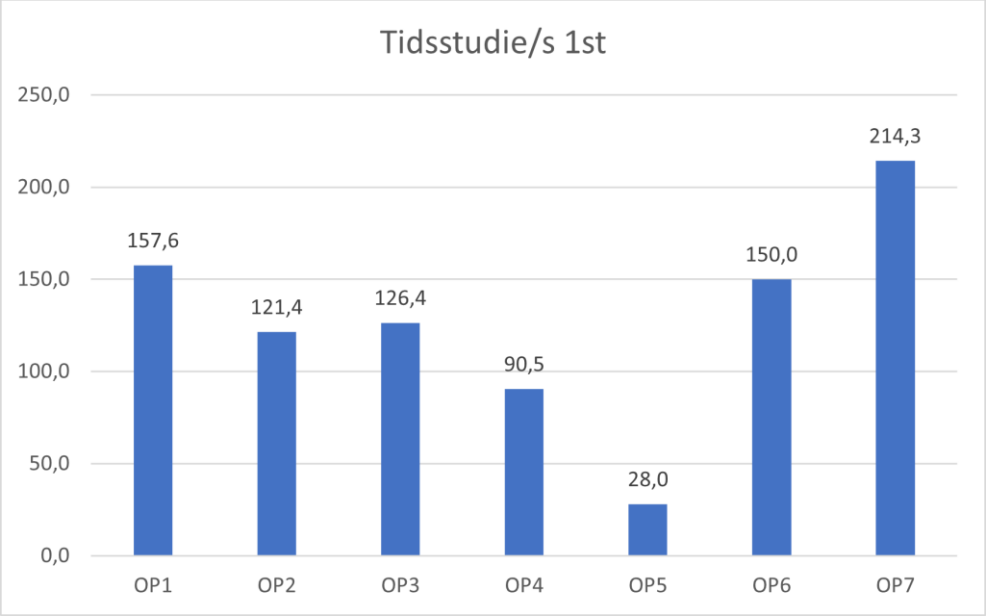
Figur 19. Tider för OP 6 om det skulle produceras tre sikten samtidigt i nuläget.



Figur 20. Uppdaterad cykeltid för OP 6 om det skulle produceras tre sikten samtidigt.

5.5 Uppdaterad takt vid insättning av uppmätta tider

För att se den resulterande förändring av takten vid implementering av automationslösningen i nulägets produktionslina används den uppdaterade cykeltiden från figur 18, som är 150 sekunder. Den nya cykeltiden för operation stoppas sedan in i figur 5. Detta ger upphov till diagrammet som kan ses i figur 21.



Figur 21. Uppdaterad tid för varje operation där cykeltiden för OP6 är uppdaterad.

Vid implementering av automationslösningen i nulägets produktionslina förblir siktets takt oförändrad då det inte är OP 6 som är flaskhalsen i produktionen, utan OP 7.

6 Slutsats

Detta kapitel redovisar slutsatser som har dragits utifrån resultatet av det genomförda arbetet samt framtida utvecklingsmöjligheter. Avsnittet innehåller även en etisk reflektion över automatisering.

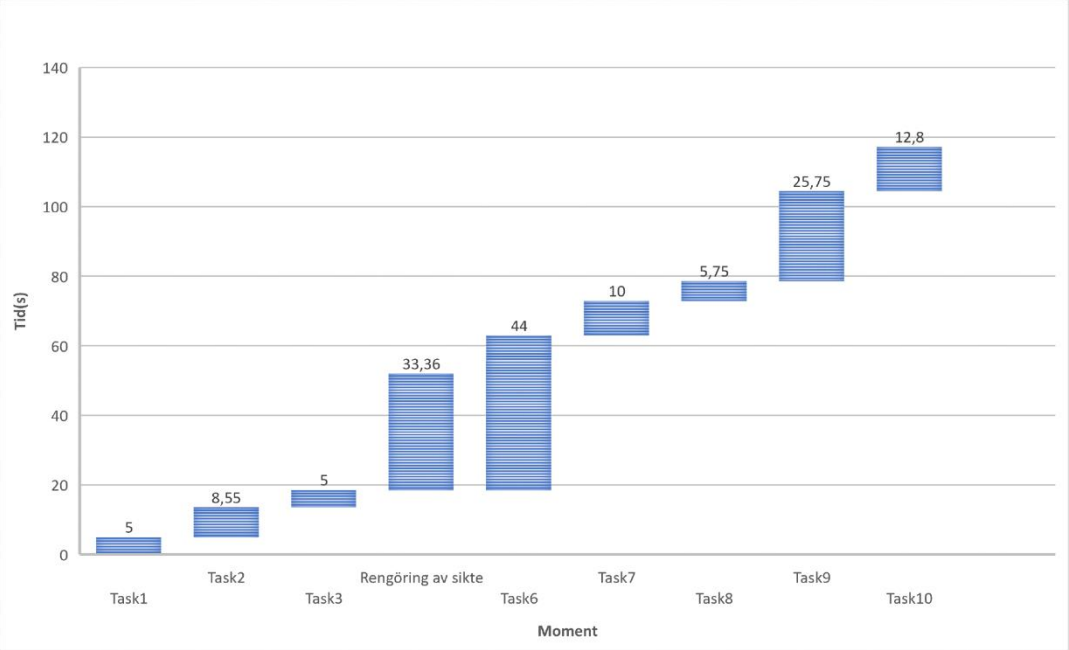
6.1 Besvarande av problemformulering

I kapitlet *1.4 Problemformulering* presenterades frågor som avsågs att besvaras i detta arbete. Dessa frågor ska nu besvaras.

1. Vilket steg i siktets produktion borde automatiseras?
   Steget som valdes ut för en implementering av en automationslösning valdes till rengöring av siktet i operation 6 för Aimpoints sikte X. Detta steg valdes då det är ett slitagemoment för montörerna samt ett steg som tar lång tid och som enligt Aimpoint varierar i tid då nyanställda har svårt att veta när de har rengjort siktet tillräckligt. Med en automatisering blir rengöringen mer koncis.

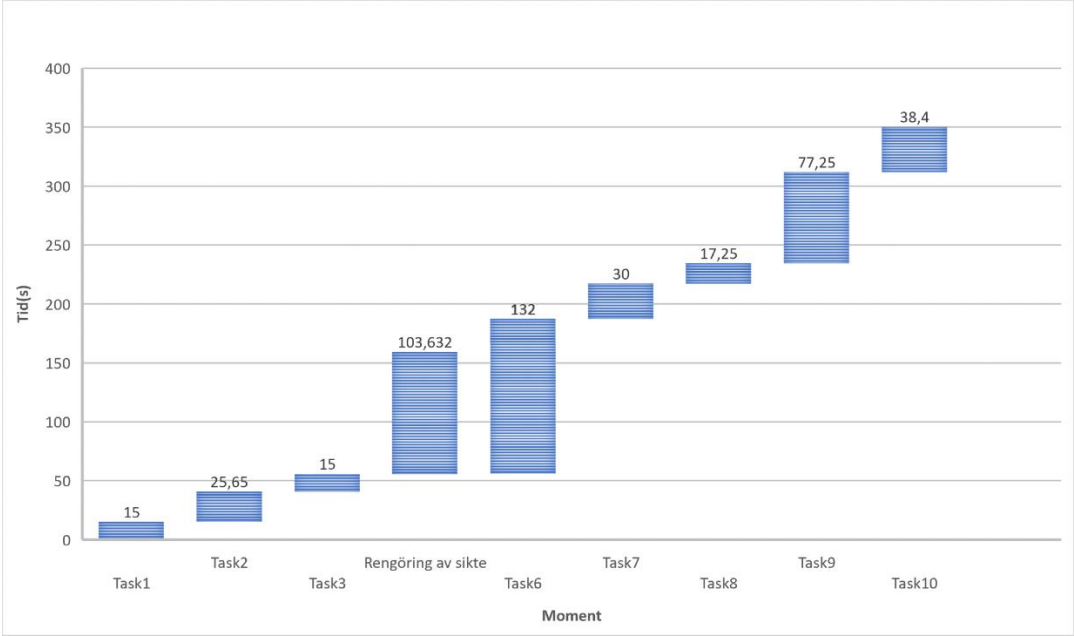
2. Vilken robot bör användas?
   Den robot som det beslutades att använda i detta arbete blev ABB:s cobot SWIFTI CRB 1100. Motiveringen till detta val grundade sig på två orsaker och presenteras i *4.1.2 Val av ABB* och *4.1.3Val av SWIFTI CRB 1100*. Kortfattat motiverades valet av en robot från just företaget ABB med att Aimpoint har tidigare erfarenhet av ABB robotar i sin produktion och ser fördelar med att använda robotar från samma företag. Valet att använda coboten SWIFTI motiveras främst av att den är kompakt och den snabbaste coboten i ABB:s utbud.

3. Vad blir förändringen av cykeltiden för momentet vid automatisering?
   Som presenterat i *5.4 Uppdaterad cykeltid vid insättning av uppmätta tider* ändras cykeltiden för OP 6 från 178 sekunder till 150 sekunder. Automationslösningen kan däremot bidra till en ännu lägre cykeltid. Då roboten genomför rengöringen av siktet lämnar det tid för montören att påbörja Task 6 i operationen. Detta är möjligt då Task 6 inte är beroende av rengöringen utan kan genomföras separat, som sagt *i 3.2.2 Välja ut operation och uppgift i operationen för automatisering* . Genom att montören påbörjar Task 6 parallellt med att coboten genomför rengöringen kan cykeltiden minska ännu mer. Detta resulterar i att cykeltiden går från 178 sekunder till 117 sekunder, vilket kan ses av diagrammet i figur 22. Detta innebär att cykeltiden skulle sänkas med 61 sekunder i stället för 28 sekunder.



Figur 22. Uppdaterat diagram med tider vid effektiv implementering.

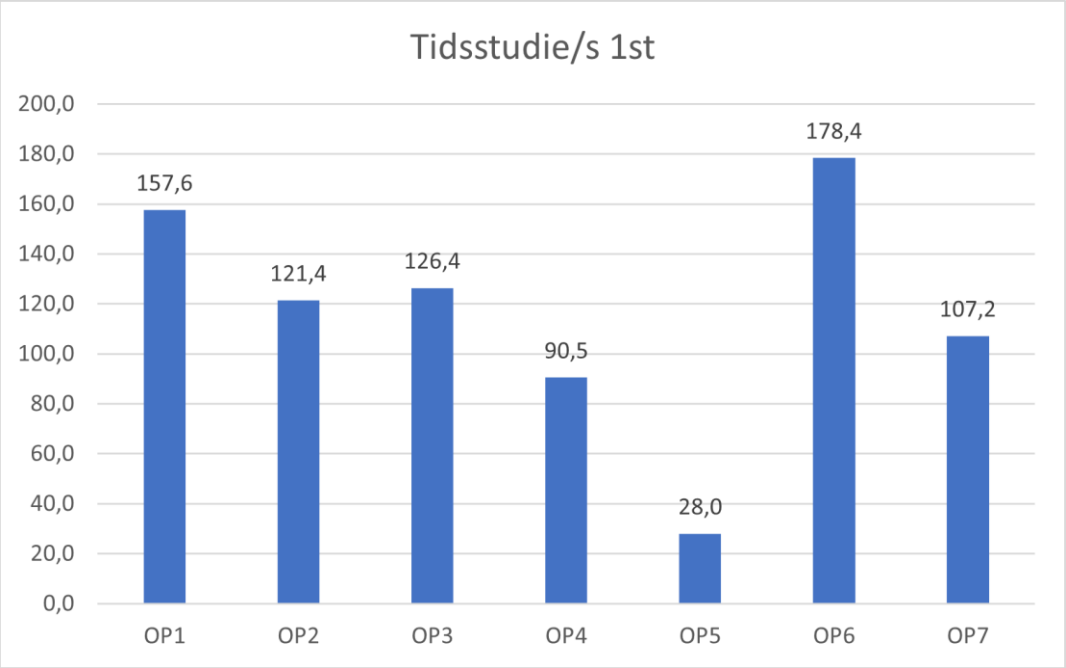
Genom att applicera samma tankesätt vid hantering av tre sikten parallellt blir cykeltiden för operationen i det fallet 350 sekunder, vilket innebär att cykeltiden minskar med 185 sekunder i stället för 81 sekunder. Detta kan ses i figur 23.



Figur 23. Uppdaterat diagram med tider vid effektiv implementering där tre sikten hanteras samtidigt.

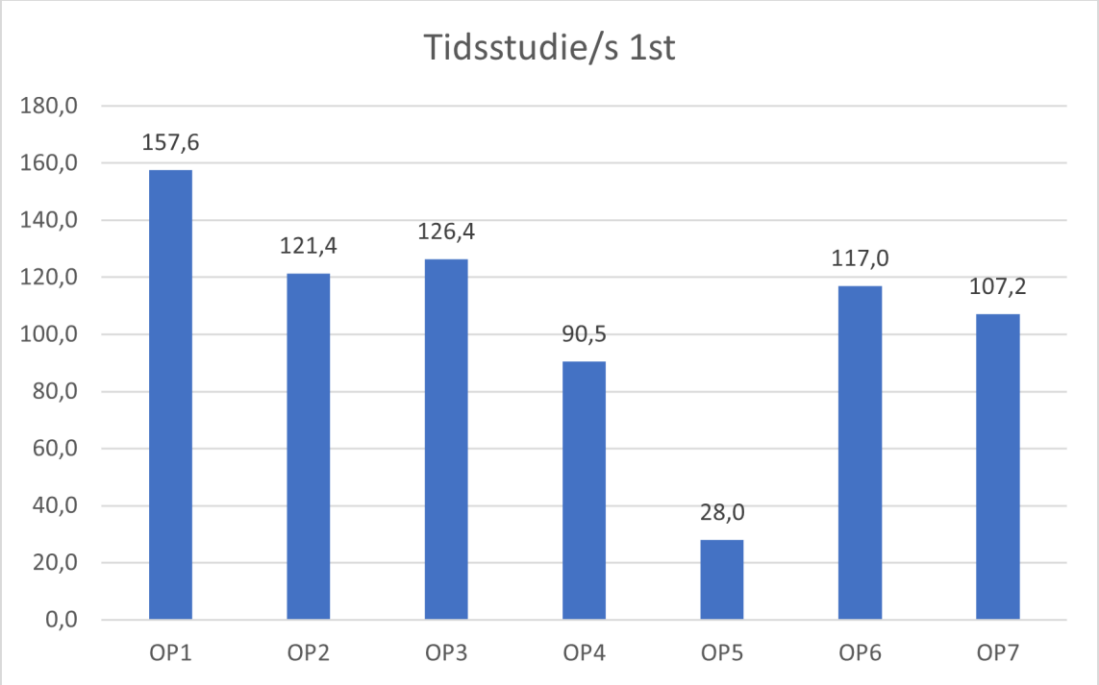
Sammanfattningsvis uppnås den maximala minskningen av cykeltiden genom att montören påbörjar Task 6 samtidigt som roboten påbörjar rengöringen av resterande del av siktet. Genom att göra på detta vis minskar operationens cykeltid vid hantering av ett sikte åt gången från 178 sekunder till 117 sekunder, vilket ger en total minskning på 61 sekunder. Denna metod gör även att cykeltiden vid hantering av tre sikten åt gången minskar från 535 sekunder till 350 sekunder, vilket är en minskning på 185 sekunder.

4. Vad blir förändringen av takten vid automatisering?
   Som det konstaterades i *5.5 Uppdaterad takt vid insättning av uppmätta tider* blir takten oförändrad vid implementering av automationslösningen, även om den mest effektiva implementeringen som ger en cykeltid på 117 sekunder används. Detta då, som kan ses i figur 5, det är OP 7 som är flaskhalsen i produktionslinan och därmed sätter takttiden för den delen av produktionen som undersöktes. Automationslösningen kan däremot bida till att öka takten ifall det införs en optimering av OP 7. Detta skulle exempelvis kunna ske genom att fördubbla OP 7 och ha två stationer som utför denna operation parallellt. Detta hade resulterat i en halvering av cykeltiden när den tas i beaktning tillsammans med de andra operationerna. Resultatet av att göra detta med OP 7 innan implementering av automationslösningen kan ses i figur 24.



Figur 24. Resulterande cykeltiden vid dubblering av OP 7.

Detta gör att flaskhalsen i operationen nu är OP 6. Genom att nu genomföra den effektiva implementeringen av automationslösningen sänks cykeltiden för OP 6 till 117 sekunder, vilket eliminerar denna operation som flaskhals. Resultatet av att göra detta gör att OP 1 numera är den operation som tar längst tid och som därmed sätter takten. Då cykeltiden för OP 1 är 157,6 sekunder har den genomsnittliga cykeltiden blivit sänkt från 214,3 sekunder till 157,6 sekunder, vilket kan ses i figur 25.



Figur 25. Produktionsflödet vid effektiv implementering av automatiseringen.

5. Vad är den ungefärliga kostnaden vid implementering av automationslösningen?
   Då det ej gick att erbjudas en offert av ABB gick det inte att besvara denna fråga i detta arbete.

6.2 Framtida utvecklingsmöjligheter

I diskussionen med skiftledaren, som presenterades i *3.2.2 Välja ut operation och uppgift i operationen för automatisering*, framkom det att en anledning till att rengöringssteget tar lång tid är att det är svårt att identifiera när ett sikte har blivit tillräckligt rent vid rengöringen. En utvecklingsmöjlighet vore därför att implementera en kamera i samspel med roboten som kan identifiera om sikte har blivit tillräckligt rent. Ett förslag är att använda en kamera som med hjälp av en bildbank kan användas för att avgöra om siktet är rent. Att implementera en sådan lösning skulle kräva ett omfattande förarbete där det tas många bilder av både rena och smutsiga sikten där bilderna sedan kategoriseras som antingen rena eller smutsiga. Med en kamera till hjälp kan det bli en mer konsekvent rengöring och en bättre tillit till att varje sikte som roboten rengör blir rent. Med hjälp av en kamera kan siktena även bedömas i förväg vare sig de är i behov av rengöring eller inte och på så sätt spara ännu mer tid.

Ett ytterligare förslag är att vidareutveckla den fixtur som håller siktet på plats vid rengöring. Vid implementering av simuleringen från RobotStudio behöver siktet just nu placeras helt rätt för att linserna ska bli ordentligt rengjorda. Det skulle behövas en fixtur där det är enkelt att sätta siktet helt rätt, utan att behöva oroa sig för att siktet har placerats för högt upp eller för lågt ner.

En vidare förbättring vore att utöka användningsområdet till att omfatta nästa uppgift i operation 6. Denna uppgift har också med rengöring att göra, vilket medför liknande ergonomiska problem för montören som rengöring av resterande delen av siktet, vilket medför att en automatisering av även denna uppgift skulle hjälpa montörerna undvika

slitskador. Detta skulle också göra att medan roboten genomför rengöringen kan montören hjälpa till vid ett annat moment i produktionen då uppgiften efter rengöring av baklins inte går att genomföra om inte baklinsen har rengjorts. Kameralösningen skulle också vara till hjälp vid denna uppgift då det även här behöver identifieras vare sig linsen har blivit tillräckligt ren eller inte.

6.3 En etisk reflektion över dilemmat vid användandet av cobots

På senare tid har alltmer automatiserats, inte minst inom världen av produktion. Denna omställning, från mänsklig arbetskraft till robotar, har gått väldigt snabbt, och till en början, utan större omtanke för dess konsekvenser. Automatisering har medfört betydande förändringar för hur produktion ser ut idag, bland annat genom att ha möjliggjort snabbare produktion men också att behovet av mänsklig arbetskraft har minskat, och det är det sistnämnda som framför allt har väckt debatt. En fråga som många ofta ställer sig är om automatisering är värt risken att lämna människor utan jobb. Anledningen till att detta är en vanlig fundering är eftersom det ibland händer att något automatiseras och till följd av det blir någon arbetslös eftersom en robot kan göra jobbet snabbare och utan paus. Det finns även företag som Aimpoint där det finns en rädsla om att gå miste om sin produktkvalitet som bland annat kommer från att använda just manuellt arbete. Däremot finns det de som menar att automatisering kan hjälpa frigöra oss människor från att göra de saker vi ändå inte hade velat göra och minskar risken för arbetsskador.

Som ett mellanting mellan dessa två ståndpunkter har konceptet av cobots blivit alltmer aktuellt på senare tid. Det som lockar framför allt företag med just cobots är att det gör det möjligt för dem att hålla sig i kapp med teknologin samtidigt som de inte behöver välja bort de fördelar som finns med manuellt arbete. Cobots är nämligen designade och utformade för att arbeta vid sidan av människor, vilket inte är möjligt ifall det inte finns några arbetare att arbeta vid sidan av. Det är på grund av denna anledning som Aimpoint och andra företag är intresserade av cobots och hur den typen av robot kan hjälpa deras produktion och deras arbetare att undvika slitskador. Cobots agerar som mellanhand mellan de två sidorna av de aspekter som dominerar automatiserings debatten idag. Detta då dessa typer av robotar möjliggör för en mer effektiv produktion samtidigt som ett av deras primära syften är att underlätta arbetet för människor utan att ersätta dem, vilket tilltalar aspekten för potentiell arbetslöshet samt aspekten gällande underlättat arbete.

7 Källförteckning

7.1 Källkritik

[1], [2]: Som källa för företaget Aimpoints historia och värdegrunder används Aimpoints egna hemsida för införskaffande av denna information. Detta har gjorts då företag själva kan anses känna till sin historia och värdegrunder bäst. Aimpoints historia är troligtvis skriven för företagets kunder då det är ett rimligt antagande att kunder vill känna till historien bakom hur företag grundades och växt till det stadiet som företaget befinner sig i idag.

[3]: Som källa för historien om begreppet Digital Tvillings ursprung användes ett kapitel i en publicerad bok. Detta kapitel är skrivet av Michael Grieves, forskaren som tog fram grunderna för vad som sedan skulle komma att kallas för Digital Tvilling, och John Vickers som är senior ledare på NASA. Grieves är en väletablerad och erkänd forskare inom sitt ämne och har gjort många publiceringar och arbeten som använts av många företag för att forma den produktionsmiljön som är idag. John Vickers är NASA:s huvudteknolog för avancerade material och tillverkning. Han har över 30 års erfarenhet inom området och har arbetat med forskning, ingenjörskonst och produktion för olika rymdrelaterade system. Bokens utgivare är ett förlag baserat i Tyskland som publicerar akademiska tidskrifter och böcker i hela världen och kan därmed anses vara pålitlig. Kapitel utdraget som har används som källa i detta arbete innehåller information om vad som är grunderna i begreppet Digital Tvilling motiveringen till varför Grieves tog fram detta koncept. Kapitlet är skrivet för andra forskare och elever som vill lära sig mer om begreppet Digital Tvilling. Boken är publicerad år 2017 och även om begreppet i sig ändras med tiden, kommer historian bakom begreppet inte att förändras vilket gör att även om denna källa är 7 år gammal är den relevant för bruk även idag.

[4]: En ytterligare källa som användes för information kring begreppet Digital Tvilling är ett kapitel skriven av samma Michael Grieves som är medförfattare till källa [3]. Detta kapitelutdrag är däremot taget ur en annan bok och är mer inriktad på vad begreppet Digital Tvilling omfattar idag snarare än dess ursprung. Boken är publicerad av American Institute of Aeronautics and Astronautics som är en professionell organisation för flygteknik och förkortas AIAA. Kapitlet är till för andra forskare och för studerande som vill lära sig mer om Digital Tvillingar och dess olika typer. Boken är skriven år 2019 och kan därmed fortfarande anses vara aktuell.

[5]: Som källa för information kring ABB:s datorapplikation RobotStudio har RobotStudios egen manual använts då den anses innehålla all tänkbar information som kan behövas gällande programmet. Manualen är skapad av företaget ABB och anses vara en pålitlig utgivare då det är företaget själva som skapat applikationen och därmed kan anses vara experter på programmet. Manualen är till för användare av programmet RobotStudio för att ge dem enkel tillgång till information de kan behöva för att komma i gång med programmet och för att få en bättre förståelse för dess funktioner. Manualen kom ut i samband med den senaste versionen av RobotStudio, vilket är år 2024.

[6]: Källan som användes för att hämta information kring cobots som koncept är en forskning artikel skriven av Claudio Taesi, Francesco Aggogeri och Nicola Pellegrini. Taesi och Pellegrini är doktorander vid institutionen för maskin- och industriell teknik på University of Brescia där Aggogeri även är professor. University of Brescia är ett italienskt universitet som bildades år 1982 i Brescia och kan därmed anses som ett seriöst institut. Artikeln är publicerad av förlaget MDPI som är en utgivare av vetenskapliga tidskrifter med öppen

tillgången som bildades år 1996. MDPI är bland de största förlagen i världen när det gäller publiceringar av tidskriftsartiklar. Materialet är skrivet för forskare och studerande som vill läsa på om ämnet cobots och har publicerats just med detta som syfte. Artikeln publicerades så sent som 2023 och kan därför anses vara aktuell.

[7]: Denna källa har använts för att ge en inblick i de tekniska specifikationerna om SWIFTI CRB 1100. Denna källa är ett datablad skapat av företaget ABB, som också skapat roboten i fråga. Då ABB är företaget som skapat roboten så är det rimligt att anta att informationen i detta datablad stämmer och är faktamässigt korrekt. Databladet innehåller information kring de tekniska specifikationerna för SWIFTI såsom dess räckvidd, tyngd, funktioner och nyttolast. Dokumentet är skapat för användare av SWIFTI eller någon som gör forskning kring roboten och vill undersöka den i mer detalj. Databladet publicerades i samband med att ABB gjorde SWIFTI tillgänglig för inköp år 2021.

[8]: Denna artikel är skriven av Wieslaw Urban och Patrycja Rogowska. Urban är docent på Bialystok University of Technology, vilket är samma universitet som där Rogowska är doktorand. Då ena författaren är docent och den andra doktorand anses de vara pålitliga författare. Bialystok University of Technology är ett 70 år gammalt universitet i nordöstra Polen och kan anses vara en seriös institution. Förlaget som publicerat artikeln är Sciendo, som är dotterbolag till det väletablerade och vetenskapliga förlaget De Gruyter. . Denna journal kan därmed anses pålitlig och seriös. Artikeln handlar om hur flaskhalsanalys kan gå till vid implementering av TOC (Theory of Constraints). Då denna källa användes för att bilda sig en uppfattning och få reda på information kring just flaskhalsanalys anses källan har relevant innehåll i relation till dess användning. Artikelns målgrupp är andra akademiker och arbetare som arbetar med produktion. Källans syfte är att informera och presentera slutsatser som författarna har kunnat dra kring ämnet flaskhalsanalys. Artikeln är publicerad år 2020 och kan därför anses vara aktuell.

7.2 Källor

[1] Aimpoint. ”Vår historia.” Aimpoint.com. [Online]
Tillgänglig: https://www.aimpoint.com/se/om-oss/vaar-historia/
[Hämtad: maj. 5, 2024]

[2] Aimpoint. ”Våra kärnvärden.” Aimpoint.com. [Online]
Tillgänglig: https://www.aimpoint.com/se/om-oss/vaara-kaernvaerden/
[Hämtad: maj. 5, 2024]

[3] M. Grieves och J.Vickers, ”Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior” i *Transdisciplinary Perspectives on Complex System*, J.Kahlen, S.Flumerfelt, A.Alves. Cham: Springer, 2017, 85-113.

[4] M. Grieves, "Virtually Intelligent Product Systems: Digital and Physical Twins," i *Complex Systems Engineering: Theory and Practice*, S.Flumerfelt, et al. American Institute of Aeronautics and Astronautics, 2019, 175-200.

[5] ABB. *Operating manual RobotStudio*, AS ed. (2024). [Online]
Tillgänglig:
https://library.abb.com/r?dkg=dkg_instructions%20and%20manuals&q=robotstudio
[Hämtad: maj. 8, 2024]

[6] C. Taesi, F. Aggogeri, N.Pellegrini, ”COBOT Applications – Recent Advances and Challenges,” *Robotics*, vol.12, no.3, 79, juni 2023. [Online]
Tillgänglig: https://doi.org/10.3390/robotics12030079
[Hämtad: maj. 8, 2024]

[7] ABB, *SWIFTI - CRB 1100 datasheet*, (2021). [Online]
Tillgänglig: https://new.abb.com/products/robotics/robots/collaborative-robots/swifti/crb-1100-swifti
[Hämtad: maj. 9, 2024]

[8] W. Urban, P. Rogowska, ”Methodology for bottleneck identification in a production system when implementin TOC,” *Sciendo*, vol.12, no.2, 74–82, juli 2020. [Online]
Tillgänglig: https://doi.org/10.2478/emj-2020-0012
[Hämtad: maj. 16, 2024]