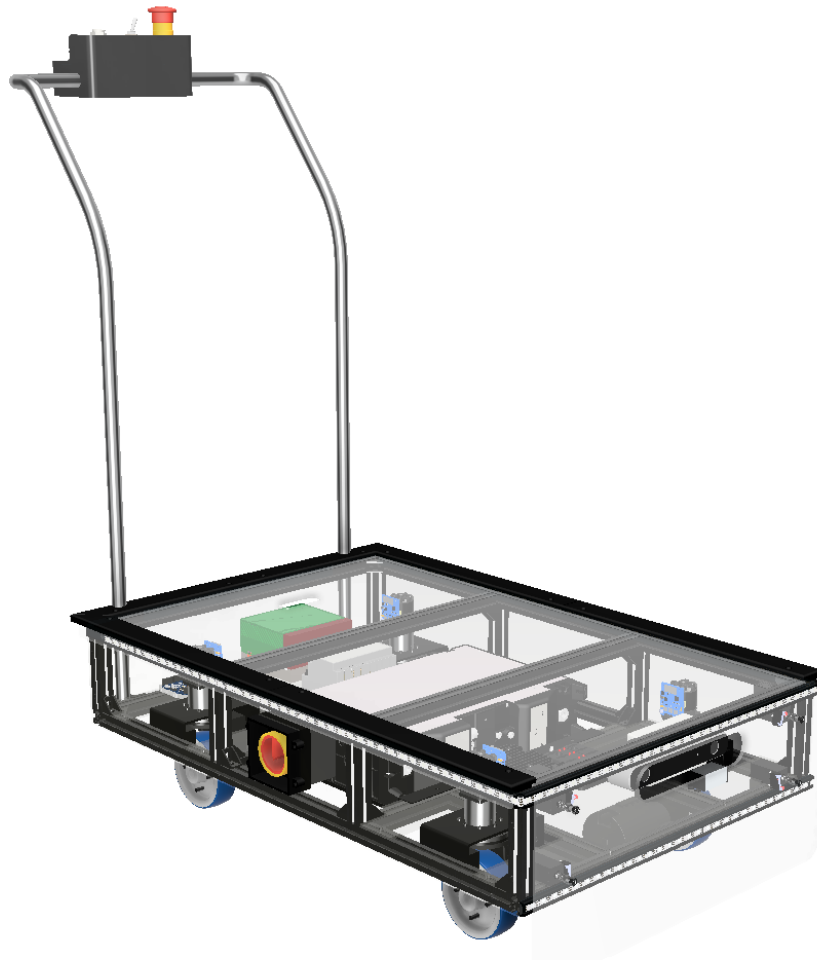




CHALMERS



Design och konstruktion av en 4WS och 4WD mobil robot

Vilgot Alexandersson, Elias Grönlund, Filip Johansson,
Hüseyin Karacan, Oskar Persson, Vincent Wallsten

INSTITUTIONEN FÖR ELEKTROTEKNIK

CHALMERS TEKNISKA HÖGSKOLA

Göteborg 2025

www.chalmers.se

KANDIDATARBETE 2025

**Design och konstruktion av en
4WS och 4WD mobil robot**

Vilgot Alexandersson

Elias Grönlund

Filip Johansson

Hüseyin Karacan

Oskar Persson

Vincent Wallsten



CHALMERS

Institutionen För Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg 2025

Design och konstruktion av en 4WS och 4WD mobil robot

Vilgot Alexandersson

Elias Grönlund

Filip Johansson

Hüseyin Karacan

Oskar Persson

Vincent Wallsten

© Vilgot Alexandersson, Elias Grönlund, Filip Johansson, Hüseyin Karacan, Oskar Persson, Vincent Wallsten, 2025.

Handledare: Knut Åkesson, Institutionen för elektroteknik

Examinator: Martin Fabian, Institutionen för elektroteknik

Kandidatarbete 2025

Institutionen för Elektroteknik

Chalmers Tekniska Högskola

SE-412 96 Göteborg

Telefon +46 31 772 1000

Omslagsbild: CAD-modell av roboten som utvecklades under arbetet

Skriven i L^AT_EX

Göteborg 2025

Sammanfattning

Detta är ett projekt som har itererats över flera år som kandidatarbete på Chalmers tekniska högskola. Projektet syftade till att vidareutveckla den tidigare robotplattformen, som varit instabil och enbart kunde styras via inbyggda styrspakar, till en mekaniskt stabil och banföljande robot. Under projektet har både hårdvara och mjukvara prioriterats för att få en bättre presterande robot. Roboten blev inte helt banföljande men fick istället fler körögen och möjligheten att köra den med en trådlös spelkontroll. Tidigare kunde roboten inte köras med ett batteri som strömkälla utan var tvungen att köras med ett nätaggregat, vilket löstes under denna iteration av projektet.

För att göra roboten mer stabil låg stort fokus på att byta ut robotens hjulmodul till ett mer hållfast alternativ. Ett av målen var även att få roboten att navigera utan att köra in i hinder. Detta gjordes med hjälp av en kamera som skulle känna igen objekt och räkna ut avståndet till dem. Detta påbörjades men på grund av andra prioriteringar i projektets slutskede kunde inte detta implementeras parallellt med styrningen av roboten. Ett intentionssystem implementerades på roboten för att kommunicera dess planerade rörelser till omgivningen. Exempel på detta är animationer för acceleration/retardation, bromsljus och blinkers.

Abstract

This is a project that has been iterated over several years as a bachelor's thesis at Chalmers University of Technology. The goal of the project was to develop the previous model from being an unstable robot that could only be operated using its onboard controls into a stable autonomous robot. During the project, both hardware and software were prioritized to create a robust autonomous robot. Unfortunately, the robot did not become fully autonomous, but instead gained additional driving modes and the ability to be operated with a wireless game controller. Previously, the robot could not be powered by a battery and had to run off a power supply; this problem was solved during this iteration of the project.

To make the robot more stable, a major focus was placed on replacing its wheel module. This allowed it to drive in a simpler and more predictable manner than before. One of the goals was also to enable it to navigate without colliding with obstacles. This was to be achieved using a camera that would recognize objects and calculate their distance. This effort was initiated but, due to long lead times on other components, could not be implemented in parallel with the robot's drive control. An intent system was implemented on the robot to communicate its planned movements to the surrounding environment. Examples of this include animations for accelerating/deceleration, brake lights, and turn signals for right and left.

Förord

Vi i gruppen vill tacka vår handledare, Knut Åkesson samt Kristian Ceder som var med under projektets gång. De har varit mycket hjälpsamma och engagerade i projektet och har till stor del bidragit till de resultat vi kan presentera i denna rapport. Vi vill också tacka Kristoffer Bengtsson på AB Volvo som har hjälpt oss med diverse frågor och bidragit med tips och komponenter till vårt projekt. Rikard Karlsson från CASE labbet var också till stor hjälp under projektet där han hjälpte till med beställningar och andra problem som uppstod under projektet. Utan dessa personer hade inte projektet uppnått detta resultat.

Akronymer

Nedan är en lista över akronymer som har använts i rapporten, listade i alfabetisk ordning:

AI	Artificiell intelligens
AMR	Autonomous mobile robot
BMS	Battery management system
CAD	Computer-Aided Design
CAN	Controller Area Network
FEM	Finita Elementmetoden
RGB	Röd-Grön-Blå
RPM	Revolutions Per Minute
SLAM	Simultaneous localization and mapping
VO	Visuell odometri

Innehåll

1	Introduktion	1
1.1	Bakgrund	1
1.2	Syfte	3
1.3	Problemställning	3
1.4	Avgränsningar	4
1.5	Samhälleliga och etiska aspekter	4
2	Datorseende	6
2.1	Val av kamera för objektigenkänning	6
2.2	Dator för objektigenkänning	6
2.3	Modellval för objektigenkänning	6
3	Styrgeometri	9
3.1	Ackermannstyrning	9
3.2	Krabbstyrning och rotation kring egen axel	10
4	Banföljning	12
4.1	Odometri	12
4.2	Hjulbaserad odometri	13
4.2.1	Visuell odometri	14
4.3	SLAM	15
4.4	Planering av vägpunktbanor	16
4.4.1	Pure Pursuit	19
5	Mekanisk konstruktion	20
5.1	Hjulmodul	20
5.1.1	Tidigare hjulmodul	21
5.1.2	Överdel hjulmodul	21
5.1.3	Svänlager	22
5.1.4	Nederdel hjulmodul	23
5.1.5	Navmotorhjul	24

5.1.6	Styrstopp	25
5.1.7	Hjuldistan	25
5.1.8	Framtaget koncept	26
5.1.9	Slutgiltig lösning	27
5.1.10	FEM-analys av hjulmodul	28
5.2	Plexiglas	33
5.3	Fästen till komponenter	35
6	Elsystem	37
6.1	Kommunikationssystem	38
6.2	Design av elsystem	39
6.3	Implementering av batteri	40
6.4	Intentionssystem	41
6.4.1	Implementering av tidigare utvecklade animationer	43
6.5	Komponentplacering	43
7	Datorprogram för styrning av robot	46
7.1	Lågnivåstyrning	47
7.2	Högnivåstyrning	49
7.3	Sändning av styrsignaler	50
8	Testning	52
8.1	Simulering av den kinematiska modellen	52
8.2	Kvalitativ Verifiering av Systemfunktionalitet	53
8.3	Tester för styrning	53
8.4	Kvantitativt test av odometri och precision	55
9	Resultat	57
9.1	Simulering av grundläggande banföljning	58
9.2	Tester	59
9.3	Implementation av objektigenkänning	60
9.4	Intentionssystem	60
9.5	Mekanisk konstruktion	60
10	Diskussion	61
10.1	Hjulmodul	61
10.2	Elsystem	62
10.3	Datorprogram för styrning av robot	63
10.4	Objektigenkänning	64

10.5	Styrgeometri	64
10.6	Vidareutveckling	65
Bilagor		i
A.1	Ritning – Överdel hjulmodul	ii
A.2	Ritning – Underdel hjulmodul (3D-utskriften)	iii
A.3	Ritning – Distans konisk	iv
A.4	Ritning – Distans lager, innerring	v
A.5	Ritning – Distans lager, yttering	vi
A.6	Ritning – Plexiglas, underdel fram	vii
A.7	Ritning – Plexiglas, underdel bak	viii
A.8	Ritning – Fäste för ZED-kamera	ix
A.9	Ritning – Övre del av konstruktion	x
A.10	Ritning – Stötfångare	xi

1 Introduktion

Projektet är en vidareutveckling av tidigare kandidatarbeten och syftar till att skapa en autonom robot med både fyrhjulsdrift och fyrhjulsstyrning för intern logistik. Arbetet fokuserar på att förbättra manövrerbarhet och säkerhet genom användning av en stereokamera. Behovet av en fyrhjulsstyrd och fyrhjulsdriven robot har uttryckts av AB Volvo, som efterfrågar en lättmanövrerad lösning för intern logistik.

1.1 Bakgrund

Projektet är i samarbete med AB Volvo. Chalmers har fått i uppgift att hjälpa dem med automatisering av transporter och enkla arbetsuppgifter. Detta sker eftersom produktionen av elektriska bilar och lastbilar ökar, vilket ökar antalet komponenter som behövs för tillverkningen. AB Volvo vill att autonoma robotar ska transportera komponenter från lagret till monteringslinan precis när de behövs för att effektivisera produktionen och minska sina kostnader på lång sikt.

Kandidatarbeten rörande att skapa en autonom mobil robot har genomförts tidigare på Chalmers tekniska högskola. År 2021 genomfördes ett liknande projekt i samarbete med AB Volvo (Dahl m. fl., 2021). Den tidigare projektgruppen utvecklade en robotmodell med sex hjul, där de två mellersta var drivande och monterade på en fjädrande svängarm. De övriga fyra hjulen var fria länkhjul placerade i robotens hörn. År 2024 genomfördes också en iteration av projektet (Billing m. fl., 2024). Denna projektgrupp valde att designa om drivlinan med endast fyra hjul där alla fyra var drivande och hade var sin dedikerad svängmotor. Detta gjorde det möjligt att utöver differentialstyrning och Ackermannstyrning även implementera fyrhjulsstyrning hos roboten. Att styra med alla fyra hjul ger fler frihetsgrader och förbättrar robotens manövreringsförmåga jämfört med en traditionell Ackermannstyrning med framhjulen. Detta möjliggör snabbare och mer precis riktningssändring, vilket gör att roboten kan navigera smidigare i trånga och komplexa miljöer.

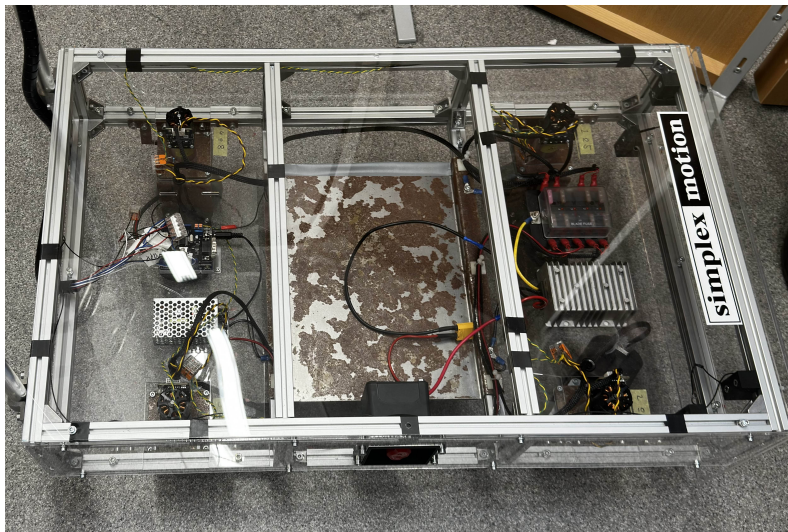
På AB Volvos tillverkningsanläggning i Tuve utanför Göteborg finns sedan 2024 en småskalig AMR-implementation. Dessa robotar liknar modellen från 2021 års kandidatarbete, med två centralt monterade drivande hjul. Robotarna navigerar i fabriken med hjälp av kameror monterade i taket som ser positionerna på robotar, människor och andra objekt.

Implementationen är en del av ett utvecklingsprojekt som AB Volvo driver. Trots en relativt lyckad implementation finns dock förbättringspotential.

För att förbättra robotens säkerhet vid navigation finns önskemål om att en stereokamera monteras på roboten. Fördelen med en stereokamera är att den har förmåga att se djup i bilden. Med denna teknik kan roboten exempelvis stanna om den kommer för nära en människa eller ett objekt. Detta gör systemet mer självständigt och mindre känsligt för fel med takkamerorna. Genom maskininlärning kan kameran identifiera vilken typ av objekt som befinner sig framför roboten (Zebra, 2025). Ett önskemål är också att implementera en radar på roboten. Genom att använda radiovågor kan radarn mycket exakt bestämma avstånd till objekt och beräkna deras relativa hastighet. En annan fördel med radar är dess robusthet i miljöer med mycket störningar, såsom damm, regn eller mörker, där kamerabaserade system kan ha svårigheter att ge tillförlitliga data. Detta gör radar särskilt användbar i en fabriksmiljö med höga nivåer av partiklar i luften (Searcy, 2020). Genom att kombinera dessa två sensorsystem förbättras robotens navigering och säkerhet, vilket möjliggör en säkrare drift och en högre grad av kollisionssundvikande.

Objektigenkänning utgör en central funktion inom detta sammanhang och har ett brett användningsområde. För den aktuella applikationen används objektigenkänning för att analysera robotens omgivning och identifiera vilken typ av objekt som befinner sig framför den. Detta möjliggör ett situationsanpassat beteende beroende på om det rör sig om exempelvis en människa, en lastpall eller ett annat hinder. Tekniken tillåter även instruktioner att tolkas visuellt, såsom tillfälliga skyltar eller lastplatser.

Årets projekt kommer att bygga vidare på tidigare års robot (se Figur 1.1.1) och kommer att fokusera på att förbättra prestandan på både den fysiska modellen och styrsystemet.



Figur 1.1.1: Tidigare års robot

1.2 Syfte

Syftet med detta arbete är att utveckla en autonom robot med fyrhjulsstyrning (Four wheel steering/4WS) och fyrhjulsdrivning (Four wheel drive/4WD) för intern logistik i samarbete med AB Volvo. Målet är att optimera materialflödet och frigöra mänskliga resurser för mer komplexa arbetsuppgifter. Arbetet fokuserar på att undersöka hur 4WS kan förbättra robotens rörlighet och navigationsförmåga i komplexa industrimiljöer. Genom att utveckla en fungerande modell och analysera dess egenskaper syftar projektet till att öka förståelsen för autonoma robotars tillämpning inom industriell automation, med särskilt fokus på AB Volvos behov och användningsområden.

1.3 Problemställning

Detta projekt är en vidareutveckling av tidigare år samt använder AB Volvos nuvarande lösning som utgångspunkt. Problemet som skulle lösas var en robot med fyra drivhjul för att förbättra manövreringsförmågan i trånga situationer. Att implementera ett säkerhets-system med radar samt en kamera var också ett viktigt delmål.

1.4 Avgränsningar

Projektet genomförs hos institutionen för Elektroteknik på Chalmers tekniska högskola och håller sig inom tidsramen 2025-01-20 till 2025-05-27. De tekniska avgränsningarna omfattar institutionens laboratorier, där alla bygg- och monteringsprocesser kommer att utföras.

De mekaniska och elektriska delarna av roboten är ett centralt område i projektet. Roboten byggs inte från grunden utan vidareutvecklas baserat på tidigare års konstruktion. Projektet fokuserar på att förbättra och optimera den befintliga roboten genom att bygga på det arbete som har genomförts av tidigare grupper. De resursmässiga avgränsningarna är en budget på 5000 SEK som institutionen har fastställt, och tillgängliga material samt maskiner som finns i institutionens laboratorier. Budgeten kan överskridas vid överenskommelse med institutionen. I laboratorierna finns bearbetningsmaskiner som kan bearbeta de flesta metaller och trä.

1.5 Samhälleliga och etiska aspekter

Att konstruera och utveckla en AMR har flera samhälleliga och etiska aspekter. Automatisering av robotar inom industrin kan leda till att manuella jobb försvinner, vilket resulterar i arbetslöshet för de individer som är anställda för dessa arbetsuppgifter. Samtidigt skapas nya arbetsområden relaterade till hantering, programmering och underhåll av robotar. Dock kommer antalet arbetare som krävs för dessa nya områden sannolikt att vara färre än de arbetsplatser som robotarna ersätter. Arbetsfördelningen och arbetskraften kan komma att ställas om och anställda kommer då att behöva nya kompetenser samt vidareutbildning för att kunna arbeta tillsammans med robotarna.

Ur företagets perspektiv är automatisering med robotar en långsiktig kostnadseffektiv lösning, med lägre produktionskostnader och högre effektivitet. Robotarna och deras kamerabaserade styrsystem kan bidra till ökad effektivitet, men samtidigt väcka frågor rörande ständig övervakning. Bildhantering kopplat till personlig integritet kan bli ett stort problem. På AB Volvos testanläggning har man löst detta med tidsbegränsad lagring som raderar bildmaterialet så fort det använts för styrningen av robotarna. Vid större implementering är det dock oklart om detta är en tillräcklig åtgärd.

En annan konsekvens som också bör tas i beaktning är de digitala klyftorna som kan komma att uppstå mellan företagen. Stora företag har resurserna som krävs för att implementera ny teknik, medan mindre företag ofta saknar både budget och möjlighet att göra det. Detta kan göra det svårt för mindre företag att konkurrera och riskerar således att hamna på efterkälken.

Den nya teknologin med autonoma robotar skapar även frågor rörande risker och olyckor som kan uppstå vid samarbete mellan robotar och människor. Vem som bär ansvaret när en robot skadar en medarbetare är oklart. Därför är det viktigt att de autonoma robotarna är utformade för att minimera olyckor och säkerställa en trygg interaktion med människor. Dock påverkas ergonomin och de fysiska påfrestningarna på de mänskliga medarbetarna i allmänhet positivt av dessa implementeringar, eftersom de underlättar hantering av tunga objekt och minskar behovet av repetitiva rörelser.

2 Datorseende

Med hjälp av en stereokamera samt en dator implementerades objektigenkänning på roboten. Med hjälp av denna teknik skapades möjligheten att kunna använda kameran som ett säkerhetssystem. Till detta tillägg hör även möjligheten att styra robotens beteende baserat på vilka objekt som identifieras i kamerabilden.

2.1 Val av kamera för objektigenkänning

Under projektet användes en ZED-kamera (första generationen) från StereoLabs. Tanken med denna, tillsammans med objektigenkänning, var att det skulle bidra till ett säkerhetssystem. Syftet med implementeringen av kameran var främst att kunna möjliggöra olika beteenden på roboten när den är i drift. Exempel på detta kan vara om en människa står i vägen för roboten, minskar den hastigheten. Om ett statiskt objekt skulle vara i vägen för roboten, ska den stanna helt.

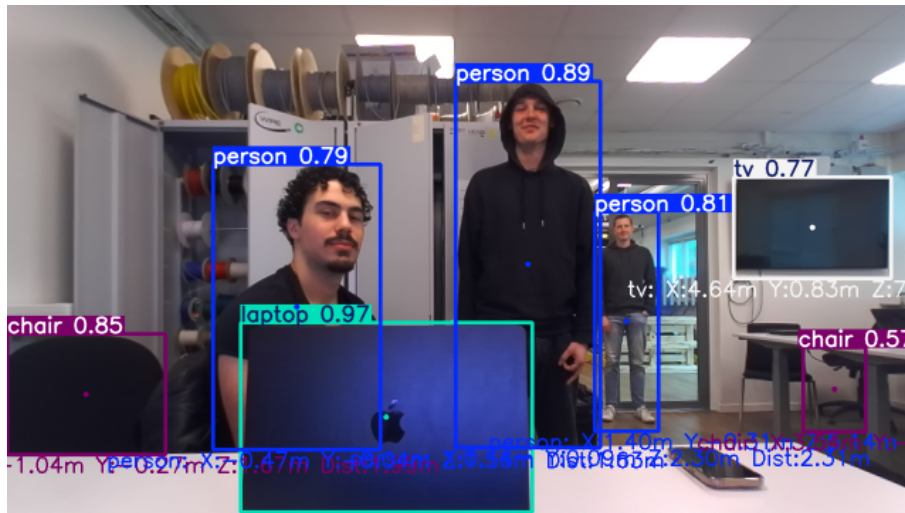
2.2 Dator för objektigenkänning

Ursprungligen satt bara en Arduino UNO R4 WIFI (Arduino, 2025) på roboten, men eftersom den saknade tillräcklig kapacitet för bildbehandling, användes en Nvidia Jetson AGX Orin (Nvidia, 2025). Denna specialiserar sig på objektigenkänning och är väl lämpad för ändamålet. Under projektets gång uppstod däremot återkommande problem med att vissa program och bibliotek inte var fullt optimerade för plattformen. Trots dessa utmaningar utvecklades till slut en Python-kod som bearbetade bilderna från kameran med hjälp av Jetson-datorn.

2.3 Modellval för objektigenkänning

För att få fram en välfungerande modell som kan känna igen objekt behövs mycket träning för en AI. För att kringgå detta problem använde sig gruppen av en redan färdigtränad modell, YOLO (You Only Look Once) och med hjälp av detta kunde olika objekt identi-

feras och klassificeras (se Figur 2.3.1). I projektet användes specifikt versionen YOLOv8 som utvecklats av Ultralytics, och finns tillgängligt med öppen källkod under en icke-kommersiell licens. YOLO är en bildklassificerare som delar upp objekt i olika klasser (V7 Labs, 2023). Detta användes då gruppen kunde nyttja befintlig kod via GitHub som var optimerad för Jetson-datorer som skulle köra detta paket (se del 2.2).



Figur 2.3.1: Objektigenkänning med YOLOv8

Från föregående år fanns redan en Arduino monterad på roboten för styrning. Denna hade däremot inte tillräcklig kapacitet för att hantera bildbehandling, genom att istället använda Jetson-datorn kunde detta möjliggöras. Jetson-datorn är särskilt anpassad för AI-tillämpningar, vilket möjliggjorde effektiv användning av ZED SDK. ZED SDK (Software Development Kit) är ett utvecklingspaket från StereoLabs som erbjuder verktyg för bland annat djupmätning, 3D-visualisering och objektsparning med ZED-kameror. Kombinationen av datorn, kameran och kamerapaketet resulterade i en avsevärt snabbare och mer kapabel lösning jämfört med vad som varit möjligt med exempelvis en Arduino.

ZED-kameran som används är en stereokamera. Stereokameror uppskattar djup genom att mäta förskjutningen mellan två bilder av samma scen, tagna från de två separata linserna (StereoLabs, 2025). Till skillnad från nyare modeller i ZED-serien har ZED (första generationen) ingen inbyggd funktion för objektigenkänning. Därför användes externa Python-bibliotek för detta ändamål. Python valdes som programmeringsspråk då det var det språk som behärskades bäst.

Då ZED-kameran inte hade någon inbyggd objektigenkänning så behövde kod skrivas för att identifiera objekten. Efter sökningar på internet kunde kod från GitHub användas

för just detta. Problemet med denna kod var främst att den använde sig av biblioteket OpenCV som skulle visa vad kameran såg i realtid. Då detta inte fungerade, hittades en annan lösning på detta, vilket var att använda sig av terminalen. Objekttypen skrevs ut i terminalen, var i rummet de befann sig samt den totala distansen till objektet från kameran.

Eftersom ZED-kameran består av två separata linser (en till vänster och en till höger) där koordinatsystemet ursprungligen var centrerat kring den vänstra linsen. För att bättre integrera med övrig positionsdata och möjliggöra beräkning av robotens position i rummet, flyttades origo till robotens mittpunkt. Målet blev därmed att kombinera positionsfunktionen med kamerans objektigenkänning. Positionsberäkningen användes för att möjliggöra visuell odometri som senare skulle kombineras med hjulens positionering.

3 Styrgeometri

Roboten har individuellt styrbara hjul, vilket möjliggör flera olika styrmodeller beroende på körsituation och behov. Beteckningar som inte återfinns i figurerna nedan återfinns i figur 4.2.1a.

3.1 Ackermannstyrning

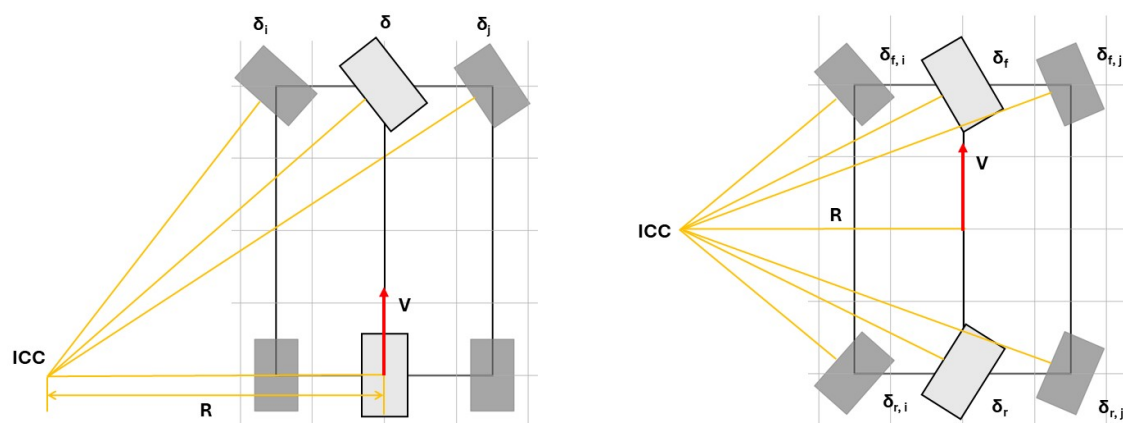
Ackermannstyrning implementeras vid svängar för att minimera slirning så mycket som möjligt. För att uppnå detta styrs de inre och yttre hjulen med olika svängradier och roterar med olika hastigheter (VivekB, 2024), se Figur 3.1.1.

I 2WS-läget svänger roboten endast med framhjulen, se Figur 3.1.1a. Det momentana rotationscentrumet (hädanefter ICC från engelska instantaneous center of curvature) hamnar då i linje med robotens bakre axel, vilket ger följande ekvationer för styrvinklar och rotationshastigheter:

$$\delta_{fi} = \arctan\left(\frac{L \cdot \tan(\delta)}{L - 0.5 \cdot W \cdot \tan(\delta)}\right), \quad \delta_{fj} = \arctan\left(\frac{L \cdot \tan(\delta)}{L + 0.5 \cdot W \cdot \tan(\delta)}\right) \quad (3.1.1)$$

$$R = \frac{L}{\tan(\delta)}, \quad \omega_{fi} = \omega_{ri} = \frac{V \cdot \left(2 - \frac{w}{R}\right)}{2 \cdot r}, \quad \omega_{fj} = \omega_{rj} = \frac{V \cdot \left(2 + \frac{w}{R}\right)}{2 \cdot r} \quad (3.1.2)$$

I 4WS-läget styr roboten även med bakhjulen, se Figur 3.1.1b. Ekvationerna ovan förblir nästan identiska, men eftersom ICC förflyttas från den bakre axeln till mittpunkten mellan fram- och bakaxeln, ersätts L i ekvationerna ovan med $L = 0.5L$. Vidare gäller för bakre styrvinklar att $\delta_{fi} = -\delta_{ri}$ och $\delta_{fj} = -\delta_{rj}$ (MATLAB, 2024).



(a) Ackermann styrning med endast framhjul. Detta ger upphov till mindre skarpa svängar. (b) Ackermann styrning med samtliga hjul. Detta ger upphov till skarpare svängar.

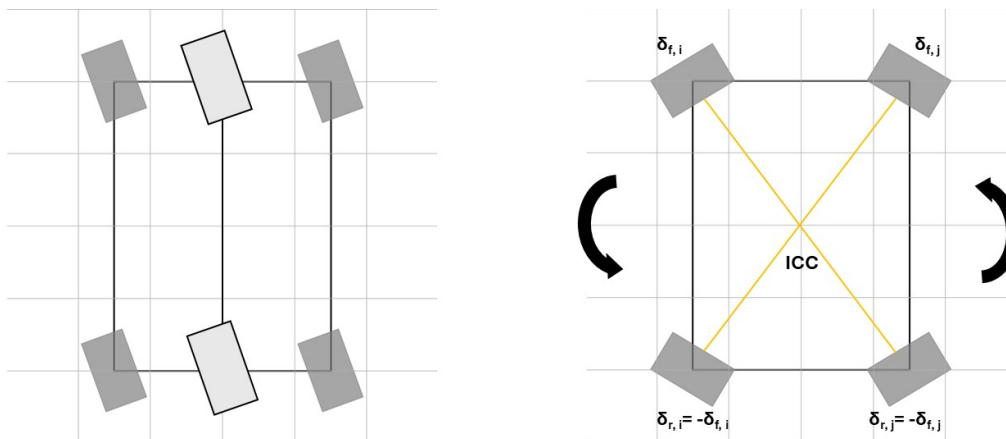
Figur 3.1.1

3.2 Krabbstyrning och rotation kring egen axel

Rotation kring egen axel och krabbstyrning är de enklaste styrmoderna för roboten. Båda illustreras i Figur 3.2.1. När roboten krabbstyr i Figur 3.2.1a styrs alla hjul i samma riktning och roterar med samma hastighet, dvs $\delta = \delta_{f_i} = \delta_{f_j} = \delta_{r_i} = \delta_{r_j}$. Detta innebär att robotens kroppsvinkel hålls konstant under körning, ingen rotation sker.

I Figur 3.2.1b roterar roboten kring sin egen mittpunkt. Alla hjul roterar då med samma hastighet, där styrvinklarna för fram- och bakhjulen beräknas enligt följande:

$$\delta_{f_i} = \arctan\left(\frac{W}{L}\right) - \frac{\pi}{2}, \quad \delta_{f_j} = \frac{\pi}{2} - \arctan\left(\frac{W}{L}\right), \quad \delta_{r_i} = -\delta_{f_i}, \quad \delta_{r_j} = -\delta_{f_j} \quad (3.2.1)$$



(a) Roboten krabbstyr genom att alla hjul på roboten har samma styrvinkel och roterar med samma hastighet.

(b) Roboten roterar kring sin egen axel genom att alla hjul är vinkelräta mot dess centrum och roterar med samma hastighet.

Figur 3.2.1

4 Banföljning

I projektet undersöktes hur robotens position kunde bestämmas med hjälp av både hjulodometri och sensorinformation från kameran. Dessa tekniker låg till grund för arbetet med att få roboten att röra sig längs förutbestämda banor, där metoder som SLAM (Simultaneous Localization and Mapping), Waypoint Mapping och Pure Pursuit studerades.

4.1 Odometri

För att skatta en robots position kan man använda sig av olika odometriska metoder. Dessa baseras på data från olika sensorer, vilket möjliggör en uppskattning av hur roboten har rört sig mellan två tidpunkter, $t_0 \rightarrow t_1$. Metoden skiljer sig från användningen av GPS eller annan teknik, eftersom referenspunkten alltid utgår från roboten. Dessutom är möjligheten att korrigera för fel som byggs upp över längre tid mer begränsad utan en fast referenspunkt (BLADE, 2024).

Vanliga odometriska metoder inkluderar:

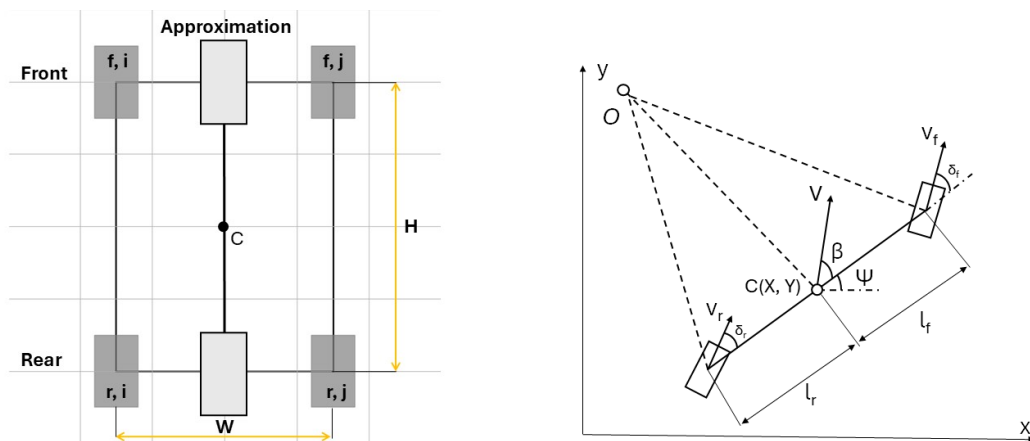
- **Hjulsensorer:** Mäter rotationen hos varje enskilt hjul eller axel för att uppskatta robotens rörelse.
- **IMU (Inertial Measurement Unit):** Använder accelerometrar och gyroskop för att skatta robotens rörelse. Hastighet och sträcka kan beräknas genom integration.
- **Optiska sensorer:** Använder optisk data för att uppskatta rörelser över ytor.
- **Visuell odometri:** Liknar optisk odometri men använder kameror för att skapa en bild av rummet och skattar rörelser genom att analysera miljödrag.

Det är vanligt att kombinera flera av dessa metoder för att utnyttja deras respektive styrkor och kompensera för svagheter. Särskilt fokus kommer att läggas på odometriska metoder med hjulsensorer samt visuell odometri i denna rapport.

4.2 Hjulbaserad odometri

Som tidigare nämnt bygger hjulbaserad odometri på mätning av hur varje enskilt hjul på roboten har roterat, för att kunna uppskatta dess rörelse över tid. De sensorer som möjliggör denna mätning kallas rotationsgivare och finns i flera olika varianter. Exempel på sådana är optiska, mekaniska, magnetiska och elektromagnetiska givare (Ahmed, 2023).

För att robotens position ska kunna uppskattas med information insamlad från motorernas sensorer krävs först en kinematisk modell för att genomföra själva uppskattningen. Eftersom projektet fokuserar på fyrhjulsstyrning kommer en förenklad 2WS cykelmodell att användas, likt den modell som ofta används för tvåhjulsstyrda bilar. I Figur 5.3.3 beskrivs hela modellen. I Figur 4.2.1a visas en bild av hela roboten sedd ovanifrån. Vinkeln på de två fram- och bakhjulen approximeras som ett enda par fram- och bakhjul, där tyngdpunkten C befinner sig i mitten mellan fram- och bakaxeln.



(a) Robotens två fram- respektive bakhjul approximeras som ett fram- och bakhjul. Roboten approximeras som en punkt (C) som rör sig. (b) Kinematisk modell över roboten. Förklaring för alla vinklar och hastigheter återfinns i texten.

Figur 4.2.1: Roboten approximeras som en 2WS cykel-modell.

Modellen över kinematiken hos roboten som beskrivs i Figur 4.2.1b antar en plan styv kroppsrörelse utan hjulslirning i ett fast X-Y-koordinatsystem. Med hänvisning till figuren definieras punkt C som fordonets tyngdpunkt, vars koordinater beskriver fordonets position i det fasta koordinatsystemet. \mathbf{V} är hastighetsvektorn vid tyngdpunkten, vilken bildar en vinkel med fordonets längdaxel och X-axeln. Ψ är fordonets kursvinkel. β är sladdvinkeln, dvs. vinkeln mellan tyngdpunktens linjära hastighet och fordonets längdaxel. v_f och v_r är hastigheterna för fram- respektive bakhjulen. Dessa beräknas med vinkelhastighe-

terna för fram- och bakhjul enligt $v = \omega r$. Vidare har vi att δ_f och δ_r är styrvinklarna för fram- och bakhjulen, definierade som vinkeln mellan varje hjuls längdaxel och fordonets längdaxel. l_f är avståndet från fordonets tyngdpunkt till framhjulsaxeln och l_r är avståndet från fordonets tyngdpunkt till bakhjulsaxeln. Med dessa definitioner kan robotens kinematik beskrivas med följande ekvationer:

$$\dot{X} = V \cos(\psi + \beta) \quad (4.2.1)$$

$$\dot{Y} = V \sin(\psi + \beta) \quad (4.2.2)$$

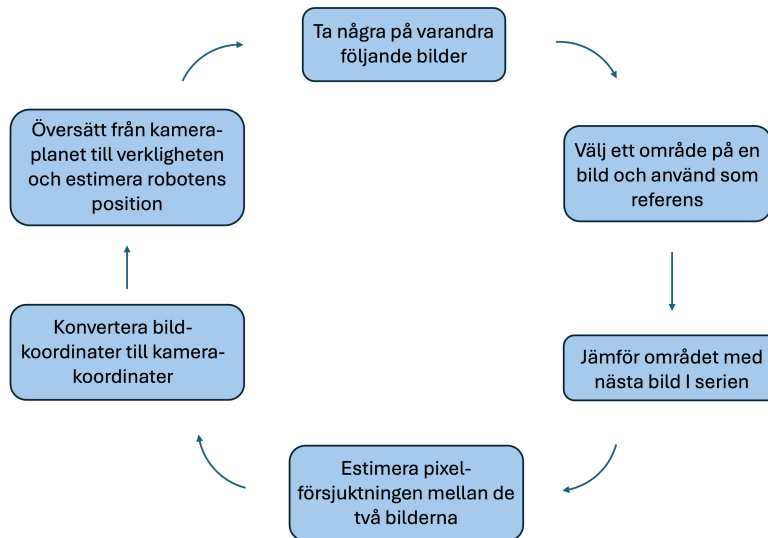
$$\dot{\psi} = \frac{V \cos \beta (\tan \delta_f - \tan \delta_r)}{l_f + l_r} \quad (4.2.3)$$

$$\beta = \tan^{-1} \left(\frac{l_r \tan \delta_f + l_f \tan \delta_r}{l_f + l_r} \right) \quad (4.2.4)$$

$$V = \frac{(r\omega_f) \cos \delta_f + (r\omega_r) \cos \delta_r}{2 \cos \beta} \quad (4.2.5)$$

4.2.1 Visuell odometri

Visuell odometri (hädanefter "VO") är en metod som bygger på att uppskatta en kameras position och rörelse utifrån en bildsekvens, och är en grundläggande teknik inom datorseende och navigation av AMR:er. Denna metod ger stegvis uppskattning av fordonets position genom att successivt skapa 3D-punktmoln och jämföra bilder i en sekvens för att sedan spara den relativa skillnaden mellan varje bild (Aqel m. fl., 2016). Metoden beskrivs visuellt i Figur 4.2.2.



Figur 4.2.2: VO flödesschema

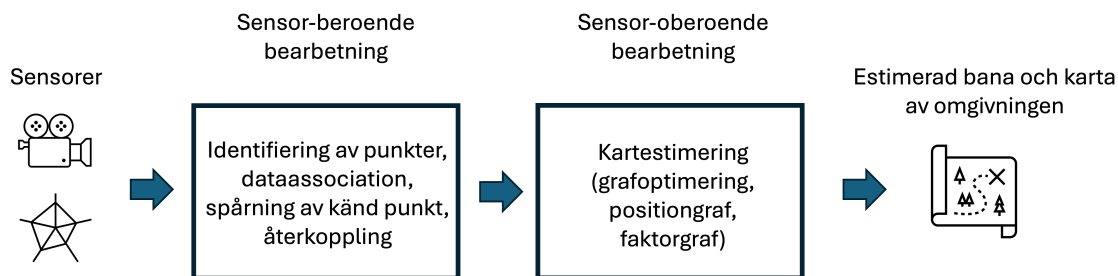
Det finns olika typer av VO där vanliga inkluderar:

- **Monokulär VO:** Använder endast en kamera och kan inte skapa ett fullt 3D-punktmoln eftersom det inte finns någon djupsensor. Istället uppskattar den djup genom att analysera förändringar mellan bilderna, detta ger dock en väldigt grov och skalfri uppskattning av djupet.
- **Stereobaserad VO:** Två kameror som används för att beräkna djup med hjälp av triangulering. Stereobaserad VO kan skapa ett tätt punktmoln genom att beräkna djup genom att jämföra djupet mellan två bilder tagna från olika vinklar.
- **RGB-D:** Kombinerar en vanlig RGB-kamera och en djupsensor för att skapa ett väldigt exakt punktmoln där varje pixel i bilden kan kopplas till ett djupvärde.

4.3 SLAM

SLAM (simultaneous localization and mapping) är en metod som används i autonoma fordon för att samtidigt bygga upp en karta över omgivningen och bestämma sin egen position i denna. Till skillnad från enbart visuell odometri innehåller SLAM ofta tekniker för att minska ackumulerande fel, exempelvis genom återknytning till tidigare besökta och kända platser.

Generellt sett finns det två olika tekniska delar till att uppnå SLAM. Den första delen är bearbetning av sensordata som bygger på att identifiera distinkta punkter och följa dem över tid. Den andra delen bygger på estimering och optimering av grafer där informationen från den första delen behandlas. Målet är att optimera hela banan och kartan som roboten byggt upp över dess gång (MathWorks, 2025). Processen beskrivs visuellt i Figur 4.3.1



Figur 4.3.1: SLAM flödesschema

I detta projekt användes en ZED stereokamera för att genomföra SLAM. Kameran genererar ett tätt punktmoln genom att analysera bilder från två separata linser, vilket möjliggör en exakt uppskattning av djup i varje bildsekvens. För att implementera funktionalitet och genomföra SLAM används kodexempel från ZED SDK (Software Development Kit), vilket är Stereolabs officiella mjukvaruutvecklingsverktyg för att hantera deras kameror (StereoLabs, 2025). Med modifiering och anpassning av kod kan robotens position uppskattas i ett globalt kartlagt rum.

4.4 Planering av vägpunktbanor

Målet var att roboten ska kunna navigera självständigt mellan förutbestämda punkter längs förutbestämda banor, vilket kallas vägpunktskartläggning (hädanefter VPK). Det finns flera olika metoder för att uppnå detta, och nedan presenteras en av metoderna som används inom ramen för projektet.

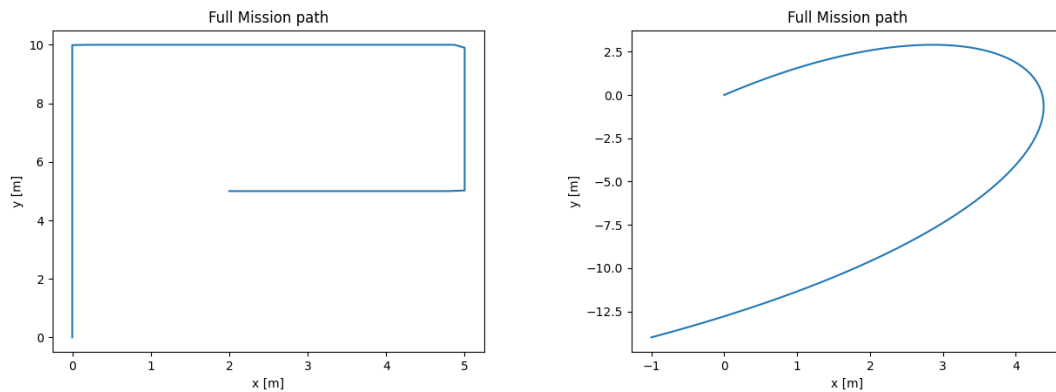
Problemet kan formuleras enligt följande: Givet en bana som roboten förväntas följa

$$P_b(s) = \begin{bmatrix} X_b(s) \\ Y_b(s) \end{bmatrix}, \quad s : s_0 \rightarrow s_n$$

där s är sträckan roboten färdas, designa en rutt för roboten så att:

$$T(t) = \begin{bmatrix} X(t) \\ Y(t) \\ \psi(t) \end{bmatrix}, \quad t : t_0 \rightarrow t_n$$

Exempel på två banor kan ses i Figur 4.4.1. Formuleringen håller för både kontinuerliga och diskontinuerliga kurvor.



(a) En bana med tre 90-graders svängar. Roboten kommer behöva stanna i varje hörn och svänga kring sin egen axeln.

(b) En mjuk och kontinuerlig bana för roboten att följa. Segmentet är en Bézier kurva, genererad med matplotlib funktioner.

Figur 4.4.1: Exempel på banor för roboten att följa.

Vad som inte framgår i figurerna ovan är hur robotens kursvinkel ψ ska variera längs med banan. Detta då det finns flera möjligheter, till skillnad från X och Y (mer om hur dessa planeras senare). De två mest praktiska villkoren för kursvinkeln är följande:

Sidohalkningsfri manöver Detta villkor för styrvinkel säkerställer att sladdvinkeln sätts till noll för hela sträckan, dvs:

$$\beta(s) = 0, \quad s : s_0 \rightarrow s_n \quad (4.4.1)$$

Enligt (4.2.4) uppnås detta om vi sätter $\delta_f = -\delta_r$ i modellen över roboten. Styrvinkeln sätts sedan för att matcha den tangentiella vinkeln av kurvan, dvs:

$$\psi(s) = \theta(s), \quad s : s_0 \rightarrow s_n \quad (4.4.2)$$

där den tangentiella vinkeln θ kan ges vid varje punkt av kurvan som:

$$\theta = \arctan \frac{dY}{dX} \quad (4.4.3)$$

Parallellstyrningstyrning Parallelstyrning, tidigare benämnt krabbstyrning, utgår ifrån att samtliga hjul på roboten styr i samma riktning. Detta innebär också att sladdvinkeln blir densamma enligt 4.2.4.

$$\delta_f = \delta_r = \beta \quad (4.4.4)$$

Som bekant innebär detta att ingen rotation uppstår, dvs:

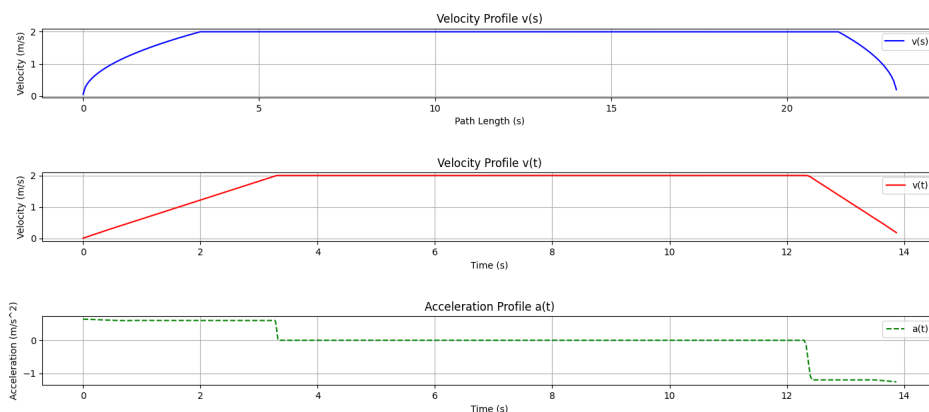
$$\psi(s) = \psi_0 \quad s : s_0 \rightarrow s_n \quad (4.4.5)$$

Acceleration och hastighetsplanering Roboten är begränsad av sin maximala acceleration, retardation och hastighet. Den sistnämnda är särskilt viktig eftersom roboten måste kunna köras säkert. De förstnämnda är främst relaterade till begränsningar i robotens hårdvara, såsom maximal effekt. Av denna anledning måste hastighetsprofiler planeras noggrant.

I Figur 4.4.2 visas ett exempel på en hastighetsprofil med $v_{max} = 2$ m/s, $a_{max} = 0.6$ m/s² och $a_{min} = -1.2$ m/s². Hastigheten som funktion av sträckan s genereras genom att integrera den maximala accelerationen vid varje punkt på kurvan. Denna acceleration kan vara godtycklig, men i detta fall har vi valt följande:

$$f_{acc} = \sqrt{2 \cdot a_{max} \cdot s + v_{start}^2} \quad f_{ret} = \sqrt{|v_{end}^2 + 2 \cdot a_{ret} \cdot (S - s)|} \quad (4.4.6)$$

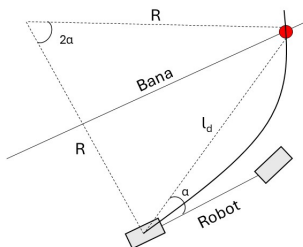
Detta resulterar i linjära hastighetsökningar över tiden, vilket kan ses i den mellersta grafen i Figur 4.4.2. Det är också värt att nämna att hastigheten kan begränsas beroende på hur skarp kurvan är, men eftersom roboten aldrig kör särskilt snabbt har detta utelämnats. Notera även att små fel uppstår vid randen på grund av att numerisk integration användes (Danwei Wang, 2001).



Figur 4.4.2: Grafen högst upp visar hastigheten som funktion av banans längd s . De två efterföljande graferna är hastigheten och accelerationen som funktion av tiden t .

4.4.1 Pure Pursuit

En pure-pursuit controller är en algoritm som använder sig av en robots (eller fordons) position, geometri och kinematik för att följa en bana. En punkt på banan agerar som en referens för roboten att styra mot, och således följer roboten approximativt banan. Metoden demonstreras i Figur 4.4.3.



Figur 4.4.3: Pure pursuit algoritmen anpassar styrvinkeln utefter den egna positionen och positionen för en punkt på den givna banan. Figuren uppvisar algoritmen för en 2WS modell.

Genom att använda den kinematiska modellen presenterad i sektion 4.2 beräknas styrvinklar enligt följande för 2WS fallet:

$$R = \frac{L}{\tan(\delta)} \Rightarrow \delta = \arctan\left(\frac{2L \sin(\alpha)}{l_d}\right) \quad (4.4.7)$$

Precis som innan gäller det även för 4WS att $L = 0.5L$ (Ding, 2020).

5 Mekanisk konstruktion

Den mekaniska konstruktionen utgör en central funktion av en AMR då den direkt påverkar systemets strukturella integritet, rörelseförmåga och integrationsmöjligheter för övriga delsystem. Med utgångspunkt i föregående års konstruktion har detta område fokuserat på att förbättra befintliga lösningar, åtgärda brister och ta fram en mer robust robotplattform.

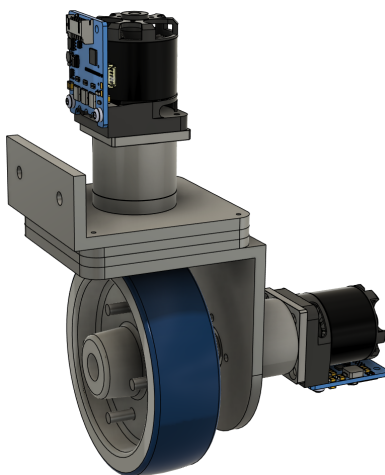
Utvecklingsarbetet har särskilt omfattat en ny hjulmodul. Den tidigare konstruktionen utvärderades och förbättrades med målet att öka robusthet och hållfasthet. För att underlätta integreringen av dessa systemkomponenter har även plexiglas på robotens övre och undre delar konstruerats om. Genom att justera form, infästningsmetod och materialanvändning har dessa element optimerats för enklare åtkomst och minskad monterings- och underhållstid. Samtidigt har komponentfästen utvecklats för att säkert montera ZED-kamera, Jetson-dator och ett nytt intentionssystem. Tillsammans bidrar dessa mekaniska lösningar till att skapa en mer flexibel, driftsäker och framtidssäkrad plattform. Den mekaniska delen av arbetet har därmed inte enbart varit en bärande struktur, utan en aktiv del i att möjliggöra samverkan mellan robotens olika tekniska delsystem.

5.1 Hjulmodul

Hjulmodulen är en central del i robotens mekaniska system och har en avgörande funktion i att möjliggöra framdrivning, manövrering och stabilitet. Utformningen av hjulmodulen påverkar rörelseförmågan, robotens mekaniska robusthet, lastbärande kapacitet och driftsäkerhet. Fokus har legat på att analysera och åtgärda tidigare svagheter i konstruktionen, samtidigt som nya lösningar har utvärderats för att uppnå en mer stabil och tillförlitlig funktion. Arbetet med hjulmodulen har inletts med en konceptfas där olika lösningar för drivning, styrning och infästning studerats och jämförts utifrån teknisk genomförbarhet, komplexitet och anpassning till övriga delar av roboten. Samtidigt har hänsyn tagits till komponentval, integrationsmöjligheter och framtida underhåll.

5.1.1 Tidigare hjulmodul

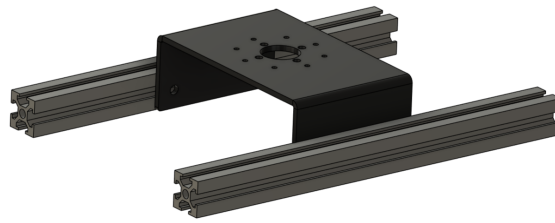
I den befintliga konstruktionen användes ett enkelt vinkelfäste monterat på en aluminiumprofil för att fästa hjulmodulen, se Figur 5.1.1. Eftersom infästningen endast gav stöd från ena sidan uppstod ett böjmoment vid belastning, vilket ledde till oönskad deformation och minskad stabilitet.



Figur 5.1.1: Tidigare hjulmodul

5.1.2 Överdel hjulmodul

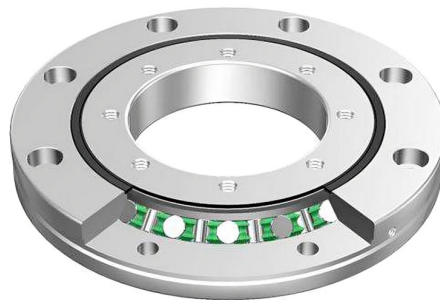
För att förbättra stabiliteten och minska oönskade böjmoment utreddes alternativa lösningar med fokus på ökad strukturell symmetri och lastfördelning. Ett centralt mål i konceptutvecklingen var att säkerställa symmetrisk infästning av hjulmodulen genom användning av två parallella aluminiumprofiler för att eliminera de böjande moment som uppstod vid ensidig infästning samt öka den mekaniska styvheten. Flera koncept av fäste som håller ihop hjulmodul med AMR:ens ram utvärderades utifrån faktorer som minimering av deformationer vid belastning samt kompatibilitet med befintliga aluminiumprofiler. Lösningen som valdes för infästningen blev ett vattenutskuret och bockat vinkelfäste i 3 mm stålplåt med dubbelsidig infästning i aluminiumprofilerna, se Figur 5.1.2. Till skillnad från den föregående lösningen vändes även fästet upp och ner för att på så sätt sänka roboten närmare marken. Detta bidrar till en lägre tyngdpunkt och ökad stabilitet. Detaljritning för hjulmodulens överdel återfinns i Bilaga A.1.



Figur 5.1.2: Vinkelfäste med aluminiumprofiler

5.1.3 Svänglager

Eftersom roboten är fyrhjulsstyrd krävs att varje hjulmodul utöver drivmotorn också har en dedikerad svängmotor samt ett lager som möjliggör rotation mellan hjulmodulens över- och underdel. I föregående års konstruktion användes ett vridbeslag för möbler och ostbrickor som svänglager, vilket resulterade i låg precision och ett betydande lagerglapp. För att förbättra detta valdes i årets konstruktion ett kryssrullager med monteringshål i både inner- och ytterring. Kryssrullager utmärker sig genom sin höga precision och styvhet, vilket uppnås genom att rullarna är placerade i ett korsmönster med 90° vinkel mot varandra (se Figur 5.1.3).



Figur 5.1.3: Kryssrullager med genomskärning (Remlagret.se, 2025).

Denna typ av lager möjliggör god upptagning av radiella, axiella och momentlast, vilket är fördelaktigt i denna typ av konstruktion. Montering av lagret underlättades dessutom av dess integrerade monteringshål och tunna profil, vilket möjliggjorde en stabil och kompakt infästning mellan modulens övre och undre del. Eftersom inner- och ytterringen har samma tjocklek krävdes två separata distanser för att möjliggöra korrekt montering av lagret – en för innerringen och en för ytterringen (se Figur 5.1.4). Likt hjulmodulens

överdel vattenskars dessa ut i 3 mm stålplåt enligt detaljritningen som återfinns i Bilaga A.4 och A.5.



Figur 5.1.4: Kryssrullager med distanser

5.1.4 Nederdel hjulmodul

Även hjulmodulens nedre del behövde omdesignas. Denna del behövde först och främst förlängas på grund av den större hjuldiametern hos navmotorhjulet. Dessutom krävdes en uppdatering av hålbilderna för montering av både lager och drivhjul, då dessa skiljde sig från den tidigare konstruktionen. För att minimera utböjning i den nedre delen designades två triangulära förstärkningsvinklar, vilka TIG-svetsades fast på båda sidor av komponenten. Den nyframtagna nederdelen tillverkades på liknande sätt som överdelen i 3 mm stålplåt och kan ses i Figur 5.1.5).

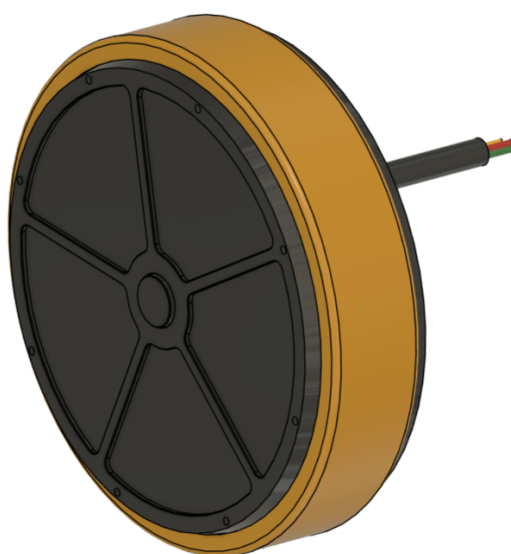


Figur 5.1.5: Underdel hjulmodul (plåt)

5.1.5 Navmotorhjul

Den tidigare konstruktionen av hjulmodulen innebar att all radiell belastning från hjulet togs upp av växellådan vars kapacitet var begränsad till 200 N per hjul. Då belastningen i praktiken inte alltid fördelas jämnt över samtliga hjulmoduler kunde detta leda till att gränsen för vad växellådan klarade överskreds, vilket begränsade robotens totala lastkapacitet. Denna konstruktion innebar också att växellådan och drivmotorn satt monterade utstickande från sidan på den nedre delen av hjulmodulen, endast ett par centimeter ovanför marken. För att skydda dessa komponenter hade ett skyddande hölje i PLA-plast monterats utanpå dessa. Dock blev denna lösning inte särskilt smidig då detta tar stor plats när hjulen svänger och höljet riskerar att ta i eventuella hinder på marken.

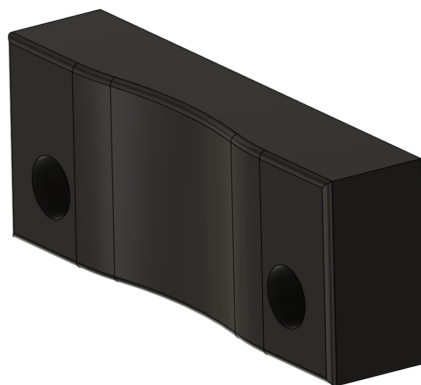
För att förbättra konstruktionen togs ett nytt koncept fram där det ursprungliga drivhjulet ersattes av ett hjul med integrerad navmotor, se figur 5.1.6 . Belastningen leds nu genom hjulets axel och lager, vilka har betydligt högre lastkapacitet. Detta minskar risken för överbelastning och ökar tillförlitligheten. Den nya lösningen säkerställer att roboten klarar en last på 50 kg med god säkerhetsmarginal. Drivhjulet som planerades att användas var DirectDrive M1502E-111 som ska klara radiell belastning på upp till 800 N (DirectDriveTech, 2024). Då hjulet levereras utan däckgummi planerades att göra 3D-utskrivna däck i TPU-plast (orange i Figur 5.1.6).



Figur 5.1.6: Hjul med navmotor

5.1.6 Styrstopp

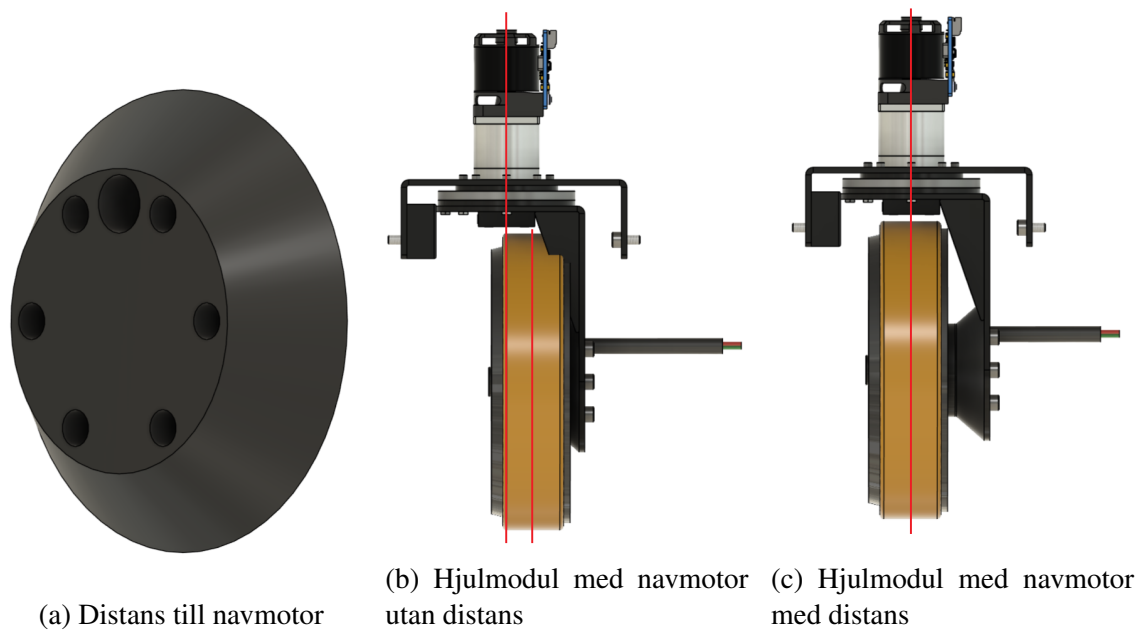
Dessutom integrerades ett mekaniskt rotationsstopp tillverkat i PLA-plast i hjulmodulen, se Figur 5.1.7. Stoppet begränsar hjulets vridning till $\pm 90^\circ$, vilket dels förhindrar överrotation och kabelslitage, men framför allt möjliggör kalibrering av svängmotorerna. Genom att skapa en fysisk begränsning kan motorsystemet identifiera en absolut nollposition, vilket underlättar för exakt styrning.



Figur 5.1.7: Styrstopp

5.1.7 Hjuldistans

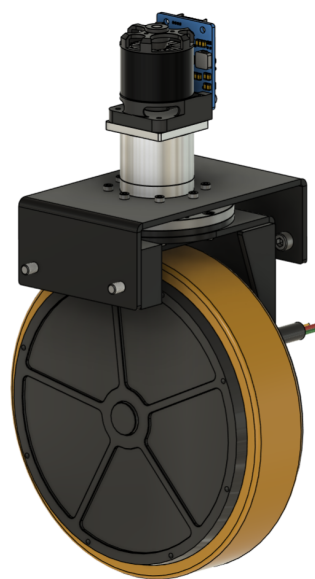
För att hjulet ska hamna rakt under svängaxeln och därmed undvika att avståndet mellan hjulen förändras vid styrning, måste navmotorn placeras med rätt distans från hjulmodulens underdel (se Figur 5.1.8b och 5.1.8c). För detta ändamål designades en konformad distans med monteringshål samt genomföring för navmotorns kablage (se Figur 5.1.8a). Distansen har även en bärande funktion genom att den fördelar lasten över en större yta av underdelen. Som visas i Figur 5.1.8c överlappar distansen även delvis med de fastsvetsade vinkelfästena, vilket bidrar till att öka styvheten i konstruktionen. Distanserna är ännu inte tillverkade, men planerades att tillverkas i PLA-plast med hjälp av 3D-utskrift enligt detaljritningen som återfinns i Bilaga A.3.



Figur 5.1.8: Distans i hjulmodul med navmotor

5.1.8 Framtaget koncept

Tillsammans resulterade de modifikationer som har presenterats i 5.1.2 till 5.1.7, i det slutgiltiga konceptet som visas i Figur 5.1.9



Figur 5.1.9: Hjulmodul koncept med navmotor

5.1.9 Slutgiltig lösning

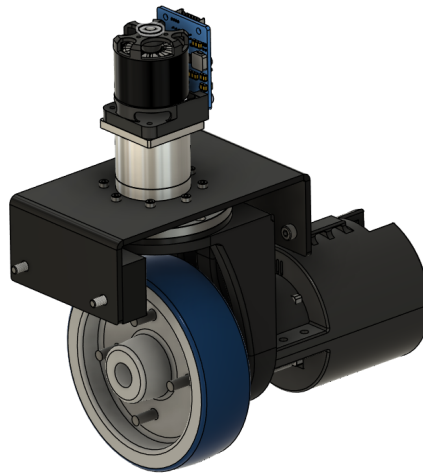
Då de planerade navmotorerna inte kunde beställas i tid, krävdes en alternativ lösning för att möjliggöra fortsatt utveckling. Den valda lösningen innebar att den nyutvecklade hjulmodulen kombinerades med motor och drivhjul från föregående års konstruktion. För att denna kombination skulle vara möjlig behövde hjulmodulens underdel konstrueras om, eftersom infästningarna för motor och svänglager skiljde sig åt, samtidigt som det tidigare drivhjulet hade en mindre diameter än navmotorhjulet.

På grund av begränsad tid tillverkades den nya underdelen i PLA-plast med hjälp av 3D-skrivare (se Figur 5.1.10), istället för att bockas och svetsas i 3 mm stålplåt som fästena som gjordes till navmotorerna. Då plastens mekaniska egenskaper är väsentligt svagare än stålets, behövde konstruktionen förses med extra materialtjocklek och förstärkas med kraftiga vinklar för att uppnå tillräcklig styvhet och minimera deformation vid last. Detaljritningen för underdelen som är 3D-utskrivna återfinns i Bilaga A.2



Figur 5.1.10: Underdel hjulmodul (3D-utskrivna)

Den slutliga hjulmodulen som monterades på roboten presenteras i Figur 5.1.11.



Figur 5.1.11: Hjulmodul med 3D utskrivna underdelar och gammal motor

Hjulmodulernas komponenter, såsom vinkelfästen och distanserna till kryssrullagren, har även blåstrats och lackerats med rostskyddsgrundfärg och svart täckfärg. Detta gäller även de fyra underdelarna i plåt som tillverkades för navmotorhjulen. Detta gör det lätt för eventuell framtida kandidatgrupp att installera navmotorhjulen. Åtgärderna vidtogs främst för att motverka rostangrepp, vilket har varit ett återkommande problem i tidigare års projekt, men även för att förbättra det visuella helhetsintrycket.

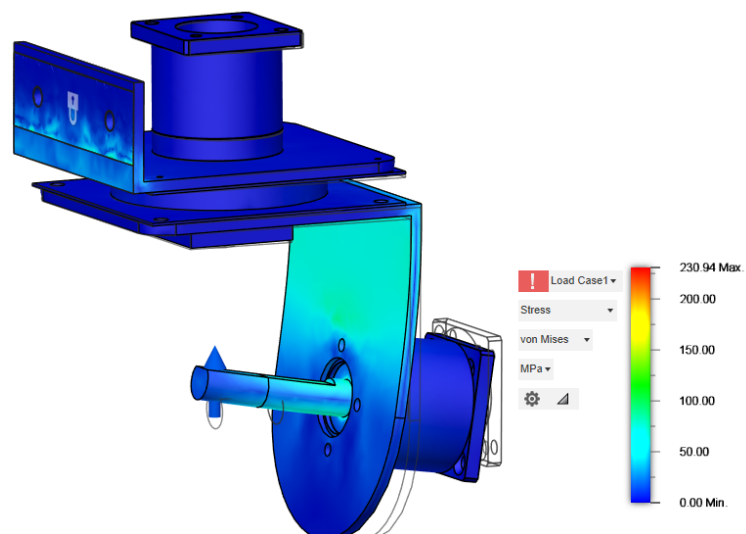
5.1.10 FEM-analys av hjulmodul

För att säkerställa att det nya hjulmodulkonceptet uppfyller stabilitetskraven och överträffar föregående års konstruktion genomfördes FEM-analyser innan tillverkning. Analyserna utfördes i programmet *Fusion 360* med fokus på materialspänningar och utböjning vid belastning. För att efterlikna en situation där roboten belastas med totalt 50 kg, inklusive egenvikten på 30 kg, applicerades en last motsvarande 200 N per hjulmodul.

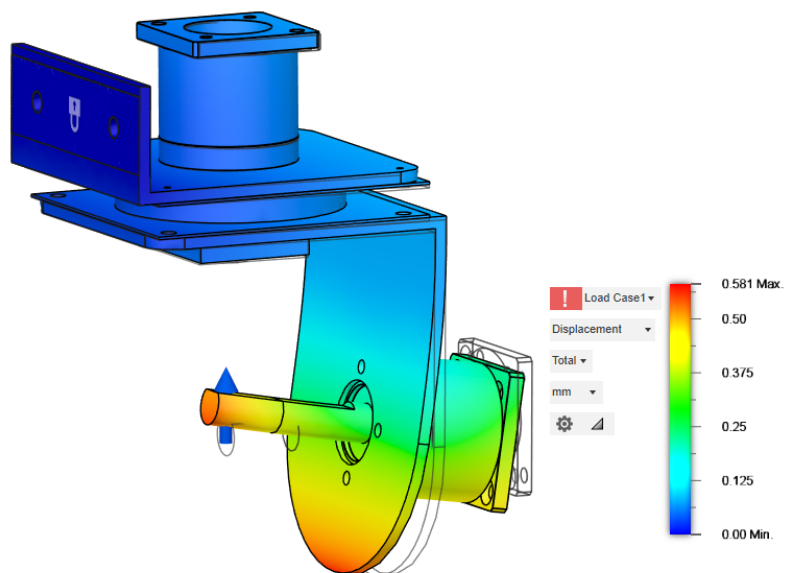
För att även ta hänsyn till det vridmoment som uppstår till följd av hjulaxelns längd i förhållande till infästningspunkten och belastningsytan på hjulet förlängdes axlarna i samtliga analyser. Den del av axeln som är infogad i hjulet modellerades med ett material med oändlig styvhet, eftersom dess deformation i praktiken beror på själva hjulets egenskaper. Då fokus i denna analys ligger på hjulmodulernas prestanda exkluderades därför hjulens flexibilitet från simuleringen.

Som referens jämfördes det nyframtagna konceptet med den tidigare hjulmodulen, se Fi-

gur 5.1.12 och Figur 5.1.13.

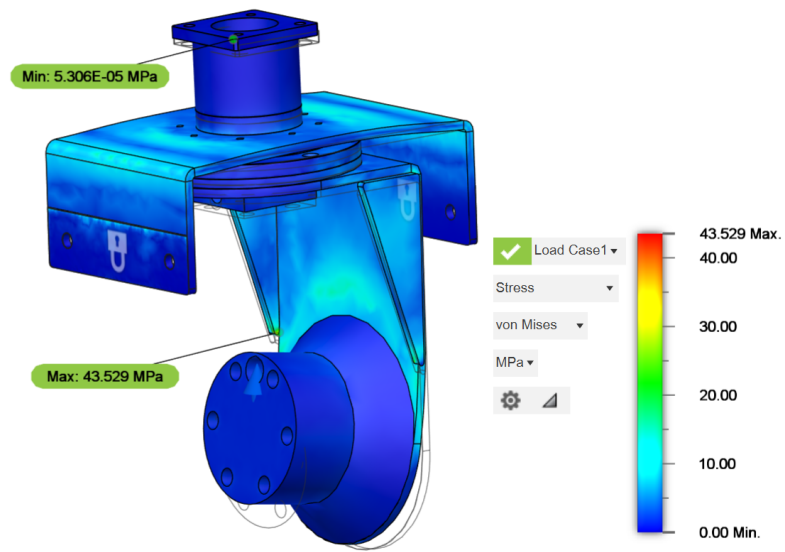


Figur 5.1.12: Materialspänning [MPa] i tidigare hjulmodul

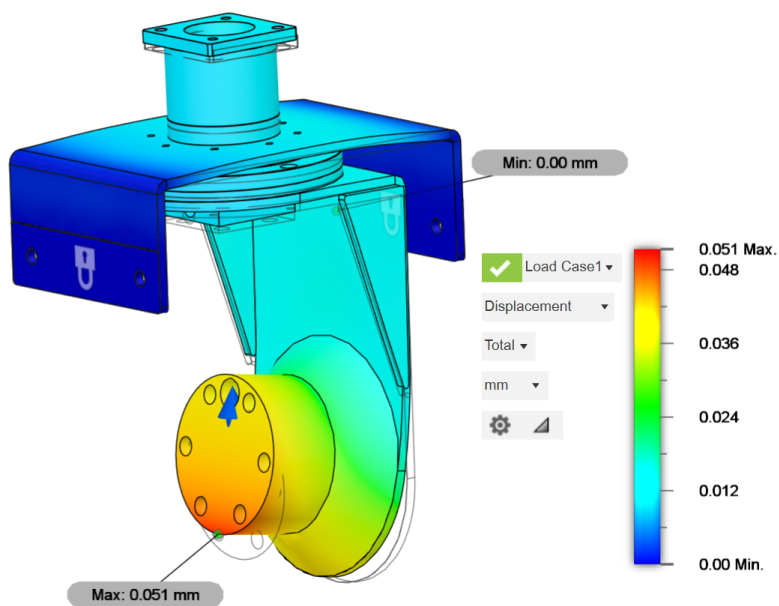


Figur 5.1.13: Utböjning [mm] i tidigare hjulmodul (överdriven i figur)

Därefter analyserades det nya konceptet med integrerad navmotor, se Figur 5.1.14 och Figur 5.1.15.

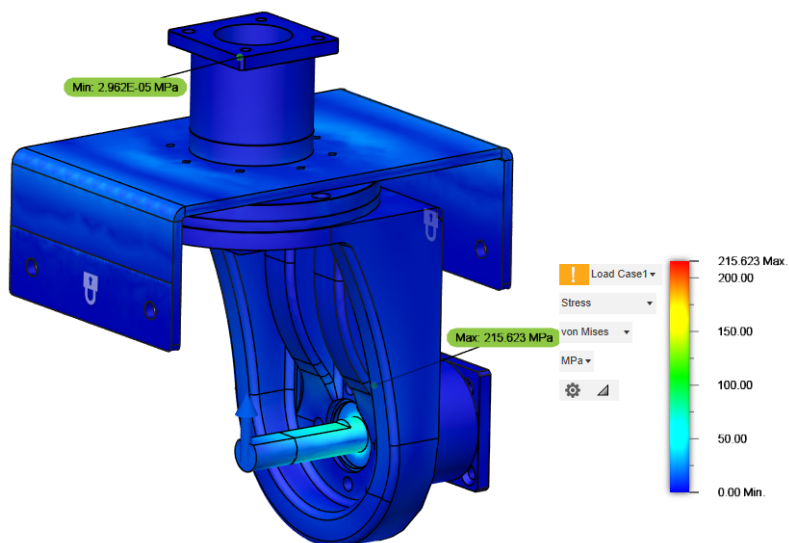


Figur 5.1.14: Materialspänning [MPa] i hjulmodul med navmotor

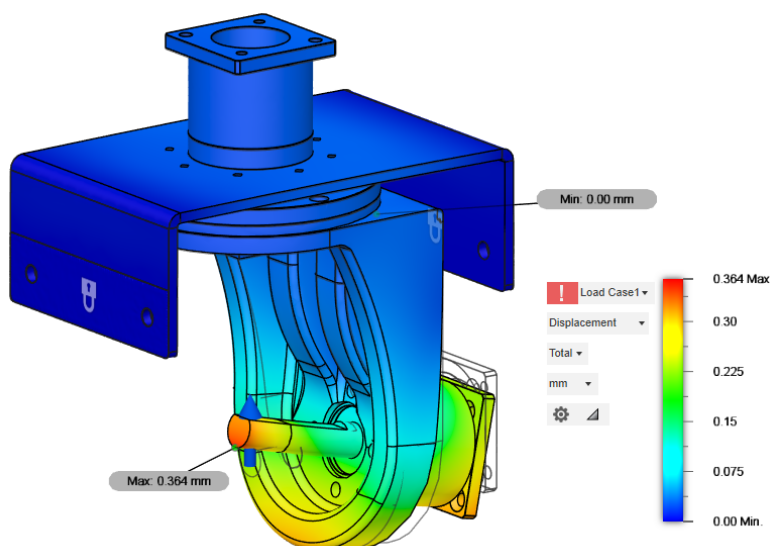


Figur 5.1.15: Utböjning [mm] i hjulmodul med navmotor (överdriven i figur)

Slutligen analyserades en version med 3D-utskrivna underdel i PLA, se Figur 5.1.16 och Figur 5.1.17.



Figur 5.1.16: Materialspänning [MPa] i hjulmodul med 3D-utskrivna underdel



Figur 5.1.17: Utböjning [mm] i hjulmodul med 3D-utskrivna underdel (överdriven i figur)

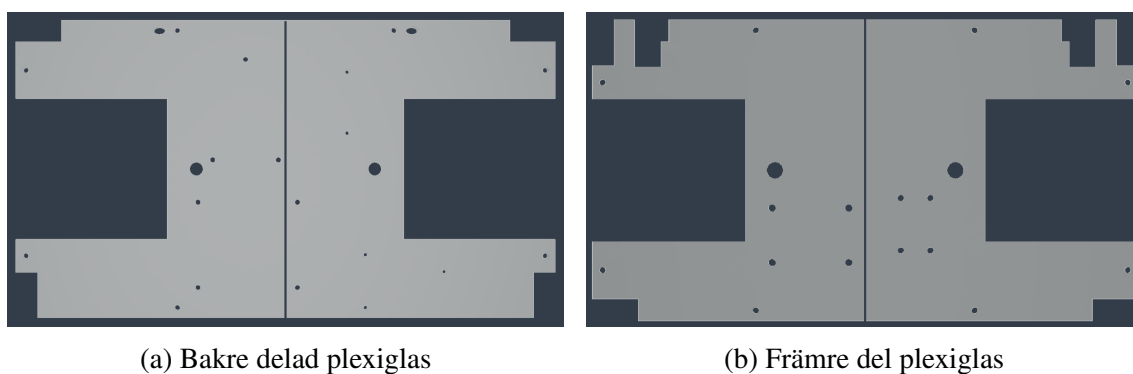
Utböjningen hos den nya hjulmodulen uppmättes till endast 0,051 mm, vilket motsvarar cirka 9% av den tidigare konstruktionens utböjning på 0,581 mm vid samma belastning. Detta indikerar en avsevärt högre styvhet och förbättrad strukturell prestanda hos den nya lösningen. Även den tillämpade varianten med 3D-utskrivna underdel i PLA uppvisade en utböjning på endast 0,364 mm, motsvarande cirka 63% av den tidigare konstruktionens utböjning. Det är samtidigt viktigt att notera att det imprecisa lagret som användes i den äldre konstruktionen också var en betydande källa till utböjning, vilket inte beaktats i denna FEM-analys.

Utöver att utböjningen har minskats avsevärt i de nya konstruktionerna har även materialspänningarna reducerats från 230,9MPa i den tidigare lösningen till enbart 43,5MPa för hjulmodulen med navmotor, samt 215,6MPa för versionen med 3D-utskrivna underdel i PLA. Den höga spänningen i den sistnämnda konfigurationen är lokaliserad till infästningen mellan växellådan och underdelen, och bedöms vara en följd av ett meshningsfel i denna zon. Problemet kvarstår trots användning av olika meshtyper och förfiningsnivåer. Vid en kvalitativ granskning av spänningsfördelningen, både visuellt via färgskalan och med hjälp av probe-verktyget i *Fusion 360*, framgår det tydligt att spänningarna i övriga delar av modulen är låga och väl inom acceptabla nivåer (< 60 MPa).

5.2 Plexiglas

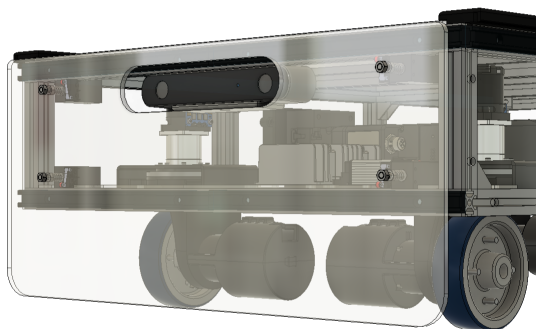
För att möjliggöra montering av den nya hjulmodulen och elsystemets komponenter krävdes nya plexiglasskivor på undersidan av roboten, både fram och bak. Genom att utnyttja en korrekt positionering av samtliga komponenter i Fusion kunde hålen för skruvar enkelt placeras genom att projektionsverktyget användes för att överföra monteringsfästena direkt till plexiglasskivornas skisser. Dessa skisser exporterades därefter som DXF-filer och fördes över till en laserskärare för tillverkning (se Bilaga A.6 och A.7 för detaljritningar).

Vid montering visade det sig vara utmanande att få plexiglasskivorna på plats utan att demontera ett flertal av robotens aluminiumprofiler. För att förenkla denna process togs beslut om att dela plexiglasets på mitten genom ytterligare en laserskärning (se Figur 5.2.1a och 5.2.1b). Detta är en effektiv lösning som underlättar för både montering och framtida demontering.



Figur 5.2.1: Laserskurna plexiglas

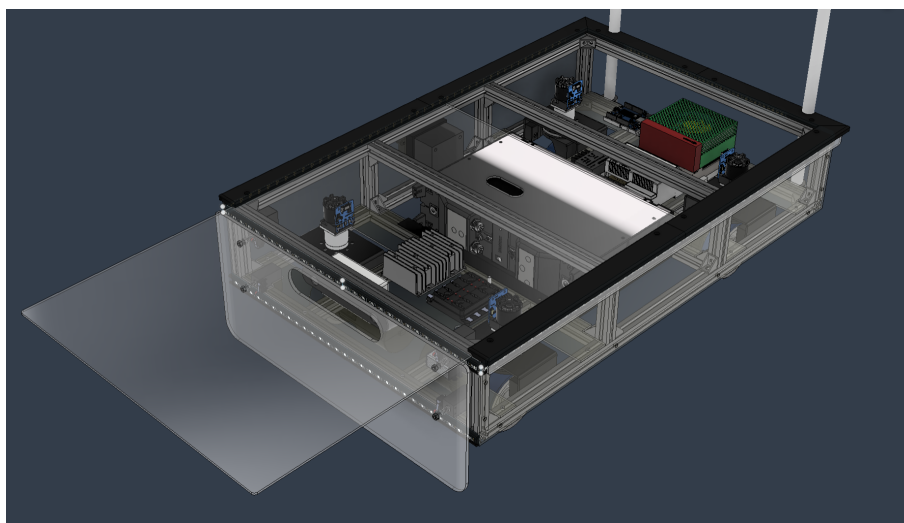
För att säkerställa fri sikt för stereokameran, som är monterad på robotens framsida, har det främre plexiglasets omkonstruerats jämfört med föregående års version. För att undvika nedsatt kamerafunktion har ett konturformat hål laserskurits i plexiglasets (se Figur 5.2.2), noggrant anpassat efter kamerans geometri. Denna modifiering möjliggör obehindrad bildinsamling, samtidigt som plexiglasets fortsatt skyddar övriga komponenter i robotens frontsektion. Detaljritning på det främre plexiglasets återfinns i Bilaga A.10.



Figur 5.2.2: Främre plexiglas

Även det övre plexiglasets konstruerades om för att förenkla montering och demontering. I den tidigare designen var plexiglasets fastskruvat direkt i ramens aluminiumprofiler med åtta skruvar, vilket gjorde demontering och återmontering både tidskrävande och opraktisk.

Den nya lösningen baserades på en skjutmontering, där plexiglasets kunde skjutas på plats med hjälp av 3D-utskrivna fästen tillverkade i PLA (se Figur 5.2.3). Dessa fästen, som beskrivs närmare i avsnitt 5.3, skruvades fast i ramen och möjliggör verktygsfri montering och demontering av det övre plexiglasets. Resultatet blev en mer användarvänlig lösning som förenklar åtkomsten till robotens interna komponenter. Detaljritning på det övre plexiglasets återfinns i Bilaga A.9.

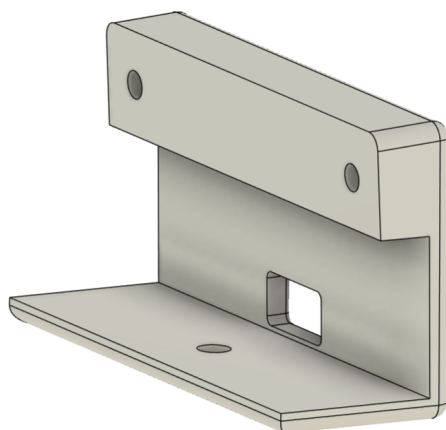


Figur 5.2.3: Övre plexiglas

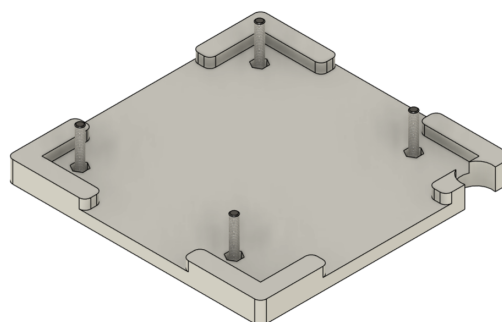
5.3 Fästen till komponenter

I samband med att nya komponenter tillfördes roboten behövdes även särskilda fästen konstrueras för att möjliggöra säker och funktionell montering. För ZED-kameran, som placerades längst fram på robotens övre del, tillverkades ett anpassat 3D-utskrivet fäste i PLA (se Figur 5.3.1a och Bilaga A.8 för detaljritning). Denna placering möjliggör fri sikt framåt utan att kamerans synfält skymdes av andra komponenter.

Ett separat fäste designades även för Jetson-datorn, med syftet att skapa en stabil men samtidigt lätt avtagbar montering. Även detta fäste tillverkades i PLA och monterades med 20 mm långa M4-skruvar. Kombinationen av fäste och skruvar förhindrar oönskad förflyttning i sid- och längdled under drift (se Figur 5.3.1b). Däremot medgav konstruktionen möjlighet att lyfta datorn rakt upp, vilket underlättar montering och demontering av datorn.



(a) Fäste för ZED v1 kamera



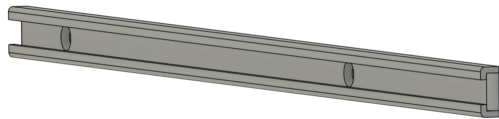
(b) Fäste för Jetson-datorn

Figur 5.3.1: CAD-modeller: Fästen för dator och kamera

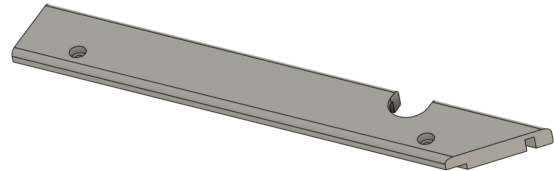
Även för intentionssystemets LED-slingor som tas upp i del 6.4, krävdes specialtillverkade fästen för att möjliggöra enkel montering och demontering (se Figur 5.3.2a). Ett alternativ hade varit att använda den fabriksmonterade klistertejpen på slingans baksida, men detta skulle ha försvårat demontering vid underhåll eller modifiering av roboten.

I fästena som primärt möjliggjorde inskjutning av det övre plexiglasets från framsidan integrerades även en skåra avsedd för montering av långsmala, frostade plexiglasskivor (se Figur 5.3.2b). Dessa placerades utanpå LED-slingorna på robotens sidor och bakre del.

Eftersom LED-slingorna är utrustade med mycket ljusstarka dioder, bidrar det frostade plexiglas till att sprida ljuset, minska bländning och förbättra det visuella helhetsintrycket. Av samma skäl valdes även frostat plexiglas till robotens främre stötfångare (se Figur 5.3.3a och 5.3.3b).

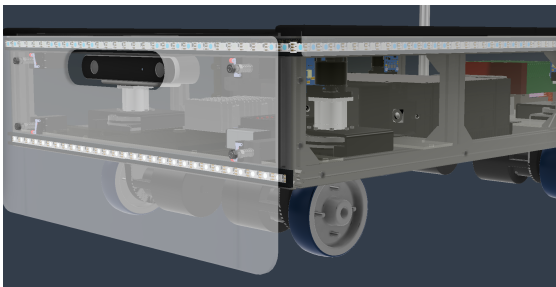


(a) Fäste för LED-slinga



(b) Fäste för frostat plexiglas

Figur 5.3.2: CAD-modeller: Fästen för plexiglas och intentionssystem



(a) Frostat plexiglas framför LED-slinga

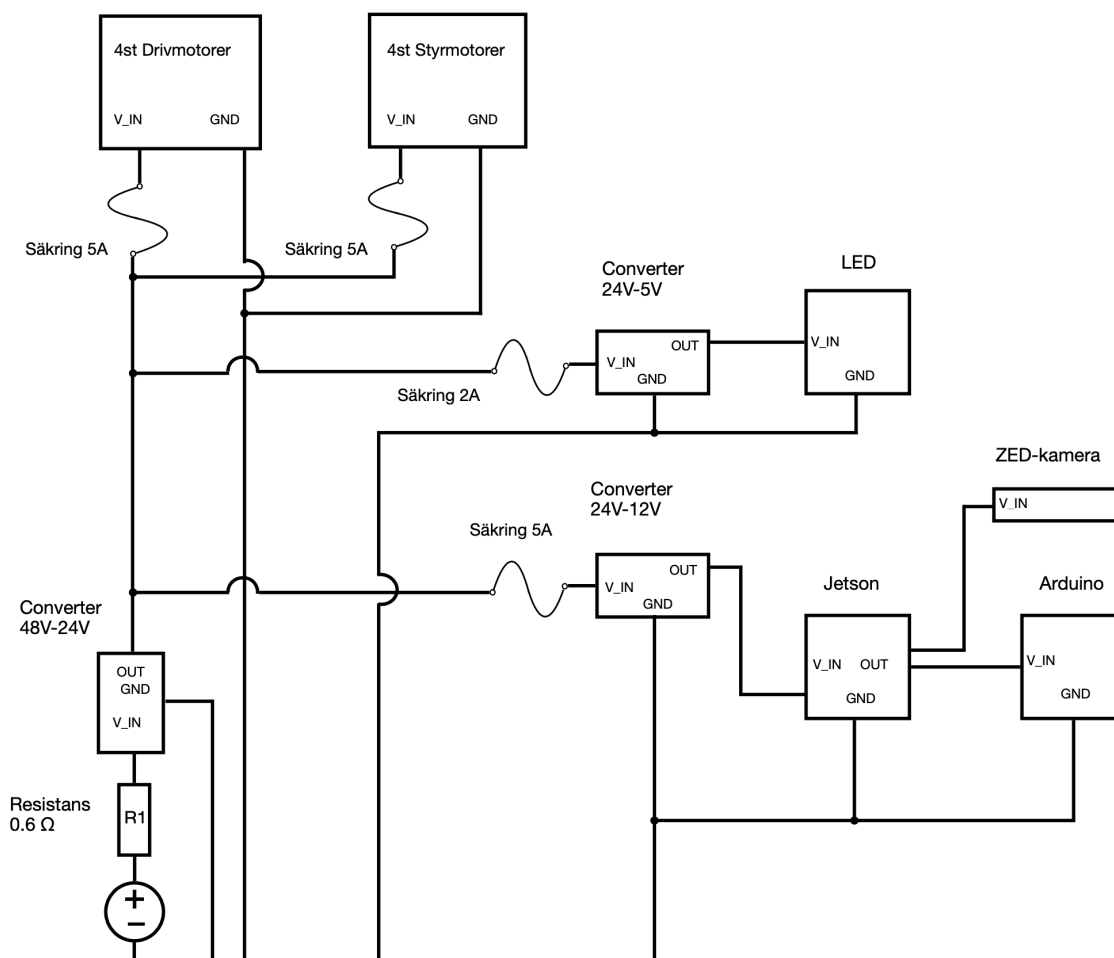


(b) LED-slinga utan frostat plexiglas

Figur 5.3.3: CAD-modeller: Framsida av robot med och utan frostat plexiglas

6 Elsystem

För att den autonoma roboten ska kunna fungera självständigt och effektivt krävs ett väl fungerande elsystem som knyter samman samtliga komponenter. Under projektets gång har systemet vidareutvecklats genom integration av nya delar och förbättring av befintliga lösningar. Batteridrift har implementerats för att kunna få roboten att fungera utan ett externt nätaggregat. Vidare har ett intentionssystem baserat på en programmerbar LED-slinga implementerats för visuell signalering. Komponentplacering har optimerats med avseende på åtkomst, kabeldragning och framtida underhåll. Fokus har legat på att konstruera ett tillförlitligt och skalbart elsystem anpassat för fortsatt utveckling. I Figur 6.0.1 illustreras elsystemets komponenter och deras inbördes kopplingar.



Figur 6.0.1: Elsystemets utformning

6.1 Kommunikationssystem

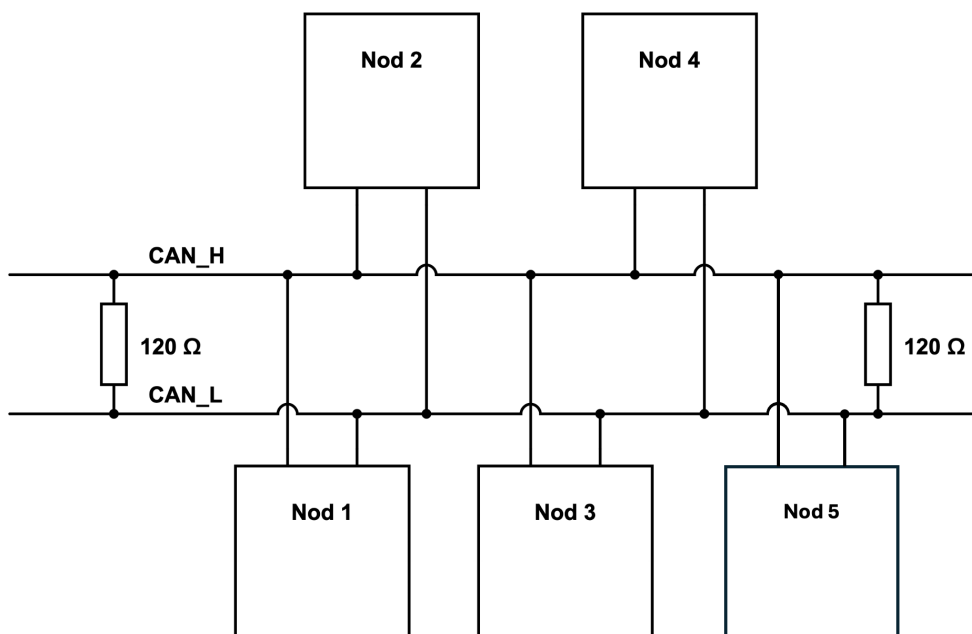
För att göra kommunikation möjlig mellan olika enheter i elsystemet används ett CAN2.0B-protokoll. Protokollet används i och med att det redan var implementerat i robotens elsystem sedan tidigare år, vilket gjorde att vi kunde återanvända hårdvara och delar av befintlig styrkod.

CAN-buss (Controller Area Network) är en standardiserad databuss för fordon som möjliggör kommunikation mellan fordonets olika elektroniska styrenheter (noder) utan en central styrenhet. Alla noder är anslutna till samma tvåtrådiga kabel som fungerar som en gemensam kommunikationslinje där endast ett meddelande kan skickas åt gången (Falch, 2025).

Kommunikationslinjen består av två ledare: CAN_H (high) och en CAN_L (low). Tillsammans skickar de meddelanden genom att variera spänningen mellan dem. När båda ledare ligger på samma spänning (typiskt 2,5 V) betyder det att ingen nod aktivt skickar data, vilket gör att andra noder får tillgång till bussen och möjlighet att skicka sina meddelanden. Detta tillstånd kallas recessivt. När meddelanden väl behöver skickas ökar spänningen på CAN_H till 3,5 V och CAN_L minskar till 1,5 V. Detta tillstånd kallas dominant och signalerar att en nod aktivt skickar data och överröstar alla recessiva signaler på bussen (Technology, n.d.).

Termineringsmotstånd på vanligtvis 120Ω behövs även vid varje ände av databussen eftersom signalerna färdas i båda riktningar. Termineringen vid vardera ände absorberar signalerna så att de inte reflekteras tillbaka, vilket skulle kunna skapa störningar och förvrängda signaler (Electronics, n.d.).

Ett system med CAN-buss möjliggör att robotens noder kan ta emot information på ett snabbt och robust sätt samtidigt som antalet kablar minimeras. En förenklad struktur av CAN-bussen för roboten visas i Figur 6.1.1 där Nod 1-4 representerar varje motor och Nod 5 är batteriet. Nytt för i år är implementationen av batteriet i CAN-bussen. Detta medför att man kan övervaka bland annat batteriets laddning och temperatur.



Figur 6.1.1: Förenklad version av CAN-buss struktur

6.2 Design av elsystem

Fokus låg på att skapa ett tillförlitligt och skalbart elsystem som stödjer robotens funktionalitet. AB Volvo använder idag 48V VARTA EasyBlade-batterier (se specifikationer i Tabell 6.2.1) i sina autonoma robotar vid fabriken Volvo Lastvagnar i Tuve. För att efterlikna AB Volvos nuvarande lösning var målsättningen att integrera samma batterityp i detta projekt.

Tabell 6.2.1: Specifikationer för Varta EasyBlade 48V-batteri

Parameter	Värde
Nominell spänning	51.8 V
Typisk kapacitet	29 Ah
Typisk energi	1502 Wh
Utladdningsström (nom.)	15.5 A
Max kontinuerlig utladdningsström	31 A
Dimensioner (L × B × H)	330 × 80 × 230 mm
Vikt	9.6 kg

För att möjliggöra integrationen av detta batteri utgjorde valet av motorer en central aspekt vid utformningen av elsystemet. Ett nyckelmoment var att välja en huvud-spänningsomvandlare

kapabel att distribuera den elektriska effekten på ett tillförlitligt sätt. Både de befintliga Simplex Motion-motorerna samt de önskade Direct Drive navmotorerna var konstruerade för att drivas med en nominell spänning på 24V. På grund av detta kvarstod beslutet att använda den befintliga spänningsomvandlaren för konvertering från 48V till 24V. Från denna 24V-grundförsörjning planerades flera sekundära spänningsomvandlingar för att uppfylla de olika delsystemens krav. En ny 24V–12V-spänningsomvandlare installerades för att förse Jetson-datorn med ström, vilken i sin tur försörjer den befintliga Arduino-enheten via USB. Dessutom installerades en 24V–5V-spänningsomvandlare för att försörja LED-belysningen i intentionssystemet.

Ett behov som tidigare saknades i robotsystemet var kommunikationen mellan robot och människa. Med fokus på användargränssnitt och säkerhet sattes därför målet att införa ett intentionssystem som tydligt visar robotens planerade färdväg. För att uppnå detta utvärderades olika koncept utifrån kompatibilitet med den befintliga robotplattformen. Valet föll på att använda LED-slingor (RGB) som den primära lösningen för visuell kommunikation med omgivningen. LED-slingor erbjuder flera tekniska fördelar som gör dem lämpade för detta ändamål. De är flexibla, lätta att integrera i robotens befintliga konstruktion och erbjuder ett brett spektrum av färger och ljusmönster, vilket underlättar tydlig och intuitiv signalering till omgivningen. Genom att kombinera ljus och olika typer av animationer kunde robotens rörelser uttryckas på ett sätt som är lättförståeligt.

6.3 Implementering av batteri

Föregående projektgrupp hade tillgång till ett batteri, men lyckades aldrig integrera det i systemet. Detta medförde att roboten enbart kunde köras via ett nätaggregat och därmed var beroende av konstant extern strömförsörjning. I denna iteration löstes problemet genom att batteriet framgångsrikt integrerades i robotens elsystem. Initialt var målet att köpa in ett batteri från tillverkaren VARTA, men på grund av bristande kommunikation från leverantörens sida kunde inget köp genomföras. Som alternativ lånades ett 48V VARTA EasyBlade litiumjonbatteri från AB Volvo i Tuve.

Integrationen av batteriet krävde en noggrann genomgång av tillverkarens tekniska dokumentation samt inköp av ett antal specifika komponenter. Två kommunikationskablar beställdes: en för laddning och en för CAN-bussanslutning till roboten. Eftersom CAN-kommunikation kräver korrekt terminering för att fungera utan extern styrning, installerades två 120 Ω termineringsmotstånd enligt specifikation. Dessutom krävdes en 0,6 Ω

resistor mellan batteriet och huvudspänningsomvandlaren för att möjliggöra uppstart, eftersom batteriets känsliga BMS inte tolererade störningar vid inkoppling. Batteriet monterades sedan centralt i roboten på den befintliga batterihållaren som tillverkades av förra årets grupp.

6.4 Intentionssystem

Eftersom en autonom mobil robot (AMR) opererar i miljöer där människor vistas, är det av stor vikt att kommunikationen mellan människa och robot är tydlig och intuitiv – något som benämns Intent Communication System (ICS). En sådan lösning är nödvändig för att skapa trygghet hos personalen och möjliggöra ett säkert och effektivt samspel mellan människa och robot. För att uppnå detta implementerades ett ICS i form av en LED-slinga monterad runt robotens chassi.

LED-slingan som valdes var Intelligent Flexible RGB Strip (RS Components, 2025). Denna specifika modell valdes på grund av dess höga flexibilitet samt kompatibilitet med Arduino-baserad styrning där flera ljusanimationer kunde programmeras. Kombinationen av tät lysdiodplacering och RGB-funktionalitet bidrog till tydliga ljusmönster och färgkodning som kunde anpassas för olika rörelser. Montering av LED-slingorna på roboten genomfördes med hjälp av 3D-utskrivna fästen som nämndes i 5.3. Eftersom LED-slingan får ström från en separat 24V-5V spänningsomvandlare anslöts LED-slingorna endast med en signalkabel till robotens Arduino, vilket möjliggjorde styrning av ljusmönster och färgkodning.

LED-slingan delades in i tre delar som har kopplats samman, den första och mest omfattande LED-slingan placerades på roboten utmed långsidorna och övre fronten. Denna är den mest synliga vilket gör att den utgör de flesta animationer. Acceleration som markeras med turkost ljus som börjar i mitten på fronten och åker på respektive sida ut mot kanten på långsidan och försvinner i flera sektioner, se bild (Figur 6.4.1). Den har även blinkers med gult blinkande ljus som visar vilken riktning den ska köra som aktiveras genom DualShock 4-kontroll. Utöver dessa finns kod för ljusanimationen för vänteläge genom att den blinkar gult och även hinder genom ett segment dioder i mitten på varje sida av LED-slingan lyser turkost men dessa har ännu ingen aktiveringsfunktion.

Den andra LED-slingan är monterad på nedre fronten och lyser kontinuerligt vitt se (Figur 6.4.1). Den visualiserar tydligt vilken riktning roboten kör och belyser ytan framför, vilket

även hjälper kamerans förmåga att detektera objekt även i svagt ljus. Den tredje LED-slingan är monterad baktill på den övre ramen och lyser kontinuerligt rött under drift. Även här är det för att visualisera vilken körriktning roboten har. Den fungerar även som bromsljus och lyser starkare rött vid inbromsning. I detta fall sker det vid retardation eftersom den inte har några bromsar då det inte krävs med den låga hastigheten roboten har vid drift. Tack vare att varje lysdiod kan adresseras individuellt är systemet modulärt och skalbart. Ytterligare LED-slingor kan integreras och nya animationer kan skapas med hjälp av kodning i Arduino. I tabell 6.4.1) visas alla funktioner, färg, placering och när den ska aktiveras vid full autonom körning.



Figur 6.4.1: Turkos accelerationsanimation längs sidorna och statiskt vitt ljus framåt

Tabell 6.4.1: Översikt av ljussignaler och deras aktivering

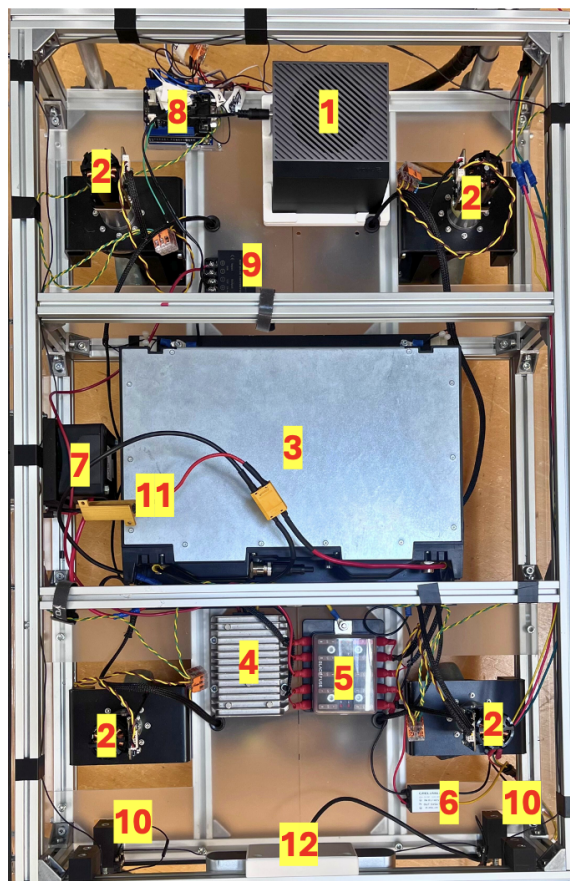
Funktion	Färg/animation	Placering	Aktiveras vid
Acceleration	Turkos segment – inåt	Övre ram, sidor	Hastighetsökning
Retardation	Turkos segment – utåt	Övre ram, sidor	Hastighetsminskning
Blinkers vänster	Blinkande gult	Främre + sidor + bak	Vänstersväng
Blinkers höger	Blinkande gult	Främre + sidor + bak	Högersväng
Framljus	Statiskt vitt	Nedre ram, främre vy	Alltid
Bakljus	Statiskt rött	Övre ram, bakre vy	Alltid
Bromsljus	Starkare rött	Övre ram, bakre vy	Inbromsning
Vänta	Blinkande gult	Runt övre ramen	Väntan på uppdrag
Hinder	Turkost segment	Sidor	Hinder detekterat

6.4.1 Implementering av tidigare utvecklade animationer

Utformningen av ljusanimationerna baserades delvis på ett tidigare kandidatarbete (Björnsdorff m. fl., 2019), vilket fokuserade på att ta fram färg- och ljusmönster för autonoma fordon med särskild hänsyn till tillgänglighet för personer med färgblindhet. Genom att utgå från detta arbete kunde utvecklingsprocessen effektiviseras, då en omfattande förundersökning redan fanns dokumenterad. Med tanke på det stora antalet möjliga situationer en autonom robot kan befinna sig i, gjordes ett urval av scenarier som ansågs mest relevanta för projektets syfte och tillgängliga funktionalitet.

6.5 Komponentplacering

I samband med att robotens funktionalitet utvecklades och fler tekniska komponenter integrerades, genomfördes flera systemanpassningar och visuella modifieringar. Dessa berörde både den interna strukturen, där elsystemets placering optimerades, och den yttre designen, där bland annat plexiglasets modifierades för att passa den nya hjulmodulen. Syftet med dessa förändringar var att förbättra funktionalitet, tillgänglighet och användarinteraktion utan att kompromissa med den tekniska prestandan. Figur 6.5.1 tillsammans med Tabell 6.5.1 visar hur dessa komponenter är fysiskt placerade i roboten.



Figur 6.5.1: Elsystemets placering

Tabell 6.5.1: Komponentförteckning till elsystemets placering, se Figur 6.5.1

Nr	Komponent
1	Jetson Orin dator
2	Styrmotorer
3	Batteri
4	Huvud-spänningsomvandlare (48V till 24V)
5	Säkringslåda
6	Spänningsomvandlare (24V till 5V)
7	Huvudbrytare
8	Arduino
9	Spänningsomvandlare (24V till 12V)
10	Brytare till stötfångare
11	Resistor
12	ZED-kamera

Implementeringen av flera nya elektroniska komponenter krävde en omfattande omstrukturering av robotens interna layout. Den nya, förstärkta hjulmodulen bidrog visserligen till en stabilare konstruktion, men upptog samtidigt mer utrymme än den tidigare lösningen, vilket minskade det tillgängliga utrymmet för övriga komponenter. Detta ställde höga krav på noggrann planering av komponentplacering för att minimera kabeldragning och säkerställa tillräckligt utrymme för kabelförläggning. Placeringen modellerades i *Fusion 360*, där de flesta komponenter kunde importeras som STEP-filer, vilket möjliggjorde hög precision i bedömningen av utrymmesbehov. Valet av placering styrdes både av fysiska begränsningar, behovet av enkel åtkomst för framtida underhåll och även en framtida implementering av en kontrollenhet för säkerhetsradar.

Komponentplacering: Främre sektion

I den främre delen av roboten monterades fyra huvudsakliga komponenter: ZED-kameran, säkringslådan, huvudspänningsomvandlaren (48V–24V) samt spänningsomvandlaren för LED-slingorna (24V–5V). Huvudspänningsomvandlaren och säkringslådan var sedan tidigare placerade i denna sektion, vilket underlättade integrationen av övriga komponenter genom korta kabelsträckor och god åtkomst. ZED-kameran tillkom i denna iteration av roboten och placerades fram till för att ge optimalt synfält.

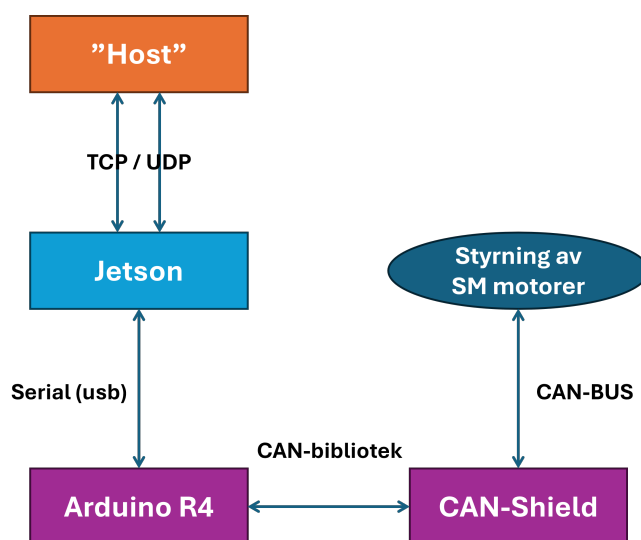
Komponentplacering: Bakre sektion

För att optimera kabeldragningen från handtagens knappkontroller, placerades Arduino-mikrokontrollern i den bakre delen av roboten, nära handtaget. Eftersom knapparnas signalledningar redan var dragna dit i föregående version, undveks onödig kabelomläggning.

Jetson-datorn placerades längst bak i roboten, en placering som valdes med hänsyn till både åtkomlighet och kylning. Eftersom det saknas plexiglas på baksidan kan användaren enkelt slå på, stänga av eller koppla ur datorn utan att behöva demontera det övre höljet. Den bakre placeringen underlättade även kommunikationskopplingen mellan Jetson-datorn och Arduino-enheten tack vare det korta avståndet mellan dem. I denna sektion installerades även spänningsomvandlaren (24V-12V) som förser Jetson-Orin-datorn med ström.

7 Datorprogram för styrning av robot

Systemets mjukvara är uppdelad i flera komponenter som körs på olika hårdvaruenheter. Figur 7.0.1 visar en översikt av hur dessa komponenter interagerar med varandra. Programmet på roboten består av en lågnivåstyrning på en Arduino, som kommunicerar med motorer och sensorer via CAN och seriell kommunikation. Jetson-datorn användes för högnivåstyrning, där styrsignaler mottas och vidareförmedlas. Dessutom finns ett externt styrprogram som körs på en värdenhet (HOST) med ansluten kontroll, vilket möjliggör manuell styrning över WiFi.

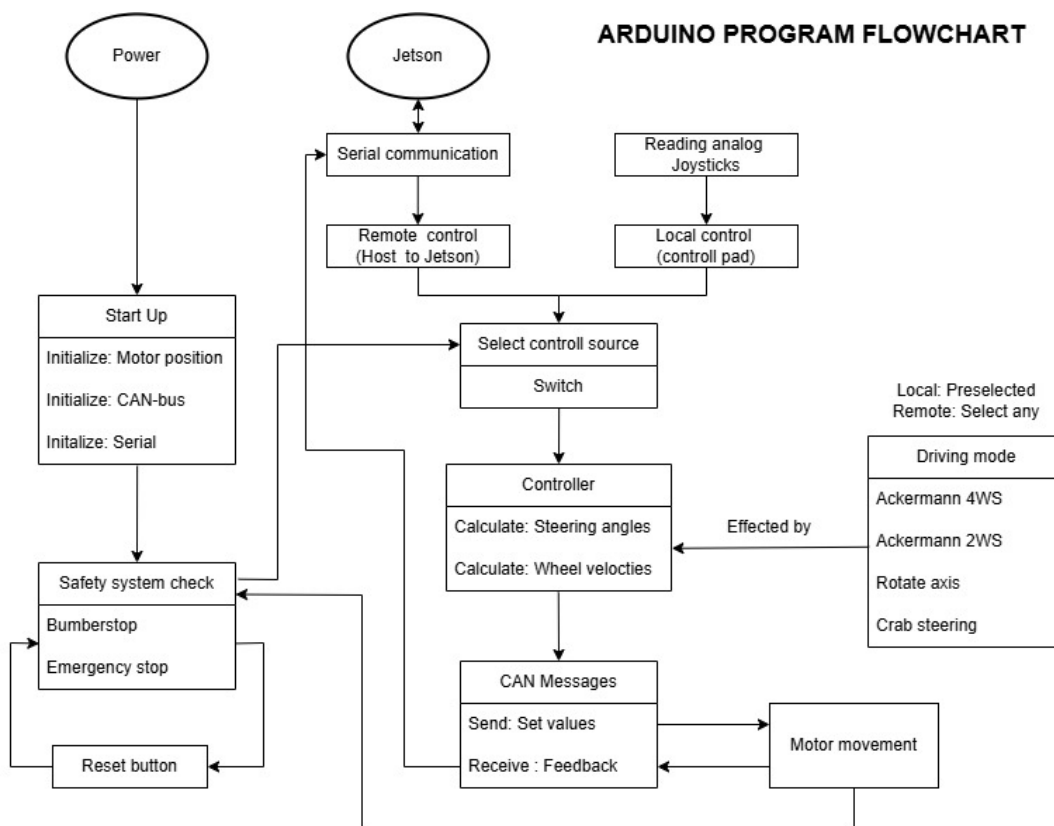


Figur 7.0.1: En överblick på hur olika komponenter och programvaror interagerar med varandra.

All kod som utvecklats inom projektet är tillgänglig via vårt offentliga GitHub-repository (Alexandersson m. fl., 2025). Repositoriet innehåller all relevant kod och dokumentation.

7.1 Lågnivåstyrning

Arduinon, tillsammans med den tillhörande CAN-skölden, ansvarar för robotens lågnivåstyrning. Alla motorer styrs via CAN-bussen, intentionssystemets LED-slingor kontrolleras, stötfångarna övervakas, samt analoga signaler från joysticks läses in. Ett överskådligt flödesschema över Arduinons programstruktur presenteras i Figur 7.1.1.



Figur 7.1.1: Flödesschema över programmet som körs på Arduinon på roboten.

Nedan följer viktiga nyckelförklaringar för några funktioner och klasser:

CANMessages För att kommunicera över CAN-bussen användes ett bibliotek vid namn Seeed Arduino CAN-BUS library (GitHub contributors, 2022) som fungerar ihop med CAN-skölden. Alla meddelanden formateras enligt SM-CAN-protokoll, vilket var speciellt för just Simplex motion-motorerna..

Funktionen `CAN.sendMessage(id, ext, rtrBit, len, buf)` användes för att skicka meddelanden över CAN-bussen. Parametern `id` syftar på motorns CAN-ID i kom-

ination med ett register enligt SM-CAN-protokollet. `ext` anger om meddelandet skickas som en extended frame (29-bitars ID, `ext = 1`) eller standard frame (11-bitars ID, `ext = 0`). `rtrBit` bestämmer om det är en läsförfrågan (1) eller en skrivning (0) till det angivna registret. `len`, ofta kallad DLC (Data Length Code), anger hur många byte som ska skickas eller tas emot, och `buf` innehåller själva datan som överförs. Exempel på användning:

```
CAN.sendMessageBuf(0x501C2, 1, 0, 2, {0xVAL, 0xVAL})
```

 skickar ett CAN-meddelande till ID `0x501C2` (motsvarar motor med id (HEX) 50 och register `1C2` -> vinkelstyrning) med extended frame (1), skrivning (0), två databyte (`len = 2`) med innehållet `0xVAL, 0xVAL`.

Funktionen `CAN.readMsgBufID(id, ext, rtr, len, buf)` användes för att motta inkommande meddelanden från Simplex motion-motorerna (feedback). Alla beteckningar är samma som ovan.

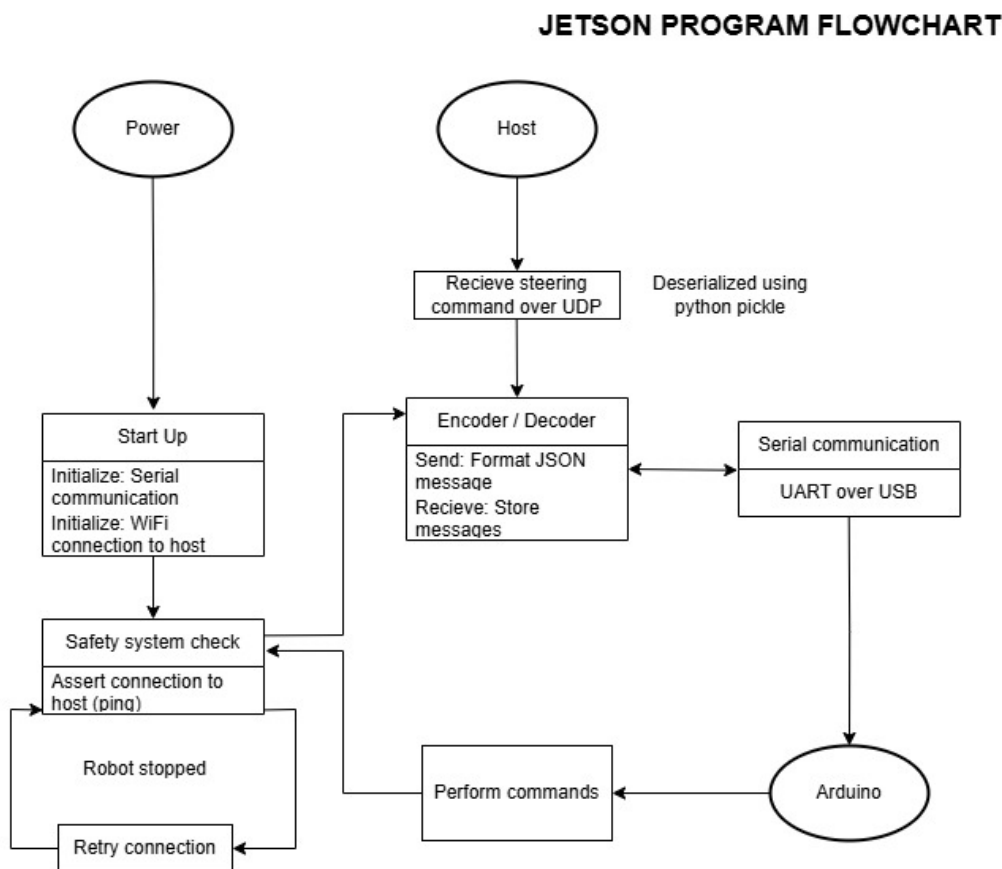
Controllers För att beräkna styrvinklar och rotationshastigheter baserat på börvärden skapades en klass för styrning. Denna klass bygger på de ekvationer som presenteras i kapitel 3 för Ackermannstyrning, krabbstyrning och rotation kring egen axel.

Serial Communication Den inbyggda `Serial`-funktionen från Arduinos standardbibliotek används för att hantera all seriell kommunikation över USB, tillsammans med ett JSON-bibliotek (GitHub contributors, 2025b). All kommunikation behandlas här, och majoriteten av alla meddelanden serialiseras med hjälp av JSON-biblioteket, både när roboten är i fjärrstyrt och lokalt styrt läge. En hög `baudrate` på 500000 används för att undvika flaskhalsar vid hög trafikmängd.

Intentionssystem Klassen och tillhörande funktioner för kontroll av intentionssystemet finns inte med i flödesschemat, då en fullständig implementering aldrig färdigställdes. Alla animationer är dock förberedda i klassen `AnimationManager`, där styrningen av de adresserbara lysdioderna utförs med `Adafruit NeoPixel Library` (GitHub contributors, 2025a). Animationerna kan styras manuellt, och visst arbete påbörjades för att möjliggöra automatisk uppspelning baserat på återkoppling från hjulens positionsgivare.

7.2 Högnivåstyrning

För implementering av högre logik såsom objektigenkänning och banplanering krävdes en kraftfull högnivå-styrenhet, vilket motiverade användningen av Jetson-datorn. Den nuvarande implementationen är dock begränsad till att ta emot och vidarebefordra styrsignaler, då det inte fanns tillräcklig tid för en fullständig implementation inom ramen för projektet. Ett flödesschema över programmet kan ses i Figur 7.2.1. Programmet startas automatiskt via en systemtjänst, vilket konfigureras på operativsystems-nivå.



Figur 7.2.1: Flödesschema över programmet som körs på Jetson-datorn.

Nedan följer viktiga nyckelförklaringar för några funktioner och klasser:

WiFi-kommunikation För att snabbt skicka styrsignaler över WiFi används UDP-protokollet, implementerat med hjälp av Pythons standardbibliotek `socket`. För att underlätta kommunikationen och få ökad kontroll användes en separat router, där statiska IP-adresser sattes för Jetson-datorn och en laptop (benämnd "HOST" i flödesschemat). Detta gjordes

för att undvika problem med föränderliga dynamiska IP-adresser på eduroam.

När programmet körs övervakas anslutningen kontinuerligt via det mer pålitliga TCP-protokollet. Om en fränkoppling upptäcks stoppas roboten automatiskt inom loppet av några hundra sekunder. Hur ofta anslutningen kontrolleras kan regleras.

Encoder / Decoder Klasserna ansvarar för att formatera och spara meddelanden som skickas mellan Arduinon och datorn. Som nämnts tidigare kommuniceras all styrdata i JSON-format, vilket hanteras av funktioner inom `Encoder`-klassen. Dessa formaterar meddelanden med hjälp av Pythons standardbibliotek `JSON`. På motsvarande sätt avkodas JSON-meddelanden av `Decoder`-klassen, där fördefinierade meddelandetyper sparas i en dictionary som alltid innehåller den senaste versionen.

Serial communication Klassen som ansvarar för seriell kommunikation använder sig av `pyserial`. Meddelanden från Arduinon kan avläsas kontinuerligt i en separat tråd i bakgrunden, och dessa vidarebefordras sedan till en instans av `decoder`-objektet som tidigare diskuterats. För att säkerställa att Arduinon inte blir överbelastad med meddelanden implementerades även en enkel handshake-funktion, där Arduinon flaggar för att data processas. Under denna tid skickas inga meddelanden. En maximal sändningsfrekvens implementerades också (båda dessa går att åsidosätta vid programmering). Genom tester uppskattades att en maximal kontinuerlig hastighet på 30 Hz var möjlig utan problem.

På grund av misstänkta svårigheter med `pyserial` och Linux-drivrutiner användes en USB-till-seriell UART-konverterare istället för den inbyggda USB-C-porten på Arduinon. Den exakta anledningen är okänd, men felet upprepades på flera Linux-datorer medan det fungerade på Windows.

7.3 Sändning av styrsignaler

Som tidigare diskuterats tar Jetson-datorn emot styrsignaler från en "Host" över WiFi via UDP-protokollet. Inom ramen för projektet agerar en laptop med en ansluten spelkontroll (i detta fall en PS4-kontroll) som värdenhet. Kontrollen programmeras med hjälp av en `inputs`-modul och Figur 7.3.1 illustrerar styruplägget.



Figur 7.3.1: Layout för PS4-kontrollen som användes för trådlös styrning av roboten (eve-resd_design, 2020).

Det är möjligt att växla mellan olika styrlägen via knappsatsen till höger, spela upp fördefinierade animationer med styrkorset (D-pad), justera hastighet och styrvinkel med joysticker samt aktivera blinkers.

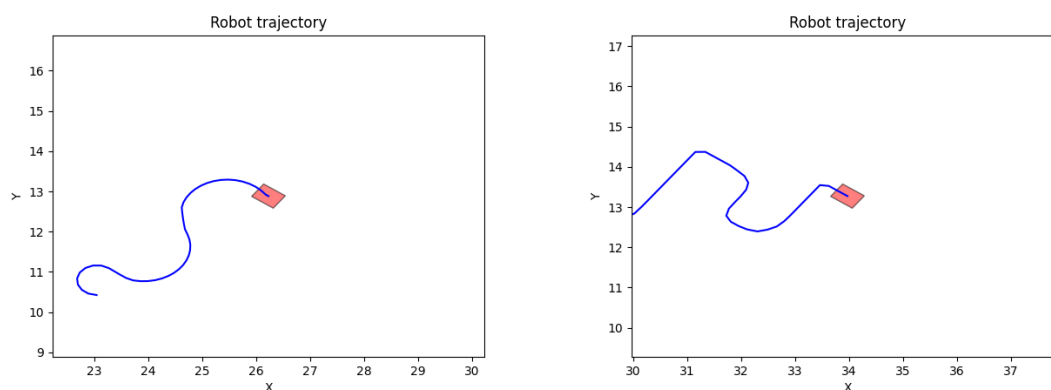
8 Testning

För att utvärdera och testa funktionaliteten och robusthet har flera tester genomförts, både i simulerad miljö och verklig drift. Testningen syftade till att undersöka hur väl robotens mekaniska, elektroniska och styrtekniska delsystem samverkar.

För att säkerställa funktionerna genomfördes en uppsättning av kvalitativa tester där olika manöverfunktioner och intentionsfunktioner observerades under drift. Dessa tester syftade främst till att verifiera att robotens manövreringsförmåga, återkoppling från hjulen och visuella intentionssystem fungerade som avsett.

8.1 Simulering av den kinematiska modellen

För att säkerställa att den kinematiska modellen ger förväntade resultat på den riktiga roboten, genomfördes först en preliminär simulering. Ett Python-script skapades där en spelkontroll (DualShock 4) användes som substitut för feedback från robotens hjul. En Robot-klass och en Model-klass utvecklades baserade på ekvationerna 4.2.1-4.2.5, där tidssteget dt fick variera i takt med simuleringen. Resultatet för tre av de fyra styrmoderna visas i Figur 8.1.1.



(a) Simulering av robotens rörelse för 2WS och 4WS ackermann. Simuleringen börjar med 4WS och avslutas med 2WS. (b) Simulering av krabbstyrning för roboten. Som väntat rör sig roboten i raka linjer utan att ändra styrvinkel.

Figur 8.1.1

Resultaten uppvisar rimliga rörelsemönster för roboten. I Figur 8.1.1a inleds robotens bana (markerad i blått) i 4WS-läget, vilket resulterar i betydligt skarpare svängradier jämfört med 2WS-läget som används mot slutet. Modellen simulerar även krabbstyrning på ett trovärdigt sätt, se Figur 8.1.1b.

8.2 Kvalitativ Verifiering av Systemfunktionalitet

För att säkerställa att robotens nya system fungerar som förväntat, samt att den kan styras både manuellt och centralt, kommer ett antal kvalitativa tester att genomföras. Syftet med dessa tester är inte att kvantitativt utvärdera systemens prestanda, utan snarare att verifiera att de fungerar som avsett.

I Figur 8.3.1 illustreras ett antal scenarier med successivt ökande komplexitet för robotens manövrering. Roboten ska i dessa scenarier kunna:

1. Styras manuellt via joysticks.
2. Styras centralt över WiFi.
3. Rapportera återkoppling från hjulen.
4. Estimera sin position med hjälp av odometri.
5. Upptäcka hinder och objekt i miljön med hjälp av sin kamera.
6. Visa att LED-baserade intentionssystemet fungerar (dock utan att genomgå rigorösa tester; dessa moment utelämnas).

8.3 Tester för styrning

För att kunna bedöma hur väl fyrhjulsstyrningen fungerade så behövde tester genomföras. Fokuset med dessa var främst att kunna hitta fall där styrningen kunde utvecklas mer. Det första testet testade främst de drivande motorerna då ingen styrning krävdes alls (se Figur 8.3.1a).

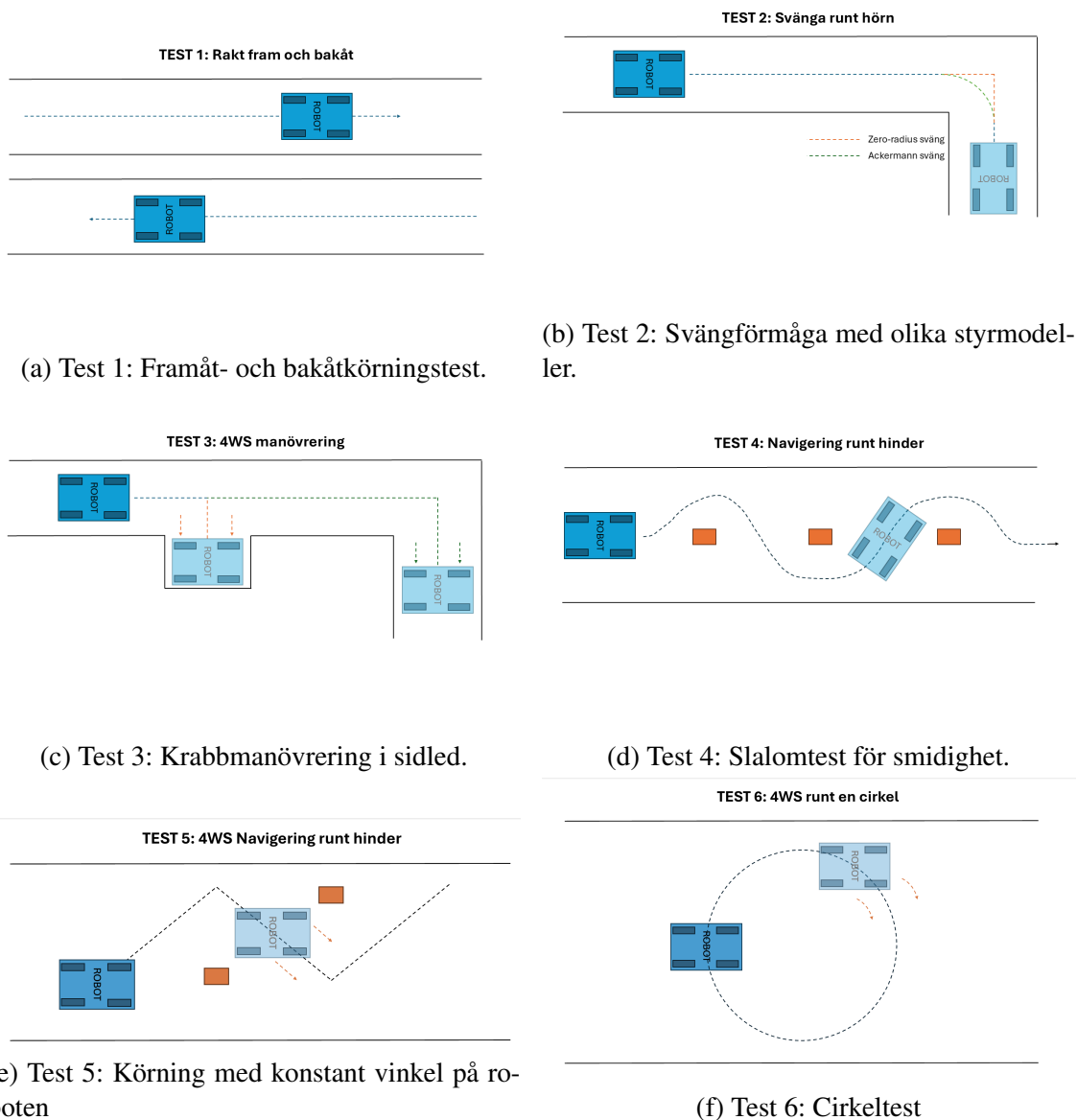
Det andra testet (se Figur 8.3.1b) undersökte hur väl roboten kunde rotera kring sin egen axel. Detta möjliggjorde skarpare svängar vid trånga utrymmen vilket var ett stort fokus under projektet.

Tredje testets fokus var att få roboten att kunna fickparkera genom att vända på alla hjul

samtidigt och krabbstyra in på en parkeringsficka (se Figur 8.3.1c). I trånga utrymmen är detta en stor fördel, då roboten inte behöver någon svängradie för att parkera. Eftersom alla fyra hjul kan styras individuellt kan roboten enkelt manövreras även i begränsade utrymmen.

I test fyra (se Figur 8.3.1d) utvärderades robotens förmåga att undvika hinder i trånga miljöer genom ett slalomtest. Eftersom robotens hjul och front inte behöver ha samma riktning, användes test 5 (se Figur 8.3.1e) för att undersöka robotens möjlighet att bibehålla en konstant orientering medan hjulen rör sig i annan riktning.

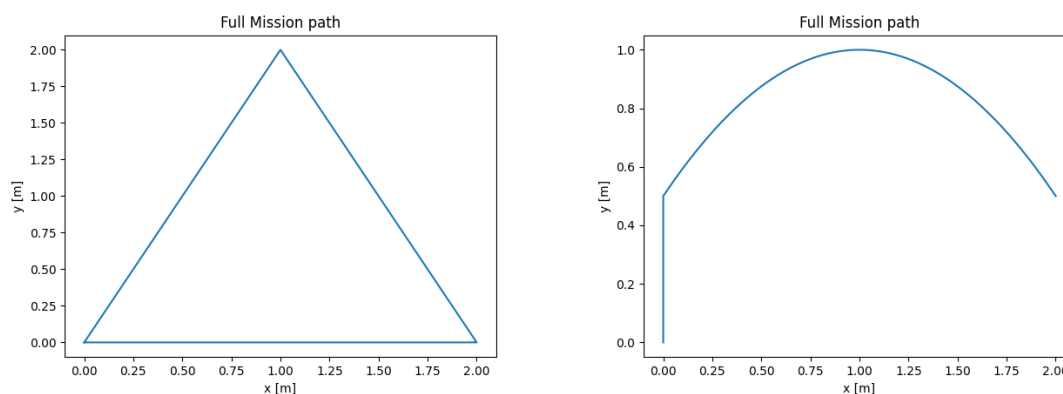
Det sjätte testet (se Figur 8.3.1f) fokuserade på robotens förmåga att utföra en vändning med noll svängradie, det vill säga en rotation kring sin egen axel. Denna manöver är särskilt användbar vid begränsade utrymmen där vanliga svängrörelser inte är möjliga, och demonstrerar robotens maximala manövrerbarhet genom simultan styrning och rotation av hjulen.



Figur 8.3.1: Översikt av tester för 4WS 4WD-robotens manövreringsförmåga.

8.4 Kvantitativt test av odometri och precision

För att utvärdera robotens odometri och precision planeras två enkla banor att följas, se Figur 8.4.1. Båda testerna utförs i både 2WS/4WS ackermann-läget och med krabbstyrning. Svängning runt egen axel tillämpas också mellan linjesegmenten.



- (a) Roboten körs i en triangel. Vid varje hörn stannar roboten och vrider sig för att hålla sig parallell med banan.
- (b) Roboten tar en mjuk kurva. Första segmentet innan av banan är för att underlätta placeringen av roboten mellan upprepningar av testet.

Figur 8.4.1: Två tester som utförs för att utvärdera roboten odometri.

Testet kommer att utföras med följande metod:

1. Robotens bana körs tio gånger. Efter varje körning markeras slutpositionen med ett 'X', och dess (x,y) -koordinater i förhållande till den förväntade målpunkten mäts. Efter varje körning återställs roboten manuellt till startpositionen.
2. Medelvärdet av de uppmätta slutpositionerna, \bar{x} och \bar{y} , beräknas och jämförs med målpunkten för att kvantifiera noggrannheten i robotens rörelse. Dessutom beräknas felet mellan robotens skattade position, baserad på odometri, och dess faktiska position för att analysera odometrins noggrannhet. Felet definieras som:

$$e_i = \sqrt{(x_{\text{odo},i} - x_{\text{verklig},i})^2 + (y_{\text{odo},i} - y_{\text{verklig},i})^2}$$

där $x_{\text{odo},i}$ och $y_{\text{odo},i}$ är den av odometrin skattade positionen, och $x_{\text{verklig},i}$, $y_{\text{verklig},i}$ är den verkliga positionen vid den i :te körningen.

3. För att kvantifiera spridningen av slutpositionerna beräknas kovariansmatrisen:

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

där σ_x^2 och σ_y^2 är variansen i x - och y -led, medan σ_{xy} anger kovariansen mellan de två variablerna. Kovariansmatrisen ger ett mått på osäkerheten i rörelsen och är direkt kopplad till systemets precision och repeterbarhet.

9 Resultat

Majoriteten av de planerade testerna kunde genomföras med manuell styrning. Dock hann den autonoma körningen inte att färdigställas inom projekttiden, men simuleringar genomfördes tidigare i arbetet. Visionssystemet implementerades, inklusive en AI-modell för objektigenkänning. Även åtta olika ljusanimationer implementerades till intentionssystemet. Den slutgiltiga konstruktionen av roboten kan ses i Figur 9.0.1.



(a) Roboten sedd bakifrån.

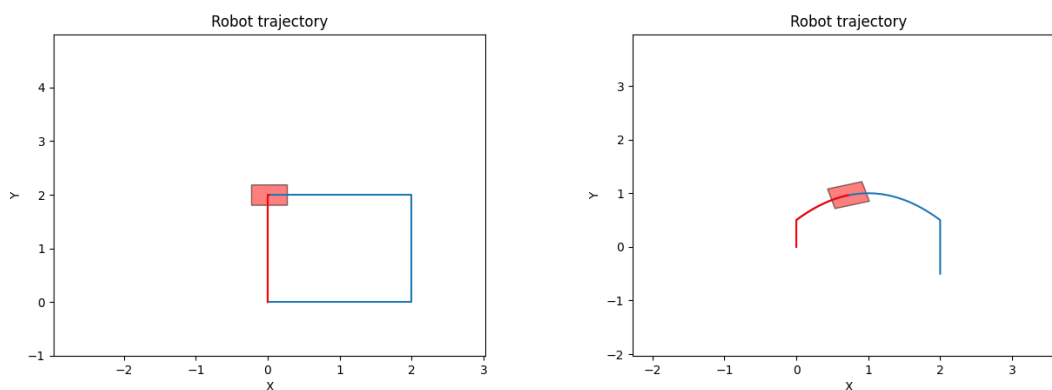


(b) Roboten sedd framifrån.

Figur 9.0.1: Nya iterationen av roboten

9.1 Simulering av grundläggande banföljning

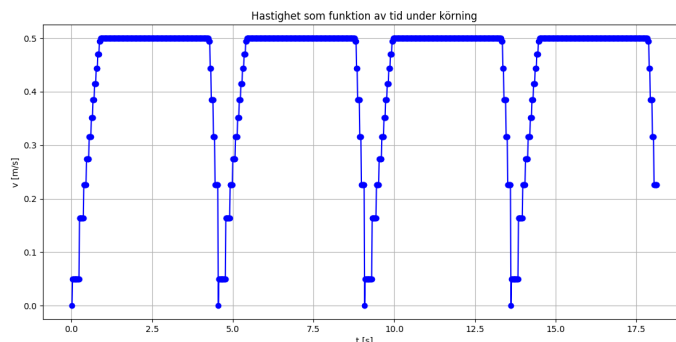
Som ett steg i arbetet med att få roboten att följa förutbestämda banor utvecklades en enklare simulering. Figur 9.1.1 visar hur roboten simulerat följer två olika banor. Simuleringen bygger på den tidigare presenterade kinematiska modellen i sektion 8.1. Återkopplingen i simuleringen tillämpas antingen med en *Pure Pursuit*-algoritm eller en PID-regulator för styrvinkeln. Hastigheten bestäms som en funktion av var den simulerade roboten befinner sig längs banan. Allt detta har tidigare diskuterats mer ingående i kapitel 4.



(a) Simulering där roboten följer en kvadratisk bana. (b) Simulering där roboten följer en bana liknande den i Figur 8.4.1b.

Figur 9.1.1: Simulerad banföljning för roboten.

Banor definieras som styckvis sammanhängande kontinuerliga segment, bestående av kurvor och linjer. Robotens position projiceras relativt banan, och med denna information kan hastigheten anpassas efter maxhastighet, acceleration och inbromsning. Ett exempel på den resulterande simulerade hastighetsprofilen visas i Figur 9.1.2 för banan i Figur 9.2.1a.



Figur 9.1.2: Simulerad hastighet som funktion av tiden under körning.

Vid en framtida implementation på den fysiska roboten var tanken att den kinematiska simuleringen skulle ersättas med återkoppling från antingen kamerabaserad SLAM eller från robotens motorpositionsgivare. Mer om detta samt återstående arbete diskuteras i kapitel 10.

9.2 Tester

Roboten kunde inte köras autonomt från punkt A till B autonomt. Däremot blev det möjligt att styra roboten trådlöst med hjälp av en DualShock 4-kontroll. Med hjälp av denna kunde olika tester göras från delkapitel 9.2.1 för att testa styrningen på roboten. De styrslägen som implementerades var framhjulsstyrning, fyrhjulsstyrning och krabbstyrning. Alla tester som sattes upp kunde slutföras utom testnummer sex vilket var ett cirkeltest, se figur 8.3.1f. Syftet med detta test var att få roboten att köra kontinuerligt i en cirkel. Detta uppnåddes inte då hjulen behövde byta körriktning efter ett kvarts varv. På grund av styrstoppet hade inte robotens hjul möjlighet att rotera 360° .



(a) Roboten lastas med två pallar och körs med styrspakar på handtaget



(b) Test för fickparkering utförs med trådlös styrning.

Figur 9.2.1: Tester för manövrering utförs på roboten. Figuren visar två av testerna.

9.3 Implementation av objektigenkänning

Objektigenkänningen som programmerades kunde inte integreras ihop med styrkoden. Detta var främst på grund av tidsbrist och prioritering av vilka delmoment som var viktigast för ett fungerande resultat. Kommunikationen mellan Arduinon och Jetson-datorn.

9.4 Intentionssystem

Intentionssystem implementerades till roboten till viss del. Alla animationer som programmerades lades inte till i koden. Detta berodde mycket på tidsbrist då varje funktion behövde testas löpande och att varje funktion behövde prioriteras på rätt sätt. Roboten kunde inte ha somliga funktioner, exempelvis objekt i vägen. Detta för att objektigenkänningen inte implementerades tillsammans med körningen. De animationer som lades till i körningen var statiskt framljus, statiskt bakljus, acceleration, retardation, bromsljus och blinkers (höger och vänster). Alla animationer kördes parallellt med styrkoden förutom blinkers, denna aktiverades med knapptryck på DualShock-kontrollern.

9.5 Mekanisk konstruktion

Efter att konstruktionen färdigställts testades roboten med en extern belastning på 500 N, utöver sin egenvikt på 300 N. Belastningen påverkade inte robotens prestanda, vilket överensstämmer med resultaten från de genomförda FEM-analyserna. Ingen märkbar förändring observerades i vare sig styrförmåga eller hastighet.

10 Diskussion

Gruppen är sammanfattningsvis nöjda med resultatet men det stora genomgående problemet i projektet var problem med beställningar och leveranser. Detta bidrog till att gruppen inte hann klart med alla delar som var tänkta. Gruppen har prioriterat att ge en kortare startsträcka till vidareutveckling av roboten. Många av delarna som inte avklarades har förberetts och behöver endast kombineras med de redan fungerande funktionerna.

10.1 Hjulmodul

Utvecklingen av den nya hjulmodulen har resulterat i en mer robust och hållfast lösning jämfört med tidigare konstruktioner. Genom att analysera strukturella svagheter i föregående års design har den mekaniska integriteten förbättrats avsevärt. Införandet av dubbelsidig infästning har eliminerat det böjmoment som tidigare uppstod vid ensidig infästning. Detta har lett till en tydlig ökning av stabilitet, vilket även bekräftas av de genomförda FEM-analyserna.

En av de mest centrala förbättringarna var ersättningen av det tidigare vridbeslaget med ett kryssrullager. Detta förbättrade både styvheten och styrprecisionen i hjulmodulen. Det lagerglapp som tidigare utgjorde ett betydande problem har i stort sett eliminerats.

Ett avgörande hinder i projektet uppstod dock när de planerade navmotorerna, kring vilka hela hjulmodulen ursprungligen konstruerats, inte kunde levereras i tid. Detta påverkade både den praktiska implementeringen och utvärderingen av det tänkta slutkonceptet. Gruppen tvingades därför återanvända motorer och drivhjul från tidigare år, vilket innebar att den nykonstruerade underdelen behövde anpassas till andra infästningar och hjulgeometrier.

Den 3D-utskrivna underdelen, som tillverkades i PLA, var en nödvändig lösning med hänsyn till tidsramen, men är inte representativ för den färdiga konstruktionens prestanda. PLA har väsentligt lägre hållfasthet än stål, vilket innebär att resultaten från praktiska tester inte fullt ut speglar den avsedda konstruktionens kapacitet. Det är därför viktigt att en framtida projektgrupp, vid tillgång till navmotorerna, slutför installationen enligt den ursprungliga designen i stål för att möjliggöra en korrekt utvärdering.

De växellådor som används till samtliga motorer uppvisar också betydande glapp. Även om detta har mindre påverkan på de drivande motorerna, är det särskilt problematiskt för styrmotorerna, där hög precision krävs. Med ett maximalt motormoment på 0,5 Nm är den nuvarande växellådan med utväxling 1:16 överdimensionerad. En växellåda med mindre glapp och eventuellt lägre utväxling bör därför övervägas i framtida iterationer.

10.2 Elsystem

En av de största förbättringarna jämfört med föregående års arbete var den framgångsrika integrationen av ett batteri, vilket gjorde roboten oberoende av extern strömförsörjning. Detta innebar ett viktigt steg mot verklig autonomi.

Gruppen hade stora problem att få tag i ett batteri som mötte kraven för roboten. Problemet löstes genom kontakt med AB Volvo, där Kristoffer Bengtsson gav möjlighet att låna ett batteri (K. Bengtsson, personlig kommunikation, 2025). Batteriet som lånades och användes i projektet var ett Varta EasyBlade 48V-batteri.

För att få systemet att samspela med batteriet behövde ändringar göras på robotens elsystem. Efter noggrannare undersökning kunde gruppen se att det uppstod en kort strömspik precis när batteriet startade. Detta gjorde att batteriets överströmsskydd löste ut och batteriets indikator började lysa fast rött. Enligt uppgift från Bengtsson används en serieresistor i AB Volvos interna system för att dämpa sådana strömspikar. Genom att implementera en liknande lösning, det vill säga en serieresistor på 0.6 Ω , kunde problemet lösas, vilket fick batteriet att fungera och möjliggjorde tester av roboten utan extern strömförsörjning. Däremot är denna lösning inte optimal, eftersom resistorn försämrar robotens drifttid något. Idealt bör ett relä användas för att koppla förbi detta motstånd efter uppstart, vilket är den lösning som AB Volvos egna robotar använder. Den tidsfördröjning som orsakades på grund av denna tekniska utmaning resulterade i att testfasen i projektet blev kortare än förväntat.

En enkel förbättring till nästa iteration av projektet är att byta ut batteriet till ett 24 V-batteri istället. Detta används dock inte för närvarande av AB Volvo. Med denna förbättring skulle även 48–24 V-omvandlaren kunna tas bort, vilket hade förenklat robotens elsystem. I övrigt anses elsystemet vara välfungerande och inga andra problem uppstod under arbetet.

10.3 Datorprogram för styrning av robot

Stora delar av programmeringen som återstår rör den slutgiltiga integrationen och implementeringen av de olika delsystemen. Flera funktionella komponenter har utvecklats separat, men har ännu inte integrerats i det huvudsakliga styrsystemet. Bland annat:

- Objektigenkänning fungerar i tester men är ännu inte kopplad till robotens beslutslogik, se figur 2.3.1.
- SLAM-algoritmen från ZED-kameran genererar en tillsynes korrekt positionsuppskattning, men används inte ännu som återkoppling till robotens rörelse i banföljningen.

De planerade testerna för banföljningen kunde inte genomföras, vilket innebär att den exakta positioneringsprecisionen med hjälp av kamera (SLAM) och odometri förblir okänd. En noggrann utvärdering av dessa delsystem bör därför prioriteras tidigt i nästa iteration av projektet.

- En kinematisk simulering av banföljning har utvecklats, men behöver kopplas till faktisk sensorfeedback (kameran och motorer) och motorstyrning. Funktioner i koden existerar för återkoppling av motorer.
- Ett omfattande arbete med intentionssystemets animationer är genomfört, men en automatisk uppspelningslogik baserat på robotens tillstånd är ännu inte färdigställd. Även integration med kameran är ej påbörjat.

Ytterligare ett område där arbete återstår är integreringen av banföljning med planering av intentionssystemets animationer. Exempelvis är det önskvärt att roboten aktiverar blinkers i god tid innan en sväng påbörjas, vilket kräver att körplanen analyseras i förväg och att animationer tidsmässigt synkroniseras med planerade manövrar. Någon implementation eller logik för detta har ännu inte påbörjats.

Sammanfattningsvis bör arbeten med kommande iterationer av roboten fokusera på att sammanföra dessa delar till ett helhetsfungerande system.

10.4 Objektigenkänning

Eftersom roboten inte kunde drivas med batteri förrän mot slutet av projektet prioriterades batteriintegrationen och en fungerande manuell styrning framför samverkan mellan styrkod och objektigenkänning. Gruppen lyckades implementera en grundläggande, men välfungerande, objektigenkänning. Däremot visade det sig vara mer utmanande än förväntat att få Arduinon att ta emot och tolka data från Jetson-datorn för att roboten skulle kunna agera utifrån bildinformationen. Denna integration var planerad som nästa steg i utvecklingen, men stabil styrning prioriterades i det fortsatta arbetet.

10.5 Styrgeometri

I projektet lades två nya styrlägen till (2WS Ackerman och rotation kring sin egen axel), och koden är strukturerad på ett sätt som gör det enkelt att lägga till ytterligare lägen vid behov. Däremot har användarupplevelsen vid manuell styrning inte analyserats djupare. I nuläget regleras styrvinkeln med vänster joystick och hastigheten med höger. Det finns dock stor potential att undersöka alternativa metoder för användarinteraktion, där exempelvis olika kombinationer av joysticks eller knappfunktioner skulle kunna förbättra körupplevelsen.

Ett annat observerat problem är att motorerna för styrning inte är helt synkroniserade. I de båda Ackermann-lägena innebär detta att det inre hjulet – som svänger mindre – når sitt målvärde tidigare än det yttre, vilket skapar en kortvarig asymmetri i hjulens vinkelpositioner. Även om detta inte är ett allvarligt problem vid normala rörelser, blir det mer påtagligt vid skarpa svängar.

Detta skulle kunna förbättras genom mjukvarulösningar, antingen genom mer aggressiv reglering av styrmotorernas position för snabbare återkoppling, eller genom att programmera efterlikna ett mekaniskt länkage som i ett traditionellt styrsystem. En sådan lösning skulle också kunna anpassas beroende på valt styrläge för att uppnå mer realistiskt eller önskvärt styrbeteende.

10.6 Vidareutveckling

Detta projekt har gått ut på att göra roboten stabilare genom ny hjulmodul, implementera objektigenkänning, införa ett intentionssystem och fixa styrning för roboten. Det finns fortfarande flera vidareutvecklingar som går att implementera för att uppnå en mer robust robot som kan köras autonomt.

Hjulmodulen använder sig fortfarande av Simplex-Motion-motorer som sticker ut en del från hjulmodulens underdel, detta vill undvikas genom att byta ut dessa motorer och drivhjul mot hjul med integrerade navmotorer. Ett däck har redan genom 3D-utskrivning i TPU tillverkats till navmotorn men kunde aldrig implementeras på grund av att navmotorn inte beställdes. Däcket var halkigt men om detta anses värt att vidareutveckla så finns greppsprej för att göra TPU mer gummiaktig och kan därmed få bättre grepp.

För objektigenkänning används enbart en stereokamera, vilket inte är tillräckligt för att uppfylla krav för säkerhetsklassning. Detta kan åtgärdas genom att komplettera systemet med en säkerhetsklassad radar, vilket skulle bidra till att uppnå en högre grad av säkerhet i robotens drift. Detta var tanken från början men på grund av tidsbristen blev det inte aktuellt. Att implementera det för nästa iteration av roboten är däremot ett bra alternativ.

Styrningen är i dagsläget enbart manuell och sker antingen via robotens styrspakar eller med en DualShock 4-kontroll. Fyra styrlägen har implementerats: tvåhjulstyrning (2WS) enligt Ackermannprincipen, fyrehjulstyrning (4WS) enligt Ackermann, rotation kring robotens egen axel samt krabbstyrning.

Det finns för närvarande ingen dedikerad knapp på roboten för att växla mellan dessa lägen. Växling sker antingen genom att ladda upp ny kod eller via knappar på PlayStation-kontrollen. En möjlig vidareutveckling är att införa ett fysiskt gränssnitt för detta, exempelvis en fyrvägsswitch.

Fullt autonom styrning hann inte implementeras inom projektets tidsram, vilket innebär att även detta är ett naturligt utvecklingsområde för framtida arbete.

Även intentionssystemet har potential för vidareutveckling. De befintliga ljusanimationerna kan finjusteras för att förbättra tydlighet och funktion. Dessutom är inte samtliga animationer fullt implementerade, eftersom vissa kräver autonom körning och en fördefinierad bana för att kunna aktiveras på rätt sätt.

Vid körning finns i nuläget ingen möjlighet att övervaka batteriets status, såsom laddningsnivå eller temperatur. En möjlig förbättring vore att installera en display på roboten som visar batteriprocent och uppskattad återstående drifttid.

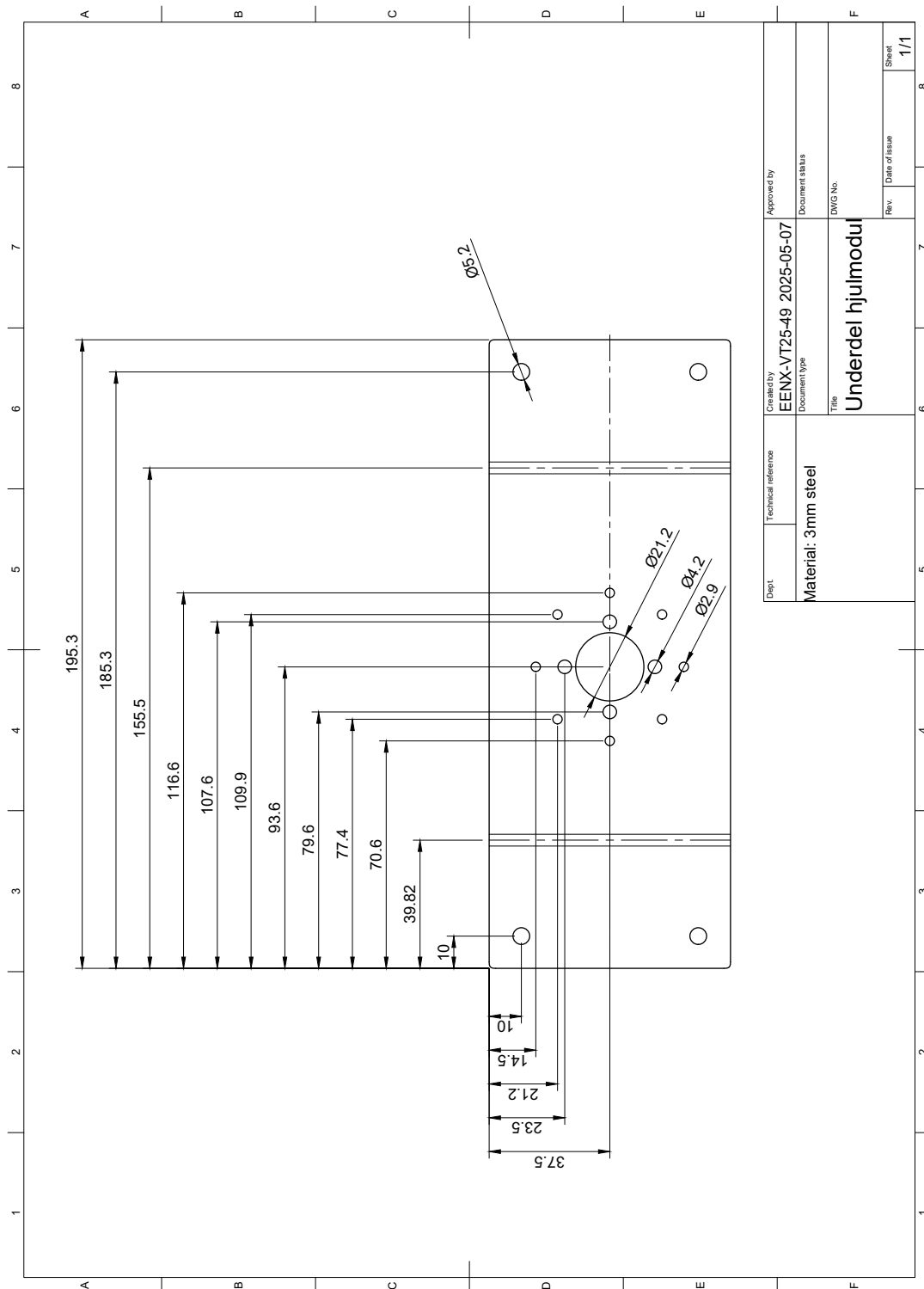
Litteratur

- Ahmed. (2023). Rotary Encoders for Odometry. *Medium*.
- Alexandersson, V., Grönlund, E., Johansson, F., Karacan, H., Persson, O., & Wallsten, V. (2025). Kandidat-arbete-grupp49. <https://github.com/Vilgotal/Kandidatarbete-grupp-49.git>
- Aqel, M., Marhaban, M., & Saripan, M. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* 5.
- Arduino. (2025). *UNO R4 WiFi*.
- Billing, K., Börjesson, Z., Enocson, G., Jabraouti, M., Janson, G., & Nilsson, O. (2024). *Design och konstruktion av en autonom mobil robot* (Technical Report). Institutionen för elektroteknik, Chalmers Tekniska Högskola. Göteborg, Sverige.
- Bjöersdorff, W., Tidblom, J. B., Göthberg, C., Nilsson, M., Nordin, C., & Rooth, L. (2019). *Design av interaktion för ett autonomt fordon: Med avseende på interntransport i fabriksområde* (Kandidatarbete inom Teknisk Design). Institutionen för industri- och materialvetenskap, Chalmers Tekniska Högskola. Göteborg, Sverige.
- BLADE. (2024). Understanding Odometry and Its Applications. *SparkFun Electronics*.
- Dahl, E., Karlsson, M., Lindström, F., Nevelius Wernholm, V., Quach, D., & Sjöstrand, S. (2021). *Development and Construction of an Autonomous Transport Robot* (Bachelor Thesis). Institutionen för elektroteknik, Chalmers Tekniska Högskola. Göteborg, Sverige.
- Danwei Wang, F. Q. (2001). *Trajectory Planning for a Four-Wheel-Steering Vehicle* (tekn. rapport). School of Electrical och Electronic Engineering, Nanyang Technological University.
- Ding, Y. (2020). Three Methods of Vehicle Lateral Control: Pure Pursuit, Stanley and MPC. *Medium*.
- DirectDriveTech. (2024). *M1502E-111 Directdrive Motor* [Hämtad: 2025-05-12]. <https://shop.directdrive.com/products/m1502e-111-driectdrive-motor>
- Electronics, C. (n.d.). *Terminal Resistor CAN Bus*. <https://www.csselectronics.com/products/terminal-resistor-can-bus>
- everesd_design. (2020). *PS4 PlayStation Controller To Play*. <https://pixabay.com/vectors/ps4-playstation-controller-to-play-5172918/>
- Falch, M. (2025). *CAN Bus Explained - A Simple Intro [2025]*. <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial>

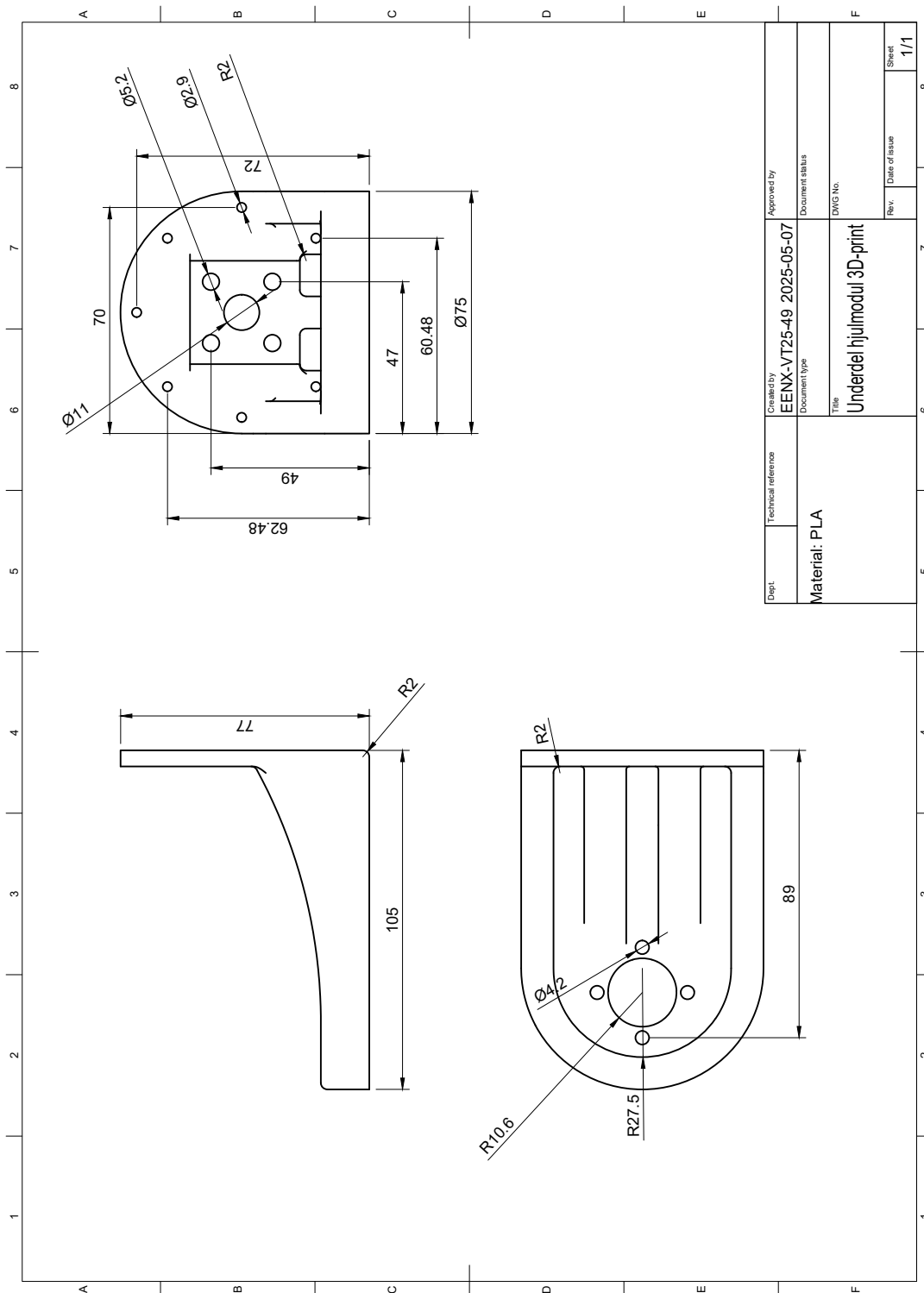
- GitHub contributors. (2022). Seeed_Arduino_CAN. https://github.com/Seeed-Studio/Seeed_Arduino_CAN
- GitHub contributors. (2025a). Adafruit_NeoPixel. https://github.com/adafruit/Adafruit_NeoPixel
- GitHub contributors. (2025b). ArduinoJson. <https://github.com/bblanchon/ArduinoJson>
- MathWorks. (2025). SLAM (Simultaneous Localization and Mapping). *MathWorks*.
- MATLAB. (2024). Kinematic Steering. *MATLAB*.
- Nvidia. (2025). NVIDIA Jetson Orin.
- Remlagret.se. (2025). *RU 85 UUC0 THK Kryssrullager 55x120x15* [Hämtad 9 maj 2025]. <https://remlagret.se/products/ru-85-uuc0-thk-55x120x15-korsat-rullager>
- RS Components. (2025). *LED Strip Light RGB, 5 V, 2000 mm, IP65* [Hämtad: 11 maj 2025]. <https://se.rs-online.com/web/p/led-strip-lights/1807500?gb=s>
- Searcy, R. (2020). The Many Advantages of Radar. *Aptiv*.
- StereoLabs. (2025). *Stereolabs Docs: API Reference, Tutorials, and Integration*.
- Stereolabs. (2025). *Depth Sensing Overview* [Hämtad 7 maj 2025]. <https://www.stereolabs.com/docs/depth-sensing>
- Technology, S. (n.d.). *CAN Bus Voltage Level: Analyzing the Electrical Potential*. <https://sitaltech.com/can-bus-voltage-level-analyzing-the-electrical-potential/>
- V7 Labs, R. K. (2023, 17. januari). *YOLO: Algorithm for Object Detection Explained [+Examples]* [Hämtad 7 maj 2025]. <https://www.v7labs.com/blog/yolo-object-detection>
- VivekB. (2024). Ackerman's Principle of Steering: Working Applications. *Skill-Lync*.
- Zebra. (2025). *What Is a Machine Vision Camera?* Hämtad 5 februari 2025, från <https://www.zebra.com/gb/en/resource-library/faq/what-is-machine-vision-camera.html#:~:text=A%20machine%20vision%20camera%20is,based%20on%20what%20they%20see>.

Bilagor

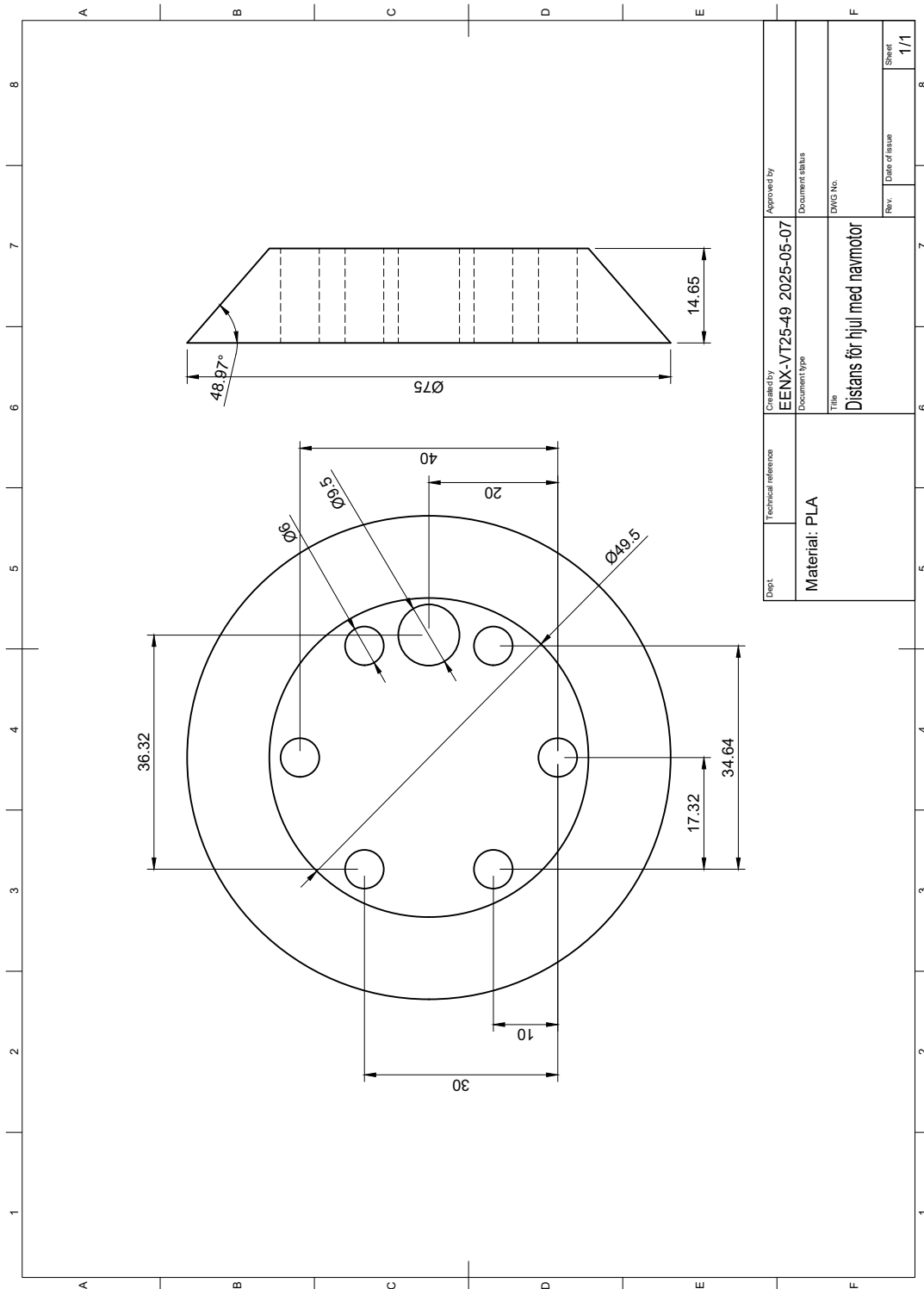
A.1 Ritning – Överdel hjulmodul



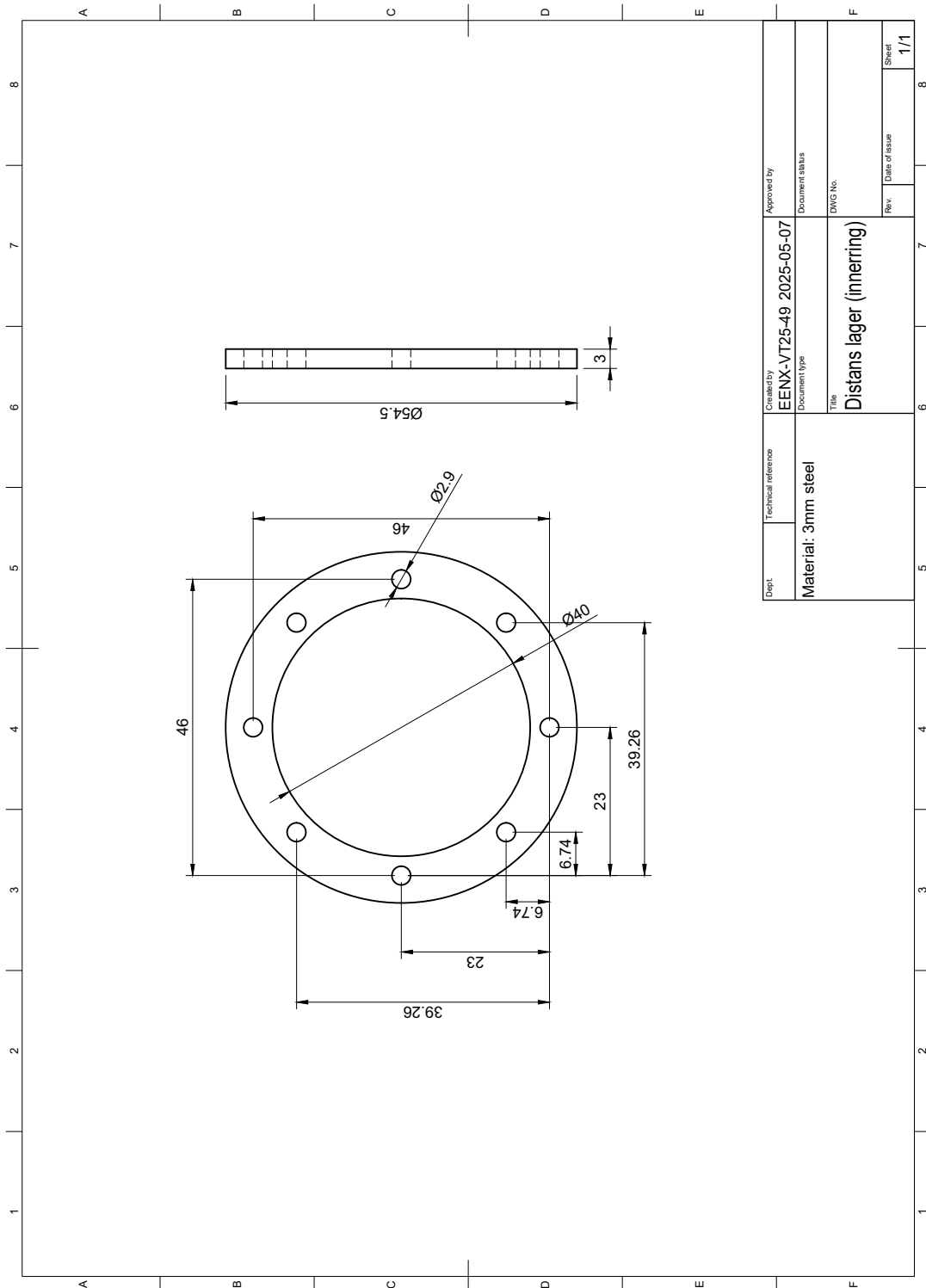
A.2 Ritning – Underdel hjulmodul (3D-utskriven)



A.3 Ritning – Distans konisk

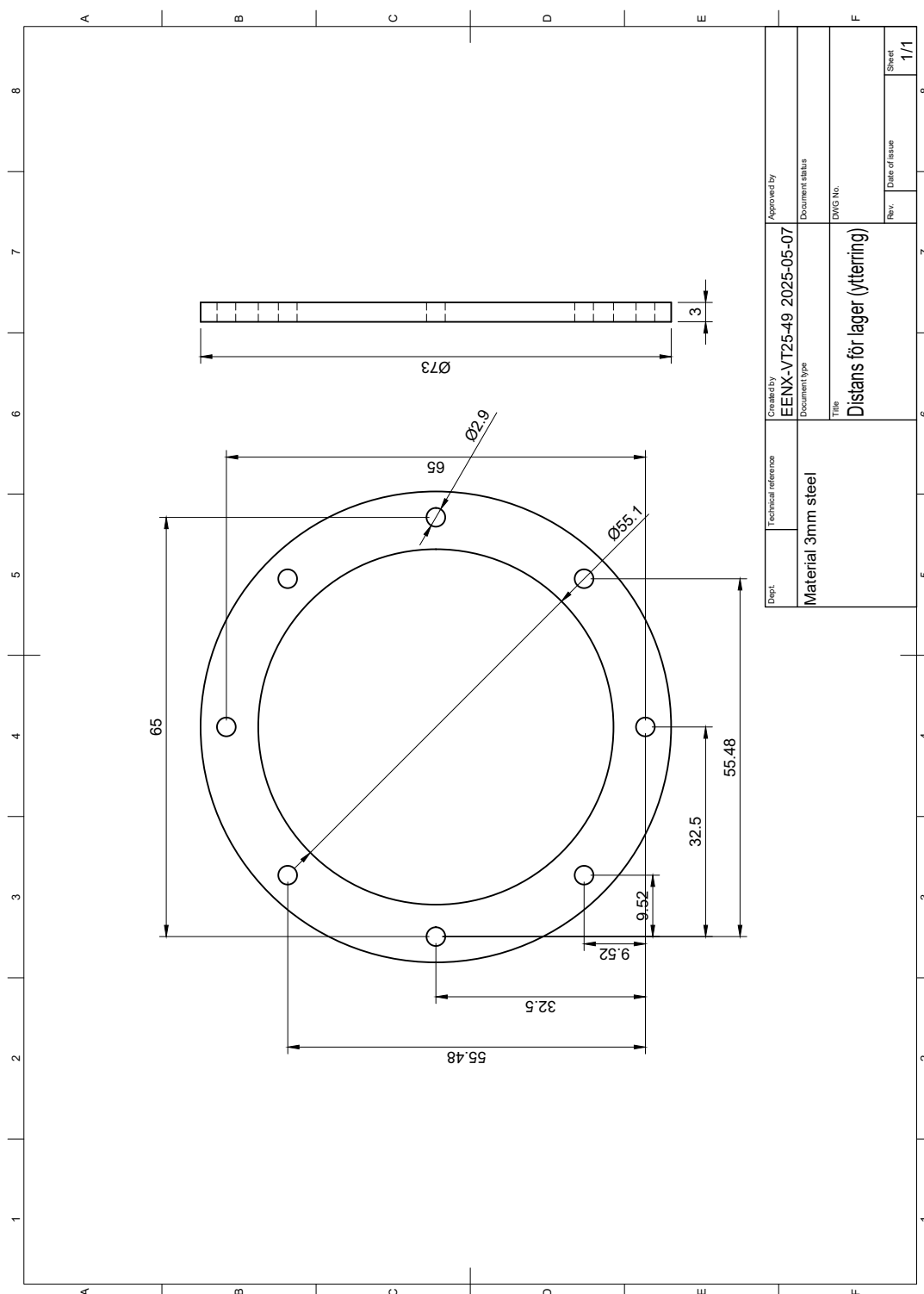


A.4 Ritning – Distans lager, innerring

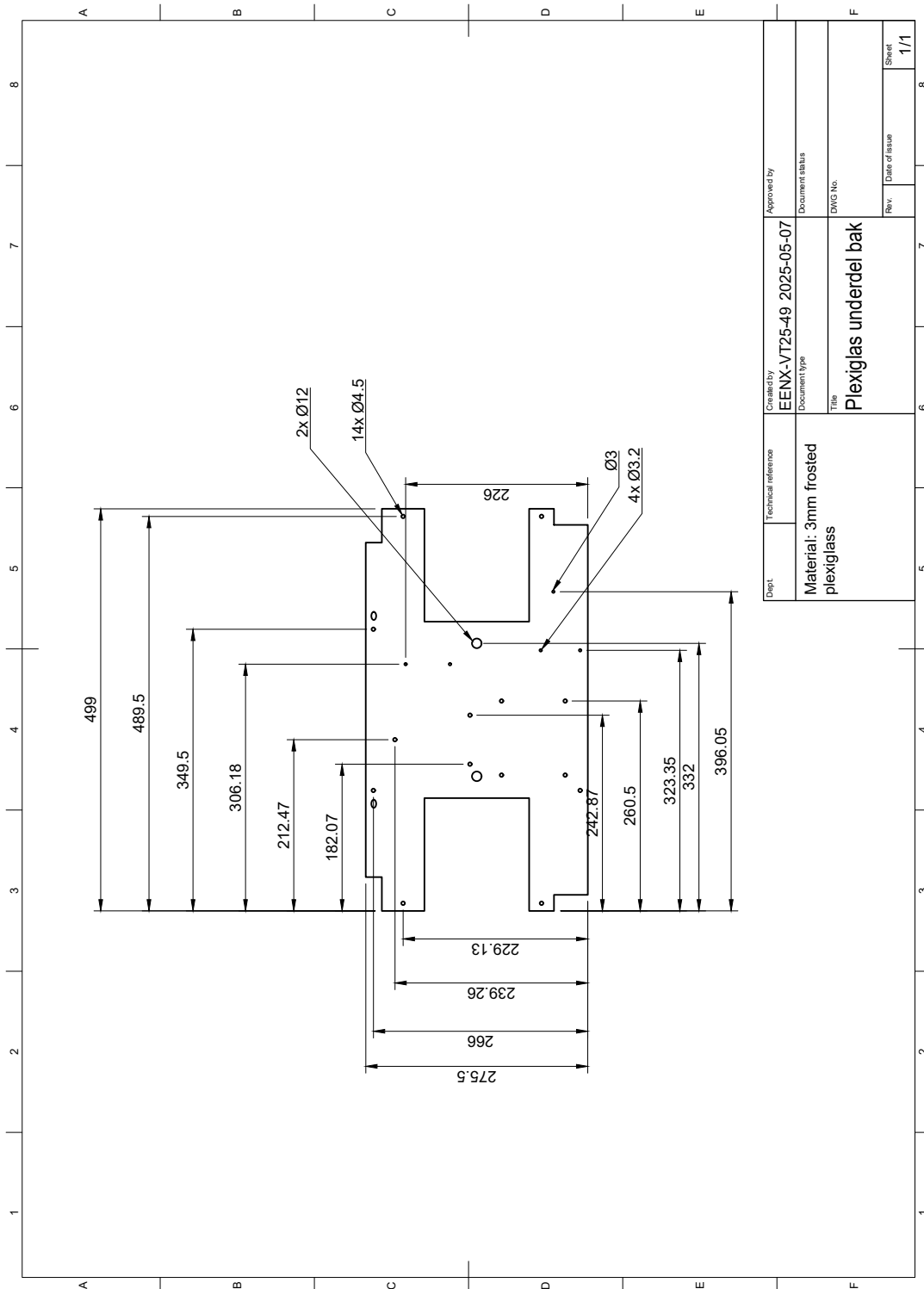


Dept.	Technical reference	Created by	Approved by
	Material: 3mm steel	EENX-VT25-49 2025-05-07	Document status
		Document type	DWG No.
		Title	Rev.
		Distans lager (innerring)	Date of issue
			Sheet
			1/1

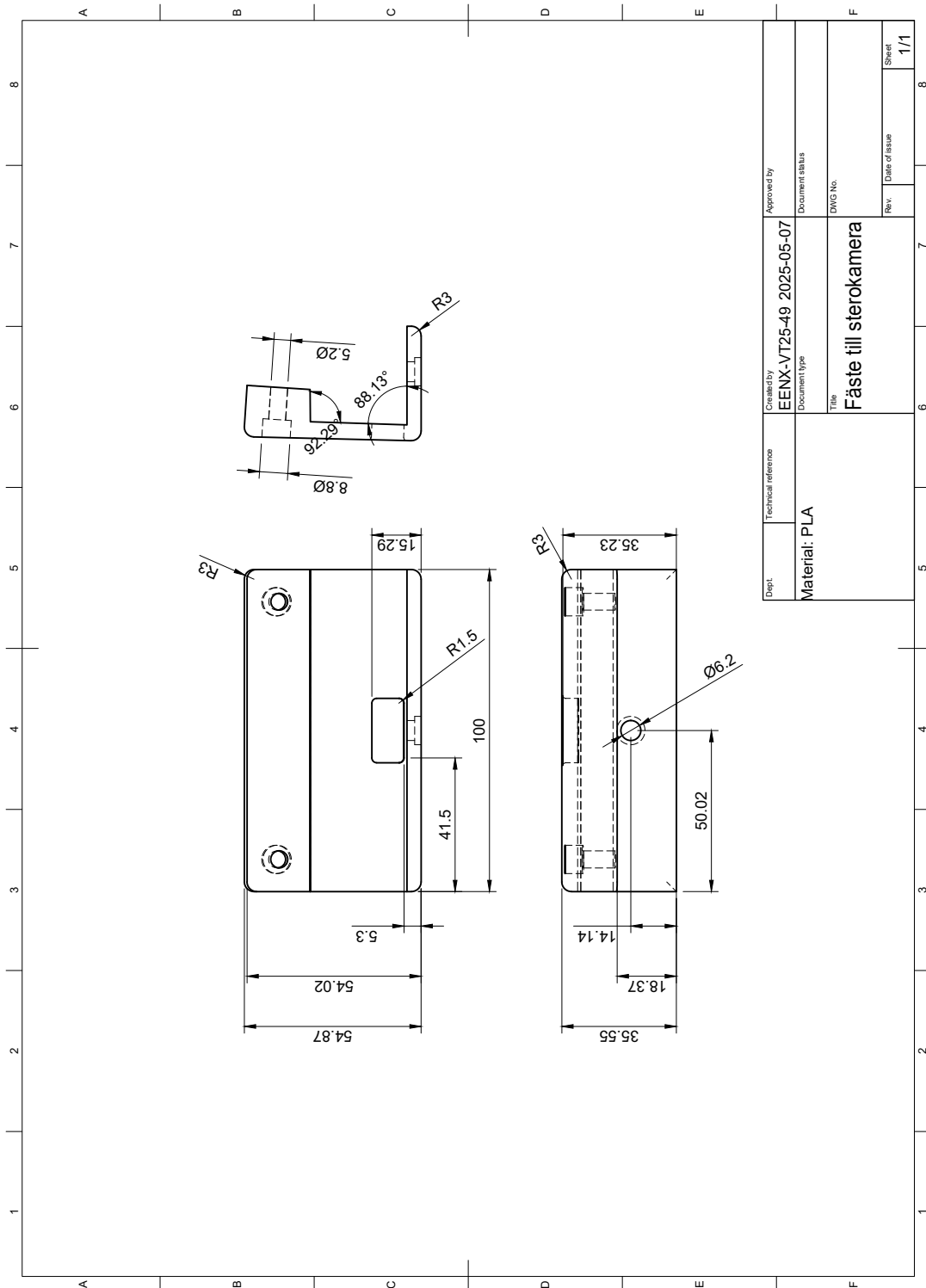
A.5 Ritning – Distans lager, ytterring



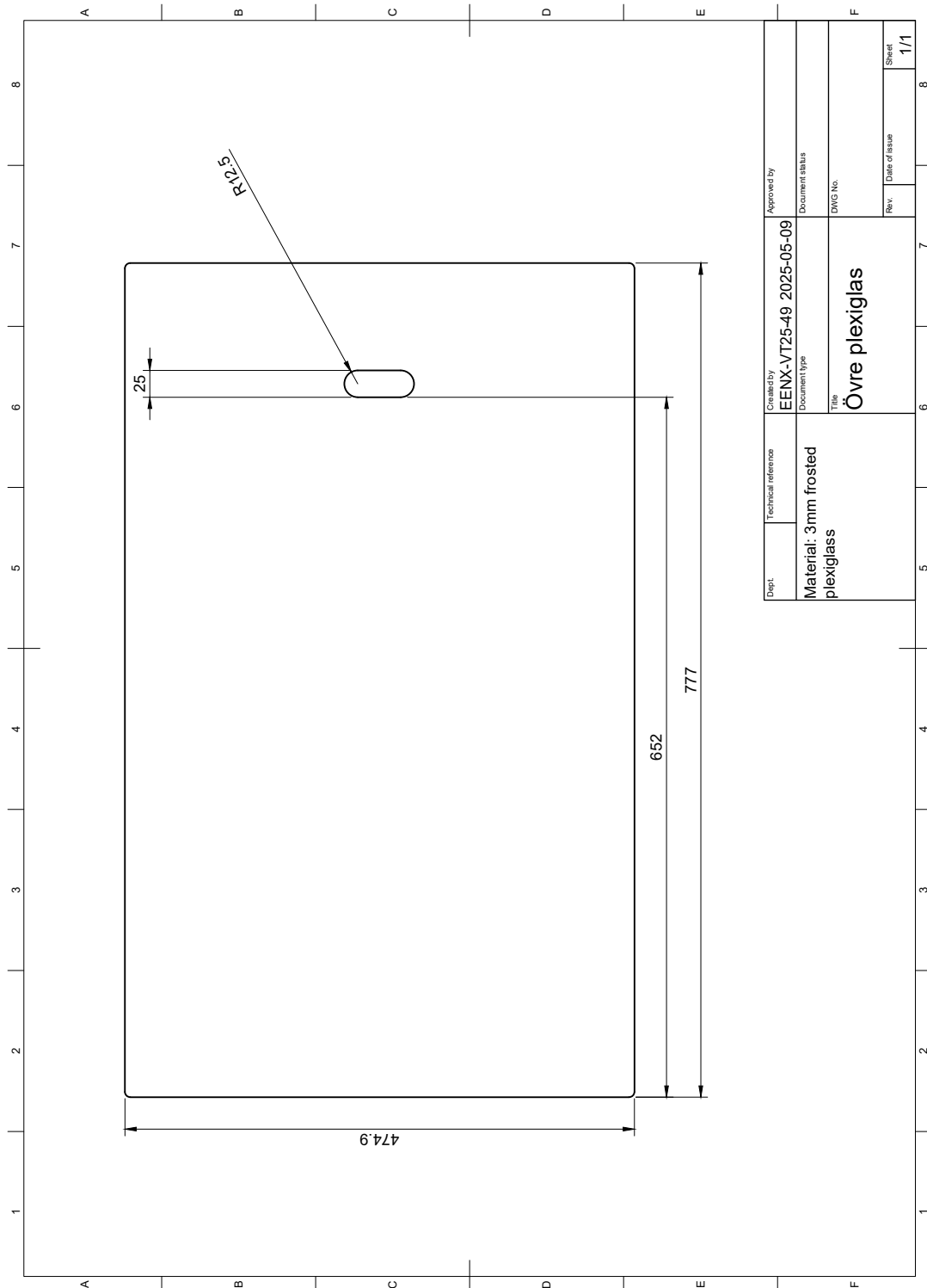
A.7 Ritning – Plexiglas, underdel bak



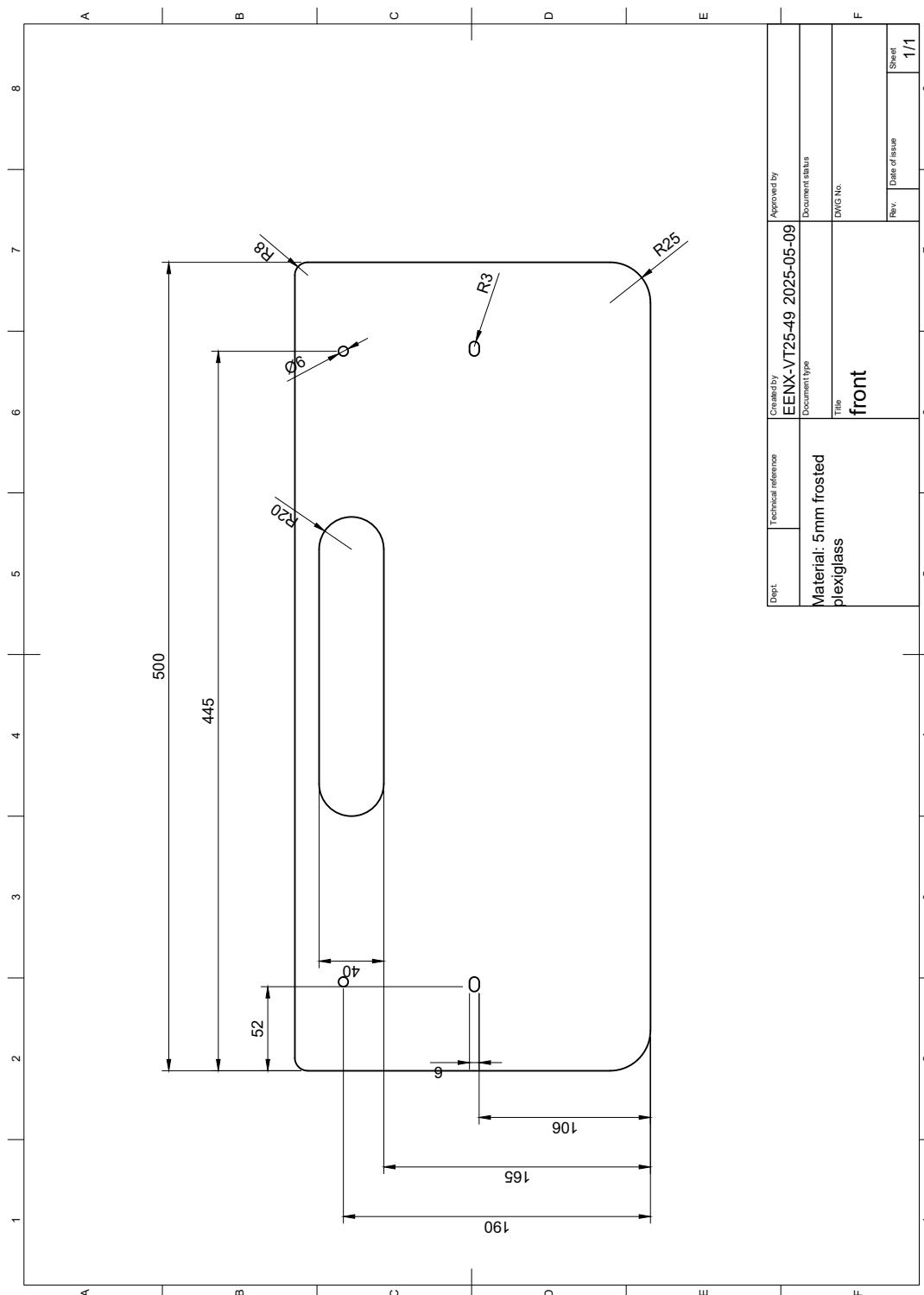
A.8 Ritning – Fäste för ZED-kamera



A.9 Ritning – Övre del av konstruktion



A.10 Ritning – Stötfångare



Dept.	Technical reference	Created by	Approved by
	Material: 5mm frosted plexiglass	EENX-VT25-49	EENX-VT25-49
		Document type	Document status
		Title	DWG No.
		front	
		Rev.	Date of issue
			Sheet
			1/1