

# Industriell robots energiförbrukning för olika val av rörelseparametrar

Simulerad förflyttning av tunga föremål i  
RobotStudio



LUNDS  
UNIVERSITET

Lunds Tekniska Högskola

LTH vid Campus Helsingborg

Avdelningen för industriell elektroteknik och automation

Examensarbete:

Alfred Jonasson

Sihe Chen



© Copyright Alfred Jonasson, Sihe Chen

LTH vid Campus Helsingborg

Lunds universitet

Box 882

251 08 Helsingborg

LTH at Campus Helsingborg

Lund University

Box 882

SE-251 08 Helsingborg

Sweden

Tryckt i Sverige

Lunds universitet

Lund 2026

## **Sammanfattning**

Detta arbete undersöker en simulerad robotarm som förflyttar tunga produkter mellan två rullband på en industriell produktionslinje. Simuleringarna utförs med en ABB-robot i programmet RobotStudio. Examensarbetet har två

huvudmål: 1. Att minska upprepade tunga lyft för personalen. 2. Öka effektiviteten i produktionen och minimera robotens energikonsumtion.

Målen uppfylldes genom att en produktionslinje modellerades i RobotStudio. Sedan definierades robotens beteende med en blandning av programmeringsspråket RAPID och signaler från logiska block, så kallade Smart Components.

Simulering påbörjades sedan med ett rullande rullband med produkter ovanpå som roboten förflyttade till ett annat rullband. Energiförbrukning och cykeltid registrerades och jämfördes för olika värden på parametrarna hastighet, acceleration och accelerationsramp.

Resultatet visar att hastighet, acceleration och accelerationsramp har en betydande inverkan på en robots energiförbrukning. Högre hastigheter, acceleration och accelerationsramp resulterar i kortare cykeltider men högre energiförbrukning och vice versa.

Resultatet indikerar att det, genom simulering, kan finnas möjligheter att optimera energiförbrukning av industrirobotar som används på produktionslinjer.

Nyckelord: automation, industriella robotar, energiminskning, simulering, RobotStudio

## **Abstract**

This paper studies a simulated robotic arm that moves heavy products between two conveyor belts on an industrial production line. The simulations are performed using an ABB robot in the program RobotStudio. The degree project

has two main goals; 1. To reduce repeated heavy lifting by employees. 2. Increases production efficiency and minimises the robot's energy consumption.

The goals were fulfilled by modelling a production line in RobotStudio. Then, the robot's behaviour was defined by a mix of the programming language RAPID and signals from block logic, so called Smart Components.

Simulation then commenced with a moving conveyor belt with products on top that the robot moved to another conveyor belt. Energy consumption and cycle time got registered and was compared for different values of the parameters velocity, acceleration and acceleration ramp.

The results show that velocity, acceleration and acceleration ramp have a considerable impact on the robot's energy consumption. Higher velocities, acceleration and acceleration ramp result in shorter cycle times but higher energy consumption, and vice versa.

The results indicate that there may be opportunities to optimise energy consumption of industrial robots that are used on production lines through simulation.

Keywords: automation, industrial robots, energy reduction, simulation, RobotStudio

## **Förord**

Detta arbete skrivs 2026 av två studenter som studerar till högskoleingenjör i datateknik respektive elektroteknik med automation, med viljan att få

kombinera deras kunskaper i ett projekt som kan bidra till att lösa utmaningar i ett verkligt scenario.

Projektet omfattar både kunskaper inom automation och programmering, samtidigt som det kräver en viss förståelse för hur industrirobotar fungerar och används i praktiken.

I början funderade vi på att bygga en produktionslinje och använda PLC-programmering. Vi saknade däremot fysisk utrustning för att kunna testa en sådan produktionslinje. Vi övervägde också ett mer allmänt fokus på programmering, men lyckades till slut hitta en bra avvägning mellan våra programinriktningar.

Vi vill tacka vår handledare som tack vare sin erfarenhet hjälpt oss att hitta ett examensarbete vi kunde utföra när inget företag erbjöd oss ett examensarbete. Handledarens hjälp har varit mycket uppskattad, med tips och återkoppling kring hur projektet kan vidareutvecklas och förbättras.

Vi vill även tacka vår examinator som har hjälpt oss att kritiskt granska examensarbetet för att det ska uppnå en hög akademisk standard.

Även fast studenterna snart tar sin högskoleingenjörsexamen kommer de att minnas sin studietid och mötet med studenter och lärare. Vi vill tacka LTH och Lunds universitet för studietiden.

## **Innehållsförteckning**

|   |           |
|---|-----------|
| <b>1. Inledning</b>                                 | <b>1</b>  |
| 1.1 Bakgrund  | 1         |
| 1.2 Syfte   | 2         |
| 1.3 Målformulering                                  | 3         |
| 1.4 Problemformulering                              | 3         |
| 1.5 Motivering av examensarbetet                    | 4         |
| 1.6 Avgränsningar                                   | 8         |
| 1.7 Arbetsfördelning                                | 10        |
| <b>2. Teknisk bakgrund</b>                          | <b>13</b> |
| 2.1 Allmänna begrepp                                | 13        |
| 2.2 Smart Components och Station Logic              | 16        |
| 2.3 RAPID   | 18        |
| <b>3. Metod</b>                                     | <b>23</b> |
| 3.1 Initialt val av arbete                          | 23        |
| 3.2 Inledning och val av slutgiltigt arbete         | 24        |
| 3.3 Genomförande av slutgiltigt arbete              | 30        |
| 3.3.1 Modellering av produktionslinje i RobotStudio | 30        |
| 3.3.2 I/O-konfiguration och Station Logic           | 33        |
| 3.3.3 RAPID   | 35        |
| 3.3.4 Mätningmetod                                  | 37        |
| 3.3.5 Behandling av mätdata för analys              |           |
| 3.3.6 Kommunikation och samarbete                   |           |
| 3.4 Manuell informationssökning                     | 39        |
| 3.5 Användning av generativ AI                      | 40        |
| 3.6 Metoddiskussion                                 |           |
| 3.7 Källkritik                                      | 41        |
| <b>4. Analys</b>                                    | <b>44</b> |
| 4.1 Val av metod                                    | 44        |
| 4.2 Analys av resultat                              | 45        |
| 4.3 Förbättringsmöjligheter                         | 51        |
| <b>5. Resultat</b>                                  | <b>53</b> |
| 5.1 Excel och mätning                               |           |
| 5.2 Diagram   |           |
| <b>6. Slutsats</b>                                  | <b>59</b> |
| 6.1 Diskussion                                      | 59        |
| 6.2 utmaning  |           |
| 6.3 Mål och problem                                 |           |
| 6.4 Reflektion över etiska aspekter                 | 62        |
| 6.5 Slutsatser                                      | 64        |
| <b>7. Källförteckning</b>                           | <b>65</b> |

|                    |           |
|--------------------|-----------|
| <b>8. Appendix</b> | <b>72</b> |
| <b>Appendix A</b>  | <b>72</b> |
| <b>Appendix B</b>  | <b>73</b> |
| <b>Appendix C</b>  | <b>75</b> |

# 1. Inledning

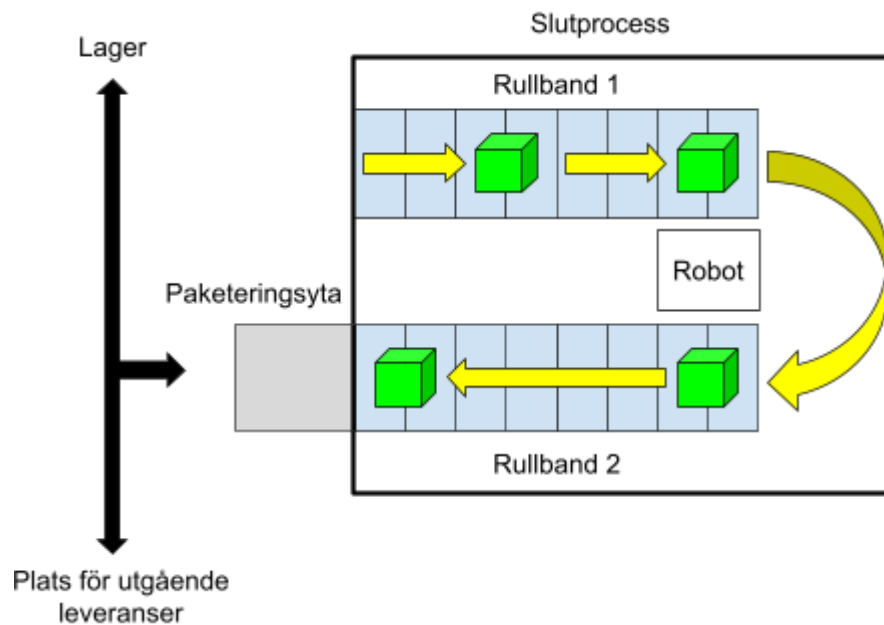
Detta kapitlet ger en bakgrund till vad examensarbetet handlar om, dess syfte, mål och vilka problem det försöker besvara. Kapitlet motiverar även examensarbetet och förklarar vilka avgränsningar som studenterna valt att göra. Slutligen beskrivs det hur arbetsuppgifterna har fördelats mellan de två studenterna.

## 1.1 Bakgrund

Detta projekt har sitt ursprung ur en tankeväckande och givande diskussion mellan studenterna och deras handledare. Arbetet utgår från ett fiktivt företag som driver en fabrik med en produktionslinje. Den sista delen av produktionslinjen är identisk med figur 1, förutom att personal arbetar där istället för en robot. Produktionslinjen består av två rullband som färdigtillverkade produkter transporteras på. Produkterna, som har en vikt på 40 kg styck, förflyttas av personal från det första rullbandet till det andra. Vid slutet av det andra rullbandet faller produkterna ner i en kartong som står på paketeringsytan. När kartongen är fylld kan den transporteras till fabriken lager med en gaffeltruck.

Uppdragsgivaren föreslår att en industriell pick-and-place-robot ska flytta produkterna mellan rullbanden istället för personal, för att minska mängden fysiskt krävande arbetsmoment personalen har. Produktionslinjen skulle då stämma överens med figur 1, Uppdragsgivaren föreslår istället att personalen kan fokusera på att transportera färdigpaketerade produkter från paketeringsytan till lagret samt från lagret till platsen för utgående leveranser med en gaffeltruck. Uppdragsgivarens förhoppning är att en sådan förändring ska ge personalen mer tid att hinna med utgående leveranser och mer tid till att utföra kvalitetskontroll. Detta skulle dels kunna förbättra ledtiden för leveranser, dels minska antalet defekta produkter som når slutkund. Detta skulle öka uppdragsgivarens effektivitet och göra deras produkt mer attraktiv för kunder.

Figur 1 visar en bild över produktionslinjens sista delar med uppdragsgivarens föreslagna förändring.



**Figur 1.** Bild över produktionslinjens sista delar, som utgörs av slutprocessen och därefter paketeringsytan, varifrån färdiga produkter transporteras vidare med gaffeltruck. En robot förflyttar tillverkade produkter mellan rullbanden istället för, som det var ursprungligen, personalen.

Uppdragsgivaren behöver samtidigt en kostnadseffektiv lösning för att se den som lönsam på lång sikt. Därför ska examensarbetet försöka minska långsiktiga kostnader för elförbrukning.

## 1.2 Syfte

Detta delkapitel förklarar det övergripande syftet och avsikten med examensarbetet.

Ett av syftena med arbetet är att minska tunga lyft som personalen behöver utföra för att förflytta produkter mellan rullbanden. Förhoppningen är att detta ska förbättra arbetsmiljön och minska risken för arbetsrelaterade skador.

Det andra syftet med examensarbetet är att minska energiförbrukningen som den tilltänkta industriroboten använder på lång sikt. Förväntat resultat är en lösning med en robot som uppfyller fabriken krav men samtidigt ger så låg energiförbrukning som möjligt.

## 1.3 Målformulering

Ett mål med detta projekt är att utifrån ett begränsat urval av värden på några parametrar som roboten använder: hastighet, acceleration och accelerationsramp, identifiera de värden som ger minst energiförbrukning när industriroboten i figur 1 utför sin uppgift.

Detta mål bidrar till syftet då det skapar ett beslutsunderlag, om än begränsat, för fabriken att ställa in parametrarna hastighet, acceleration och accelerationsramp för att minska onödig energiförbrukning. Fabriken kan då välja parametrar som ger minimal energiförbrukning men fortfarande uppfyller krav på en viss cykeltid.

Ytterligare ett mål är att kunna automatisera slutprocessen, som markeras av rektangeln med markerade svarta kanter i figur 1, för att personalen istället ska kunna lägga mer tid på att transportera kartonger med hjälp av gaffeltruck och utföra kvalitetskontroll innan produkterna säljs. Målet att automatisera slutprocessen bidrar till arbetets syfte genom att målet, om det uppfylls, tar bort ett fysiskt krävande arbetsmoment för personalen.

## **1.4 Problemformulering**

Studenterna vill försöka besvara följande frågor, med slutprocessen i figur 1 som utgångspunkt:

1. Hur påverkar parametrarna hastighet, acceleration och accelerationsramp cykeltid och robotens energiförbrukning?
2. Stämmer resultaten överens med slutsatser från vetenskapliga artiklar som gjort liknande undersökningar?

## **1.5 Motivering av examensarbetet**

Arbetet motiveras bland annat med behovet att förbättra arbetsmiljön för de anställda i fabriken. Det nuvarande arbetssättet kräver att fabriken personal lyfter tunga produkter för hand och flyttar dessa mellan rullbanden på produktionslinjen. Studenterna och företaget ser risker med detta.

Risken för smärta i den nedre delen av ryggen ökar när en allt större del av arbetstiden läggs på tunga lyft [1]. Speciellt upprepade tunga lyft i kombination med vridning av kroppen, som är en av personalens arbetsuppgifter, har identifierats som en riskfaktor för ryggsmärta [2].

Den totala kostnaden för samhället på grund av sjukdomar orsakade av ryggont beräknades till 22,5 miljarder kronor/år i Sverige på 1990 års prisnivå, inkluderat både direkta och indirekta kostnader. Ungefär 95 % av denna kostnad uppstod utanför sjukvården [3].

Arbetsmiljöproblem har inte bara kortsiktiga konsekvenser. Smärta i både över- och underkroppen har kopplats till sjukfrånvaro och permanent oförmåga att arbeta. Hög mekanisk exponering på arbetet ökar denna risk ytterligare [4]. Belastningsskador är dessutom svåra att behandla och svåra att bli återhämtade från [5] och arbetsrelaterade muskuloskeletal sjukdomar är en av de stora orsakerna till tidig pensionering [6]. Lyftningsmomentet tros därför ha en negativ inverkan, både för personalen och arbetsgivaren.

Av de arbetsskador under 1985 som anmäldes till informationssystemet om arbetsskador (ISA) var 30 % belastningsskador. Vid en studie genomförd av Arbetsmiljöinstitutet undersöktes 181 skadeanmälningar närmare med arbetsplatsbesök, enkätundersökning och registerutdrag från Försäkringskassans handläggning av de aktuella ärendena [7].

Vid samtliga genomförda arbetsplatsbesök bedömdes arbetstagaren ha utsatts för belastning på jobbet. Enkätundersökningen genomfördes tre år efter anmälan och bland de som vid skadetillfället hade arbete inom tillverkningsarbete, maskinskötsel med mera, svarade cirka en fjärdedel att de lämnat sitt arbete [7].

Arbetsmiljöverket skriver i en rapport år 2024, som främst baseras på telefonintervjuer av personer i åldern 15-74 år som en del av Arbetskraftsundersökningarna (AKU) [8], att 6 % av sysselsatta upplevt besvär på grund av olyckshändelse på arbetsplatsen under det senaste året. Hälften av dessa svarar att varken de själva eller arbetsgivaren anmält olyckan eller olyckorna som orsakat besvären till Försäkringskassan [9]. Mörkertalet innebär att konsekvenserna för olyckor kopplade till tunga lyft i Sverige kan vara högre än vad officiell statistik pekar mot.

I en undersökning med data från Guangdong-provinsen i Kina visades att användning av industriella robotar kan förbättra arbetsmiljön och på så sätt minska antalet arbetsskador [10]. Arbetsskador ses som en drivande faktor till att fler industriella robotar används, bland annat kopplat till arbetsskadornas belastning på den offentligt finansierade sjukvården [10].

Utifrån ovanstående information anser studenterna att det finns möjligheter att förbättra arbetsmiljön i fabriken och minska sjukfrånvaro och tidig pensionering genom att använda en industrirobot på produktionslinjen. Detta skulle vara fördelaktigt för både arbetstagarna och arbetsgivaren, arbetstagarna släpper mindre inkomst på grund av sjukfrånvaro och arbetsgivaren får lättare att bemanna fabriken och möjligtvis minskad personalomsättning.

Det finns även motiv till att göra en grundlig genomgång av olika val fabriken kan göra i samband med att de driftsätter en industrirobot. Att planera för, köpa in, driftsätta och underhålla en industriell robot skulle innebära en uppenbar kostnad för företaget. Givet att elpriset i Sverige har varierat kraftigt mellan 2014 och 2025 för elkunder som inte är hushåll [11], så antas optimerad energiförbrukning kunna göra stor ekonomisk skillnad för företaget.

Tillverkningsindustrin stod för 37 % av den totala energiförbrukningen år 2018 [15]. Robotar förbrukar ungefär 8 % av den elektriska energin som går åt vid produktionsprocesser [16]. En genomgång av data från 38 länder och 17 olika tillverkningssektorer har visat att användning av industriella robotar kan minska energiintensiteten [15], som beskriver mängden tillförd energi per enhet av BNP [17].

Tidigare har inte minskning av energiförbrukning från robotar och automatiska system varit ett designuppdrag [12]. Utformning och användning av dessa system utgör en stor möjlighet för minskad energianvändning inom industrin [13]. Satsningar för att minska energiförbrukningen hos industriella robotar och automationssystem ökar i snabb takt, med klimatförändringar och ekonomiska aspekter som drivande faktorer [14].

Robotar och maskiner körs ofta för att maximera produktionshastigheten. Detta leder till både energiförluster och energiöverskott vid acceleration respektive deceleration [13]. Industriella robotar använder vägar som inte är optimala ur ett energiperspektiv och en liten ökning av cykeltiden kan möjliggöra markanta energibesparingar [12].

Det verkar alltså finnas stora möjligheter att minska robotens energiförbrukning om den används på ett optimalt sätt och denna studie kan bidra till en ökad förståelse för hur olika parametrar samspelar, vilket även är användbar information för andra industriföretag och robottillverkare. Det finns andra vinster med användning av industriella robotar, som ökad effektivitet [15], [18] och produktivitet och minskad risk för fel [18], vilket ökar företagens

konkurrenskraft på marknaden och kan motivera användningen av roboten istället för personal för den specifika arbetsuppgiften.

RobotStudio skapar automatiskt tidsoptimala vägar för uppgifter som kräver att roboten rör sig [16] men mjukvara för att utforma robotceller och robotens arbetsyta saknar funktioner för att optimera energiförbrukningen på ett tidigt stadium i designprocessen [19]. Därför bedömer studenterna att det krävs ytterligare arbete för att uppnå en lösning som förbrukar mindre energi än de förvalda inställningarna för robotens rörelser som ställs in automatiskt i RobotStudio, som studenterna tänkt att använda för att simulera roboten.

En artikel drar slutsatsen att robotens hastighet varken ska vara för låg eller för hög för optimal energiförbrukning [20] medan en annan kommer fram till att hastigheten bör sättas mellan 50 % och 100 % för optimal energiförbrukning [21].

Samtidigt förekommer olika syn på accelerationens påverkan på energiförbrukningen. [13] ser att acceleration leder till energiförlust och beskriver accelerationen som en av de vanligaste metoderna för att optimera energiförbrukning [19] medan en annan studie visade att acceleration hade en försumbar påverkan på energiförbrukningen [21].

På grund av att olika studier har dragit olika slutsatser så anser studenterna att det vore värdefullt med ytterligare studier inom området. Det förekommer dessutom skillnader mellan olika robotar beroende på skillnader i deras konstruktion [19] som det idealt bör tas hänsyn till och studenterna har inte hittat någon studie med den robot som är tänkt att användas i fabriken.

En ytterligare motivering till detta arbete är behovet av tillgång till data. Många tidigare studier diskuterar energieffektivisering och gör mätningar men redovisar inte alla mätdata. Mer fullständiga mätdata skulle krävas för att avgöra vilken lösning som är bäst lämpad för produktionslinjen i figur 1 och företagets potentiella krav på cykeltid.

Arbetet är även relevant då det finns goda möjligheter till vidare arbete. Bland annat kan fler hastigheter testas. Dessutom förväntas linjära och olinjära rörelser av TCP ha olika energiförbrukning [21] och kan därför jämföras. Placeringen av roboten kan påverka

energiförbrukningen [12], [19] och även avståndet från roboten till det objekt som roboten ska arbeta med [19].

## 1.6 Avgränsningar

Arbetet kommer endast att utföra simuleringar och inga tester med riktiga robotar i någon verklig produktionsmiljö. Det saknas ekonomiska resurser och testmiljö för detta inom ramen för arbetet.

De tillverkade produkterna som rullar på rullbanden är i samtliga simuleringar av en och samma typ. Endast slutprocessen i figur 1 kommer att simuleras. Detta då denna del av tillverkningsprocessen i och med roboten blir helt automatiserad och deterministisk.

Givare som används för att detektera produkter i simuleringen kommer inte att vara riktiga givare utan kommer istället utgöras av simulerade signaler som implementeras med hjälp av Smart Components och Station Logic i RobotStudio.

Arbetet undersöker bara två olika lastvikter på roboten: en för endast verktyget och en lastvikt för både verktyg och produkt. I verkliga tillämpningar kan lastvikt ha en betydande påverkan på robotens energiförbrukning [21].

I denna virtuella miljö beaktar och antar experimentet endast roboten funktion under normala arbetsförhållanden, och tar inte hänsyn till slitage eller extrema och plötsliga situationer som strömavbrott eller spänningsfall.

Slutligen behandlar arbetet inte ekonomiska analyser av att roboten används istället för personal för den specifika arbetsuppgiften. Här finns det både engångskostnader för förstudier och driftsättning av systemet men även återkommande kostnader för energiförbrukning och underhåll. Dessa kostnader måste sedan vägas mot vinsten i produktivitet, effektivitet och personalens arbetsmiljö.

Projektet kan inte heller på grund av tidsbegränsningar och arbetets omfattning ta fram någon utförlig metod för fullständig optimering av parametrar som studenterna undersöker. Detta skulle sannolikt, utifrån studier av vetenskapliga artiklar, kräva omfattande statistisk analys

och framtagning av energi- eller kostnadsfunktioner [14], [16] eller modeller av systemet och identifiering av olika parametrar [12], [16], [19], [20], [21], [22].

## 1.7 Arbetsfördelning

En översikt över hur examensarbetet har fördelats mellan studenterna visas i tabell 1 med ytterligare kommentarer längre ner.

**Tabell 1.** Arbetsfördelning av examensarbetet efter arbetsområde.

|                                       | Utfört arbete (%) |           |
|---------------------------------------|-------------------|-----------|
|                                       | Alfred Jonasson   | Sihe Chen |
| <b>Arbete i RobotStudio</b>           |                   |           |
| Modellering av produktionslinjen      | 70                | 30        |
| Programmering                         | 80                | 20        |
| Insamling av mätdata genom simulering | 60                | 40        |
| <b>Rapportskrivning</b>               | 60                | 40        |
| Inledning                             | 50                | 50        |
| Teknisk bakgrund                      | 70                | 30        |
| Metod                                 | 70                | 30        |
| Analys                                | 20                | 80        |
| Resultat                              | 50                | 50        |
| Slutsats                              | 30                | 70        |
| <b>Övrigt arbete</b>                  | 50                | 50        |
| Informationssökning                   | 50                | 50        |
| Poster                                | 30                | 30        |
| Presentation                          | 30                | 70        |

## **Inledande fas**

I denna fas fördelades arbetet ungefär lika mellan studenterna. Studenterna arbetade tillsammans för att skapa en grundläggande förståelse för RobotStudio och programmets funktioner. Fokus låg på centrala verktyg för modellering av en robotcell samt användning av Smart Components, Station Logic och RAPID.

Under denna fas genomfördes övningar och tester med avgränsade mål, till exempel: 1. Skapa ett rullband som går att starta och stoppa. 2. Styr rullbandet med en givare. 3. Justera roboten till en viss ställning.

Studenterna började så småningom förstå hur robotens rörelser kan styras och hur roboten kan interagera med produkterna på produktionslinjen. Syftet med denna fas var att ha grundförutsättningarna för att sedan kunna skapa en sammanhängande fungerande lösning som representerar en enkel produktionslinje.

## **Modellering och programmering**

Till en början arbetade Alfred Jonasson med att modellera utrustningen i fabriken, som bestod av roboten, rullbanden samt givare. Sihe Chen modellerade samtidigt, med hjälp av verktyget Autodesk Fusion 360, produkterna som skulle produceras. Produkterna modellerades som räbblock med två cylinderformade insättningar på ovansidan.

Efter att utformningen av slutprocessen ändrades till en enklare process utan ytterligare montering av delar på produkterna så ersattes dock denna modell med enklare geometri utan några insättningar som skapades direkt i RobotStudio.

Studenterna spenderade ungefär lika stor tid med att lära sig RobotStudio och experimentera. Den slutgiltiga slutprocessen modellerades huvudsakligen av Alfred Jonasson efter förslag och diskussioner med Sihe Chen.

Alfred Jonasson tog huvudansvar för arbetet med RAPID, Smart Components och Station Logic. Trots detta var Sihe Chen mycket delaktig och stöttande i denna delen av arbetet, särskilt vid problemlösning och när tekniska utmaningar eller oklarheter uppstod. Sihe Chen hjälpte också till att översätta pseudokod till RAPID-kod.

## **Mätningar**

Denna delen fördelades ungefär lika mellan studenterna efter att Alfred Jonasson hittat en mätningssmetod som underlättade mätningarna och minskade risken för att tidigare mätningar påverkade efterföljande mätningar.

## **Analysfas**

Under analysfasen arbetade studenterna gemensamt med att tolka och diskutera resultaten från simuleringarna. Sihe Chen ansvarade huvudsakligen för bearbetning och visualisering av insamlad data, med förslag och återkoppling från Alfred Jonasson.

## **Rapportskrivningsfas**

Rapportskrivningen genomfördes gemensamt av båda studenterna. Båda deltog i att skriva, granska och revidera samtliga delar av rapporten. Arbetet innefattade att formulera tydliga texter, strukturera innehåll och att säkerställa språklig precision som uppfyller akademisk standard.

Sihe Chen arbetade mycket med att starta upp skrivandet och att ge rapporten en tidig struktur. Studenterna fördelade skrivandet efter vad de arbetat mest med. Speciellt mot slutet av arbetet läste studenterna rapporten många gånger för att ha en god förståelse för arbetet och gjorde justeringar för att säkerställa en enhetlig, tydlig och sammanhängande text.

## **Poster och presentation**

Poster och presentation skrevs huvudsakligen av Sihe Chen med förslag och justeringar av Alfred Jonasson gällande innehåll, fokus och grafisk formgivning av postern.

## 2. Teknisk bakgrund och terminologi

Målet med följande kapitel är att förklara några av de begrepp och verktyg som används i examensarbetet för att ge en ökad förståelse för rapportens kommande delar. I vissa fall markeras svenska översättningar till termer inom parentes, om studenterna gjort en egen översättning av termen som används senare i rapporten. Kapitlet delas in i tre delar: allmänna begrepp, Smart Components och Station Logic samt avslutningsvis RAPID.

### 2.1 Allmänna begrepp

Följande delkapitel beskriver några generella begrepp inom robotik och några begrepp och verktyg i RobotStudio.

#### **Industrirobot**

En automatiskt styrd, omprogrammerbar robotarm med flera användningsområden. Roboten har tre eller fler axlar och används inom industriell miljö [23].

Industrirobotar består vanligtvis av flera huvudkomponenter: en mekanisk arm med flera leder, elektriska motorer som driver rörelserna och en styrenhet [18]. Dessa komponenter samverkar för att roboten ska kunna utföra sina uppgifter. Industrirobotars energiförbrukning kan påverkas av många faktorer, till exempel hastighet, acceleration, lastvikt, rörelsevägar och friktion i robotens leder [21].

#### **Cobot - collaborative robot (kollaborativ robot)**

En robot som är utformad för att på ett säkert sätt arbeta tillsammans med människor [Kumar]

#### **Robotcell**

Ett system som består av minst en robot och övrig utrustning och verktyg. Systemet är utformat för att utföra specifika uppgifter, till exempel materialhantering inom bland annat tillverkning [24].

#### **Styrenhet**

Uppsättning hård- och mjukvara som möjliggör styrning av en robots beteende [23].

### **Virtuell styrenhet**

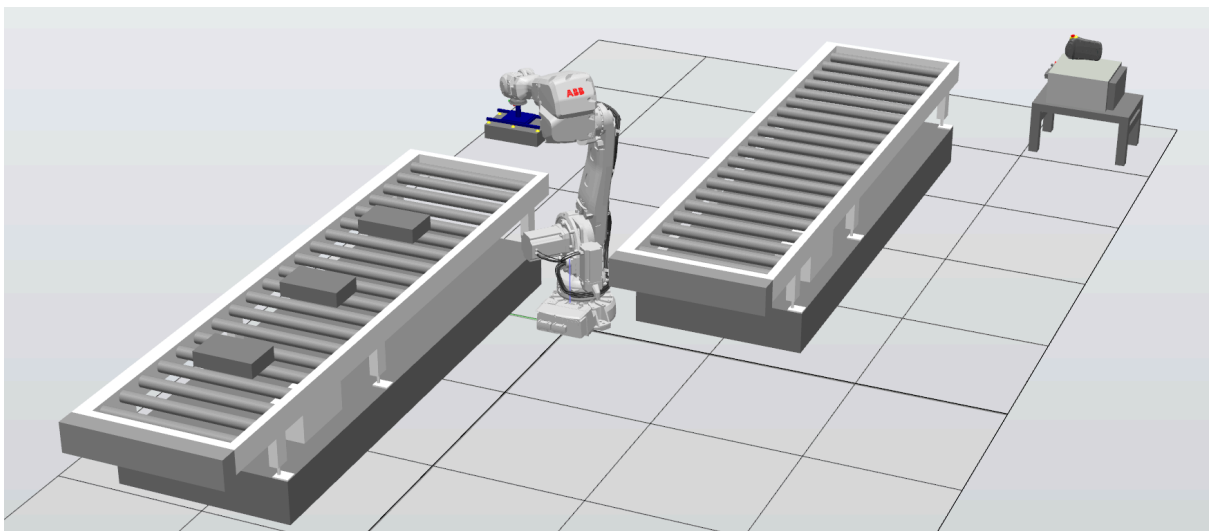
Mjukvara som efterliknar den riktiga styrenheten [25].

### **Cykeltid**

Cykeltiden är tiden som spenderas med en produkt vid en viss arbetsstation under en produktionsprocess [26].

### **RobotStudio**

RobotStudio är ett verktyg som bland annat kan användas för att modellera robotceller, programmera ABB-robotar och utföra simuleringar [25]. Robotceller kan visualiseras i en tredimensionell vy som användaren kan panorera, rotera och zooma in och ut i [25]. Denna vy, med projektets robotcell som exempel, visas i figur 2.



**Figur 2.** Vy över projektets robotcell medan simulering pågår.

### **ABB Smart Gripper**

Mekanism som kan ha två gripklor vars avstånd kan justeras [27]. Smart Gripper kan fästas till robotarmen och sedan användas för att greppa tag om föremål.

### **RobotWare**

En uppsättning mjukvaror som används till att konfigurera en styrenhet [25].

### **Pack and Go**

En funktion i RobotStudio som kan skapa ett paket av en robotcell i filformatet .rspag. Filen innehåller använda bibliotek och virtuella styrenheter. Filen kan sedan delas för att låta andra personer ta del av projektet [25].

### **World Coordinate System - WCS**

Ett koordinatsystem som används i representationen av robotcellen och som är högst upp i hierarkin av olika koordinatsystem i ett projekt. Alla övriga koordinatsystem relateras till World Coordinate System [25].

### **I/O System**

Del av programmet RobotStudio som används för att konfigurera en styrenhets in- och utgångar [28].

### **FlexPendant**

En enhet med skärm som bland annat kan användas till att styra robotarmen samt köra och skriva program [29].

### **Signal Analyzer**

Verktyg i RobotStudio som används för att samla in och lagra signaler från en robot och dess virtuella styrenhet [25]. De insamlade signalerna kan sedan studeras för vidare analys.

### **TCP - Tool Center Point**

Är den punkt som robotens position bestäms relativt till. TCP är origo till koordinatsystemet för verktyget och används när roboten utför sina uppgifter. Verktygets egna koordinatsystem definierar i sin tur verktygets position och vridning [25]. TCP för ett par gripklor kan till exempel vara positionen mittemellan de två gripklorerna [30].

### **Target (mål)**

Är ett objekt i RobotStudio som används för att spara information om en position och vridning av robotens TCP. Ett Target kan synkroniseras till RAPID-kod och lagras då i form av datatypen `robtarget` [25] (se kapitel 2.3 RAPID).

### **Path**

Ett rörelsemönster som en robot utför med rörelseinstruktioner som exekveras i sekventiell ordning, där varje stoppunkt längs vägen beskrivs av ett Target [25].

### **User library (användarbibliotek)**

Biblioteksfiler som har importerats till RobotStudio [25].

### **PLC - Programmable Logic Controller**

En typ av dator med in- och ut signaler som används på bred front i industriella styrsystem. En PLC arbetar cykliskt: insignalen läses av, logik exekveras och ut signaler ställs in [31] baserat på tillstånd och logik.

## **2.2 Smart Components och Station Logic**

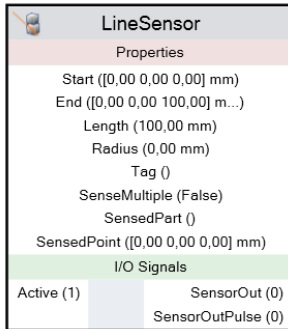
Detta delkapitel beskriver Smart Components och dessa hänger ihop med Station Logic i RobotStudio.

### **Smart Components (smarta komponenter)**

Smart Components är block med bland annat ingångar, utgångar och egenskaper, som kan användas för att simulera olika komponenter i RobotStudio, med undantag för den virtuella styrenheten [25].

Det finns inbyggda Smart Components, så kallade Base Smart Components, som bland annat kan användas för att simulera givare och deras signaler, förflyttning av objekt [25] samt logiska grindar. Användare kan skapa egna Smart Components och dessa kan i sin tur bestå av flera inkapslade Smart Components [25]. På detta vis kan mer avancerad logik abstraheras.

Ett exempel på en inbyggd Smart Component, kallad LineSensor, visas i figur 3. Givaren utgörs av en linje och är aktiverad när den digitala signalen Active är hög. När ett föremål korsar linjen så blir den digitala signalen SensorOut hög [25], vilket tolkas som att givaren detekterar ett föremål.



**Figur 3.** Den inbyggda smarta komponenten LineSensor med olika egenskaper, bland annat Start, End, Radius och SensedPart. LineSensor har en digital insignal Active och de två digitala utsignalerna SensorOut och SensorOutPulse.

Nedan beskrivs några Smart Components. För detaljer om deras signaler, egenskaper och användningsområden hänvisar studenterna till RobotStudios användarmanual [25].

### Timer

Genererar, när den är aktiverad, en klockpuls. Om egenskapen Repeat är satt till sann så upprepas klockpulsen med intervallet Interval mellan pulserna [25].

### Source

En Smart Component som klonar ett specifikt objekt [25].

### Positioner

Kan användas till att flytta och rotera ett specificerat objekt till en ny position [25].

### Queue

Representerar en FIFO-kö av objekt [25] som kan manipuleras som en grupp [32].

### LinearMover

Förflyttar ett specificerat objekt i en viss riktning med en angiven hastighet [25].

### PoseMover

Kan användas för att simulera förflyttning av en robot [25] till en viss ställning [33].

## Attacher

Kan användas för att fästa ett föremål till ett annat objekt [25].

## Detacher

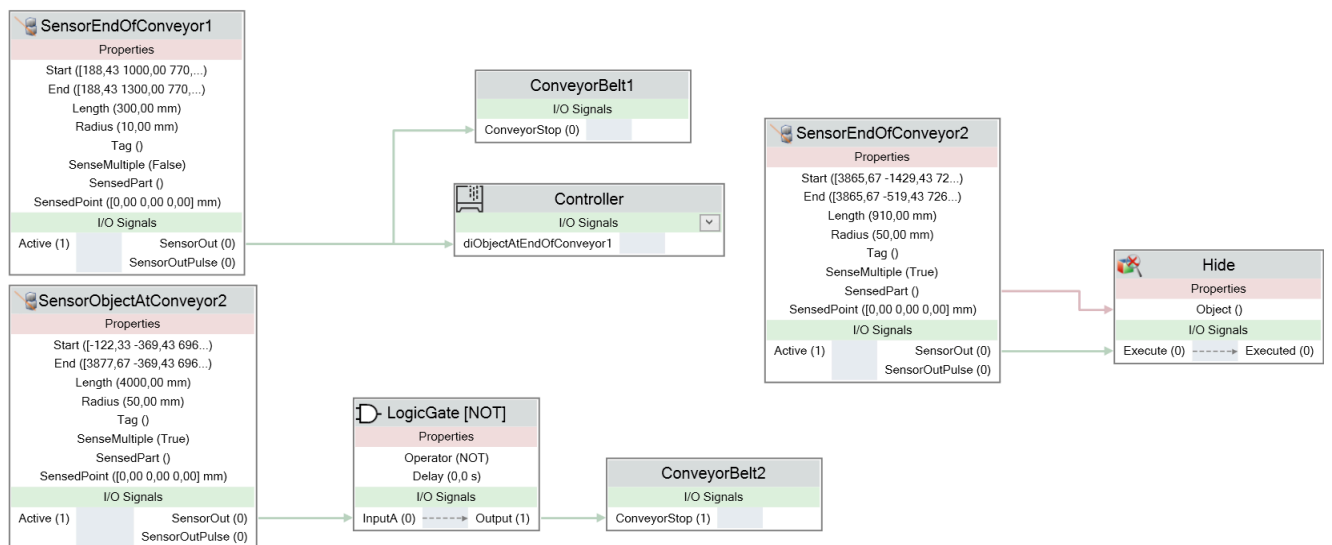
Kan användas för att lossa ett fäst föremål från objektet det är fäst till [25].

## ClosestObject

Kan utifrån ett bestämt objekt eller en bestämd punkt hitta närmaste objekt, del och avstånd [25].

## Station Logic

Station Logic är en vy där det kan bestämmas hur alla Smart Components, den virtuella styrenheten och övriga så kallade RobotStudio-objekt som en station består av, kopplas samman [25]. Ett exempel på hur Station Logic kan se ut visas i figur 4.



**Figur. 4** Station Logic med en virtuell styrenhet, tre givare, två rullbandsmekanismer, en NOT-grind och en Smart Component kallad Hide som döljer objekt som når slutet av det andra rullbandet. Signaler kopplas samman via anslutningar som markeras med gröna pilar. Egenskaper kopplas samman via bindningar som markeras med röda pilar [25].

## 2.3 RAPID

Denna delen förklarar vad RAPID är och ger exempel på nyckelord för vanliga åtgärder som deklarerering av variabler, konstanter och funktioner utan returvärde. Delkapitlet förklarar även hur flödeskontroll och iterering kan utföras. Några instruktioner och datatyper som används i

RAPID för att styra en robots rörelser och styrenhetens digitala ut signaler introduceras också. Studenterna rekommenderar att läsare läser denna delen innan de studerar appendix B.

RAPID är ett programmeringsspråk som används för styrenheter från ABB [25]. RAPID kan bland annat användas för att förflytta roboten och ställa in utgångar på styrenheten [30]. RAPID har inbyggda datatyper som bland annat `num`, `bool` och `string` för tal, booleska värden och strängar [33]. Data kan deklarerars som en variabel med nyckelordet `VAR`, ett bestående värde som lagras och uppdateras mellan körningar av programmet med nyckelordet `PERS` och som en konstant med nyckelordet `CONST` [30]. Programflödet kan bland annat styras med nyckelorden `IF`, `FOR`, `WHILE` och genom funktionsanrop [30].

### **Procedure**

Är en funktion/rutin som kan anropas i RAPID men som inte har ett returvärde. Deklareras med nyckelordet `PROC` och avslutas med nyckelordet `ENDPROC` [30]. Ett RAPID-program börjar i en `Main procedure` [30]. Ett exempel på ett Hello world-program skrivet med RAPID som skriver ut text på FlexPendant visas nedan:

```
MODULE Module1
    PROC main()
        TPWrite "Hello world";
    ENDPROC
ENDMODULE
```

### **signaldo**

Datatyp i RAPID som används för en digital utsignal [33].

### **dionum**

Datatyp i RAPID som används för att representera en digital signal [33].

### **SetDO**

En RAPID-instruktion som ställer in en digital utsignal på styrenheten. `SetDO` har två obligatoriska argument, `Signal` av datatypen `signaldo` respektive `Value` av datatypen `dionum` [33]. Exempel på användning av instruktionen `SetDO` är

```
SetDo doCycleDone, 1;
```

där do indikerar att signalen är en digital utsignal.

### **robtarget**

Är en datatyp som används i rörelseinstruktioner [33] för att bland annat specificera position och vridning av robotens TCP [25].

### **speeddata**

Är en datatyp som bland annat specificerar med vilken hastighet som TCP ska röra sig under en rörelseinstruktion. Det finns fördefinierade konstanter av typen speeddata, till exempel v1000, som innebär att TCP rör sig med en hastighet på 1000 mm/s vid exekvering av en rörelseinstruktion [33].

### **zonedata**

Datatyp som anger med vilken precision som roboten måste nå en specificerad position i en rörelseinstruktion. Det finns fördefinierade konstanter av typen zonedata, till exempel fine, som innebär att roboten måste nå den specificerade positionen och stå stilla innan exekveringen fortlöper [33].

### **tooldata**

Datatyp som beskriver verktygets koordinatsystem, se TCP - Tool Center Point, samt verktygets lastvikt [30].

### **MoveJ**

Rörelseinstruktion som förflyttar robotens TCP med en rörelse som inte är garanterat linjär, samtidigt som alla axlar på roboten når deras slutpositioner samtidigt. Har fyra obligatoriska argument: ToPoint av datatypen robtarget, Speed av datatypen speeddata, Zone av datatypen zonedata och Tool av datatypen tooldata [33].

### **MoveL**

Rörelseinstruktion som förflyttar robotens TCP med en linjär rörelse. Har fyra obligatoriska argument: `ToPoint` av datatypen `robtarget`, `Speed` av datatypen `speeddata`, `Zone` av datatypen `zonedata` och `Tool` av datatypen `tooldata` [33].

### TPWrite

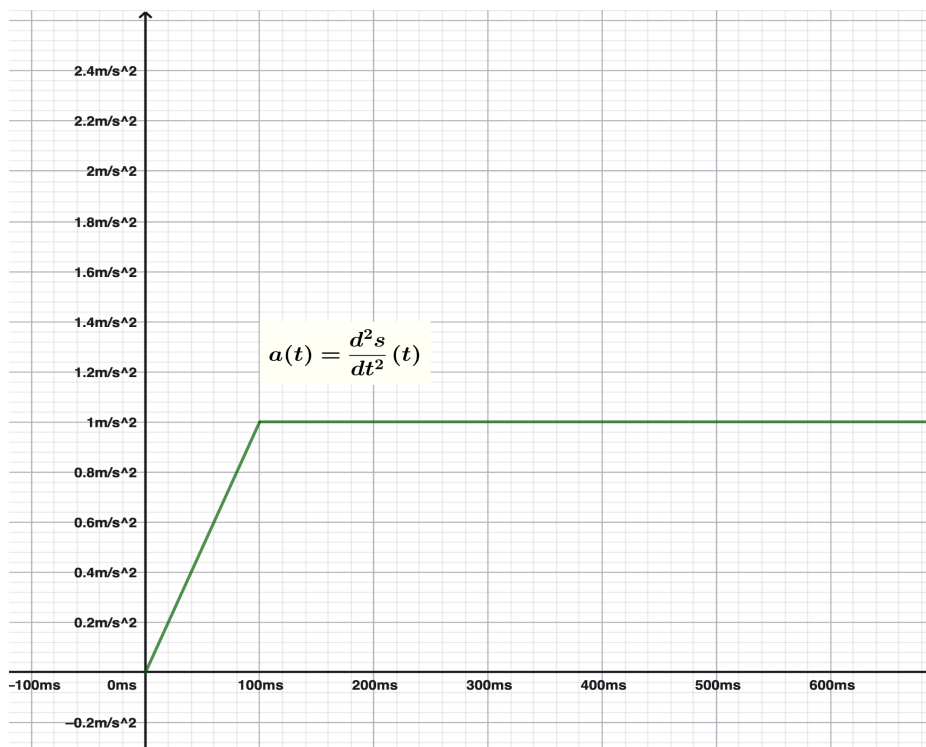
Instruktion som används för att skriva ut text på FlexPendant. `TPWrite` har ett obligatoriskt argument: `String` av datatypen `string` [33].

### WaitTime

RAPID-instruktion som används för att pausa exekvering under en specificerad tid. `WaitTime` har ett obligatoriskt argument `Time` av datatypen `num` som anger tiden, i sekunder, som programmets exekvering ska pausas [33].

### Accelerationsramp

Tiden under vilken accelerationen ökar tills den önskade accelerationen har nåtts. Studenterna tolkar accelerationsrampen, så som den används i RAPID, som en tidsperiod inom vilken derivatan av accelerationen är en konstant  $C$ ,  $C \neq 0$  [33]. Ett exempel på en accelerationsramp visas i grafen i figur 5.



**Figur 5.** Graf med accelerationsramp då  $0\text{ ms} \leq t \leq 100\text{ ms}$ . Inom detta intervall är derivatan av accelerationen konstant, vilket skapar en ”ramp” i grafen.

### **AccSet**

RAPID-instruktion som används för att ställa in robotens acceleration och accelerationsramp. `AccSet` har två obligatoriska argument, `Acc` och `Ramp`, båda av datatypen `num`. Dessa argument anger värde, i procent av det normala värdet, för acceleration respektive accelerationsramp [33].

### **clock**

Är en datatyp som kan användas för att mäta tid, likt ett tidtagarur [33].

### **ClkStart**

Instruktion som startar en klocka. Har argumentet `clock` av datatypen `clock` [33].

### **ClkStop**

Instruktion som stoppar en klocka. Har argumentet `clock` av datatypen `clock` [33].

### **ClkReset**

Instruktion som återställer en klocka. Har argumentet `clock` av datatypen `clock` [33].

### **ClkRead**

Funktion som returnerar en klockas uppmätta tid i sekunder i form av datatypen `num`. Har ett obligatoriskt argument `clock` av datatypen `clock` [33].

### **NumToStr**

Funktion som har två obligatoriska argument: `Val` och `Dec`, båda av datatypen `num`. Returnerar talet `Val` konverterat till datatypen `string` med en precision på `Dec` stycken decimaler [33].

### **3. Metod**

Detta kapitel beskriver hur examensarbetet har utförts, vilka delar arbetat omfattat och vad som gjorts i varje del. Arbetsmetoderna motiveras och studenterna förklarar hur de genomfört informationsinhämtning och behandlat mätdata. Kapitlet består av sju delar som beskriver följande: hur det initiala valet av examensarbete gick till, hur det slutgiltiga valet av arbete gjordes, hur det slutgiltiga arbetet sedan genomfördes, hur informationssökningen gick till manuellt respektive med generativ AI, metoddiskussion och avslutningsvis källkritik.

#### **3.1 Initialt val av arbete**

Följande delkapitel beskriver hur studenterna bestämde en initial inriktning för examensarbetet.

Arbetet började med att studenterna behövde bestämma vad examensarbetet skulle bestå av, då studenterna inte hittade något företag som kunde erbjuda dem ett examensarbete. Därför diskuterades ett arbete inom universitetets verksamhet med den som senare blev studenternas handledare.

Handledaren gav bland annat två förslag på arbeten som hade löst problem i undervisningen på LTH:

1. Att skapa ett program som kan simulera en laboration med två vattentankar som används inom kursen EIEF30 Styr- och reglerteknik.
2. En lösning för ett gränssnitt mellan dator och PLC som krävs för programmering till en annan laboration, även denna i kursen EIEF30. Laborationen kräver datorer med Windows XP då det saknas nyare drivrutiner för programmet som laborationen använder.

Studenterna var osäkra på om alternativ 2 var genomförbart inom examensarbetets tidsram. Ett sådant arbete skulle sannolikt kräva baklängeskonstruktion (reverse engineering) för att nå framgång. Studenterna är inte säkra på att detta skulle vara förenligt med det immaterialrättsliga skydd programmet i fråga har [34].

Studenterna uteslöt även alternativ 1 då ett sådant projekt skulle haft ett väldigt stort fokus på programmering. Studenterna var dessutom osäkra på vilket programmeringsspråk som hade

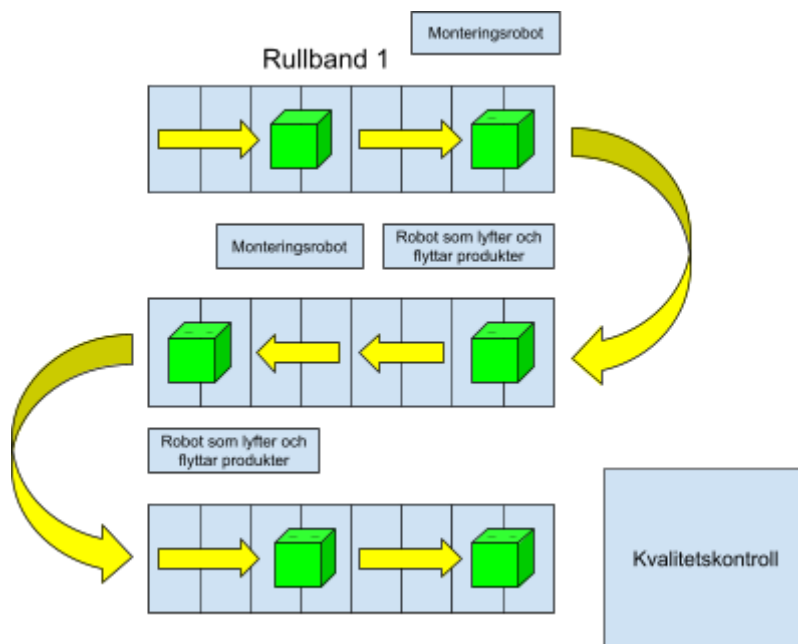
varit lättast att använda för att uppnå arbetets mål givet studenternas förkunskaper. En ändring av programmeringsspråk längre fram i processen skulle sannolikt bli kostsam tidsmässigt.

Handledaren föreslog även att studenterna kunde arbeta med programmet RobotStudio för att formulera och lösa ett problem med hjälp av industriella robotar. Studenterna testade att använda programmet lite för att få en uppfattning om programmets funktionalitet och hur svårt det skulle vara att använda för examensarbetet. Studenterna gick vidare med detta alternativ då detta var en mer flexibel lösning, där examensarbetets exakta syfte och mål- och problemformulering kunde utarbetas efterhand som studenterna såg vad som var möjligt inom den givna tidsramen.

### **3.2 Inledning och val av slutgiltigt arbete**

Detta delkapitel beskriver hur arbetet gick till i den första fasen efter att studenterna bestämt att utföra ett arbete med RobotStudio. Det beskrivs även hur valet av slutgiltigt arbete gick till.

Studenternas första idé var att utforma en industriell produktionslinje som illustreras i figur 6 och därefter undersöka olika prioriteringsalgoritmer vid ett ojämnt inflöde av inkommande produkter på rullband 1 som rullar med hög hastighet.



**Figur 6.** Bild av produktionslinjen som ursprungligen var målet att implementera i RobotStudio. Varje monteringsrobot skruvar fast en skruv på produkten.

Studenterna använde RobotStudio via licens från skolan med hjälp av webbportalen AppsAnywhere, som kan användas av universitetets studenter för att få tillgång till programvara [35]. Versionen som installerades via AppsAnywhere var RobotStudio 2019.2 (64-bit) version 7.0.8504.0151. Därefter kunde studenterna installera RobotStudio 2025.5 (64-bit) version 25.5.11605.0 när programmet hittade en uppdatering av programvaran. Licensen visade sig fungera även för den nya versionen av programmet.

För att börja lära sig använda RobotStudio tittade sedan studenterna på videogenomgångar som ABB placerat i kategorin "Get started in 30 minutes" [36]. Då dessa videogenomgångar använder en version av programmet som är mer lik RobotStudio 2025.5 än 2019.2 valde studenterna att använda den då senaste versionen, 2025.5, för projektet. Studenterna tog även del av några videogenomgångar som publicerats på YouTube av allmänheten om att förflytta objekt med rullband [37], [38], [39] samt en video om olika typer av givare [40].

Studenternas första mål var att placera ut de tre rullband och robotar som visas i figur 6. För att kunna plocka upp produkter så användes gripklon ABB Smart Gripper. En gripklo

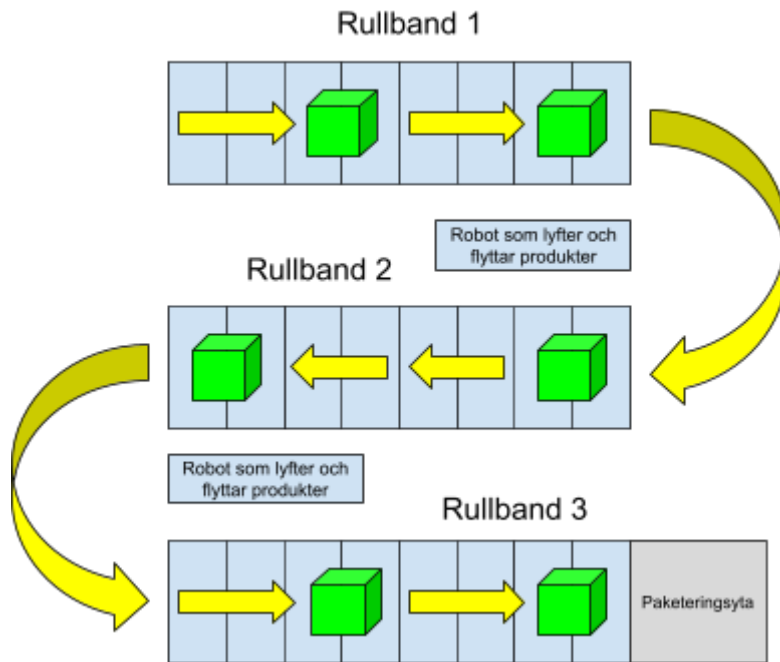
fästes sedan på fästningspunkten för verktyget på respektive robot som förflyttar produkter mellan rullbanden. Detta verktyg valdes på grund av att det var det enda inbyggda verktyget för att plocka upp och lyfta föremål som studenterna kunde se i det inbyggda biblioteket.

Därefter infogades en modell av ett rätblock från fliken Modeling → Solid → Box som var tänkt att tillfälligt representera en produkt. Denna modell ersattes sedan av en ny modell som modellerats med Autodesk Fusion 360 och som hade två cylinderformade insättningar på ovansidan av rätblocket. Tanken var att monteringsrobotorna skulle montera vars en cylinder i insättningarna på rätblocket.

Efter ett tag insåg studenterna att monteringssteget skulle bli svårare än förväntat att implementera och gick då tillbaka till en enklare modell av ett rätblock utan insättningar. Beslutet motiverades även av insikten att det skulle bli svårt att avgöra ifall produkter som nådde kvalitetskontroll hade blivit korrekt monterade av monteringsrobotarna.

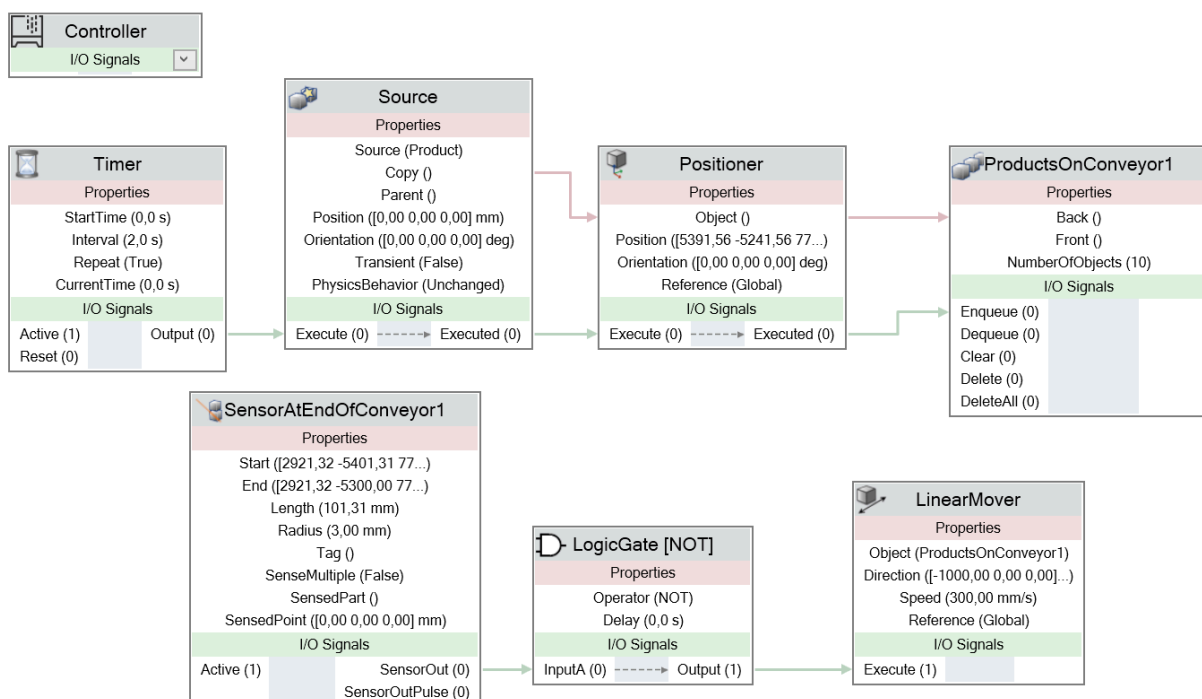
Någon form av kontrollmetod skulle då behöva utvecklas, vilket väntades ta tid från arbetets ursprungliga huvudsyfte, nämligen effektiv prioritering av robotarnas arbetsuppgifter. Beslutet innebar att monteringsrobotarna togs bort från produktionslinjen.

Dessutom insåg studenterna att de skulle behöva ändra mening med examensarbetet, då det inte skulle finnas tid att implementera och testa olika schemalägningsalgoritmer, på grund av att inlärningsprocessen för RobotStudio var betydligt längre än väntad. Istället bestämde studenterna att fokusera på att få en produktionslinje likt den i figur 7 att fungera. Samtidigt funderade de på ett nytt syfte och vad den färdiga produktionslinjen skulle kunna användas till för att motivera examensarbetet.



**Figur 7.** Önskad utformning av produktionslinjen efter några veckors arbete med projektet.

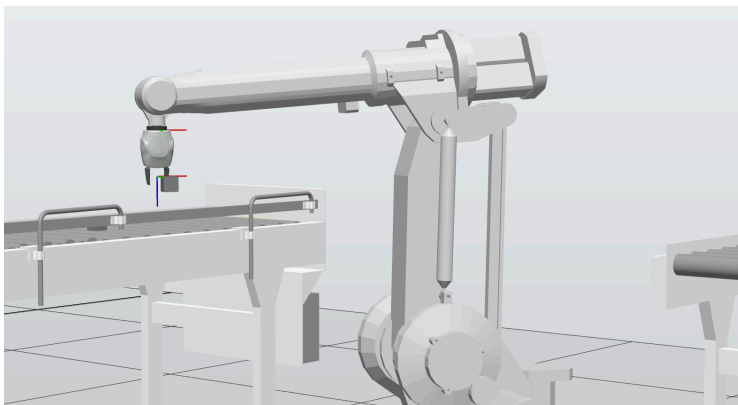
Studenterna började arbeta med olika Smart Components för att hitta en fungerande lösning som kunde simulera rullbandens funktion. Till en början användes bland annat de smarta komponenterna Source, Positioner, Queue och LinearMover för att driva produkterna på rullband 1 och stoppa rullbandet när en produkt närmade sig slutet av bandet, se figur 8.



**Figur 8.** Station Logic för den första fungerande versionen av rullband 1.

Studenterna hade inte vid denna tidpunkt insett hur Smart Components-logiken som styrde rullband 1 skulle kombineras och synkroniseras med styrning av roboten. Hur detta kunde göras klarnade efter att studenterna tittat på två videor som demonstrerar Station Logic [41], [42].

Upptäckten av Station Logic ledde till framsteg och nu kunde roboten börja röra sig baserat på signaler från en givare på slutet av rullband 1 som indikerade att det fanns en produkt att förflytta till rullband 2. De smarta komponenterna Attacher och Detacher började då användas men produkterna på rullband 1 fästes inte på ett sätt som såg rätt ut i simuleringen, se figur 9.



**Figur 9.** Robot och produkt som inte har fästs till varandra ordentligt vid simulering.

Studenterna finjusterade positionen på givaren på slutet av rullband 1 och försökte även justera robotens position och avståndet mellan griplorna på Smart Gripper för att roboten och produkten skulle passa ihop så bra som möjligt och ha ordentlig kontakt. Trots detta bestod det felaktiga beteendet vid simulering. Ett försök med en ny typ av lösning för att förflytta produkterna på rullbandet, en så kallad rullbandsmekanism med namnet ConveyorBelt, testades men gav liknande resultat.

Vid denna tidpunkt började studenterna även lära sig RAPID och använde programmeringsspråket för att styra utsignaler på styrenheten. Ett annat problem uppstod när produkterna nått rullband 2 och skulle lossas från robotens griplor. En givare på början av rullband 2 indikerade för Detacher vilket objekt som skulle lossas men produkterna lossades inte från griplorna som förväntat. Utformningen av Station Logic vid denna tidpunkt, som var mer komplicerad än den slutgiltiga lösningen, visas i figur A i Appendix A.

Studenterna löste problemet med Detacher genom att montera en givare på Smart Gripper som alltid kunde indikera för Detacher vilket objekt som skulle lossas, även innan produkten nått rullband 2. De märkte också att de kunde behålla givaren på början av band 2, som det var ursprungligen, med en mindre justering så att den inte gav falska utslag orsakade av rullbandet självt.

Rullband 1 och förflyttningen av produkter från rullband 1 till rullband 2 fungerade alltså nu, förutom att produkterna inte fästes riktigt ordentligt med Smart Gripper, vilket nämndes tidigare, se figur A i Appendix A. Rullband 1 och rullband 2 styrdes inte heller oberoende av varandra. Detta innebar att om rullband 1 stannade så stannade också rullband 2, även om produkterna kunnat rulla vidare på rullband 2 utan problem.

Då en stor del av tiden för projektet redan passerat och syftet och mål- och problemformuleringar fortfarande inte var helt bestämda och tydliga så valde studenterna att ta en ny diskussion med handledaren. Detta resulterade i att studenterna bestämde sig för att fokusera på en enklare produktionslinje med två rullband, vilket också blev den slutgiltiga idén. Ett nytt syfte togs även fram, nämligen energiförbrukning, som tydligare kunde kopplas till riktiga förekommande problem inom industriell produktion [13], [15].

Studenterna kunde fortfarande valt att gå vidare med att utveckla produktionslinjen i figur 7, vilket kommer att diskuteras vidare i kapitel 6. Produktionslinjen som studenterna slutligen valde att undersöka är slutprocessen i figur 1. Fokus blev sedan att mäta robotens energiförbrukning som uppstod när den utförde sitt arbete, dvs att förflytta produkter från rullband 1 till rullband 2. Robotcellen studenterna hade konstruerat vid denna tidpunkt var en produktionslinje enligt figur 1, förutom att produkterna inte fästes till Smart Gripper ordentligt, se figur 9.

Roboten som använts fram till denna tidpunkt var IRB 1410. Denna typ av robot valdes då den hade tillräcklig räckvidd för att kunna nå båda rullbanden. Studenterna insåg dock att IRB 1410 hade en viktgräns på 5 kg [43], vilket var problematiskt då ett av projektets nya mål var att roboten skulle kunna utföra tunga lyft av produkterna för att minska den fysiska belastningen för personalen under arbetspassen. Viktgränsen på 5 kg innefattar vikten för verktyget som fästs på fästningspunkten för verktyg på roboten och lasten som eventuellt kopplas till verktyget [23], [43].

Studenterna blev därför tvungna att välja en annan robot. Valet föll på IRB 4600-60/2.05 då denna hade tillräcklig räckvidd och en viktgräns på 60 kg [44], vilket mötte kraven. Det finns en funktion i RobotStudio som kan användas för att byta ut en robot mot en annan [25] men studenterna fick felmeddelandet “Failed to replace robot” när de försökte använda funktionen för att byta från roboten IRB 1410 till IRB 4600-60/2.05. Därför valde studenterna att börja om med ett nytt RobotStudio-projekt.

Av en slump hade studenterna tidigare sett att det finns ett alternativ med olika demo-stationer i RobotStudio. Ett av dessa projekt var SC Infeeder finished. I detta projekt använde roboten ett verktyg som använde flera vakuumsugkoppar för att fästa objekt till roboten. Detta löste ett problem som tidigare identifierats, nämligen att Smart Gripper bara kan ha ett avstånd på upp till 5 cm mellan griplorna [27], vilket begränsar möjligheten att lyfta tyngre och större föremål. Verktøget med vakuumsugkoppar kunde täcka en area på 40 cm × 20 cm med sugkoppar i hörnen, vilket gav större realistiska möjligheter att representera ett tyngre föremål.

Den tidigare produktionslinjen återskapades med den nya roboten IRB 4600-60/2.05 och det nya verktyget med vakuumsugkoppar, som kallas Vacuum Tool. Efter detta kördes simuleringar där produkterna nu blev fästa ordentligt till robotens verktyg. Det fanns nu alltså en fungerande produktionslinje och simuleringar kunde nu köras för att undersöka robotens energiförbrukning och olika cykeltider.

### **3.3 Genomförande av slutgiltigt valt arbete**

Detta delkapitel förklarar hur arbetet genomfördes efter att en slutgiltig inriktning, syfte och mål- och problemformuleringar var bestämda. Delkapitlet beskriver modelleringen i RobotStudio, I/O-konfiguration och Station Logic, RAPID-programmering, mätningmetod, hur mätdata behandlats för analys och slutligen hur studenterna gjort för att kunna arbeta tillsammans med arbetet.

#### **3.3.1 Modellering av produktionslinje i RobotStudio**

Följande delkapitel förklarar hur robotcellen modellerades i RobotStudio. För samtliga steg användes förvalda värden för de parametrar och alternativ vars konfiguration inte uttryckligen specificeras i detta delkapitel.

Ett nytt tomt RobotStudio-projekt kallat Project skapades utan att inkludera en robot och virtuell styrenhet. Roboten IRB 4600-60/2.05 lades till och placerades på positionen (186, 575, 0, 0) i World Coordinate System. Sedan skapades en virtuell IRC5-styrenhet för roboten med alternativet From Layout och RobotWare-versionen 6.16.03.00.

Ett nytt RobotStudio-projekt skapades tillfälligt genom att Pack & Go-filen för demo-stationen SC Infeeder finished packades upp. Vacuum Tool i demo-stationen sparades sedan som ett bibliotek som kunde importeras till Project. Vacuum Tool fästes sedan till IRB 4600-60/2.05 i projektet Project.

Därefter lades två rullband med namnet Conveyor till i robotcellen från användarbiblioteket. Båda rullbanden hade bredden 950 mm och längden 4 m. Rullband 1 placerades i punkten (- 3 373, 099, 555, 676, 291, 864) i WCS. Motsvarande punkt för rullband 2 blev (- 194, 329, - 1 429, 427, 291, 864), också i World Coordinate System.

En produkt, representerad av ett är rätblock med dimensionerna 25 cm × 40 cm × 10 cm skapades, med hörnpunkten (- 3 385, 10, 808, 731, 726, 86) i World Coordinate System. Modellen fick namnet Product.

Därefter placerades modeller av upphöjningar under rullbanden. Dessa modeller skapades genom Modeling → Solid → Box. Dimensionerna på dessa upphöjningarna var 3 430 mm × 1 169, 97 mm × 291, 86 mm. Upphöjningen till rullband 1 placerades med hörnpunkt i punkten (- 2 928, 928, 445, 861, 0) i WCS. Motsvarande hörnpunkt för upphöjningen till rullband 2 blev (250, 958, - 1 539, 459, 0).

För att kunna skapa en rullbandsmekanism skapades en tom Smart Component. Sedan drogs rullband 1 i Layout-vyn till den nyskapade smarta komponenten i samma vy, vilket gjorde att rullband 1 blev en del av denna Smart Component. Sedan skapades en rullbandsmekanism med namnet ConveyorBelt1 via Modeling → Mechanism → Create Conveyor. Den smarta komponenten som innehöll rullbandsmodellen valdes som geometri för rullbandsmekanismen. Längden på rullbandsmekanismen bestämdes till 5,4 m. Anledningen till att längden på rullbandsmekanismen är längre än rullbandsmodellen på 4 m, är att objekt som skapats av rullbandsmekanismen tycktes raderas automatiskt då dessa nådde slutet av

rullbandsmekanismen. Detta hade dessutom skapat problem när studenterna försökte fästa produkter till robotens verktyg tidigare. Produkterna behövde också kunna transporteras vidare på rullband 2, vilket också motiverade den valda längden på rullbandsmekanismen.

Efter att rullbandsmekanismen skapats så kunde produkten som kallades Product i Layout-vyn dras till mappen Object source, som fanns under rullbandsmekanismen i samma vy. Sedan sattes hastigheten på rullbandet till 500 mm/s. Allt detta tillsammans innebär nu att produkter kunde skapas och sedan förflyttas längs rullbandet.

För att rullband 1 ska stanna när en produkt når slutet av bandet så skapades en LineSensor i Station Logic. Startpunkten för givaren var (318,43, 880, 730) och slutpunkten (318,43, 1180, 730). Radien sattes till 10 mm. Den smarta komponenten fick namnet SensorEndOfConveyor1. I Station Logic gjordes en anslutning från LineSensors digitala utgång SensorOut till rullbandsmekanismens digitala ingång ConveyorStop.

Efter detta placerades en till givare, denna gången på början av rullband 2. Detta var också en LineSensor, med startpunkt i (- 23, 54, - 1 088, 64, 756, 86) och slutpunkt i (- 23, 54, - 788, 64, 756, 86). Radien för denna givare ställdes också in till 10 mm.

Sedan användes en kombination av Mechanism Joint Jog och Mechanism Linear Jog för att ställa in roboten i den position den skulle ha när den skulle fästa en produkt på slutet av rullband 1 till verktyget med vakuumsugkopparna. Denna position kallas AtConveyor1. Mechanism Joint Jog användes först och sedan finjusterades positionen med Mechanism Linear Jog. Hur detta gjordes visas i tabell 2. Sedan användes alternativet Home → Teach Target för att skapa ett nytt Target utifrån robotens nuvarande ställning.

**Tabell 2.** Värden på parametrar för att ställa roboten i positionen AtConveyor1. Observera att värdena för Mechanism Joint Jog ändras när Mechanism Linear Jog används.

| Mechanism Joint Jog  |       |       |       |       |       |
|----------------------|-------|-------|-------|-------|-------|
| Led 1                | Led 2 | Led 3 | Led 4 | Led 5 | Led 6 |
| 90°                  | 0°    | 0°    | 0°    | 90°   | 90°   |
| Mechanism Linear Jog |       |       |       |       |       |

| X   | Y     | Z   | RX   | RY | RZ    |
|-----|-------|-----|------|----|-------|
| 186 | 1 026 | 827 | 180° | 0° | -180° |

Efter att robotens ställning ställts in till AtConveyor1 så användes Mechanism Linear Jog för att ställa in Z till 832. Sedan skapades ett nytt mål, JustAboveConveyor1, med Teach Target. Detta är robotens viloposition när den inte har någon produkt att förflytta men väntar på att en sådan ska nå fram till slutet av rullband 1. Ytterligare en ställning kallades HighAboveConveyor1 och används för att roboten och produkten inte ska slå i någon annan produkt på rullband 2 under rörelsen från rullband 1 till rullband 2. Efter att roboten ställts in till målet JustAboveConveyor1 med funktionen Jump To Target så användes Mechanism Linear Jog och Z ställdes in som 1020. Sedan användes återigen funktionen Teach Target för att spara ställningen som ett Target.

Efter detta ställdes roboten i sin hemposition, genom att högerklicka på roboten i Layout-vyn och sedan välja Move To Pose → Jump Home. Sedan skapades målet AtConveyor2 på liknande sätt som AtConveyor1, genom att första använda Mechanism Joint Jog, sedan Mechanism Linear Jog enligt tabell 3 och slutligen Teach Target. AtConveyor2 beskriver ställningen när roboten är framme och har stannat vid rullband 2 och är redo att lossa produkten från verktyget.

**Tabell 3.** Värderna på parametrar för att ställa roboten i positionen AtConveyor2.

| Mechanism Joint Jog  |       |       |       |       |       |
|----------------------|-------|-------|-------|-------|-------|
| Led 1                | Led 2 | Led 3 | Led 4 | Led 5 | Led 6 |
| -90°                 | 0°    | 0°    | 0°    | 90°   | 90°   |
| Mechanism Linear Jog |       |       |       |       |       |
| X                    | Y     | Z     | RX    | RY    | RZ    |
| 0                    | -946  | 860   | 180°  | 0°    | 180°  |

Slutligen skapades ett sista Target. Med Mechanism Linear Jog ställdes Z till 1020. Efter detta användes Teach Target för att spara målet, som kallades för AboveConveyor2.

Efter att samtliga Targets sparats så skapades en Path som bestod av samtliga Targets. Ett tomt RAPID-program skapades sedan genom RAPID-fliken och sedan RAPID → T\_ROB1 → Module1 i vyn över styrenheten. Sedan användes alternativet Synchronize to RAPID med alternativet att synkronisera mappen Paths & Targets markerat. Detta resulterade i att en procedure skapades i RAPID med rörelseinstruktioner till samtliga Targets som skapats.

### 3.3.2 I/O-konfiguration och Station Logic

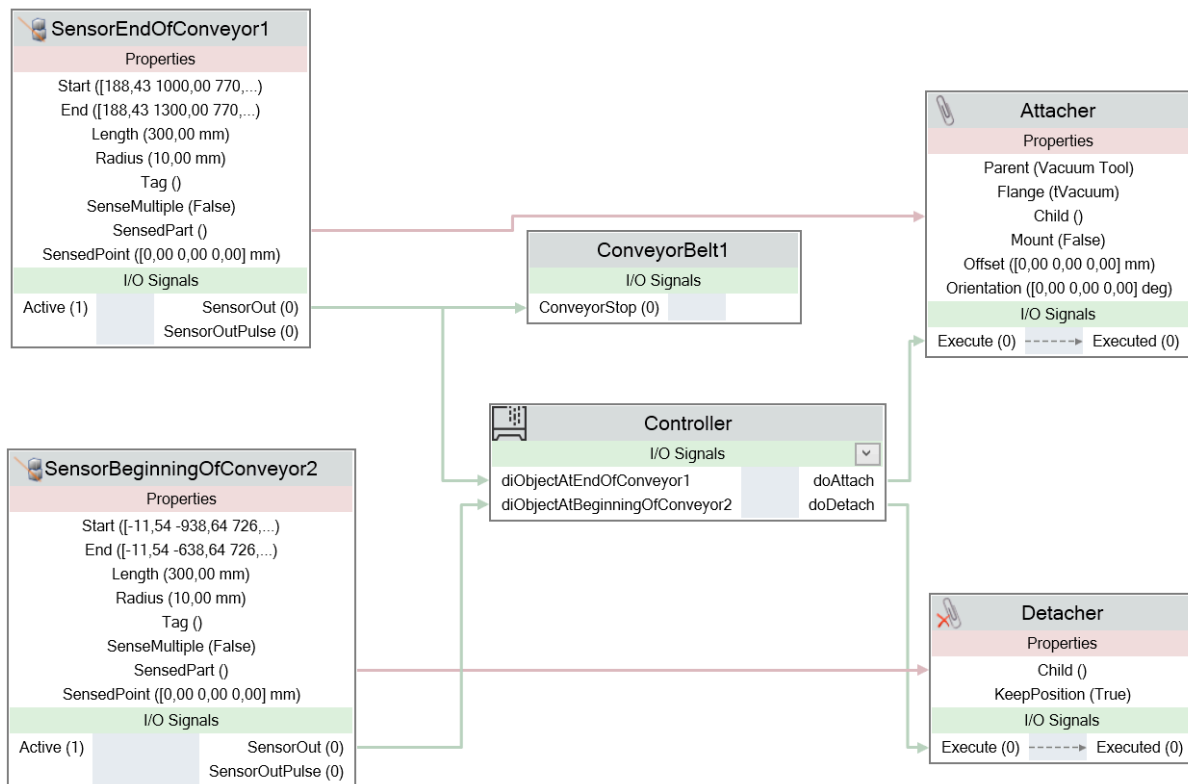
Denna del förklarar hur insignaler och utsignaler på styrenheten konfigurerades och kopplades samman med omgivande smarta komponenter och RobotStudio-objekt genom Station Logic.

Station Logic utformades på så vis att styrenheten konfigurerades med fyra nya signaler. Av dessa är två stycken digitala insignaler: diObjectAtEndOfConveyor1 och diObjectAtBeginningOfConveyor2. De andra två är digitala utsignaler: doAttach och doDetach. Signalerna skapades från fliken Controller → Configuration → I/O System → Signal. Namn och typ av signal valdes. För inställningen Access Level valdes alternativet All då detta alternativ används i en videogenomgång från ABB [28].

Efter en omstart av den virtuella styrenheten kunde dess nyskapade signaler exponeras i Station Logic. Två nya Smart Components lades till i Station Logic: Attacher och Detacher. Dessa används för att, rent visuellt i simuleringen, fästa och lossa produkter från robotens verktyg med vakuumsugkoppar.

Den smarta komponenten Attacher konfigurerades så att Parent är mekanismen för verktyget med vakuumsugkoppar, det vill säga Vacuum Tool.

Efter detta så utformade studenterna hur de smarta komponenterna, rullbandsmekanismen och styrenheten skulle interagera med varandra i Station Logic. Hur Station Logic såg ut efter denna process visas i figur 10.



**Figur 10.** Den slutgiltiga utformningen av projektets Station Logic.

Utsignalen **SensorOut** på givaren **SensorEndOfConveyor1** kopplas med en anslutning till insignalen **ConveyorStop** på rullbandsmekanismen. Detta gör att rullband 1 rullar om och endast om det inte finns ett objekt på slutet av rullbandet som kan detekteras av givaren.

**SensorOut** från **SensorEndOfConveyor1** går även till styrenheten, för att roboten ska veta när det finns en produkt som är redo att bli upplockad från bandet. Givarens egenskap **SensedPart** kopplas också till egenskapen **Child** på **Attacher**, för att **Attacher** ska veta vilket objekt som ska fästas till robotens verktyg.

Egenskapen **SensedPart** för givaren **SensorBeginningOfConveyor2** kopplas till egenskapen **Child** på den smarta komponenten **Detacher**. Detta görs för att **Detacher** behöver veta vilket föremål som ska lossas från robotens verktyg. **SensorOut** på **SensorBeginningOfConveyor2** ansluts till **diObjectAtBeginningOfConveyor2** på styrenheten. Detta behövs för att styrenheten ska veta att **Detacher** får information om vad som ska lossas. Detta görs för att undvika att **Execute** på **Detacher** sätts till hög utan att det finns någon definierad **Child**-egenskap.

Avslutningsvis är `doAttach` och `doDetach` på styrenheten kopplad till `Execute` på `Attacher` respektive `Detacher`. Detta gör att det går att styra när objekt fästs och lossas från robotens verktyg via RAPID-kod.

### 3.3.3 RAPID

Syftet med detta delkapitel är att närmare förklara och motivera hur RAPID har använts för att ge upphov till mätdata som möjliggör en kvantitativ undersökning.

Det fullständiga RAPID-programmet som examensarbetet använder finns i figur B, appendix B. Programmet har skrivits med hjälp av officiell dokumentation om programmeringsspråket RAPID.

`cycleTimer` används för att mäta cykeltiden. I projektet har studenterna valt att definiera cykeltiden som tiden det tar att flytta en produkt från rullband 1 till rullband 2, från och med att det finns en produkt på slutet av rullband 1, även inkluderat tiden det tar för roboten att ta sig från rullband 2 tillbaka till sin viloposition `JustAboveConveyor1`. En ny cykel kan inte börja förrän den föregående cykeln har slutat.

`executionTimer` mäter den totala exekveringstiden för 5 cykler adderat med tiden det tar för den första produkten att förflyttas från början till slutet av rullband 1. Timerna styrdes sedan med instruktionerna `ClkStart`, `ClkStop`, `ClkRead` och `ClkReset`.

`AccSet` används för att ställa in värden på parametrarna `acceleration` och `accelerationsramp`. Variabeln `velocity` användes för att ändra hastigheten för rörelseinstruktioner på en global nivå i programmet.

För att kunna styra de smarta komponenterna `Attacher` och `Detacher` via styrenheten så användes `SetDO` för att ställa in styrenhetens digitala utsignaler `doAttach` och `doDetach`. `WaitTime` användes för att skapa fördröjningar mellan att en digital utsignal ändras och att nästa instruktion exekveras. Detta gjordes för att se till att en ändring av en utsignal i RAPID-koden hann propagera till kopplade smarta komponenter, då simuleringarna utgår från cykliska beräkningar [25].

Programmet använder två stycken tooldata: `tVacuum` respektive `vacuumToolWithLoad`. `tVacuum` ingick i biblioteksfilen för verktyget med vakuumsugkopparna och representerar verktyget när roboten inte lyfter någon produkt.

`vacuumToolWithLoad` är identisk med `tVacuum` förutom att värdet på parametern `tooldata.loaddata.mass` ökas med 40 kg, för att simulera lastvikten när en produkt är kopplad till verktyget. I rörelseinstruktionerna kunde studenterna sedan växla mellan `tVacuum` och `vacuumToolWithLoad` beroende på om en produkt är kopplad till verktyget eller inte för en viss rörelse.

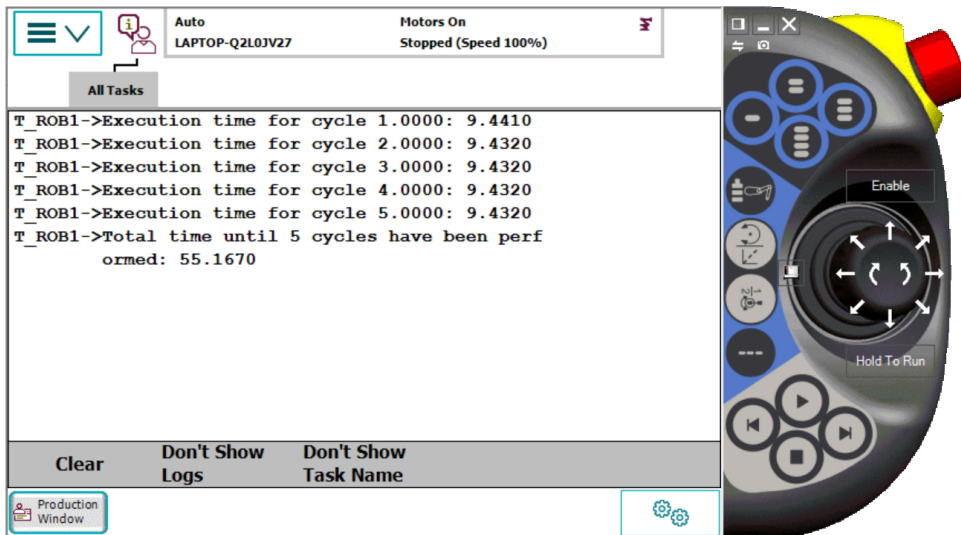
Samtliga rörelseinstruktioner är av varianten `MoveJ`. Anledningen till detta är att `MoveL` ej var möjlig att använda för alla rörelseinstruktioner, då kravet på att TCP ska utföra en linjär rörelse inte alltid kunde uppfyllas givet placeringarna av målen.

### 3.3.4 Mätningssmetod

Detta delkapitel förklarar vilken mätningssmetod arbetat har använt och vilka data som har samlats in.

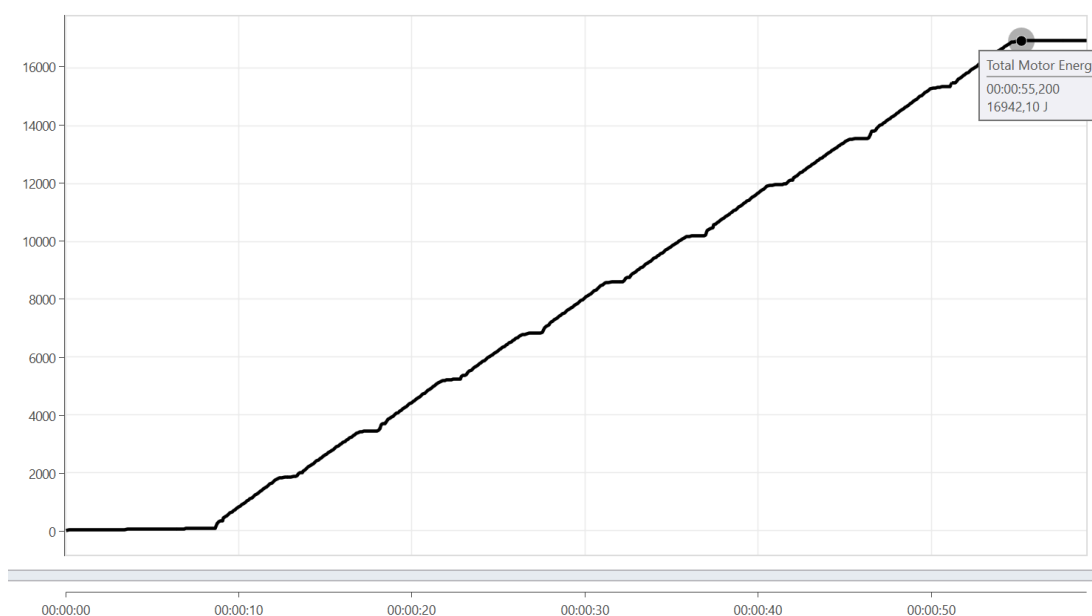
Arbetet undersöker problemformuleringarna i kap. 1.4 med hjälp av en kvantitativ metod, i form av simulering som ger upphov till mätdata. Acceleration, accelerationsramp och hastighet undersöktes som oberoende variabler. Beroende variabler var cykeltid för cykel 1-5, exekveringstid, avläst tid och den totala mängd energi som robotens motorer förbrukar. Under fliken `Simulation` → `Signal Setup` kunde signalen `Total Motor Power` bockas i så att signalen inkluderas i mätningarna som spelas in med `Signal Analyzer`.

Cykeltid och den totala exekveringstiden skrevs ut på `FlexPendant` med `RAPID`-instruktionen `TPWrite`. Studenterna kunde sedan läsa av dessa värden efter simuleringen. Ett exempel på hur mätresultat på `FlexPendant` kan se ut visas i figur 11.



**Figur 11.** Exempel på mätdata med cykeltid för cykel 1-5 och total exekveringstid som visas på FlexPendant. Observera att den totala exekveringstiden inkluderar tiden som roboten väntar tills den första produkten kommer fram till slutet av rullband 1.

Med Signal Analyzer läste studenterna av avläst exekveringstid och motorernas totala energiförbrukning, vilket exemplifieras i figur 12.



**Figur 12.** Exempel på användning av Signal Analyzer för att se avläst exekveringstid och total energiförbrukning från robotens motorer. Grafen är motorernas effektförbrukning

integrerad [25] över tiden, alltså  $E(t) = \int_0^t P(\tau) d\tau$ . Avläst exekveringstid inkluderar tiden

som roboten väntar tills den första produkten kommer fram till slutet av rullband 1.

Totalt undersöktes tre hastigheter på TCP: 1 m/s, 1,5 m/s och 2 m/s. För acceleration och accelerationsramp testades vardera sex inställningar: 100 %, 90 %, 80 %, 70 %, 60 % och 50 %. Mätningar genomfördes för alla möjliga kombinationer av dessa värden på de oberoende variablerna. Mätdata skrevs in i tabell C1, C2 och C3 i appendix C1, C2 respektive C3. För varje unik kombination av hastighet, acceleration och accelerationsramp genomfördes två mätningar för att kunna studera hur olika mätningar med samma parametrar avviker från varandra. Valet att göra just två mätningar och inte fler motiveras i kapitel 4.

Studenterna valde att mäta total tid och motorens totala energiförbrukning för att förflytta fem produkter. Då blev skillnaden mellan olika val av hastighet, acceleration och accelerationsramp tydligare. Varje gång en mätning hade gjorts startades RobotStudio om utan att spara några eventuella ändringar. Detta gjordes för att se till att mätningarna alltid startade med det önskade starttillståndet så att grundförutsättningarna var likadana mellan mätningarna. Tidigare hade studenterna märkt att signaler från givare kunde sparas mellan körningar och att instanser av produkter kopplade till rullbandsmekanismen kunde råka sparas, vilket ledde till oönskat beteende när programmet exekverades på nytt.

### 3.3.5 Behandling av mätdata för analys

Denna del förklarar hur mätdata i tabell C1, C2, C3 i Appendix C1, C2 respektive C3 har behandlats för att visualisera skillnader och identifiera mönster.

De uppmätta parametrarna inkluderar energiförbrukning, avlastningstid, total tid och cykeltid. Dessa komponenter lagras som uppmätta data i excel och används för jämförelse. Genom att först jämföra total tid (körtid) och energiförbrukning, drar man slutsatsen att när hastighets-, accelerations- och rampvärdena är små, leder längre arbetstid till lägre energiförbrukning.

En andra jämförelse visar att när hastighets-, accelerations- och rampvärdena är stora, leder kortare arbetstid till högre energiförbrukning. Därför kan man dra slutsatsen att arbetstid och energiförbrukning har ett linjärt samband. I nästa steg sätts hastighet som en variabel och acceleration och ramp som konstanter, till exempel genom att jämföra energiförbrukning och arbetstid vid hastigheterna  $v_{1000}$  och  $v_{1500}$ , båda med accelerations- och rampvärden på 50. Resultaten visar att skillnaden i energiförbrukning vanligtvis är signifikant, vanligtvis runt 30 %.

Nästa steg är att sätta både hastighet och acceleration/ramp som konstanter, med acceleration satt som variabel X och energi/ramp som variabel y. Detta används för jämförelse, och resultaten visar ett linjärt samband: ju större acceleration, desto större energiförbrukning och desto kortare drifttid. Skillnaden i energiförbrukning är dock inte signifikant, ungefär 2 % till 5 %, vilket indikerar en begränsad påverkan av acceleration på energi.

Nästa steg är att återigen använda hastighet som konstant. Den här gången är accelerationen också konstant, men rampen blir en variabel. Effekten av ramp förändringar observeras på energiförbrukningen när hastigheten är 1000, 1500 och 2000, och accelerationen är 50/60/70/80/90/100. Till exempel, jämför man en hastighet v1000 med en acceleration på 100 och en ramp på 100 kontra en ramp på 90. Resultaten visar att rampen har en ännu mindre påverkan på energiförbrukningen, ungefär endast 1 %, och det linjära sambandet kvarstår under de flesta förhållanden.

Samma metod användas för att jämföra cykeltid och energi/ramp, och resultaten visar samma linjära samband som tidigare. En längre cykeltid resulterar i lägre energiförbrukning och vice versa.

Det är viktigt att förklara här att total tid är den tid som förbrukas av RAPID-enheten med en enda timer, medan avläst tid är den tid som förbrukas av signalanalysatorn. Skillnaden ligger i de olika metoder som används för att beräkna tiden, vilket resulterar i variationer mellan total tid och avläst tid. Det är dock värt att notera att total tid (körtid) vanligtvis är något mindre än avläst tid, vanligtvis med en till två decimaler. För att säkerställa konsekvens i experimentell analys används total tid (körtid) istället för avläst tid.

Det är också viktigt att förtydliga att cykeltid endast tar hänsyn till robotarmens rörelsetid, medan total tid (körtid) och avläst tid vanligtvis också tar hänsyn till avståndet produkten färdas från början av bandet till robotarmen och avståndet från det att robotarmen sänker produkten vid band 2 till slutpunkten. Därför är summan av cykeltid inte lika med totaltid eller avlästtid. Detta är viktigt att notera.

Vid dataanalys måste energiförbrukningen jämföras och analyseras separat från andra tidsperioder; annars påverkas de experimentella slutsatserna.

### 3.3.6 Kommunikation och samarbete

Denna del förklarar hur studenterna kommunicerat med varandra och vilka verktyg som har krävts för att genomföra arbetet tillsammans.

Kommunikationen mellan studenterna har skett genom fysiska möten, SMS och Discord. SMS och Discord har främst använts för att planera möten och ställa frågor till varandra utanför kontorstid. Under fysiska möten, som skett minst en gång i veckan, har studenterna uppdaterat varandra om vilka framsteg som gjorts och vilka utmaningar som uppstått. Vid större problem har studenterna arbetat tillsammans för att försöka hitta lösningar. Även planering och arbetsfördelning har genomförts under mötena.

För att kunna arbeta tillsammans med RobotStudio exporterades RobotStudio-projektet som en Pack & Go-fil som sedan kunde laddas upp till en mapp på OneDrive med gemensam åtkomst.

Studenterna kunde inte arbeta parallellt med samma projekt i realtid. Istället gjorde de en kopia av önskad version av projektet lokalt på datorn och packade upp filen. Efter ändringar gjorts så fick projektet exporteras om till Pack & Go-format och kopieras till mappen på OneDrive. Sedan fick studenterna meddela varandra om vilken version som var intressant att arbeta med framåt.

För rapportskrivning användes huvudsakligen Google Docs med konvertering till .docx-format för finjustering av formatering.

## 3.4 Manuell informationssökning

Följande delkapitel förklarar hur studenterna sökt information under examensarbetet.

Studenterna använde flera webbplatser och tjänster som rekommenderats av universitetet för att söka vetenskapliga artiklar, såsom Finn, EBSCOhost och Google Scholar. När intressanta artiklar påträffades så undersöktes dess referenser för att lättare hitta fler källor inom samma område.

Studenterna använde bland annat nyckelord som robots, RobotStudio, cobots, cobots manufacturing, energy reduction, energy efficiency, energy optimisation, energy consumption, simulation, robot trajectories, linear movement, joint movement, movement type, types of movement, manufacturing och production line för att hitta källor som handlar om industriella robotar och dess energiförbrukning inom tillverkningsindustrier.

För att hitta källor kopplade till arbetsmiljö, ergonomi och skador från tunga lyft använde studenterna bland annat följande nyckelord: heavy lifting, injury, injuries, workplace, occupational safety, safety, back pain, musculoskeletal disease, manufacturing och industry. Studenterna försökte även använda specialiserade termer som till exempel MoveJ och MoveL, vilket inte gav önskat resultat. Då kunde studenterna ta hjälp av generativ AI, vilket redogörs för i delkapitel 3.5.

När en intressant artikel hittades lästes först abstract för att bekräfta om innehållet stämde överens med vad som eftersöktes. Även om innehållet inte riktigt stämde överens med det som studenterna sökte så kunde det finnas användbara termer eller nyckelord som kunde användas för vidare informationssökning.

Studenterna kompletterade även de vetenskapliga artiklarna med teknisk dokumentation, myndighetsrapporter, standarder och information från International Federation of Robotics. Användningen av RobotStudio begränsade många av källorna om programmet till dess utgivare ABB. Detta diskuteras vidare i delkapitel 3.7. När studenterna ville mäta exekveringstid använde de till exempel söktermen “timer” i ABB:s dokumentation av instruktioner, funktioner och datatyper i RAPID [33].

När studenterna sökte artiklar om automatisering med robotar och påverkan på arbetsmarknaden användes söktermerna robots replacing work/workers och driving factors AI manufacturing.

Studenterna tog också hjälp av vetenskapliga artiklar för inspiration kring arbetets inriktning samt mål- och problemformuleringar. Informationssökningen bidrog också till val av metod och gav studenterna en uppfattning om vilka parametrar som var möjliga och relevanta att ändra med RAPID-kod. Studenterna försökte även se om det fanns kopplingar mellan deras resultat och tidigare resultat från vetenskapliga undersökningar.

### 3.5 Användning av generativ AI

Följande delkapitel beskriver hur generativ AI, i form av så kallade chattbottar, använts i detta examensarbete.

Generativ artificiell intelligens har i viss utsträckning använts i detta examensarbete som ett verktyg för att hitta relevant information, om försök att hitta sådan information genom vetenskapliga artiklar eller officiell dokumentation från ABB enligt tillvägagångssättet i delkapitel 3.4 misslyckats. Med ett misslyckat försök menas att studenterna inte sett någon annan lösning än generativ AI som tillräckligt effektiv för att lösa problemet inom rimliga tidsramar givet arbetets omfattning, alltså om manuell informationssökning inte var tillräckligt effektiv.

Med hjälp av AI kunde studenterna hitta vetenskapliga artiklar som beskrev följande begrepp: cykeltid, PLC, robotcell och industrirobot. Många andra vetenskapliga artiklar och ABB:s dokumentation nämnde dessa begrepp men förklarade inte deras innebörd. Studenterna hade en uppfattning om vad begreppen betydde baserat på tidigare erfarenheter eller sammanhang men behövde källor för att styrka dessa definitioner. Därför användes generativ AI i dessa fall.

Generativ AI har även använts för att komplettera mängden vetenskapliga artiklar som undersöker vissa frågor, till exempel skillnaden mellan linjär och olinjär förflyttning av TCP, när studenternas val av nyckelord ej lett till relevanta artiklar.

Generativ AI hjälpte även studenterna att hitta portalen för ABB:s bibliotek för olika typer av dokumentation. Sedan kunde de använda söktermer som RobotStudio, Smart Gripper och RAPID i detta bibliotek för att hitta relevant dokumentation.

Studenterna kontrollerade alltid källorna som generativ AI föreslog. Information från dessa källor, som används i rapporten, bygger helt på ursprungskällan eftersom det finns risk att generativ AI återger information bristfälligt eller direkt felaktigt i sina svar. Detta har varit en grundläggande förutsättning för att säkerställa arbetets akademiska integritet trots att generativ AI förekommit i arbetsprocessen.

## 3.6 Metoddiskussion

Meningen med denna del är att motivera och diskutera arbetets metod och vilka fördelar, nackdelar och brister som valet av metod har.

### 3.6.1 Simulering och datakvalitet

Detta delkapitel resonerar kring hur väl simulering lämpar sig som metod för att samla in mätdata som sedan utgör en grund för arbetets analys och slutsats.

Simuleringsmodeller bygger på förenklingar. Det kan påverka noggrannheten i de erhållna resultaten, vilket åtminstone en tidigare studie visat [14].

En fallstudie utförd hos Volvo på en station med 4 robotar som utför svetsning visade att optimering av fyra rörelseparametrar kopplade till hastighet och acceleration kunde minska den totala energianvändningen och toppeffekten med vardera 12 %. I fallstudien utfördes mätningar på de fyra robotarna på plats i fabriken men även genom simuleringar.

Effektförbrukningen vid simulering underskattades jämfört med den riktiga uppmätta förbrukningen men forskarna anser att precisionen ändå motiverar simulering av energiförbrukning under optimeringsprocessen [14].

I en annan undersökning mättes energiförbrukning och då överskattades den förbrukade energin i simuleringen jämfört med den riktiga roboten, samtidigt som båda graferna följde samma trend [19]. I denna undersökning mättes även exekveringstid och även denna egenskap överskattades något i simuleringen [19].

Med dessa tidigare studier i åtanke tänker studenterna att simulering kan vara ett bra sätt att undersöka hastighet, acceleration och accelerationsramp och dessa parametrars ungefärliga inverkan på robotens energiförbrukning och cykeltid.

Mätningar visade sig ta lång tid att genomföra, speciellt på grund av att RobotStudio och den virtuella styrenheten startades om mellan varje mätning. En lösning som hade automatiserat testning och registrering av mätdata hade varit användbar men studenterna kan inte se något sådant verktyg i RobotStudio [25].

### 3.6.2 Andra faktorerers inverkan

Denna del nämner några andra faktorer som kan ha inverkan på produktionslinjens faktiska energiförbrukning som fabriken behöver betala för.

Endast energiförbrukning kopplad till roboten studeras i detta arbete. Energiförbrukning från andra delar av produktionssystemet studeras inte, även om denna förbrukning indirekt kan påverkas av robotens rörelsemönster.

Som studenterna nämner i kapitel 1.6 så finns alternativa metoder att identifiera sätt att minska energiförbrukningen på, som till exempel energi- eller kostnadsfunktioner [14], [16]. Dessa hade kunnat resultera i bättre förslag på potentiella val av rörelseparametrar för optimal energiförbrukning, utan samma behov av uttömmande prövning av parameterkombinationer. Information om en robots parametrar för tröghet och friktion kan vara information som robotens tillverkare håller konfidentiell [20], vilket dock kan försvåra användningen av vissa av dessa alternativa metoder.

Avståndet från roboten till rullbanden valdes så att avståndet var ungefär 50 % av robotens räckvidd i  $x$ -led när roboten rörde sig mellan rullband 1 och 2. En studie visar att denna procentandel av robotens räckvidd ger optimal energiförbrukning för några olika ABB-robotar [19].

Valet av placering av rullbanden i förhållande till roboten i höjded valdes också så att ungefär 50 % av robotens räckvidd i detta led användes. Detta valet byggde på en misstolkning av studiens resultat. Studenterna tolkade det som att den optimala arbetspositionen för roboten var 50 % i både  $x$ - och  $z$ -led utifrån artikelns abstract. Senare i texten blir det tydligt att positionen för de flesta undersökta robotarna hade lägre energiförbrukning när deras arbetsposition var i höjd med robotens bas eller nedanför denna [19]. Därför hade energiförbrukningen kanske kunnat minskas ytterligare om placeringen av roboten i höjded justerats.

### 3.6.3 Jämförelse med andras resultat

Studenterna utförde endast tester med roboten IRB 4600-60/2.05 då det är den som studenterna bedömer passar bäst för den aktuella produktionslinjen, se kapitel 3.2. Jämförelsen mellan studenternas resultat och resultat från vetenskapliga artiklar kommer att vara kort på grund av att produktionslinjen och roboten som studenterna använder skiljer sig åt från tidigare studerade system. Studenterna bedömer det därför som olämpligt att jämföra resultaten med statistiska metoder.

De experimentella resultaten gäller endast för den undersökta industriroboten och den aktuella produktionslinjen och går inte att generalisera till alla industrirobotar, vilket är en brist med den valda metoden. Detta beror på att det förekommer skillnader i energiförbrukningen mellan olika robotar på grund av skillnader i deras konstruktion [19]. För att underlätta jämförelser med andra vetenskapliga undersökningar borde studenterna ha valt en annan robot som det finns fler studier kring.

### 3.6.4 RAPID, RobotWare och RobotStudio-version

När roboten ska fästa eller lossa en produkt från verktyget med vakuumsugkoppar så används fördröjningar i RAPID för att se till så att det smarta komponenterna Attacher respektive Detacher hinna reagera på signaler från styrenheten. Konsekvensen av detta blir dock att det finns en tid då roboten står stilla i en position utan att röra sig. Detta leder nog till lite onödig energiförbrukning som hade kunnat minskas.

När en mindre fördröjning användes så fungerade fästning och lossning av produkten ibland men inte alltid. Därför var studenterna generösa med fördröjningen för att se till att få en lösning som konsekvent fungerade inför att mätningarna påbörjades. En längre fördröjning som fortfarande fungerade hade säkert kunnat tas fram genom uttömmande prövning av lägre och lägre val av fördröjning. Däremot hade detta tagit tid från andra delar av arbetet i en fas när arbetet snart behövde vara klart.

Att robotprogrammet som används för att göra mätningarna endast har rörelseinstruktioner av typen `MoveJ` och inte `MoveL` kan dessutom innebära ytterligare energibesparingar då rörelsetypen kan förväntas ha stor påverkan på den resulterande energiförbrukningen [21].

Anledningen till detta är dock att `MoveL` ej var möjlig att använda för alla rörelseinstruktioner, då kravet på att TCP ska utföra en linjär rörelse inte alltid kunde uppfyllas givet placeringarna av målen.

När `ClkRead` används för att läsa av en timer så finns det ett valfritt argument `\HighRes` som gör att en avläst tid returneras med en högre tidsupplösning. Däremot är tidsupplösningen redan hög, med en precision på tre decimaler. Därför anser studenterna att ytterligare precision ej är nödvändigt för detta arbete, även fast det hade kunnat ge mer exakta mätdata. För studenternas ändamål blir skillnaden i tid försumbar.

Efter att mätningarna hade påbörjats så släpptes en ny version av RobotStudio: 2026.1. Det kan vara så att denna version har mer realistiska simuleringar eller på annat vis skulle kunna påverka resultatet.

Versionen av RobotWare som användes är, som tidigare nämnts, 6.16.03.00. Initialt valdes samma version som användes i den andra videogenomgången från ABB [36], vilket var version 6.15.05.00. RobotStudio uppmärksammade efter ett tag att en uppdatering var tillgänglig, varefter studenterna installerade denna och sedan gick över till version 6.16.03.00. Detta är den senaste versionen av RobotWare 6.x [32].

Den senaste versionen av RobotWare är 7.21.0. För att använda denna krävs dock en annan typ av styrenhet kallad OmniCore, som är annorlunda jämfört med IRC5, som är den typ av styrenhet som används med RobotWare 6.16.03.00. Det är möjligt att en OmniCore-styrenhet med RobotWare-version 7.21.0 hade resulterat i lägre energiförbrukning.

### 3.6.5 Informationssökning och bearbetning av mätdata

Metoden för informationssökning anser studenterna generellt fungerade bra när det gällde att hitta vetenskapliga artiklar. Vid behov kunde manuell sökning kompletteras med förslag på artiklar från generativ AI.

Att hitta information om RobotStudio var dock svårare, även med hjälp av AI. Officiell dokumentation från ABB var till hjälp men studenterna känner att det hade varit bra med fler

exempel på RAPID-programmering så att personer kan överföra kunskaper från andra programmeringsspråk till RAPID snabbare. Studenterna fick pussla ihop information för att få en samlad bild, vilket tog tid

Mätdata lagras i ett Excel-ark och säkerhetskopierades för att förhindra dataförlust under databehandlingsprocessen. Spridningsdiagram med trendlinje valdes eftersom varje datapunkt visualiseras och kan relateras till andra, samtidigt som arbetet att identifiera mönster i datan underlättas. Samtidigt blir det tydligt när en datapunkt avviker från det generella mönstret.

Eftersom det finns tre oberoende variabler: hastighet, acceleration och accelerationsramp, så hade det varit önskvärt att använda en 3D-graf. Då kan två av dessa variabler varieras och den tredje vara konstant, samtidigt som förhållandet till energiförbrukning, cykeltid och exekveringstid kan studeras.

Då studenterna hade några veckor kvar med arbetet och inte hade några förkunskaper om att använda program för att visualisera data i 3D-grafer så användes traditionella 2D-grafer. Istället fick datapunkterna etiketter som visade värdet på den andra variabeln som varierades.

En 3D-graf hade sannolikt tydligare visat sambanden mellan de tre oberoende variablerna och energiförbrukning, exekveringstid, samt cykeltid. Lösningen studenterna använde gjorde att datapunkterna hamnade i kluster där de blev svåra att skilja åt, se till exempel figur x.

### **3.7 Källkritik**

Avsikten med detta delkapitel är att motivera varför källor som arbetet använder är tillförlitliga. Studenterna identifierar också några möjliga brister med källorna, trots att de i sin helhet ses som tillförlitliga.

Samtliga vetenskapliga artiklar som har valts till detta arbete har genomgått en peer review-process, vilket innebär att andra forskare och sakkunniga har granskat innehållet innan artikeln publiceras. Studenterna anser att detta stärker källornas trovärdighet. Ingen av författarna till de vetenskapliga artiklarna har heller uppgivit några intressekonflikter. Detta innebär så klart inte att det faktiskt kan förekomma dolda intressekonflikter, så därför har studenterna försökt hitta flera vetenskapliga artiklar inom samma område.

Liknande artiklar eftersöktes också för att minska risken att felaktiga slutsatser dras utifrån enskilda artiklars data och resultat. Att källornas övergripande slutsatser kring arbetsmiljö var samstämmiga, att tunga lyft på arbetsplatsen är problematiskt, stärker dessa källors trovärdighet. Även artiklarna om energiförbrukning var generellt samstämmiga i sina slutsatser.

Studenterna ser risker i peer review-processen för [12], [13], [18], [19] och [24]. Respektive vetenskaplig tidskrift använder ett system där granskare, som lämnar in grundliga granskningsrapporter i tid, får kuponger som ger rabatt på avgiften vid publicering i valfri vetenskaplig tidskrift som MDPI ger ut [MDPI]. Studenterna ser därför en risk att kvaliteten på granskningarna sänks för ekonomiska vinster.

Källan om MDPI:s riktlinjer för granskare [45] anser studenterna som trovärdig då den kommer från den officiella webbsidan för utgivaren av artiklarna [12], [13], [18], [19] och [24] riktar sig till personer som ska utföra en granskning av någon annans arbete..

Ett antal källor kommer också från myndigheter. Studenterna anser att dessa också har hög trovärdighet då de har lagstadgade krav på saklighet och opartiskhet [46], [47]. Flera av dessa källor källhänvisar också när de inte använder sin egen data eller statistik [3], [7], [8], [9], [17] vilket ökar transparensen.

Några lagar används också som källor, dels för att styrka varför ett alternativt arbete uteslöts, dels varför källorna från myndigheter är trovärdiga. Dessa anses som mycket trovärdiga då informationen kommer från Riksdagens webbsida, som även stämmer överens med Regeringskansliets rättsdatabaser. Regeringskansliets rättschef är utgivare av Svensk författningssamling [Författning], som Riksdagen har beslutat att lagar och förordningar ska kungöras i [Lag].

Dokumentation, videogenomgångar och information i RobotStudio används också som källor. Dessa källor har ABB skapat och studenterna anser att de är trovärdiga då ABB är utvecklarna bakom programmet RobotStudio. Däremot har ABB affärsintressen av hur deras produkter och tjänster framställs. Därför är det inte nödvändigtvis så att deras beskrivningar

är helt objektiva och sakliga. Därför har studenterna själv prövat de påståenden som de hänvisar till, för att bekräfta att de stämmer.

Videogenomgångarna från allmänheten ser studenterna inte som lika trovärdiga givet att studenterna inte vet vilka som ligger bakom informationen och därför inte vilka intressen de har. Därför har studenterna bara använt dessa källor som vägledning för att lära sig RobotStudio. Sedan har information bekräftats med mer trovärdiga källor som ABB:s officiella dokumentation.

ISO-standarden som används anser studenterna är trovärdig, på grund av att ISO-standarder tas fram och förhandlas inom kommittéer där experter från flera intressenter kan delta, däribland företag, konsumentföreningar, icke-statliga organisationer, universitet och forskningsinstitut [48]. Detta innebär förhoppningsvis en standard och definitioner som många intressenter accepterar och förhåller sig i sitt språkbruk.

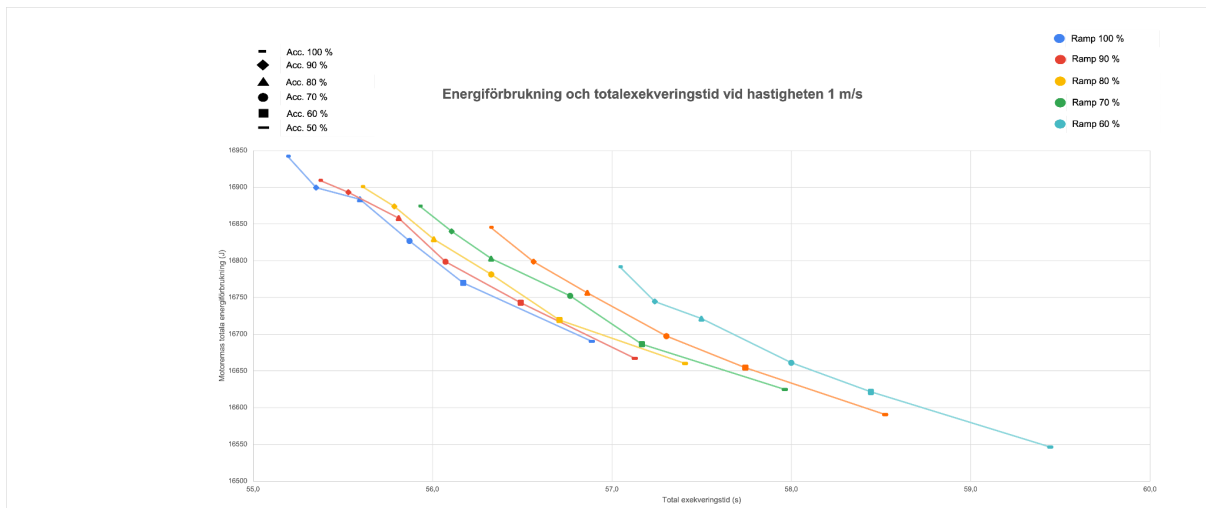
Avslutningsvis har information från IFR - International Federation of Robotics använts för statistik om antalet industriella robotar som är i bruk. IFR är en organisation med medlemmar från olika företag inom robotindustrin och forskningsinstitut som bland annat arbetar för att främja robotindustrin och informera allmänheten och makthavare [51]. Studenterna ser en uppenbar intressekonflikt här när det gäller organisationens framställning av industrirobotar som framgångsrika. Dock bedömer studenterna att informationen som de hänvisar till troligtvis är sann då den stämmer överens med den generella bilden som [15], [Liu] och [Mahajan] ger.

De flesta källorna i form av vetenskapliga källor kopplade till robotik är max 5 år gamla. Detta är viktigt då utvecklingen inom robotik går snabbt [Liu]. Vissa vetenskapliga artiklar och myndighetsrapporter är äldre. Däremot bygger slutsatserna på en gemensam bild där både äldre och nyare källor är samstämmiga: 1. Det finns risk för belastningsskador på jobbet som kan påverka hälsa och arbetsförmåga. 2. Det finns möjligheter att minska robotars energiförbrukning som är värda att undersöka.

## 4. Analys

### 4.1 Analys av resultat

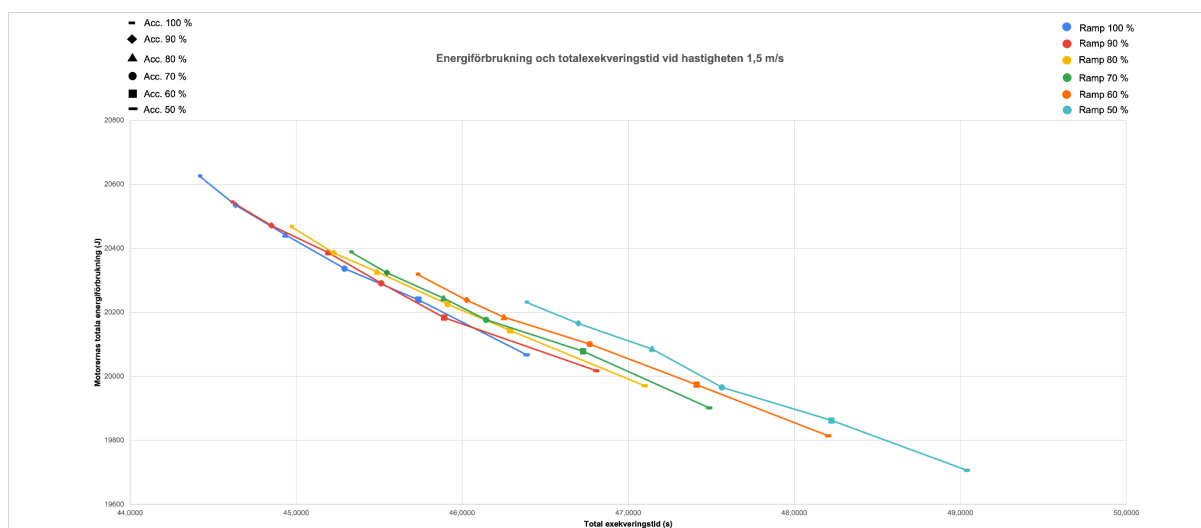
Genom att erhålla de uppmätta värdena hittades följande resultat: För det första är tid och hastighet positivt korrelerade. Ju större energiförbrukning, desto kortare arbetstid. Samtidigt korrelerade tid och acceleration också linjärt positivt. Näst är tid och accelerationsramp också linjärt positivt korrelerade. Detta innebär att ju längre körtid och ju lägre hastighet, desto lägre energiförbrukning. Men om hastigheten är för låg kan det lätt påverka produktionseffektiviteten. Detta kan lätt orsaka förseningar i produktionsprocessen, vilket resulterar i förluster.



**Figur 14.**  $y$ -axeln är energiförbrukning (J) och  $x$ -led är körtid (s). Ju högre energiförbrukningen är, desto kortare blir körtiden.

Vid en hastighet på 1000 visar detta diagram all data. När accelerationen minskar, som visas i diagrammet, ökar den totala arbetstiden gradvis medan energiförbrukningen kontinuerligt minskar. Vidare, om man jämför samma acceleration, som Acc100, är hastigheterna desamma; den enda skillnaden är accelerationsrampen, som ändras från 100 till 50. Resultaten är desamma: arbetstiden ökar medan energiförbrukningen minskar.

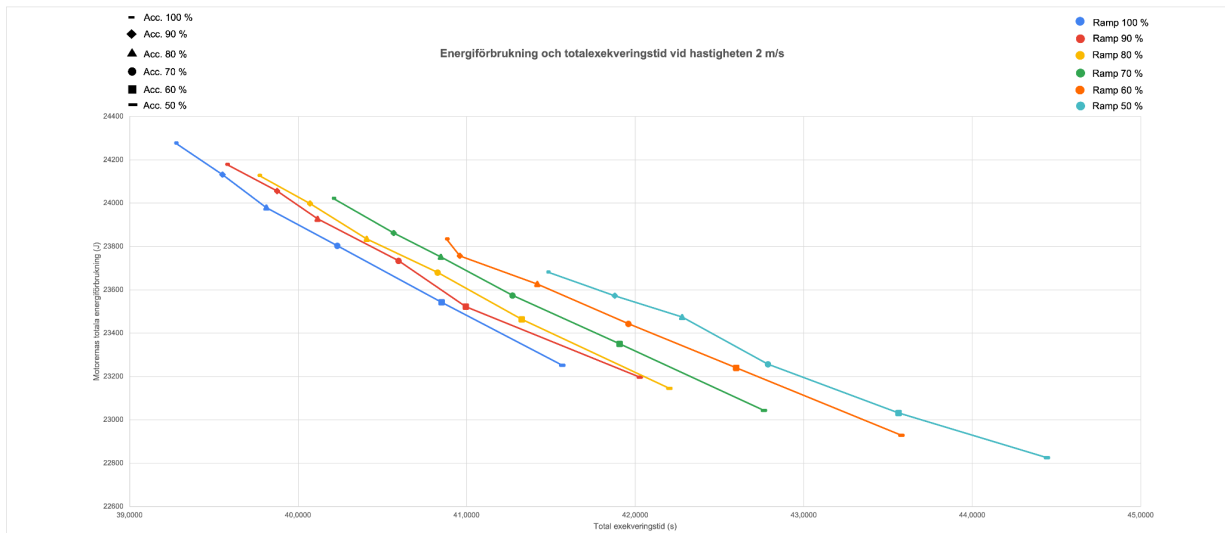
Nedan finns tre diagram som visar hastighet 1500, 2000 samt alla data.



**Figur 15.** V1500 y-axeln är energiförbrukning (J) och x-led är körtid (s). Energiförbrukning har linjär relation med arbetstid

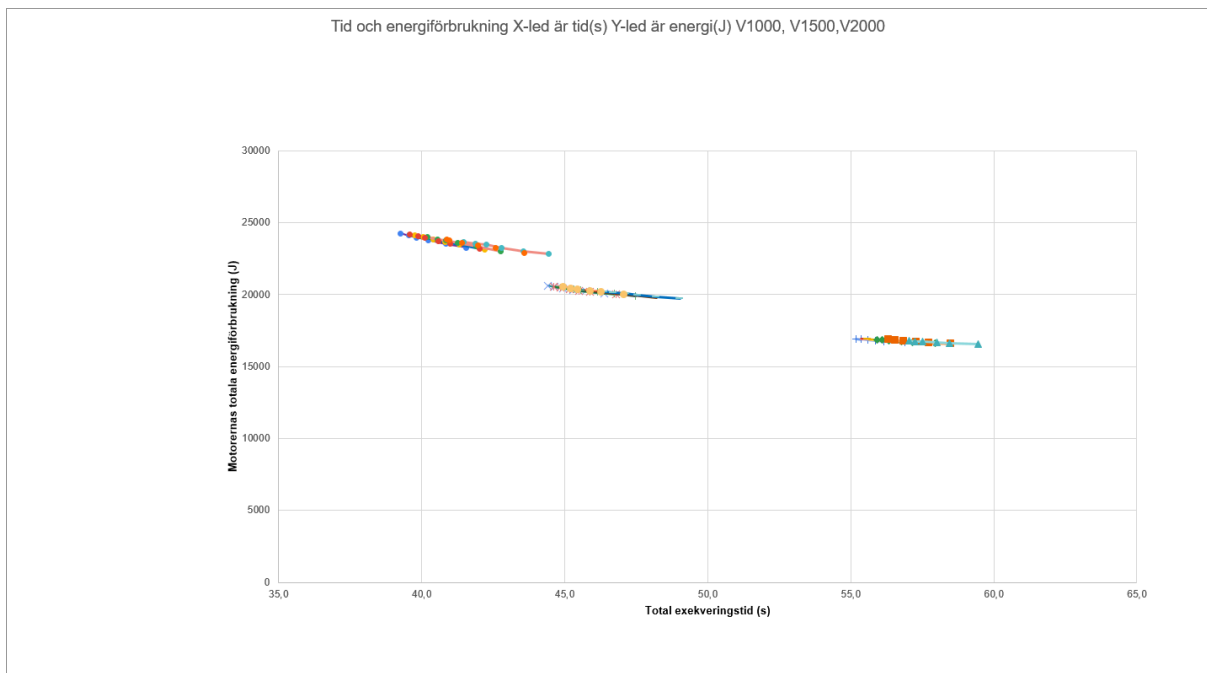
Grafmönstret för V1500 liknar V1000:s och visar också ett linjärt samband: högre energiförbrukning, kortare driftstid och större hastighet, acceleration och accelerationsramp.

Samtidigt, vid en hastighet på 1500, är den maximala energiförbrukningen 20626,25 J för ramp 100 och acceleration 100, med en körtid på 44,415 sekunder. Den lägsta energiförbrukningen är 19706,94 för ramp 50 och acceleration 50, med en körtid på 49,039 sekunder.



**Figur 16.** V2000 y-axeln är energiförbrukning (J) och x-led är tid. Arbetstids skillnad är tydligare än **Figur 15** och **14**

När hastigheten är 2000 liknar hela processen en regression till linjäritet. ju lägre hastighet, desto lägre acceleration, desto lägre accelerationsramp, desto mindre energi förbrukas och desto längre tid.



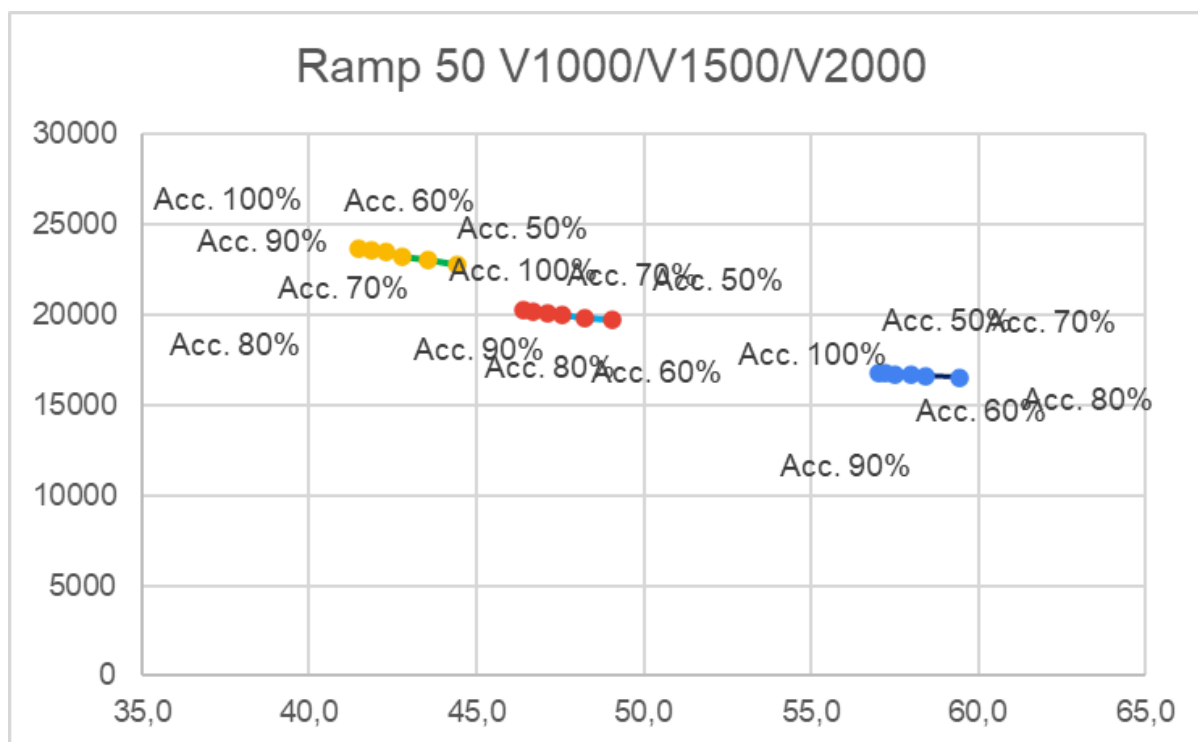
**Figur 17.** V1000, V1500, V2000 y-axeln är energiförbrukning (J) och x-led är tid

Resultatet visar tydligt att energiförbrukning blir hög när hastigheten blir hög. Samtidigt kan energiförbrukning bli relativt låg när hastigheten och accelerationen är låg även om det tar längre tid.

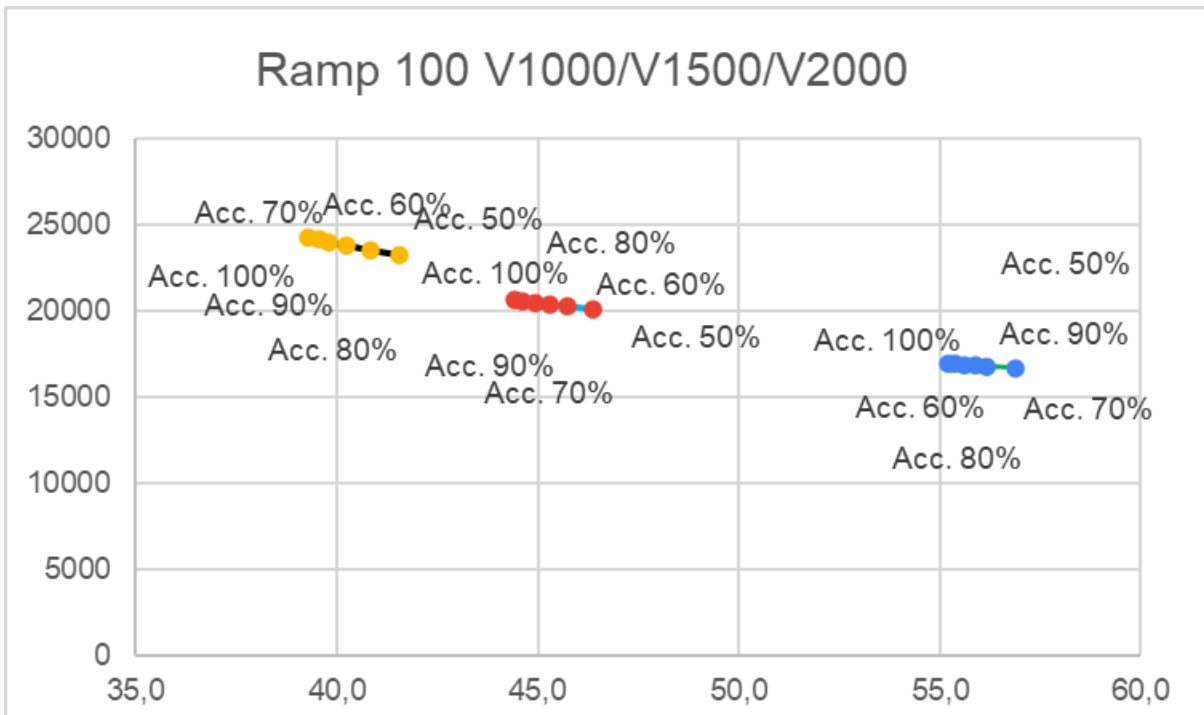
Maximal energiförbrukning är 24277,91J när hastighet är 2000, accelerationsramp är 100 samt acceleration är 100. Tid används 39,271S. Minium energiförbrukning är 16546,28J när hastighet är 1000, accelerationsramp är 50 samt acceleration är 50. Tiden är 59,443S.

Mätresultaten visar att energiförbrukning och tid har ett linjärt samband, men detta beror på variablerna hastighet/acceleration och accelerationsramp.

För att säkerställa mätnoggrannheten jämfördes återigen effekterna av olika hastigheter på energiförbrukning och tid då rampen var mycket låg.

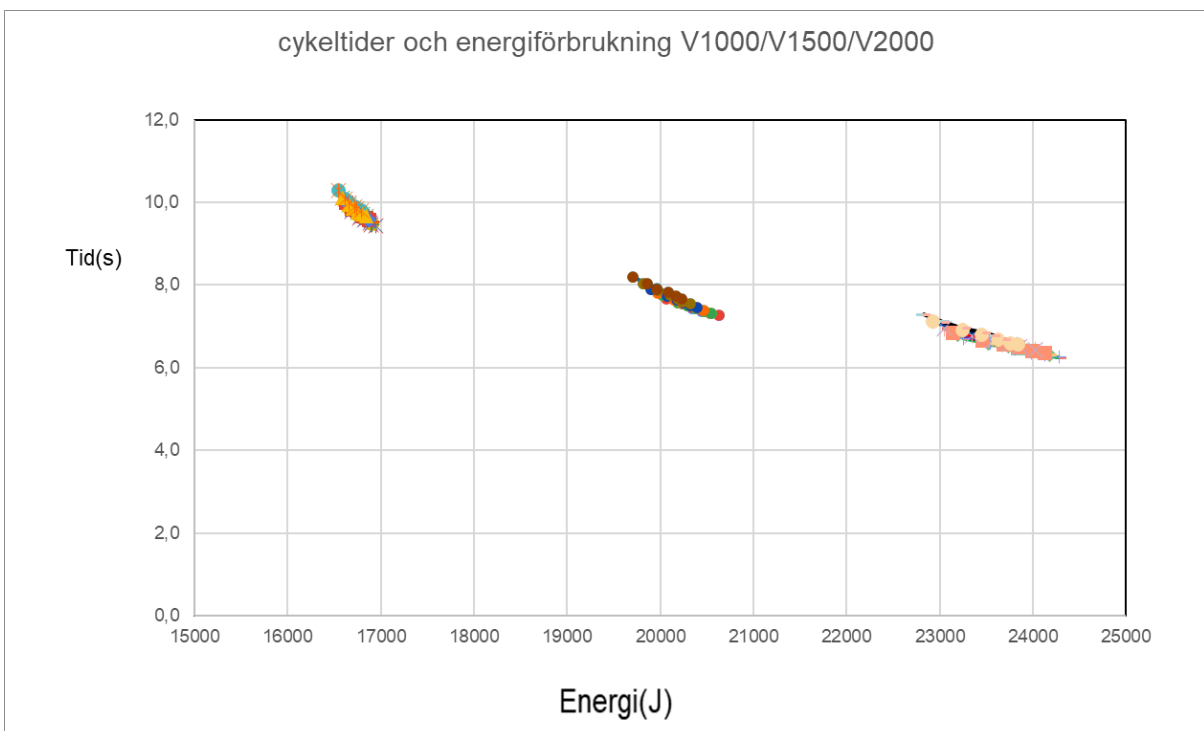


**Figur 18.** Acceleration ramp 50% gul färg är v2000, röd färg är v1500 och blå färg är v1000



**Figur 19.** Acceleration ramp 100% gul punkter är v2000, röda punkter är v1500, blåa punkter är v1000

Härnäst jämförs sambandet mellan energiförbrukning och cykeltid.



**Figur 20.** X-led är energiförbrukning och Y-led är Cykeltider

Härnäst jämförs sambandet mellan energiförbrukning och cykeltid. Ju längre cykeltiden varar, desto mindre energi förbrukar den.

Resultaten är följande: med accelerationsramp på ett fast värde varierades hastigheten och tids- och energiförbrukningen observerades. Resultaten visar att accelerationsramp, tiden och energiförbrukningen förblir linjärt relaterade; det vill säga ju mindre accelerationsramp är, desto mindre energi förbrukas och desto längre arbetstid. Omvänt, ju mer energi förbrukas och desto kortare arbetstid.

För att säkerställa resultatens noggrannhet, stabilitet och relativa vetenskapliga validitet refererade studenterna till flera liknande experimentella data, varav den mest typiska är [49], som beskrivs och analyseras i deras artikel. Andra simuleringsresultat visar också att bra val av hastighet och acceleration kan ge betydande minskning av energiförbrukning [49]. Även om denna artikel endast nämner effekterna av hastighet och acceleration på resultaten, och inte nämner accelerationsramp bekräftar denna slutsats indirekt noggrannheten i våra experimentella resultat. Den styrker också vår experimentella hypotes. Valet av hastighet och acceleration har en betydande inverkan på energiförbrukningen.

För att se till att korrekt mätvärde fås och undvika tidigare körning påverkar mätvärdena i denna gång brukar robotstudio startas om efter att mätvärde data fås. Reset knapp i robotstudio kan inte riktigt startar om alla komponenter och ledde till att statistik och värde är har mycket systematisk fel.

Resultaten visar att minskningen av hastighet har störst inverkan på energiförbrukningen, medan minskningarna av acceleration och accelerationsramp har begränsade effekter. När hastigheten minskar från 2000 till 1000, och accelerationen och accelerationsramp förblir konstanta vid 100, minskar energiförbrukningen från 24277,91 J till 16942,28 J, en minskning med cirka 30 %. När hastigheten förblir konstant, till exempel vid 1000, och accelerationen minskar från 100 till 50 medan accelerationsramp förblir konstant vid 100, minskar energiförbrukningen från 16942,28 J till 16690,13 J, en minskning med cirka 1.4 %. Om hastigheten förblir konstant vid 1000, och accelerationen förblir konstant vid 100, och accelerationsramp minskar från 100 till 50, minskar energiförbrukningen från 16942,28 J till 16791,79 J, en minskning med cirka 0.89 %.

Baserat på de experimentella resultaten kan slutsatsen därför dras att när experimentmiljön och belastningen är desamma, och systemet fungerar normalt utan problem, kommer hög

hastighet, hög acceleration och hög accelerationsramp att få roboten att generera större vridmoment på kort tid. Detta innebär att robotens effekt blir mycket hög på kort tid, och därmed kommer energiförbrukningen att bli mycket hög på mycket kort tid.

Tvärtom, under låg hastighet eller låg acceleration, kommer motorns vridmoment inte att vara särskilt högt inom en kort tidsperiod resulterar i låg effekt och låg energiförbrukning.

Att justera sensorns position var också en utmaning. Ursprungligen plan är att placera sensorn under bandet så att den skulle upptäcka produkten som passerade ovanför. Men på grund av felaktig placering eller problem med stationens logik misslyckades sensorn ibland med att upptäcka produkten och kunde inte aktivera RobotsArm i tid.

Att välja rätt bandstorlek var också ett problem. Tidigare, på grund av emulatorbegränsningar, rörde bandet inte marken utan svävade faktiskt i luften för överföring. Denna allvarliga fysikbrytande bugg upptäcktes under våra tester, så bandstorleken ändrades.

I början av experimentet bestämdes planen för att mäta förflyttningen av en produkt från band 1 till band 2. Resultaten visade dock att det var svårt att mäta skillnaden i energiförbrukning, eftersom skillnaden var mycket liten, ibland bara några få joule. Därför gjorde vi två förbättringar. Först designades storlek om produkten och ökade dess storlek och vikt från 5 kg till 40 kg. För det andra ändrades förflyttningen av en produkt till att flytta fem produkter. Den här gången blev skillnaden i mätresultaten mycket stor, och skillnaden och magnituden observerades lätt från både bilder och data. Denna mätmetod tar också fullt ut hänsyn till produktionslinjens intensitet och skala. Att flytta fler produkter skulle ytterligare förstärka skillnaden i energiförbrukning, vilket gör optimeringen av vår mätmetod avgörande.

Emellertid uppstår ett nytt problem: att flytta fem produkter gör det svårt att manuellt styra när robotarmen stannar och registrera data utan att använda RAPID. Att bara flytta en produkt är relativt enkelt att styra; titta helt enkelt på simulatören när robotarmen rör sig från band 1 till band 2, placerar produkten, flyttar tillbaka till band 1 och sedan klickar på "stopp". Att flytta fem produkter gör det dock svårt att kontrollera när man ska stoppa. I slutändan, på grund av betydande skillnader i flera mätningar och resultat, övervägdes det att lägga till en timer för att automatiskt styra när robotarmen stannar. Robotarmen plockar upp produkten

från band 1, flyttar den till band 2, placerar den och återgår till band 1 en gång, och upprepar denna fram-och-tillbaka-rörelse fem gånger. Vid den sista rörelsen registrerar timern tiden och stoppar robotarmen. Detta minimerar felet för varje rörelse, vilket gör det tillräckligt för vår forskning.

För att åtgärda problemet med att gripdonet ibland inte lyckades plocka upp produkten utfördes ett flertal tester, och vissa ABB-manualer konsulterades under processens gång. Resultaten visade att felaktig gripdonsplacering var bidragen till att produkten inte kunde greppas; specifikt fanns det ett mellanrum mellan gripdonet och produkten, vilket förhindrade att gripdonet fick kontakt.

Angående frågan om Mount, Det är svårt att förstå den från början till slut, mount enligt [25] förklarar, om det är sant kommer objekt fastna på parentes. Men i projektet funkade det inte. När Mount stängdes den funkade det.

I allmänhet kan man inte blint sträva efter låg energiförbrukning, eftersom detta också kommer att leda till låg effektivitet Det är inte så bra och man kan inte heller överdrivet sträva efter effektivitet på bekostnad av hög energiförbrukning. Därför måste en balans hittas mellan energiförbrukning och hastighet för att beakta både energiförbrukning och effektivitet.

Under hela projektet hindrade även vissa utmanande problem den experimentella processen. Ett klassiskt exempel är att robotar med låg belastning konsekvent har mycket låg energiförbrukning, vilket resulterar i mycket små skillnader i mätresultaten. Att använda robotar med högre belastningar ökar dock variationen i energiförbrukning, vilket ökar sannolikheten för att observera parametrar och deras samband med resultaten. Därför togs IBR 1410-seriens robot med en maximal nyttolast på 5 kg bort, och IBR 4600-seriens robot med en maximal nyttolast på 60 kg återskapades och installerades på lämplig plats.

Sammanfattningsvis visar analysen och resultaten att rörelseparametrar som hastighet, acceleration och accelerationsramp har en betydande inverkan på energiförbrukning och

cykeltid. Resultaten belyser också vikten av att optimera dessa parametrar för att uppnå en balans mellan energieffektivitet och produktivitet i industriella robotsystem.

### **4.3 Förbättringsmöjligheter**

Om projektet hade haft tillräckligt med tid skulle projektet ha kunnat utvecklas mycket mer. Till exempel skulle lösningar ha undersökts för vår ursprungliga plan. En stor anledning till att den ursprungliga planen gavs upp var tidsbristen. Planen omfattade två typer av fyra robotar, så det tog mycket tid att förstå och lära oss hur man styr olika robotar och får dem att samarbeta med varandra. Samtidigt kommer MOVEJ och MOVEL att beaktas i projektet. Vi kommer också att observera hur MOVEJ och MOVEL påverkar energiförbrukning och tid.

Sammanfattning Tvist visade resultaten att lägre acceleration och lägre ramp gav längre körtid och längre cykeltid i samtliga testerna. Samtidigt minskade energiförbrukningen successivt när hastighet, acceleration eller ramp sänktes. De högsta energivärdet återfanns i test v2000, medan lägre energivärden registrerades vid lägre acceleration och lägre ramp. De upprepade mätningarna visade endast små avvikelser mellan körningarna.

## **5. Resultat**

### **5.1 Excel och Mätning**

Om mätning 1 och mätning 2

De två upprepade mätningarna gav genomgående liknande resultat. Skillnaderna mellan mätning 1 och mätning 2 var i de flesta fall små för både körtid, avläst tid, energiförbrukning och cykeltid. Anledningen är att undvika systematiska fel så mycket som möjligt. Efter att ha utfört en mätning i Mätning 1 måste du starta om RobotStudio innan du kan utföra en mätning i Mätning 2. Alla mätningar skrivs i Excel och sparas i Appendix C.

### **5.2 Diagram**

Med hjälp av Excel används spridningsdiagram och linjediagram för att jämföra data. Linjardiagram kan bättre hjälpa till att analysera datatrender och spridningsdiagram kan bättre hjälpa till att observera förändringar i datastorlek. Effekterna av cykeltid och energiförbrukning, och effekterna av körtid (totaltid) på energiförbrukning jämfördes.

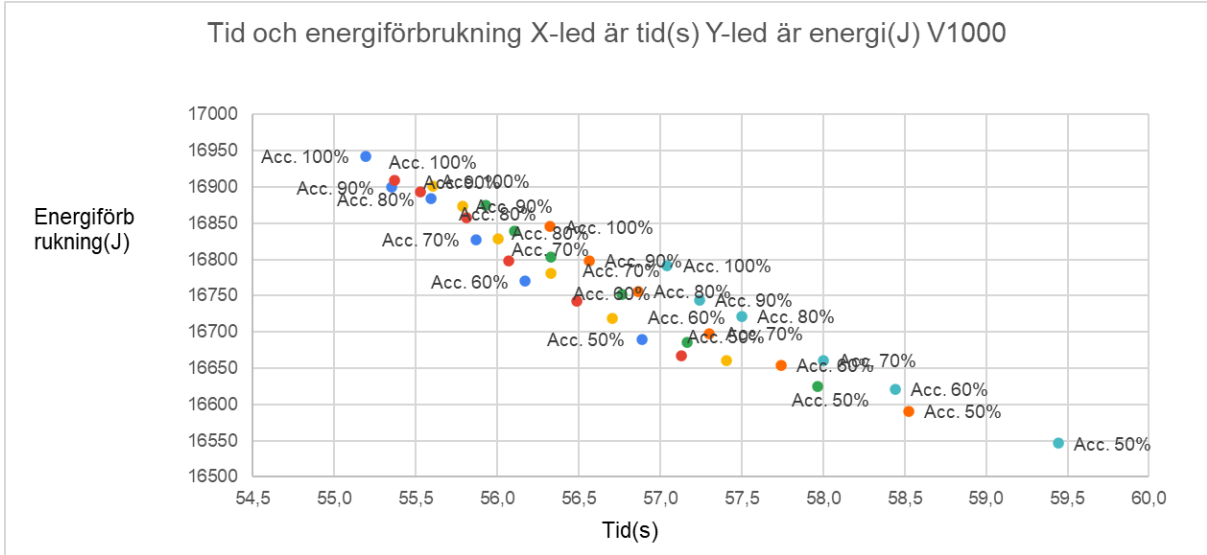
I grundserien kunde det observeras att körtiden ökade när acceleration och ramp sänktes. Vid Ramp 100 % och Acc. 100 % var körtiden 55.191 s i mätning 1, medan motsvarande värde vid Ramp 100 % och Acc. 50 % var 56,887 s. Samtidigt minskade energiförbrukningen från 16 942,28 J till 16690,13 J. Cykeltiderna ökade under samma jämförelse från cirka 9,434 s till cirka 9,772 s. Följande är alla data i diagram i V1000. **Figur 21** visar bara punkter utan linjär och **Figur 14** visar punkter och linjär

I **Figur 15** och **22** v1500 var körtiderna generellt kortare än i grundserien. Vid Ramp 100 % och Acc. 100 % registrerades en körtid på 44,415 s och en energiförbrukning på 20626,25 J. När inställningarna sänktes till Ramp 50 % och Acc. 50 % ökade körtiden till 49,039 s, medan energiförbrukningen minskade till 19706,94J.

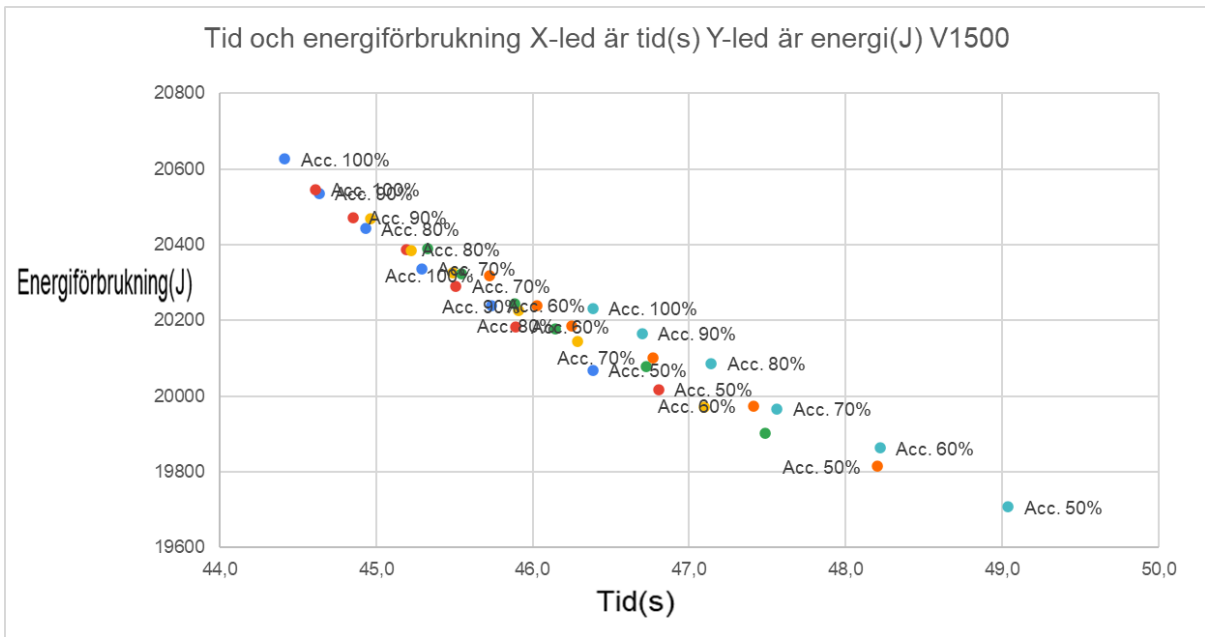
I **Figur 23** och **Figur 16** kan man se att v2000 registrerades de lägsta cykeltiderna i materialet. Vid Ramp 100 % och Acc. 100 % var cykeltiden 6,257 s och energiförbrukningen 24277,91 J. Vid Ramp 50 % och Acc. 50 % ökade cykeltiden till 7,279 s, medan energiförbrukningen minskade till 22825,99 J. Körtiden ökade under samma jämförelse från 39,271 s till 44,43 s.

Som visas i **figur 24, 25** och **26** har cykeltiden och energiförbrukningen ett linjärt samband. Ju högre energiförbrukningen är, desto kortare är cykeltiden.

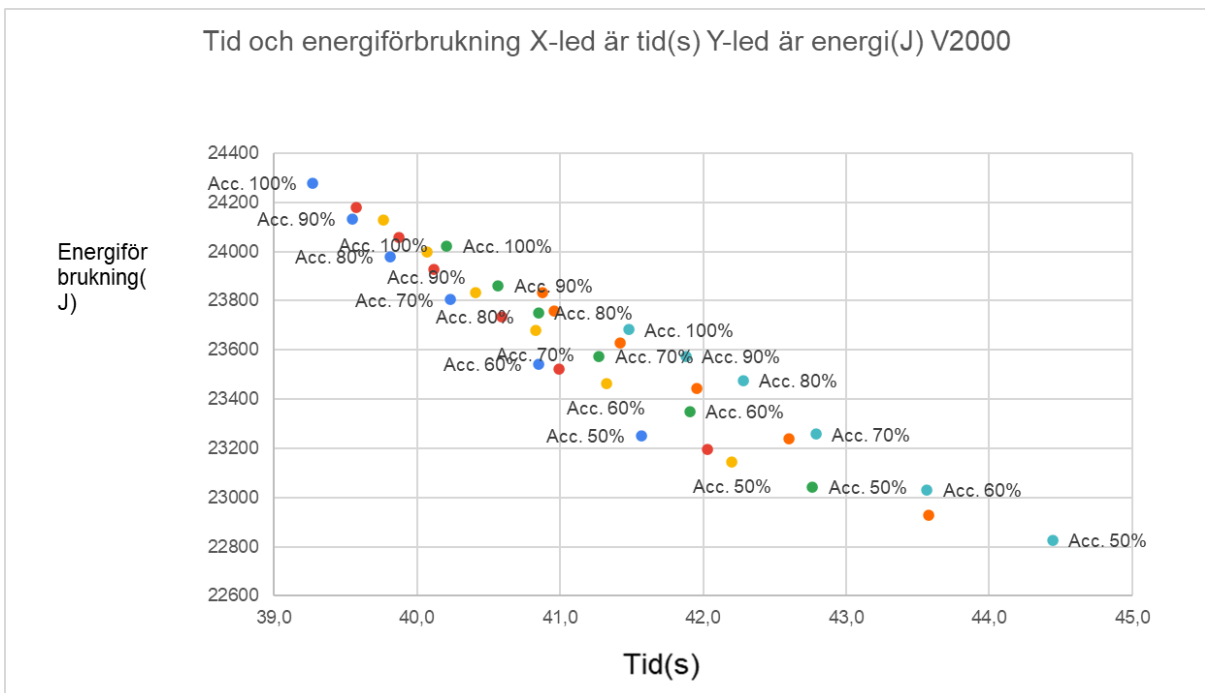
Parametrarna med lägst energiförbrukning är hastighet 1000, acceleration 50 och ramp 50.



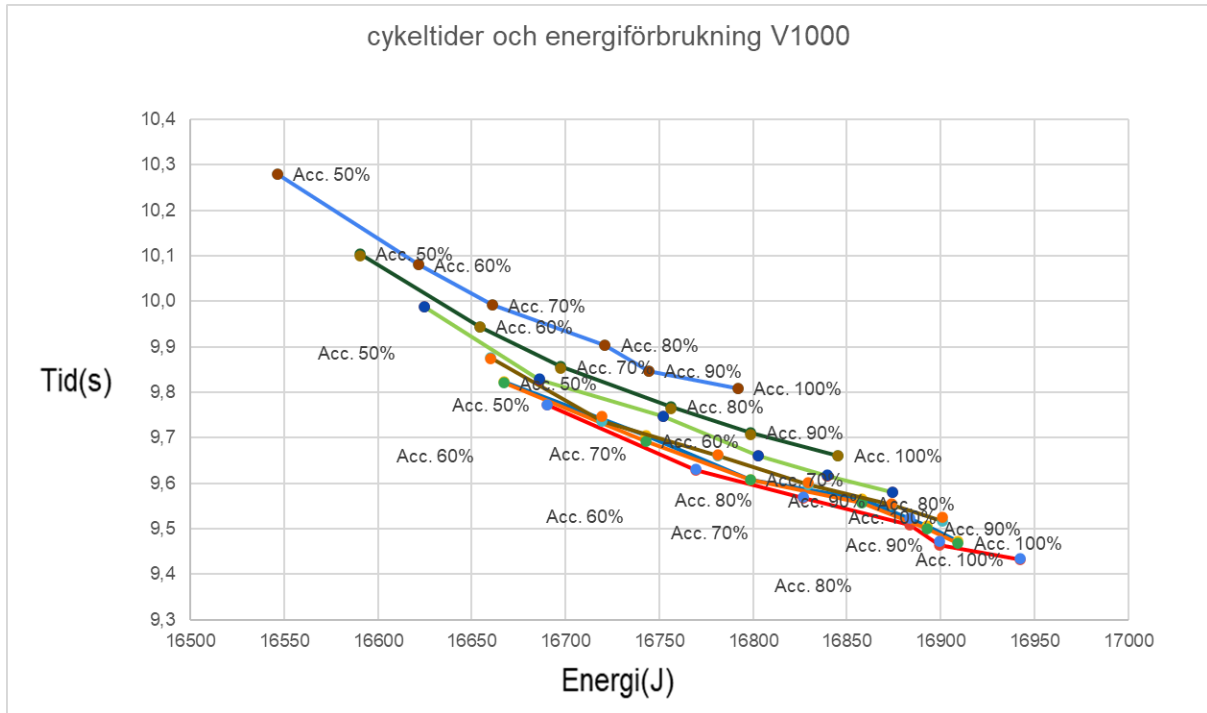
**Figur 21.** V1000 Ramp 100% till 50% punktdiagram



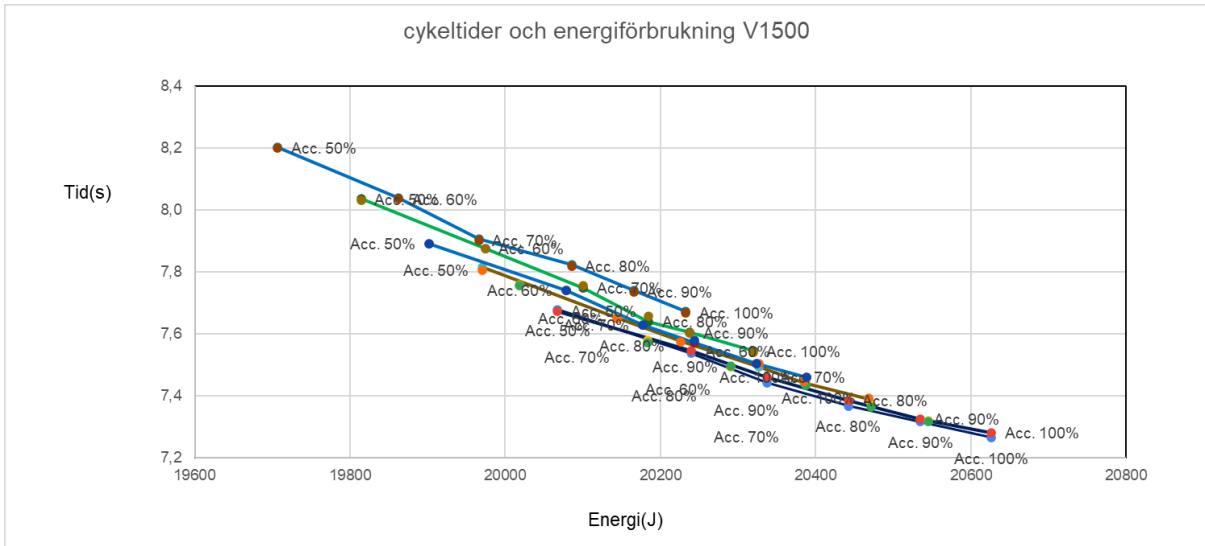
Figur 22. V1500 Ramp 100% till 50% punktdiagram



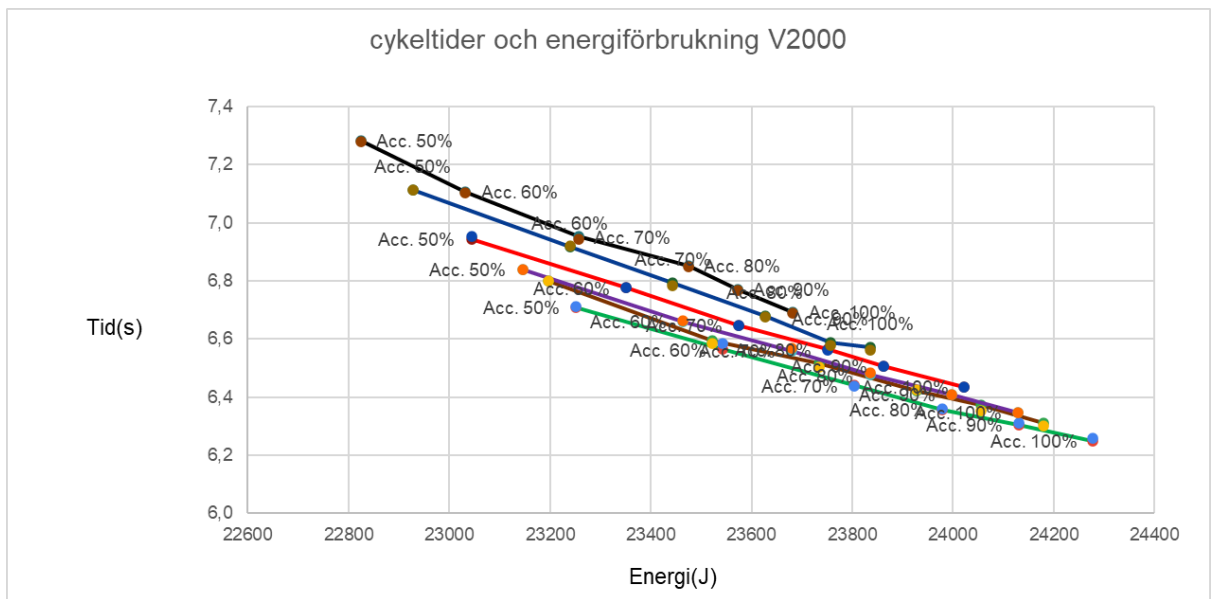
**Figur 23. V2000 Ramp 100% till 50% punktdiagram**



**Figur 24. Cykeltider och Energiförbrukning när hastighet är 1000**



**Figur 25. Cykeltider och Energiförbrukning när hastighet är 1500**



**Figur 26. Cykeltider och Energiförbrukning när hastighet är 2000**

## 6. Slutsats

### 6.1 Diskussion

Hela projektets experimentperiod var mindre än fem månader, vilket inte var särskilt lång, men resultaten var generellt värdefulla och relativt noggranna. Resultaten är rimliga.

Hela den experimentella processen var kantad av diverse svårigheter. Efter att ha övervunnit otaliga problem och utmaningar erhöles mödosamt förvärvade experimentella data och resultat.

Forskningen har fortfarande många begränsningar. Till exempel saknades riktiga industrirobotar i verkligheten eller produktionslinjer i verkligheten för felsökning och inläring. Alla projektkomponenter och kod felsöktes i en simulator. Därför kan detta ha en viss inverkan på våra experimentella resultat. Till exempel tog simulatorm inte hänsyn till effekterna av luftmotstånd eller friktion på experimentet, och inte heller till situationer som kan uppstå i verkligheten, såsom otillräcklig spänning, plötsliga strömavbrott eller andra oväntade händelser. Alla experimentella förhållanden var under ideala förhållanden.

Den praktiska betydelsen av detta experiment ligger i att utforska och adressera effekten av robots rörelsens acceleration och accelerationsramp på energiförbrukningen. Många nuvarande artiklar och studier fokuserar främst på roboteffektivitet eller förhållandet mellan effektivitet och energi och försummar eller underskattar påverkan av acceleration och accelerationsramp på energiförbrukningen. I dagens värld med minskande fossila bränslen och behovet av att minska koldioxidutsläppen blir det allt viktigare att minska energiförbrukningen hos industrirobotar. Med utvecklingen av AI teknik kommer en dag en stor del av den globala produktionen att vara beroende av industrirobotar vilket resulterar i enorm energiförbrukning. Därför är det, inför denna stora robotutveckling, både teoretiskt

nödvändigt och meningsfullt att utforska denna fråga, särskilt i praktisk produktion och vardagsliv.

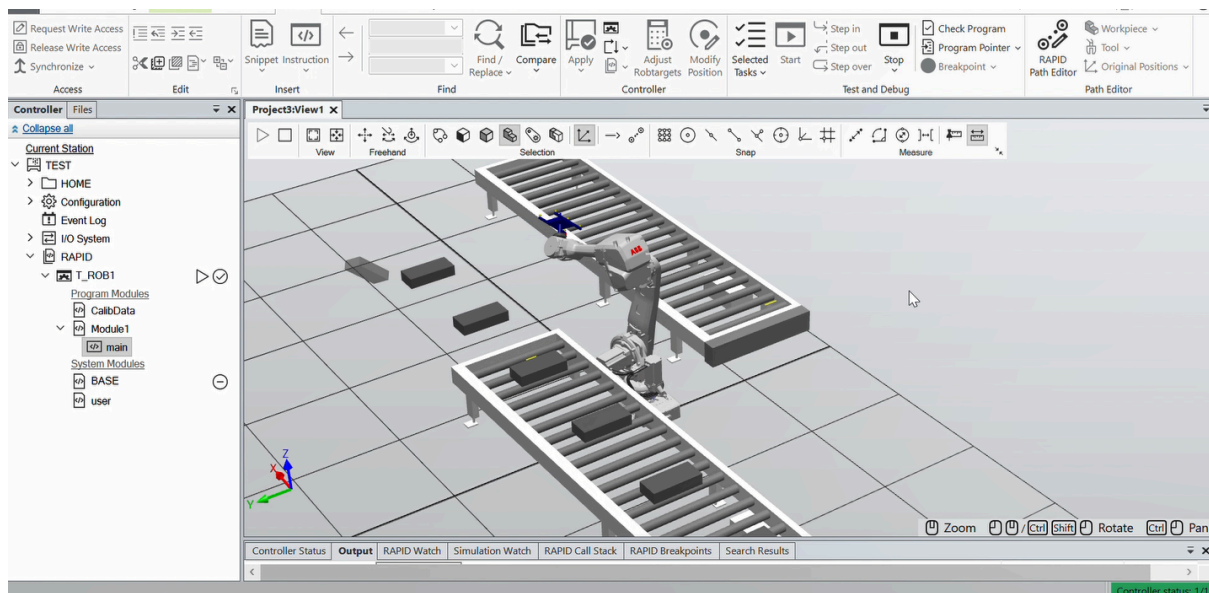
För den framtida utvecklingen av detta projekt kan olika värden och robotbanor om tiden tillåter testas för att undersöka effekten av olika banfaktorer på energiförbrukningen, samtidigt som data beaktas acceleration och accelerationsramp. Dessutom kommer projektet att inkludera fler produktionslinjer och robotar för att göra det så likt en riktig fabrik som möjligt i vardagen. Dessutom, om förhållandena tillåter är det lämpligt att utforska och felsöka en riktig RobotArm IRB4600-serierobot snarare än att bara fokusera på simulatordrift, och därmed beakta faktorer som luftmotstånd och friktion för att hantera energiförbrukningsproblem i verklig produktion och vardagsliv så mycket som möjligt.

När det gäller generaliseringen av resultaten kan resultaten av detta experiment fungera som en objektiv referens för små produktionslinjer eller akademisk forskning. Detta beror på att experimentet, som endast använder en robotarm i IRB4600-serien, har vissa begränsningar. För mindre robotar, såsom 1410-serien, som bara kan lyfta 5 kg, är resultatens inverkan begränsad. Generaliserbarheten för detta projekt är relativt rimlig, men det är också baserat på en specifik modell och en specifik uppgift (att flytta produkten från transportband 1 till transportband 2). De experimentella resultaten är baserade på den specifika robotmodellen IRB4600-serien och den specifika produkten 40 kg; därför kan resultaten tolkas som en indikation på en allmän trend.

## **6.2 Utmaning**

Projektet har många utmaningar när Robotstudio användes. Till exempel var robotar mens bana i vissa fall väldigt konstig; efter att ha bestämt startpunkten och slutpunkten roterade robotarmen kontinuerligt längs banan. Dessutom bröt produkten ibland mot fysikens lagar, eller så nådde den givaren men stannade inte, och fortsatte att röra sig framåt även efter att ha överskridit bandets räckvidd. Dessutom fungerade doattach felaktigt, och robotarmen kunde inte fästas vid produkten.

Genom kontinuerliga misslyckanden och utforskning slutfördes slutligen experimentet med ihärdig ansträngning.



**Figur 27.** Ett misstag i Robotstudio

Ur ett industriellt perspektiv är detta projekt baserat på vetenskapliga experiment och industriell inriktning. Därför är dessa resultat användbara för beslutsfattande under robotprogrammering och kan inspirera operatörer eller ingenjörer, som kan använda denna information som grund för ett projekt. För olika problem, såsom energieffektivitet eller produktionshastighet, kan lämpliga parametrar och data väljas från våra resultat. Detta hjälper till att undvika onödiga eller liknande fel i vår projektprocess.

Gällande optimering var optimeringen i detta projekt inte idealisk. Eftersom studenterna använde ABB RobotStudio för första gången och inte var särskilt skickliga, uppstod teknisk komplikation ofta under användningen. Till exempel, trots att fästet var anslutet till sugkoppen i logiskschemat, fastnade inte sugkoppen fortfarande vid produkten under rörelse. Efter flera modifieringar och kodefelsökning visade det slutliga experimentella resultatet att robotarmen kunde fungera, men det fanns fortfarande ett potentiellt fel i den digitala displayens position. Sammantaget, med tillräckligt med tid, kan detta projekt optimeras och förbättras ytterligare. Det hade varit intressant att även studera produktionslinjen i figur 6.

Inledningsvis var planen att konstruera en produktionslinje bestående av tre transportband och fyra olika robotar. Syftet var att undersöka interaktionen mellan robotar och band med särskilt fokus på effekterna av varierande bandhastigheter. Denna plan avskrevs dock eftersom målsättningen bedömdes vara för otydlig och projektet ansågs sakna tillräcklig komplexitet för att adressera verkliga industriella problem. Att endast justera hastighetsparametrar ansågs inte uppfylla kraven för en examensarbetsnivå. Som ett resultat av detta reviderades strategin till förmån för den nuvarande projektplanen.

Även om det ursprungliga Projekt-projektet lades ner, behölls det som en potentiell framtida integration med det befintliga Projekt-projektet för att skapa ett mer avancerat, komplext och tekniskt sofistikerat projekt.

### **6.3 Mål och problem**

Mål finns två 1. Begränsa värdeintervallet och hitta parametervärdet med den lägsta energiförbrukningen inom det intervallet.

2. Målet är att automatisera den slutliga processen.

Problemen liknar de som beskrivs i Problemformulering:

1 Hur påverkar parametrarna hastighet, acceleration och accelerationsramp cykeltid och robotens energiförbrukning?

2 Stämmer resultaten överens med slutsatser från vetenskapliga artiklar som gjort liknande undersökningar?

Dessa två problem har till stor del lösts. Vi drog slutsatsen att effekterna av acceleration, ramp och acceleration på energiförbrukningen är linjära; ju större energiförbrukningen är,

desto större är effekterna av acceleration, ramp och acceleration. Acceleration har störst inverkan på energiförbrukningen, följt av acceleration och sedan rampning.

Vi hittade även andra artiklar som bekräftar våra resultat. Därför kan vi säga att dessa två problem i huvudsak är lösta.

Eftersom den ursprungliga projektplanen var för stor (i den ursprungliga planen övervägde vi minst 3 till 4 robotband och minst 2 till 3 olika typer av robotar) var vi tvungna att ta hänsyn till för många faktorer i början. Som ett resultat upptäckte vi att vissa processer kanske inte kunde slutföras i tid efter att projektet startat. Därför var vi tvungna att minska projektets storlek ytterligare, och slutligen, efter ytterligare minskning, blev det en situation med två robotband och en robotarm. Om vi har möjlighet att fortsätta projektet i framtiden kan de nuvarande resultaten tjäna som en bra grund för dess framtida utveckling.

## **6.4 Reflektion över etiska aspekter**

Detta delkapitel reflekterar över det etiska dilemma som uppstår när robotar utför arbetsuppgifter som traditionellt utförts för hand av personal på en arbetsplats. Finns det en risk att personer kan bli arbetslösa på grund av detta? En vanlig uppfattning, som säkert är bekant, är nämligen att digital teknik kommer ersätta människor [Kumar] och deras arbetsuppgifter.

Att allt fler industriella robotar används inom tillverkningsindustrin är ingen nyhet utan snarare en trend. År 2017 var ungefär 3 gånger så många industriella robotar i bruk inom tillverkningsindustrin runt om i världen jämfört med år 2002. Från 2009 till och med år 2017 så har antalet robotar inom denna industri ökat för varje år [15]. Från 2017 till och med år

2024 så har även antalet industriella robotar som används totalt, alla industrier sammanslagna, ökat för varje år [50].

Studenterna ser därför inga tydliga indikationer på att det skulle gå att bromsa denna utveckling, då det finns möjligheter till ökad produktivitet och minskad risk för fel [15], [18] som sannolikt motiverar företag att använda robotar.

En litteraturstudie visar att det finns stöd för påståendet att marknaden för kollaborativa robotar ökar [Liu]. Kollaborativa robotar är robotar som är utformade för att, på ett säkert sätt, kunna arbeta nära och tillsammans med människor utan behov av säkerhetsstängsel [Kumar]. Detta skulle kunna möjliggöra nya arbetsuppgifter.

Olika studier har dragit olika slutsatser kring vilken påverkan automatisering med robotar får för arbetstillfällena. Utifrån en bibliometrisk analys argumenterar en artikel att rollen för produktionsoperatörer kommer att förändras men inte helt ersättas av robotar [Kumar]. En studie säger att förbättring av robotteknologi leder till lägre löner och arbetstillfällen samtidigt som andra uppgifter kan ge fler jobb när produktiviteten ökar [Acemoglu].

Exempel på nya jobb är programmerare och utvecklare som hjälper fabriker att använda robotar i sin produktion. Några andra exempel är underhållspersonal och personal som jobbar för robottillverkare. När produktiviteten ökar som en konsekvens av införandet av robotar kan marknaden för företagets varor också växa, vilket kan ha positiva effekter för andra parter i försörjningskedjan och på så vis skapa nya jobb.

Studenterna ser artificiell intelligens som kanske den viktigaste faktorn för hur stor påverkan robotar kommer ha på framtidens arbetsmarknad. Artificiell intelligens hjälper till att driva effektivitet och produktivitet i tillverkningsindustrin [Mahajan] och därför kommer AI troligtvis att börja användas av fler tillverkare utifrån ekonomiska incitament. En litteraturstudie har visat att avancerad artificiell intelligens kommer minska sannolikheten att människor deltar i beslutsprocesser [Liu].

Sammantaget ser studenterna examensarbetet som en del av en större trend mot automatisering och användning av artificiell intelligens, en trend som inte heller tycks gå att stoppa. Vilka konsekvenser denna trend får för arbetsmarknaden verkar inte helt klarlagd.

Därför arbetar studenterna istället för att personalen ska kunna dra så stora fördelar av förändringen som möjligt. Som nämnts i kapitel 1.5 så för arbetet med sig tydliga fördelar för personalens arbetsmiljö. För att minska risken för att personalen förlorar sina arbeten så har studenterna även identifierat alternativa arbetsuppgifter, såsom kvalitetskontroll och förflyttning av färdigpaketerade produkter med gaffeltruck.

## 6.5 Slutsatser

Studien har kartlagt hur rörelseparametrar specifikt acceleration, hastighet och accelerationsramp samverkar för att definiera en industrirobots energiprofil i en simulerad miljö. samtidigt som cykeltider registreras och jämförs.

Resultaten visar en tydligt förhållande mellan energiförbrukningen och tid. Det framgår att konfigurationer optimerade för hög produktivitet exempelvis med en hastighet på 2000 och maximerad ramp oundvikligen leder till en eskalering av energiförbrukningen. I kontrast till detta genererar inställningar med lägre intensitet hastighet 1000 och reducerad ramp betydande energibesparingar, om än på bekostnad av cykeltiden. Det betyder att om endast energiförbrukningen beaktas, kommer en noggrannhet på 50, en ramp på 50 och en hastighet på 1000 att förbruka minst energi. Tvärtom, om bara effektivitet fokuseras på, så kommer en noggrannhet på 100, en ramp på 100 och en hastighet på 2000 att resultera i den maximalt genomflöde, men också den största energiförbrukningen.

Därför visar vår projektforskning ett trade-off samband mellan energiförbrukning och produktivitet, vilket innebär att valet av rörelseparametrar bör baseras på antingen lägre energiförbrukning eller högre produktivitet. Slutligen drogs slutsatsen att ABB Rotbotstudio är ett effektivt verktyg för att analysera och optimera robotar. Det kan hjälpa till att studera hur robotik och programmering påverkar energiförbrukningen i industriell produktion. Valet av hastighet, acceleration eller accelerationsramp har stor betydelse för energiförbrukningen. Hastighet har störst inverkan på energiförbrukningen, följt av acceleration och sedan accelerationsramp. Hastighet större än acceleration större än accelerationsramp.

## 7. Källförteckning

[1] J. I. Halonen, R. Shiri, L. L. Magnusson Hanson, och T. Lallukka, "Risk and Prognostic Factors of Low Back Pain: Repeated Population-based Cohort Study in Sweden," *SPINE*, vol. 44, nr. 17, ss. 1248-1255, Sep. 2019, doi:10.1097/BRS.0000000000003052.

[2] M. L. Skovron, "Epidemiology of low back pain," *Baillière's Clinical Rheumatology*, vol. 6, nr. 3, ss. 559-573, Okt. 1992, doi:10.1016/S0950-3579(05)80127-X.

[3] "Ont i ryggen - orsaker, diagnostik och behandling," Statens Beredning för Utvärdering av medicinsk metodik, Stockholm, Sverige, SBU Utvärderar nr. 108, 1991.

[4] T. G. Sommer, S. W. Svendsen, och P. Frost, "Sickness absence and permanent work disability in relation to upper- and lower-body pain and occupational mechanical and psychosocial exposures," *Scandinavian Journal of Work, Environment & Health*, vol. 42, nr. 6, ss. 481-489, Jun. 2016, doi:10.5271/sjweh.3600.

[5] A. Koltan, "An Ergonomics Approach Model to Prevention of Occupational Musculoskeletal Injuries," *International Journal of Occupational Safety and Ergonomics*, vol. 15, nr. 1, ss. 113-124, Jan. 2009, doi:10.1080/10803548.2009.11076793.

[6] K. A. Hafez, "Occupational discomfort and injuries among automotive technicians in Jeddah, Saudi Arabia: A cross-sectional study," *WORK*, vol. 73, nr. 4, ss. 1203-1216, Dec. 2022, doi:10.3233/WOR-210522.

[7] K. Kemmlert, M. Dallner Öreljus, Å. Kilbom och F. Gamberale, "Treårsuppföljning av 195 arbetsskadeanmälningar av belastningskaraktär," *Arbetsmiljöinstitutet, Solna, Sverige, Arbeta och Hälsa* 1990:47, 1990.

- [8] ”Arbetsmarknads-situationen för befolkningen, 15-74 år, AKU 2025,” SCB, Örebro, Sverige, AMFT urn:nbn:se:scb-2026-amftbr2601\_pdf, 2026. [Online]. Tillgänglig: [https://www.scb.se/contentassets/2d9522a3556e448a59b0ccfa94acf8171/am0401\\_2025a01\\_br\\_amftbr2601.pdf](https://www.scb.se/contentassets/2d9522a3556e448a59b0ccfa94acf8171/am0401_2025a01_br_amftbr2601.pdf), Hämtad: 2026-04-16.
- [9] ”Arbetsorsakade besvär 2024,” Arbetsmiljöverket, Solna, Sverige, Arbetsmiljöstatistik Rapport 2025:03, 2025. [Online]. Tillgänglig: <https://www.av.se/globalassets/filer/statistik/arbetsorsakade-besvar-2024/arbetsmiljostatistik-arbetsorsakade-besvar-2024.pdf>, Hämtad: 2026-04-16.
- [10] S. Yang, F. Liu, J. Lu, och X. He, “Does occupational injury promote industrial robot applications?,” *Technology in Society*, vol. 70, Aug. 2022, art.nr. 101998, doi:10.1016/j.techsoc.2022.101998.
- [11] SCB, “Pris på el för övriga kunder (ej hushåll) efter förbrukarkategori. Halvår 2014H2 - 2025H2,” 2026. [Online]. Tillgänglig: [https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START\\_EN\\_EN0301\\_EN0301A/SSDHalvarElIcke/](https://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START_EN_EN0301_EN0301A/SSDHalvarElIcke/) (hämtad: 2026-04-27).
- [12] Ł. Gruszka, och M. Bartyś, “A New Energy-Efficient Approach to Planning Pick-and-Place Operations,” *Energies*, vol. 15, nr. 23, Dec. 2022, art.nr. 8795, doi:10.3390/en15238795.
- [13] G. Carabin, E. Wehrle, och R. Vidoni, “A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems,” *Robotics*, vol. 6, nr. 4, Dec. 2017, art.nr. 39, doi:10.3390/robotics6040039
- [14] M. Hovgard, B. Lennartson, K. Bengtsson, “Applied energy optimization of multi-robot systems through motion parameter tuning,” *CIRP Journal of Manufacturing Science and Technology*, vol. 35, ss. 422-430, Nov. 2021, doi:10.1016/j.cirpj.2021.07.012.
- [15] E.-Z. Wang, C.-C. Lee och Y. Li, “Assessing the impact of industrial robots on manufacturing energy intensity in 38 countries,” *Energy Economics*, vol. 105, Jan. 2022, art.nr. 105748, doi:10.1016/j.eneco.2021.105748.

- [16] S. Riazi, O. Wigström, K. Bengtsson, och B. Lennartson, "Energy and Peak Power Optimization of Time-Bounded Robot Trajectories," *IEEE Transactions on Automation Science and Engineering*, vol. 14, nr. 2, ss. 646-657, Jan. 2017, doi:10.1109/TASE.2016.2641743.
- [17] "Energiindikatorer 2025," Energimyndigheten, Eskilstuna, Sverige, Energiindikatorer ER 2025:05, 2025. [Online]. Tillgänglig: <https://energimyndigheten.a-w2m.se/Arkitektkopia/ViewTemplate?tid=68756b7ec3d8469c8e5b855646bf6b9e>, Hämtad: 2024-04-17.
- [18] Ş. Çiğdem, I. Meidute-Kavaliauskiene, och B. Yıldız, "Industry 4.0 and Industrial Robots: A Study from the Perspective of Manufacturing Company Employees," *Logistics*, vol. 7, nr. 1, Mar. 2023, art.nr. 17, doi:10.3390/logistics7010017.
- [19] R. Ruzarovsky, T. Horak, och R. Bocak, "Evaluating Energy Efficiency and Optimal Positioning of Industrial Robots in Sustainable Manufacturing," *Journal of Manufacturing and Materials Processing*, vol. 8, nr. 6, Dec. 2024, art.nr. 276, doi:10.3390/jmmp8060276.
- [20] A. Liu, H. Liu, B. Yao, W. Xu, och M. Yang, "Energy consumption modeling of industrial robot based on simulated power data and parameter identification," *Advances in Mechanical Engineering*, vol. 10, nr. 5, ss. 1-11, Maj 2018, doi:10.1177/1687814018773852.
- [21] R. Rustici Garcia, A. Carvalho Bittencourt, och E. Villani, "Relevant factors for the energy consumption of industrial robots," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 40, nr. 9, Sep. 2018, art.nr. 464, doi:10.1007/s40430-018-1376-1.
- [22] M. Hovgard, B. Lennartson, K. Bengtsson, "Simulation Based Energy Optimization of Robot Stations by Motion Parameter Tuning," i *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, Kanada, 2019, ss. 456-461. [Online]. Tillgänglig: <https://doi.org/10.1109/COASE.2019.8843152>, Hämtad: 2026-04-17.

- [23] *Robotik - Terminologi*, ISO 8373:2021, IDT, Svenska institutet för standarder, Stockholm, Sverige, 2021. Tillgänglig:  
<https://www.sis.se/produkter/terminologi-och-dokumentation/ordlistor/produktionsteknik-ordlistor/SS-ISO-83732021/>.
- [24] R.-A. Sánchez-Sosa, och E. Chavero-Navarrete, “Robotic Cell Layout Optimization Using a Genetic Algorithm,” *Applied Sciences*, vol. 14, nr. 19, Okt. 2024, art.nr. 8605, doi:10.3390/app14198605.
- [25] ABB, “Operating manual - RobotStudio,” 2025. [Online]. Tillgänglig:  
<https://search.abb.com/library/Download.aspx?DocumentID=3HAC032104-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=AZ>  
(hämtad: 2026-02-17).
- [26] T. Komenda, M. Brandstötter, och S. Schlund, “A comparison of and critical review on cycle time estimation methods for human-robot work systems”, *Procedia CIRP*, vol. 104, ss. 1119-1124, Nov. 2021, doi:10.1016/j.procir.2021.11.188.
- [27] ABB, “Product manual - Grippers for IRB 14000,” 2025. [Online]. Tillgänglig:  
<https://search.abb.com/library/Download.aspx?DocumentID=3HAC054949-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=N>  
(hämtad: 2026-05-06).
- [28] YouTube, “IRC5 and OmniCore™ Controllers - I/O Signal Setup,” 2022. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=8BxiQS-1E7Q> (hämtad: 2026-05-06).
- [29] ABB, “Product specification - Controller IRC5 with FlexPendant (DSQC1000, RW 5.6x, RW 6),” 2025. [Online]. Tillgänglig:  
<https://search.abb.com/library/Download.aspx?DocumentID=3HAC047400-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=AN>  
(hämtad: 2026-05-15).
- [30] ABB, “Technical reference manual - RAPID overview,” 2026. [Online]. Tillgänglig:  
<https://search.abb.com/library/Download.aspx?DocumentID=3HAC050947-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=X>

(hämtad: 2026-05-01).

[31] H. Unniyankal, D. Ancona, A. Ferrando, F. Parodi, A. Alessi, och F. Bottino, “Runtime Verification of Program Organization Units in Safe Programmable Logic Controller Systems,” i *2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume (DSN-S)*, Neapel, Italien, 2025, ss. 112-118. [Online]. Tillgänglig:

<https://doi.org/10.1109/DSN-S65789.2025.00050>, Hämtad: 2026-05-08.

[32] RobotStudio, 2025.5 (64-bit) version 25.5.11605.0, [Mjukvara], Västerås, Sverige: ABB Robotics, 2026.

[33] ABB, “Technical reference manual - RAPID Instructions, Functions and Data types RW 6,” 2026. [Online]. Tillgänglig:

<https://search.abb.com/library/Download.aspx?DocumentID=3HAC050917-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=AB>

(hämtad: 2026-05-01).

[34] *Lag om upphovsrätt till litterära och konstnärliga verk*, SFS 1960:729, Justitiedepartementet, Stockholm, Sverige: Regeringskansliet, Dec. 1960. [Online].

Tillgänglig:

[https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/lag-1960729-om-upphovsratt-till-litterara-och\\_sfs-1960-729/](https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/lag-1960729-om-upphovsratt-till-litterara-och_sfs-1960-729/).

[35] Lunds Tekniska Högskola, ”Programvara,” 2020. [Online]. Tillgänglig:

<https://www.student.lth.se/stoed-och-service/it-tjanster-och-support/programvara/>

(hämtad: 2026-05-06).

[36] ABB, “RobotStudio® tutorials,” 2026. [Online]. Tillgänglig:

<https://www.abb.com/global/en/areas/robotics/products/software/robotstudio-suite/robotstudio-tutorials> (hämtad: 2026-05-06).

[37] YouTube, “Att simulera conveyor belt i robotstudio,” 2025. [Online]. Tillgänglig:

<https://www.youtube.com/watch?v=DkUPqsywvKI> (hämtad: 2026-05-06).

- [38] YouTube, "RobotStudio Conveyor," 2020. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=29w309jRJEc> (hämtad: 2026-05-06).
- [39] YouTube, "How to Create Conveyor Tracking in RobotStudio," 2021. [Online]. Tillgänglig: [https://www.youtube.com/watch?v=\\_R3TEO-PiWE](https://www.youtube.com/watch?v=_R3TEO-PiWE) (hämtad: 2026-05-06).
- [40] YouTube, "RobotStudio Smart Components #10 - Plane Sensor, Volume Sensor, Position Sensor," 2021. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=yT7sx18mq6c> (hämtad: 2026-05-06).
- [41] YouTube, "Amazing ABB RobotStudio Tutorial on Gripper Modeling, Pick and Place Simulation, and Rapid Coding," 2022. [Online]. Tillgänglig: <https://www.youtube.com/watch?v=JVivVuuogHo> (hämtad: 2026-05-06).
- [42] YouTube, "smart component RobotStudio," 2018. [Online]. Tillgänglig: [https://www.youtube.com/watch?v=1vnxYdN\\_cLQ](https://www.youtube.com/watch?v=1vnxYdN_cLQ) (hämtad: 2026-05-06).
- [43] ABB, "IRB 1410, Product specification," 2023. [Online]. Tillgänglig: <https://search.abb.com/library/Download.aspx?DocumentID=3HAC026366-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=H> (hämtad: 2026-05-08).
- [44] ABB, "Product manual, IRB 4600," 2026. [Online]. Tillgänglig: <https://search.abb.com/library/Download.aspx?DocumentID=3HAC033453-001&LanguageCode=en&DocumentPartId=&Action=Launch&DocumentRevisionId=AK> (hämtad: 2026-05-01).
- [45] MDPI, "Guidelines for Reviewers," 2026. [Online]. Tillgänglig: <https://www.mdpi.com/reviewers> (hämtad: 2026-05-18).
- [46] *Förvaltningslag*, SFS 2017:900, Justitiedepartementet, Stockholm, Sverige: Regeringskansliet, Sep. 2017. [Online]. Tillgänglig: [https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/forvaltningslag-2017900\\_sfs-2017-900/](https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/forvaltningslag-2017900_sfs-2017-900/).

[47] *Kungörelse om beslutad ny regeringsform*, SFS 1974:152, Justitiedepartementet, Stockholm, Sverige: Regeringskansliet, Feb. 1974. [Online]. Tillgänglig: [https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/kungorelse-1974152-om-beslutad-ny-regeringsform\\_sfs-1974-152/](https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/kungorelse-1974152-om-beslutad-ny-regeringsform_sfs-1974-152/).

[48] P. Neuhäusler, och K. Blind, “The impact of ISO standards on the diffusion of scientific knowledge,” *Research Policy*, vol. 55, nr. 1, Jan. 2026, art.nr. 105371, doi:10.1016/j.respol.2025.105371.

[49] M. Gadaleta, M. Pellicciari, och G. Berselli, “Optimization of the energy consumption of industrial robots for automatic code generation,” *Robotics and Computer-Integrated Manufacturing*, vol. 57, ss. 452-464, Jun. 2019, doi:10.1016/j.rcim.2018.12.020.

[50] International Federation of Robotics, “Market Presentation - World Robotics 2025 Report,” 2025. [Online]. Tillgänglig: [https://ifr.org/downloads/press\\_docs/PressConference2025\\_presentation.pdf](https://ifr.org/downloads/press_docs/PressConference2025_presentation.pdf) (hämtad: 2026-05-20).

[51] International Federation of Robotics, “About IFR,” 2026. [Online]. Tillgänglig: <https://ifr.org/association> (hämtad 2026-05-20).

[Kumar] P. Kumar, S. Gupta, N. Virmani, och S. Jagtap, “Applications of Cobots in manufacturing: A bibliometric analysis,” *International Journal of Innovation Studies*, vol. 10, nr. 1, ss. 88-93, Mar. 2026, doi:10.1016/j.ijis.2025.10.002.

[Liu] L. Liu, F. Guo, Z. Zou, och V. G. Duffy, “Application, Development and Future Opportunities of Collaborative Robots (Cobots) in Manufacturing: A Literature Review,” *International Journal of Human-Computer Interaction*, vol. 40, nr. 4, ss. 915-932, Apr. 2024, doi:10.1080/10447318.2024.2301895.

[Acemoglu] D. Acemoglu, och P. Restrepo, “Robots and Jobs: Evidence from US Labor Markets,” *Journal of Political Economy*, vol. 128, nr. 6, ss. 2188-2244, Jun. 2020, doi:10.1086/705716.

[Mahajan] A. Mahajan, G. Singh, S. Devgan, G. Singh, och P. Kaur, "Next-generation manufacturing: leveraging AI for industrial innovation and growth," *Computers & Industrial Engineering*, vol. 211, Jan. 2026, art.nr. 111618, doi:10.1016/j.cie.2025.111618.

[Förordning] *Förordning med instruktion för Regeringskansliet*, SFS 1996:1515, Statsrådsberedningen, Stockholm, Sverige: Regeringskansliet, Dec. 1996. [Online].

Tillgänglig:

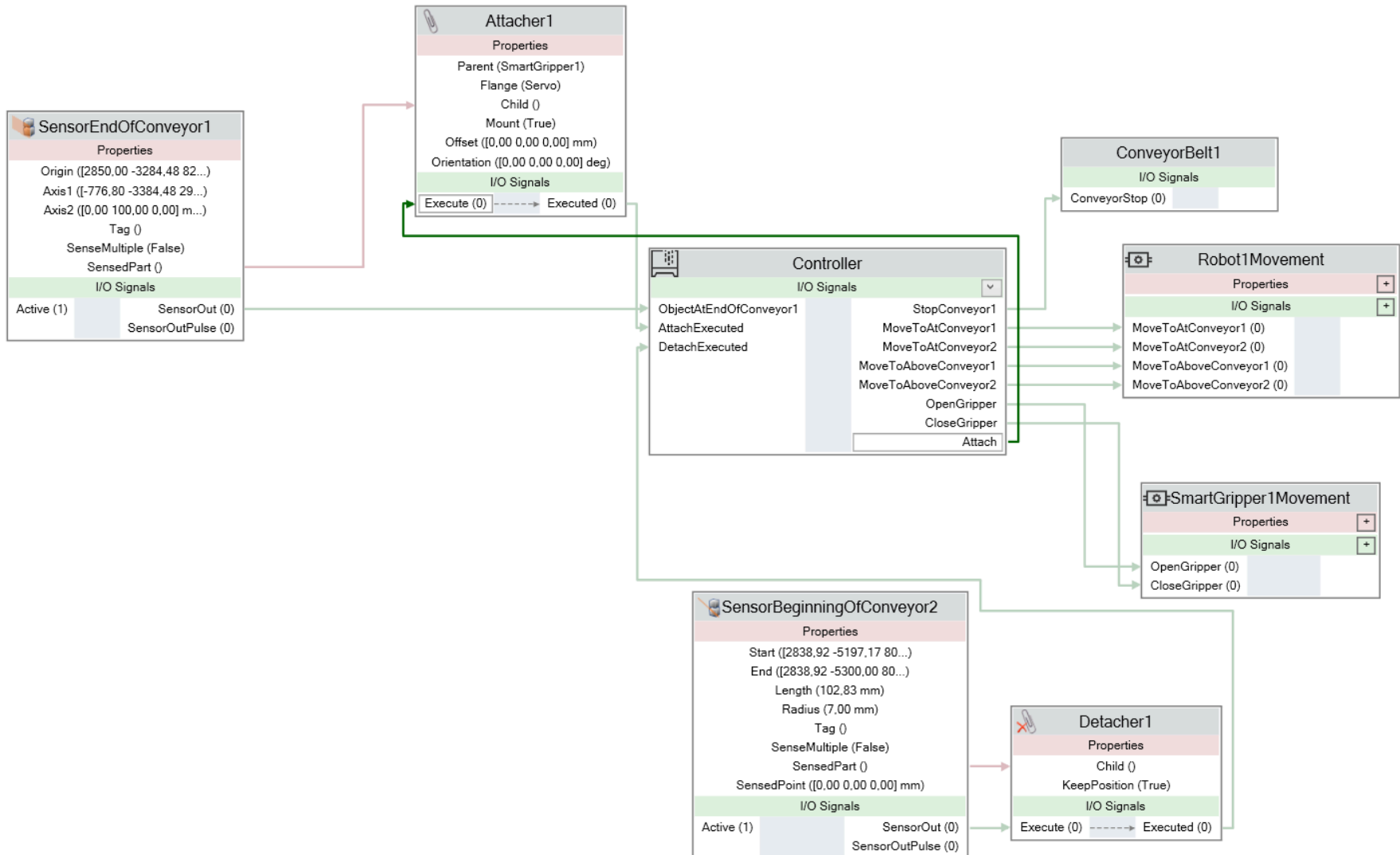
[https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/forordning-19961515-med-instruktion-for\\_sfs-1996-1515/](https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/forordning-19961515-med-instruktion-for_sfs-1996-1515/)

[Lag] *Lag om kungörande av lagar och andra författningar*, SFS 1976:633,

Justitiedepartementet, Stockholm, Sverige: Regeringskansliet, Jun. 1976. [Online].

Tillgänglig:

[https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/lag-1976633-om-kungorande-av-lagar-och-andra\\_sfs-1976-633/](https://www.riksdagen.se/sv/dokument-och-lagar/dokument/svensk-forfattningssamling/lag-1976633-om-kungorande-av-lagar-och-andra_sfs-1976-633/)



## B. RAPID-kod

```
MODULE Module1
!*****
! Targets translated to robtargets by synchronizing
! A path with the desired targets to RAPID
!*****

CONST robtarget AtConveyor1:=

    [ [-0.575629841,1026.000068972,827.001256684],
      [0,-0.000000298,1,0.000000403], [1,-1,1,0],
      [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09] ];

CONST robtarget AtConveyor2:=

    [ [-186.575446499,-946.000092282,860.000196466],
      [0.000000089,-0.00000019,1,-0.000000049], [-2,0,-2,0],
      [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09] ];

CONST robtarget AboveConveyor2:=

    [ [-186.57536732,-946.000320078,1020.001088093],
      [0.000000104,0.000000085,1,0.000000025], [-2,0,-2,0],
      [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09] ];

CONST robtarget HighAboveConveyor1:=

    [ [-0.575202829,1026,1020],
      [0,0,1,0], [1,-1,1,0],
      [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09] ];

CONST robtarget JustAboveConveyor1:=

    [ [-0.575629695,1026.000033959,832.00132135],
      [0,-0.000000298,1,0.000000373], [1,-1,1,0],
      [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09] ];
```

```
!*****
! tooldata used for move instructions that are performed with a
! product attached to the vacuum tool. The vacuum tool from the
! demo station "SC Infeeder finished" has a weight of 9 kg.
! Another 40 kg corresponding to the product weight is added.
!*****
```

```
PERS tooldata vacuumToolWithLoad:=[TRUE, [[0,0,150],[1,0,0,0]
      [49,[0,0,100],[1,0,0,0],0,0,0]]];
```

```
! Timer used to measure cycle times
VAR clock cycleTimer;
VAR num cycleTime;
```

```
! Cycle counter
VAR num cycle := 1;
```

```
! Timer used to measure the execution time
VAR clock executionTimer;
VAR num totalTime;
```

```
VAR clock temp;
```

```
! Changes speeddata globally for the program
```

```
VAR speeddata velocity := v1000;
```

```
PROC main()
```

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

```
! Resets output signals after previous runs to ensure
! a consistent initial state
```

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

```
SetDO doAttach, 0;
SetDO doDetach, 0;
```

```
! Sets percent of normal acc. and acc. ramp
AccSet 100, 100;
```

```
ClkStart executionTimer;
```

```

WHILE TRUE DO
    IF diObjectAtEndOfConveyor1 = 1 THEN
        MoveProductFromConveyor1To2;
    ENDIF
ENDWHILE
ENDPROC

PROC MoveProductFromConveyor1To2()
    ClkStart cycleTimer;

    ! Sets doAttach to signal to the Smart Component Attacher
    WaitTime 0.15;
    SetDO doAttach, 1;
    WaitTime 0.15;
    SetDO doAttach, 0;
    WaitTime 0.15;

    ! Moves product to conveyor belt 2
    MoveJ HighAboveConveyor1,velocity,fine,vacuumToolWithLoad;
    MoveJ AboveConveyor2,velocity,fine,vacuumToolWithLoad;
    MoveJ AtConveyor2,velocity,fine,vacuumToolWithLoad;

    ! Sets doDetach to signal to the Smart Component Detacher
    WaitTime 0.15;
    SetDO doDetach, 1;
    WaitTime 0.15;
    SetDO doDetach, 0;
    WaitTime 0.15;

    ! Moves robot back to conveyor belt 1
    MoveJ AboveConveyor2,velocity,fine,tVacuum;
    MoveJ HighAboveConveyor1,velocity,fine,tVacuum;
    MoveJ JustAboveConveyor1,velocity,fine,tVacuum;

```

```
ClkStop cycleTimer;
cycleTime := ClkRead(cycleTimer);
TPWrite "Execution time for cycle " + NumToStr(cycle, 4) +
": " + NumToStr(cycleTime, 4);
```

```
IF cycle = 5 THEN
    TPWrite "Total time until 5 cycles have been" +
        "Performed: " + NumToStr(ClkRead(executionTimer), 4);
    Stop;
ENDIF
```

```
cycle := cycle + 1;
ClkReset cycleTimer;
```

```
ENDPROC
```

```
ENDMODULE
```





