



Petter Jemtland

våren 2010

Masteroppgave:

Dokkingssystem for automatisk selvstyrt vogn (AGV)

Institutt for produksjons- og kvalitetsteknikk
Fakultet for ingeniørvitenskap og teknologi

 **NTNU**
Det skapende universitet

MASTEROPPGAVE
Våren 2010
stud. techn. Petter Jemtland

DOKKINGSYSTEM FOR AUTOMATISK SELVSTYRT VOGN (AGV)
(Docking system for automatic guided vehicle (AGV))

Ved instituttet utvikles det et lavkost AGV (Automatic Guided Vehicle) system som del av et ph.d. studium. Denne AGV løsningen bruker en rekke kamera montert i taket til å finne de enkelte AGV'enes posisjoner og gi dem hovedinformasjon for navigering. Denne informasjonen er ikke presis nok til å gi en sikker og nøyaktig dokking for på- og avlasting av AGV'en ved de arbeidsstasjonene den skal betjene.

I denne oppgaven skal det undersøkes hvilke metoder som kan brukes for å posisjonere en AGV nøyaktig ved en på-/avlastingsstasjon. Både posisjonsmåling, presisjonskjøring inn mot målet og mekanismer for å låse AGV fast under lasteoperasjonen er tema for oppgaven. Det skal gjøres både teoretiske analyser og praktiske forsøk med eksisterende AGV'er i laboratoriet.

Arbeidet skal omfatte:

1. En analyse av problemstillingen ved presis posisjonering (dokking) av AGV'er ved laste- og lagerplasser, verktøymaskiner og montasjeceller. Spesielt skal det analyseres hvilke spesifikke problemstillinger som er knyttet til presis dokking av instituttets eksperimentelle AGV'er.
2. En litteraturstudie for å skaffe oversikt over metodene for styringa av AGV'er og en dypere studie av metoder og teknikker for måling av posisjon og styring av bevegelse inn mot presis målposisjon for AGV.
3. Uvikling av forslag til teknisk løsning for et dokkingsystem for instituttets AGV'er. Systemet skal primært anvendes for opp/avlastingsstasjoner for AGV og for dokking i montasjeceller.
4. Så langt det er mulig innenfor oppgavens tidsramme, skal det gjennomføres oppbygging og testing av elementer i dokkingsystemet. Beslutning om hvilke elementer som skal bygges og testes tas i samråd med veileder.

Oppgaveløsningen skal basere seg på eventuelle standarder og praktiske retningslinjer som foreligger og anbefales. Dette skal skje i nært samarbeid med veiledere og fagansvarlig. For øvrig skal det være et aktivt samspill med veiledere.

Innen tre uker etter at oppgaveteksten er utlevert, skal det leveres en forstudierapport som skal inneholde følgende:

- En analyse av oppgavens problemstillinger.
- En beskrivelse av de arbeidsoppgaver som skal gjennomføres for løsning av oppgaven. Denne beskrivelsen skal munne ut i en klar definisjon av arbeidsoppgavens innhold og omfang.
- En tidsplan for fremdriften av prosjektet. Planen skal utformes som et Gantt-skjema med angivelse av de enkelte arbeidsoppgavens terminer, samt med angivelse av milepæler i arbeidet.

Forstudierapporten er en del av oppgavebesvarelsen og skal innarbeides i denne. Det samme skal senere fremdrifts- og avviksrapporter. Ved bedømmelsen av arbeidet legges det vekt på at gjennomføringen er godt dokumentert.

Besvarelsen redigeres mest mulig som en forskningsrapport med et sammendrag både på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse etc. Ved utarbeidelsen av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger for korresponderende steder i tekst, tabeller og figurer anføres på begge steder. Ved bedømmelsen legges det stor vekt på at resultatene er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte og diskuteres utførlig.

Materiell som er utviklet i forbindelse med oppgaven, så som programvare eller fysisk utstyr er en del av besvarelsen. Dokumentasjon for korrekt bruk av dette skal så langt som mulig også vedlegges besvarelsen

Eventuelle reiseutgifter, kopierings- og telefonutgifter må bære av studenten selv med mindre andre avtaler foreligger.

Hvis kandidaten under arbeidet med oppgaven støter på vanskeligheter, som ikke var forutsett ved oppgavens utforming og som eventuelt vil kunne kreve endringer i eller utelatelse av enkelte spørsmål fra oppgaven, skal dette straks tas opp med instituttet.

Innleveringsfrist: 14. juni 2010

Besvarelsen skal innleveres i 1 elektronisk eksemplar (CD) og 2 eksemplar (innbundet).

Ansvarlig faglærer/veileder ved NTNU: Professor Terje K. Lien
Telefon: 73 59 37 88
Mobiltelefon: 970 83 144
E-post: terje.k.lien@ntnu.no

**INSTITUTT FOR PRODUKSJONS-
OG KVALITETSTEKNIKK**



Per Schjølberg

førstemanuensis/instituttleder



Terje K. Lien
faglærer

Forord

Denne rapporten er resultatet arbeidet med masteroppgaven *Dokkingssystem for automatisk selvstyrt vogn* gjennomført i løpet av våren 2010. Oppgaven er utført ved Institutt for produksjons- og kvalitetsstyring, Fakultet for ingeniørvitenskap og teknologi ved Norges teknisk-naturvitenskaplige universitet.

Jeg vil gjerne få takke professor Terje K. Lien for veiledning med oppgaven, Ph.D.-stipendiat Olivier Roulet-Dubonnet for hans verdifulle hjelp og verkstedspersonalet på verkstedet ved instituttet under ledelse av avdelingsingeniør Jan Terje Hakvåg. I tillegg vil jeg få takke Morten Lind, Simen Bjørnstad og Vegard Brøtan for gode innspill og diskusjoner. Til slutt vil jeg også få takke alle ansatte ved Institutt for produksjons- og kvalitetsteknikk som har hjulpet meg i løpet av prosessen med masteroppgaven.

Trondheim 14. juni 2010

Stud.techn. Petter Jemtland

Sammendrag

Masteroppgaven *Dokkingssystem for automatisk selvstyrt vogn (AGV)* har som mål å utvikle en automatisk selvstyrt vogn (AGV) og å skissere et automatisk dokkingssystem for denne AGV-en.

En ny AGV har blitt utviklet med fokus på at konstruksjonen skal være så enkel som mulig og at det skal benyttes lett tilgjengelige komponenter. AGV-en er basert på en rammekonstruksjon laget av aluminiumprofiler. Den drives og styres med differensialdrift med to elektriske børstemotorer montert direkte i rammens vanger.

Den nye AGV-en laget for bruk i instituttets automatiseringslaboratorium presenteres i oppgaven. AGV-ens og dens hovedkomponenter som motorer, gir og motorkontrollere beskrives, og det gis også en kort gjennomgang av hvordan man styrer AGV-ens motorer via motorkontrollerne.

En introduksjon av teorien bak angivelse av en mobil robots posisjon og orientering, til sammen kalt positur gis. To modeller for kinematikken for roboter med differensialdrift og en modell for banestyring av mobile roboter presenteres.

Grunnlaget for å utvikle dokkingløsningen gjøres gjennom en analyse av ulike metoder og sensorer som kan benyttes i dokkingoperasjonen. Virkemåten beskrives enkelt, og styrker og svakheter vurderes opp mot hverandre. Blant sensorene som omhandles er en laserbasert områdesensor, maskinsyn basert på digitale kameraer, infrarøde distansesensorer og ultrasoniske distansesensorer. Ut fra denne analysen blir det bestemt å utvikle en dokkingmetode basert på en type infrarøde sensorer som benytter seg av et trianguleringsprinsipp for å måle avstand. Dette er interessant særlig siden disse sensorene er meget billige.

En løsning for å gjennomføre dokkingoperasjonen med disse infrarøde sensorene presenteres. Løsningen baserer seg på bruk av seks infrarøde sensorer, to med et måleområde fra 20 til 150cm og fire med et måleområde fra 4 til 30cm. De to sensorene med langt måleområde monteres i front og brukes for å bestemme AGV-ens positur relativt til dokken før dokkingoperasjonen begynner. De fire sensorene med kort måleområde monteres langs AGV-ens to sider og benyttes for å finjustere AGV-ens posisjon i forhold til veggene i dokken.

For å bekrefte at det infrarøde sensorene holder mål blir de grundig testet både i forhold til signalkvalitet og i forhold til oppløsning. En anbefaling for hvordan sensorene kan brukes på en måte som er tilfredsstillende i forhold til behovet blir presentert med bakgrunn i disse testene.

Summary

This master thesis called *Docking system for automatic guided vehicle (AGV)* has as a goal to develop an automatic guided vehicle (AGV) and to suggest an automatic docking system for this AGV.

A new AGV has been developed focusing on keeping the construction being as simple as possible, and on using easily available components. The AGV is based on a frame-construction made from aluminium profiles. It is driven and steered using a differential drive with two electric brushed motors mounted directly in the sidewall of the frame.

The new AGV is made for being used in the automation laboratory is presented and its main components such as motors gears and motor controllers are described. A short walkthrough of the control of the AGV's motors using the motor controllers is also given.

An introduction is given into the basics behind the representation of a mobile robot's position and orientation, together called its pose. Two models on the forward kinematics for differential drive mobile robots and a model for open loop control of mobile robots are presented.

The basis for the development of the docking system is developed through an analysis of different methods and sensors that can be used in the docking procedure. Their mode of operation, their strengths and weaknesses are assessed and compared. Among the sensors covered is a laser-based range finder, machine vision systems using digital cameras, infrared range sensors and ultrasonic range sensors. Using this analysis as a basis it is decided to develop a docking system using a certain type of infrared range sensors that use the principle of triangulation to measure the distance to an object. This is especially interesting since these sensors are very low cost.

A solution on how to do the docking using these infrared sensors is presented. The solution is based on using six infrared sensors; two sensors with a range from 20 to 150 cm and four sensors with a range from 4 to 30 cm. The two long range sensors are mounted in front of the AGV and are used for determining the AGV's pose relative to the dock prior to the docking procedure. The four short range sensors are mounted along the two sides of the AGV and are used to fine tune the AGV's position relative to the inner walls of the dock.

To confirm that the infrared sensors perform adequately they are thoroughly tested both in signal quality and resolution. A recommendation on how the sensors should be used in a way that produce a satisfactory result is presented with the basis of these tests.

Innhold

Oppgavetekst	i
Forord	iv
Sammendrag	v
Summary	vi
Figurliste	ix
Tabelliste	xi
Eksempler	xi
Vedlegg	xii
Elektroniske vedlegg	xiii
1 Innledning	1
1.1 Oppgavens oppbygning	1
1.2 Figurer, tabeller og eksempler	2
1.3 Ordliste	2
1.4 Referanser	2
1.5 Vedlegg	2
2 Bakgrunn	3
3 Problemanalyse og krav til dokkingløsning	4
3.1 Begreper	4
3.2 Ønsker og krav til dokkingløsning	4
3.2.1 Nøyaktighet og repeterbarhet	4
3.2.2 Hastighet	5
3.2.3 Kostnad	5
3.2.4 Fleksibilitet / Rekonfigurerbarhet	5
3.2.5 Bygge kvalitet	5
3.3 Forutsetninger	6
3.3.1 Dokkens utseende	6
3.3.2 Eksterne forhold	7
4 Oppdatering av design	8
4.1 Ramme	9
4.1.1 Alternativt rammekonsept laget med tynnplate	10
4.2 Hjul	11
4.2.1 Plassering av hjul	11
4.3 Framdrift	13
4.3.1 Motor og gir	13
4.3.2 Enkodere	14
4.4 Motorkontrollere	15
4.4.1 Kommunikasjon	16
4.4.2 Styring	17

4.4.3	Automatisk innstilling av parametere for motorkontroller	19
4.4.4	Odometri	19
4.5	Tilkoblingskabler	20
4.6	Batterier	20
4.7	SICK områdesensor	20
4.8	Datamaskin	21
4.8.1	Seriellport-utvider	21
4.9	Diverse	22
4.9.1	Hastighet	22
4.9.2	Pris	22
5	Positur og styring	23
5.1	Kinematikk	23
5.2	Differensialstyring	23
5.3	Bestikknavigasjon (dead reconing eller odometri)	25
5.4	Kinematikkmodell I	27
5.4.1	Kinematikk for en mobil robot med differensialstyring	28
5.5	Kinematikkmodell II	30
5.6	Invers kinematikk	32
5.7	Dynamikk	32
5.8	Banestyring	32
5.8.1	Banestyring, løsning fra [7]	32
5.9	Baneplanlegging	36
6	Metoder og sensorer	38
6.1	Sensorer og målemetoder	38
6.1.1	Områdesensor	38
6.1.2	Digitalt kamera / Maskinsyn	46
6.1.3	Induktiv linjefølgning	48
6.1.4	Passiv linjefølgning	49
6.1.5	Elektromagnetiske felt	49
6.1.6	Induktive sensorer	50
6.1.7	Ultrasoniske sensorer	51
6.1.8	Laser-sensor	52
6.1.9	Infrarøde avstandssensorer	52
6.2	Sammenlikning av tekniske spesifikasjoner for ulike sensorer	55
6.3	Videre lesning om sensorer og målemetoder for mobile roboter	56
6.4	Konseptforslag	56
6.4.1	Bildegjenkjenning vha. kamera montert i rommet	56
6.4.2	Bildegjenkjenning vha. kamera montert på AGV	57
6.4.3	Benytte områdesensor	57
6.4.4	Benytte flere infrarøde avstandssensorer	57
6.4.5	Oppsummering av de foreslåtte konseptene	57

7	Dokking ved hjelp av infrarøde sensorer	59
7.1	Konseptuelle sensorkonfigurasjoner	60
7.1.1	Konsept 1	60
7.1.2	Konsept 2	62
7.1.3	Konsept 3	63
7.1.4	Konsept 4	64
7.2	Skissert dokkingprosedyre	65
7.2.1	Beregning av posisjon i forhold til senterlinje	66
7.2.2	Beregning av bane mellom utgangspunkt og punkt på senterlinje	67
7.3	Testing av sensorer, SHARP GP2D120	69
7.3.1	Signalkvalitet	70
7.3.2	Oppløsning	74
7.3.3	Frekvens og oversampling	79
7.3.4	Oppsummering	80
7.4	Testing av sensorer, SHARP GP2Y0A02YK	81
7.4.1	Oppløsning	83
7.4.2	Oppsummering	84
7.5	Montering av sensorer	85
7.6	Oppsummering	85
8	Videre arbeid	87
9	Oppsummering og konklusjoner	88
10	Engelsk-Norsk Ordliste	89
11	Referanser	90

Figurliste

Figur 1	– Illustrasjon av dokkens utseende	6
Figur 2	– Illustrasjon av AGV	8
Figur 3	– AGV sett forfra	9
Figur 4	– Illustrasjon av ramme	10
Figur 5	– Illustrasjon av alternativt konsept i tynnplate	11
Figur 6	– Vange med to posisjoner for montering av motor	12
Figur 7	– AGV sett fra siden med drivhjulene montert i bakerste posisjon	12
Figur 8	– Eksplodert figur av sammenstillingen av enkoder, motor, gir og hjul.	14
Figur 9	– Skjematisk beskrivelse av framdriftssystemet	15
Figur 10	– Akselerasjon og hastighetskurve for akselerasjon som følger \sin^2	16
Figur 11	– Rekkefølge kommandoer ved hastighetskontrollmodus med forhåndsbestemt akselerasjonsprofil, hentet fra [5].	18
Figur 12	– SICK områdesensor (https://www.mysick.com/saqqara/wrapper.aspx?id=im0025864)	21
Figur 13	– Plassering av områdesensor	21

Figur 14 – Ulike hjulkonfigurasjoner for mobile roboter med differensialstyring (løst basert på [7])	24
Figur 15 – Illustrasjon av differensialdrift	24
Figur 16 – Usikkerhet i positur (basert på [7, s. 191]).	26
Figur 17 – Usikkerhet rundt det virkelige gripepunktet på drivhjulene gir en usikkerhet på $2w$ i hjulavstanden l . [8]	26
Figur 18 – Globalt og lokalt koordinatsystem. Basert på [7]	27
Figur 19 – Rotasjon om ICC	30
Figur 20 – Banestyring	33
Figur 21 – Illustrasjon av begrenset bevegelsesrom for dokking	37
Figur 22 – Illustrasjon av synsfeltet til områdesensor	39
Figur 23 – Illustrasjon av teoretisk resultat fra områdesensor	40
Figur 24 – Plottede verdier fra områdesensor	41
Figur 25 – Bilde av AGV-ens retning ved avlesning av områdesensor	41
Figur 26 – Plottede verdier fra områdesensor	42
Figur 27 – Panorama av laboratorium	42
Figur 28 – Lineær oppløsning områdesensor	43
Figur 29 – Avstand mellom målepunkter og diameter på målepunkt (vinkel $\beta=0^\circ$)	44
Figur 30 – Oppløsning for en avstand, d , på 1000mm.	45
Figur 31 – Illustrasjon av måleområde for ultrasonisk ovenfra (hentet fra [30])	51
Figur 32 - Illustrasjon av infrarød sensor	53
Figur 33 – Illustrasjon av måleområde for infrarød sensor (hentet fra [1])	54
Figur 34 – Illustrasjon av dokk.	59
Figur 35 – Illustrasjon av måleområdet til infrarød sensor	60
Figur 36 – Illustrasjon av sensorplassering i konsept 1	61
Figur 37 – Illustrasjon av sensorplassering i konsept 2	62
Figur 38 – Konsept 2, detalj fra bakfra	63
Figur 39 – Illustrasjon av sensorplassering i konsept 3	63
Figur 40 – Illustrasjon av sensorplassering i konsept 4	64
Figur 41 – Hjelpesfigur for vinkelberegning	66
Figur 42 – Illustrasjon av en kubisk Bézier-kurve	67
Figur 43 – Illustrasjon av bevegelsesmønstre	68
Figur 44 – Infrarød avstandssensor, SHARP GP2D120	69
Figur 45 – Timing diagram for infrarød sensor (hentet fra Vedlegg 6)	70
Figur 46 – Testing av sensor, forsøk 1	70
Figur 47 – Testing av sensor, forsøk 2	71
Figur 48 – Testing av sensor, forsøk 3	71
Figur 49 – Koblingsskjema av fysisk lavpassfilter	72
Figur 50 – Testing av sensor, forsøk 4	72
Figur 51 – Koblingsskjema sensor	73
Figur 52 – Koblingsskjema spenningsregulator	74
Figur 53 – Karakteristikk av utspenning for infrarød sensor SHARP GP2D120, hentet fra Vedlegg 6	75
Figur 54 – Gjennomsnittsmålinger ved ulike distanser, sensor 1	76
Figur 55 – Sammenlikning av avlest data for Sensor 1 og Sensor 2	78
Figur 56 – Infrarød avstandssensor, SHARP GP2Y0A02YK	81

Figur 57 – Karakteristikk av utspenning for infrarød sensor SHARP GP2Y0A02YK, hentet fra Vedlegg 7	82
Figur 58 – Sammenlikning av avlest data for Sensor 1 og Sensor 2 (GP2Y0A02YK).....	83
Figur 59 – Anbefalt montering av infrarød sensor, hentet fra Vedlegg 6.....	85

Tabelliste

Tabell 1 – Kostnadsoversikt.....	22
Tabell 2 – Utdrag av tekniske spesifikasjoner for områdesensor, hentet fra [13].....	43
Tabell 3 – Lineær oppløsning ved ulike vinkler.....	44
Tabell 4 – Tekniske spesifikasjoner induktive sensorer.....	50
Tabell 5 – Tekniske spesifikasjoner SICK UM18-51112 (hentet fra www.sick.com).....	51
Tabell 6 – Tekniske spesifikasjoner SICK OD2-N120W60A2 (hentet fra www.sick.com)	52
Tabell 7 – Tekniske spesifikasjoner, SHARP infrarøde distansesensorer.....	55
Tabell 8 – Sammenlikning av de egenskapene til ulike distansesensorer	55
Tabell 9 – Oversikt over forsøk med infrarød distansesensor.....	73
Tabell 10 – Målinger ved ulike distanser (1000 datapunkter) Sensor 1	77
Tabell 11 – Målinger ved ulike distanser (1000 datapunkter) Sensor 2.....	77
Tabell 12 – Lineær oppløsning ved ulike avstander fra sensor.....	79
Tabell 13 – Målinger ved ulike distanser (1000 datapunkter) Sensor 1, GP2Y0A02YK.....	82
Tabell 14 – Målinger ved ulike distanser (1000 datapunkter) Sensor 2, GP2Y0A02YK.....	83
Tabell 15 – Lineær oppløsning ved ulike avstander fra sensor GP2Y0A02YK.....	84
Tabell 16 – Sammenlikning av absolutt oppløsning med 10-bits oppløsning	85
Tabell 17 – Engelsk-Norsk ordliste.....	89

Eksempler

Eksempel 1 – Bruk av libEpos med Python.....	19
Eksempel 2 – bruk av klassen DistanceTable.....	76
Eksempel 3 – Oversampling.....	80

Vedlegg

- Vedlegg 1 – Teknisk dokumentasjon motor, Maxon RE 40, hentet fra <http://www.maxonmotor.com/>
- Vedlegg 2 – Teknisk dokumentasjon gir, Maxon GP 52 C, hentet fra <http://www.maxonmotor.com/>
- Vedlegg 3 – Teknisk dokumentasjon enkoder, Maxon MR 500IMP, hentet fra <http://www.maxonmotor.com/>
- Vedlegg 4 – Teknisk dokumentasjon svivelhjul, Tente 2470PJH075P40, hentet fra <http://www.tente-as.no>
- Vedlegg 5 – Arbeidstegninger av komponenter til AGV
- Vedlegg 6 – Teknisk dokumentasjon infrarød distansesensor, SHARP GP2D120 ¹
- Vedlegg 7 – Teknisk dokumentasjon infrarød distansesensor, SHARP GP2Y0A02YK²
- Vedlegg 8 – Script for mottak og logging av verdier fra seriellport, skrevet i Python 2.6
- Vedlegg 9 – Script for avlesning av sensor med mikrokontroller, skrevet i Wiring
- Vedlegg 10 – Eksempler fra utvidet versjon av biblioteket libEPOS
- Vedlegg 11 – Script for å beregne avstand fra avlest sensorverdi med interpolasjon, skrevet i Python 2.6
- Vedlegg 12 – Sammenlikning av tekniske spesifikasjoner for ulike sensorer
- Vedlegg 13 – General Application Note: Distance Measuring Devices³ [1]
- Vedlegg 14 – Mellomregning for posisjon
- Vedlegg 15 – Forstudierapport

På grunn av rettighetsproblemer er vedleggene 6, 7 og 13 ikke tilgjengelige i den elektroniske versjonen av denne oppgaven. De er imidlertid tilgjengelige fra internettdressene angitt i fotnotene. Vedleggene er også tilgjengelige som enkeltfiler på vedlagte CD-ROM.

¹ Tilgjengelig fra: https://www1.elfa.se/data1/wwwroot/assets/datasheets/gnSharp_Distanzsensoren-GP-2D120J000F_EN.pdf

² Tilgjengelig fra: https://www1.elfa.se/data1/wwwroot/assets/datasheets/gpSharp_Distanzsensoren-GP-2Y0A02YK0F_EN.pdf

³ Tilgjengelig fra: <http://www.seattlerobotics.org/WorkShopRobot/level2/datasheets/Sharp-DistanceSensorAppNote.pdf>

Elektroniske vedlegg

Disse vedleggene finnes på den vedlagte CD-ROM-platen.

- **CAD**

CAD-filer og arbeidstegninger til de ulike delene som AGV-en er bygget av.

- **CAD-filer**

CAD-filer av de delene som er produsert. AGV-en er tegnet med programmet Autodesk Inventor 2010. Hovedsammenstillingen av alle delene finnes i filen "Assembly1.iam".

- **Arbeidstegninger**

Arbeidstegninger til de ulike delene som ble laget for AGV-en. Disse tegningene er oppdatert etter produksjon for å rette mindre feil, og kan i visse tilfeller avvike noe fra de produserte delene.

- **Dokumentasjon**

Dokumentasjon for de ulike komponentene som inngår i AGV-en:

- **Enkoder**

- **EPOS (motorkontroller)**

Inneholder dokumentasjonen som ble levert fra produsenten MAXON, denne finnes også på egen CD-ROM

- **Gir**

- **Infrarøde sensorer**

- **Motor**

- **SICK Range Finder**

- **Svivelhjul**

- **libEPOS, utvidet og forbedret versjon**

Revisjon 5, hentet fra SourceForge [2] 5. juni 2010. Den nyeste versjonen av biblioteket kan også til enhver tid finnes på bibliotekets nettside på SourceForge.

- **Programmer**

- **Interpolasjon**

Program for omregning av sensorverdier til distanse

- **Les seriell**

Programmer for testing av sensor med mikrokontroller via seriell-port

- **Datamaskin**

Skriptet som kjøres på datamaskinen for å ta imot og lagre de avleste verdiene

- **Mikrokontroller**

Programmet som kjører på mikrokontrolleren som leser de analoge verdiene fra sensoren og sender de til datamaskinen via seriell kommunikasjon

- **Prosjektrapport**

- **Vedlegg**

1 Innledning

Masteroppgaven *Dokkingsystem for automatisk selvstyrt vogn (AGV)* har som hovedmål å ende opp med en fungerende ny AGV, samt et forslag til teknisk løsning for dokking av denne AGV-en.

Utdrag fra oppgaveteksten:

I denne oppgaven skal det undersøkes hvilke metoder som kan brukes for å posisjonere en AGV nøyaktig ved en på-/avlastingsstasjon. Både posisjonsmåling, presisjonskjøring inn mot målet og mekanismer for å låse AGV fast under lasteoperasjonen er tema for oppgaven. Det skal gjøres teoretiske analyser og praktiske forsøk med eksisterende AGV'er i laboratoriet.

Oppgave 1

En analyse av problemstillingen ved presis posisjonering (dokking) av AGV'er ved laste- og lagerplasser, verktøymaskiner og montasjeceller. Spesielt skal det analyseres hvilke spesifikke problemstillinger som er knyttet til presis dokking av instituttets eksperimentelle AGV'er.

Oppgave 2

En litteraturstudie for å skaffe oversikt over metodene for styringa av AGV'er og en dypere studie av metoder og teknikker for måling av posisjon og styring av bevegelse inn mot presis målposisjon for AGV.

Oppgave 3

Utvikling av forslag til teknisk løsning for et dokkingsystem for instituttets AGV'er. Systemet skal primært anvendes for opp/avlastingsstasjoner for AGV og for dokking i montasjeceller.

Oppgave 4

Så langt som det er mulig innenfor oppgavens tidsramme, skal det gjennomføres oppbygging og testing av elementer i dokkingsystemet. Beslutninger om hvilke elementer som skal bygges og testes tas i samråd med veileder.

Punktet i oppgaven som sier at mekanismer for låsing under dokkingoperasjonen er ikke behandlet oppgaven. Grunnen til dette er at det å skissere delen av dokkingsystemet som tar for seg posisjonering og kjøring inn i dokk hadde høyere prioritet enn det mekanisk låsing hadde. I tillegg til dette er at det er ønskelig å kartlegge i hvor stor grad det er behov for mekanisk låsing ved å gjennomføre tester med den ferdige AGV-en. Å undersøke behov og metoder for mekanisk låsing av AGV-en under dokking er nevnt som alternativt videre arbeid.

I tillegg til de fire oppgavene beskrevet i oppgaveteksten var en del av oppgaven å tegne og bygge en ny AGV.

1.1 Oppgavens oppbygging

Som oppgaven beskriver tar denne oppgaven for seg utviklingen av et dokkingsystem for en automatisk selvstyrt vogn eller AGV. I tillegg til dokkingsystemet har en ny AGV blitt utviklet.

Selve oppgavebesvarelsen er delt inn i fem hovedkapitler:

- *Kapittel 3* gjør en kort problemanalyse og en gjennomgang av krav og ønsker til dokkingløsningen.
- *Kapittel 4* dokumenterer utformingen av den nye AGV-en og de ulike komponentene som er benyttet for å bygge denne. I tillegg beskrives metodene som benyttes for å styre de to motorkontrollerne.
- I *Kapittel 5* gjøres et litteraturstudie som tar for seg prinsipper bak beskrivelse av positur og styring av mobile roboter. To ulike kinematikk-modeller for roboter med differensialdrift og en metode for banestyring av mobile roboter presenteres.
- I *Kapittel 6* undersøkes ulike sensorer og metoder som det er aktuelt å benytte i dokkingen. Sensorenes virkemåte, styrker og svakheter presenteres kort, og de ulike sensorenes egenskaper sammenliknes og vurderes for å finne ut hvilke som er best egnet til å brukes i en dokkingoperasjon.
- I *Kapittel 7* skisseres først en metode for dokking basert på infrarøde distansesensorer. Etter at løsningen er skissert utføres en testing av sensorene for å undersøke hvorvidt de er gode nok til å kunne benyttes i den skisserte metoden.

I tillegg til disse hovedkapitlene kommer et kapittel som kort tar for seg bakgrunnen bak oppgaven (Kapittel 2). Et kapittel som anbefaler videre arbeid (Kapittel 8) og et kapittel med oppsummering og konklusjoner er plassert slutt i oppgavebesvarelsen (Kapittel 9).

1.2 Figurer, tabeller og eksempler

Illustrasjoner, fotografier og grafer som er benyttet i oppgaven er for enkelthets skyld alle merket under etiketten "Figur". Tabeller er merket og nummerert under etiketten "Tabell", og eksempler er nummerert som "Eksempel". Lister over disse tre typene er gitt i de innledende sidene over.

1.3 Ordliste

En ordliste som klargjør sammenhengen mellom engelsk og norske faguttrykk finnes som Kapittel 10 til slutt i oppgaven.

1.4 Referanser

Referanser til eksterne kilder som blir brukt blir fortløpende referert med et nummer i klammer i teksten. Den nummererte listen over alle de refererte kildene finnes i Kapittel 11 helt sist i oppgaven.

1.5 Vedlegg

Vedlegg til oppgaven omfatter utvalgt teknisk dokumentasjon, dataprogrammer samt mellomregning det ikke er hensiktsmessig å inkludere i selve oppgaven. De skriftlige vedleggene er nummerert fra 1 til 15, og en liste over disse er gitt over. Vedleggene er ikke utstyrt med sidetall, men skilles fra hverandre med nummererte skillerark.

I tillegg til de skriftlige vedleggene er det også vedlagt en CD-ROM med ytterligere vedlegg. Denne inneholder ytterligere teknisk dokumentasjon i tillegg til den som er vedlagt i oppgaven. Alle dataprogrammer er vedlagt, det samme er CAD-filer av AGV-ens maskinerte komponenter.

2 Bakgrunn

Rene logistikkoppgaver som å flytte deler og produkter mellom maskiner, og mellom maskiner og lager i en produksjonssituasjon er oppgaver som kan være tidkrevende og som ikke er verdiskapende, det vil si at operasjonen ikke øker verdien på det produktet som transporteres. Oppgaven er også relativt enkel med tanke på at den ikke krever høy kunnskap for å utføres. Den krever heller ikke veldig stor presisjon (i forhold til mange av de andre operasjoner som utføres i en produksjonsprosess). På grunn av dette er det meget interessant å automatisere logistikkoppgaver. Automatisering av logistikkoppgaver er da heller ikke en ny tanke, det ble gjort først gjennom samlebåndet og senere med såkalte AGV-er. En AGV er en Automatisert Selvstyrt Vogn (Automated Guided Vehicle). AGV-er er i dag i bruk innenfor ulike områder i samfunnet. De brukes i primært i produksjonsbedrifter men er også blitt tatt i bruk på for eksempel sykehus. Blant produksjonsbedriftene er det i hovedsak de bedriftene som har høy grad av spesialisering (smalt produktspekter) og/eller meget høyt volum som benytter AVG-er. En av grunnene til dette er teknologien som de fleste AGV-systemer benytter; systemene er dyre å installere, og de er dyre å endre på. Dette gjør dem lite egnet for moderne produksjonsbedrifter som streber etter å oppnå høy fleksibilitet for å betjene stadig mer usikre markeder. Slike bedrifter har behov for å raskt og rimelig kunne endre på eller rekonfigurere produksjonssystemene sine for å tilpasse seg endringer og å introdusere nye produkter.

I dette segmentet åpner det seg et behov for metode å automatisere logistikkoppgavene på en måte som også ivaretar fleksibiliteten og muligheten til rekonfigurering av produksjonssystemene uten at det medfører for høye kostnader.

Det er dette behovet man prøver å løse gjennom å utvikle en AGV kan styres gjennom et meget fleksibelt distribuert styresystem [3]. Dette styresystemet benytter digitale kameraer som observerer AGV-en i dens arbeidsområde og ut fra dette styrer den til ønsket sted, eller andre fleksible navigasjonssystemer som områdekartlegging med en laserbasert områdesensor. Disse systemene gir i dag en god nok presisjon til å navigere rundt i et produksjonsmiljø. I noen tilfeller er imidlertid kravet til presisjon høyere enn det systemet kan levere. Derfor er det behov for et system som benyttes når AGV-en skal ha en mer nøyaktig posisjonering. Situasjonene der det er behov for nøyaktig posisjonering er eksempelvis ved av- og pålasting av AGV-en, kjøring av AGV-er i heis og ved automatisk opplading av AGV-ens batterier. Operasjonen som utføres i disse situasjonene kalles dokking.

I tillegg til de konvensjonelle bruksområdene åpner det seg nye bruksområder for AGV-er med et høyt fleksibelt styringssystem. Man ser for eksempel for seg at flere AGV-er sammen kan benyttes som en bevegelig montasjelinje, der produktene som skal monteres fraktes mellom arbeidsposisjonene på AGV-ene og at alle operasjoner utføres med AGV-en som montasjebord. Slik trenger ikke produktene som skal monteres en gang å lastes av AGV-en før montasjen er helt ferdig. Dette vil muliggjøre det å flytte montasjelinjen uten å gjøre fysiske endringer på produksjonen. Det vil også endre forutsetningene for hvordan produktstrømmene kan være. Det muliggjør kryssende strømmer av produkter, noe som er vanskelig i dag da linjene må passere over og under hverandre. Det vil også lette fletting av flere konvergerende eller divergerende produktstrømmer.

3 Problemanalyse og krav til dokkingløsning

I dette kapitlet gjennomføres en analyse av ønsker og krav til dokkingløsningen. Men, først beskrives en del begreper som vil bli benyttet i oppgaven.

3.1 Begreper

Før selve analysen av problemet begynner er det noen sentrale begreper som bør klargjøres:

AGV

AGV-en eller den automatisk selvstyrte vognen er den mobile roboten som skal *dokkes*. Det er en batteridrevet mobil robot som styres på et tilnærmet flatt underlag.

Dokking

Operasjonen kalt dokking eller dokkingoperasjonen innebærer at den mobile roboten som omhandles skal bevege seg fra en posisjon foran selve *dokken* og inn i denne. Inne i dokken skal den stoppe på en nøyaktig posisjon/positur.

Dokk

En dokk er området og apparaturen rundt der *AGV-en* skal stoppe. Mer om dokkens utseende finnes i Avsnitt 3.3.1.

Dokkingposisjon/positur

Dokkingposisjonen/posituren er den plasseringen *AGV-en* skal ha når har stoppet. For mer om forskjellen mellom posisjon og positur se Kapittel 5.1.

3.2 Ønsker og krav til dokkingløsning

3.2.1 Nøyaktighet og repeterbarhet

Hvilke krav til nøyaktighet i positur som stilles til dokkingløsningen er avhengig av hvilke bruksområde det er snakk om. Ulike aktuelle bruksområder vil ha ulike krav til nøyaktighet:

- **Plukking fra AGV med robot (med maskinsyn)**

Dersom plukkingen foregår med en robot utstyrt med maskinsyn er nøyaktigheten mindre viktig. Maskinsyn kan benyttes for å bestemme AGV-en er posisjonert, og ut fra dette beregne hvor plukkeposisjonene er. Her vil det kanskje holde om posisjonen har en nøyaktighet på $\pm 10\text{mm}$, avhengig av hvilke type del som skal plukkes og hvordan maskinsyn-systemet fungerer.
- **Plukking fra AGV med robot (uten maskinsyn)**

Dette vil være veldig avhengig av delen og griperen som skal plukke delen. For enkelte deler kan kravet til nøyaktighet være $\pm 1\text{mm}$, mens for andre deler kan det være langt mindre.
- **Løfting av palett fra AGV**

Hvilken nøyaktighet som kreves her er veldig avhengig av utformingen på løftemekanismen, men det vil være å anta at nøyaktigheten bør være i området rundt $\pm 3\text{-}5\text{mm}$.
- **Skyving av palett fra AGV**

Nøyaktighetskravet vil også her avhenge mye av mekanismen som benyttes til avskyving, men det er også her mulig å anta at nøyaktigheten bør ligge i området $\pm 3\text{-}5\text{mm}$.

- **Lading av robotens batterier**

Ved lading av robotens batterier er det fullt mulig å lage en mekanisme som ikke krever meget høy nøyaktighet. Det bør være mulig å designe en ladestasjon som tillater en feil på opptil $\pm 5\text{-}10\text{mm}$.

I dette tilfellet er det ønskelig å oppnå en posisjonsnøyaktighet bedre enn $\pm 1,5\text{mm}$. Dette vil bety at det er mulig å plukke deler fra AGV-en med en industrirobot uten å ha behov for et fordyrende maskinsyn-system.

3.2.2 Hastighet

Hastigheten på dokkingoperasjonen er et sentralt punkt. Særlig i en produksjonssituasjon er hastigheten en viktig faktor. Dersom hastigheten på dokkingen er lav vil man også kunne operere systemet med et lavere antall AGV-er, og dermed redusere investerings- og driftskostnadene ved systemet. Hastigheten på dokkingoperasjonen er derimot ikke den aller viktigste faktoren i denne oppgaven, da det er mer viktig å prøve å oppnå en tilfredsstillende nøyaktighet.

3.2.3 Kostnad

Kostnaden tilknyttet installasjon og bruk av AGV-systemet (både investerings- og driftskostnader) vil selvsagt også være en meget viktig egenskap. Det er viktig at kostnadene er så lave som mulig, særlig med tanke på at prosjektet går ut på å lage et lavkostnads AGV-system.

Lav kostnad kan oppnås ved å utforme en dokkingløsning basert på enkel teknologi, som samtidig er pålitelig og effektiv i bruk.

Man vil også ha et kostnadsspørsmål i forhold til antallet dokker og antallet AGV-er. Dersom antallet dokkingposisjoner overstiger antallet AGV-er vil det være en fordel om mesteparten av utstyret i dokkingoperasjonen befinner seg på AGV-en. Dersom antallet AGV-er overstiger antallet dokkingposisjoner vil det være motsatt. Dersom man også tenker på fleksibiliteten er det en fordel dersom selve dokken inneholder minst mulig utstyr. Den blir da antakelig enklere og mindre kostnadskrevende å flytte.

3.2.4 Fleksibilitet / Rekonfigurerbarhet

Et hovedfokus med dokkingløsningen som skal utvikles er at den skal ha høy fleksibilitet. Dette også er en av hovedtankene bak utviklingen av det nye styringssystemet der man ønsker å utvikle et AGV-system som er fleksibelt med tanke på rekonfigurerbarhet. Man skal kunne endre på hele produksjonssystemet uten at det medfører store kostnader i tid eller penger. Når det gjelder dokkingen menes med høy grad av fleksibilitet at det skal være enkelt både å installere nye dokkingposisjoner og å flytte eksisterende posisjoner.

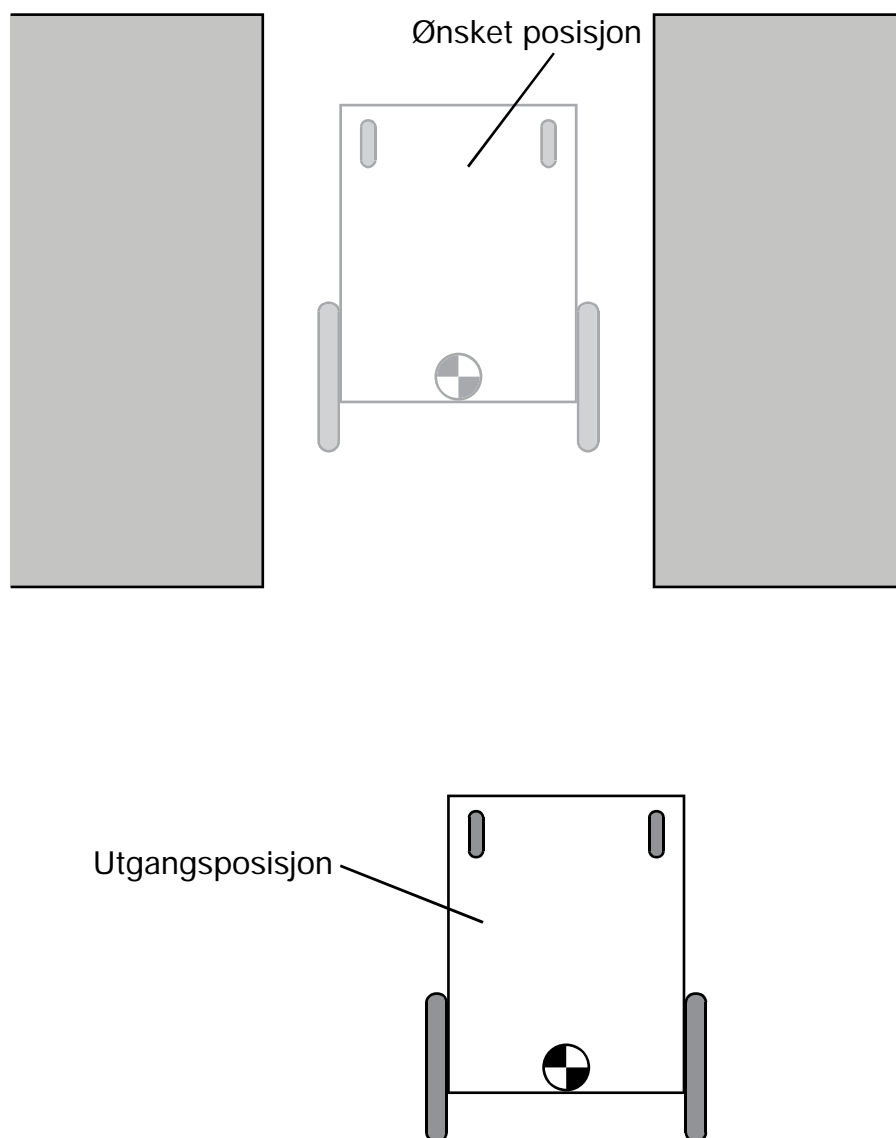
3.2.5 Bygge kvalitet

Et annet moment er at løsningen skal være holdbar over tid. Den bør kreve lite vedlikehold og ha en robust løsning som tåler bruk over tid. Dokken bør ha en høy oppetid og kunne vedlikeholdes raskt. Dette er i likhet med høy hastighet ikke et hovedfokus i denne oppgaven siden hovedutfordringen først er å utvikle en løsning som virker.

3.3 Forutsetninger

3.3.1 Dokkens utseende

Stedet der AGV-en skal dokke har form som en korridor uten en endevegg som vist i Figur 1. Det antas at korridoren har vegger som er like høye som selve AGV-en. AGV-en skal dokke på en posisjon som ligger midt mellom de to sideveggene.



Figur 1 – Illustrasjon av dokkens utseende

3.3.2 Eksterne forhold

- **Underlag**

Underlaget antas å være tilnærmet plant, og ha en jevnhet som et typisk støpt fabrikkgulv.

- **Krav til plass**

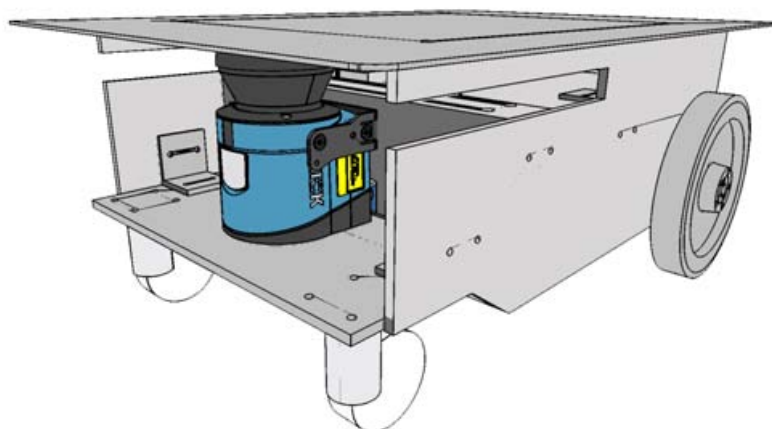
Det antas at det er tilstrekkelig med plass foran selve dokken til at dokkingen kan gjennomføres

- **Lysforhold**

Lysforholdene antas å være stabile, uten for eksempel overdrevne innslag av sollys. Dette kan være viktig dersom dokkingløsningen innebærer maskinsyn eller infrarøde sensorer.

4 Oppdatering av design

For å kunne teste det styresystemet som er under utvikling for instituttets AGV-system har det fra før blitt laget noen forsøks-AGV-er for å kunne eksperimentere med. Disse AGV-ene baserer seg på elektriske rullestoler som er ribbet for å kunne benytte motorer, hjul og styring. Det var imidlertid et ønske om å bygge en ny og oppdatert versjon av disse forsøks-AGV-ene. Først og fremst for å få nok en AGV å eksperimentere med, men også for å få en utforming som er mer robust og som i større grad benytter standard deler slik at den blir enklere å produsere flere av dersom det skulle bli nødvendig. Det var også et ønske om å benytte deler av høyere kvalitet enn tidligere, særlig når det gjaldt framdrift og motorkontroll. Det var også et viktig poeng at AGV-en skulle være enkel å montere og demontere slik at det skal være enkelt å bytte ut deler dersom man ønsker det på et senere tidspunkt. På denne måten kan man si at AGV-en til en viss grad er modulær i og med at ulike deler eller moduler lett kan legges til eller fjernes.



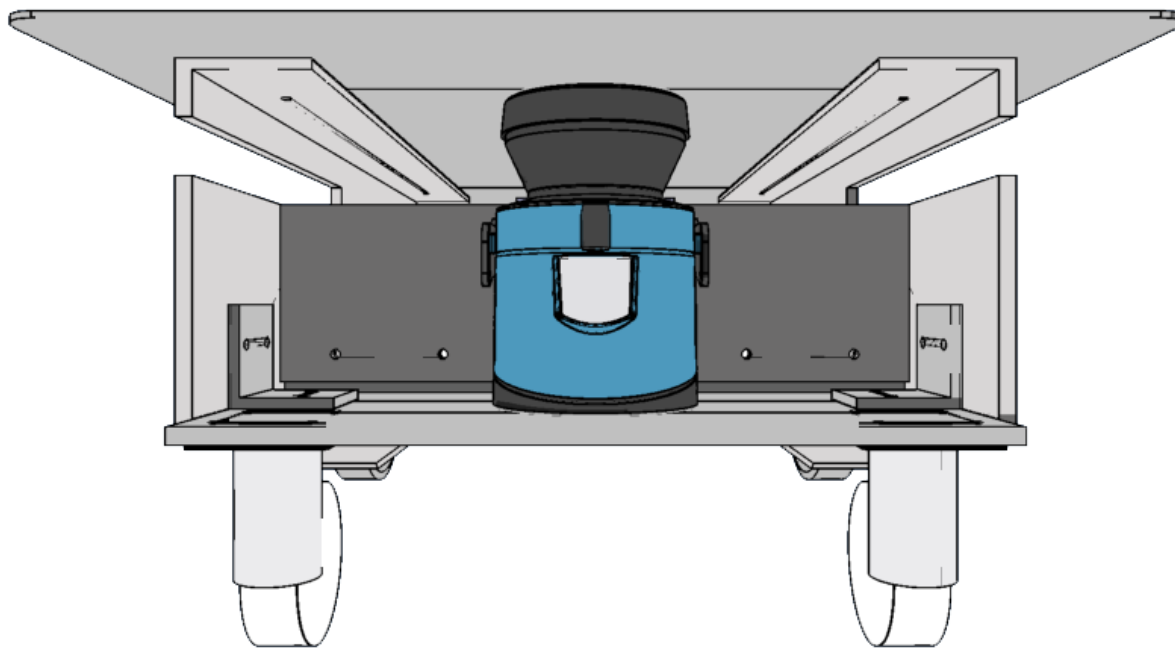
Figur 2 – Illustrasjon av AGV

AGV-en er, med en lasteflate på 450 x 650mm, relativt liten sammenliknet med de fleste konvensjonelle AGV-er som finnes på markedet i dag. Den er på denne måten egnet for å transportere små kasser med komponenter, mindre partier med komponenter og små eller mellomstore palletter. Målsetningen var at den skulle klare å bære en nyttelast 100kg, men dette har blitt nedjustert til ca. 80kg fordi det gjorde at utformingen ble betydelig enklere som beskrevet under. Aluminium ble, av flere grunner, valgt som det primære konstruksjonsmaterialet. Aluminium er et mye lettere materiale enn stål, som var alternativet. Den reduserte vekten dette medfører fører først og fremst til at den tillatte maksimale nyttelasten blir større. I tillegg vil en lettere AGV føre til et lavere forbruk av energi til forflytning.

Det primære bruksområdet for denne AGV-en er som forsøks-AGV i automatiseringslaboratoriet. Mye på grunn av dette har ikke mye vekt blitt lagt på å utvikle en AGV som kan bevege seg med høye hastigheter. Dersom man skal ha en AGV som skal benyttes til logistikkoppgaver i en produksjonsbedrift er lageret AGV-ene betjener som regel plassert en viss distanse unna selve produksjonsutstyret. Korridorer forbeholdt AGV-er forbinder lageret

med produksjonsområdet, og i disse korridorene beveger AGV-ene seg med hastigheter opp mot 5 m/s. En mobil robot som beveger seg med en slik hastighet vil imidlertid i stor grad påvirkes av dynamiske effekter. En konfigurasjon med drivhjulene plassert lengst bak er, som omhandlet i Kapittel 4.2.1, imidlertid ikke ideelt for bevegelse med høy hastighet.

Resten av dette kapittelet tar for seg detaljene rundt utformingen av den nye AGV-en.

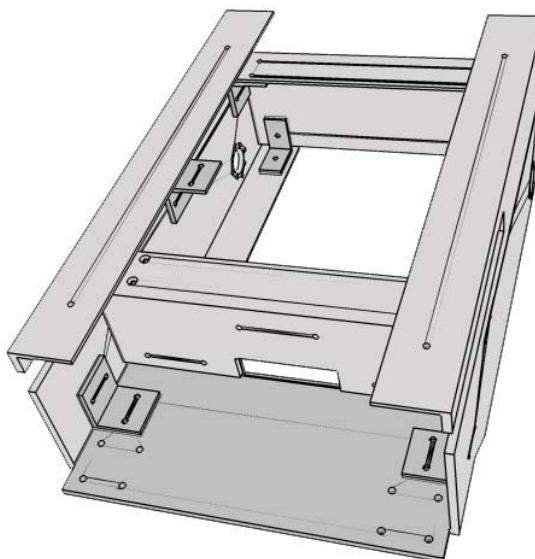


Figur 3 – AGV sett forfra

4.1 Ramme

Rammen er laget for å være så enkel og stiv som mulig. Den er laget av aluminiumprofil som er kappet og frest. Aluminiumprofilen som er benyttet er 200x60x4x8mm. Utformingen er som et chassis bestående av to vanger med tversgående stag. De ulike delene i rammen monteres sammen med vinkler laget av 50x50x5mm aluminiumvinkel og bolter.

De tversgående stagene er maskinert ut fra samme profil som vangerne er laget fra. Dette ble gjort siden det i denne omgang kun skulle produseres en AGV, og det da reduserte mengden materialer som det var behov for å kjøpe inn.

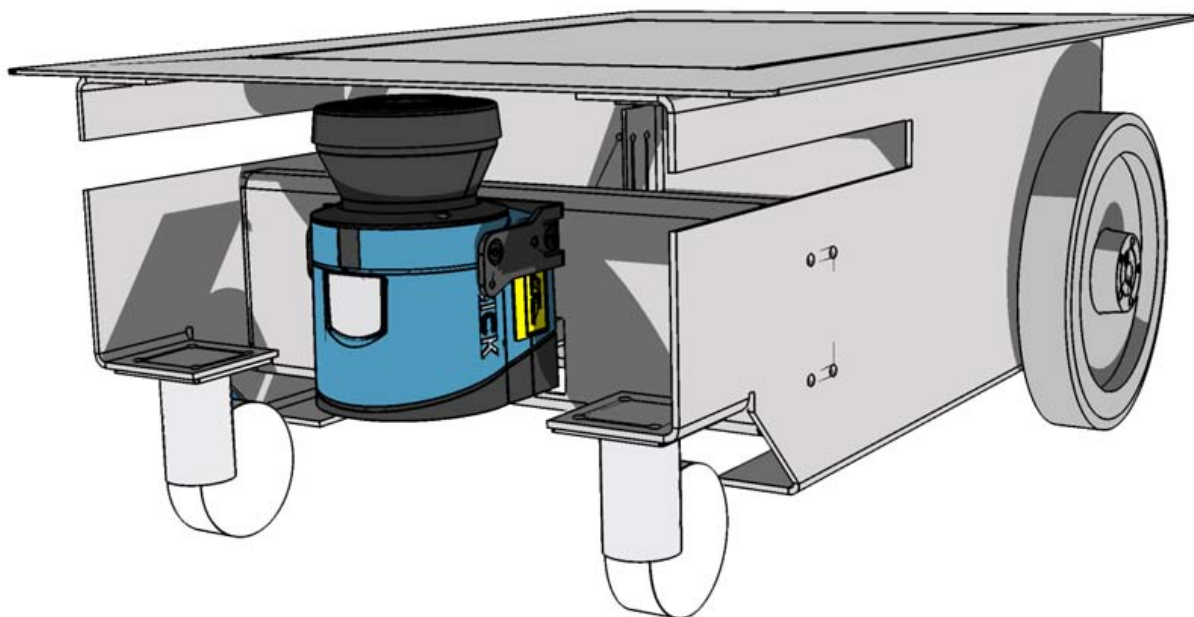


Figur 4 – Illustrasjon av ramme

Konseptet baserer seg på at giret med hjulakselen monteres direkte på innsiden av vangene. Dette krever at giret tåler en betydelig belastning i radiell retning, men det forenkler også konstruksjonen mye, siden det ikke krever ytterligere lagring av motorakslingen. Som nevnt førte denne løsningen til at maksimal tillatt last ble noe redusert siden det valgte giret ikke hadde høy nok tillatt radiell belastning.

4.1.1 Alternativt rammekonsept laget med tynnplate

Et alternativt konsept laget i tynnplate ble også utviklet. Dette konseptet har i stor grad samme enkle utforming, men har fordelen av at det krever færre deler. Dette fører til mindre maskinering og færre deler som må monteres. Produksjon i tynnplate ville sannsynligvis også vært en relativt kostnadseffektiv prosess. Designet i tynnplate baserer seg på at chassiset stanses eller laserskjæres, og deretter knekkes. Dette konseptet ble forkastet og ikke videre utviklet på grunn at det var ønskelig at instituttets eget verksted skulle benyttes til maskineringen. Verkstedet har ikke utstyret som er nødvendig for å produsere AGV-en etter dette konseptet på en effektiv måte. En illustrasjon av dette konseptet er vist i Figur 5.



Figur 5 – Illustrasjon av alternativt konsept i tynnplate

4.2 Hjul

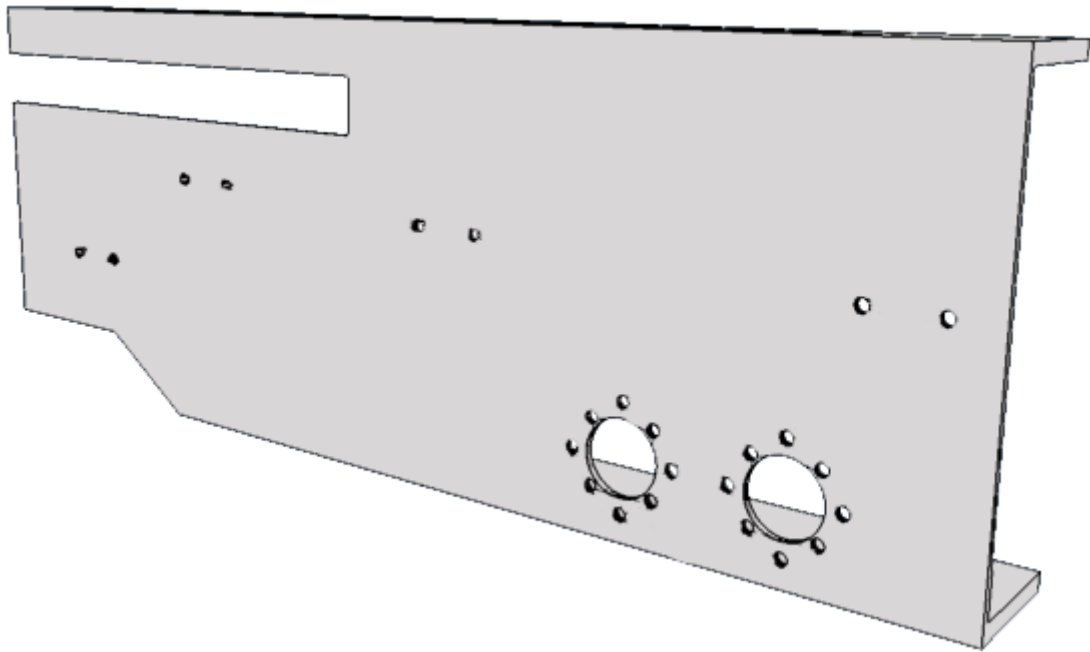
Hjulkonfigurasjonen er den samme som på de tidligere AGV-ene, med to drivhjul bak og to frie svivelhjul i front som vist i Figur 7. Ved bruk av to frie og to faste hjul regner man med maks belastning per hjul som totalvekt/3. Dette er fordi man aldri kan være sikker på at alle fire hjulene er i kontakt med underlaget til enhver tid. Dimensjonerer man for en tredjedel av vekten tar man hensyn til dette, og man tar hensyn til at underlaget ikke er helt plant.

Svivelhjulene er av typen 2470PJH075P40 fra produsenten Tente. De har en diameter på 75mm og en lastekapasitet på 75/150 kg (dynamisk/statisk). For dokumentasjon på svivelhjulene se Vedlegg 4.

De to drivhjulene har en diameter på 8 tommer (203mm), og kommer fra produsenten Colson. Disse er utstyrt med en overgang maskinert i aluminium for å kunne montere dem på akslingen som kommer fra giret.

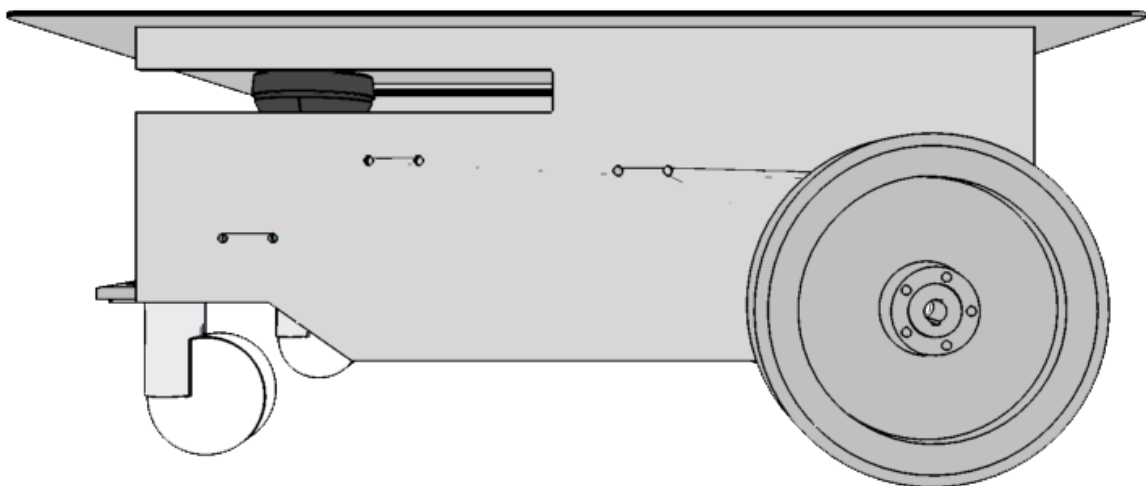
4.2.1 Plassering av hjul

AGV-en slik den ser ut nå har begge hjulene plassert lengst bak på vane. Når AGV-en beveger seg med høy hastighet kan dette imidlertid få negative innvirkninger på dynamikken. Ideelt sett burde drivhjulene vært plassert på midten av AGV-en med frihjul foran og bak som vist i Figur 14. Denne konfigurasjonen ville imidlertid vært mer utstabil, og vil kreve imidlertid bruk av fjæring på frihjulene for å unngå at AGV-en vipper mellom de to frihjulene. Mot slutten av prosjektet kom det et ønske om å ha muligheten til å flytte drivhjulene framover for å kunne eksperimentere med hvordan AGV-ens dynamikk påvirkes av posisjonen på drivhjulene. Derfor ble det boret opp et ekstra sett med festehull for motorene 85mm lenger fram på vane. En illustrasjon av venstre vane med denne ekstra posisjonen for montering av motor vises i Figur 6. Oppdaterte tegninger av vane finnes også i Vedlegg 5.



Figur 6 – Vange med to posisjoner for montering av motor

Bruken av denne fremre posisjonen for montering av motor krever at man er mer bevisst på at lasten ikke plasseres for langt bak på AGV-en. Noe som i så fall kan føre til at AGV-en tipper bakover.



Figur 7 – AGV sett fra siden med drivhjulene montert i bakerste posisjon

4.3 Framdrift

AGV-en er utstyrt med to motor-gir pakker som driver de to store hjulene. Disse er montert på samme akse og sørger slik både for AGV-ens framdrift og styring. Styringen foregår gjennom en kontrollsløfve med en enkoder og en motorkontroller for hver motor, noe som er illustrert i Figur 9.

4.3.1 Motor og gir

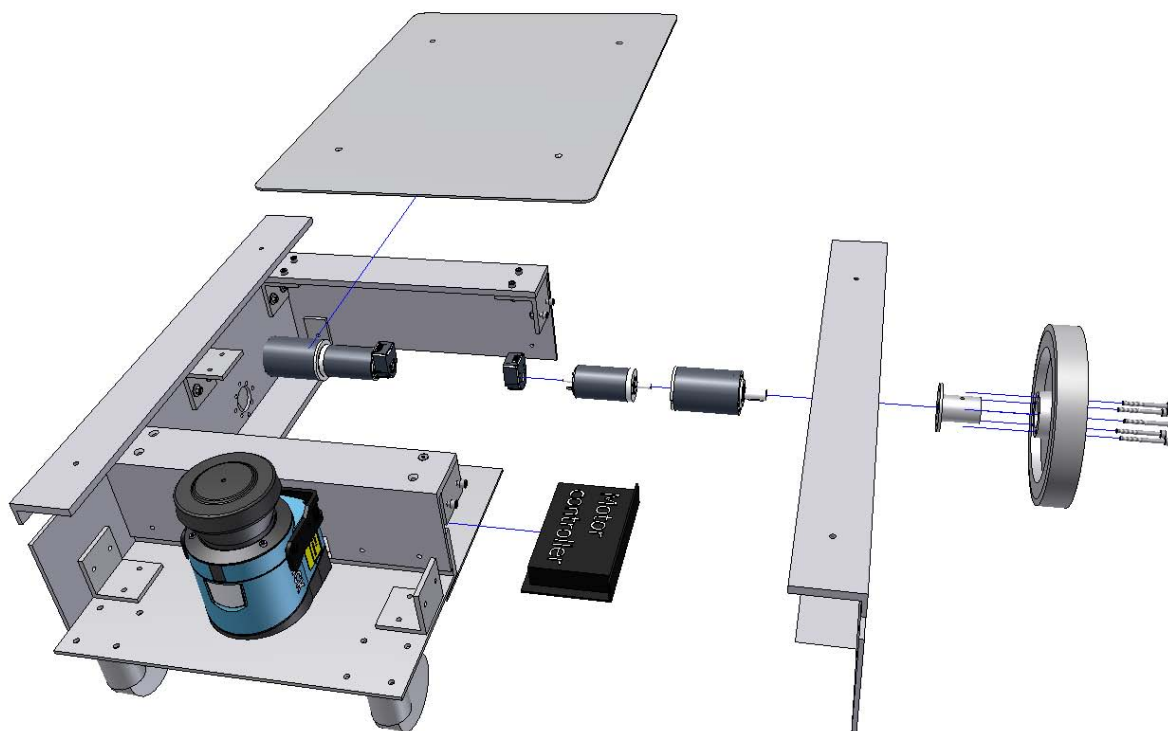
Motoren som benyttes er børstemotoren 148867 av typen RE 40 fra produsenten Maxon. Den tekniske dokumentasjonen for motoren finnes i Vedlegg 1. Motoren har en pålydende spenning på 24V. Den leverer et oppstartsmoment (stall torque) på 2280 mNm. De har en hastighet uten last på opp til 7580 omdreininger per minutt (rpm).

Giret leveres fra samme produsent og er av modellen 223093 av typen GP 52 C. Dette er et planetgir med utveksling på 81:1. Dette giret ble valgt på grunn av dets høye evne til å tåle radiell kraft, noe som gjør at man ikke behøver å lagre opp akslingen ytterligere når den har blitt montert i vangen. Dette medvirker til å gjøre hele konstruksjonen svært mye enklere. Den tekniske dokumentasjon for giret finnes i Vedlegg 1. Giret har en oppgitt maksimal effektivitet på 75 % noe som gir et oppstartmoment på 1710 mNm ($2280 \text{ mNm} \cdot 75 \%$). Giret har en anbefalt motorhastighet på $< 6000 \text{ rpm}$.

Dersom man kjører med den anbefalt maksimale motorhastighet på 6000 rpm vil man da man med en hjuldiameter på 203mm kunne oppnå den maksimale hastigheten v_{maks} .

$$v_{maks} = \frac{6000 \text{ r}/\text{min}}{60 \text{ s}/\text{min}} \cdot m \cdot (0,203 \cdot \pi) \text{ m}/\text{r} \cdot \left(\frac{1}{81}\right) = 0,787 \text{ m}/\text{s} \quad (4.1)$$

Dette er noe lavere enn den hastigheten man ønsker seg fra en AGV som skal benyttes til logistikkoppgaver i en produksjonssammenheng, men det er en tilfredsstillende hastighet for laboratoriebruk.

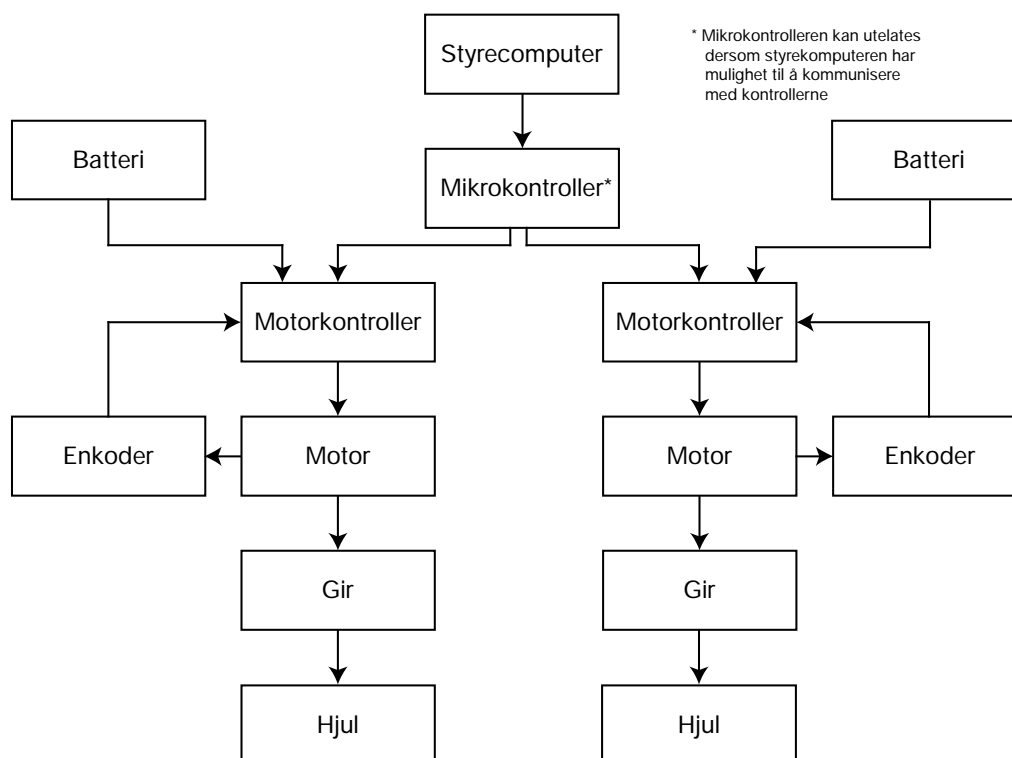


Figur 8 – Eksplodert figur av sammenstillingen av enkoder, motor, gir og hjul.

4.3.2 Enkodere

Enkoderene, modellen 228452 av typen MR 500IMP, leveres også fra Maxon. De har en oppløsning på 500 steg per omdreining. Enkoderene er koblet til akslingen på motoren, og på den måten vil det gi en effektiv oppløsning på 40500 steg per hjulomdreining (500 steg pr. motoromdreining * 81 motoromdreining pr. hjulomdreining).

Enkoderene er koblet til motorkontrollerne som styrer motorene. Motorkontrollerne styres av AGV-ens styrecomputer gjennom en mikrokontroller. I Figur 9 er framdriftssystemet beskrevet skjematisk.



Figur 9 – Skjematisk beskrivelse av framdriftssystemet

4.4 Motorkontrollere

Kontrollerne som benyttes er av typen EPOS 70/10 fra produsenten MAXON. Den har en mengde ulike operasjonsmodi, men mest interessant her er hastighetskontrollmodusen (Velocity mode) eller modusen for hastighetskontroll med forhåndsbestemt akselerasjonsprofil (Profile Velocity Mode), i begge sendes ønsket hastighet til kontrolleren, og den styrer motoren gjennom tilbakekoplingsløyfen med enkoderen. Hastighetsmodus med kontrollert akselerasjonsprofil vil være det beste valget siden det gir mykere start og stopp enn den vanlige hastighetskontrollmodusen som akselererer maksimalt til motoren har nådd korrekt hastighet.

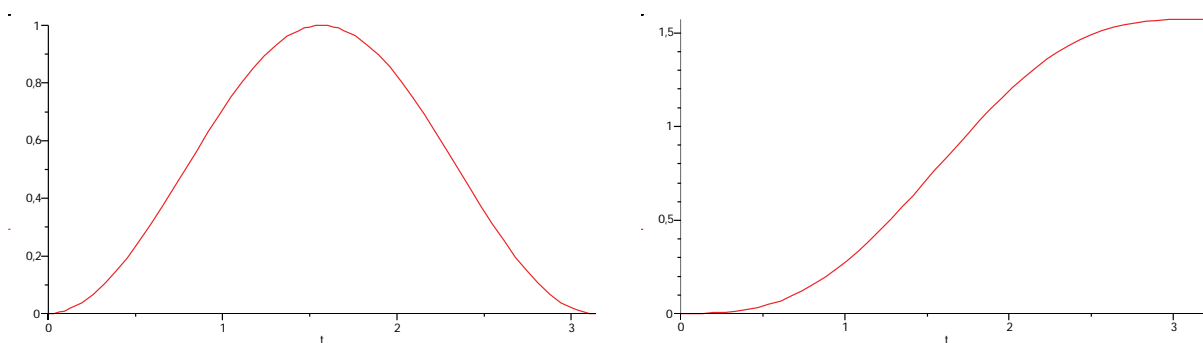
Når man bruker hastighetskontroll med forhåndsbestemt akselerasjonsprofil kan man velge mellom to ulike akselerasjonsprofiler. Den ene er en profil med konstant akselerasjon som vil gi en lineær hastighetsøkning, mens den andre er en akselerasjonsprofil som følger en funksjonen \sin^2 . Denne akselerasjonsprofilen er vist i Figur 10, som også viser korresponderende hastighetskurve. Det er denne akselerasjonskurven som benyttes på AGV-en.

Dersom akselerasjonen følger likningen:

$$a = \sin^2(t), \quad (4.2)$$

Vil hastigheten være:

$$v = \frac{t - \sin(t)\cos(t)}{2} \quad (4.3)$$



Figur 10 – Akselerasjon og hastighetskurve for akselerasjon som følger \sin^2

Denne funksjonen vil selvfølgelig måtte tilpasses med ønskede parametere, men den vil ha den samme prinsipielle formen for alle parametrene.

Motorkontrollerne er montert inne i AGV-en på en plate. Denne platen er montert slik at den lett tas av uten å først demontere motorkontrollerne. Dette letter monteringen av motorkontrollerne samtidig som det letter tilgangen til motorkontrollerne etter at de har blitt montert.

Fyldig dokumentasjon om motorkontrollerne finnes på den vedlagte CD-ROM samt på en egen CD-ROM levert av produsenten.

4.4.1 Kommunikasjon

For å kommunisere med kontrollerne benyttes en seriell RS232 tilkobling fra en PC eller en annen prosesseringsenhet, for eksempel en mikrokontroller, med mulighet for slik seriell kommunikasjon. Kommunikasjonen foregår gjennom tre ledninger, to for signal og en for jord. Kontrollerne har også mulighet for kommunikasjon via CANopen-protokollen men dette benyttes ikke. For mer informasjon om CAN se CAN in Automation (CiA) [4].

Sending av data til kontrolleren gjøres sekvensielt i et spesielt rammeverk som består av et hode (header), et datafelt av varierende lengde og til en slutt en CRC (cyclic redundancy check) for å kunne sjekke dataintegriteten.

Hode: Hodet består av to byte. Den første sier hvilken type rammeverk som sendes eller mottas, mens den andre inneholder lengden til datafeltet.

Datafelt: Datafeltet inneholder parametrene til den sendte kommandoen.

CRC: En 16-bits sjekksum som beregnes ut fra en algoritme kalt CRC-CCITT. Denne sjekksummen beregnes basert på innholdet i kommandoen som sendes, og på den måten kan kontrolleren sjekke at innholdet i den mottatte kommandoen er korrekt.

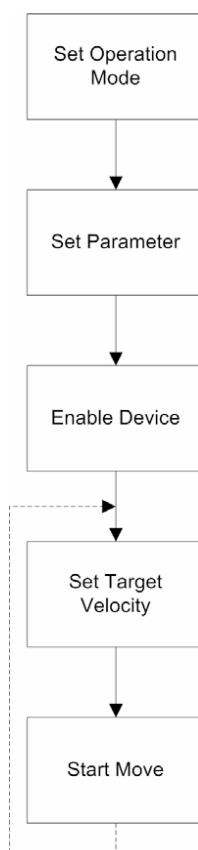
Dokumentasjon angående kommunikasjon finnes i kapittel 6 i motorkontrollerens dokumentasjon

4.4.2 Styring

Fra produsenten Maxon leveres et bibliotek med kommandoene som benyttes for å kommunisere med kontrollerne. Dette biblioteket er imidlertid kun beregnet for bruk på datamaskiner utstyrt med operativsystemet Microsoft Windows. Styringssystemet på AGV-ene er ikke basert på Microsoft Windows men på Linux. Det originale biblioteket som leveres med kontrollerne er derfor ikke kompatibelt med styresystemet på AGV-en. Det finnes imidlertid et bibliotek skrevet med tanke på Linux kalt libEPOS [2]. Dette biblioteket er utviklet av Marcus Hauser ved LSW Heidelberg. Biblioteket er skrevet i programmeringsspråket C. Biblioteket libEPOS er skrevet for posisjonskontrollmodus, men inneholder de grunnleggende funksjonene for kommunikasjon som beskrevet over, og kan derfor benyttes som grunnlag også for hastighetskontrollmodus eller hastighetskontrollmodus med kontrollert akselerasjonsprofil.

På bakgrunn av dette ble libEPOS utvidet med de funksjonene som er nødvendige for å styre motorene med hastighetskontrollmodus. I tillegg ble biblioteket klargjort for å kunne brukes sammen med programmeringsspråket Python som AGV-ens styresystem er skrevet i. Denne jobben ble i hovedsak gjort av Olivier Roulet-Dubonnet. Dette gjør at motorene kan kontrolleres fra Linux med programmeringsspråket Python. Biblioteket libEPOS finnes i Vedlegg 11 samt på den vedlagte CD-ROM. I tillegg kan nyeste versjon lastes ned fra SourceForge [2].

For å benytte hastighetskontrollmodusen med forhåndsbestemt akselerasjonsprofil er det en gitt rekkefølge kommandoene må sendes til kontrolleren. Denne rekkefølgen er illustrert i Figur 11. Først må operasjonsmodus gis (Set Operation Mode), deretter må parametere for oppsett sendes (Set Parameter). Disse parametrene setter blant annet akselerasjons- og retardasjonsprofilene og maksimal tillat hastighet. Deretter må en kommando sendes for å initialisere motoren (Enable Device). Disse tre trinnene må gjøres hver gang motoren initialiseres. Etter at disse trinnene er gjennomført går man som figuren viser inn i en løkke der man sender hastigheten man ønsker at motoren skal holde etterfulgt av en startkommando. Programmet vil her gå i løkke og man kan sende nye oppdaterte hastigheter, og kontrollerne vil oppdatere hastigheten når de får nye kommandoer. Når man ønsker å stoppe motoren kan det gjøres med en normal stoppkommando som gjør at motoren retarderer etter den definerte hastighetsprofilen eller det kan gjøres med en hurtigstoppkommando som gjør at motoren stanses så fort som mulig. Det anbefales ikke å benytte denne hurtigstoppkommandoen ved vanlig kjøring av AGV-en siden den vil føre til en meget brå stopp som vil være uheldig for de mekaniske delene, i tillegg til at det kan føre til at lasten forflyttes.



Figur 11 – Rekkefølge kommandoer ved hastighetskontrollmodus med forhåndsbestemt akselerasjonsprofil, hentet fra [5].

For å kunne følge denne sekvensen har funksjonene blitt implementert i libEPOS for å kunne kalles fra Python. Eksempel 1 viser hvordan man bruker biblioteket libEPOS (revisjon 5) fra Python. Eksemplet viser hvordan man setter opp kommunikasjonen mot en motorkontroller (kalt *left*). Motoren blir startet med en målhastighet på 4000 omdreininger per minutt (rpm). Målhastigheten holdes på 4000 rpm i tre sekunder før den reduseres til 3000 rpm. Til slutt stoppes motoren, og kommunikasjonen med motorkontrolleren avsluttes på korrekt måte. For hver av kommandoene som ligger i dette eksempelet sendes en rekke kommandoer til motorkontrolleren etter protokollen beskrevet over. Disse kommandoene befinner seg i bibliotekene `epos` og `test` som importeres helt først i eksempelet. Disse vil ikke bli omhandlet her men er gjengitt i vedlegget.

Eksempel 1 – Bruk av libEpos med Python

```
from time import sleep
import epos
import test

if __name__ == "__main__":
    left = epos.EPOS("/dev/ttyUSB0")
    left.openEPOS()
    try:
        test.init(left)

        left.moveWithVelocity(4000)
        sleep(3)

        left.setTargetVelocity(1000)
        sleep(3)

    finally:
        try:
            left.haltVelocityMovement()
        finally:
            left.closeEPOS()
```

Dersom det er behov for å styre motorkontrollerne fra Microsoft Windows finnes det biblioteker for å gjøre dette på CD-ROM som ble levert sammen med kontrollerne.

4.4.3 Automatisk innstilling av parametere for motorkontroller

Motorkontrolleren har tre ulike regulatorer. For for strømkontroll (Current Regulator) har den en regulator med proporsjonal + integral-regulator (PI-regulator). Den har også en proporsjonal + integral-regulator for hastighetskontroll (Velocity Regulator) og for posisjonskontroll (Position Regulator) har den en proporsjonal + integral + derivat-regulator (PID-regulator). Alle disse reguleterne har parametere som bestemmer virkningen av de ulike komponentene i reguleteren. Disse parametrene er avhengig av motor og oppsett av tilstøtende komponenter. For å justere disse parametrene finnes er medfølgende program som gjøre en automatisk søking (auto-tuning) for å finne parametere som er godt egnet for det oppsettet man har. Dette programmet kjører en serie av tester med motorene med ulike parametere og bestemmer ut fra disse testene hvilke parametere som er godt egnet. Instruksjoner om hvordan man utfører den automatiske innstillingen av parametere finnes i kapittel 6.4 dokumentasjonen til motorkontrolleren (se vedlagt CD-ROM).

4.4.4 Odometri

Innebygget i motorkontrollerne er en funksjon som gjør at man kan lese av enkoderens verdi. Denne funksjonen benyttes til å bestemme hvor langt AGV-en har beveget seg. Dette kalles, som beskrevet i Kapittel 0, odometri eller bestikknavigasjon. Siden enkoderene har en høy oppløsning per hjulomdreining gir en denne metoden en meget nøyaktig avlesning av hvor mange omdreininger motorakslingen har gjort. Nøyaktigheten i forhold til lengden AGV-en har beveget seg er imidlertid noe dårligere, noe som også forklares i Kapittel 0

4.5 Tilkoblingskabler

Kabler for sammenkobling av komponentene i drivsystemet; batterier, motorkontrollere, datamaskin, motorer og enkodere, kommer også fra produsenten Maxon. Kablene som leveres fra produsenten er alle tre meter lange, noe som fører til at det blir mye unødvendig kabel liggende inne i AGV-en. Dette fører til at det blir trangt inne i AGV-en i tillegg til at det øker vekten til AGV-en. Ved bygging av flere AGV-er anbefales det å kjøpe inn et sett med nødvendige tilkoblingsplugger som også leveres fra Maxon (ordrenummer: 276248), som kan benyttes til å lage egne kable med mer hensiktsmessig lengde.

4.6 Batterier

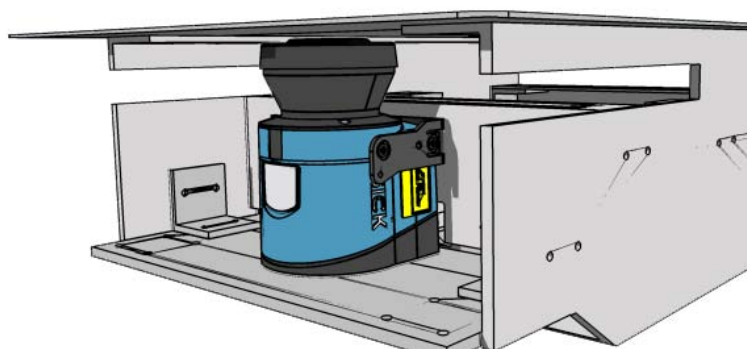
Batteriene som driver AGV-ens to motorer er såkalte LI-PO (Lithium-ion Polymer) oppladbare batterier. De gir en utspenning på 11,1V og har en kapasitet på 4000 mAh. To av disse er koblet i serie til hver motorkontroller. I tillegg benyttes et eget batteri for å drive områdesensoren. Batteriene har den fordelen av at de er meget små og lette, noe som sparer vekt og som gjør de lette å plassere inne i AGV-en. Batteripakkene er montert inne i AGV-en bak motorene med en borrelås som gjør at det er enkelt å ta de ut og sette de inn før og etter lading, noe som foreløpig må gjøres eksternt.

4.7 SICK områdesensor

Foran på AGV-en sitter det en områdesensor (range finder) fra produsenten SICK (vist i Figur 12). For mer om sensorens virkemåte se Kapittel 6.1.1. Områdesensoren har et synsfelt på 270°, og på grunn av dette har vangene på roboten utskårne spor, som vist i Figur 13, for at områdesensoren skal kunne sitte beskyttet mellom vangene og samtidig benytte hele sitt synsfelt. På grunn av dens utsatte posisjon foran på AGV-en og dens høye pris er områdefinneren montert på en slik måte at den er beskyttet i framkant. Dette gjør den mindre utsatt ved en eventuell kollisjon med et annet objekt. Dokumentasjon for områdesensoren finnes på den vedlagte CD-ROM. Områdesensoren kobles til datamaskinen om bord i AGV-en via Ethernet-porten på datamaskinen.



Figur 12 – SICK områdesensor (<https://www.mysick.com/saqqara/wrapper.aspx?id=im0025864>)



Figur 13 – Plassering av områdesensor

4.8 Datamaskin

Datamaskinen som sitter om bord må ha muligheten for å kjøre operativsystemet Linux. Den må ha minst én USB-port for å koble til en seriellport-utvider samt en Ethernet-port for å koble til områdesensoren. Datamaskinen må være utstyrt for trådløs nettverkskommunikasjon (WiFi) for å kunne kommunisere med det overordnede kommunikasjonssystemet som tar seg av navigasjonen av AGV-ene. Siden datamaskinen monteres inne i AGV-ens skrog bør den ha mulighet for å benytte en ekstern nettverksantenne som kan monteres bak eller på siden av AGV-en, siden mottaksforholdene inne i skroget mest sannsynlig er meget dårlige siden metallet i skroget vil blokkere de trådløse signalene.

4.8.1 Seriellport-utvider

Siden datamaskinen kommuniserer med de to motorkontrollerne med seriell kommunikasjon er den avhengig av å ha to seriellportere. De fleste av dagens datamaskiner har ikke en gang en slik seriellport. På grunn av dette benyttes en seriellport-utvider for å gi datamaskinen seriellportene den trenger. Seriellport-utvideren har fire seriellportere og kobles til datamaskinen med en USB-tilkobling.

4.9 Diverse

4.9.1 Hastighet

Basert på girets anbefalte innhastighet er den maksimale motorhastigheten AGV-en kan kjøre med er 6000 rpm. Dette gir AGV-en en maksimal hastighet på 0,79 m/s.

4.9.2 Pris

I Tabell 1 vises en oversikt over prisen på delene som er benyttet i AGV-en. Prisene er oppgitt uten merverdiavgift og frakt. Prisen på de fabrikkerte delene er ikke tatt med. Montasje er heller ikke tatt med.

Tabell 1 – Kostnadsoversikt

Beskrivelse	Pris per enhet	Antall	Sum	Sum (NOK)
Drivverk ⁴				
Motor – Gir - Enkoder	631,77 €	2	1263,54 €	9 982 kr
Kontroller	427,89 €	2	855,78 €	6 761 kr
Kabler	125,65 €	1	125,65 €	993 kr
				17 736 kr
Hjul				
Drivhjul	ca. 100kr	2	200 kr	200 kr
Frihjul	40 kr	2	80 kr	80 kr
				280 kr
Andre deler				
Seriellport-utvider	274 kr	1	274 kr	274 kr
Områdesensor	Rundt 20 000 kr	1	20 000 kr	20 000 kr
Datamaskin	Rundt 2000 kr	1	2000 kr	2 000 kr
				22 274 kr
Totalt				
Totalt				40 290 kr

⁴ Beregnet med en kurs på 1 € = 7,90 kr.

5 Positur og styring

Denne delen av oppgaven tar for seg teorien bak hvordan man beskriver posisjonen og positur til en mobil robot og to kinematikkmodeller for å beskrive forholdet mellom bevegelse og positur. En modell for banestyring blir også presentert.

Teorien som finnes i dette kapittelet er ikke begrenset til AGV-er, men gjelder generelt for såkalte mobile roboter. Mobile roboter er et samlebegrep som omhandler alle roboter med mulighet for bevegelse i omgivelsene. Begrepet mobile roboter er altså meget bredt, og omfatter roboter som går til lands, til vanns og i luften. AGV-er er altså en type mobile roboter. Mye av den teorien som omhandles i dette kapittelet er gyldig for alle mobile roboter, noe er begrenset til mobile hjuldrevne roboter, mens noe er begrenset til mobile hjuldrevne roboter med differensialdrift.

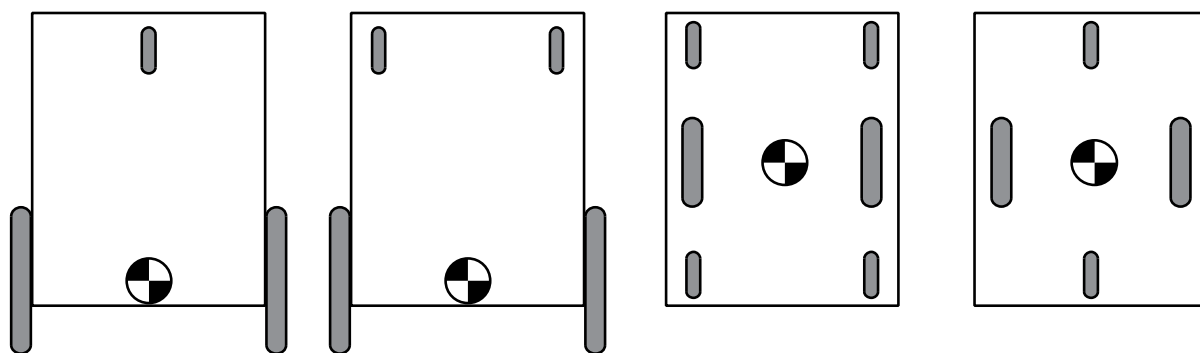
5.1 Kinematikk

Kinematikk er læren om legemers bevegelse i rom og tid uten hensyn til de krefter som fremkaller bevegelsen [6]. I dette tilfellet vil det si at kinematikken beskriver hvor roboten befinner seg basert på hvordan hjulene beveges.

I tilfellet med en AGV som beveger seg på gulvet i et produksjonslokale, og i de aller fleste andre tilfeller med mobile roboter som beveger seg langs et flatt underlag, vil det være naturlig å beskrive posisjonen kun i ett plan (xy-planet). Disse robotene beveger seg kun i dette planet og har ingen mulighet til å endre sin høyde i forhold til planet. Robotens posisjon i et globalt koordinatsystem er dens kartesiske koordinater i dette koordinatsystemet. I det todimensjonale tilfellet det er snakk om her vil det si koordinatene til roboten i et fast referansekoordinatsystem. Siden de ikke kan bevege seg i alle retninger er det for de aller fleste mobile roboter, i tillegg til posisjon, også nødvendig å angi robotens orientering. Derfor brukes en eller annen angivelse av orienteringen til roboten i forhold til en referanse. Til sammen kalles variablene posisjon og orientering robotens positur (pose).

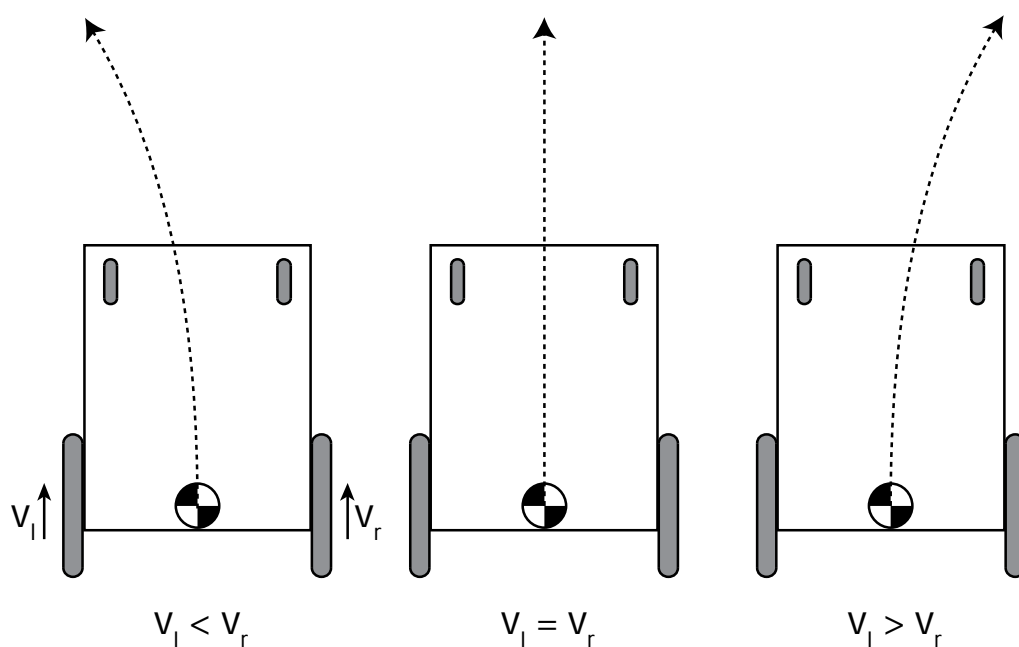
5.2 Differensialstyring

Differensialstyring er en veldig vanlig måte å styre mobile roboter som beveger seg innendørs. Det går ut på at man har to individuelt styrte drivhjul montert på samme akse. I tillegg til disse to drivhjulene har man en eller flere trinser eller svivelhjul som roterer fritt som vist som små hjul i Figur 14, der de store hjulene er drivhjul, mens de små hjulene er fritt roterende trinser.



Figur 14 – Ulike hjulkonfigurasjoner for mobile roboter med differensialstyring (løst basert på [7])

Når de to drivhjulene kjøres med samme hastighet vil roboten bevege seg rett, og dersom man kjører de i ulik hastighet (eventuelt ulik retning) vil roboten svinge i en bue mot det hjulet som beveger seg saktest. Dette er illustrert i Figur 15, der v_l og v_r er henholdsvis venstre og høyre hjuls hastighet. Drivhjulene kan enten være plassert midt på roboten med frihjul foran og bak eller i ene enden av roboten med frihjul i den andre enden. I det første tilfellet med drivhjulene i senter vil roboten kunne rotere rundt sin egen senterakse, og på denne måten har en meget høy manøvrerbarhet. I det andre tilfellet, der drivhjulene ikke er montert i senter, slik som på instituttets AGV, vil roboten alltid ha en viss svingradius.



Figur 15 – Illustrasjon av differensialdrift

5.3 Bestikknavigasjon (dead reconing eller odometri)

Det engelske begrepet dead reconing kommer opprinnelig fra sjøfart (bestikknavigasjon) og går ut på at man fra en kjent posisjon ved et tidligere tidspunkt benytter en kjent kurs og en kjent hastighet til å beregne hvor man befinner seg. I en mobil robot vil gjøres dette ved at man teller omdreiningene på drivhjulene (med kjent dimensjon) for å beregne tilbakelagt distanse fra et kjent punkt. Man benytter også kursen, enten fra styrehjul eller fra drivhjulene (dersom man har differensialstyring), og på den måten kan man regne ut et estimat av posituren til roboten. De aller fleste landbaserte mobile roboter bruker i dag denne metoden som en del av navigasjonen [8, s. 35].

I et generelt, ideelt tilfelle der man kan se bort fra feil vil man med en hastighetsvektor $\frac{d\mathbf{x}}{dt}$ ha posituren [9]:

$$\mathbf{x} = \int_{t_0}^{t_f} \frac{d\mathbf{x}}{dt} dt, \quad (5.1)$$

Der bevegelsen finner sted fra tidspunktet t_0 til t_f .

For en robot med differensialdrift som benytter en sensor som der man måler antall pulser med en enkoder slik som AGV-en gjør vil man kunne beregne endringen i variablene posisjon (x og y), og orientering (θ) med (antatt at det ikke sitter et gir mellom motor og hjul):

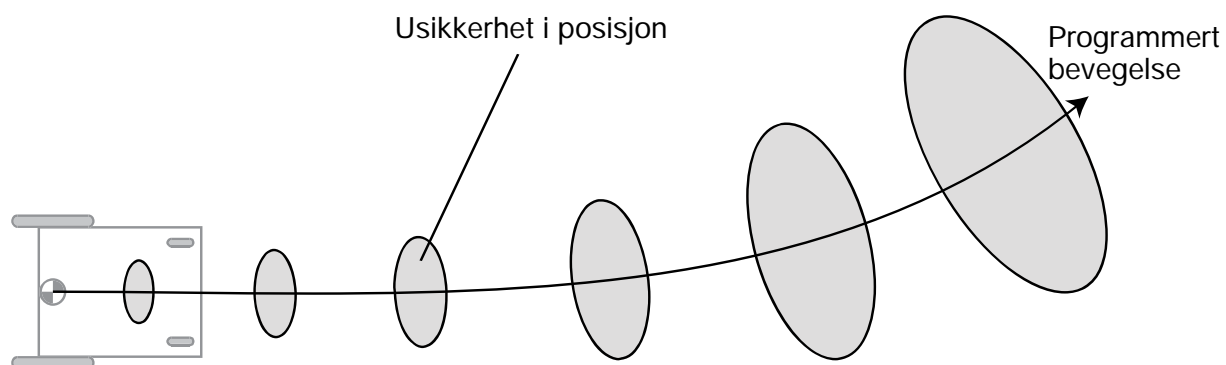
$$\Delta x = R \cos(\theta)(T_1 + T_2) \quad (5.2)$$

$$\Delta y = R \sin(\theta)(T_1 + T_2) \quad (5.3)$$

$$\Delta \theta = 2\pi \frac{T_1 - T_2}{T_R} \quad (5.4)$$

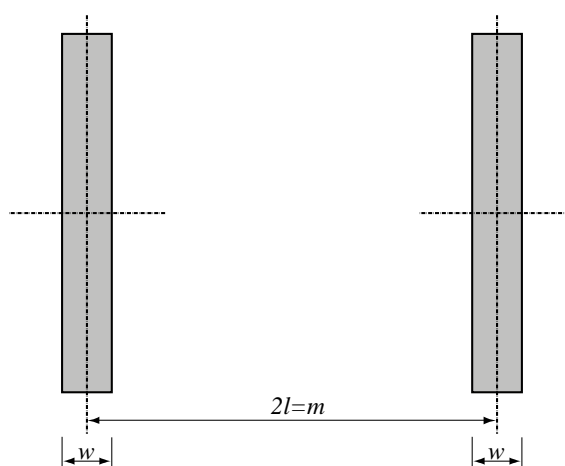
Der T_1 er enkoderens posisjon ved bevegelsens begynnelse og T_2 er enkoderens posisjon ved bevegelsens slutt. R er hjulets radius og T_R er antallet pulser fra enkoderen gjør på en hel hjulomdreining. Mer om dette beskrives i Kapittel 0 og Kapittel 0.

En slik metode der man kun benytter beregninger for å bestemme positur vil i alle reelle tilfeller ha en del unøyaktighet. Dette kommer som en naturlig følge blant annet av at sensorer har en begrenset nøyaktighet; man vil heller ikke kunne ta høyde for at et hjul for eksempel spinner på underlaget og på den måten registrerer unøyaktig tilbakelagt distanse. Det er også usikkerhet knyttet til hjulets eksakte diameter og hjulets gripepunkt som illustrert i Figur 17. Motoren vil heller aldri gå med helt nøyaktig den hastigheten som er programmert. Man har altså både systematiske og eksterne feil. Disse feilene vil forplante seg, og jo lenger roboten beveger seg jo større vil usikkerheten i posituren være. Unøyaktigheten er generelt større i beregningen av retningen, og usikkerheten i positur vil derfor få formen som en ellipse som vist (noe overdrevet) i Figur 16.



Figur 16 – Usikkerhet i posisjon (basert på [7, s. 191]).

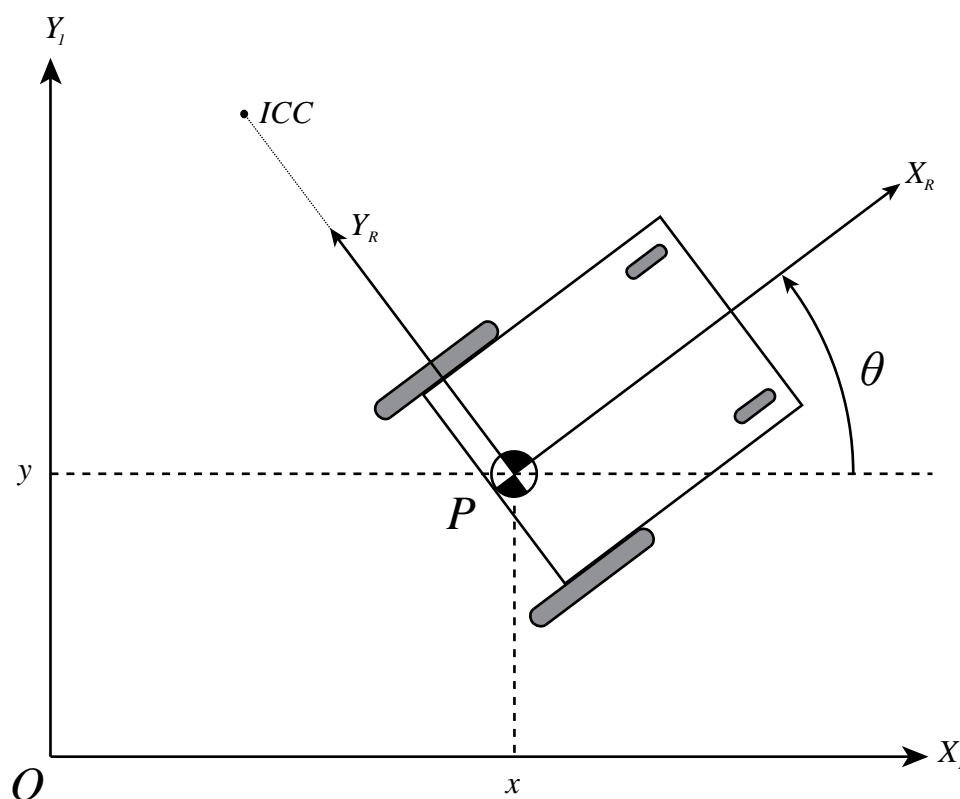
På grunn av den store usikkerheten som finnes i de fleste bestikknavigasjonssystemer suppleres de i de aller fleste tilfeller av et sekundært system for å bestemme posituren. Dette systemet oppdaterer posituren i forhold til en kjent referanse for å eliminere de akkumulerte feilene som har oppstått.



Figur 17 – Usikkerhet rundt det virkelige gripepunktet på drivhjulene gir en usikkerhet på $2w$ i hjulavstanden l . [8]

5.4 Kinematikkmodell I

Beskrivelsen og beregningen av kinematikken er hentet fra Siegwart and Nourbakhsh [7, kap. 3] samt Everett [8, kap. 2.3.1].



Figur 18 – Globalt og lokalt koordinatsystem. Basert på [7]

Aksene X_I og Y_I spesifiserer et koordinatsystem basert på et opprinnelsepunkt $O = \{X_I, Y_I\}$. Punktet på roboten angir dens referansepunkt, som er plassert midt på aksene mellom de to hjulene. Ut fra punktet P defineres aksene $\{X_R, Y_R\}$ på robotens chassis. Disse utgjør robotens lokale koordinatsystem. Posisjonen til P i det globale koordinatsystemet spesifiseres med x og y , og rotasjonen til det lokale koordinatsystemet i forhold til det globale koordinatsystemet angis med vinkelen θ . Vi kan definere robotens positur som en vektor med disse tre elementene:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (5.5)$$

Vi er også nødt til å overføre bevegelse i det globale koordinatsystemet over i det lokale koordinatsystemet. Dette gjør vi ved hjelp av en ortogonal rotasjonsmatrise:

$$R(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Denne matrisen kan vi så bruke for å overføre bevegelse i det globale koordinatsystemet til det lokale koordinatsystemet:

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \quad (5.7)$$

5.4.1 Kinematikk for en mobil robot med differensialstyring

For å beregne $\dot{\xi}_I$ behøver vi et sett likninger som beskriver hvordan robotens geometri og egenskaper påvirker dens bevegelser. For en robot med differensialstyring har vi to drivhjul med diameter r . Punktet P befinner seg som sagt på akse midt mellom disse to hjulene. Avstanden fra P til hvert hjul er lengden l . For gitte r , l , θ og rotasjonshastighetene φ_1 og φ_2 for de to hjulene, vil en kinematisk modell kunne finne hastighet i det globale koordinatsystemet:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \varphi_1, \varphi_2) \quad (5.8)$$

Robotens bevegelse i det globale koordinatsystemet transformert til det lokale koordinatsystemet vil ut fra likning (5.7) være: $\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$. Det som da trengs er påvirkningen til de to hjulenes bevegelse i det lokale koordinatsystemet. I følge Siegwart and Nourbakhsh [7, s. 61] vil ikke frie trinsehjul ha noen påvirkning på robotens kinematikk, man trenger derfor ikke hensynta disse ved beregning av kinematiske begrensninger.

Dersom man tenker seg at roboten er orientert slik at den beveger seg langs positiv X_R -akse slik som i Figur 18. Dersom et hjul ruller mens det andre står stille vil punktet P, siden det ligger midt på akse mellom hjulene, bevege seg med halvparten av hastigheten:

$$\dot{x}_{r1} = \frac{r\dot{\varphi}_1}{2} \quad (5.9)$$

,og

$$\dot{x}_{r2} = \frac{r\dot{\varphi}_2}{2}. \quad (5.10)$$

For en robot med differensialstyring vil man kunne legge sammen disse for å få \dot{x}_R -komponenten av $\dot{\xi}_R$. Siden bevegelse langs y -aksen aldri vil forekomme vil den andre komponenten, \dot{y}_R , nødvendigvis alltid være null.

Den siste komponenten, $\dot{\theta}_R$, må også beregnes. Også her kan de to hjulene behandles uavhengig og resultatene adderes. Rotasjon framover på det høyre hjulet (hjul 1) vil føre til en rotasjon ved punktet P mot klokka. Rotasjonshastigheten ω_1 ved punkt P kan beregnes siden hjulet beveger seg langs en sirkel med radius $2l$.

$$\omega_1 = \frac{r\dot{\phi}_1}{2l} \quad (5.11)$$

Den samme utregningen gjelder også for venstre hjul. Her vil imidlertid rotasjon framover føre til rotasjon med klokka.

$$\omega_2 = \frac{-r\dot{\phi}_2}{2l} \quad (5.12)$$

Når vi kombinerer alle disse resultatene får vi en kinematikkmodell for en robot med differensialstyring:

$$\dot{\zeta}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix} \quad (5.13)$$

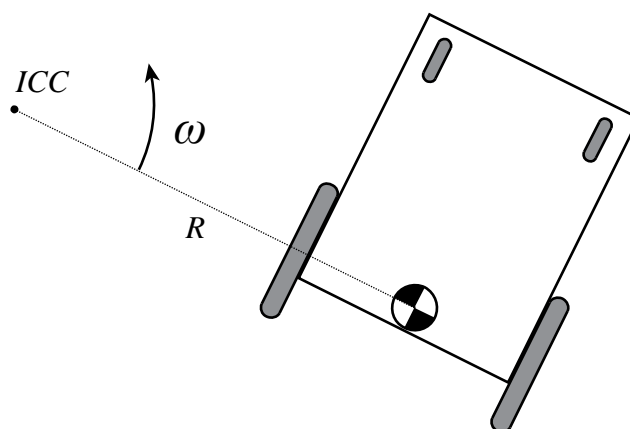
Vi trenger også $R(\theta)^{-1}$:

$$R(\theta)^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.14)$$

$$\dot{\zeta}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} + \frac{r\dot{\phi}_2}{2} \\ 0 \\ \frac{r\dot{\phi}_1}{2l} + \frac{-r\dot{\phi}_2}{2l} \end{bmatrix} \quad (5.15)$$

5.5 Kinematikkmodell II

Det er den andre av de to kinematikkmodellene som presenteres. Denne modellen er hentet fra Dudek and Jenkin [9].



Figur 19 – Rotasjon om ICC

På et tidspunkt der roboten svinger vil robotens venstre og høyre hjul ha samme vinkelhastighet i forhold til punktet ICC (instantaneous centre of curvature), og vi har:

$$\begin{aligned}\omega \left(R + \frac{m}{2} \right) &= v_r \\ \omega \left(R - \frac{m}{2} \right) &= v_l\end{aligned}\tag{5.16}$$

Der l er avstanden mellom de to hjulene og R er avstanden fra ICC til senterpunktet på den imaginære akse mellom de to hjulene. Det venstre hjulet beveger seg med hastigheten v_l mens det høyre beveger seg med hastigheten v_r . Dersom vi løser disse med hensyn på R og ω får vi:

$$R = \frac{m (v_l + v_r)}{2 (v_r - v_l)}\tag{5.17}$$

$$\omega = \frac{v_r - v_l}{m}\tag{5.18}$$

Vi vil her ha noen interessante tilfeller:

1. Hvis $v_l = v_r$ vil radien R være uendelig stor og roboten vil bevege seg i en rett linje.
2. Hvis $v_l = -v_r$ vil radien R være null og roboten roterer omkring punktet på akse mellom hjulene.
3. Når hastighetene har andre verdier vil roboten bevege seg i en kurve omkring et punkt (ICC) som ligger en avstand R fra punktet mellom hjulene.

Selv om det er relativt åpenbart er det verdt å merke seg at det ikke finnes noen kombinasjoner av v_l og v_r som vil føre til en bevegelse parallelt med akse som går gjennom hjulene.

På et gitt tidspunkt befinner roboten seg i punktet x, y i det globale koordinatsystemet med en rotasjon θ i forhold til x-aksen (se Figur 18).

$$ICC = [x - R \sin(\theta), y + R \cos(\theta)] \quad (5.19)$$

Ved tiden $t + \delta t$ er robotens positur gitt ved:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega \delta t) & -\sin(\omega \delta t) & 0 \\ \sin(\omega \delta t) & \cos(\omega \delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega \delta t \end{bmatrix} \quad (5.20)$$

Integrasjon av (5.20) fra en utgangspositur (x_0, y_0, θ_0) gjør det mulig å beregne robotens teoretiske positur ut fra kontrollparametrene v_l og v_r .

Generelt for en robot som kan beveger seg i en retning $\theta(t)$ med en hastighet $V(t)$ har vi:

$$\begin{aligned} x(t) &= \int_0^t V(t) \cos[\theta(t)] dt \\ y(t) &= \int_0^t V(t) \sin[\theta(t)] dt \\ \theta(t) &= \int_0^t \omega(t) dt \end{aligned} \quad (5.21)$$

Og for tilfellet med differensialstyring vil (5.21) gi:

$$\begin{aligned} x(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \cos[\theta(t)] dt \\ y(t) &= \frac{1}{2} \int_0^t [v_r(t) + v_l(t)] \sin[\theta(t)] dt \\ \theta(t) &= \frac{1}{l} \int_0^t [v_r(t) - v_l(t)] dt \end{aligned} \quad (5.22)$$

Dette er løsningen på framoverkinematikken som gir oss muligheten til å vite hvor vi er dersom vi vet hvor vi startet og hvilke hastigheter vi har hatt på hjulene. Mer interessant vil det være å finne modellen for den inverse kinematikken; altså hvordan vi må kontrollere roboten for å nå en gitt positur i det globale koordinatsystemet.

5.6 Invers kinematikk

Systemet med differensialstyring har såkalte ikke-holonomske begrensninger når man skal bestemme posituren. Ikke-holonomske begrensninger er definert som "ikke integrerbare lineære begrensninger i hastighetsrommet til et system" [10]. Dersom den skulle vært holonomisk måtte vi hatt muligheten til å endre alle frihetsgradene simultant og uavhengig. Ikke-holonomske roboter har begrensninger på hvordan man kan endre dens positur. For eksempel kan roboten aldri bevege seg parallelt med aksen gjennom drivhjulene.

Vi kan dog tenke oss at for å nå et gitt positur kan roboten først roteres slik at den peker mot den posisjonen den skal til, deretter kjører den rett framover til den når posisjonen og til slutt roterer den for å oppnå korrekt orientering.

Altså har vi tilfellet der $v_l = -v_r = v$, og roboten roterer rundt seg selv:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta + 2v \delta t / l \end{pmatrix} \quad (5.23)$$

Vi har også tilfellet der $v_l = v_r = v$, der roboten beveger seg i en rett linje:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x + v \cos(\theta) \delta t \\ y + v \sin(\theta) \delta t \\ \theta \end{pmatrix} \quad (5.24)$$

5.7 Dynamikk

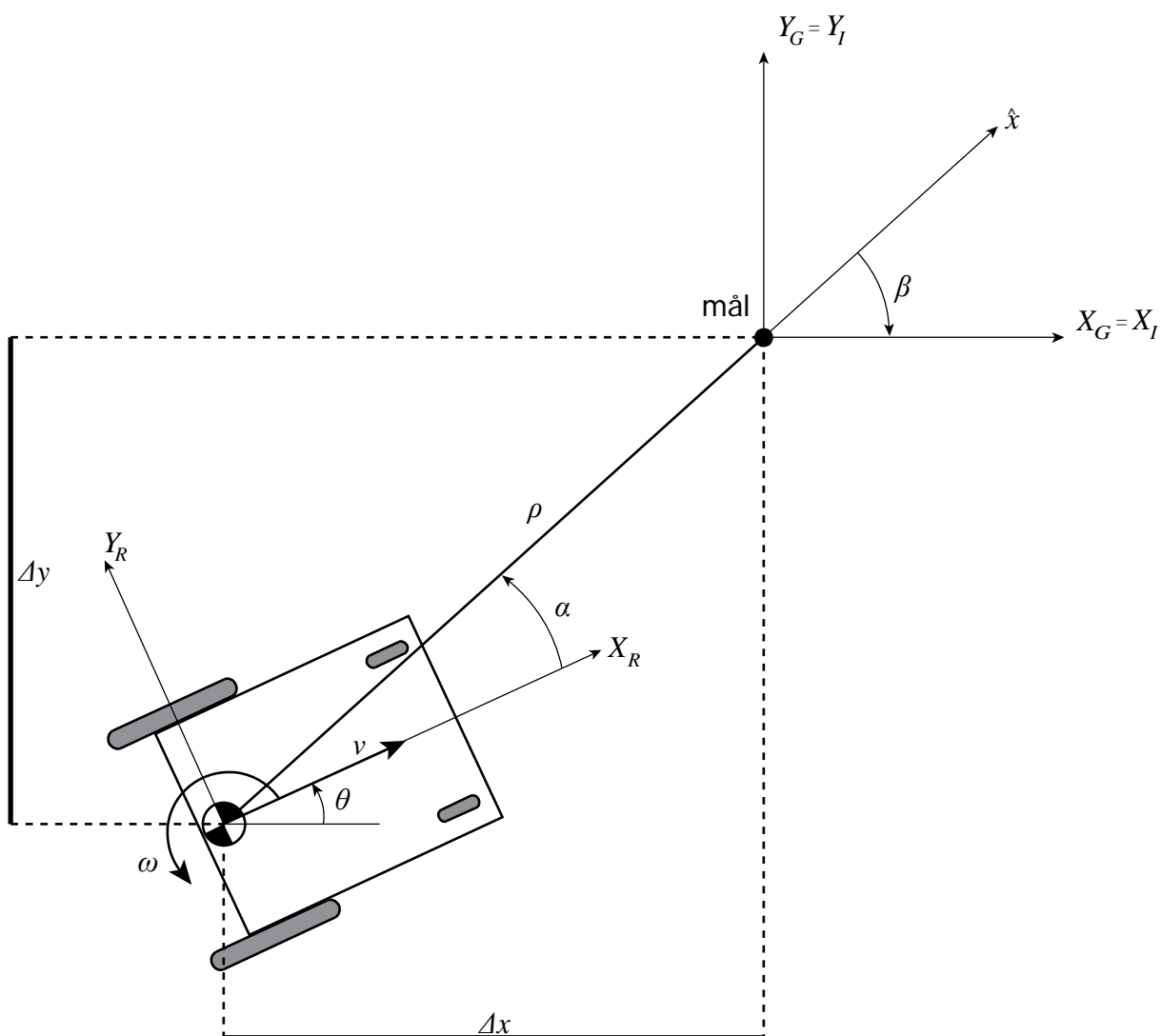
I motsetning til industriroboter, som ofte beveger seg opp mot maksimal hastighet, beveger mange mobile roboter, slik som AGV-en, seg med en relativt lav hastighet. Dette gjør at man sjelden behøver å ta hensyn til dynamiske begrensninger når man styrer robotene. På grunn av AGV-ens relativt begrensede hastighet er det mindre interessant å omhandle dens dynamiske egenskaper.

5.8 Banestyring

Når man kontrollerer en robot langs en definert bane er den beste måten å benytte en tilbakekoplingskontroller. Med en slik kontroller reduseres kontrollproblemet til å sette mellompunkter på banen som roboten skal bevege seg. I avsnitt 5.8.1 presenteres en løsning hentet fra Siegwart and Nourbakhsh [7] (basert på Astolfi [11] og Lee et al. [12]).

5.8.1 Banestyring, løsning fra [7]

Vi tar utgangspunkt i Figur 20 med en mobil robot i en vilkårlig positur, og med et definert mål. Vektoren som angir feilen i positur i robotens koordinatsystem $\{X_R, Y_R, \theta\}$ er $e = {}^R [x, y, \theta]^T$, der x, y og θ er robotens målkoordinater.



Figur 20 – Banestyring

Kontrollerens oppgave blir da å finne en matrise, K , hvis den eksisterer:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \end{bmatrix}, \quad (5.25)$$

med $k_{ij} = k(t, e)$, slik at når den kontrollerer $v(t)$ og $\omega(t)$:

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = K \cdot e = K \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad (5.26)$$

får feilen, e , til å gå mot null⁵.

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (5.27)$$

Vi antar at målet er i origo av det globale koordinatsystemet. Positurvektoren $[x, y, \theta]^T$ er heretter definert i dette koordinatsystemet.

Kinematikken til en robot med differensialstyring er beskrevet i det globale koordinatsystemet ved:

$${}^I \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (5.28)$$

der \dot{x} og \dot{y} er de lineære hastighetene i retningene X_I og Y_I i det globale koordinatsystemet.

Vi lar α beskrive vinkelen mellom X_R i robotens koordinatsystem og vektoren \hat{x} som forbinder senterpunktet mellom robotens hjule med målposituren. Hvis $\alpha \in I_1$ der:

$$I_1 = \left(-\frac{\pi}{2}, \frac{\pi}{2} \right), \quad (5.29)$$

kan vi gjøre transformasjonen til polarkoordinater:

$$\rho = \sqrt{\Delta x^2 + \Delta y^2} \quad (5.30)$$

$$\alpha = -\theta + \text{atan2}(\Delta x, \Delta y) \quad (5.31)$$

$$\beta = -\theta - \alpha \quad (5.32)$$

Dette gir oss på systembeskrivelsen på vektorform:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -\cos \alpha & 0 \\ \frac{\sin \alpha}{\rho} & -1 \\ -\frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (5.33)$$

⁵ Det er viktig å huske at $v(t)$ alltid peker i retningen X_R i robotens koordinatsystem på grunn av den ikke-holonomske begrensningen.

der ρ er avstanden fra senterpunktet mellom robotens hjul og målpunktet. Θ angir vinkelen mellom aksen X_R og aksen X_I . Hastighetene v og ω er henholdsvis tangentiell hastighet og vinkelhastighet.

Dersom på den annen side $\alpha \in I_2$, der

$$I_2 = \left(-\pi, -\frac{\pi}{2} \right] \cup \left[\frac{\pi}{2}, \pi \right), \quad (5.34)$$

noe som vil snu retningen til roboten ved å sette $v = -v$, får vi et system som beskrives på vektorform som:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ -\frac{\sin \alpha}{\rho} & 1 \\ \frac{\sin \alpha}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.35)$$

5.8.1.1 Bemerkninger til den kinematiske modellen i polarkoordinater

Den kinematiske modellen i polarkoordinater er gitt ved linkningene (5.33) og (5.35).

- Koordinattransformasjonen er ikke definert for $x = y = 0$, [siden determinanten i Jacobien-matrisen ikke er definert i dette punktet.]
- For $\alpha \in I_1$ peker robotens bevegelsesretning mot målet, for $\alpha \in I_2$ peker den vekk fra målet.
- Ved å definere bevegelsesretningen som den initiale posituren er det alltid mulig å ha $\alpha \in I_1$ ved $t = 0$. Dette betyr imidlertid ikke at α forblir i I_1 for hele tiden t . Derfor bør man lage kontrolleren slik at α er en del av I_1 for hele t når $\alpha(0) \in I_1$ (se stabilitetsspørsmål under).

5.8.1.2 Kontrollreglene

De signalene vi kontrollerer, v og ω , må nå velges slik at de flytter roboten fra sin opprinnelige positur, $(\rho_0, \alpha_0, \beta_0)$, til målposituren. Likning (5.33) har en diskontinuitet for $\rho = 0$.

Hvis vi ser på den lineære kontrollregelen:

$$v = k_\rho \rho \quad (5.36)$$

$$\omega = k_\alpha \alpha + k_\beta \beta \quad (5.37)$$

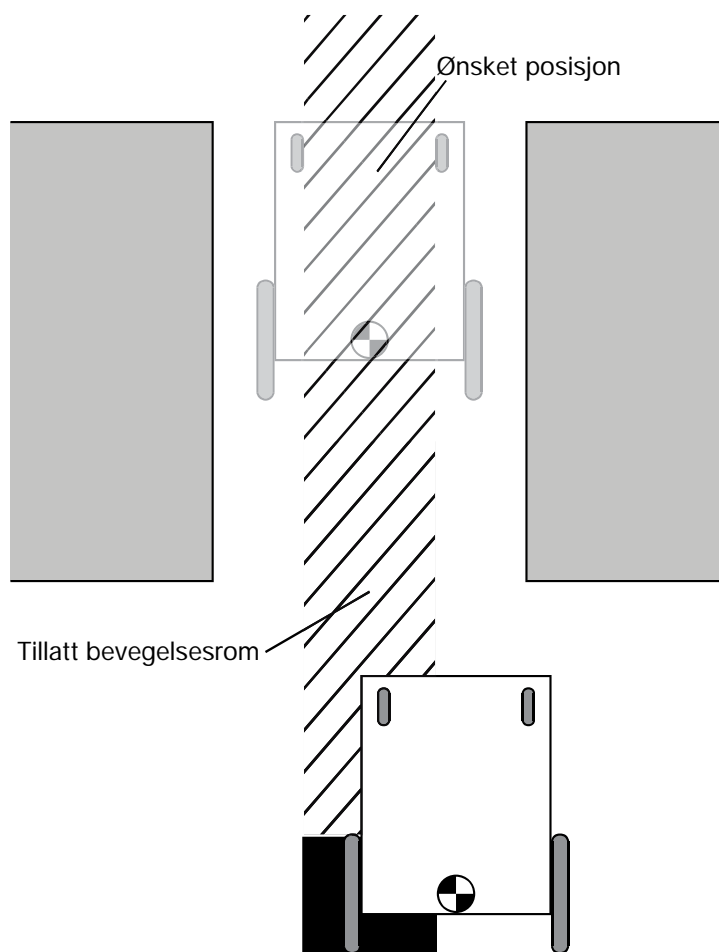
Får vi med likning (5.33) system med lukket reguleringsløyfe beskrevet av:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -k_{\rho}\rho \cos \alpha \\ k_{\rho} \sin \alpha - k_{\alpha}\alpha - k_{\beta}\beta \\ -k_{\rho} \sin \alpha \end{bmatrix} \quad (5.38)$$

Dette systemet har ingen singularitet for $\rho = 0$ og har et unikt likevektspunkt i $(\alpha, \beta, \rho) = (0, 0, 0)$. Det vil derfor kjøre roboten til denne posituren som også er målposituren.

5.9 Baneplanlegging

Baneplanlegging vil si hvordan man planlegger banen som roboten skal følge for å komme seg til ønsket punkt. Dette er særlig viktig når det finnes hindringer i veien som roboten må passere utenom eller når roboten må bevege seg inn mot målet i et begrenset bevegelsesrom. Dette siste tilfellet vil sannsynligvis være det relevante i dokkingssituasjonen. Der roboten må bevege seg inn mot endelig mål langs en linje eller en relativt smal korridor som illustrert i Figur 21. Figuren illustrerer det tillatte området som senterpunktet mellom hjulene kan bevege seg i. Dersom senterpunktet beveger seg utenfor dette området vil roboten kollidere med konstruksjonen som ligger rundt. For å oppnå dette må robotens bane planlegges slik at den hele tiden ligger innenfor dette området.



Figur 21 – Illustrasjon av begrenset bevegelsesrom for dokking

Planleggingen av banen roboten skal velge gjøres i de fleste tilfeller ut fra en rekke regler, og med en rekke begrensninger som for eksempel det skraverte bevegelsesrommet i Figur 21.

Mange komplekse modeller for baneplanlegging finnes (se for eksempel Dudek and Jenkin [9, Kap. 5.3] eller Siegwart and Nourbakhsh [7, Kap. 6]), men i tilfellet med dokking er det et relativt enkelt tilfelle. Dette isolerte tilfellet er omhandlet i Kapittel 7.2.2.

6 Metoder og sensorer

I dette kapitlet presenteres først en rekke metoder og sensorer som kan brukes eller brukes til å bestemme en mobil robot sin posisjon. En sammenlikning og vurdering av de ulike teknologiene blir presentert. Til slutt i kapitlet presenteres og vurderes de mest spennende konseptene som kan benyttes for å gjennomføre dokkingen av AGV-en.

6.1 Sensorer og målemetoder

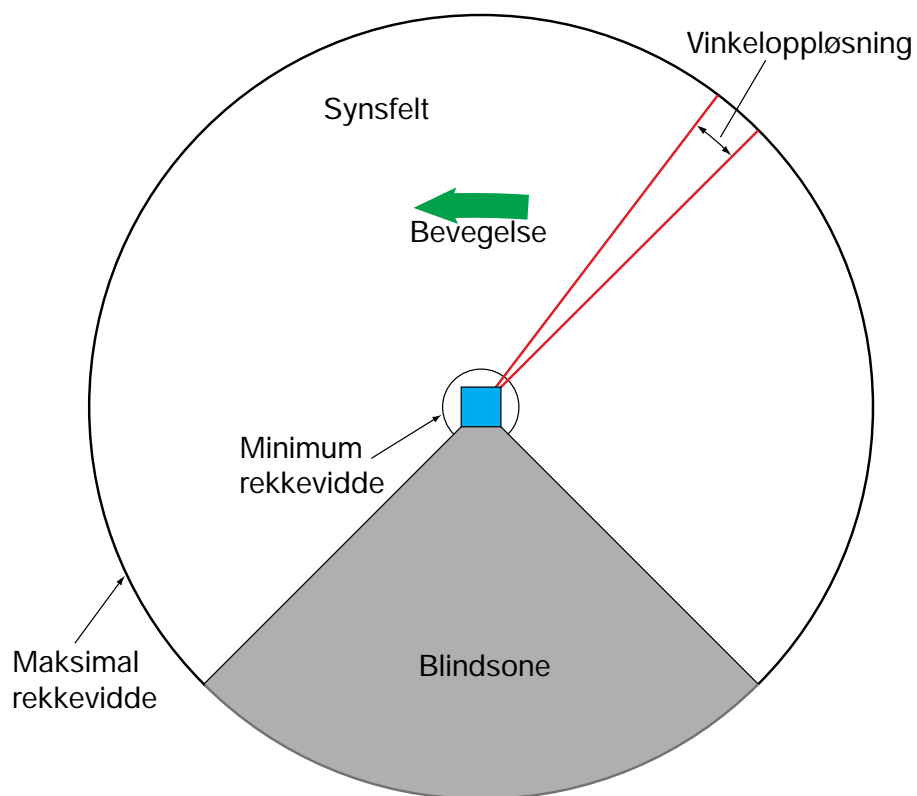
Det blir her gitt en oversikt over ulike sensorer og teknologier det vil være aktuelt å benytte til avstandsbestemmelse i forbindelse med dokkingoperasjonen. De ulike teknologiene presenteres, og deres virkemåte blir kort beskrevet. Fordeler og ulemper med de ulike teknologiene blir presentert.

6.1.1 Områdesensor

AGV-en er allerede utstyrt med en såkalt områdesensor (range finder) som benytter laserlys for å måle avstander i et felt i framkant av AGV-en (se Kapittel 4.7). Denne områdesensoren er montert på AGV-en for å benyttes til detektering av hindringer foran AGV-en. I tillegg til at man undersøker muligheten for å benyttes sensoren til navigasjon.

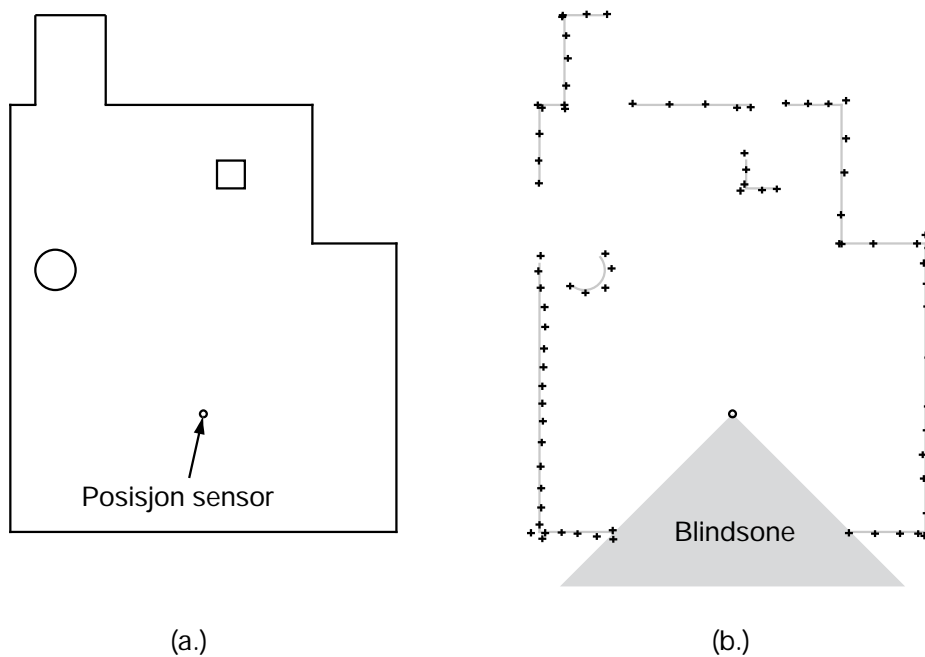
Områdesensoren virker ved at den sender ut en laserpuls som beveger seg gjennom luften til den treffer en flate. Noe av laserlyset vil reflekteres fra flaten, tilbake i den retningen det kom fra. Dette lyset vil bevege seg tilbake til sensoren som tar imot lyssignalet. Tiden lyset har brukt fra det ble sent til det kom tilbake, refleksjonstiden (time of flight, TOF), registreres. Siden laserlyset har en tilnærmet konstant hastighet i luft kan det ut fra denne tiden beregnes hvor langt laserlyset har beveget seg, og dermed også avstanden fra sensoren til flaten.

På vei ut av sensoren reflekteres laserstrålen et roterende speil. Ved å rotere speilet ev viss vinkel mellom hver måling får sensoren et bredt synsfelt. Modellen som sitter på AGV-en har et synsfelt på 270° som illustrert i Figur 22. Områdesensoren kan programmeres til å gjøre målinger med en vinkelopløsning på enten 0,25 eller 0,5 grader. Områdesensoren har en oppgitt maksimal måleavstand på 18m, men dette vil være noe kortere på AGV-en siden sensoren er montert nære gulvet.



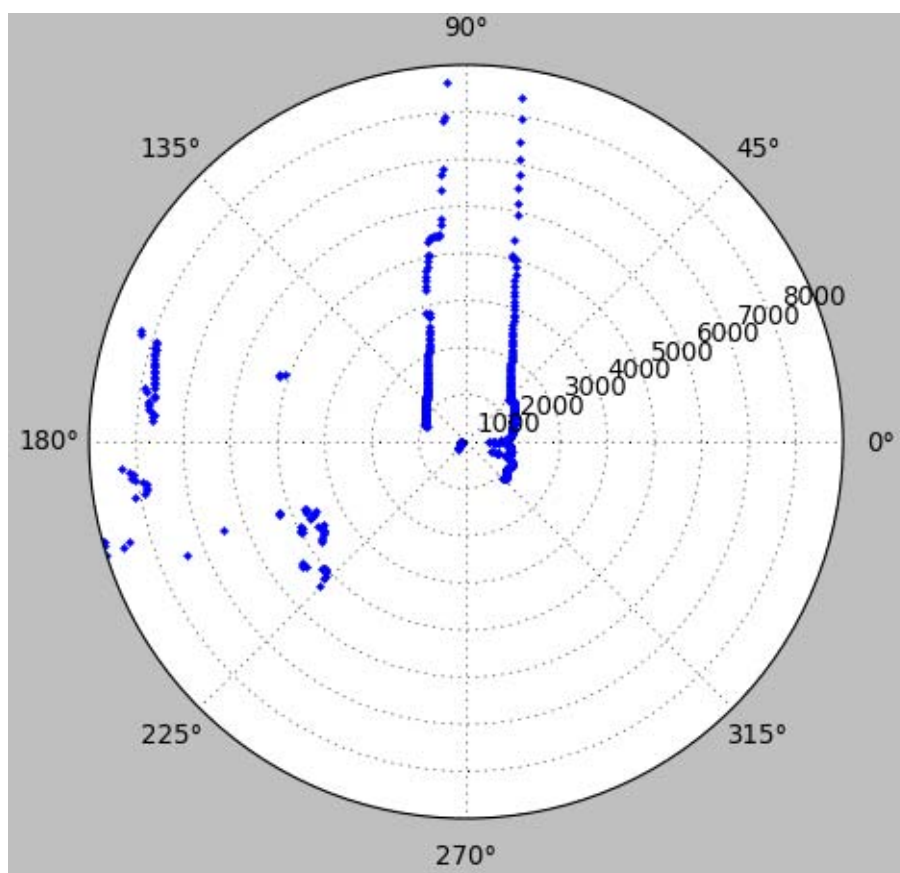
Figur 22 – Illustrasjon av synsfeltet til områdesensor

På denne måten produseres en punkter som representerer målingene sensoren har gjort i de forskjellige retningene i synsområdet. Sensoren returnerer en serie med datapunkter som inneholder vinkel og avstand, og på ut fra disse punktene kan man lage et plott av hvordan landskapet rundt AGV-en ser ut. I Figur 23 ser vi en representasjon av hva områdesensoren teoretisk kan se. Siden den kun sender ut laserstråler fra et punkt vil den naturlig nok ikke kunne se bak objekter. I figuren viser (a.) rommet sensoren befinner seg i og (b.) prinsipielt hvordan den resulterende punkttskyen fra områdesensoren kan se ut.



Figur 23 – Illustrasjon av teoretisk resultat fra områdesensor

I Figur 24 vises et plott av de faktiske dataene som blir hentet fra områdesensoren. Hvert av de blå punktene representerer en måling. I plottet er 90° rett fram på AGV-en. Avlesningen av sensoren ble gjort i automatiseringslaboratoriet på Institutt for produksjons- og kvalitetsteknikk ved NTNU. AGV-en stod rettet slik som det er indikert i det sammensatte panoramaet i Figur 25. Fotografiet har ikke 270° synsvinkel, men fotografiet er tatt i retningen som er indikert med 90° i Figur 24.



Figur 24 – Plottede verdier fra områdesensor

Som vi kan se klarer områdesensoren å se den smale gangen den har foran seg som også kan sees på fotografiet.

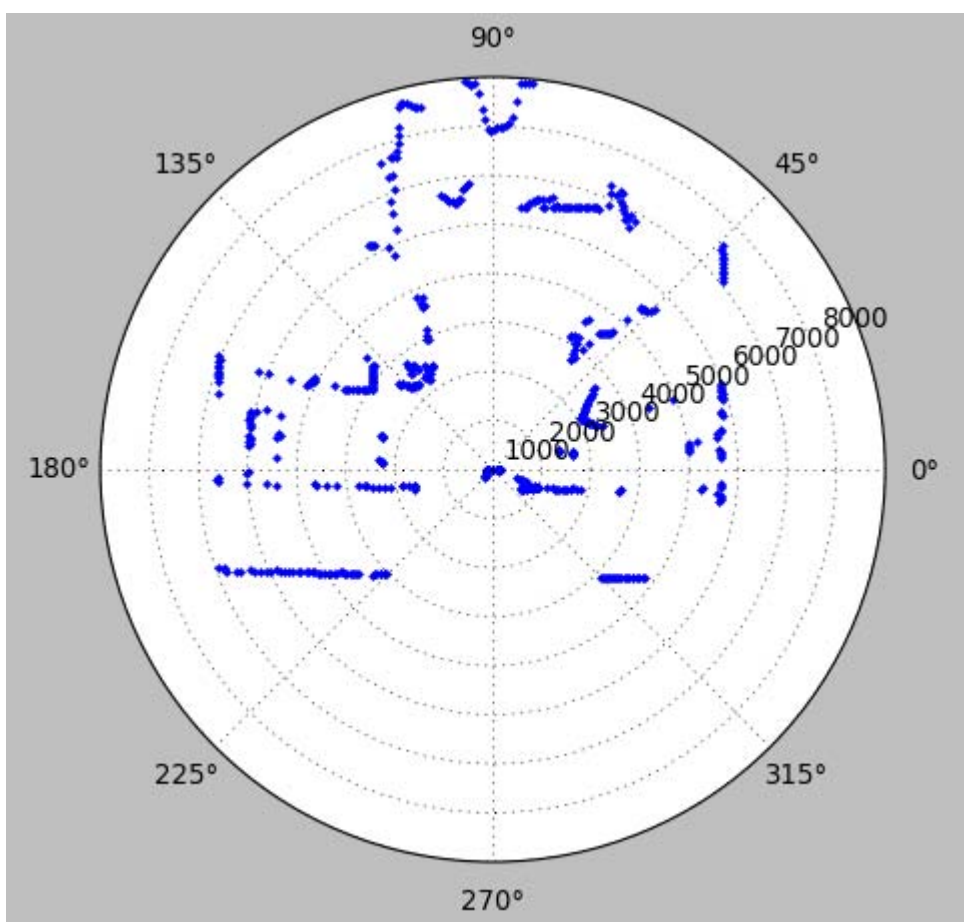


Figur 25 – Bilde av AGV-ens retning ved avlesning av områdesensor

I Figur 26 har AGV-en blitt rotert 90° mot klokka i forhold til forrige avlesning, og peker nå inn i selve laboratoriet. Dette er et mer kaotisk område med mange objekter spredd utover gulvet.

Gangen fra den første avlesningen kan fortsatt sees ut på hver side av plottet. Området foran AGV-en inneholder en rekke objekter, og dette gjør at plottet fra områdesensoren blir kaotisk. I Figur 27 vises et sammensatt panorama tatt fra områdesensorens posisjon i Figur 26. Panoramaet dekker i overkant av 180° av områdesensorens synsfelt på 270°.

Denne sensoren gir en helt unik mulighet til å kartlegge omgivelsene i et stort område rundt AGV-en på en rask måte. Hvor kaotisk dette plottet understreker imidlertid hvor stor utfordring det er å benytte denne type sensor til navigasjon. Plottet vil også se ulikt ut alt avhengig av hvilken positur man har, noe som ytterligere kompliserer situasjonen. Som nevnt er det lagt ned mye forskning akkurat på dette emnet. Se for eksempel Siegwart og Nourbakhsh [7, Kap. 5].



Figur 26 – Plottede verdier fra områdesensor



Figur 27 – Panorama av laboratorium

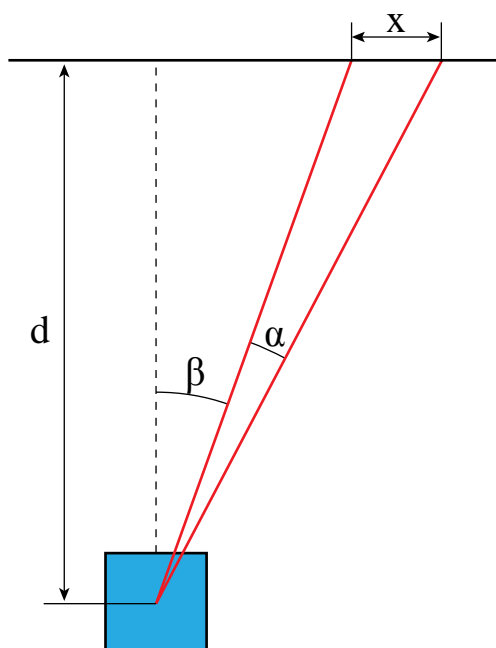
6.1.1.1 Tekniske spesifikasjoner

De mest interessante spesifikasjonene til områdesensoren er vist i Tabell 2 (hentet fra [13]).

Tabell 2 – Utdrag av tekniske spesifikasjoner for områdesensor, hentet fra [13]

	Minimum	Typical	Maximum
Functional data			
Scan angle			270°
Scanning frequency	25 Hz		50 Hz
Angular resolution			
With 25 Hz	0.25°		0.5°
With 50 Hz		0.5°	
Measurement error 1st reflected pulse⁶			
Systematic error		± 30mm	± 40mm
Temperature drift	0 mm/°C		0.32 mm/°C
Statistical error (1σ)		12mm	20mm

Det mest interessante her er hvor stor feil områdesensoren oppgis å ha. En systematisk feil på opp til ± 40mm i tillegg til en statistisk feil ved ett standardavvik opp til 20mm er betydelig. Dette selv om det ikke er oppgitt hvorvidt den statistiske feilen er avhenger av måleavstanden.

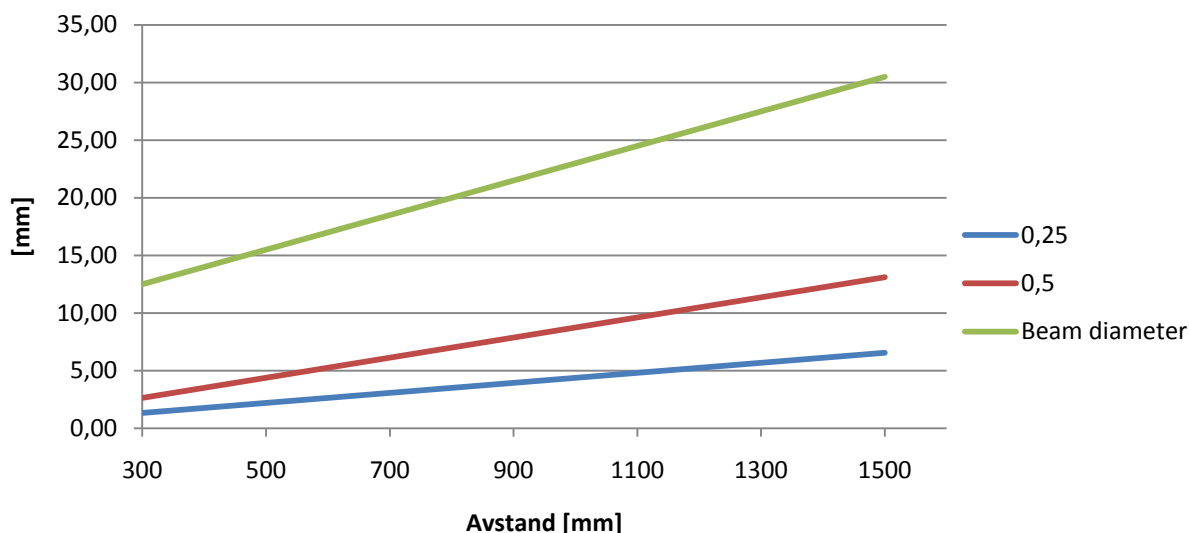


Figur 28 – Lineær oppløsning områdesensor

⁶ The time after the first reflected pulse from which measurement can be performed with full accuracy is dependent on the target that reflected the first reflected pulse.

Figur 28 viser en prinsippfigur av en områdesensor som måler en plan vegg. Distansen d er avstanden fra områdesensoren til vegg den måler. Vinklene α og β er henholdsvis vinkeloppløsningen og vinkelen fra linja fra områdesensoren som står vinkelrett på vegg.

I Figur 29 ser vi hvordan avstanden mellom målepunktene endres med avstanden når vinkelen β (se Figur 28) holdes lik null grader.



Figur 29 – Avstand mellom målepunkter og diameter på målepunkt (vinkel $\beta=0^\circ$)

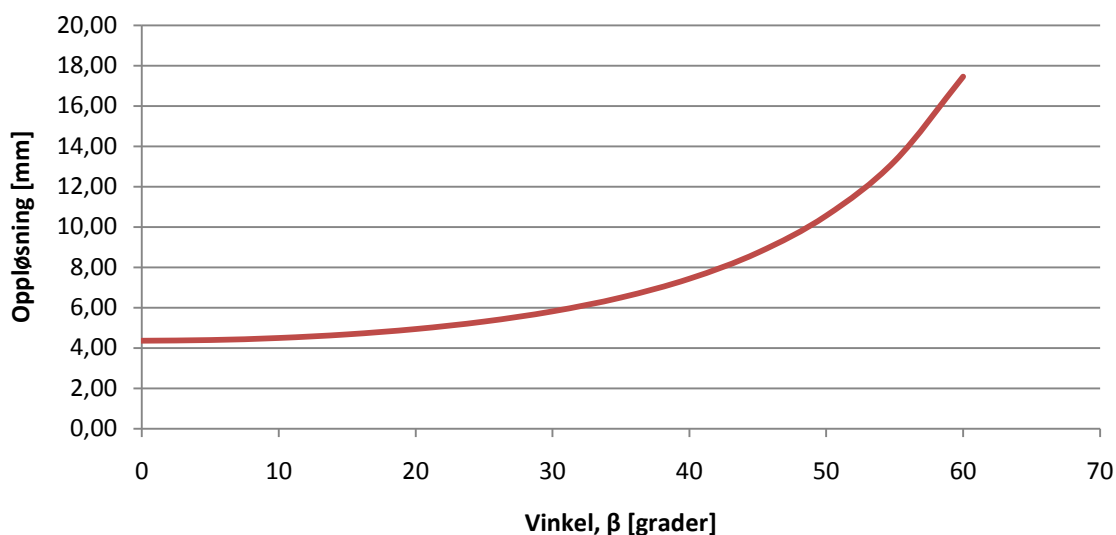
Tabell 3 viser en oversikt over avstanden mellom målepunktene ved ulike verdier av vinkelen β og avstanden d , som vist i Figur 28.

Tabell 3 – Lineær oppløsning ved ulike vinkler

Vinkel β	x [mm]	
	(d = 1000mm)	(d = 1500mm)
0°	4,36	6,54
5°	4,40	6,60
10°	4,50	6,75
15°	4,68	7,01
20°	4,94	7,41
25°	5,31	7,97
30°	5,82	8,73
35°	6,50	9,75
40°	7,44	11,15
45°	8,73	13,09
50°	10,56	15,84
55°	13,26	19,89

60°	17,45	26,18
-----	-------	-------

Forholdet mellom oppløsningen og vinkelen β for en distanse $d=1000\text{mm}$ er plottet i Figur 30.



Figur 30 – Oppløsning for en avstand, d , på 1000mm.

Det er ikke mange eksempler i litteraturen på bruken av slike områdesensorer spesifikt for dokking, men Gao et al. [14] demonstrerer en metode for dokking av en elektrisk rullestol. I dette eksemplet er imidlertid selve områdesensoren plassert på dokken og ikke på rullestolen.

6.1.1.2 Vurdering

Fordeler

- Det at det uansett sitter en områdesensor på AGV-en er åpenbart et poeng som taler for at den bør benyttes i dokkingen dersom det er mulig.
- Fremtidsrettet teknologi
- Påvirkes lite av støv/røyk eller lysforhold i rommet.
- Meget stort måleområde fra 300mm til 18m

Ulemper

- Denne typen sensor er relativt høy; prisen ligger i dag på rundt 30 000 NOK. Utbredelsen er imidlertid kraftig økende, og prisene er på vei ned.
- Antakelig vis for lav nøyaktighet til å bli brukt som eneste metode for nøyaktig dokking.
- Store utfordringer på programmeringssiden.
- Nærgrense på 300mm/500mm, noe som vil bety 100-300mm ut fra hver side på roboten.

Mye forskning er lagt ned for å kunne benytte områdesensorer som navigasjonshjelpemiddel på mobile roboter. Mye av denne teknologien kan sannsynligvis benyttes i dokkingproblemet. Problemstillingene er i stor grad like. De går begge ut på å bevege seg relativt til omgivelsene basert på data fra sensoren. Mange av applikasjonene som finnes i litteraturen benytter forhåndsdefinerte kart for å navigere etter. Man kan se for seg at dette er en mulig metode å

benytte også for dokkingen. Dersom man kan lage dokkingposisjonen på en slik måte at den lett kan kjennes igjen av områdesensoren kan man lagre denne geometrien som et kart og benytte dette til å navigere etter. For mer bakgrunn om hvordan man for eksempel ekstraherer linjer fra sensordata se [15] og [16].

6.1.2 Digitalt kamera / Maskinsyn

Bildegjenkjenning ved hjelp av digitalt kamera, ofte kalt maskinsyn eller vision-teknologi, er en stadig mer benyttet teknologi. Bildegjenkjenning benyttes i dag innenfor en rekke ulike industrier til oppgaver som kvalitetskontroll, sortering og komponentgjenkjenning og lokalisering for plukking med industrirobot. Bruk av maskinsyn i mobile roboter blir blant annen behandlet av Desouza and Kak [17] samt Bonin-Font et al. [18]. Barnes and Liu [19, Kapittel 2] gir også en god oversikt over dagens bruk av maskinsyn innen mobile roboter og forskning innen området. Et av de store problemene med denne teknologien har vært at det kreves mye prosesseringskraft for å behandle bildene som blir tatt, noe som har vært et hinder for å bruke denne teknologien i sanntidsstyring av systemer [20]. Prisreduksjonen på prosesseringskraft man har sett de siste årene har imidlertid gjort dette til en mer attraktiv teknologi.

Utstyrmessig er teknologien relativt enkel. Et ordinært kamera kobles til en datamaskin som tar imot og behandler bildene. Mer komplisert er det hvordan datamaskinen behandler bildene, og det er her den store utfordringen med denne teknologien. Det er anslått at opp mot 50 % av menneskehjernens kapasitet går med til å behandle visuell informasjon [9], og da sier det seg selv at det ikke er trivielt å reprodusere denne prosessen med en datamaskin. Store deler av den forskningen som er lagt ned innenfor området bildegjenkjenning er innefor filtrering av bilder og algoritmer for å kjenne igjen spesielle trekk i bildene på en pålitelig måte.

Et typisk digitalt kamera basert på en CCD-brikke (charge-coupled device) tar rundt 30 bilder i sekundet. CCD-brikken består av et rutenett med lysfølsomme elementer, såkalte piksler. Brikkene har fra 20000 til flere millioner piksler. CCD-brikken har enten ett lag, som vil gi bilder i gråtoneskala eller tre lag, som vil gi bilder med kombinasjoner av primærfargene rød, grønn og blå. Lysintensiteten blir målt for hver piksel og det er disse tallverdiene som lagres i bildefilen. Dersom det er mulig benyttes gråtoner framfor farger siden det gir en betydelig lavere mengde informasjon å prosessere siden hver piksel bare har en tredjedel av informasjonen ved bruk av gråtoner.

Maskinsyn kan benyttes til dokking på flere måter. Det er i hovedsak to prinsipielt forskjellige metoder som er aktuelle. Kameraet eller kameraene kan sitte på selve roboten (såkalt ego-docking) og observere verden fra robotens synspunkt. I den andre metoden sitter kameraet eller kameraene på faste posisjoner andre steder og observerer hvordan roboten beveger seg i arbeidsrommet (såkalt eco-docking).

Den første metoden der kameraet sitter på roboten baserer seg på gjenkjenning av såkalte landemerker. Landemerker er objekter eller kjennetegn i rommet eller landskapet som roboten beveger seg i. Eksempler på objekter kan være møbler, maskiner eller annet inventar som har en fast plass, mens andre kjennetegn kan være dørkarmer, vinduer eller liknende. Felles for landemerkene er at de har en fast posisjon, slik at roboten kan relatere sin positur i forhold til kjente punkter. Roboten vil plukke ut karakteristiske trekk i landskapet den ser rundt seg og sammenlikne disse trekkene med en database med landemerker. Ut fra dette vil den kunne

beregne seg fram til hvor den befinner seg i forhold til de ulike landemerkene. Landemerkene kan finnes i en fast database som ikke oppdateres, eller de kan ligge i en dynamisk database som oppdateres med erfaringsdata. Når man bruker erfaringsdata vil roboten legge inn nye landemerker som den ser hele tiden mens den kjører. De landemerkene som den ser mange ganger blir regnet som mer sikre enn andre. Dermed vil for eksempel en dørkarm som alltid befinner seg på samme sted være regnet som en bedre kilde fordi roboten har sett den mange ganger. Et menneske som beveger seg gjennom robotens synsfelt vil bli regnet som et dårlig landemerke siden roboten kun vil se det én gang. Denne typen teknologi bør kunne overføres til tilfellet med dokking der man ser for seg dokken i seg selv som et landemerke.

Landemerkene kan også være laget spesielt for å være lette å kjenne igjen med maskinsyn (kunstige landemerker). Ofte er dette objekter laget med det ene formålet å være lett gjenkjennelige for maskinsyn. Dette gjøres for at de lettere skal kjennes igjen og for at det skal være lettere å bestemme retning og avstand ut fra bildene som kameraet tar [21]. Flere eksempler på at dokkingproblemet er løst med slike kunstige landemerker finnes. Santos-Victor and Sandini [22] gjør både ego- og eco-dokking med kunstige landemerker. Low et al. [23] benytter også kunstige landemerker når de utvikler en dokkingmetode inspirert av honningbiers landingsmetode.

En meget enkel implementering av kombinasjonen av maskinsyn og landemerker presenteres av Cassinis et al. [24]. Dette systemet benytter et navigasjonsprinsipp bruk av seilere så lang tilbake som middelalderen. Det baserer seg på to lyskilder plassert bak hverandre, og som på den måte gjør det mulig å bestemme om man befinner seg på riktig innseglingskurs eller ikke ved å se om disse lyskildene sees direkte over hverandre.

I den andre metoden sitter kameraet på en fast posisjon som ser hvordan roboten beveger seg i landskapet. Et eller flere kameraer er koblet til en eller flere servere som behandler bildene. Roboten kjennes igjen på sin geometri eller ved kjennetegn som er montert på roboten for å lette oppgaven med å kjenne den igjen. Disse kjennetegnene er markører enten med et spesielt mønster eller farge. Aktive markører som sender ut lys (lysdioder) kan også benyttes som kjennetegn på roboten. Denne andre metoden er grunnlaget for styresystemet som allerede benyttes på AGV-systemet.

6.1.2.1 Vurdering

Fordeler

- Alt kan i prinsippet være landemerker, selv om noen vil være bedre egnet andre.
- Lav utstyrskostnad
- Mye forskning er allerede gjort på emnet
- Høy fleksibilitet
- Multifunksjonalitet, siden kameraet også kan benyttes til andre oppgaver utover dokkingen

Ulemper

- Kan kreve så mye prosessorkraft at datamaskinen om bord ikke kan prosessere bildene raskt nok for å bli brukt til sanntidsbevegelse.

- Påvirkes mye av lysforholdene i omgivelsene. Roboten vil ha problemer med å takle ulike lysforhold.
- Vil ha en begrenset oppløsning som ikke er tilstrekkelig i alle tilfeller
- Vil kunne ha problemer med å fokusere korrekt ved ulike avstander
- Lite robust system på grunn av typen algoritmer som benyttes for bildegjenkjenning

6.1.2.2 Navigasjon med maskinsyn

En mengde litteratur innen emnet maskinsyn for mobile roboter finnes allerede. Størsteparten handler om navigasjon ved hjelp av maskinsyn. Mye av den samme teknologien kan imidlertid sannsynligvis overføres til å brukes dokkingoperasjonen.

Desouza and Kak [17] deler maskinsyn for navigasjon av mobile roboter inn i to hovedgrupper, innendørs og utendørs navigasjon. I dette tilfellet er utendørs navigasjon ikke interessant og behandles ikke. Innendørs navigasjon deles videre inn i tre grupper:

- *Kartbasert navigasjon.* Systemer som benytter ferdiglagde geometriske modeller eller topologiske kart som beskriver omgivelsene.
- *Kartlagende navigasjon.* Systemer som benytter sensorer til å lage sine egne geometriske modeller eller topologiske kart og deretter benytter disse til navigasjon.
- *Kartløs navigasjon.* Systemer som ikke benytter noen eksplisitt representasjon av omgivelsene, men som kjenner igjen objekter i omgivelsene og navigerer ut fra disse observasjonene.

Av disse er det kartbasert navigasjon som er mest relevant å behandle siden en kjent dokk-geometri kan ses på som en type kart.

Kartbasert navigasjon ved hjelp av maskinsyn baserer seg på å forsyne roboten med et kart, en serie med landemerker den kan forvente å se på sin ferd. Systemets oppgave er da å søke etter og kjenne igjen disse landemerkene i bildene [17]. De nødvendige prosessstrinnene er i følge Borenstein et al. [25]:

1. *Samle sensordata.* For maskinsyn vil dette si å ta og å digitalisere bilder.
2. *Finne landemerker.* Dette vil ofte innebære kantgjenkjenning, utglatting, filtrering og segmentering av områder basert på forskjeller i gråtone, farge, dybde eller bevegelse.
3. *Finne samsvar mellom observasjon og forventning.* Systemet prøver å identifisere de uthentede landemerkene ved å søke i databasen (kartet) for å finne samsvar i basert på et gitt kriterium.
4. *Beregne positur.* Når samsvar har blitt funnet beregner systemet robotens positur som en funksjon av de observerte kjennetegnenes posisjon in databasen.

6.1.3 Induktiv linjefølgning

Dette er en av de mest utbredte metodene for styring av AGV-er. En elektrisk leder legges under underlaget og gjennom denne kjøres det vekselstrøm med en gitt frekvens. Dette produserer et magnetisk felt som detekteres av induktive sensorer på AGV-en. Flere ulike frekvenser kan benyttes samtidig for å bedre identifisere ulike løkker av ledere, og på denne måten kan AGV-

ene manøvreres mellom disse løkkene. Dette gjør at AGV-ene ikke er låst til å følge en fast rute hele tiden.

6.1.3.1 Vurdering

Fordeler [26]

- Utmerket pålitelighet
- Lang levetid
- Lite vedlikeholdsbehov siden ledningene ligger beskyttet under gulvet

Ulemper

- Meget høy installasjonskostnad.
- Høy kostnad og tidsbruk forbundet ved å gjøre endringer på linjene.
- Er sensitiv til forstyrrelser fra metall i nærheten av linja.
- Lav fleksibilitet i hvilke ruter AGV-ene kan ta.

6.1.4 Passiv linjefølgning

I motsetning til induktiv linjefølgning, innebærer passiv linjefølgning innebærer at en linje legges oppå underlaget der roboten skal kjøre. Roboten har en sensor som detekter linja. Sensoren har et noe bredere søkefelt enn selve linja, på den måten kan den detektere når linja svinger og styre roboten i den retningen. Passiv linjefølgning finnes i flere varianter: rent optiske linjer, såkalte kjemiske linjer, metallinjer og rutenett av linjer.

6.1.4.1 Vurdering

Fordeler

- Lav kostnad (i forhold til aktiv linjefølgning)
- Kan flyttes relativt enkelt.

Ulemper [26]

- Linjene på underlaget samt sensoren må holdes rene nok for å sikre stabil operasjon.
- Fokuslengden mellom sensor og linje må opprettholdes tilnærmet konstant.
- Linjene på underlaget kan lett skades som følge av normal slitasje fra trafikk.

6.1.5 Elektromagnetiske felt

Dokking ved hjelp av føling av elektromagnetiske felt benyttes blant annet av Husqvarna i selvgående gressklippere i Automower-serien [27].

Husqvarna har tatt patent på denne teknologien [28]. Som patentet beskriver virker denne metoden ved at en signalgenerator sender en strøm gjennom en plate med elektriske ledere som genererer et magnetisk felt. Strømmen genererer et magnetfelt som varierer i rom og tid. Roboten måler dette magnetfeltet ved hjelp av en sensor, et magnetometer, og styres etter hvordan den oppfatter magnetfeltet.

I praksis ligger det foran dokkingstasjonen en plate på ca. 1000mm x 500mm x 5mm som inneholder de elektriske lederne.

6.1.5.1 Vurdering

Fordeler

- Dokken kan lett flyttes
- Billige deler

Ulemper

- Vil sannsynligvis påvirkes meget lett av eksterne magnetfelt fra maskiner, roboten selv eller andre installasjoner i nærheten av dokkingstasjonen
- Patentert teknologi

6.1.6 Induktive sensorer

Induktive sensorer er sensorer som kan detektere metalliske objekter uten fysisk kontakt. De benyttes i hovedsak til detektering av objekter, men kan også benyttes som relativt nøyaktige distansesensorer på korte avstander. Eksempelvis leverer produsenten Festo en sensor, med et måleområde fra 0 til 20mm fra sensoren, som måler med en oppløsning på 0,005mm og en repeterbarhet på 0,3mm. Dette er den sensoren fra Festo som har størst måleområde.

I denne sammenhengen er det mest interessant å benytte induktive sensorer for å kunne finjustere posisjonen. Dersom man har to sensorer kan man også finjustere vinkelen.

6.1.6.1 Tekniske spesifikasjoner

Induktive sensorer leveres fra en rekke ulike leverandører. Som eksempel her brukes to av FESTO sine sensorer (M18 og M30).

Spesifikasjoner:

Tabell 4 – Tekniske spesifikasjoner induktive sensorer

Modell	M18	M30
Måleområde [mm]	0..10	0..20
Oppløsning [mm]	0,002	0,005
Retpetisjonsnøyaktighet [mm]	0,3	0,3
Repetisjonsnøyaktighet under konstante forhold [mm]	±0,02	±0,05

Som vi ser har disse et meget begrenset måleområde, men en høy nøyaktighet.

6.1.6.2 Vurdering

Fordeler

- Høy nøyaktighet
- Relativ lav kostnad

Ulemper

- Meget begrenset måleområde (opp til cirka 30mm)
- Krever overflate av metall å måle imot i dokkingposisjonen

6.1.7 Ultrasoniske sensorer

En vanlig ultrasonisk sensor har et stort måleområde, typisk fra et par centimeter opp til flere meter. Nøyaktigheten oppgis relativt, og kan komme så lavt som 0,05 % for høypresisjons-sensorer [29]. For en avstand på 1000mm vil dette tilsvare 0,5mm.

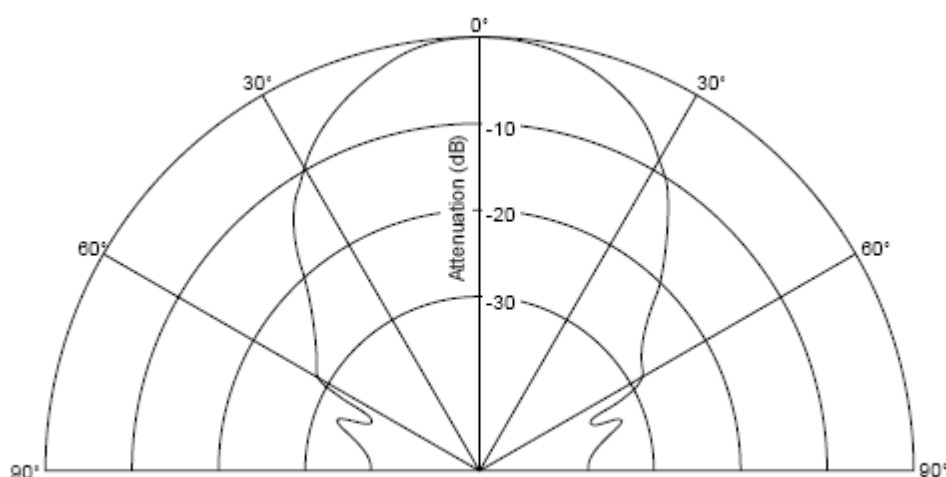
Et eksempel på en mer "standard" ultrasonisk sensor viser at nøyaktigheten typisk er noe dårligere. SICK oppgir nøyaktigheten på sin UM18-51112 til å være mindre enn 2 %:

Tabell 5 – Tekniske spesifikasjoner SICK UM18-51112 (hentet fra www.sick.com)

Måleområde	30mm ... 350mm
Oppløsning	0,36 mm
Repeterbarhet	Typ. ± 0.15 % of final value
Presisjon	$\leq 2\%$ of final value (Tilsvare 2mm på 100mm)

Billige ultrasoniske sensorer er også tilgjengelige. Et eksempel er URM37 fra YeRobot [30]. Disse har en relativt begrenset oppløsning (rundt 1 cm), men et stort måleområde fra 4cm til 3m. Den store ulempen med disse sensorene er at måleområdet har et stort tverrsnittsareal som vist i Figur 31. Dette gjør at den er lite egnet til å benyttes mot et lite måleobjekt, og dermed mindre egnet til bruk i dokkingen.

Tong et al. [31] presenterer en dokkingmetode for en enkel mobil robot der de benytter en blanding av triangulering og refleksjonstid (TOF) for å lage en nøyaktig ultrasonisk sensor.



Figur 31 – Illustrasjon av måleområde for ultrasonisk ovenfra (hentet fra [30])

Fordeler

- Stort måleområde
- Uavhengig av overflate

Ulemper

- Måleområdet har stort tverrsnittsareal

6.1.8 Laser-sensor

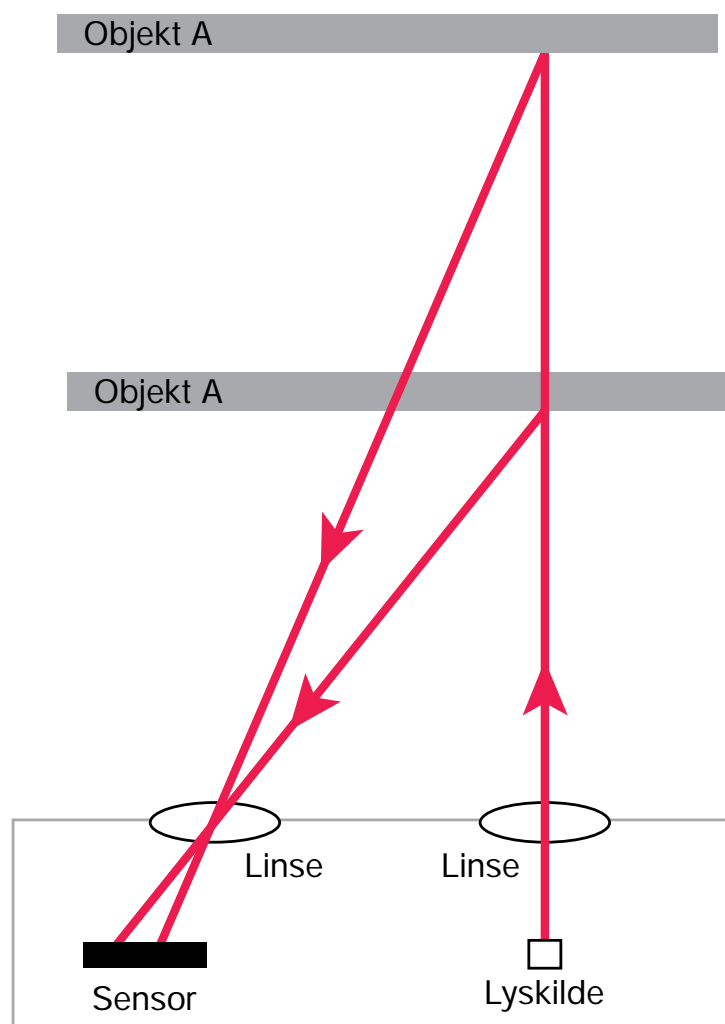
Det finnes også sensorer som benytter samme prinsipp som områdesensoren, men som kun måler i en retning i motsetning til områdesensoren som måler i et bredt område rundt seg. Disse gir en meget god oppløsning og en meget god nøyaktighet over stort et måleområde. Nøyaktigheten disse sensorene gir vil imidlertid være i overkant av hva som er behovet for denne applikasjonen. Disse sensorene har også en meget høy pris.

Tabell 6 – Tekniske spesifikasjoner SICK OD2-N120W60A2 (hentet fra www.sick.com)

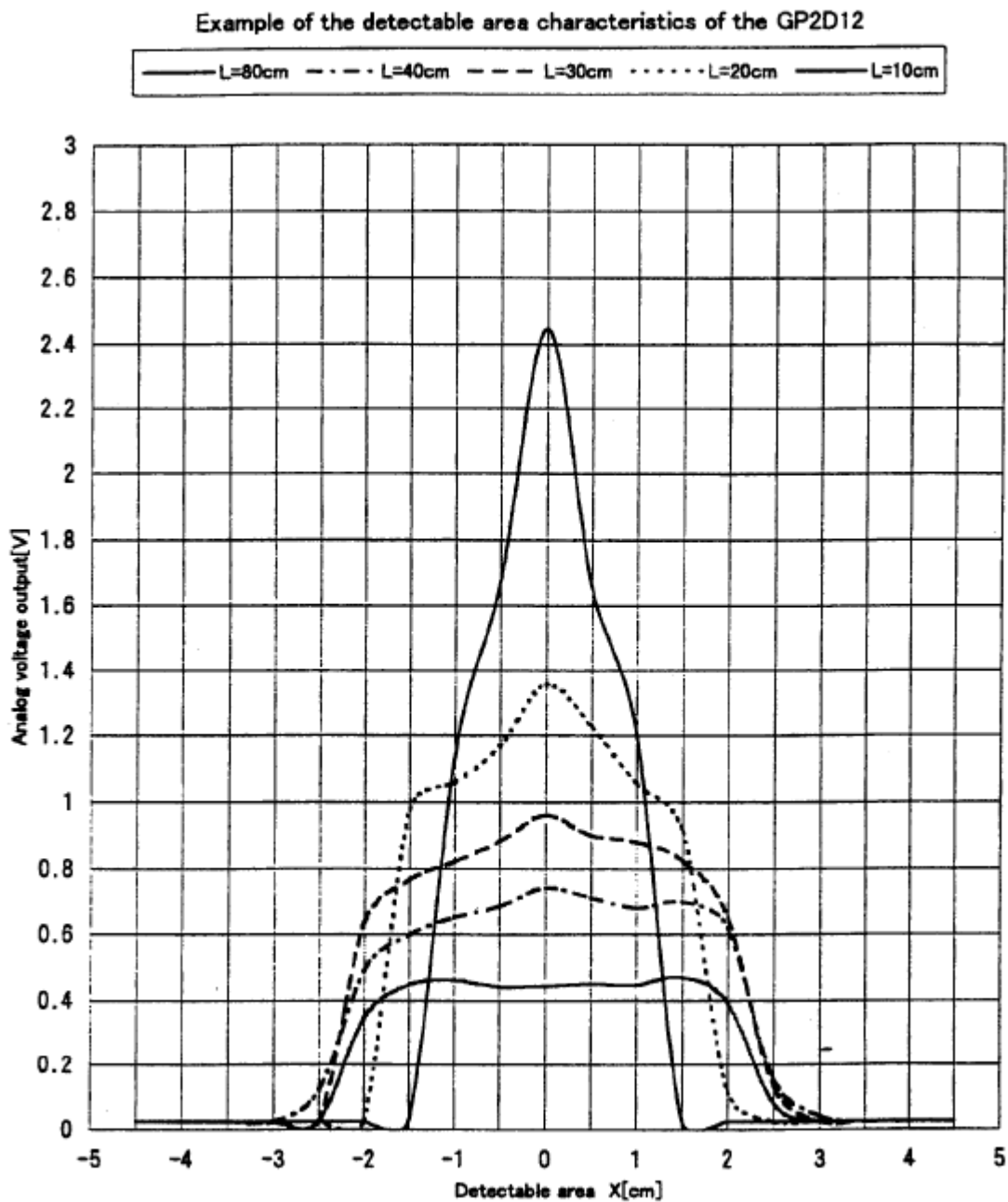
Måleområde 6 ... 90 %	60 mm ... 180 mm
Oppløsning 6 ... 90 %	30 µm (At set averaging medium (10 / 13 ms))
Repeterbarhet 6 ... 90 %	90 µm (At set averaging medium and constant conditions)
Presisjon 6 ... 90 %	± 300 µm (Equivalent to 0.25 % MS, MS= Measuring span: OD2- _120...= 120 mm) (Linearity for 90 % remission is equivalent to 0.1 % MS)

6.1.9 Infrarøde avstandssensorer

Typen infrarøde sensorer som omhandles her er en type som benytter triangulering for å måle avstand til et objekt. Som vist i Figur 32 har den en lyskilde som sender ut en puls av infrarødt lys og en sensor som tar imot lyset. Denne sensoren er en CCC-brikke lik den vi finner i digitale kamera (se Avsnitt 6.1.2). Ut fra målingen denne sensoren gjør er det mulig å bestemme vinkelen lyset kommer inn på sensoren med. Dette er illustrert i figuren der vi ser hvordan målingene fra to objekter, Objekt A og Objekt B, med ulik distanse fra sensoren vil treffe sensoren på ulike punkter og på den måten gi forskjelling verdi. Ut fra den målte vinkelen benyttes så enkel triangulering for å beregne avstanden til objektet. Måten disse sensorene virker på gjør, som vist i Figur 33, at tverrsnittsarealet på måleområdet er lite, og de er derfor egnet for retningsbestemte målinger mot små flater eller små måleobjekter. Tradisjonelle infrarøde sensorer måler intensiteten av det reflekterte lyset, noe som gjør de meget avhengige av den lysreflekterende evnen (reflektivitet) til objektet de måler [9, s. 58-59]. Når man benytter triangulering slipper man dette problemet, og man blir mye mindre avhengig av overflaten til objektet.



Figur 32 - Illustrasjon av infrarød sensor



Figur 33 – Illustrasjon av måleområde for infrarød sensor (hentet fra [1])

Disse sensorene er meget billige i forhold til ultrasoniske og laser-sensorer. De har et mindre måleområde enn de fleste ultrasoniske sensorer, men en høyere båndbredde og de er mindre sensitive for forstyrrelser utenfra.

Tabell 7 – Tekniske spesifikasjoner, SHARP infrarøde distansesensorer

Sensor	Måleområde	Endring i utspenning	Teoretisk oppløsning	Responstid	Strømforbruk
GP2D120	40..300mm	2,25V	0,087 V/cm	39 ms	33 mA
GP2Y0A02YK	200..1500mm	2,05V	0,016 V/cm	39 ms	33 mA

En løsning på dokkingproblemet ved hjelp av infrarøde sensorer ble utviklet av [32]. De oppnådde en nøyaktighet på ± 5 mm i x- og y-retning og $\pm 1^\circ$ i θ . Bruken av infrarøde avstandssensorer har også blitt undersøkt av [33] og [34]. Disse tre applikasjonene bruker imidlertid alle tradisjonelle infrarøde sensorer som måler lys-intensitet.

Prismessig er disse sensorene meget gunstige. De to sensorene som er omtalt over koster 77,70 kr for GP2D120 (eksklusiv merverdiavgift) og 99,60 kr for GP2Y0A02YK (eksklusiv merverdiavgift) [35].

Fordeler

- Meget lav pris
- Ulike måleområder tilgjengelig
- Måleområdet har lite tverrsnittsareal

Ulemper

- Lav oppløsning lengst unna sensoren

6.2 Sammenlikning av tekniske spesifikasjoner for ulike sensorer

I Vedlegg 12 vises en oppsummering av de tekniske spesifikasjonene til de ulike sensorene beskrevet ovenfor. I Tabell 8 er de mest sentrale egenskapene i forhold til denne applikasjonen oppsummert og vurdert.

Tabell 8 – Sammenlikning av de egenskapene til ulike distansesensorer

	Nøyaktighet	Måleområde	Pris	Pris
Laser-sensor	Meget høy	Middels	Meget høy	I/T
Induktiv sensor	Høy	Meget lite	Middels/Høy	500-1200kr
Ultrasonisk sensor	Middels	Middels	Høy	2000-3000kr
Infrarød sensor	Lav	Middels	Meget lav	50-200kr

Lasersensoren har en meget høy nøyaktighet, men prisen er alt for høy til at den skal være et reelt alternativ.

De induktive sensorene har en høy nøyaktighet, en pris som er overkommelig, men de har et måleområde som er veldig lite.

Den ultrasoniske sensoren har et tilfredsstillende måleområde og en god nok nøyaktighet. Den har imidlertid en noe høy pris.

Den infrarøde sensoren har en meget lav pris og et tilfredsstillende måleområde, men nøyaktigheten er i laveste laget.

6.3 Videre lesning om sensorer og målemetoder for mobile roboter

For en mer inngående beskrivelse av virkemåte, samt en bredere oversikt over ulike sensorer som benyttes i mobile roboter anbefales Everett [8] med "Sensors for mobile robots: theory and application". Fisher and Konoliges "Range Sensors" [16] tar også for seg en del aspekter rundt distansesensorer både innenfor teori og bruksområder.

Siegwart and Nourbakhsh [7] med "Introduction to: Autonomous Mobile Robots" er også en utmerket innføring i emnet mobile roboter. Den tar for seg grunnleggende kinematikk og styring, sensorer, lokalisering og navigasjon.

Everett [36] sin "Survey of collision avoidance and ranging sensors for mobile robots" gir også en god oversikt over tradisjonelle metoder for avstandsmåling.

6.4 Konseptforslag

Her presenteres de mest interessante konseptene basert på sensorene og metodene presentert ovenfor. Disse konseptene tar utgangspunkt i at AGV-en befinner seg i området foran dokkingposisjonen før dokkingen begynner. Fra denne posisjonen må den bestemme sin egen positur i forhold til dokkingposisjonen for å kunne beregne banen den må bevege seg for å nå denne posituren. Deretter må den bevege seg inn til dokkingposisjonen uten å krasje med noe før den stopper i korrekt posisjon med tilstrekkelig nøyaktighet.

Konseptene som er beskrevet i dette avsnittet plukket ut basert på flere vurderinger. De er alle konsepter som regnes som gjennomførbare. De er også alle løsninger som fører til en høy grad av fleksibilitet, noe som har vært en svært viktig faktor i utvelgelsen. Kompleksiteten og prisen på de ulike konseptene har også vært en del av vurderingen i forhold til hvilke som blir presentert her.

Innovasjonsgraden til de forskjellige konseptene har også vært et vurderingskriterium; løsninger som allerede er godt utviklet og i bruk i dag er ikke like interessante å undersøke i denne sammenhengen. Av de metodene som er beskrevet i dette kapitlet finnes det flere metoder enn de som presenteres under som kan benyttes eller som allerede benyttes i en viss utstrekning. Linjefølgning er et eksempel på en av metodene som er beskrevet over, men som ikke vurderes her. Dette er på grunn av at linjefølgning allerede er en godt utviklet metode for nøyaktig posisjonering og derfor ikke like interessant å undersøke.

I tillegg til hovedkonseptene presentert her vil det kanskje bli nødvendig å benytte supplerende sensorer for eksempel induktive sensorer for å øke nøyaktigheten i endelig positur dersom den opprinnelige løsningen ikke har tilfredsstillende nøyaktighet.

6.4.1 Bildegjenkjenning vha. kamera montert i rommet

Denne løsningen benytter det samme systemet som allerede prøves ut for å styre AGV-en til dokkingposisjonen (eco-docking). Systemet utvides med et eller flere kameraer plassert på, eller i nærheten av, dokkingstasjonen. På denne måten oppnås en høyere nøyaktighet i akkurat dette området. En løsning på dette problemet er skissert av Santos-Victor and Sandini [22].

Denne løsningen har den fordel at den ikke blir stort mer komplisert etter at dokkingen er implementert i systemet. En stor ulempe er imidlertid at den krever etter eller flere kamera per dokkingstasjon, noe som vil øke prisen på løsningen og samtidig redusere fleksibiliteten.

6.4.2 Bildegjenkjenning vha. kamera montert på AGV

Denne løsningen går ut på at et kamera plassert på AGV-en benyttes for å gjennomføre dokkingen ved å kjenne igjen spesielle kjennetegn ved dokkingstasjonen, og navigere etter disse (såkalt ego-docking).

Den store fordelen ved denne løsningen er at den ikke krever aktivt utstyr på selve dokkingstasjonen. Dette vil være en fordel da det drastisk øker fleksibiliteten til dokkingssystemet. Det er mulig at dokkingstasjonen må utformes på en slik måte at lett kjennes igjen av AGV-en, men dette vil medføre lavere kostnader enn det vil gjøre å utsyre dokkingstasjonen med aktivt utstyr som er dyrt i innkjøp og som må vedlikeholdes jevnlig. I tillegg er det ingen krav til at forskjellige dokkingstasjoner er identiske så lenge AGV-en er i stand til å kjenne den igjen, noe som er meget gunstig fra et fleksibilitetssynspunkt. Denne metoden for dokking har blitt testet tidligere blant annet av Low et al. [23] og Santos-Victor and Sandini [22].

6.4.3 Benytte områdesensor

Denne metoden baserer seg på å benytte områdesensoren AGV-en er utstyrt med (som beskrevet i Avsnitt 6.1.1) for å lese omgivelsene rundt AGV-en når den befinner seg foran dokkingposisjonen. Områdesensoren vil gi en punktsky med data om omgivelsene foran AGV-en. Ut fra denne punktskyen vil det være mulig å beregne AGV-ens positur i forhold til dokkingstasjonen. Ut fra dette beregnes hvilken bane roboten må bevege seg for å komme inn i dokkingstasjonen. Denne løsningen forutsetter at geometrien i høyden områdesensoren arbeider i er kjent og at den er lett å "se" for sensoren. Ulempen er at områdesensoren har en relativt begrenset nøyaktighet i tillegg til at det vil være en utfordring på programmeringssiden.

6.4.4 Benytte flere infrarøde avstandssensorer

Flere infrarøde avstandssensorer, som beskrevet i Avsnitt 6.1.9, med ulikt måleområde benyttes for å måle avstanden i forskjellige vinkler foran og på sidene av roboten, og på denne måten bestemme AGV-ens positur i forhold til dokkingposisjonen.

Den store fordelen med denne løsningen er at man kan benytte mange sensorer på grunn av sensorenes meget lave pris. Ulike sensorer med ulikt måleområde kan benyttes samtidig. Ulempen er at sensorene har relativt lav oppløsning og dårlig repeterbarhet og det er usikkert om de er gode nok til å kunne gjennomføre dokkingen.

6.4.5 Oppsummering av de foreslåtte konseptene

Alle disse forslagene som er presentert her må regnes om meget interessante. Det mest spennende er kanskje å benytte områdesensoren. Den har imidlertid en oppgitt nøyaktighet som er for lav. I tillegg vil utvikling av denne metoden i all hovedsak bli en utfordring på programmeringssiden, noe som ville blitt for ressurskrevende å gjennomføre i dette prosjektet.

Å benytte et kamera montert på AGV-en er også en meget interessant løsning, men i likhet med å benytte områdesensoren innebærer det en stor programmeringsjobb som overgår den kompetansen og de ressursene som er tilgjengelig i prosjektet. Det er i tillegg svært usikkert hvilken nøyaktighet det er mulig å oppnå med denne løsningen.

Å benytte kamera montert i området rundt AGV-en er en mer realistisk metode å gjennomføre dokkingen på. Mye av de tunge programmeringsjobben er allerede gjort siden dette allerede er en del av prosjektet. Problemet med denne løsningen er at det også her er lite trolig at det er mulig å oppnå en tilfredsstillende nøyaktighet.

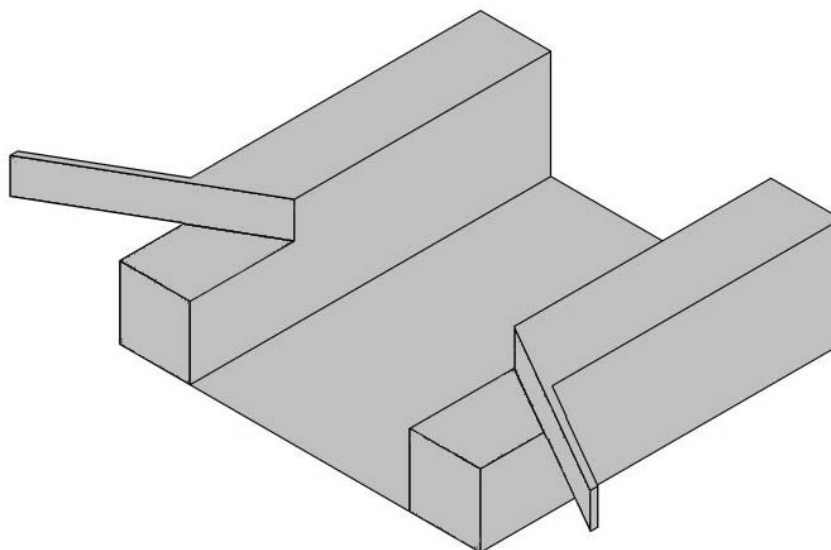
Løsningen der infrarøde distansesensorer benyttes for å bestemme distansen til objekter rundt AGV-en er interessant fordi den på grunn av sensorenes meget lave pris vil være meget billig. Den er også relativt enkel programmeringsmessig sett. Utfordringene med denne løsningen er at det sannsynligvis er behov for et spesialkonstruert anlegg rundt selve dokken. Det vil være en utfordring å implementere sensorene slik at de yter best mulig i forhold til signalkvalitet, presisjon og hastighet. Ikke mindre er denne løsningen såpass interessant, særlig ut fra et kostnadssynspunkt, at det er den som blir valgt for å undersøke ytterligere.

7 Dokking ved hjelp av infrarøde sensorer

På grunn av sensorenes meget lave pris er det som nevnt meget interessant å gjøre forsøk for å undersøke hvorvidt de er gode nok til å gjennomføre dokkingoperasjonen med slike sensorer. Seks sensorer er innkjøpt; to med mellomlangt måleområde og fire med et kort måleområde. Disse sensorene er identiske med de som er beskrevet i Tabell 7. Datablad for de to sensorene finnes i Vedlegg 6 og Vedlegg 7. Et notat omkring anvendelse av disse sensorene er også vedlagt som Vedlegg 13. I Kapittel 7.1 presenteres flere ulike konseptuelle konfigurasjoner for plassering av sensorene på AGV-en for å gjennomføre dokkingen. Det største usikkerhetsmomentet angående disse sensorene er som nevnt om de har høy nok nøyaktighet for å kunne benyttes til dokkingoperasjonen. For å vurdere hvorvidt sensorene kan levere tilfredsstillende oppløsning blir sensorene grundig testet, noe som er beskrevet i Kapittel 7.3.

For å kunne benytte de infrarøde sensorene er man avhengige av at selve dokken er utformet slik at sensorene relativt enkelt kan kjenne igjen enkelte flater eller punkter. En illustrasjon av en slik dokk er vist i Figur 34. Denne dokken har en skråstilt flate på begge sider av åpningen. Disse skråstilte flatene er laget for at sensorene med langt måleområde skal ha en flate å sikte imot.

Denne flaten strekker seg ikke helt ned til gulvet; under den skråstilte flaten stikker den plane flaten ut. Denne flaten vil sensorene med kort rekkevidde måle imot. For å få til dette monteres sensorene med langt måleområde høyt oppe på AGV-en, i samme høyde som de skråstilte flatene, mens sensorene med kort måleområde monteres lavt på AGV-en slik at de sikter mot de plane flatene.



Figur 34 – Illustrasjon av dokk

For å bevege seg inn i dokken er det som nevnt nødvendig å bestemme AGV-ens positur relativt til dokken. Når vi vet posituren relativt til dokken kan vi benytte dokken som et midlertidig globalt koordinatsystem. Vi kan nå overføre AGV-ens positur til dokk-koordinater ved hjelp av metoden beskrevet i Kapittel 0. Dette gjør at vi kan bevege oss mot den ønskede posituren inne

i dokken som også er nullpunktet i det globale koordinatsystemet. Dette gjør også at dokkens posisjon i det globale koordinatsystemet ikke behøver å måles på forhånd. AGV-en vil under dokkingoperasjonen da kun basere bevegelsene sine på dokkens plassering. En ønsket startposisjon for dokkingen må imidlertid bestemmes i det globale koordinatsystemet. Det er denne posisjonen AGV-en kjøres til før selve dokkingprosedyren begynner.

For å sikre nøyaktig posisjonering når man har kommet fram til punktet man ønsker å nå kan det være aktuelt å benytte ytterligere en sensor. Dette kan for eksempel være en induktiv sensor som måler mot et spesielt punkt på veggen inne i dokken eller en optisk sensor som for eksempel føler en stripe på gulvet inne i dokken.

Når man har kommet fram til nøyaktig posisjon kan det også være interessant å låse AGV-en fast mekanisk. Dette ønsker man å gjøre for å sikre seg nøyaktig posisjonering, samt at man sikrer seg at AGV-en ikke beveger seg mens man utfører operasjonen den er dokket for.

7.1 Konseptuelle sensorkonfigurasjoner

Her presenteres noen konseptuelle måter for hvordan de infrarøde sensorene kan monteres på roboten for å kunne gjennomføre dokkingoperasjonen. Seks sensorer er tilgjengelige; to sensorer med et måleområde fra 20 til 150 cm og fire sensorer med et måleområde fra 4 til 30 cm. De to sensorene med langt måleområde må åpenbart plasseres slik at de måler framover. De vil bli benyttet for at AGV-en skal kunne bestemme sin positur relativt til dokken før bevegelsen inn i dokken begynner, samt å følge bevegelsen mens AGV-en beveger seg mot dokken. De fire sensorene med kort måleområde vil brukes til å bestemme AGV-ens positur når den har kommet inn i selve dokken.

Figur 35 illustrerer måleområdet til sensorene. Det grå området er det "blinde" området der sensoren ikke kan måle, mens det røde området er sensorene måleområde.



Figur 35 – Illustrasjon av måleområdet til infrarød sensor

7.1.1 Konsept 1

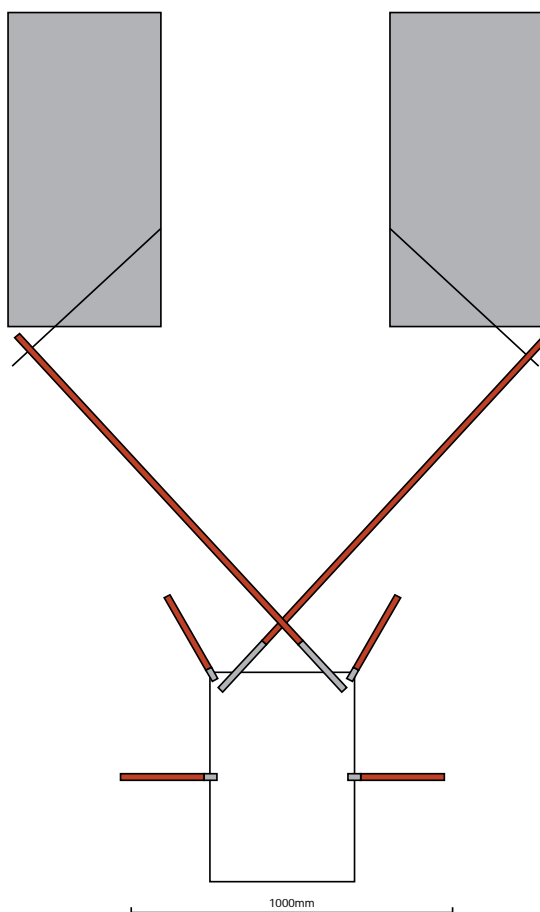
Konsept 1 er illustrert i Figur 36. Her er de to sensorene med store måleområdet plassert foran på AGV-en med en vinkel på 60 grader. På begge sider av innkjøringen til dokkingposisjonen er det montert skrå flater som sensorene siktes mot. Dette gjøres for at signalet skal bli mer pålitelig, siden de infrarøde sensorene gir en mer pålitelig avlesning når de siktes mot en flate

som er vinkelrett i forhold til sensoren. AGV-en vil befinne seg på senterlinjen når de to sensorene gir den samme avstandsavlesningen. Gjennom å utføre en algoritme med ulike bevegelse vil også være mulig å beregne hvor langt vekk fra den senterlinjen AGV-en befinner seg.

To av de fire sensorene med mindre måleområde er plassert foran på AGV-en pekende framover. Disse tar over når AGV-en er nær nok dokken. Disse sensorene har en bedre nøyaktighet og vil på den måten gi en høyere nøyaktighet i utregnet positur. I og med at disse sensorene ikke står rett på flaten de måler mot vil de ha høyere usikkerhet i målingene enn hvis de hadde stått vinkelrett i forhold til flatene inne i dokken.

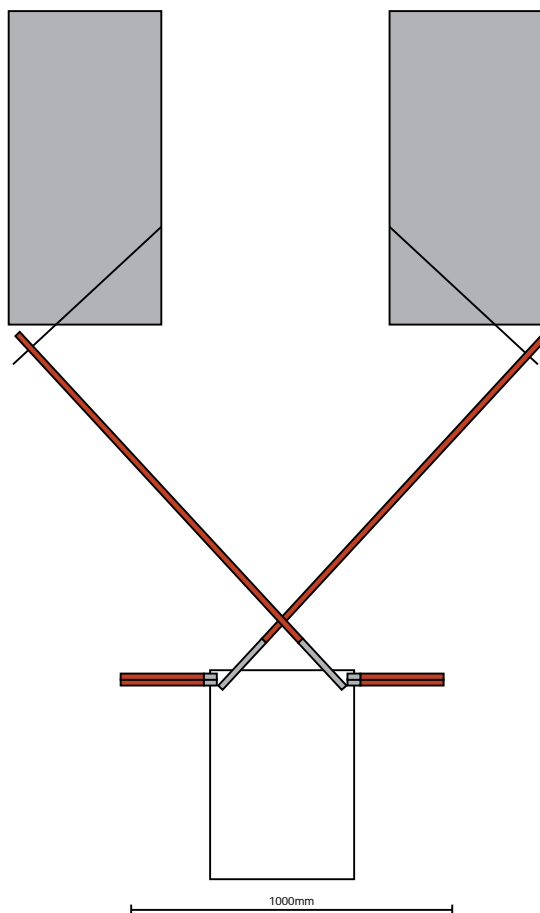
Det siste paret sensorer sitter noe lenger bak og vil være rettet direkte ut mot sidene. Disse sensorene vil i motsetning til de som er plassert lengst framme være plassert vinkelrett i forhold til flaten de måler og på den måten ha en høyere nøyaktighet i målingene.

Overflaten til anlegget AGV-en kjører inn i bør være dekket av et materiale som har en høy refleksivitet for infrarøde stråler. Selv om sensorene har god toleranse for materialer med ulik refleksivitet vil det være en fordel med høy refleksivitet, særlig når sensoren er plassert med en viss vinkel i forhold til objektet.



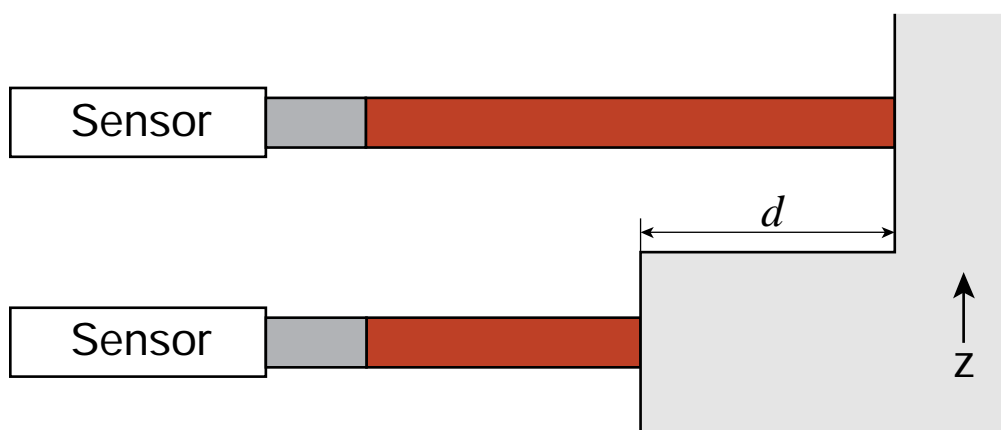
Figur 36 – Illustrasjon av sensorplassering i konsept 1

7.1.2 Konsept 2



Figur 37 – Illustrasjon av sensorplassering i konsept 2

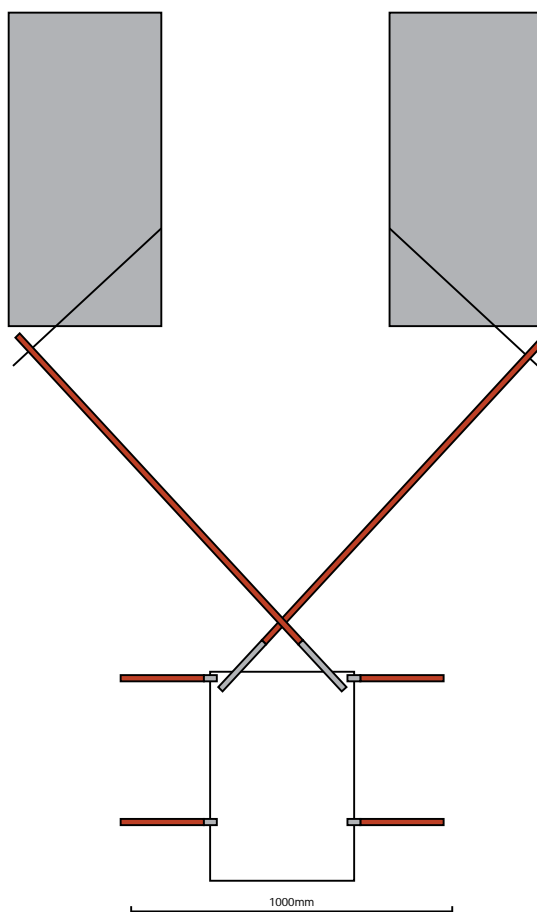
Dette andre konseptet har, som vist i Figur 37, den samme plasseringen av de to sensorene med lang rekkevidde som det første konseptet. Plasseringen av de andre sensorene er imidlertid forskjellig. Her er to sensorer plassert direkte over hverandre på begge sider av AGV-en. På selve dokkingstasjonen er det to nivåer i samme høyde som de to sensorene. De to nivåene har en horisontal avstand, d , mellom seg som vist i Figur 38. Denne avstanden starter som relativt høy ved innkjøringen til dokken, men jo lenger inn i dokken man kommer jo mindre blir avstanden d . På denne måten kan man beregne hvor langt inn i dokken man har kjørt i tillegg til at man vet avstanden til veggene på begge sider av AGV-en. Dette kan benyttes til å bestemme hastigheten på AGV-en (det vil være naturlig å redusere hastigheten når man nærmer seg den ønskede posisjonen).



Figur 38 – Konsept 2, detalj fra bakfra

7.1.3 Konsept 3

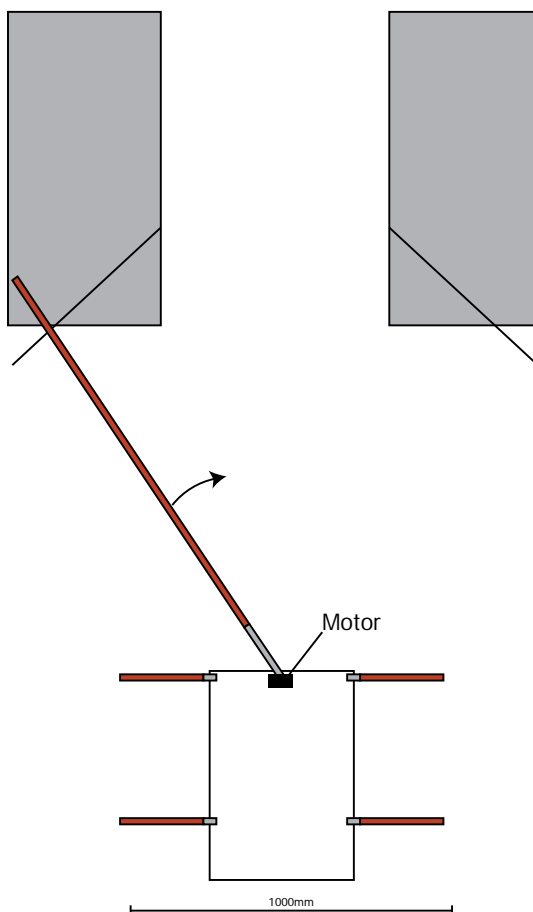
Konsept 3, som vist i Figur 39, har i likhet med de to første konseptene de to sensorene med lang rekkevidde plassert foran i en krysset konfigurasjon. De to parene med sensorer med kort rekkevidde er plassert i samme høyde, henholdsvis lengst foran og et stykke bak på AGV-en. Dette gjør at AGV-en både kan bestemme posisjon mellom de to veggene i dokken samt sin egen vinkel i forhold til veggene ved å sammenlikne målingene fra fremre og bakre sensor.



Figur 39 – Illustrasjon av sensorplassering i konsept 3

7.1.4 Konsept 4

Konsept 4, illustrert i Figur 40, har i motsetning til de andre konseptene ikke to kryssede sensorer i front, men heller én sensor montert med en motor slik at den kan beveges fra side til side. Denne konfigurasjonen gir de infrarøde sensorene tilnærmet samme virkning som områdesensoren. Løsningen blir uavhengig av å utføre bevegelse med AGV-en for å bestemme posituren relativt til dokken. Løsningen er imidlertid avhengig av at motoren som styrer sensoren har en høy nøyaktighet.



Figur 40 – Illustrasjon av sensorplassering i konsept 4

Blant disse fire konseptene er det Konsept 2 eller 3 som er de mest gjennomførbare. Å montere sensoren på en motor som skissert i Konsept 4 vil innføre en stor mulig feilkilde med selve systemet som roterer sensoren, noe som ikke er ønskelig. Konsept 2 vil gi en angivelse av hvor langt inn i dokken AGV-en befinner seg, men konseptet vil ikke kunne bestemme AGV-ens vinkel slik som Konsept 3 kan. På grunnlag av dette er det Konsept 3 som er det beste konseptet.

Ulempen med alle disse prosedyrene er at de krever en viss nøyaktighet før start av prosedyren. Dette siden sensorene har et begrenset måleområde. Konseptene 1,2 og 3 vil også kreve at AGV-en beveger seg for å bestemme sin positur i forhold til dokken.

Dersom en industrirobot skal benyttes for å plukke deler fra AGV-en kan man se for seg en ekstra anvendelse for sensorene dersom man benytter Konsept 3. De fire sensorene plassert på

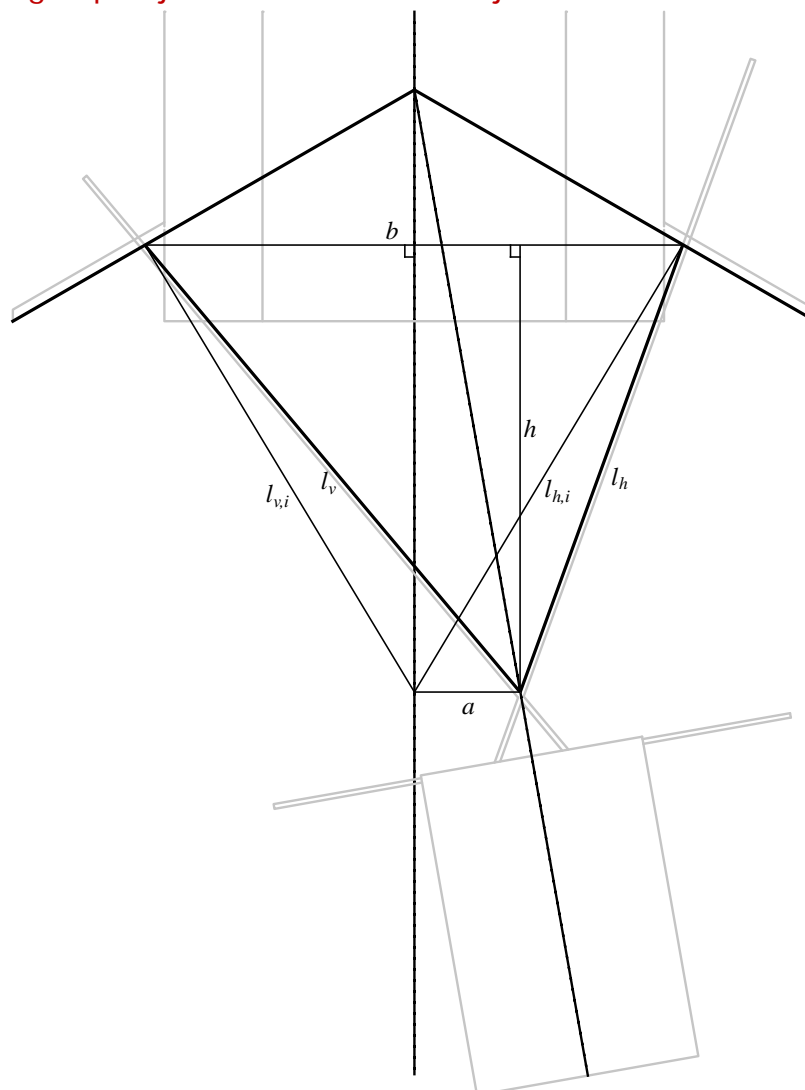
de to sidene av AGV-en kan med en relativt høy nøyaktighet bestemme AGV-ens positur inne i dokken. Denne posituren kan så sendes til industrirobotens styresystem som kan benytte denne informasjonen til å justere sitt koordinatsystem for AGV-en slik at den kan plukke rett fra AGV-en.

7.2 Skissert dokkingprosedyre

Her presenteres en prosedyre med utgangspunkt i Konsept 3 og dokken illustrert i Figur 34 over. Denne prosedyren består av en rekke trinn:

1. Finn avstand fra senterlinje
 - a. Roter til høyre til venstre sensor gir maksimal avlesning
 - i. Lagre vinkel
 - ii. Roter til utgangspunktet
 - b. Roter til venstre til høyre sensor gir maksimal avlesning
 - i. Lagre vinkel
 - c. Beregn vinkelen mellom disse to målingene
 - d. Roter til denne vinkelen. AGV-en vil nå peke mot skjæringspunktet mellom de to skråflatene slik som illustrert i Figur 41.
 - e. Gjør måling av begge sensorene
 - i. Lagre resultatene
 - ii. Bestem ut fra disse målingene om man befinner seg til høyre eller venstre for senterlinjen
 - f. Beregn avstanden fra senterlinjen (som beskrevet under, i Kapittel 7.2.1)
2. Gjennomfør dokking
 - a. Beregn bevegelse (som beskrevet under, i Kapittel 7.2.2)
 - b. Følg banen til AGV-en befinner seg på senterlinja.
 - c. Følg senterlinja mens de fremoverrettede sensorene kontinuerlig avleses
 - i. Juster kursen etter sensorene, begge sensorene skal gi samme avstand.
 - d. Når sensorene med kort måleområde kommer innenfor dokken benytt disse for justering av kurs.

7.2.1 Beregning av posisjon i forhold til senterlinje



Figur 41 – Hjelpefigur for vinkelberegning

Ideelt sett ønsker man å befinne seg på senterlinjen som går gjennom selve dokken. Vi vet at vi befinner oss en distanse, a , vekk fra denne senterlinja (her beregnet ut fra punktet der de infrarøde strålene krysses). Ut fra figuren over kan vi regne oss fram til a , b og h (mellomregningen finnes i Vedlegg 14):

$$b = \sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)} \quad (7.1)$$

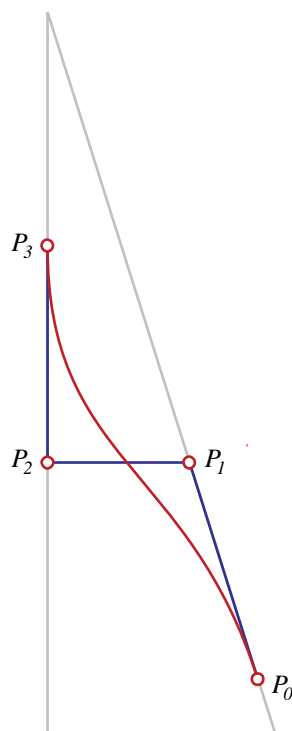
$$h = \frac{l_h l_v \sin(60^\circ)}{\sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)}} \quad (7.2)$$

$$a = \sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)} - \left(\sqrt{l_h^2 - h^2} + \sqrt{l_v^2 - h^2} \right) \quad (7.3)$$

Disse beregningene kan enten gjøres på mikrokontrolleren eller på datamaskinen. Med mindre man benytter en meget kraftig mikrokontroller, er imidlertid en fordel å gjøre mest mulig av arbeidet på datamaskinen. Datamaskinen har en mye større prosesseringskraft og muligheten for å kjøre flere prosesser samtidig.

7.2.2 Beregning av bane mellom utgangspunkt og punkt på senterlinje

For å få en fin bevegelse fra startpunktet som ligger ved siden av senterlinja og inn på senterlinja ønsker man å bevege seg etter en kurve som starter tangent med AGV-ens utgangspositur, og som ender tangent med senterlinja i punktet der AGV-en møter denne linja. Dette kan oppnås ved å bruke en såkalt kubisk Bézier-kurve der de to styrepunktene P_1 og P_2 ligger på de linjene man ønsker at kurven skal være tangent med. Dette illustreres i Figur 42 der punktene P_0 og P_3 er henholdsvis start- og slutt punktet for bevegelsen, mens punktene P_1 og P_2 er såkalte styrepunkter. Punktene P_1 og P_2 er her plassert slik at linja P_1P_2 krysser linja P_0P_3 på dens halveringspunkt.



Figur 42 – Illustrasjon av en kubisk Bézier-kurve

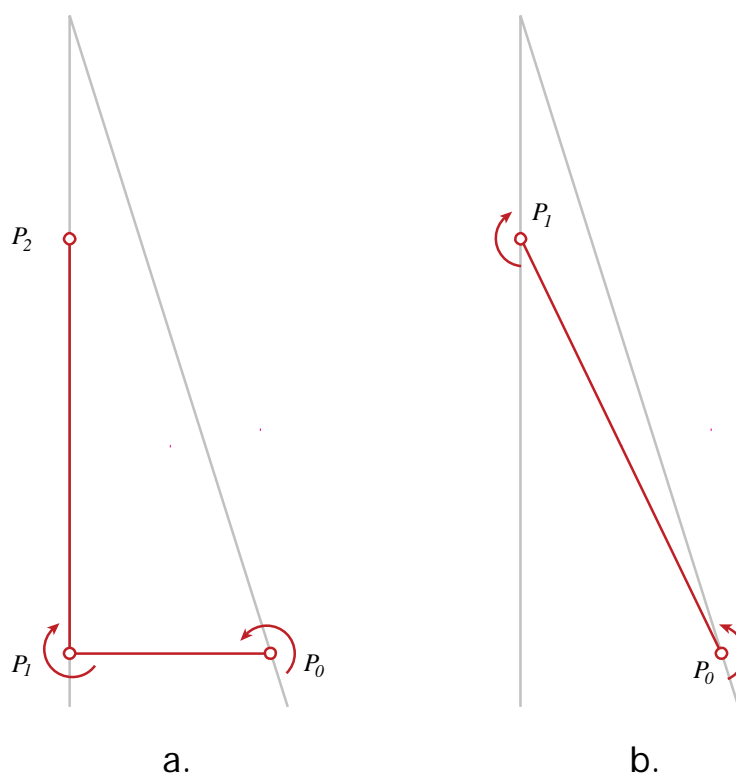
Likningen for kurven mellom P_0 og P_3 , som er en Bézier-kurve med fire punkter beskrives ved [37]:

$$\mathbf{C}(t) = (1-t)^3 \mathbf{P}_0 + 3(1-t)^2 t \mathbf{P}_1 + 3(1-t) t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3, \quad t \in [0,1] \quad (7.4)$$

Der $t = 0$ ved P_0 og $t = 1$ ved P_3 .

Det som er viktig ved bruk av denne likningen i dokkingtilfellet er at punktet P_3 velges slik at AGV-en fortsatt har noe klaring mellom seg og dokken når den befinner seg i punktet. Dette vil redusere sannsynligheten for at AGV-en kolliderer med dokken dersom det er mye feil i odometrien.

Dette er en god løsning på hvordan AGV-en kan bevege seg for å plassere seg på senterlinja. Denne fremgangsmåten krever imidlertid at man har meget nøyaktig kontroll over AGV-ens bevegelse, noe som ikke nødvendigvis er tilfelle. I tilfelle at AGV-en ikke kan styres så nøyaktig som det kreves her kan man benytte seg av mindre kompliserte metoder for å bevege seg til målet. To slike metoder er illustrert i Figur 43:



Figur 43 – Illustrasjon av bevegelsesmønstre

Bevegelsesmetode a. i figuren går ut på å først rotere slik at man peker direkte mot senterlinja, deretter bevege seg til senterlinja for så å rotere slik at man står parallelt med denne. Etter dette kan den bevege seg framover langs senterlinja og inn i dokken. Bevegelsesmetode b. går ut på å rotere slik at man er rettet direkte mot punktet P_1 som befinner seg på senterlinja, for så å bevege seg dit før man roterer slik at man står parallelt med senterlinja.

Begge disse to metodene benytter kun bevegelse langs rette linjer, noe som bør være mer gjennomførbart for de fleste mobile roboter sammenliknet med bevegelse langs Bézier-kurven som beskrevet over.

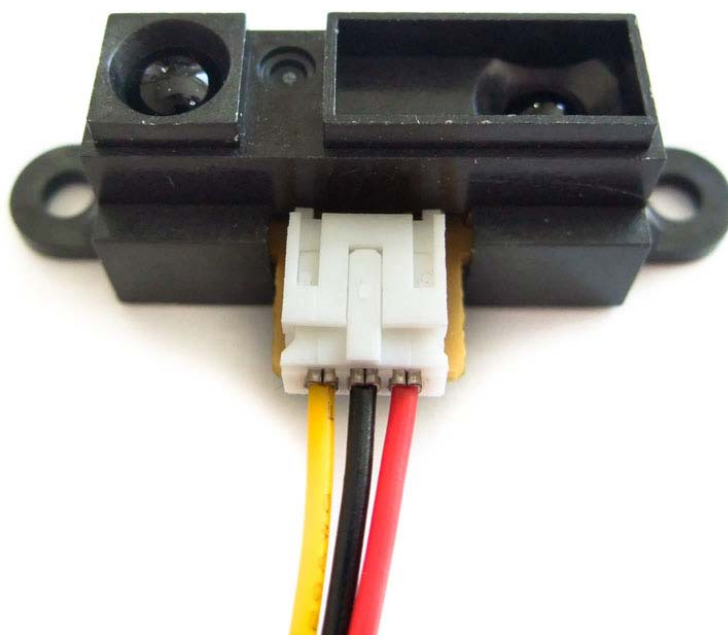
7.3 Testing av sensorer, SHARP GP2D120

For å teste nøyaktigheten til sensorene ble det utformet et testoppsett der sensorene ble koblet til en mikroprosessor som leser av det analoge signalet og sender det via seriell kommunikasjon til en datamaskin. Som databladet til sensoren (Vedlegg 6) anbefaler er en $1\mu\text{F}$ kondensator mellom matespenning og jord. Mikrokontrolleren har en analog-til-digital-omsetter med en oppløsning på 10 bit (fra 0 til 1023). Denne avleste verdien sendes så til datamaskinen via en emulert seriell port. Et enkelt skript skrevet i Python tar imot verdiene og lagrer de til en tekstfil som kan importeres til for eksempel Microsoft Office Excel for videre analyse. Skriptet er skrevet for versjon 2.6 av Python, og er gjengitt i Vedlegg 8 samt på vedlagte CD-ROM.

Mikrokontrolleren som ble benyttet er en Arduino Decimilla [38]. Den er basert på ATmega168 fra Atmel som bygger på AVR-arkitekturen. En FTDI FT232RL-brikke sørger for muligheten til seriell kommunikasjon over USB. Denne mikrokontrolleren er laget for meget enkel prototyping gjennom å ha et meget enkelt og raskt programmeringsgrensesnitt med et høy-nivå språk kalt wiring. Siden dette er et høy-nivå språk inkluderes det en del grunnleggende funksjoner når programmet kompiles til C. Det resulterende programmet vil på denne måten være langt fra optimalt når det gjelder effektivitet, men for å teste sensorene, som uansett ikke er meget raske, er det godt nok. Programkoden for mikrokontrolleren som ble benyttet for å teste sensorene finnes i Vedlegg 9. I denne koden blir signalet til analog-til-digital-omsetteren preskalert med 64 (i stedet for 128 som er standard), noe som gir en høyere samplingsfrekvens enn standard.

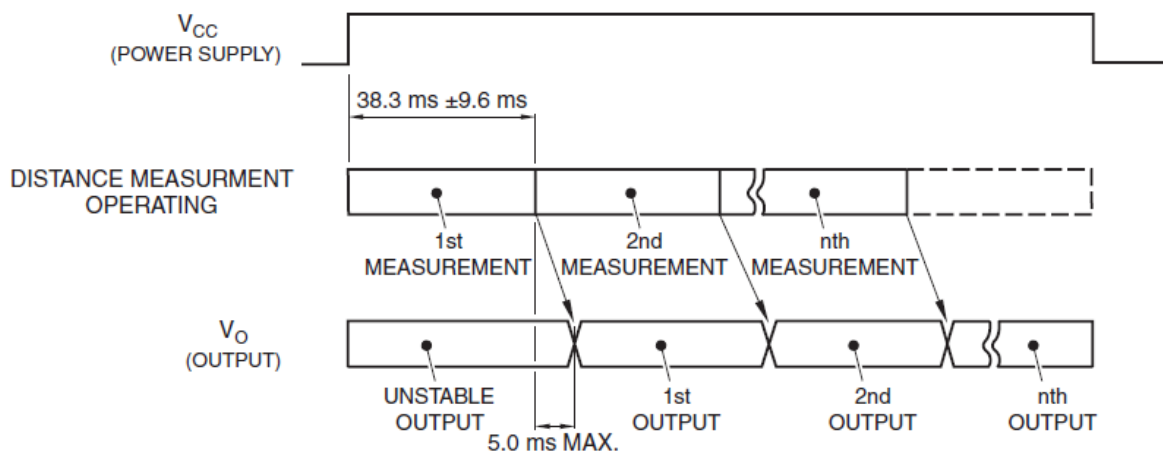
Sensoren, som vist i Figur 44 har tre tilkoblingsledninger:

- Rød: Matespenning
- Svart: Jord
- Gul: Analogt utsignal



Figur 44 – Infrarød avstandssensor, SHARP GP2D120

Sensoren har en oppgitt typisk responstid på 39ms, dette er også vist i Figur 45. Dette gir en frekvens på 25,64Hz, som igjen gir en Nyquist-frekvens på 51,28Hz. Dette er frekvensen sensorene bør samples med.

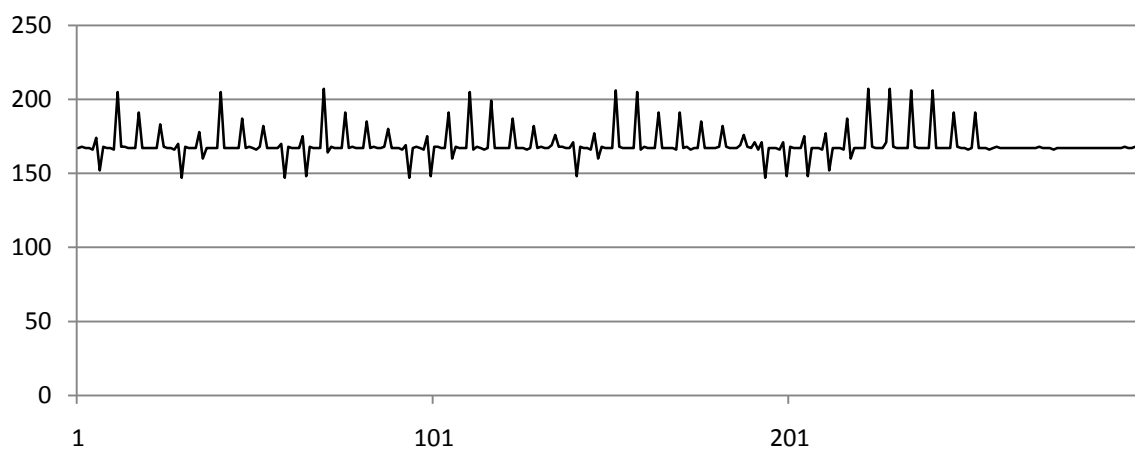


Figur 45 – Timing diagram for infrarød sensor (hentet fra Vedlegg 6)

7.3.1 Signalkvalitet

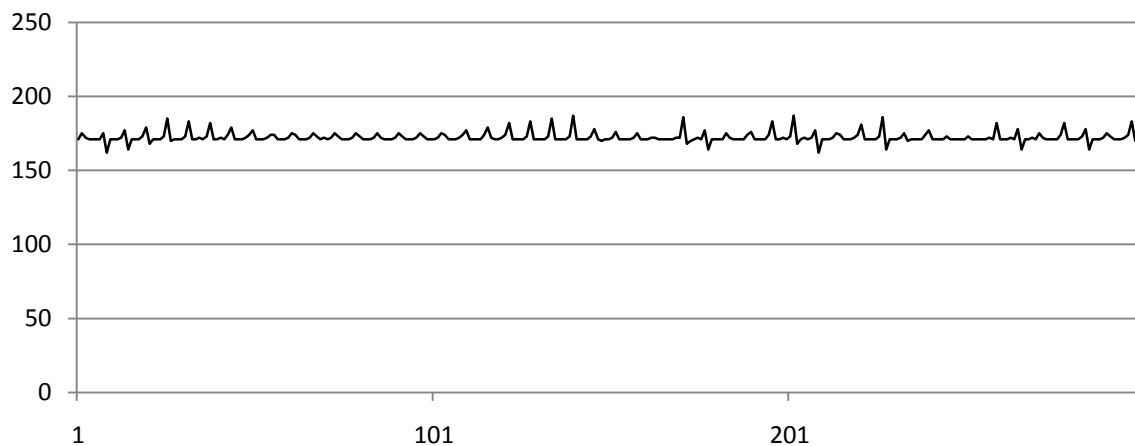
Den første testen gikk ut på å teste hvor stabilt signalet var når sensoren stod i et fast oppsett med en fast distanse fra avstandssensoren til en flate dekket med vanlig hvitt papir. Avstanden fra sensoren til måleoverflaten var cirka 150mm, noe som er omtrent midt i måleområdet til sensoren.

Den første testen viste et signal som relativt ustabil, med kraftige hopp som vist i Figur 46. På en data serie med 1000 punkter var variansen på 85,81.



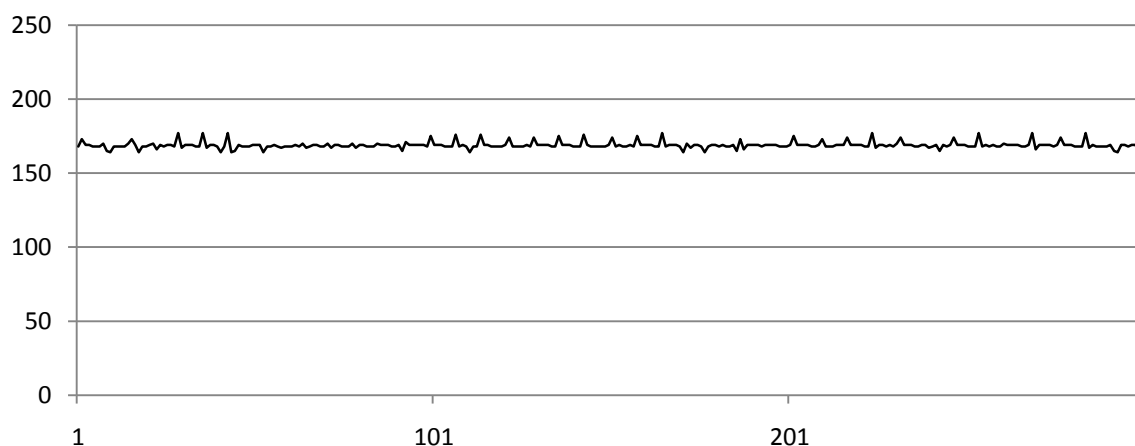
Figur 46 – Testing av sensor, forsøk 1

På grunn av virkemåten til sensoren trekker den ganske mye strøm når det sender ut lyspulser. Dette kan være noe av årsaken til de høye toppene som ble observert i det første forsøket. For å redusere dette ble kondensatoren på 10 μF byttet ut med en på 100 μF . Dette ga, som Figur 47 viser, et bedre resultat og toppene var helt klart mindre enn i det første forsøket. Variansen over 1000 datapunkter ble nå redusert til 12,88.



Figur 47 – Testing av sensor, forsøk 2

I testoppsettet i de to første forsøkene ble strømmen til mikrokontrolleren tilført fra datamaskinen gjennom samme den USB-kabelen som kommunikasjonen gikk på. Mikrokontrolleren har også muligheten for strømtilførsel gjennom en ekstern strømkilde. En transformator som leverte 9V og 0,5 A ble koblet til og benyttet som strømkilde. Dette ga et enda mindre varierende signal som Figur 48 viser. Variansen ble nå redusert til 5,18.



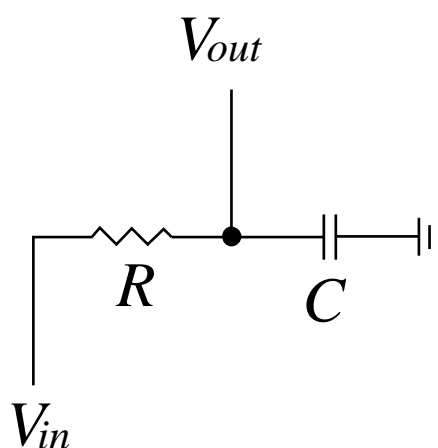
Figur 48 – Testing av sensor, forsøk 3

For å redusere det ustabile resultatet ble det implementert et fysisk lavpassfilter. Et lavpassfilter har i prinsippet samme virkningen som å ta et løpende gjennomsnitt [39]. Et løpende

gjennomsnitt vil si å kontinuerlig ta gjennomsnittet av de siste x målingene. På denne måten vil et slikt filter glatte ut signalet ved å redusere vikningen av hurtige endringer i signalet. Et fysisk lavpass filter lages ved å benytte en motstand og en kondensator på måten vist i Figur 49. Et slikt lavpassfilter gir en knekkfrekvens gitt ved:

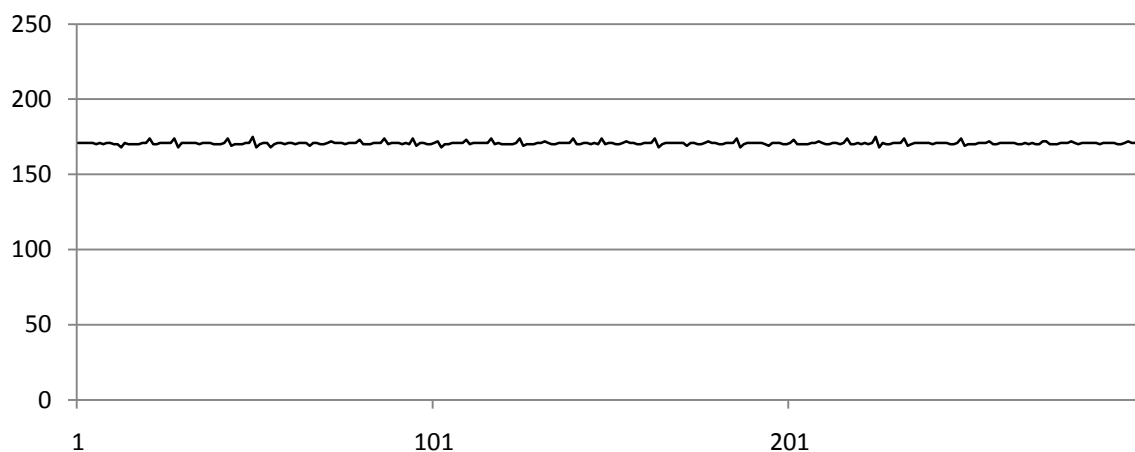
$$f_c = \frac{1}{2\pi RC} \quad (7.5)$$

Det implementerte lavpassfilteret hadde en motstand på $100 \text{ } \Omega$ og en kondensator på $47 \text{ } \mu\text{F}$, noe som gir en knekkfrekvens på $33,86 \text{ Hz}$. Dette lavpassfilteret ble plassert på det analoge utsignalet.



Figur 49 – Kablingsskjema av fysisk lavpassfilter

Et nytt forsøk ble gjort og som Figur 50 viser var når signalet mye mer stabilt. Variansen over 100 datapunkter var nå nede i 1,21.



Figur 50 – Testing av sensor, forsøk 4

Dette lavpassfilteret gir imidlertid en lav responstid. Komponentene i lavpassfilteret ble derfor byttet til en motstand på 150 Ω og en kondensator på 10 μF , noe som ifølge likning (7.5) vil gi en knekkfrekvens på 106,10Hz. Dette oppsettet gav også en varians på 1,21, men samtidig en responstid som var mye bedre.

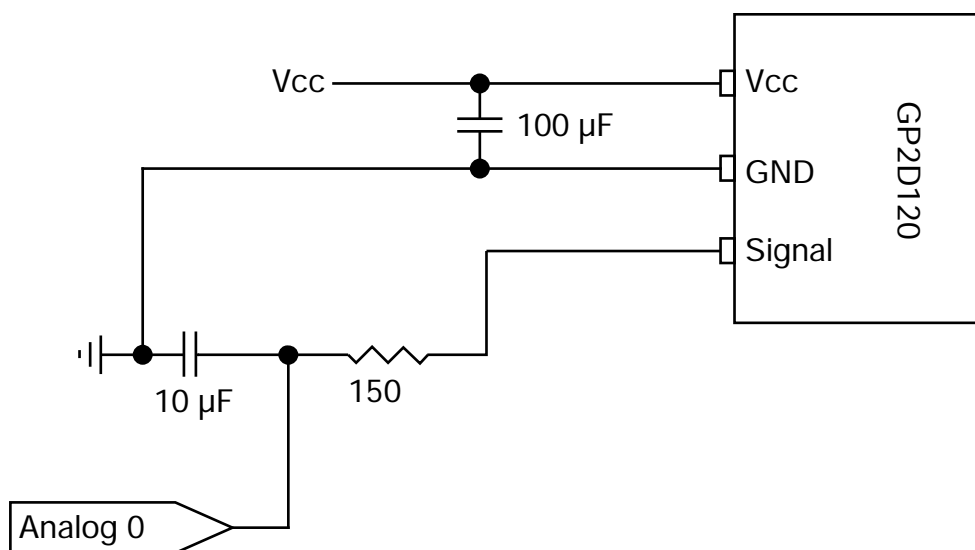
Oppsummering

Tabell 9 viser en oversikt over de fire forsøkene og hva som var forskjellen, og hvilket resultat som ble oppnådd.

Tabell 9 – Oversikt over forsøk med infrarød distansesensor

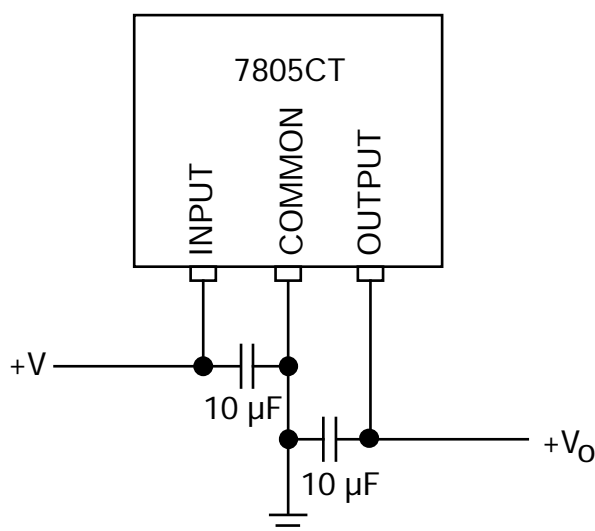
Forsøk	Beskrivelse	Varians
Forsøk 1		85,81
Forsøk 2	Økt til 100 μF kondensator	12,88
Forsøk 3	Endret til ekstern strømforsking 9V	5,18
Forsøk 4	Implementert lavpassfilter	1,21

Det endelige koblingsskjemaet er vist i Figur 51, der Analog 0 refererer til porten på mikrokontrolleren.



Figur 51 – Koblingsskjema sensor

Strømforsyningen til sensorene reguleres med en spenningsregulator som leverer 5 V. Spenningsregulatoren er av typen 7805CT, og kobles som vist i Figur 52, med to kondensatorer, og til felles jord med mikrokontrolleren. Spenningsregulatoren er i følge databladet godkjent for inn-spennning på opp til 35 V.

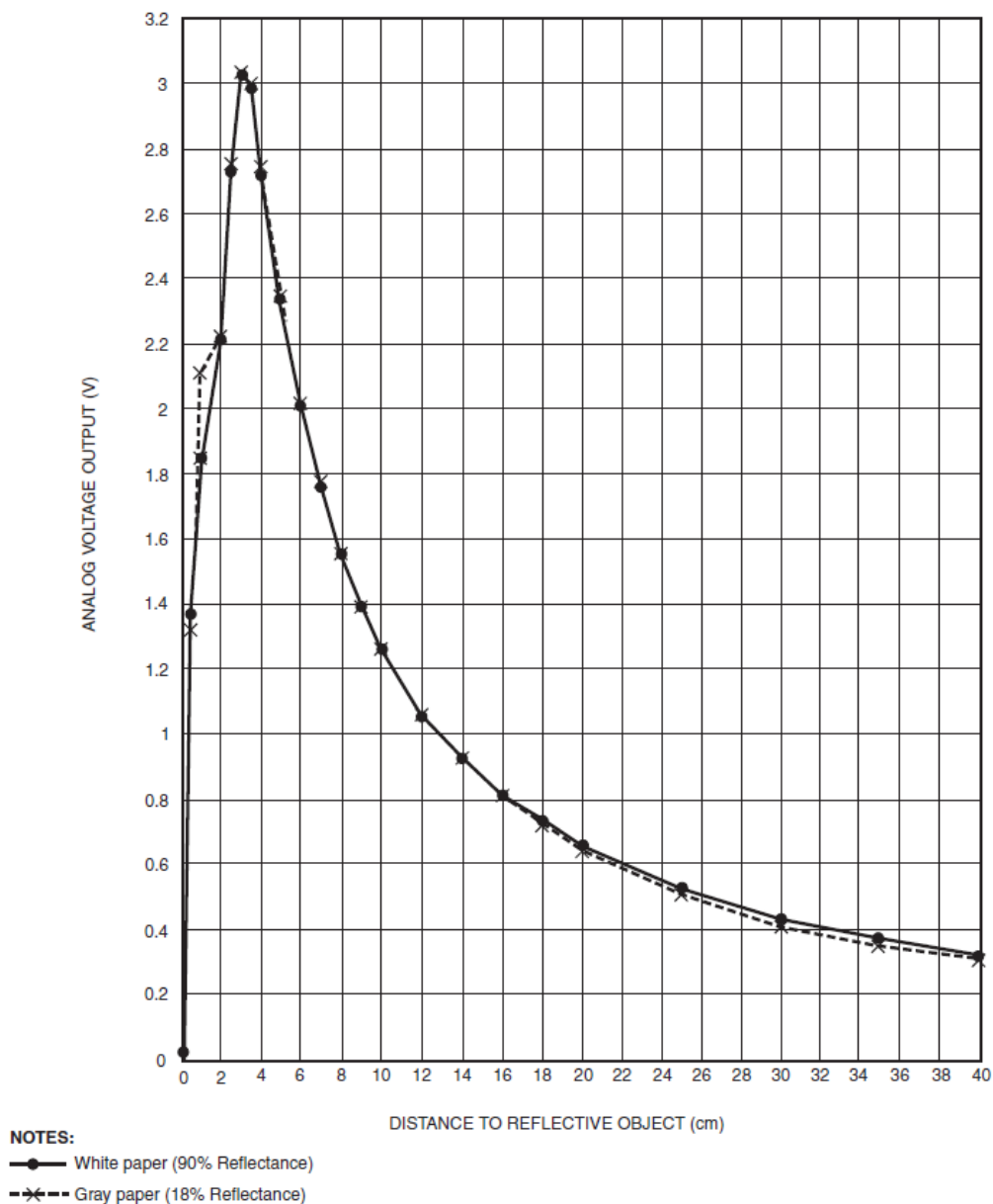


Figur 52 – Koblingskjema spenningsregulator

7.3.2 Oppløsning

Kurven i Figur 53 viser hvordan utspenningen varierer i forhold til avstanden fra sensoren til måleobjektet. Vi ser at den synker bratt fra rundt 4 cm som er den korteste måledistansen, mens den flater ut når den nærmer seg 30 cm som er den lengste måledistansen. Siden mikrokontrollere har en absolutt måleskala vil dette si at oppløsningen er høyere jo nærmere måleobjektet er sensoren (dette vises også i Tabell 12). Den maksimale utspenningen fra sensoren er ifølge figuren rett over 3V. På bakgrunn av dette vil det være naturlig å benytte en ekstern referanse til mikrokontrollerens analog-til-digital-omsetter. Omsetteren virker ved at den sammenlikner det analoge signalet (fra sensoren) med en referansespenning. Dersom man ikke benytter en ekstern referansespenning vil spenningen på 5V bli benyttet som referanse. I dette tilfellet benyttes en ekstern referanse på 3,3V som spenningen fra sensoren blir sammenliknet med. Basert på likning (7.6) gjør dette at minste måleverdi blir redusert fra 4,8828mV ($5V/2^{10}$) til 3,2227mV ($3,3V/2^{10}$).

$$\text{minste måleverdi} = \frac{\text{referansespenning}}{\text{oppløsning}} \quad (7.6)$$



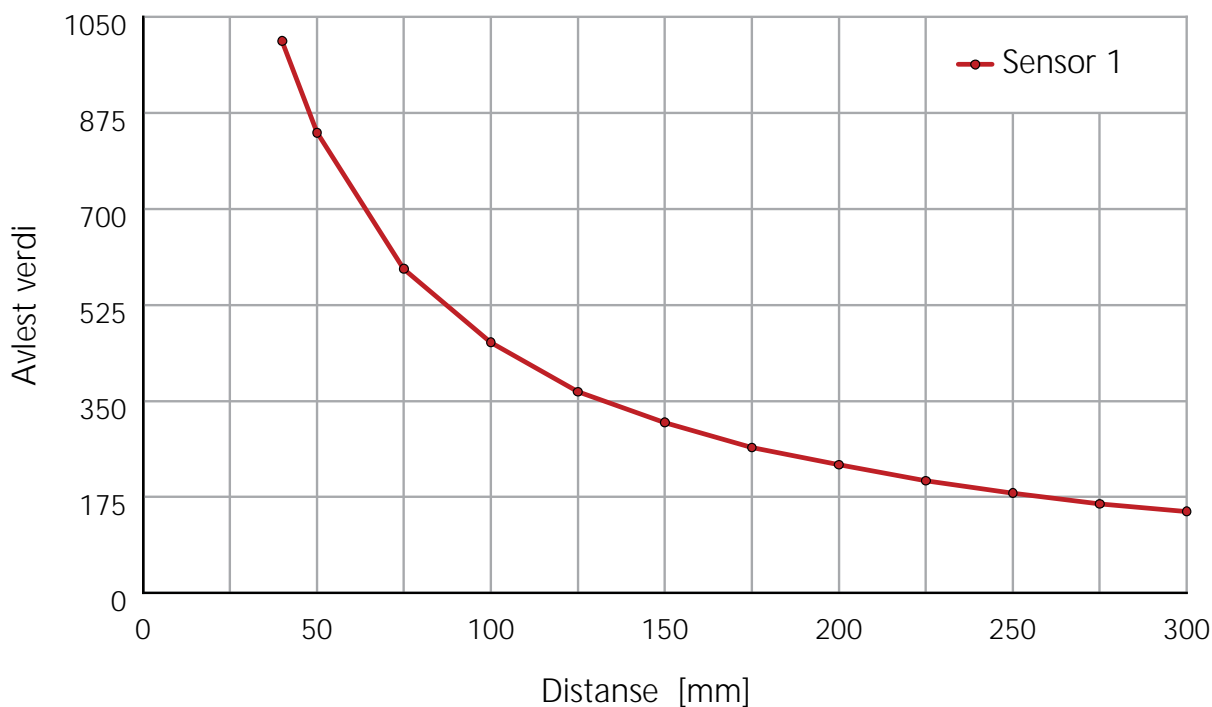
Figur 53 – Karakteristikk av utspenning for infrarød sensor SHARP GP2D120, hentet fra Vedlegg 6

For å beregne den faktiske avstanden ut fra avlesningene fra sensorene er det beste alternativet å benytte en tabell med testede verdier for kjente distanser og interpolere ut fra disse verdiene. Et metode for å gjøre dette med Python er vedlagt i Vedlegg 11 samt på vedlagte CD-ROM. Et objekt lages med `DistanceTable(points)` der `points` er listen med de korresponderende parene med verdiene og distanseverdiene. Distanse for en gitt avlest verdi kan deretter hentes ved å benytte `table[avlest verdi]`:

Eksempel 2 – bruk av klassen DistanceTable

```
import DistanceTable

sensor1 = ((40, 572.3), (50, 480.0), (75, 329.3), (100, 252.0))
tableSensor1 = DistanceTable(sensor1)
distance = tableSensor1[300]
```



Figur 54 – Gjennomsnittsmålinger ved ulike distanser, sensor 1

Tabell 10 og Tabell 11 viser sammendraget av testene som ble gjort på to av de fire innkjøpte sensorene. Den gjennomsnittlige avleste verdien er også plottet i Figur 55. Som vi kan se i tabellene og i figuren gir sensorene litt ulike verdier for samme distanse, men dette er sannsynligvis på grunn av små unøyaktigheter i måleoppsettet. Dette taler for å kalibrere hver sensor individuelt, noe som er fullt mulig med det interpolasjonsskriptet som ble presentert over siden det implementerer selve interpolasjonen som et objekt. For å benytte ulik kalibrering for de ulike sensorene konstruerer man bare et objekt forskjellig interpolasjonstabell for hver av dem.

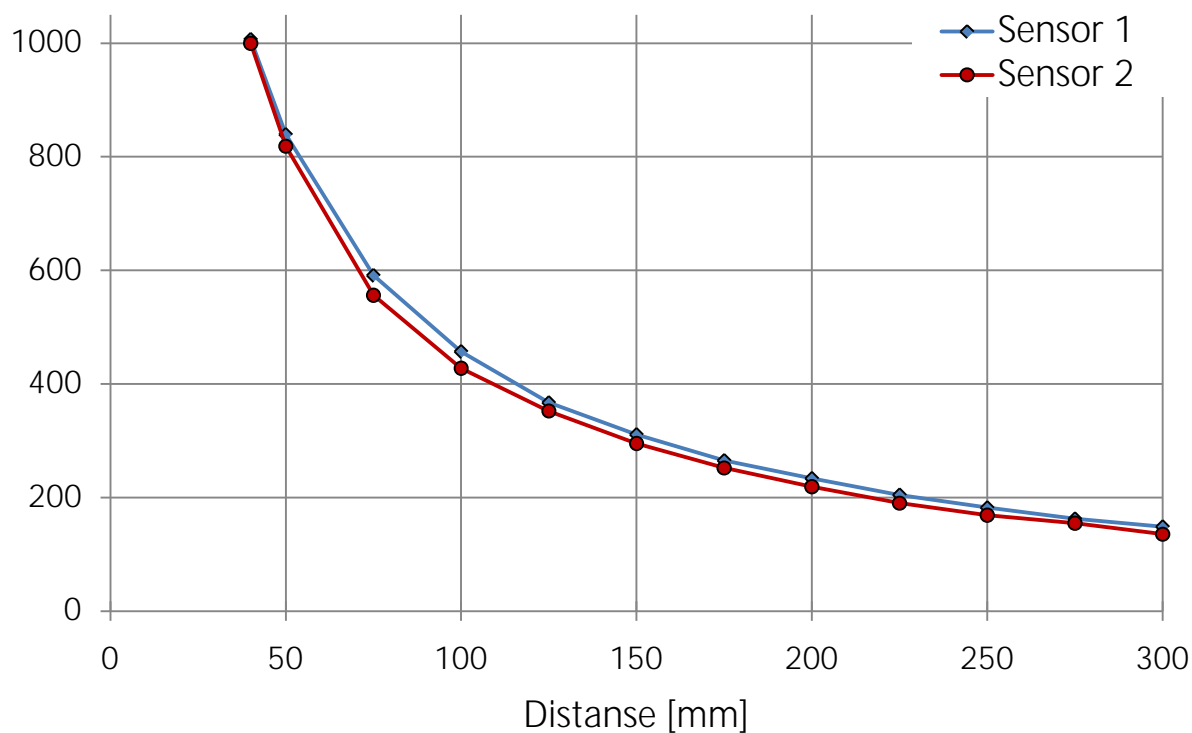
Tabell 10 – Målinger ved ulike distanser (1000 datapunkter) Sensor 1

Distanse [mm]	40	50	75	100	125	150	175	200	225	250	275	300
Gjennomsnittlig avlest verdi	1006,4	839,0	591,0	457,0	367,1	310,9	265,2	233,7	204,7	182,2	162,6	148,7
Varians	3,23	0,28	1,11	2,74	1,91	0,66	10,63	3,79	1,00	7,45	0,77	1,67
Standardavvik	1,80	0,53	1,05	1,65	1,38	0,81	3,26	1,95	1,00	2,73	0,88	1,29
Utregnet spenning [V]	3,25	2,71	1,91	1,47	1,18	1,00	0,86	0,75	0,66	0,59	0,52	0,48

Tabell 11 – Målinger ved ulike distanser (1000 datapunkter) Sensor 2

Distanse [mm]	40	50	75	100	125	150	175	200	225	250	275	300
Gjennomsnittlig avlest verdi	999,6	818,5	556,3	427,8	352,3	295,3	252,1	219,2	190,1	168,9	154,8	135,4
Varians	0,63	0,42	0,23	1,19	0,24	0,22	8,66	0,18	0,21	0,13	0,87	6,62
Standardavvik	0,80	0,65	0,48	1,09	0,48	0,47	2,94	0,43	0,45	0,37	0,93	2,57
Utregnet spenning [V]	3,22	2,64	1,79	1,38	1,14	0,95	0,81	0,71	0,61	0,54	0,50	0,44

Det som er mer interessant å merke seg er for det første at standardavviket ikke synes å variere stort over måleområdet, noe som er svært positivt med tanke på å benytte sensorene i dokkingoperasjonen. Den andre tingen som er verdt å merke seg er at de to sensorene viser en noe ulik karakteristikk når det gjelder hvor mye signalet varierer. Forskjellen i hvor mye signalet varierer er imidlertid ikke så stor at det vil medføre problemer i bruk.



Figur 55 – Sammenlikning av avlest data for Sensor 1 og Sensor 2

I Tabell 12 er oppløsningen for ulike intervaller beregnet. Som beskrevet over vil oppløsningen reduseres jo lenger unna sensoren man måler. I det nærmeste intervallet (mellom 40 og 50mm fra sensoren) er oppløsningen 54,0 mV per millimeter. Oppløsningen i det intervallet som ligger lengst borte fra sensoren (mellom 275 og 300mm) er oppløsningen nede i 1,8 mV per millimeter noe som er en trettiendedel av nærmeste intervallet. Sammenliknes dette med minste måleverdi som ble beregnet til 3,2227mV kan vi enkelt se at den høyeste mulige oppløsningen i det nærmest intervallet er 0,061mm, mens den høyeste mulige oppløsningen, den teoretisk minste målbare endringen, i det intervallet som ligger lengst unna er 1,827mm.

Tabell 12 – Lineær oppløsning ved ulike avstander fra sensor

Avstand [mm]	Utspenning [V]	Oppløsning [mV/mm]	Minste målbare endring [mm]
40	3,247	54,001	0,060
50	2,707	32,003	0,101
75	1,906	17,285	0,186
100	1,474	11,610	0,278
125	1,184	7,246	0,445
150	1,003	5,900	0,546
175	0,855	4,062	0,793
200	0,754	3,746	0,860
225	0,660	2,903	1,110
250	0,588	2,528	1,275
275	0,524	1,795	1,796
300	0,480		

7.3.3 Frekvens og oversampling

Mikrokontrolleren basert på en AVR ATmega168 har en analog-til-digital-omsetter med 10-bits oppløsning. Det er imidlertid mulig å øke denne oppløsningen ved å benytte såkalt oversampling. Oversampling foregår ved at flere verdier slås sammen og bit-skiftes for å gi en verdi med høyere oppløsning [40]. Når man oversampler kan man si at man øker oppløsningen på signalet ved å ofre samplingshastigheten. Oversampling krever en liten mengde støy for å produsere gode resultater, og dersom det ikke er nok støy på signalet er det anbefalt å legge til kunstig støy for å sørge for at man får den lille støyen man har behov for. Dersom det ikke er nok støy til å få oversamplingen til å virke tilfredsstillende kan man eventuelt fjerne lavpassfilteret.

I forsøkene som ble gjort over leverte mikrokontrolleren verdier til datamaskinen med en frekvens på rundt 70 Hz. Dersom man godtar at frekvensen reduseres til rundt 20 Hz, kan man med oversampling oppnå en høyere oppløsning på 11 bit. Dette gjør man ved å summere fire og fire avleste verdier. Den binære representasjonen av denne summen bit-skiftes en posisjon til høyre. Den bit-skiftede verdien vil nå være det avleste resultatet med 11-bits oppløsning. Et

eksempel på hvordan man utfører oversampling vises i Eksempel 3. Ved å benytte en raskere omsetter kan man bruke oversampling men fortsatt opprettholde en høy frekvens.

Eksempel 3 – Oversampling

Vi har de avleste verdiene med 10-bits presisjon hentet fra sensoren:

Desimalverdi	215	216	215	213
Binærverdi (10-bit)	0011010111	0011011000	0011010111	0011010101

Summen av disse verdiene er 859, som tilsvarer en binærverdi på 1101011011. Dersom vi bit-skifter denne verdien en plass til høyre får vi verdien 0110101101, som tilsvarer desimalverdien 429. Dette er nå den nye sensorverdien med 11-bits presisjon

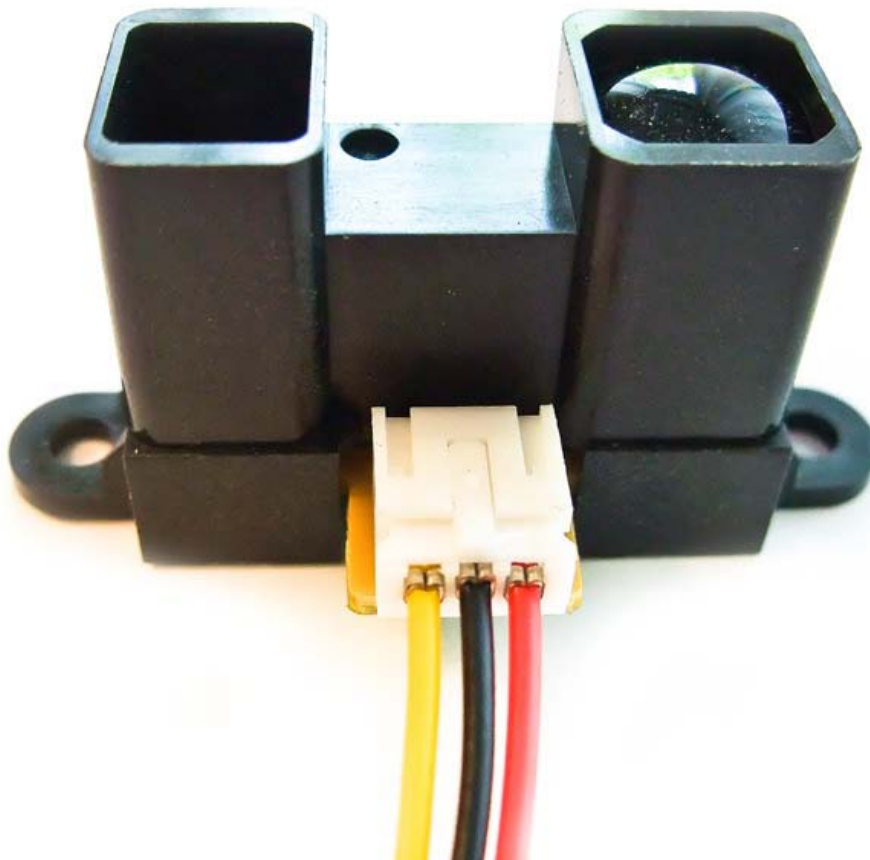
7.3.4 Oppsummering

Etter noen forsøk ble signalkvaliteten på disse sensorene god nok til at man kan benytte de til å styre AGV-en etter. Det som er viktig når man plasserer sensorene er at man benytter en regulert strømkilde som leverer nok kapasitet til å drive sensorene. Å benytte en 100 kondensator nære sensoren var også en faktor som økte kvaliteten på signalet. I tillegg gjør et enkelt lavpassfilter en ytterligere støyreduksjon på signalet som avleses av mikrokontrolleren.

Oppløsningen til sensoren virker også å være tilstrekkelig god selv med en analog-til-digital-omsetter med en oppløsning på 10-bit. Den dårligste oppløsningen på 1,827mm utgjør 0,609 % av distansen på 300mm.

7.4 Testing av sensorer, SHARP GP2Y0A02YK

Sensoren SHARP GP2Y0A02YK av samme type som GP2D120 som ble testet over. Forskjellen er at den har et større måleområde. Utformingen er også noe annerledes. Som Figur 56 viser er den noe større enn den første sensoren.

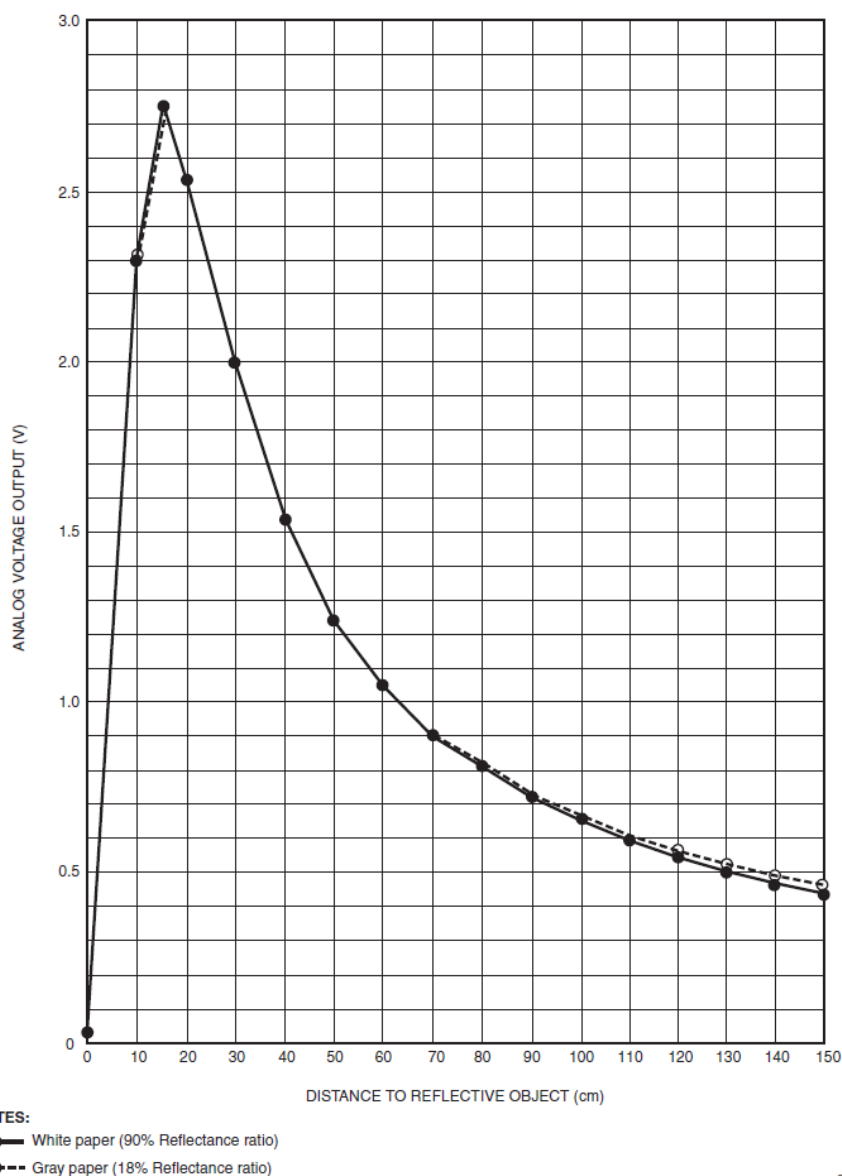


Figur 56 – Infrarød avstandssensor, SHARP GP2Y0A02YK

Databladet til sensoren finnes i Vedlegg 7.

Siden sensorene er like bortsett fra størrelsen på måleområdet ble det samme oppsettet benyttet for å teste disse sensorene som det som ble benyttet for å teste sensorene beskrevet over.

Som vist i Figur 57, har denne sensoren en noe forskjellig karakteristikk på utspenningen enn det den andre sensoren (vist i Figur 53) har. Den har samme prinsipielle form med en utspenning som synker eksponentielt i måleområdet. Nærmere sensoren enn den nærgrensen til sensoren ynker utspenningen bratt. Det mest interessante er at denne sensoren har en mindre avstand mellom høyeste og laveste utspenning enn den andre sensoren. Den høyeste utspenningen er lavere og den laveste utspenningen er høyere. Dersom man benytter den samme mikrokontrolleren til å måle utspenningen fra de to sensorene vil dette nødvendigvis føre til at oppløsningen til denne sensoren blir lavere enn den andre sensoren, siden mikrokontrolleren kun kan opereres med en referansespenning for analog-til-digital-omsetteren.



Figur 57 – Karakteristikk av utspenning for infrarød sensor SHARP GP2Y0A02YK, hentet fra Vedlegg 7

Som Tabell 13 og Tabell 14 viser har denne sensoren en karakteristikk som likner den første sensoren. Variasjonen i signalet er i samme størrelsesorden for begge sensorene, og den varierer uten noen spesielle trender over måleområdet. Vi ser at også disse sensorene har noe ulik karakteristikk fra sensor til sensor, noe som taler for at individuell kalibrering er nødvendig.

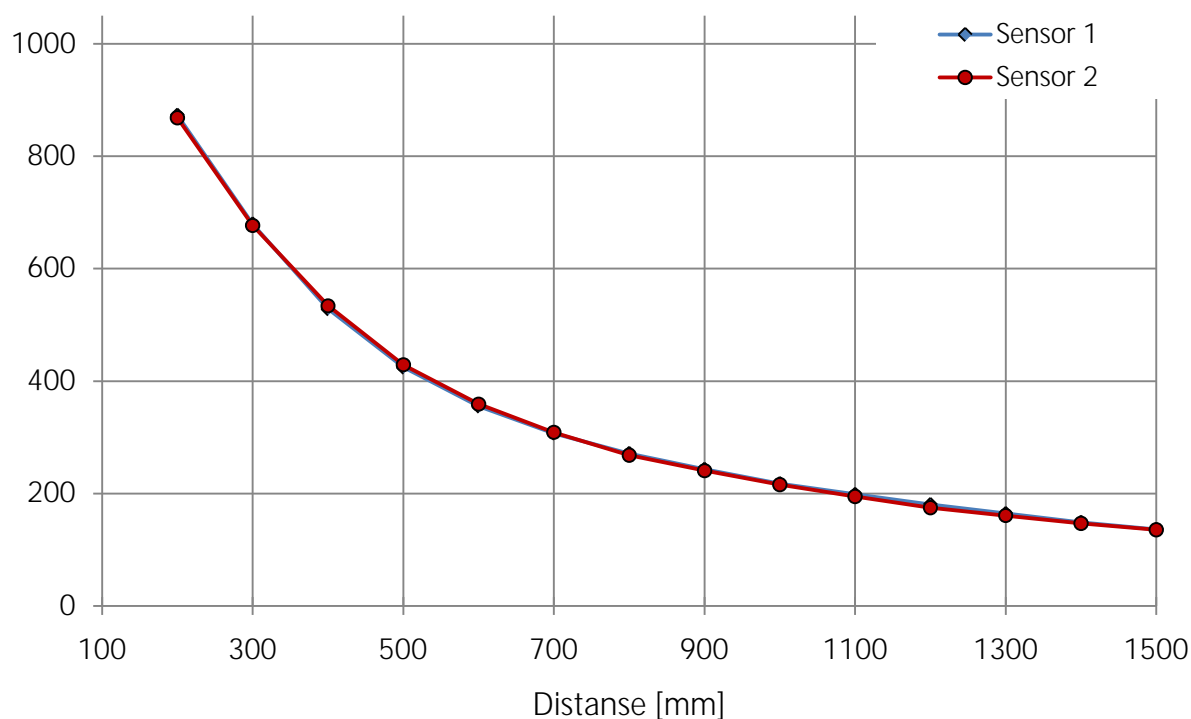
Tabell 13 – Målinger ved ulike distanser (1000 datapunkter) Sensor 1, GP2Y0A02YK

Distanse [mm]	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500
Gjennomsnittlig avlest verdi	872,9	679,4	528,2	424,5	355,2	307,2	271,3	243,0	217,7	198,4	180,4	164,3	148,8	135,9
Varians	0,87	0,28	8,77	0,64	0,16	0,37	0,62	0,14	9,70	9,31	1,23	4,84	9,81	7,78
Standardavvik	0,93	0,53	2,96	0,80	0,39	0,60	0,79	0,38	3,11	3,05	1,11	2,20	3,13	2,79
Utrechnet spenning [V]	2,82	2,19	1,70	1,37	1,15	0,99	0,88	0,78	0,70	0,64	0,58	0,53	0,48	0,44

Tabell 14 – Målinger ved ulike distanser (1000 datapunkter) Sensor 2, GP2Y0A02YK

Distanse [mm]	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400	1500
Gjennomsnittlig avlest verdi	868,5	676,9	533,9	429,0	359,1	308,9	268,1	240,6	215,9	194,7	174,9	160,7	147,0	135,5
Varians	0,30	0,32	0,43	9,39	0,48	1,74	0,90	11,86	1,53	2,83	3,56	3,85	6,25	18,36
Standardavvik	0,55	0,57	0,66	3,06	0,69	1,32	0,95	3,44	1,24	1,68	1,89	1,96	2,50	4,28
Utregnet spenning [V]	2,80	2,18	1,72	1,38	1,16	1,00	0,86	0,78	0,70	0,63	0,56	0,52	0,47	0,44

7.4.1 Oppløsning



Figur 58 – Sammenlikning av avlest data for Sensor 1 og Sensor 2 (GP2Y0A02YK)

Tabell 12 oppløsningen i de ulike distanseintervallene fra sensoren på samme måte som Tabell 12 gjør det for den andre sensoren. I det nærmeste intervallet (mellom 200 og 300mm fra sensoren) er oppløsningen 6,24 mV per millimeter. Oppløsningen i det intervallet som ligger lengst borte fra sensoren (mellom 1400 og 1500mm) er oppløsningen nede i 0,416 mV per millimeter noe som er en femtededel av nærmeste intervallet. Sammenlikner dette med minste måleverdi som ble beregnet til 3,2227mV kan vi enkelt se at den høyeste mulige oppløsningen i det nærmest intervallet er 0,516mm, mens den høyeste mulige oppløsningen, den teoretisk minste målbare endringen, i det intervallet som ligger lengst unna er 7,741mm.

Tabell 15 – Lineær oppløsning ved ulike avstander fra sensor GP2Y0A02YK

Distanse [mm]	Utspenning [V]	Oppløsning [mV/mm]	Høyeste oppnåelige oppløsning [mm]
200	2,816	6,240	0,516
300	2,192	4,876	0,661
400	1,704	3,346	0,963
500	1,369	2,236	1,441
600	1,146	1,550	2,079
700	0,991	1,158	2,784
800	0,875	0,911	3,539
900	0,784	0,818	3,941
1000	0,702	0,621	5,192
1100	0,640	0,583	5,527
1200	0,582	0,517	6,230
1300	0,530	0,501	6,430
1400	0,480	0,416	7,741
1500	0,438		

7.4.2 Oppsummering

Oppløsningen på 7,741mm utgjør 0,516 % av den totale lengden på 1500mm. Sammenlikner vi dette med 0,609 % som var tilsvarende verdi for den andre sensoren kan vi konkludere med at de to sensorene relativt sett har en ganske lik ytelse. Det er også viktig å merke seg at sensoren med det minste måleområdet har en referansespenning som er bedre tilpasset sensorens utspenning.

Dersom man benytter en mikrokontroller med en analog-til-digital-omsetter med høyere presisjon (for eksempel 12, 14 eller 16-bit) vil man kunne øke minste måleverdi, og på den måten øke oppløsningen fra sensoren. Man kan også oppnå en høyere oppløsning ved å benytte en meget rask omsetter og oversample som beskrevet i Kapittel 7.3.3.

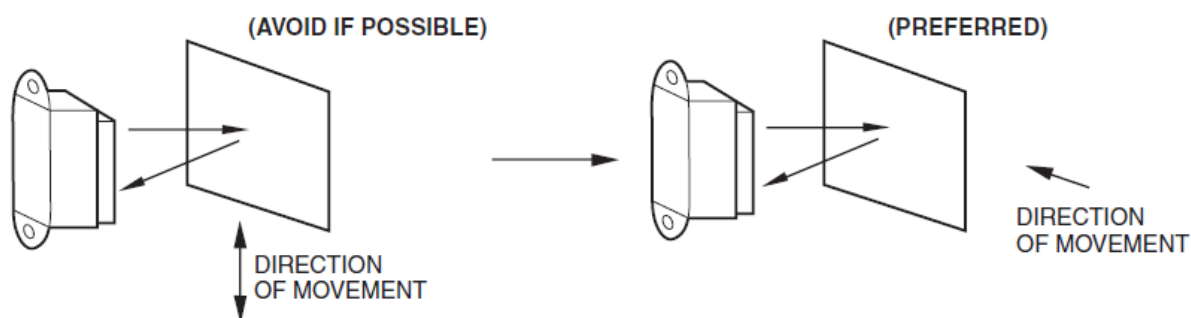
Dersom en 12-bit blir benyttet vil man basert på likning (7.6) kunne få en minste måleverdi på:

$$\text{minste måleverdi} = \frac{3,3V}{2^{16}} = 8,057 \cdot 10^{-4}V \quad (7.7)$$

Dette gir en oppløsning i intervallet mellom 1400 og 1500mm på 1,92mm. Benytter man en omsetter med 16-bits oppløsning kan man oppnå en oppløsning på 0,12mm, noe som kanskje er i overkant av hva det er behov for.

7.5 Montering av sensorer

I produsentens datablad (Vedlegg 6 og Vedlegg 7) er det illustrert hvordan sensorene bør monteres i forhold til bevegelsesretningen når sensoren skal beveges relativt til objektet den skal måle. Sensoren bør da vinkles slik at en står på høykant som vist i Figur 59.



Figur 59 – Anbefalt montering av infrarød sensor, hentet fra Vedlegg 6

7.6 Oppsummering

Som nevnt har de to sensorene relativt sett en ganske lik ytelse. Dersom vi sammenlikner den absolutte ytelsen til de to sensorene, som vist i Tabell 16, er det tydelig at sensoren med det minste måleområdet har den beste ytelsen når det gjelder oppløsning.

Tabell 16 – Sammenlikning av absolutt oppløsning med 10-bits oppløsning

	Beste oppløsning		Dårligste oppløsning	
	Oppløsning	Intervall	Oppløsning	Intervall
GP2D120	0,060mm	40 - 50mm	1,796mm	275 - 300mm
GP2Y0A02YK	0,516mm	200 - 300mm	7,741mm	1400 - 1500mm

Begge sensorene med den foreslåtte instrumenteringen, med en regulert strømtilførsel, kondensator og lavpassfilter, en signalkvalitet som er god nok for å kunne benyttes til styring av AGV-en.

Når det gjelder oppløsning er sensoren med det korteste måleområdet helt klart god nok for oppgaven, selv med en analog-til-digital-omsetter med 10-bits oppløsning. Sensoren med det lange måleområdet er imidlertid kanskje for dårlig til å brukes til dokkingoppgaven, særlig hvis man ønsker å benytte den skisserte prosedyren for dokking. Siden oppløsningen er så dårlig i den ytterste delen av måleområdet vil det bli vanskelig å benytte den skisserte løsningen og få nøyaktige svar. Dersom en annen prosedyre utvikles er det imidlertid kanskje mulig å benytte disse sensorene allikevel. Man vil også kunne øke oppløsningen ved å benytte en analog-til-digital-omsetter med høyere oppløsning. Å benytte en omsetter med en oppløsning på 12-bit vil som nevnt kunne øke oppløsningen til mer akseptable 1,92mm, noe som gjør det mye mer interessant å benytte disse sensorene.

Som en konklusjon på testingen av sensorene anbefales det at sensorene monteres med en regulert strømforsyning, en 100 μ F kondensator så nære sensoren som mulig og et lavpassfilter på utsignalet til sensoren. De infrarøde sensorene bør også implementeres med en analog-til-digital-omsetter med en oppløsning på minst 12-bit, og en samplingsfrekvens på minst 50Hz.

8 Videre arbeid

Med en analog-til-digital-omsetter med tilstrekkelig oppløsning og hastighet vil man kunne benytte de infrarøde sensorene til dokkingprosedyren. Denne prosedyren er imidlertid relativt komplisert, noe som gjør at den er både tidkrevende og mindre pålitelig enn det som er ønskelig. Basert på dette vil det være interessant å kunne teste hvorvidt det er mulig å benytte områdesensoren for å gjennomføre den første delen av dokkingen der AGV-en bestemmer sin positur relativt til dokken. Man kan legge inn en beskrivelse av hvordan dokken ser ut og hvor man ønsker at AGV-en skal kjøre inn. Datamaskinen om bord vil kunne bruke dette kartet og sammenlikne med avlesningene fra områdesensoren for å finne ut hvor AGV-en befinner seg, og hvor den må bevege seg for å kjøre inn i dokken. De infrarøde sensorene vil fortsatt ha en overlegen nøyaktighet når AGV-en først er kommet inn i dokken. Dette vil sannsynligvis være en mer robust og mer fleksibel løsning. Den er fleksibel siden man lettere kan benytte flere dokker med forskjellig utforming. I tillegg er den mer robust siden prosessen med å bestemme positur relativt til dokken vil gå fortere når den ikke involverer bevegelser med AGV-en. Man blir også mer fleksibel i forhold til startavstand for dokkingoperasjonen siden områdesensoren har et mye større måleområde enn det de infrarøde sensorene har.

Mekanisk låsing av AGV-en under dokking er også aktuelt som videre arbeid; å kartlegge hvorvidt det er behov for låsing av AGV-en under dokking, og å bestemme hvilke metoder som kan brukes for å låse AGV-en.

9 Oppsummering og konklusjoner

En ny AGV har blitt tegnet, bygget og satt i drift. AGV-en kjøres nå i automatiseringslaboratoriet ved Institutt for produksjons- og kvalitetsteknikk ved NTNU. AGV-en styres ved hjelp av et styresystem utviklet for uttesting av distribuert agentbasert styring. AGV-ens konstruksjon og de komponentene som inngår i AGV-en er beskrevet i denne oppgaven. En driver har blitt utviklet for å kunne styre AGV-ens motorkontrollere fra operativsystemet Linux.

En rekke metoder og sensorer som er aktuelle å benytte i en dokkingoperasjon har blitt kartlagt. De ulike sensorenes virkemåte, styrker og svakheter har blitt presentert. Ut fra vurderinger som ble gjort ble det bestemt å kjøpe inn en rekke infrarøde distansesensorer. Særlig var det sensorenes meget lave pris som gjorde at det var interessant å benytte dem. En metode for dokking av AGV-en ble utviklet med tanke på å benytte disse sensorene. De infrarøde sensorene ble grundig testet for å forsikre at de kunne yte tilfredsstillende for kunne brukes til å styre AGV-en. Konklusjonen fra denne testingen er en anbefaling om hvordan sensorene bør implementeres. Dersom denne anbefalingen følges vil sensorene ha en høy nok oppløsning og presisjon til å kunne benyttes til å styre AGV-en i dokkingoperasjonen.

Den skisserte løsningen baserer seg på å bestemme AGV-ens relative posisjon i forhold til dokken ved hjelp av de infrarøde sensorene. Deretter kjører AGV-en til en posisjon på dokkens senterlinje for å kunne kjøre inn i dokken. Når AGV-en er inne i dokken tar et nytt sett infrarøde sensorer over og posisjonerer AGV-en nøyaktig inne i selve dokken.

Noen problemer har også oppstått under arbeidet med oppgaven; sen levering av komponenter og en mer komplisert implementering av motorkontrollerne har ført til at bygging og utprøving av dokkingssystemet ikke har kommet så langt som ønskelig. Delen av oppgaven som gikk på bygging av ny AGV viste seg også å være mer tidkrevende enn først antatt.

10 Engelsk-Norsk Ordliste

På grunn av at størstedelen av litteraturen som er brukt i arbeidet med oppgaven er på engelsk, mens oppgaven skrives på norsk, oppstår det en del usikkerhet i forhold til fagterminologi som finnes på engelsk, men som ikke har en god oversettelse på norsk. I de tilfellene der et godt norsk ord ikke har blitt funnet har det kommet forslag på fornorskinger eller oversettelinger. En oversikt over disse oversettingene finnes i denne engelsk-norske ordlisten.

Tabell 17 – Engelsk-Norsk ordliste

Engelsk	Norsk
AGV, Automated Guided Vehicle	automatisk selvstyrt vogn
analog to digital converter (ADC)	analog-til-digital-omsetter
bit shift	bit-skift
dead reckoning	bestikknavigasjon
differential drive	differensialdrift
encoder	enkoder
forward kinematics	kinematikkmodell
ICC, instantaneous centre of curvature	momentant rotasjonscenter
inverse kinematics	inverskinematikk
nominal voltage	pålydende spenning
non-holonomic	ikke-holonomisk
odometry	odometri
oversampling	oversampling
pose	positur
range finder	områdesensor
stall torque	oppstartsmoment
target velocity	målhastighet
time of flight (TOF)	refleksjonstid
vision	maskinsyn

11 Referanser

1. SHARP Corporation, *General Application Note: Distance Measuring Sensors*. 2003, Opto-Electronic Devices Division,.
2. GeekNet Inc. *libEPOS - command an EPOS motor control*. 2010 [hentet 25. april 2010]; Tilgjengelig fra: <http://sourceforge.net/projects/libepos/>.
3. Roulet-Dubonnet, O., et al., *Development of a Holonic Free-Roaming AGV System for Part Manufacturing*. 2009. p. 215-224.
4. CAN in Automation (CiA). *CAN in Automation (CiA): Controller Area Network (CAN)*. 2010 [hentet 5. mai 2010]; Tilgjengelig fra: <http://can-cia.org/>.
5. Maxon Motor ag, *EPOS Positioning controller, Application Note "Device Programming"*. 2007.
6. Berulfsen, B. and D. Gundersen. *Blå fremmedordbok*. 2010 [hentet 26. februar 2010]; Tilgjengelig fra: <http://www.ordnett.no>.
7. Siegwart, R. and I.R. Nourbakhsh, *Introduction to autonomous mobile robots*. 2004, Cambridge, Mass.: MIT Press. IX, 321 s.
8. Everett, H.R., *Sensors for mobile robots: theory and application*. 1995, Natick, Mass.: A.K. Peters. xiv, 528 p.
9. Dudek, G. and M.R.M. Jenkin, *Computational principles of mobile robotics*. 2000, Cambridge: Cambridge University Press. XII, 280 s.
10. Minguez, J., F. Lamiroux, and J.-P. Laumond, *Motion Planning and Obstacle Avoidance*. 2008. p. 827-852.
11. Astolfi, A. *Exponential Stabilization of a Mobile Robot*. i *3rd European Control Conference*. 1995. Rome.
12. Lee, S.O., et al. *A stable target-tracking control for unicycle mobile robots*. i *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2000.
13. SICK AG - Division Auto Ident, *Operation Instructions, Laser Measurement Systems of the LMS100 Product Family*. 2009.
14. Gao, C., et al., *Autonomous docking of a smart wheelchair for the Automated Transport and Retrieval System (ATRS)*. *Journal of Field Robotics*, 2008. **25**(4): p. 203.
15. Christensen, H.I. and G.D. Hager, *Sensing and Estimation*. 2008. p. 87-107.
16. Fisher, R.B. and K. Konolige, *Range Sensors*. 2008. p. 521-542.
17. Desouza, G.N. and A.C. Kak, *Vision for mobile robot navigation: a survey*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2002. **24**(2): p. 237-267.
18. Bonin-Font, F., A. Ortiz, and G. Oliver, *Visual Navigation for Mobile Robots: A Survey*. *Journal of Intelligent and Robotic Systems*, 2008. **53**(3): p. 263-296.
19. Barnes, N. and Z.-Q. Liu, *Knowledge-based vision-guided robots*. 2002, Heidelberg: Physica-Verlag. XII, 234 s.
20. Borenstein, J., *Mobile robot positioning: Sensors and techniques*. *Journal of robotic systems*, 1997. **14**(4): p. 231.
21. Amat, J., et al. *Optimal landmark pattern for precise mobile robots dead-reckoning*. i *2001 ICRA IEEE International Conference on Robotics and Automation*. 2001.
22. Santos-Victor, J. and G. Sandini, *Visual Behaviors for Docking*. *Computer Vision and Image Understanding*, 1997. **67**(3): p. 223-238.
23. Low, E.M.P., I.R. Manchester, and A.V. Savkin, *A biologically inspired method for vision-based docking of wheeled mobile robots*. *Robotics and Autonomous Systems*, 2007. **55**(10): p. 769-784.
24. Cassinis, R., et al., *Docking and charging system for autonomous mobile robots*. Tech. Rep. R.T.2005-02-4. DEA, University of Brescia, Brescia, Italy,, 2005.
25. Borenstein, J., H.R. Everett, and L. Feng, eds. *Navigating Mobile Robots: Systems and Techniques*. 1996, AK Peters: Wellesley, Mass.

26. Castleberry, G.A., *The AGV handbook: a handbook for the selection of automated guided vehicle systems*. 1991, Ann Arbor, Mich.: Braun-Brumfield. XIX, 263 s.
27. Husqvarna Group. *Robotgressklippere for villaeiere | Husqvarnas Automower®*. 2009 [hentet 6. mars 2009]; Tilgjengelig fra: <http://www.husqvarna.com/no/homeowner/products/robotic-mowers/husqvarna-robotic-mowers-for-homeowners/>.
28. Petterson, U. and B.-A. Bergvall, *Electronic Directing System*, W.I.P. Organization, Editor. 2003, Aktiebolaget Electrolux.
29. Migatron Corp. *RPS-412A HIGH ACCURACY SENSOR*. 2010 [hentet 22. april 2010]; Tilgjengelig fra: <http://www.migatron.com/products/rps-412a-v-1/index.htm>.
30. YeRobot Technology Ltd. *URM37 V3.2*. 2009 [hentet 6.mai 2009]; Tilgjengelig fra: <http://www.yerobot.com/urm37-v3-2.html>.
31. Tong, F., S.K. Tso, and T.Z. Xu, *A high precision ultrasonic docking system used for automatic guided vehicle*. *Sensors and Actuators A: Physical*, 2005. **118**(2): p. 183-189.
32. Vaz, P.M., et al. *Docking of a mobile platform based on infrared sensors*. i *ISIE 97 Proceeding of the IEEE International Symposium on Industrial Electronics ISIE-97*. 1997.
33. Benet, G., et al., *Using infrared sensors for distance measurement in mobile robots*. *Robotics and Autonomous Systems*, 2002. **40**(4): p. 255-266.
34. Mohammad, T., *Using Ultrasonic and Infrared Sensors for Distance Measurement*. World Academy of Science, Engineering and Technology, 2009. **51**: p. 293-298.
35. ELFA Elektronik AS. *Distansesensorer*. 2010 [hentet 25. april 2010]; Tilgjengelig fra: https://www1.elfa.se/elfa3~no_no/elfa/init.do?shop=ELFA_NO-NO#toc=20392;
36. Everett, C.H.R., *Survey of collision avoidance and ranging sensors for mobile robots*. *Robotics and Autonomous Systems*, 1989. **5**(1): p. 5-67.
37. Weisstein, E.W. *"Bézier Curve" From MathWorld--A Wolfram Web Resource*. . 2010 [hentet 15. april 2010]; Tilgjengelig fra: <http://mathworld.wolfram.com/BezierCurve.html>.
38. Arduino Team. *ArduinoBoardDecimilla*. 2010 [hentet 5. mai 2010]; Tilgjengelig fra: <http://arduino.cc/en/Main/ArduinoBoardDiecimila>.
39. Wilmshurst, T.H., *Low-pass filtering and visual averaging*, in *Signal Recovery from Noise in Electronic Instrumentation, Second Edition*. 2010, Taylor & Francis.
40. Atmel Corporation, *AVR121 : Enhancing ADC resolution by oversampling, Application Note*. 2005.

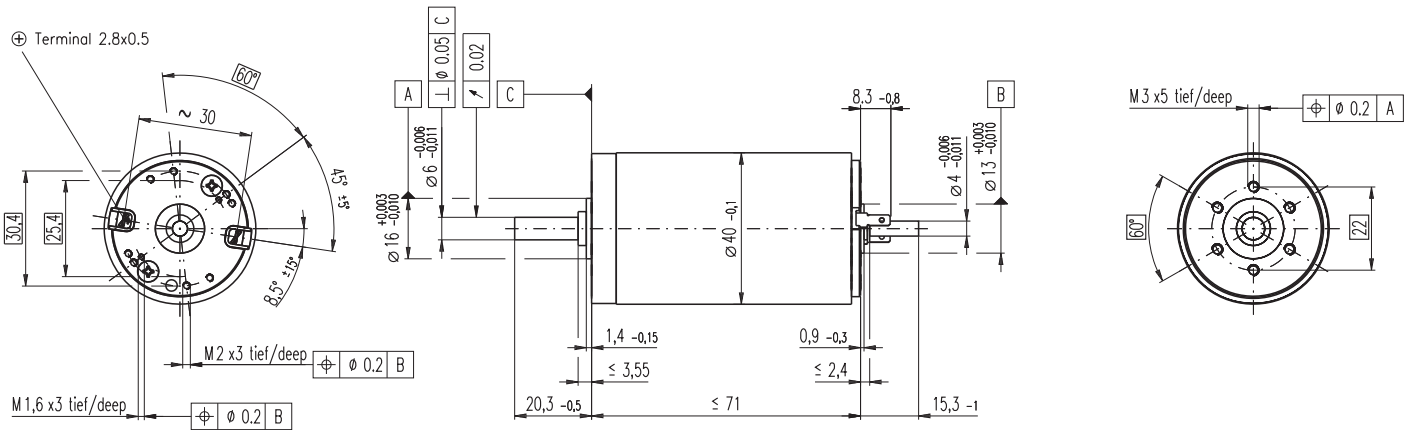
Vedlegg 1

Teknisk dokumentasjon motor, Maxon RE 40

Ordrenummer: 148867

[hentet fra <http://www.maxonmotor.com/>]

RE 40 Ø40 mm, Graphite Brushes, 150 Watt



M 1:2

- Stock program
- Standard program
- Special program (on request)

Order Number

148866 148867 148877 218008 218009 218010 218011 218012 218013 218014

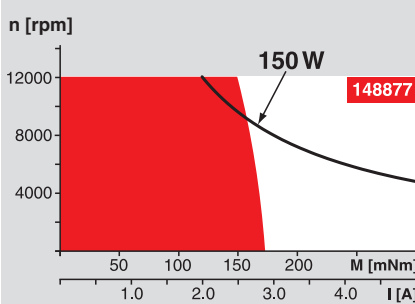
Motor Data

Values at nominal voltage		148866	148867	148877	218008	218009	218010	218011	218012	218013	218014
1	Nominal voltage	V	12.0	24.0	48.0	48.0	48.0	48.0	48.0	48.0	48.0
2	No load speed	rpm	6920	7580	7580	6420	5560	3330	2690	2130	1420
3	No load current	mA	241	137	68.6	53.7	43.7	21.9	16.7	12.5	7.77
4	Nominal speed	rpm	6370	6930	7000	5810	4920	2700	2050	1500	1080
5	Nominal torque (max. continuous torque)	mNm	94.9	170	184	183	177	187	189	189	188
6	Nominal current (max. continuous current)	A	6.00	5.77	3.12	2.62	2.20	1.38	1.12	0.898	0.721
7	Stall torque	mNm	1680	2280	2500	1990	1580	995	796	641	512
8	Starting current	A	102	75.7	41.4	28.0	19.2	7.26	4.68	3.00	1.92
9	Max. efficiency	%	88	91	92	91	91	89	88	87	86
Characteristics											
10	Terminal resistance	Ω	0.117	0.317	1.16	1.72	2.50	6.61	10.2	16.0	24.9
11	Terminal inductance	mH	0.0245	0.0823	0.329	0.460	0.612	1.70	2.62	4.14	6.40
12	Torque constant	mNm / A	16.4	30.2	60.3	71.3	82.2	137	170	214	266
13	Speed constant	rpm / V	581	317	158	134	116	69.7	56.2	44.7	35.9
14	Speed / torque gradient	rpm / mNm	4.15	3.33	3.04	3.23	3.53	3.36	3.39	3.35	3.37
15	Mechanical time constant	ms	6.03	4.81	4.39	4.36	4.35	4.31	4.31	4.31	4.32
16	Rotor inertia	gcm ²	139	138	138	129	118	123	121	123	120

Specifications

- Thermal data**
- 17 Thermal resistance housing-ambient 4.65 K / W
 - 18 Thermal resistance winding-housing 1.93 K / W
 - 19 Thermal time constant winding 41.6 s
 - 20 Thermal time constant motor 1120 s
 - 21 Ambient temperature -30 ... +100°C
 - 22 Max. permissible winding temperature +155°C
- Mechanical data (ball bearings)**
- 23 Max. permissible speed 12000 rpm
 - 24 Axial play 0.05 - 0.15 mm
 - 25 Radial play 0.025 mm
 - 26 Max. axial load (dynamic) 5.6 N
 - 27 Max. force for press fits (static) 110 N
 - (static, shaft supported) 1200 N
 - 28 Max. radial loading, 5 mm from flange 28 N

Operating Range



Comments

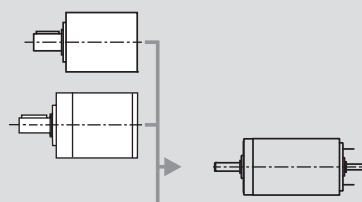
- Continuous operation**
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.
- Short term operation**
The motor may be briefly overloaded (recurring).
- Assigned power rating**

Other specifications

- 29 Number of pole pairs 13
 - 30 Number of commutator segments 13
 - 31 Weight of motor 480 g
- Values listed in the table are nominal.
Explanation of the figures on page 49.
- Option**
Preloaded ball bearings

maxon Modular System

- Planetary Gearhead**
Ø42 mm
3 - 15 Nm
Page 240
- Planetary Gearhead**
Ø52 mm
4 - 30 Nm
Page 243



- Recommended Electronics:**
- ADS 50/5 Page 282
 - ADS 50/10 283
 - ADS_E 50/5 283
 - ADS_E 50/10 283
 - EPOS2 24/5 303
 - EPOS2 50/5 303
 - EPOS 70/10 303
 - EPOS P 24/5 306
 - Notes 18

Overview on page 16 - 21

- Encoder MR**
256 - 1024 CPT,
3 channels
Page 265
- Encoder HED_ 5540**
500 CPT,
3 channels
Page 268 / 270
- Brake AB 28**
Ø45 mm
24 VDC, 0.4 Nm
Page 316
- Industrial Version Encoder HEDL 9140**
Page 273
- Brake AB 28**
Page 317

Vedlegg 2

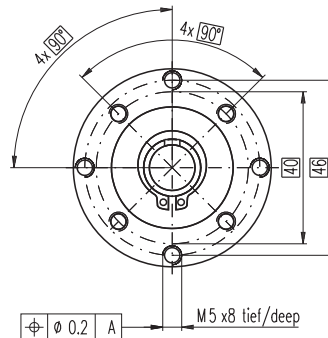
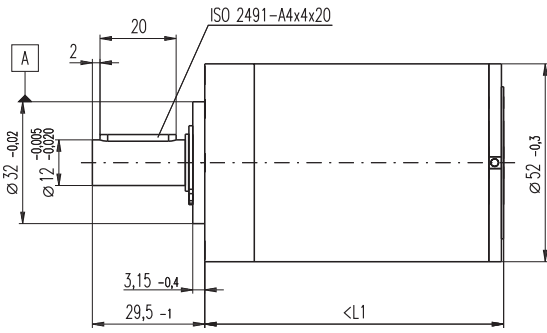
Teknisk dokumentasjon gir, Maxon GP 52 C

Ordrenummer: 223093

[hentet fra <http://www.maxonmotor.com/>]

Planetary Gearhead GP 52 C $\varnothing 52$ mm, 4 - 30 Nm

Ceramic Version



M 1:2

Technical Data

Planetary Gearhead	straight teeth
Output shaft	stainless steel
Bearing at output	preloaded ball bearings
Radial play, 12 mm from flange	max. 0.06 mm
Axial play at axial load	< 5 N 0 mm > 5 N max. 0.3 mm
Max. permissible axial load	200 N
Max. permissible force for press fits	500 N
Sense of rotation, drive to output	=
Recommended input speed	< 6000 rpm
Recommended temperature range	-20 ... +100°C
Extended area as option	-35 ... +100°C
Number of stages	1 2 3 4
Max. radial load, 12 mm from flange	500 N 700 N 900 N 900 N

- Stock program
- Standard program
- Special program (on request)

Gearhead Data

	223080	223083	223089	223094	223097	223104	223109
1 Reduction	3.5 : 1	12 : 1	43 : 1	91 : 1	150 : 1	319 : 1	546 : 1
2 Reduction absolute	$7/2$	$49/4$	$343/6$	91	$2401/16$	$637/2$	546
3 Mass inertia	gcm ² 20.7	17.6	17.3	16.7	17.3	16.8	16.4
4 Max. motor shaft diameter	mm 10	10	10	10	10	10	10
Order Number	223081	223084	223090	223095	223099	223105	223110
1 Reduction	4.3 : 1	15 : 1	53 : 1	113 : 1	186 : 1	353 : 1	676 : 1
2 Reduction absolute	$13/3$	$91/6$	$637/12$	$338/3$	$4459/24$	$28561/81$	676
3 Mass inertia	gcm ² 12	16.8	17.2	9.3	17.3	9.4	9.1
4 Max. motor shaft diameter	mm 8	10	10	8	10	8	8
Order Number		223085	223091	223096	223101	223106	223111
1 Reduction		19 : 1	66 : 1	126 : 1	230 : 1	394 : 1	756 : 1
2 Reduction absolute		$169/9$	$1183/18$	126	$8281/36$	$1183/3$	756
3 Mass inertia	gcm ²	9.5	16.7	16.4	16.8	16.7	16.4
4 Max. motor shaft diameter	mm	8	10	10	10	10	10
Order Number		223086	223092	223098	223102	223107	223112
1 Reduction		21 : 1	74 : 1	156 : 1	257 : 1	441 : 1	936 : 1
2 Reduction absolute		21	$147/2$	156	$1029/4$	441	936
3 Mass inertia	gcm ²	16.5	17.2	9.1	17.3	16.5	9.1
4 Max. motor shaft diameter	mm	10	10	8	10	10	8
Order Number		223087	223093		223103	223108	
1 Reduction		26 : 1	81 : 1		285 : 1	488 : 1	
2 Reduction absolute		26	$2197/27$		$15379/54$	$4394/9$	
3 Mass inertia	gcm ²	9.1	9.4		16.7	9.4	
4 Max. motor shaft diameter	mm	8	8		10	8	
5 Number of stages		1	2	3	3	4	4
6 Max. continuous torque	Nm	4	15	30	30	30	30
7 Intermittently permissible torque at gear output	Nm	6	22.5	45	45	45	45
8 Max. efficiency	%	91	83	75	75	68	68
9 Weight	g	460	620	770	770	920	920
10 Average backlash no load	°	0.6	0.8	1.0	1.0	1.0	1.0
11 Gearhead length L1	mm	49.0	65.0	78.5	78.5	92.0	92.0

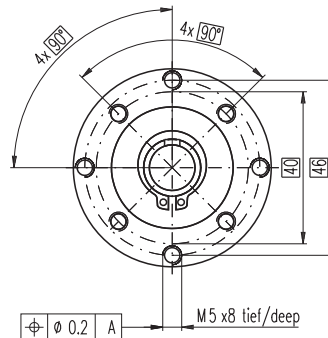
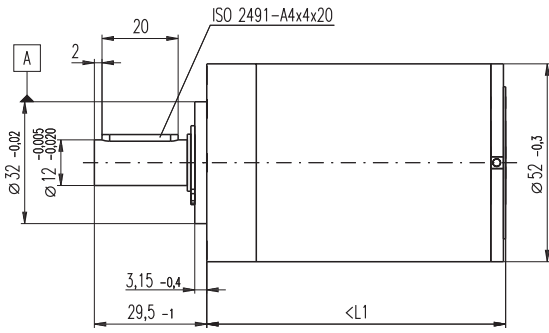


maxon Modular System

+ Motor	Page	+ Sensor	Page	+ Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor / brake) + assembly parts						
RE 40, 150 W	82					120.1	136.1	149.6	149.6	163.1	163.1	163.1
RE 40, 150 W	82	MR	265			131.5	147.5	161.0	161.0	174.5	174.5	174.5
RE 40, 150 W	82	HED_5540	268/270			140.8	156.8	170.3	170.3	183.8	183.8	183.8
RE 40, 150 W	82	HEDL 9140	273			174.1	190.1	203.6	203.6	217.1	217.1	217.1
RE 40, 150 W	82			AB 28	316	156.2	172.2	185.7	185.7	199.2	199.2	199.2
RE 40, 150 W	82			AB 28	317	164.2	180.2	193.7	193.7	207.2	207.2	207.2
RE 40, 150 W	82	HED_5540	268/270	AB 28	316	173.3	189.3	202.8	202.8	216.3	216.3	216.3
RE 40, 150 W	82	HEDL 9140	273	AB 28	317	184.6	200.6	214.1	214.1	227.6	227.6	227.6
RE 50, 200 W	83					157.1	173.1	186.6	186.6	200.1	200.1	200.1
RE 50, 200 W	83	HED_5540	269/270			177.8	193.8	207.3	207.3	220.8	220.8	220.8
EC 40, 120 W	157					119.1	135.1	148.6	148.6	162.1	162.1	162.1
EC 40, 120 W	157	HED_5540	269/271			137.5	153.5	167.0	167.0	180.5	180.5	180.5
EC 40, 120 W	157	Res 26	278			145.7	161.7	175.2	175.2	188.7	188.7	188.7
EC 40, 120 W	157			AB 28	316	149.9	165.9	179.4	179.4	192.9	192.9	192.9
EC 45, 150 W	158					160.3	176.3	189.8	189.8	203.3	203.3	203.3
EC 45, 150 W	158	HEDL 9140	273			175.9	191.9	205.4	205.4	218.9	218.9	218.9
EC 45, 150 W	158	Res 26	278			160.3	176.3	189.8	189.8	203.3	203.3	203.3
EC 45, 150 W	158			AB 28	317	167.7	183.7	197.2	197.2	210.7	210.7	210.7
EC 45, 150 W	158	HEDL 9140	273	AB 28	317	184.7	200.7	214.2	214.2	227.7	227.7	227.7
EC 45, 250 W	159					193.1	209.1	222.6	222.6	236.1	236.1	236.1
EC 45, 250 W	159	HEDL 9140	273			208.7	224.7	238.2	238.2	251.7	251.7	251.7
EC 45, 250 W	159	Res 26	278			193.1	209.1	222.6	222.6	236.1	236.1	236.1
EC 45, 250 W	159			AB 28	317	200.5	216.5	230.0	230.0	243.5	243.5	243.5
EC 45, 250 W	159	HEDL 9140	273	AB 28	317	217.5	233.5	247.0	247.0	260.5	260.5	260.5

Planetary Gearhead GP 52 C $\varnothing 52$ mm, 4 - 30 Nm

Ceramic Version



M 1:2

Technical Data

Planetary Gearhead	straight teeth
Output shaft	stainless steel
Bearing at output	preloaded ball bearings
Radial play, 12 mm from flange	max. 0.06 mm
Axial play at axial load	0 mm
	> 5 N max. 0.3 mm
Max. permissible axial load	200 N
Max. permissible force for press fits	500 N
Sense of rotation, drive to output	=
Recommended input speed	< 6000 rpm
Recommended temperature range	-20 ... +100°C
Extended area as option	-35 ... +100°C
Number of stages	1 2 3 4
Max. radial load, 12 mm from flange	500 N 700 N 900 N 900 N

- Stock program
- Standard program
- Special program (on request)

Gearhead Data

	223080	223083	223089	223094	223097	223104	223109
1 Reduction	3.5 : 1	12 : 1	43 : 1	91 : 1	150 : 1	319 : 1	546 : 1
2 Reduction absolute	$7/2$	$49/4$	$343/6$	91	$2401/16$	$637/2$	546
3 Mass inertia	gcm ² 20.7	17.6	17.3	16.7	17.3	16.8	16.4
4 Max. motor shaft diameter	mm 10	10	10	10	10	10	10
Order Number	223081	223084	223090	223095	223099	223105	223110
1 Reduction	4.3 : 1	15 : 1	53 : 1	113 : 1	186 : 1	353 : 1	676 : 1
2 Reduction absolute	$13/3$	$91/6$	$637/12$	$338/3$	$4459/24$	$28561/81$	676
3 Mass inertia	gcm ² 12	16.8	17.2	9.3	17.3	9.4	9.1
4 Max. motor shaft diameter	mm 8	10	10	8	10	8	8
Order Number		223085	223091	223096	223101	223106	223111
1 Reduction		19 : 1	66 : 1	126 : 1	230 : 1	394 : 1	756 : 1
2 Reduction absolute		$169/9$	$1183/18$	126	$8281/36$	$1183/3$	756
3 Mass inertia	gcm ²	9.5	16.7	16.4	16.8	16.7	16.4
4 Max. motor shaft diameter	mm	8	10	10	10	10	10
Order Number		223086	223092	223098	223102	223107	223112
1 Reduction		21 : 1	74 : 1	156 : 1	257 : 1	441 : 1	936 : 1
2 Reduction absolute		21	$147/2$	156	$1029/4$	441	936
3 Mass inertia	gcm ²	16.5	17.2	9.1	17.3	16.5	9.1
4 Max. motor shaft diameter	mm	10	10	8	10	10	8
Order Number		223087	223093		223103	223108	
1 Reduction		26 : 1	81 : 1		285 : 1	488 : 1	
2 Reduction absolute		26	$2197/27$		$15379/54$	$4394/9$	
3 Mass inertia	gcm ²	9.1	9.4		16.7	9.4	
4 Max. motor shaft diameter	mm	8	8		10	8	
5 Number of stages		1	2	3	3	4	4
6 Max. continuous torque	Nm	4	15	30	30	30	30
7 Intermittently permissible torque at gear output	Nm	6	22.5	45	45	45	45
8 Max. efficiency	%	91	83	75	75	68	68
9 Weight	g	460	620	770	770	920	920
10 Average backlash no load	°	0.6	0.8	1.0	1.0	1.0	1.0
11 Gearhead length L1	mm	49.0	65.0	78.5	78.5	92.0	92.0



maxon Modular System

+ Motor	Page	+ Sensor	Page	+ Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor / brake) + assembly parts						
EC-max 40, 120 W	173					137.1	153.1	166.6	166.6	180.1	180.1	180.1
EC-max 40, 120 W	173	MR	265			153.0	169.0	182.5	182.5	196.0	196.0	196.0
EC-max 40, 120 W	173	HEDL 9140	273			160.5	176.5	190.0	190.0	203.5	203.5	203.5
EC-max 40, 120 W	173			AB 28	315	177.1	193.1	206.6	206.6	220.1	220.1	220.1
EC-max 40, 120 W	173	HEDL 9140	273	AB 28	315	200.5	216.5	230.0	230.0	243.5	243.5	243.5
EC 90 flat, 90 W	199					81.4	97.4	110.9	110.9	124.4	124.4	124.4
EC 90 flat, 60 W	200					84.1	100.1	113.6	113.6	127.1	127.1	127.1

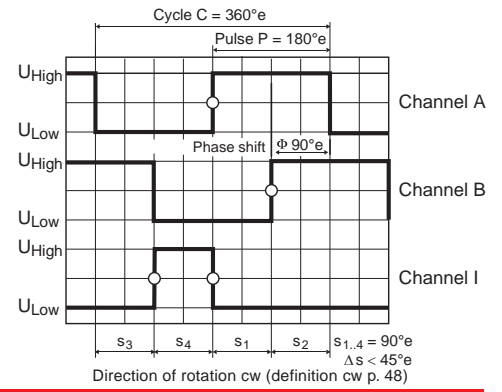
Vedlegg 3

Teknisk dokumentasjon enkoder, Maxon MR 500IMP

Ordrenummer: 228452

[hentet fra <http://www.maxonmotor.com/>]

Encoder MR, Type L, 256 - 1024 CPT, 3 Channels, with Line Driver



- Stock program
- Standard program
- Special program (on request)

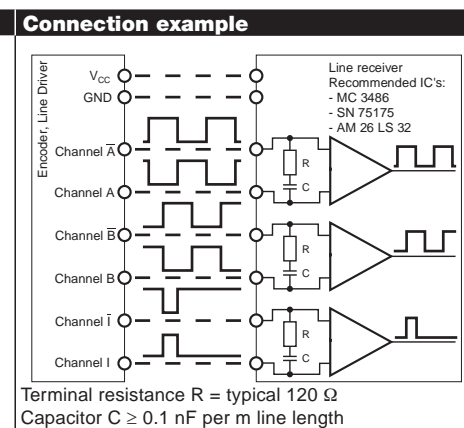
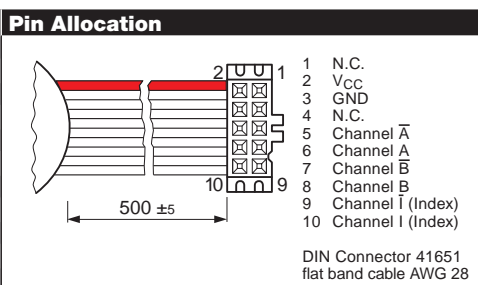
Order Number				
225783	228452	225785	228456	225787

Type					
Counts per turn	256	500	512	1000	1024
Number of channels	3	3	3	3	3
Max. operating frequency (kHz)	80	200	160	200	320
Max. speed (rpm)	18750	24000	18750	12000	18750



maxon Modular System						Overall length [mm] / • see Gearhead				
+ Motor	Page	+ Gearhead	Page	+ Brake	Page					
RE 30, 60 W	80					79.4	79.4	79.4	79.4	79.4
RE 30, 60 W	80	GP 32, 0.75 - 6.0 Nm	232/233			•	•	•	•	•
RE 30, 60 W	80	GP 32 S	249-251			•	•	•	•	•
RE 30, 60 W	80	GP 32, 0.75 - 4.5 Nm	230			•	•	•	•	•
RE 35, 90 W	81					82.4	82.4	82.4	82.4	82.4
RE 35, 90 W	81	GP 32, 0.75 - 4.5 Nm	230			•	•	•	•	•
RE 35, 90 W	81	GP 32, 0.75 - 6.0 Nm	232/233			•	•	•	•	•
RE 35, 90 W	81	GP 32, 4.0 - 8.0 Nm	235			•	•	•	•	•
RE 35, 90 W	81	GP 42, 3 - 15 Nm	238			•	•	•	•	•
RE 35, 90 W	81	GP 32 S	249-251			•	•	•	•	•
RE 40, 150 W	82					82.4	82.4	82.4	82.4	82.4
RE 40, 150 W	82	GP 42, 3 - 15 Nm	238			•	•	•	•	•
RE 40, 150 W	82	GP 52, 4 - 30 Nm	241			•	•	•	•	•
A-max 32	110/112					72.7	72.7	72.7	72.7	72.7
A-max 32	110/112	GP 32, 0.75 - 6.0 Nm	232/234			•	•	•	•	•
A-max 32	110/112	GS 38, 0.1 - 0.6 Nm	237			•	•	•	•	•
A-max 32	110/112	GP 32 S	249-251			•	•	•	•	•
EC-max 40, 70 W	168					73.9	73.9	73.9	73.9	73.9
EC-max 40, 70 W	168	GP 42, 3 - 15 Nm	239			•	•	•	•	•
EC-max 40, 120 W	169					103.9	103.9	103.9	103.9	103.9
EC-max 40, 120 W	169	GP 52, 4 - 30 Nm	242			•	•	•	•	•
EC-i 40, 50 W	190					42.0	42.0	42.0	42.0	42.0
EC-i 40, 50 W	190	GP 32, 1 - 6 Nm	234			•	•	•	•	•
EC-i 40, 50 W	190	GP 32 S	249-251			•	•	•	•	•
EC-i 40, 70 W	191					52.0	52.0	52.0	52.0	52.0
EC-i 40, 70 W	191	GP 32, 1 - 6 Nm	234			•	•	•	•	•
EC-i 40, 70 W	191	GP 32 S	249-251			•	•	•	•	•

Technical Data	
Supply voltage	5 V ± 5 %
Output signal	TTL compatible
Phase shift Φ	$90^\circ \pm 45^\circ e$
Index pulse width	$90^\circ \pm 45^\circ e$
Operating temperature rang	-25 ... +85°C
Moment of inertia of code wheel	$\leq 1.7 \text{ gcm}^2$
Output current per channel	max. 5 mA



The index signal I is synchronised with channel A or B.

Vedlegg 4

Teknisk dokumentasjon svivelhjul, Tente 2470PJH075P40

[hentet fra <http://www.tente-as.no>]

Product Information

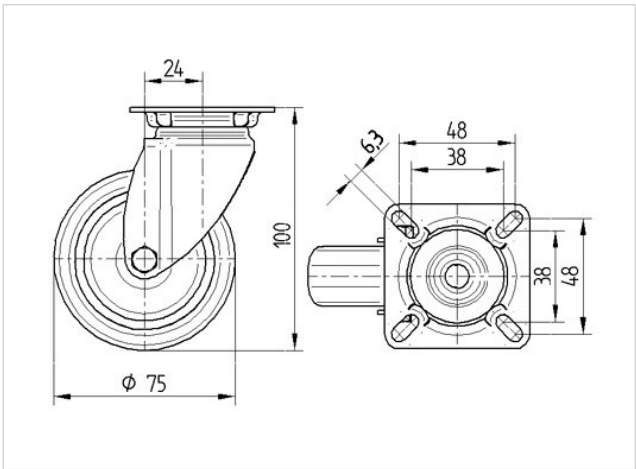


2470PJO075P40

EAN 4031582010404

Swivel castor, housing made of pressed steel, zinc-chromatised, double ball bearing swivel head. Wheel centre made of Polypropylene, Tread: TENTEprene (thermoplastic rubber), grey non-marking, plain bearing, plate fitting

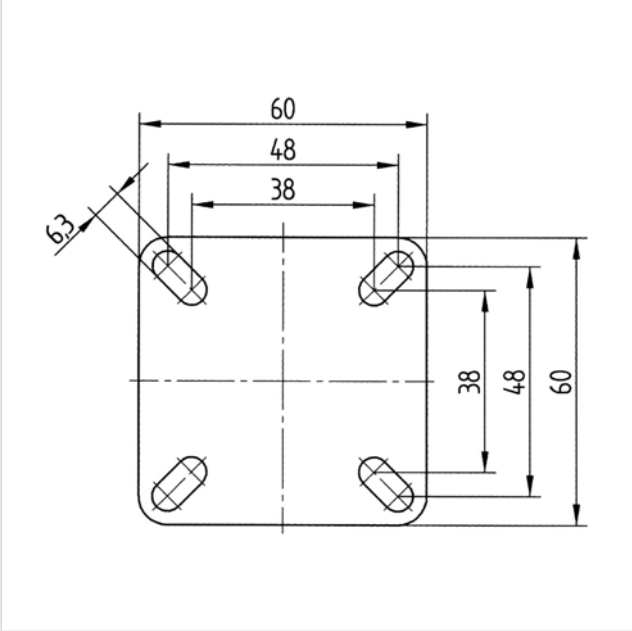
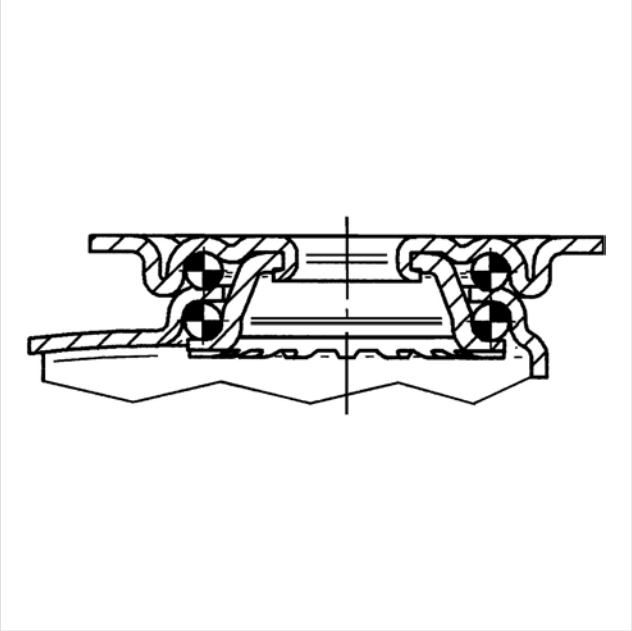
Technical Data	
Wheel diameter	75 mm
Width of Tread	25 mm
Size of Plate	60 x 60 mm
Plate hole centres	48/38 x 48/38 mm
Plate hole	6.3 mm
Offset	24 mm
Swivel Interference	123 mm
Overall height	100 mm
Temperature	- 20 / + 60 °C
Standard	EN 12530
Weight	0.242 kg
Dynamic Load capacity	75 kg
Static Load capacity	150 kg
Swivel Radius	61.5 mm
Hardness of tread	A 87 Shore



Roll performance	+	+	+	+	+
Movement noise	+	+	+	+	+
Attrition	+	+	+	+	+
Corrosion resistance	+	+	+	+	+

2470PJO075P40

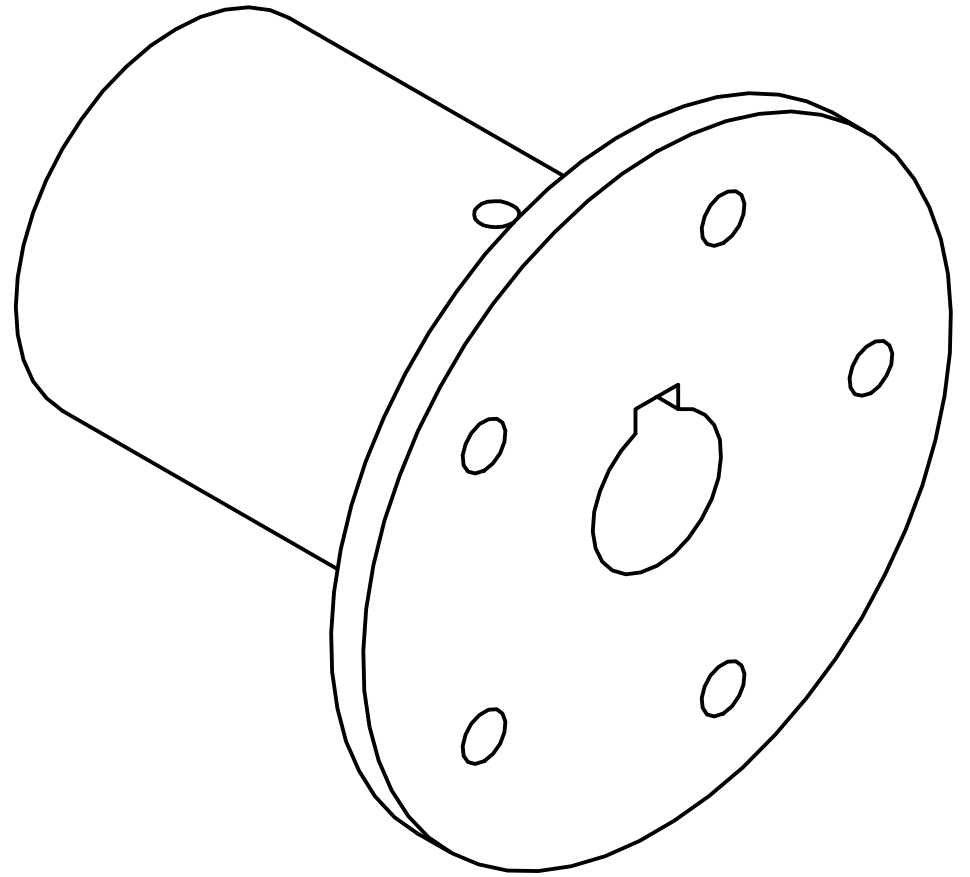
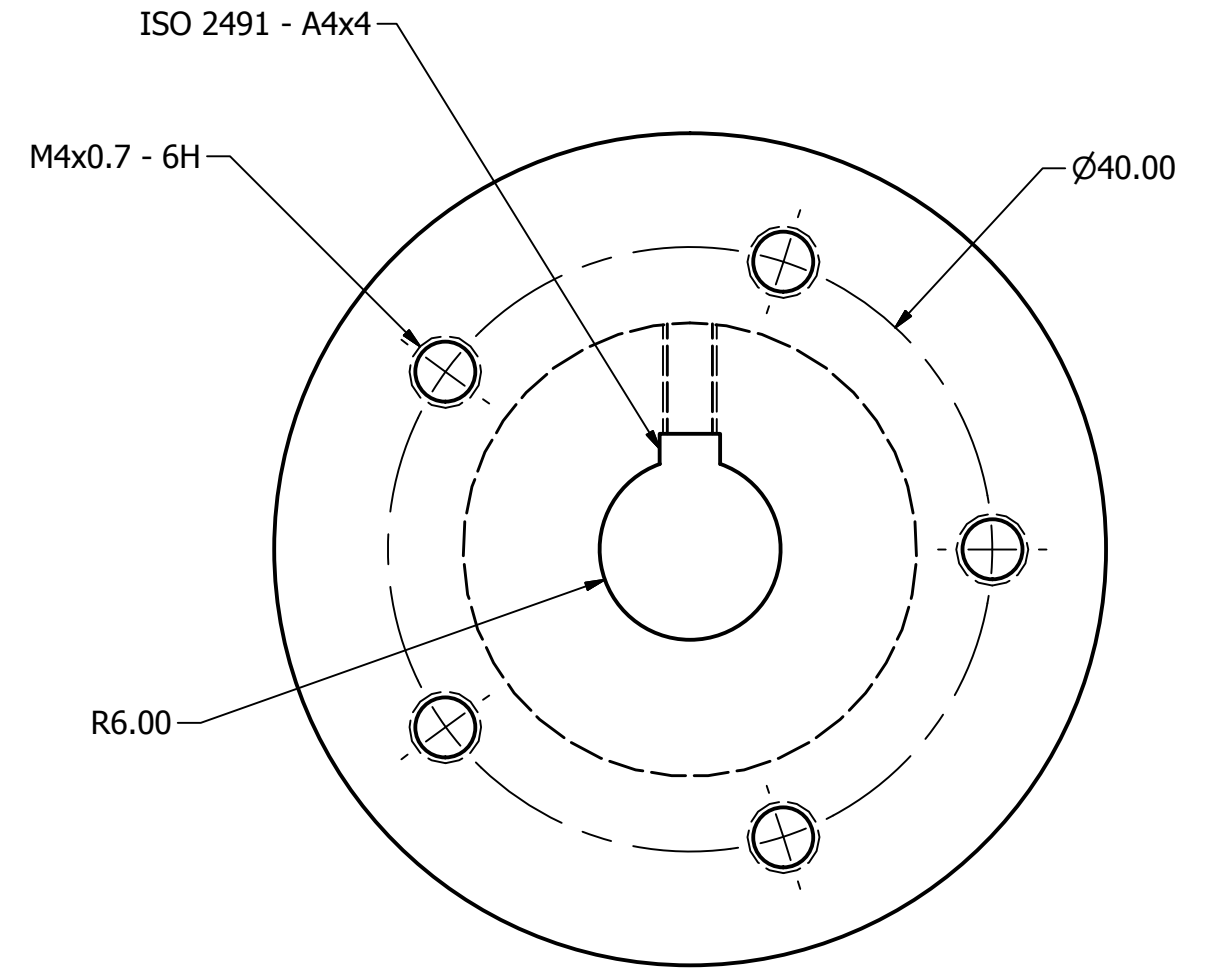
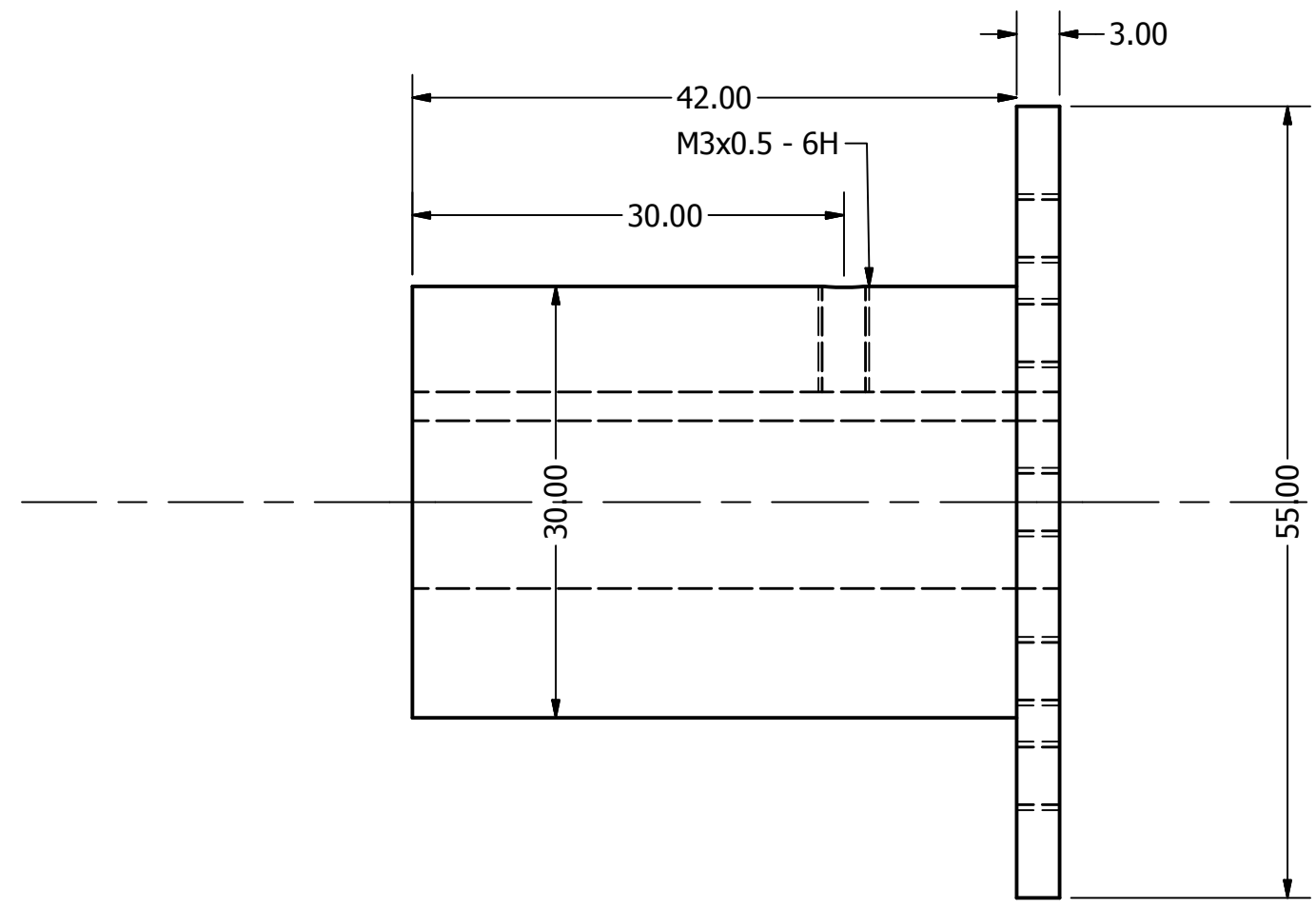
EAN 4031582010404



Vedlegg 5

Arbeidstegninger av komponenter til AGV

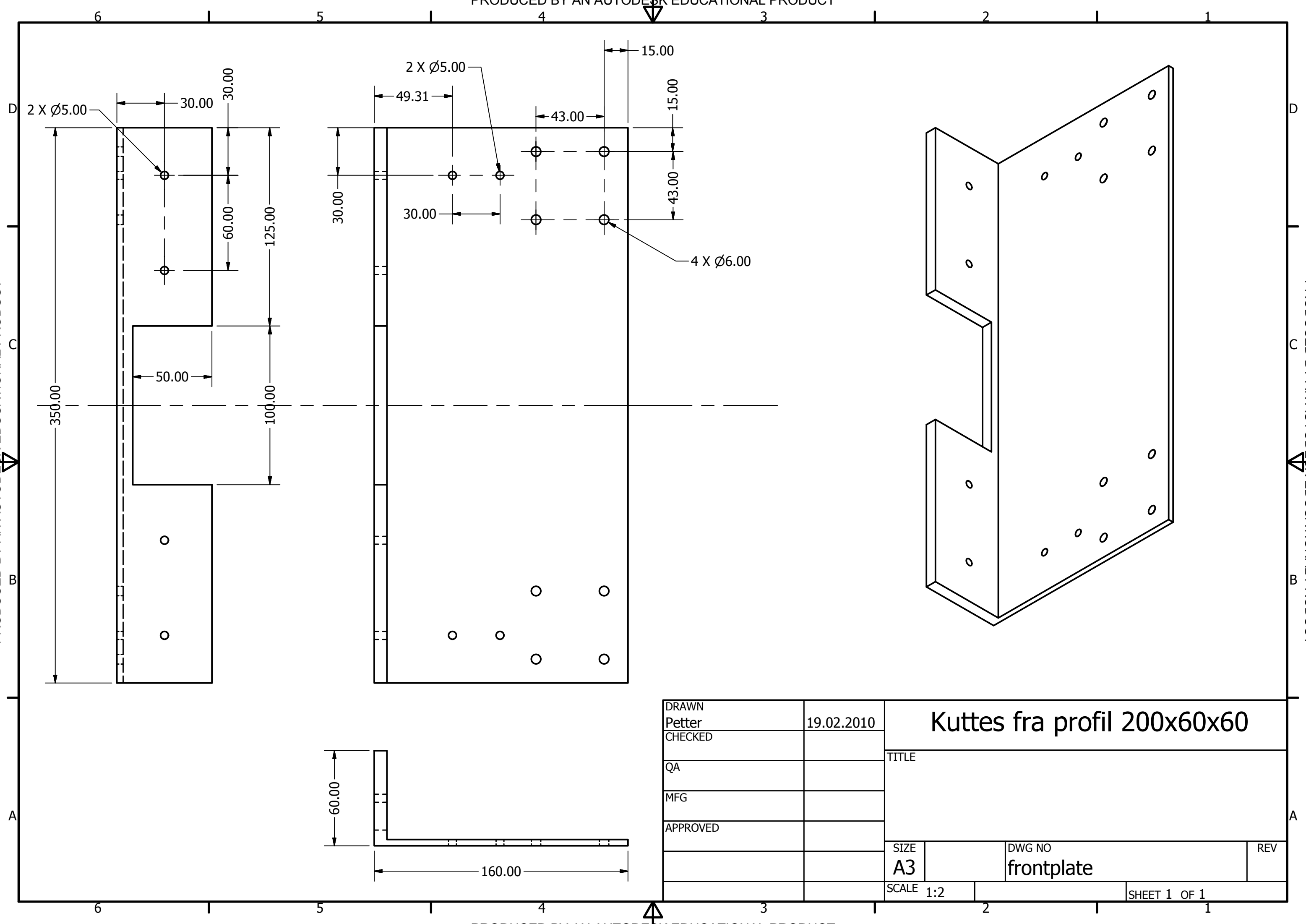
Tegningene finnes også i A3-format på vedlagte CD-ROM



DRAWN	Petter	19.02.2010		
CHECKED			TITLE	
QA				
MFG				
APPROVED				
			SIZE	DWG NO
			A3	boss
			SCALE	REV
			2:1	
			SHEET 1 OF 1	

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT



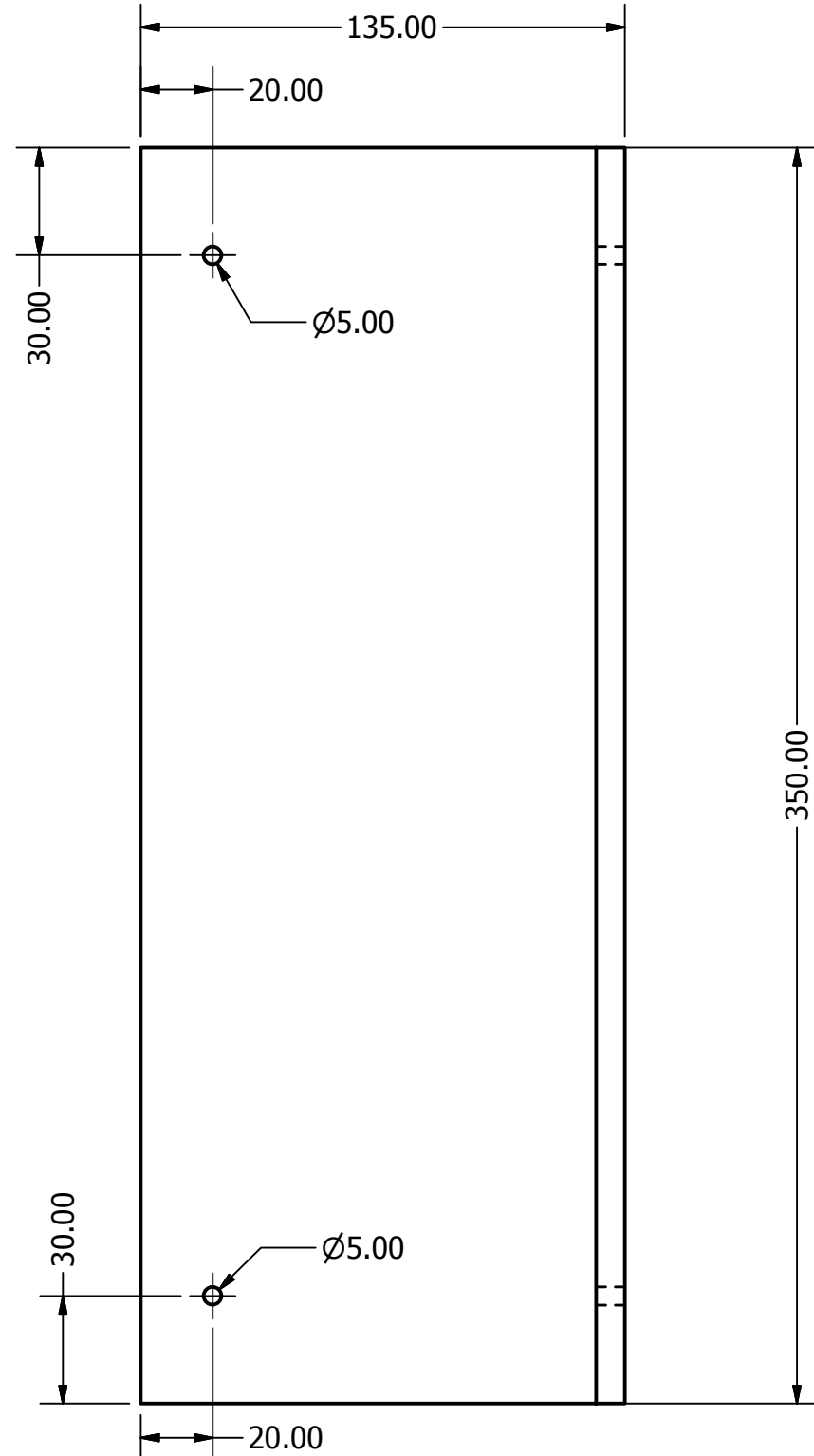
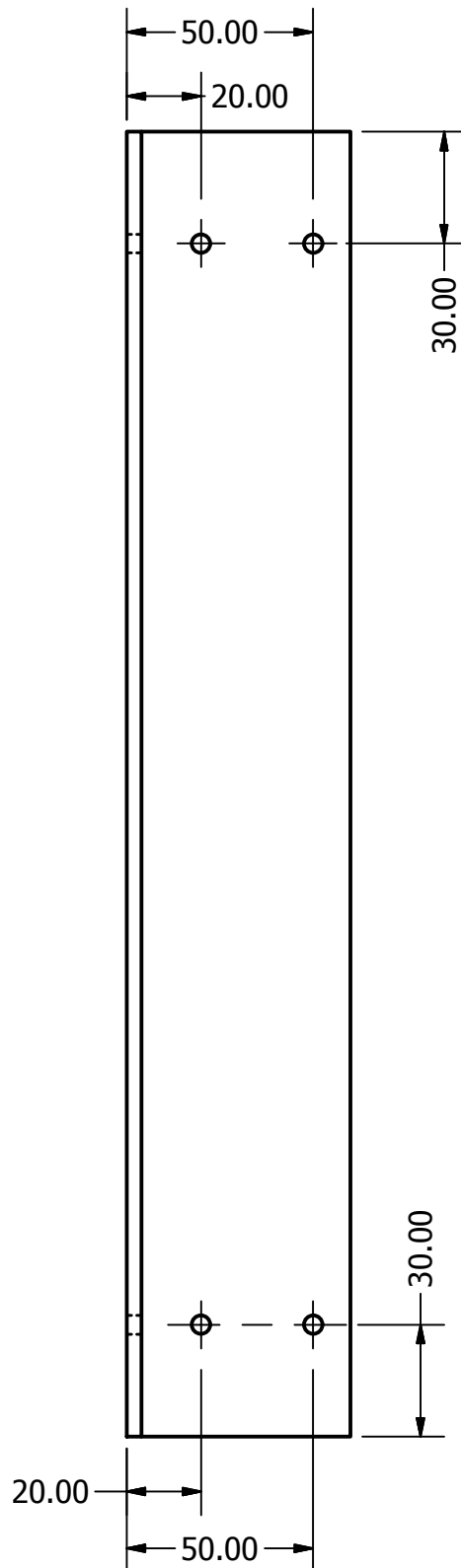
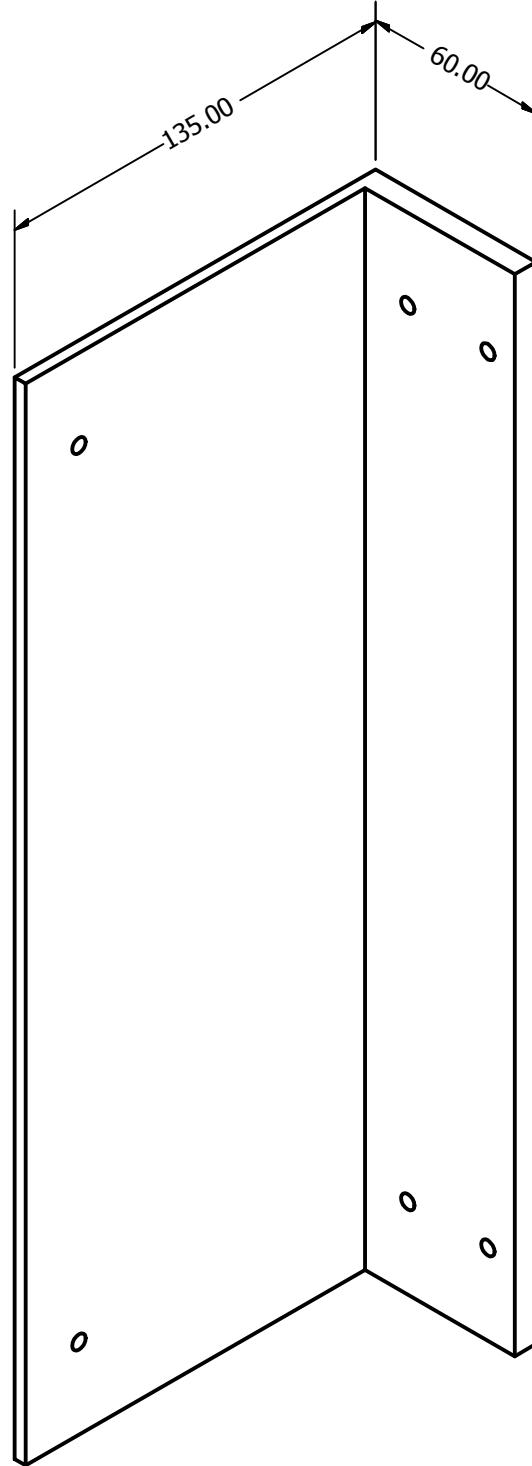
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

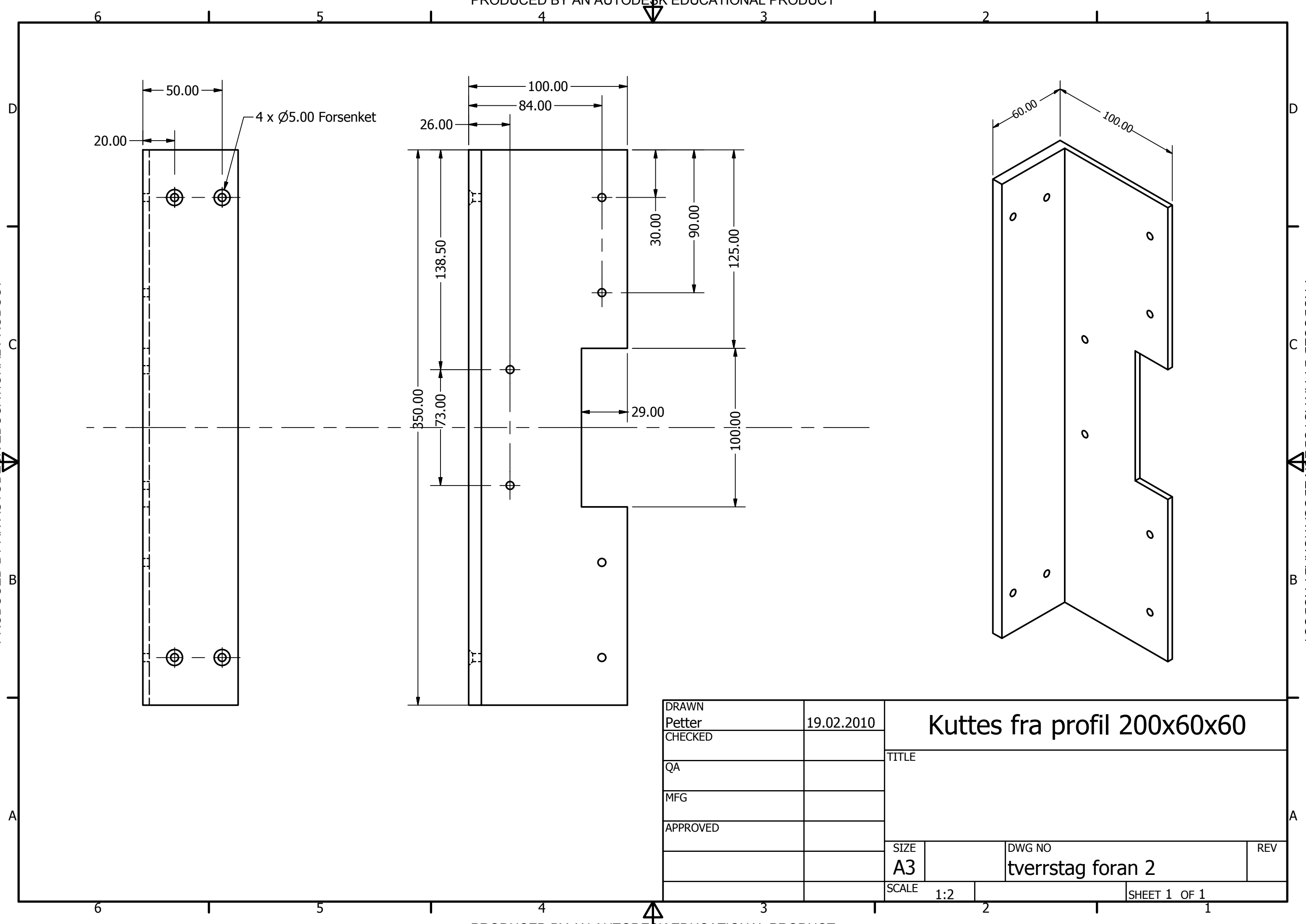
DRAWN Petter	19.02.2010	Kuttes fra profil 200x60x60		
CHECKED				
QA		TITLE		
MFG				
APPROVED		SIZE A3	DWG NO frontplate	REV
		SCALE 1:2	SHEET 1 OF 1	

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

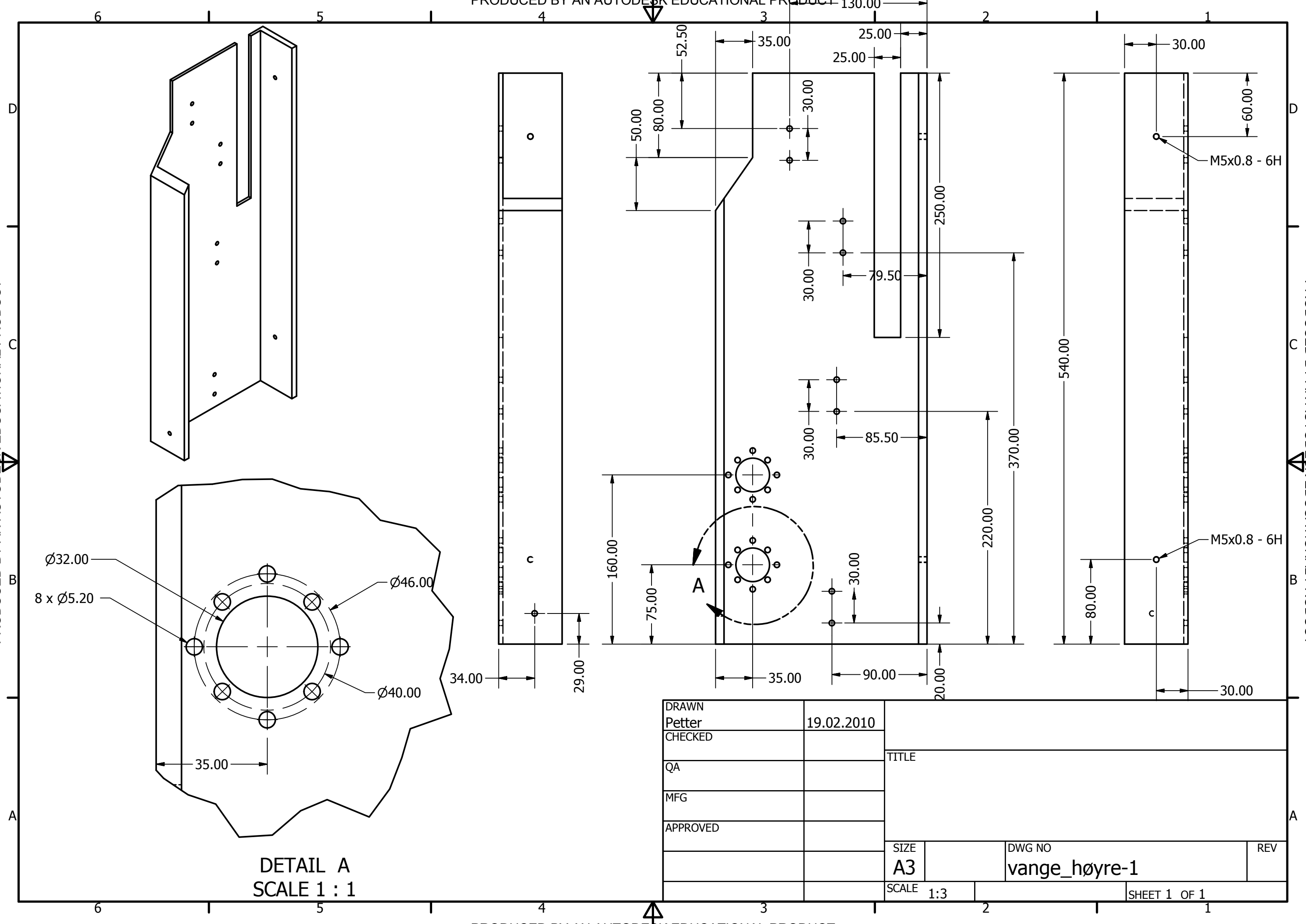
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT



DRAWN Petter	21.02.2010	Kuttet fra profil 200x60x60		
CHECKED				
QA		TITLE		
MFG				
APPROVED				
		SIZE A3	DWG NO tverrstag bak	REV
		SCALE 1:2	SHEET 1 OF 1	



DRAWN	Petter	19.02.2010	Kuttet fra profil 200x60x60		
CHECKED	QA				
	MFG		TITLE		
	APPROVED				
			SIZE	DWG NO	REV
			A3	tverrstag foran 2	
			SCALE	SHEET 1 OF 1	
			1:2		

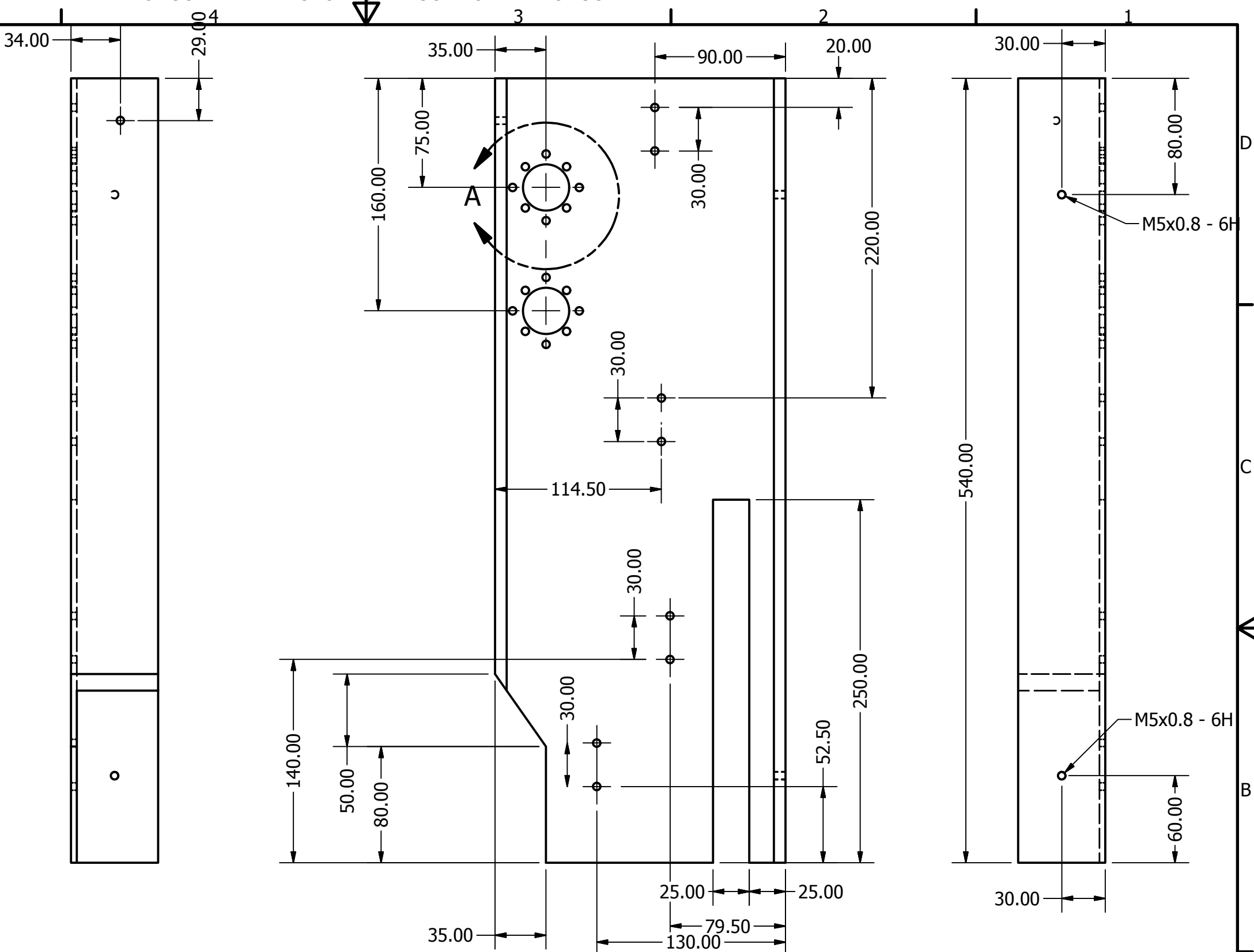
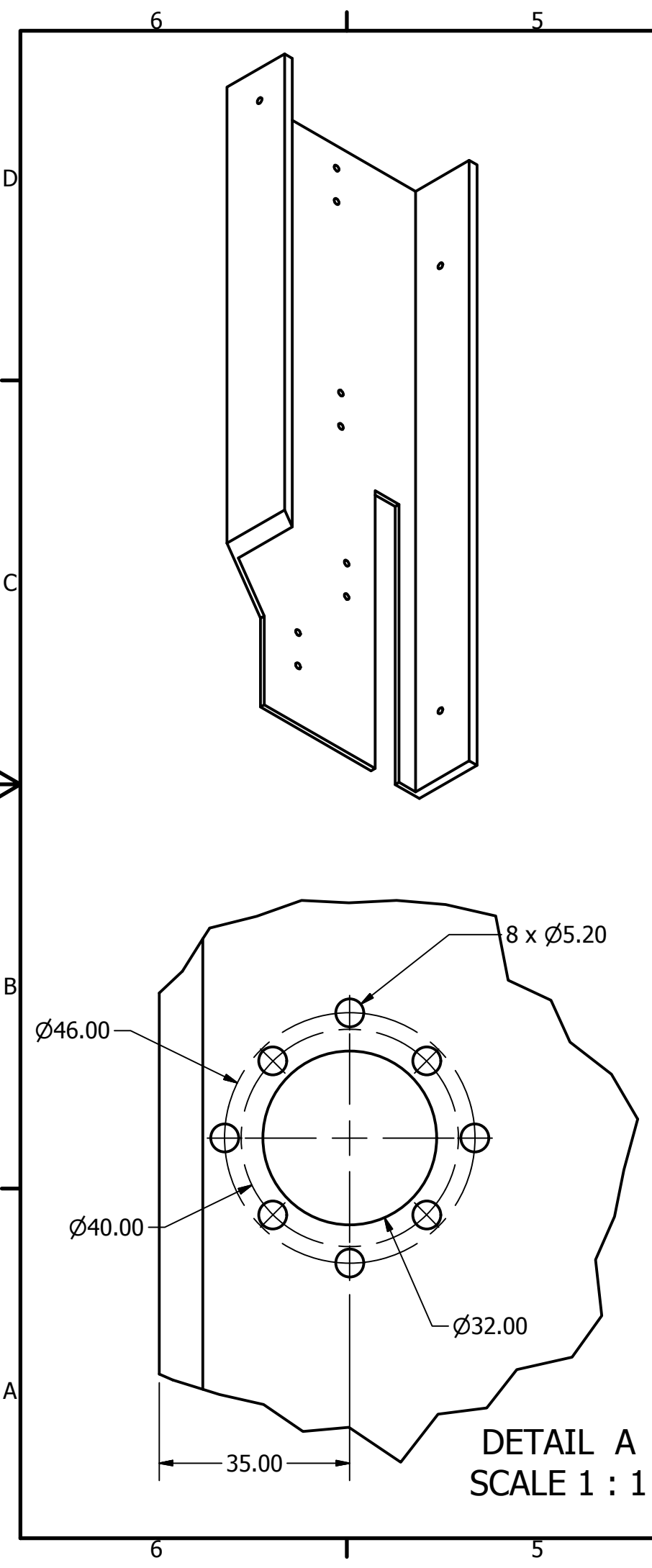


DETAIL A
SCALE 1 : 1

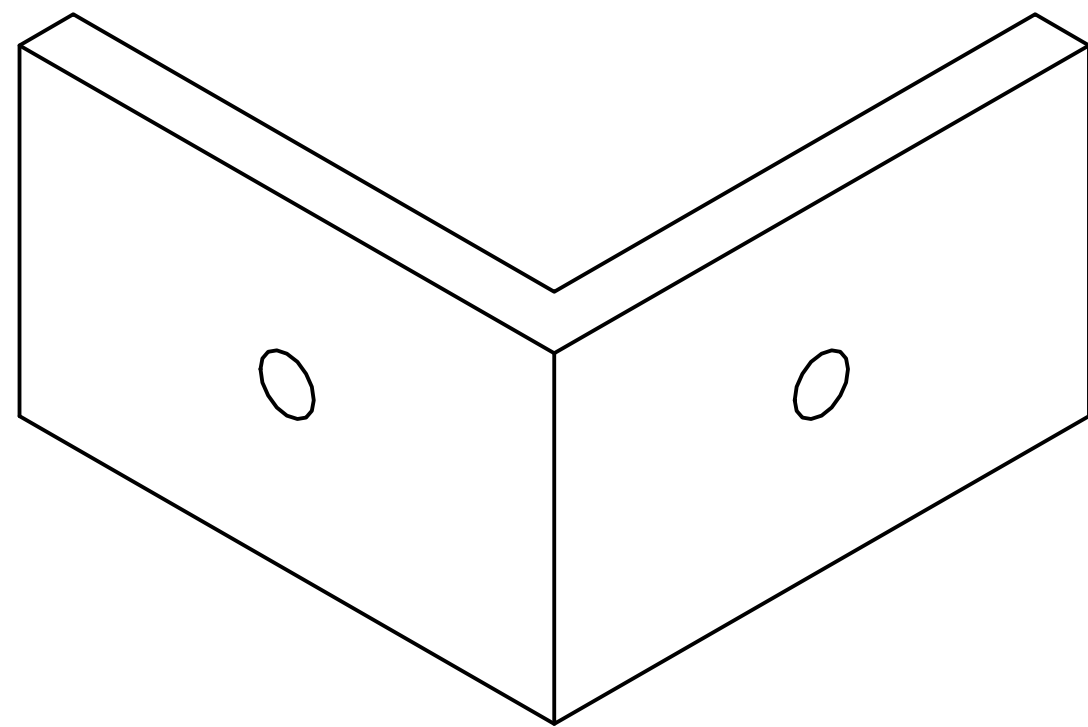
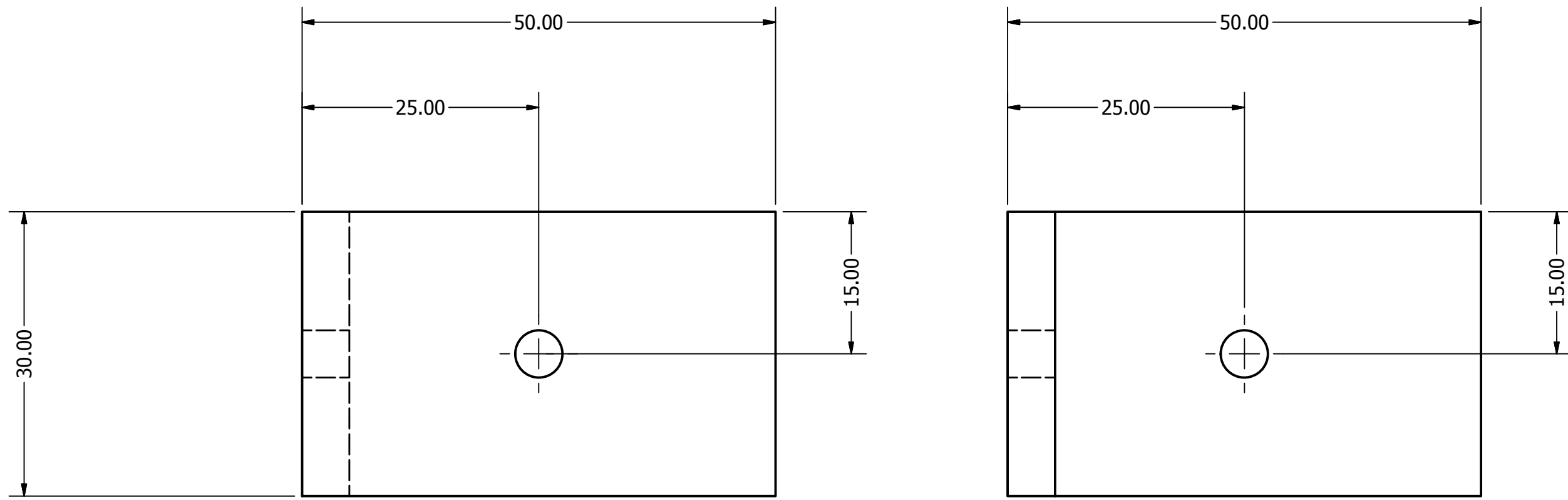
DRAWN	Petter	19.02.2010	TITLE		
CHECKED	QA				
	MFG				
	APPROVED				
	SIZE	A3	DWG NO	vange_høyre-1	REV
	SCALE	1:3	SHEET 1 OF 1		

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT



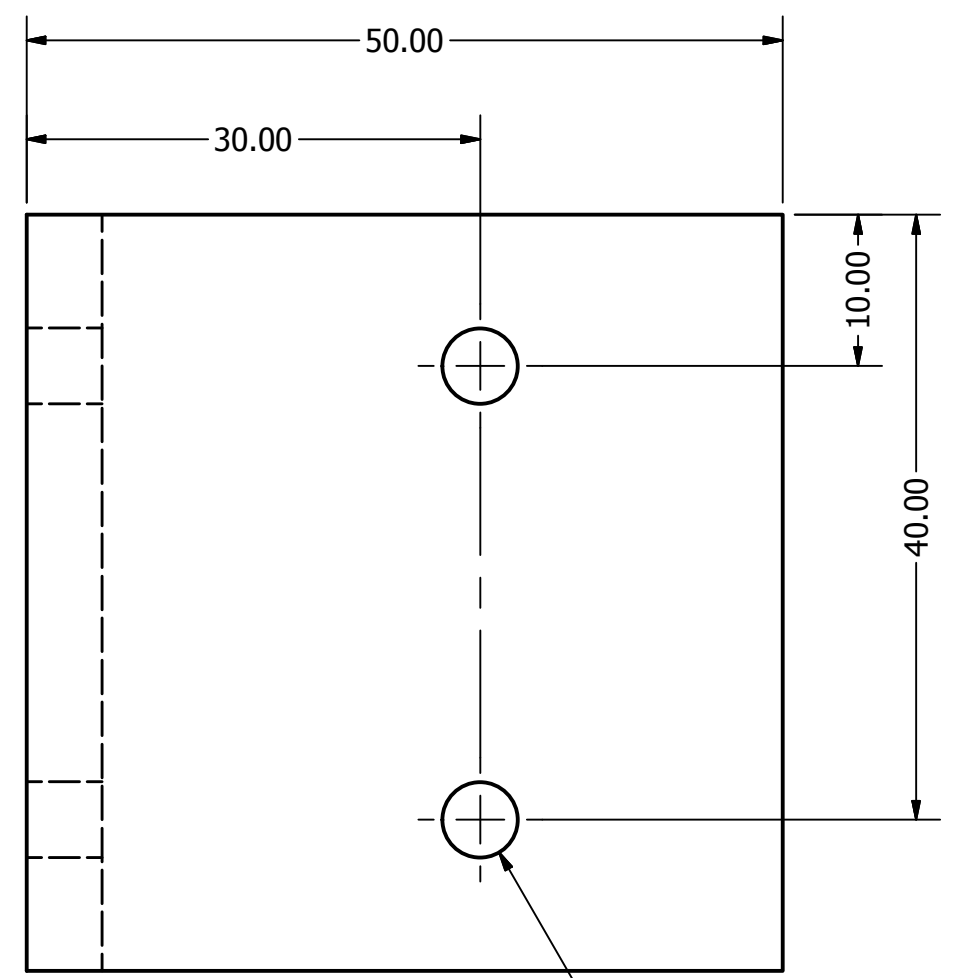
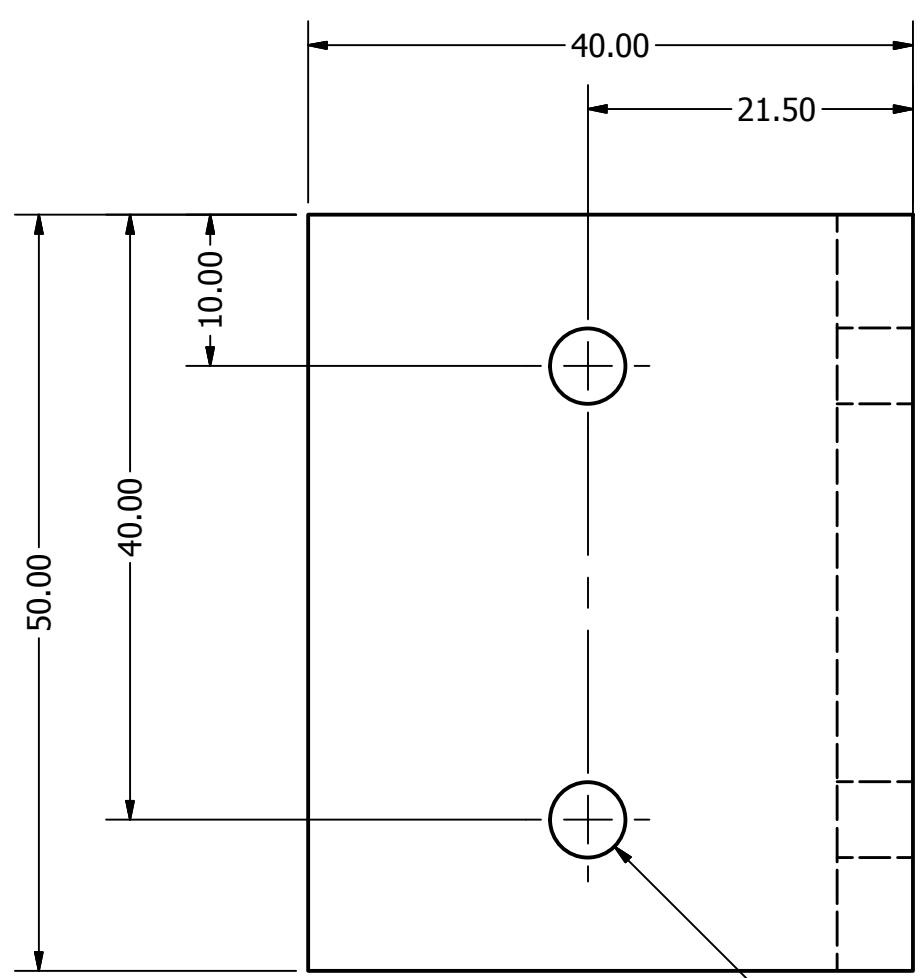
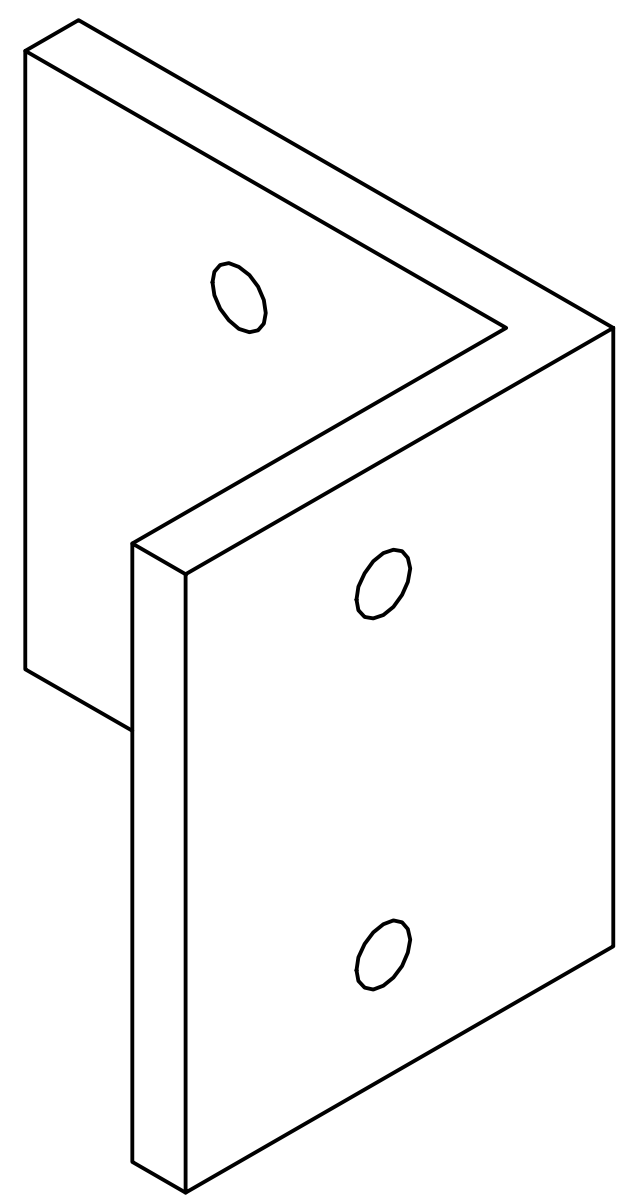
DRAWN	Petter	21.02.2010		
CHECKED			TITLE	
QA				
MFG				
APPROVED				
			SIZE	DWG NO
			A3	vange_venstre
			SCALE	REV
			1:3	
			SHEET 1 OF 1	



DRAWN	Petter	22.02.2010		
CHECKED			TITLE	
QA				
MFG				
APPROVED				
			SIZE	DWG NO
			A3	vinkel 50x50x5x30
			SCALE 2:1	REV
				SHEET 1 OF 1

PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT

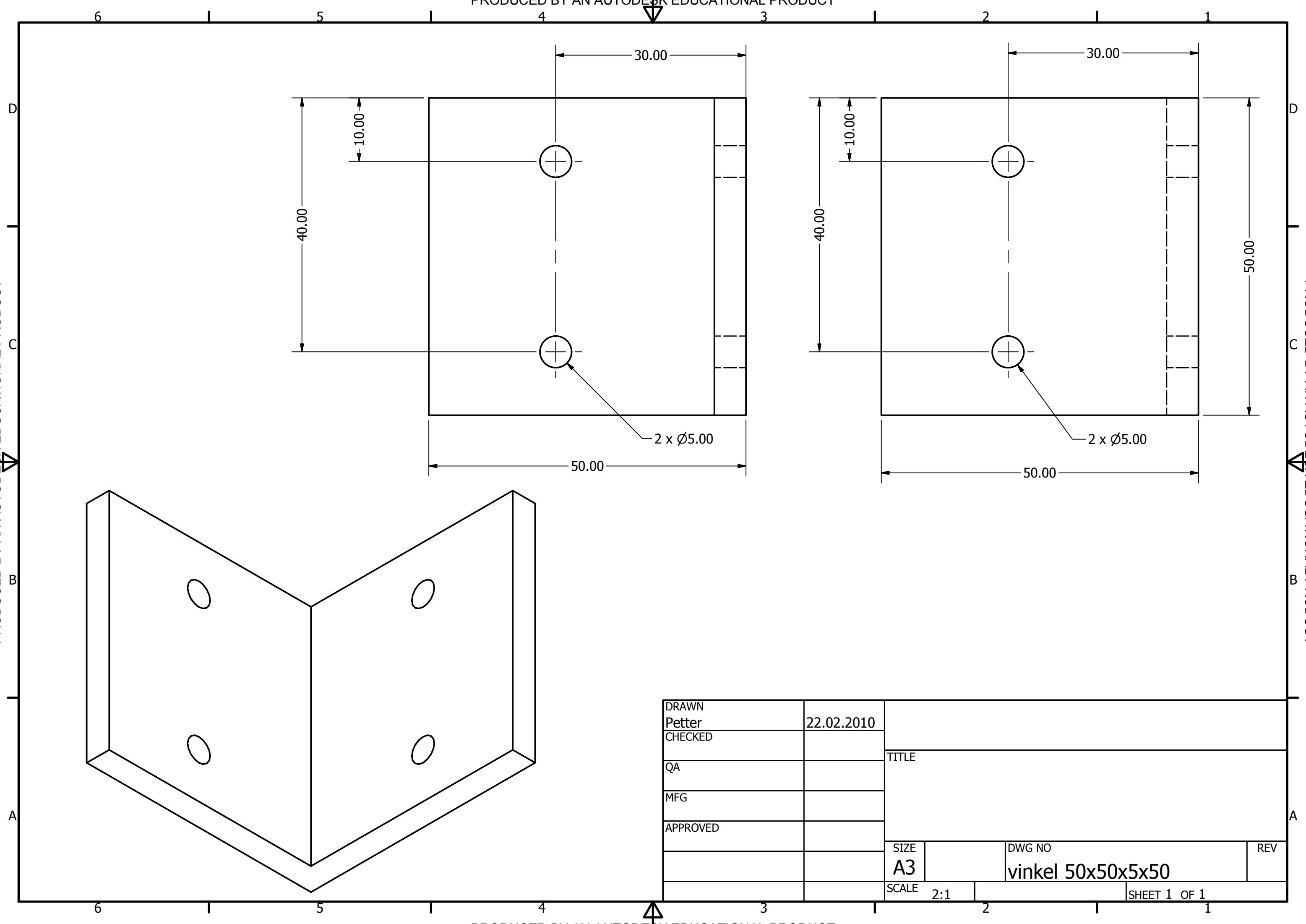
PRODUCED BY AN AUTODESK EDUCATIONAL PRODUCT



2 x Ø5.00

2 x Ø5.00

DRAWN	Petter	22.02.2010	Kuttet fra vinkel 50x50x5		
CHECKED					
QA			TITLE		
MFG					
APPROVED					
			SIZE	DWG NO	REV
			A3	vinkel 50x50x5x50 modifisert-1	
			SCALE 2:1	SHEET 1 OF 1	



DRAWN	Petter	22.02.2010		
CHECKED			TITLE	
QA				
MFG				
APPROVED				
			SIZE	DWG NO
			A3	vinkel 50x50x5x50
			SCALE	REV
			2:1	
			SHEET 1 OF 1	

Vedlegg 6

Teknisk dokumentasjon infrarød distansesensor, SHARP GP2D120

Tilgjengelig fra:

https://www1.elfa.se/data1/wwwroot/assets/datasheets/gnSharp_Distanzsensoren-GP-2D120J000F_EN.pdf

På grunn av rettighetsproblemer er ikke dette vedlegget tilgjengelig i den elektroniske versjonen av oppgaven.

Vedlegg 7

Teknisk dokumentasjon infrarød distansesensor,
SHARP GP2Y0A02YK

Tilgjengelig fra:

https://www1.elfa.se/data1/wwwroot/assets/datasheets/gpSharp_Distanzsensoren-GP-2Y0A02YK0F_EN.pdf

På grunn av rettighetsproblemer er ikke dette vedlegget tilgjengelig i den elektroniske versjonen av oppgaven.

Vedlegg 8

Script for mottak og logging av verdier fra seriellport,
skrevet i Python 2.6


```
#!/usr/bin/python
# -*- coding: cp1252 -*-
# Filename : les_seriell.py

import sys
from time import time
from Arduino_Monitor import SerialData

#####
# Program for mottak av og logging sensorverdier over seriellport
# les_seriell.py
# Petter Jemtland, 2010
#####

#PARAMETERE#

#bestem om entekstfil skal genereres eller ikke
tekstDump = True

#bestem antall verdier som skal lagres
antall = 1000

#FUNKSJONER#
def kobleTilSeriell():
    global datagen
    '''kobler til seriellporten'''
    try:
        datagen = SerialData()
    except:
        print 'Kunne ikke opprette kommunikasjon \nAvslutter'
        sys.exit(1)

def lagTekstFil(distanse):
    '''Lager en tekstfil som resultatene lagres i'''
    global filnavn
    filnavn = 'textDump-%smm.txt' %(distanse)
    fil = open(filnavn, 'w')
    print 'Tekst-fil %s klargjort' %(filnavn)
    fil.close()

def skrivTekst(streng):
    '''Skriver linje til tekstfil'''
    with open(filnavn, 'a') as fil:
        streng = streng + '\n'
        fil.write(streng)
```

```
#HOVEDPROGRAM#
if __name__=='__main__':
    print 'Program for mottak av sensorverdier over seriellport
\nProgrammet avbrytes når som helst under kjøring ved å trykke ctrl+c \n
\n'
    kobleTilSeriell()
    distanse = raw_input('Tast inn distansen i mm:')
    if tekstDump:
        lagTekstFil(distanse)
    while 1:
        try:
            print 'Venter på mikrokontroller'
            i = 0
            while i < antall:
                data = datagen.next()
                if (tekstDump & (data != -1) & (data[0] != '')):
                    tekst = "%s,%s,%s" % (time(), data[0], data[1])
                    skrivTekst(tekst)
                    print 'Sensor 1: %s Sensor 2: %s' %(data[0], data[1])
                    i = i + 1
                #else:
                #print 'Bogus data:', linje
            except:
                #avslutter kommunikasjonen dersom programmet avbrytes eller
                feil oppstår
                print 'Programmet ble avbrutt'

        finally:

            print 'Program slutt,', i, 'verdier ble logget'
            datagen.__del__()
            sys.exit(1)
            #avslutt()
            break
```

```
#!/usr/bin/python
# -*- coding: cp1252 -*-
# Filename : arduino_monitor.py

# Denne koden basert på kode skrevet av "gregpinero" som er tilgjengelig
fra:
# http://github.com/gregpinero/ArduinoPlot

"""
Listen to serial, return most recent numeric values
Lots of help from here:
http://stackoverflow.com/questions/1093598/pyserial-how-to-read-last-line-s
ent-from-serial-device
"""
from threading import Thread
import time
import serial

last_received = ''
def receiving(ser):
    global last_received
    buffer = ''
    while True:
        buffer = buffer + ser.read(ser.inWaiting())
        if '\n' in buffer:
            lines = buffer.split('\n') # Guaranteed to have at least 2
entries
            last_received = lines[-2]
            #If the Arduino sends lots of empty lines, you'll lose the
            #last filled line, so you could make the above statement
conditional
            #like so: if lines[-2]: last_received = lines[-2]
            buffer = lines[-1]

class SerialData(object):
    def __init__(self, init=50):
        try:
            self.ser = serial.Serial(
                port='COM6',
                baudrate=38400,
                bytesize=serial.EIGHTBITS,
                parity=serial.PARITY_NONE,
                stopbits=serial.STOPBITS_ONE,
                timeout=0.1,
                xonxoff=0,
                rtscts=0,
                interCharTimeout=None
            )
            print "Serial connection initiated"
```

```
        try:
            self.ser.flushInput()
            self.ser.flushOutput()
        except:
            print ' Did not flush'

    except serial.serialutil.SerialException:
        print "No serial connection"
        #no serial connection
        self.ser = None
    else:
        self.traad = Thread(target=receiving, args=(self.ser,))
        self.traad.start()
        print self.traad

def next(self):
    if not self.ser:
        return -1 #return anything so we can test when Arduino isn't
connected
    #return a float value or try a few times until we get one
    for i in range(40):
        raw_line = last_received
        try:
            #todo: make array of sensor values
            return raw_line.strip().split(',')
            #return raw_line.strip()
        except ValueError:
            print 'bogus data',raw_line
            return -1
            time.sleep(.005)
    return 0.

def __del__(self):
    if self.ser:
        print 'Killing'
        self.traad.killed = True
        print "Closing serial"
        self.ser.close()
    else:
        print "No serial"

if __name__=='__main__':
    s = SerialData()
    for i in range(500):
        time.sleep(.015)
        print s.next()
```

Vedlegg 9

Script for avlesning av sensor med mikrokontroller,
skrevet i Wiring

```

#define FASTADC 1

#ifndef cbi
#define cbi(sfr,bit) (_SFR_BYTE(sfr) &= ~_BV(bit))
#endif
#ifndef sbi
#define sbi(sfr,bit) (_SFR_BYTE(sfr) |= _BV(bit))
#endif

void setup() {
  //Definer ekstern strømkilde som referansespenning for ADC
  analogReference(EXTERNAL);

#if FASTADC
  // sett "prescaler" for ADC til 16
  sbi(ADCSRA,ADPS2) ;
  sbi(ADCSRA,ADPS1) ;
  cbi(ADCSRA,ADPS0) ;
#endif

  //start seriell kommunikasjon med PC
  Serial.begin(38400) ;
  Serial.flush();
}

void loop() {
  // Gjør målinger og send resultater separert med komma via seriellport
  Serial.print(analogRead(0)) ;
  Serial.print(',');
  Serial.println(analogRead(1));
}

```

Vedlegg 10

Eksempler fra utvidet versjon av biblioteket libEPOS

I hovedsak skrevet av Olivier Roulet-Dubonnet.

Delen av biblioteket som er skrevet i C finnes på vedlagte CD-ROM

```
#!/usr/bin/python

import libepos

class EPOS(object):
    """
    objectify libepos api
    all methods not explicitly defined in the class get
    called from libepos with epos pointer as arg !!
    """
    def __init__(self, device):
        self._device = device
        self._epos = None

    def openEPOS(self):
        self._epos = libepos.openEPOS(self._device)
        if self._epos:
            return True
        else:
            print "Error opening connection to EPOS device"
            return False

    def closeEPOS(self):
        if self._epos:
            libepos.closeEPOS(self._epos)
            self._epos = None

    def printEPOScontrolword(self, word):
        """ reimplement to avoid __getattr__ , thus epos arg addition"""
        libepos.printEPOScontrolword(word)

    def __getattr__(self, name):
        if self._epos:
            attr = libepos.__getattr__(name)
            if hasattr(attr, '__call__'):
                def _catchall(*args, **kwargs):
                    #first add _epos arg as needed in c lib
                    args = list(args)
                    args.insert(0, self._epos)
                    # now method from libepos
                    ret = attr(*args, **kwargs)
                    #now pythonify return values
                    if type(ret) == int :
                        if ret == 0:
                            return True
                        else:
                            return False
                    elif type(ret) in (list, tuple):
                        code = ret[0]
                        if code != 0:
                            raise Exception(ret)
                    else:

```



```
        ret = ret[1:]
        if len(ret) < 2:
            return ret[0]
        else:
            return ret[1:]
    return _catchall
else:
    return attr
else:
    print "EPOS not initialized"

if __name__ == "__main__":
    e = EPOS("/dev/ttyS0")
    e.openEPOS()
    e.closeEPOS()
```

```
#!/usr/bin/env python

from time import sleep, time
import epos
import test

if __name__ == "__main__":
    print "opening connection left"
    left = epos.EPOS("/dev/ttyUSB0")
    left.openEPOS()
    print "opening connection right"
    right = epos.EPOS("/dev/ttyUSB1")
    right.openEPOS()
    try:
        print "init"
        test.init(left)
        test.init(right)
        print "config"
        test.config(left)
        test.config(right)
        print "start move"

        left.moveWithVelocity(4000)
        right.moveWithVelocity(-1000)
        print "sleeping"
        sleep(1)
        print time()
        left.setTargetVelocity(4000)
        right.setTargetVelocity(2000)
        print time()
        left.setTargetVelocity(1000)
        right.setTargetVelocity(100)
        print time()
        sleep(3)
    finally:
        print "cleanup"
        try:
            left.haltVelocityMovement()
        finally:
            left.closeEPOS()
        try:
            right.haltVelocityMovement()
        finally:
            right.closeEPOS()
```

Vedlegg 11

Script for å beregne avstand fra avlest sensorverdi med interpolasjon,
skrevet i Python 2.6

```
#!/usr/bin/python
# -*- coding: cp1252 -*-
# Filename : interpolering.py

class DistanceTable(object):

    """Provides interpolated distance values from infrared sensor values.
    Point(distance, sensorValue)"""

    def __init__(self, points):
        self.points = sorted(points) #lowest distance as first point in array
        print self.points

    def __getitem__(self, x):
        if x > self.points[0][1] or x < self.points[-1][1]:
            raise ValueError
        below, above = self._GetAboveBelow(x)
        return self._Interpolate(x, below, above)

    def _GetAboveBelow(self, x):
        """Get the table point above and below the given x"""
        below = None
        above = self.points[0]
        for point in self.points[1:]:
            below = above
            above = point
            if x <= above[1]:
                break
        return below, above

    def _Interpolate(self, x, below, above):
        """Interpolate a distance value for the given x"""
        slope = (float(above[0] - below[0]) /
                (above[1] - below[1]))
        return below[0] + (slope * (x - below[1]))

#Usage example
points = ((40, 572.3), (50, 480.0), (75, 329.3), (100, 252.0))
table = DistanceTable(points)
print table[300]
```

Vedlegg 12

Sammenlikning av tekniske spesifikasjoner for ulike sensorer

Vedlegg 12 – Sammenlikning av tekniske spesifikasjoner for ulike sensorer

Sensor	Produsent	Måleområde [mm]		Oppløsning	Nøyaktighet
		Minimum	Maksimum		
Områdesensor	SICK	300/500 ¹	20000	0,25°	Systematisk feil: typ ±30mm max. ±40mm Standard feil (1σ): typ. 12mm max. 20mm
Induktiv sensor	FESTO	0	20	0,005mm	Repeterbarhet: 0,3mm
Ultrasonisk sensor	SICK	30	350	0,36mm	Repeterbarhet: Typ. +-0.15 % of final value Presisjon: =< 2% of final value
Infrarød sensor	SHARP	40	300	1,9 v	Ikke oppgitt
Laser-sensor	SICK	60	180	30 μm	Repeterbarhet: 90 μm Presisjon: ± 300 μm

Presisjon – Accuracy

Repeterbarhet – repeatability / repeat accuracy

¹ Oppgis forskjellig på to ulike steder i dokumentasjonen.

Vedlegg 13

General Application Note: Distance Measuring Devices

Tilgjengelig fra:

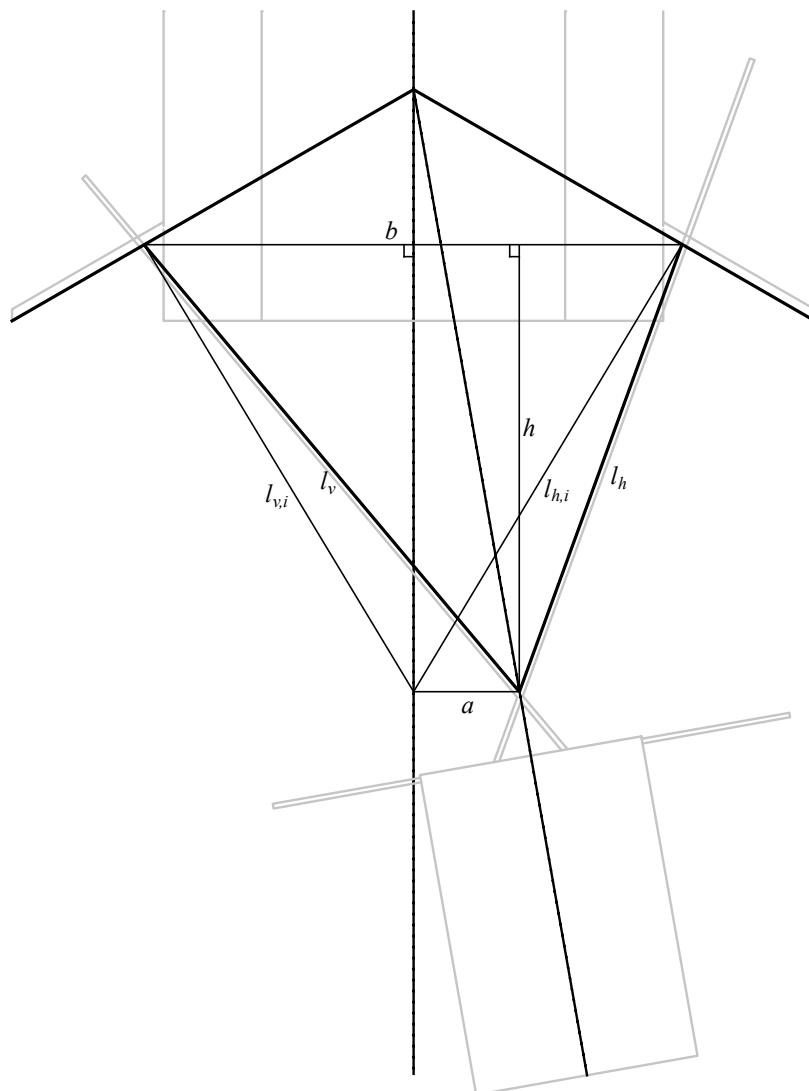
<http://www.seattlerobotics.org/WorkShopRobot/level2/datasheets/Sharp-DistanceSensorAppNote.pdf>

På grunn av rettighetsproblemer er ikke dette vedlegget tilgjengelig i den elektroniske versjonen av oppgaven.

Vedlegg 14

Mellomregning for posisjon

Vedlegg 14 – Mellomregning



Figur 1 – Hjelpesfigur 1

Vinkelen mellom linjene l_v og l_h er 60° .

Benytter cosinussetningen som sier oss at:

$$b^2 = l_v^2 + l_h^2 + 2l_v l_h \cos(60^\circ) \quad (1)$$

$$b = \sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)} \quad (2)$$

Vinkelen v_v er vinkelen mellom linja b og linja l_v , og vinkelen v_h er vinkelen mellom linja b og linja l_h .

$$\frac{l_h}{b} = \frac{\sin(v_v)}{\sin(60^\circ)} \quad (3)$$

$$\frac{l_v}{b} = \frac{\sin(v_h)}{\sin(60^\circ)}$$

$$\sin(v_v) = \frac{l_h}{b} \sin(60^\circ) \quad (4)$$

$$\sin(v_h) = \frac{l_v}{b} \sin(60^\circ)$$

Høyden, h , kan uttrykkes som:

$$h = l_h \sin(v_h) = l_v \sin(v_v) \quad (5)$$

Setter inn (4):

$$h = \frac{l_v l_h}{b} \sin(60^\circ) = \frac{l_v l_h}{b} \sin(60^\circ) \quad (6)$$

Setter inn (2) for b :

$$h = \frac{l_h l_v \sin(60^\circ)}{\sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)}} \quad (7)$$

Vi vet også at linja l_{vi} kan uttrykkes som:

$$l_{vi} = \sqrt{\left(\frac{b}{2}\right)^2 + h^2} \quad (8)$$

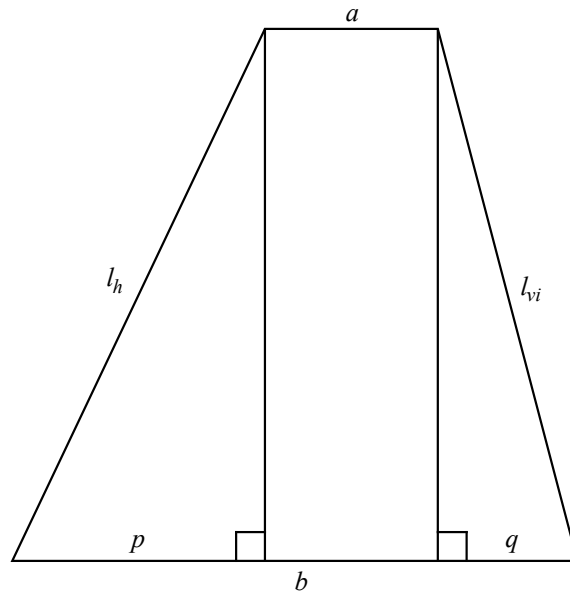
Vi vet nå tre av sidene i trapeset med sidekanter l_h , l_{vi} , b og a som vist i Figur 2. Siden a er gitt ved:

$$l_h^2 = p^2 - h^2 \quad (9)$$

$$l_{vi}^2 = q^2 - h^2$$

$$p = \sqrt{l_h^2 + h^2} \quad (10)$$

$$q = \sqrt{l_{vi}^2 + h^2}$$



Figur 2 - Hjelpefigur trapes

Som vi kan se fra figuren er:

$$a = b - p - q \quad (11)$$

Setter inn (10) og får:

$$a = b - \left(\sqrt{l_h^2 - h^2} + \sqrt{l_{vi}^2 - h^2} \right) \quad (12)$$

Som innsatt b fra likning (2) gir:

$$a = \sqrt{l_v^2 + l_h^2 - 2l_v l_h \cos(60^\circ)} - \left(\sqrt{l_h^2 - h^2} + \sqrt{l_{vi}^2 - h^2} \right) \quad (13)$$

Vedlegg 15

Forstudierapport

Forstudierappport

Masteroppgave:

Dokkingstsyem for automatisk
selvstyrt vogn (AGV)

Petter Jemtland

Vår 2010

Institutt for produksjons- og kvalitetsteknikk
NTNU

Forstudierapport

Forord

Dette er en forstudierapport til masteroppgaven med tittelen "Dokkingssystem for automatisk selvgående vogn (AGV)" som en del av min mastergrad innen Produktutvikling og Produksjon ved NTNU. Prosjektoppgaven skrives ved Institutt for Produksjons- og kvalitetsteknikk fagområde Produksjonsteknikk. Forstudierapporten inneholder bakgrunnen bak oppgaven samt presiseringer av de ulike deloppgavene. I tillegg inneholder den en plan for arbeidet med oppgaven.

Innhold

Forord.....	1
1 Innledning	2
2 Involverte i prosjektet.....	2
2.1 Petter Jemtland.....	2
2.2 Terje K. Lien.....	2
2.3 Olivier Roulet-Dubonnet	2
3 Presisering av oppgavene og arbeidsmetodikk	2
3.1 Deloppgave 1	3
3.2 Deloppgave 2	3
3.3 Deloppgave 3	3
3.4 Deloppgave 4	4
4 Mål	4
5 Tidsplan og omfang.....	4
5.1 Milepæler	4
5.2 WBS (Work Breakdown Structure)	5
Vedlegg 1 - POS ("Project Overview Statement")	6
Vedlegg 2 – Tidsplan, Gantt-diagram.....	7
Vedlegg 3 - KTR-ark	8

1 Innledning

AGV står for "automated guided vehicle", eller "automatisk, selvgående vogn" på norsk. AGV'er benyttes i dag hovedsakelig som lastbærere innen flere ulike næringer. De benyttes i havnearbeid, på sykehus, i varehus og i ulike typer produksjonsanlegg. Gjennom å bruke AGV'er kan man automatisere vareflyten og på den måten spare kostnader samt øke effektiviteten og nøyaktigheten, og redusere skader og feil på produkter som kan oppstå gjennom logistikksystemet. De fleste AGV'er benytter en styresløyfe (oftest optisk eller elektromagnetisk) som er lagt på eller i underlaget som AGV'en styres etter. Per i dag representerer et AGV-system en betydelig investering, men samtidig kan investeringskostnaden være lavere enn et dedikert materialhåndteringssystem som ikke har samme fleksibiliteten man har med et AGV-system.

Som en del av et PhD-studium utvikles det ved Institutt for produksjon og kvalitetsteknikk en lavkost AGV. Denne AGV'en styres ved hjelp av en rekke kameraer montert i taket som lokaliserer AGV'en, og ut fra dette kan bestemmes hvilken retning den beveger seg. Styresignalene sendes til AGV'en via trådløst nettverk. Ut fra at AGV'ene ikke styres etter faste baner kan de karakteriseres som såkalte "free-ranging" eller "free-roaming" AGV'er.

Styremetoden som benytter kameraer har imidlertid en begrenset nøyaktighet som i en del tilfeller kan være utilstrekkelig, og da særlig i forhold til dokking. Derfor er det ønskelig med en løsning som gir AGV'en muligheten til nøyaktig posisjonering inn mot et punkt, og dette vil være hovedoppgaven i dette prosjektet.

2 Involverte i prosjektet

2.1 Petter Jemtland

Stud. techn. ved NTNU. Går femte året ved studiet Produktutvikling og Produksjon, og er tilknyttet Institutt for produksjons- og kvalitetsteknikk med fordypningsretning produksjonssystemer.

2.2 Terje K. Lien

Professor ved Institutt for produksjons- og kvalitetsteknikk innenfor fagområdet produksjonssystemer. Er veileder for oppgaven.

2.3 Olivier Roulet-Dubonnet

PhD stipendiat ved Institutt for produksjons- og kvalitetsteknikk. Har vært med på utviklingen av de eksisterende prototypene av AGV'en, og jobber fortsatt med prosjektet.

3 Presisering av oppgavene og arbeidsmetodikk

I tillegg til de oppgavene som er presisert i den utdelte oppgaveteksten vil den første oppgaven i masteroppgaven være å bygge en ny AGV. Denne AGV'en skal benytte det samme styringssystemet som har blitt utviklet i prosjektet, men hardware-delen skal utvikles og bygges. Bakgrunnen er at de AGV'ene som har blitt bygget til nå i prosjektet har vært bygget på en eksisterende elektrisk scooter. Det er dog mer ønskelig å bygge AGV'en ved å bruke mer standard komponenter, noe som vil bedre byggekvaliteten, og lette arbeidet ved å produsere ytterligere AGV'er. Ved å bruke standard komponenter og et relativt enkelt design vil man også forenkle vedlikehold og modifikasjoner av AGV'en.

3.1 Deloppgave 1

En analyse av problemstillingen ved presis posisjonering (dokking) av AGV'er ved laste- og lagerplasser, verktøymaskiner og montasjeceller. Spesielt skal det analyseres hvilke spesifikke problemstillinger som er knyttet til presis dokking av instituttets eksperimentelle AGV'er.

Denne oppgaven består først i å finne bakgrunnen bak behovene for presis posisjonering, i hvilke situasjoner er presis posisjonering nødvendig, og hvor viktig er nøyaktig posisjon i de ulike tilfellene. En naturlig fremgangsplan vil være å først finne ut hvilke bruksområder AGV'en er tiltenkt, for deretter finne ut hvilke krav som stilles til nøyaktighet innenfor hvert av disse bruksområdene.

På grunn av det utradisjonelle styringssystemet med bruk av kameraer vil instituttets AGV ha andre utfordringer enn tradisjonelle styreløsninger som for eksempel bruker en styresløyfe i underlaget. Disse problemstillingene må også avdekkes, og er dermed en del av oppgaven.

3.2 Deloppgave 2

En litteraturstudie for å skaffe oversikt over metodene for styringa av AGV'er og en dypere studie av metoder og teknikker for måling av posisjon og styring av bevegelse inn mot presis målposisjon for AGV.

Litteraturstudiet brukes i utgangspunktet for å skaffe en oversikt over emnet styring, og skal dermed ikke være veldig dyptgående på dette området. Med styring menes styring av den enkelte AGV, det vil si styring til ønsket posisjon (banestyring). Delen av styringen som omfatter trafikk kontroll, køorganisering, ekspedisjon (dispatching) av AGV'er og liknende temaer innen overordnede styring av et system med AGV'er skal ikke behandles.

Delen som omfatter måling og styring av bevegelse inn mot posisjon er desto viktigere, og vil utgjøre grunnlaget for utviklingen av et dokkingsystem i deloppgave 3. Det vil være naturlig å undersøke hvilke metoder for måling av posisjon og styring av bevegelse som enten brukes i dag eller som finnes på konseptstadiet for å få et godt grunnlag til utviklingsoppgaven (deloppgave 3). Det vil være naturlig med en viss grad av overlapping mellom deloppgave 2 og 3. Evaluering av de ulike konseptene som blir funnet i løpet av litteraturstudiet vil foregå sammen med utviklingen i deloppgave 3.

Det vil også være nyttig å få noe input fra Olivier i denne fasen siden han har jobbet med dette på instituttets AGV. Dette ville i så fall redusere tidsbruken noe og samtidig sannsynligvis øke kvaliteten på resultatet.

3.3 Deloppgave 3

Utvikling av forslag til teknisk løsning for et dokkingsystem for instituttets AGV'er. Systemet skal primært anvendes for opp/avlastingsstasjoner for AGV og for dokking i montasjeceller.

Her vil det være naturlig å bruke den opparbeidede kunnskapen om ulike metoder for kjøring mot posisjon fra deloppgave 2 og analysere og evaluere de ulike løsningene for eventuell bruk i dokkingsystemet. I tillegg til å evaluere de allerede eksisterende løsningene, vil det være naturlig å gjøre en undersøkelse av om det er andre, hittil ubenyttede prinsipper man kan benytte i dokkingsystemet. Dette kan for eksempel gjøres med utgangspunkt i en såkalt morfologisk matrise

der problemet først dekomponeres i små deler før alle tenkelige fysiske prisnipp for å løse disse evalueres.

Etter å ha valgt en teknisk løsning vil det kanskje være nødvendig med testing og/eller verifisering for å bekrefte at løsningen er teknisk gjennomførbar. Etter eventuell testing/verifisering vil neste trinn i prosessen være å utforme en komplett dokkingløsning der det valgte prinsippet implementeres. Dette innebærer konstruksjon av teknisk løsning, samt implementering med AGV'en styresystem.

Viktige ting å ta hensyn til i utviklingen er at dokkingssystemet skal være løsningens pålitelighet, produksjonskostnad og installasjonskostnad. Det dermed ønskelig med en løsning som er robust, relativt rimelig å produsere og samtidig enkel å installere.

3.4 Deloppgave 4

Så langt som det er mulig innenfor oppgavens tidsramme, skal det gjennomføres oppbygging og testing av elementer i dokkingssystemet. Beslutninger om hvilke elementer som skal bygges og testes tas i samråd med veileder.

Denne oppgaven skal som indikert gjennomføres dersom det blir tid til det. Dette vil nødvendigvis avhenge av tidsbruken på de foregående oppgavene. I tidsplanen som blir lagt til grunn for arbeidet med masteroppgaven er det satt tatt høyde for at denne oppgaven skal gjennomføres. En utfordring kan komme til å være å skaffe til veie nødvendige komponenter i tide for å få gjennomført byggingen før prosjektet avsluttes.

4 Mål

- Ferdigstillelse av oppdatert AGV
- Ferdigstillelse av løsning for dokking av AGV
- Ferdigstillelse av prototyp av dokkingløsning
- Alle tidsfrister overholdt

5 Tidsplan og omfang

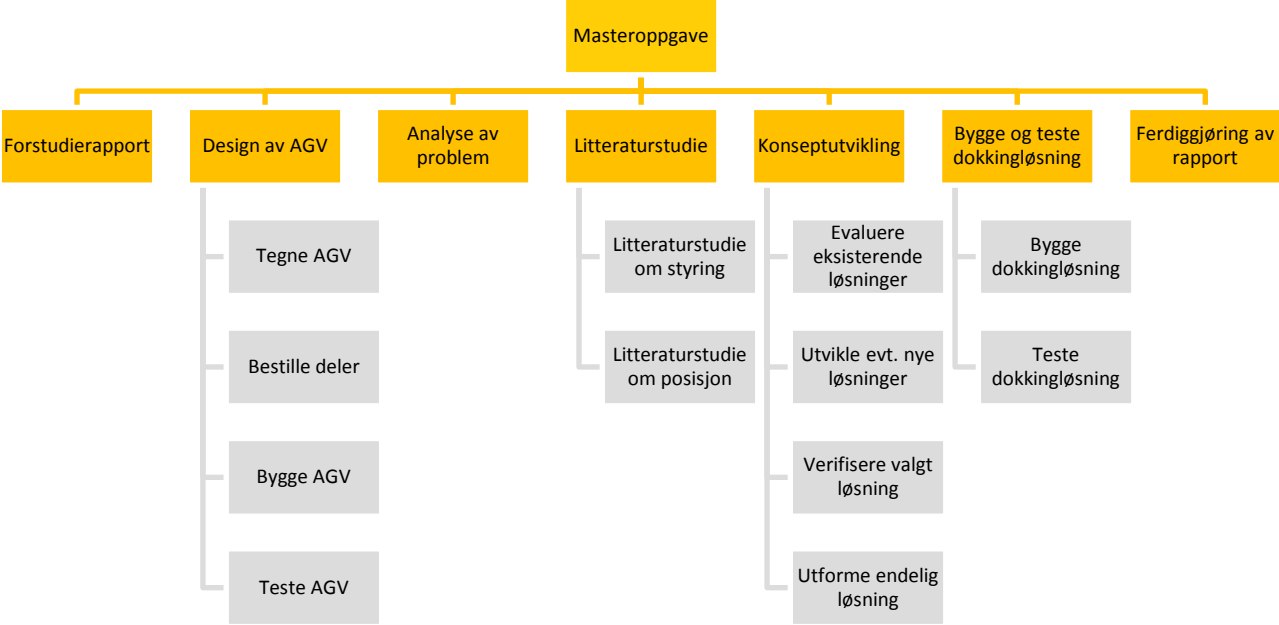
Tidsplanen som legges til grunn for prosjektet finnes i et Gantt-diagram i Vedlegg 2. Ett semester er normert til 48 belastningstimer per uke, og det er dette som ligger til grunn for tidsplanen.

5.1 Milepæler

Dato	Beskrivelse	Type
5. februar	Levere forstudierapport	Innlevering
7. februar	"Produksjonsunderlag" for oppdatert AGV ferdig	Ferdigstillelse
14. mars*	AGV ferdig	Ferdigstillelse
25. februar	Analyse av problem ferdig	Ferdigstillelse
14. mars	Litteraturstudie ferdig	Ferdigstillelse
30. april	Utvikling av dokkingkonsept ferdig	Ferdigstillelse
3. juni	Bygging og testing av dokkingkonsept ferdig	Ferdigstillelse
14. juni	Innlevering av masteroppgave	Innlevering

* Denne datoen er noe usikker siden den i stor grad avhenger av levering og produksjon av delene. Byggingen bør imidlertid være ferdig innen konseptutviklingen er ferdig.

5.2 WBS (Work Breakdown Structure)



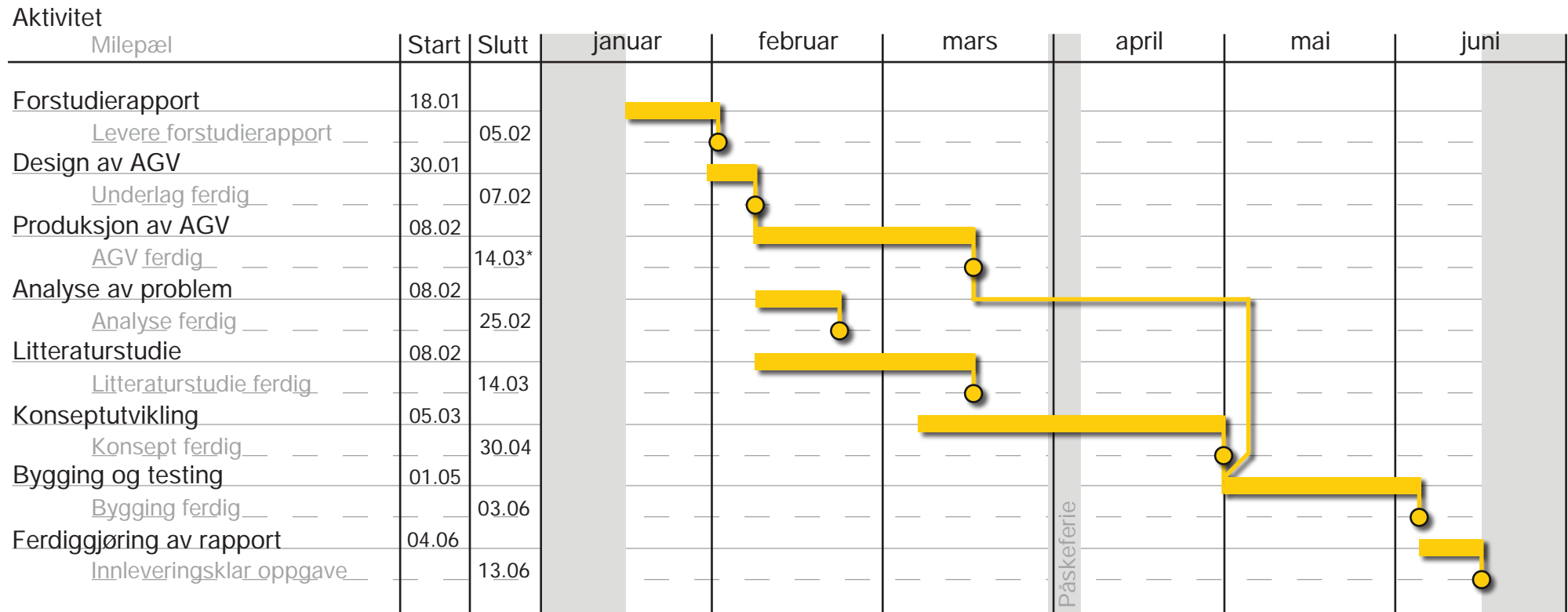
Vedlegg 1 - POS ("Project Overview Statement")

POS	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	Dato: 7.september 2009
Problem: Instituttets AGV styres ved hjelp av kameraer i taket. Denne styringen er imidlertid ikke nøyaktig nok for å brukes til dokking. Derfor må et eget system utvikles for at AGV'en skal kunne oppnå en nøyaktig posisjon.		
Hovedmål: Å utvikle, bygge og teste en dokkingløsning for å sikre nøyaktig posisjonering av instituttets AGV. Arbeidet skal resultere i et ferdig dokkingløsning samt en rapport omkring temaet og arbeidet.		
Delemål: <ul style="list-style-type: none">- Designe og bygge oppdatert versjon av AGV- Gjennomføre problemanalyse og litteraturstudie omkring emnene styring og nøyaktig posisjonering av AGV'er.- Utvikle fungerende dokkingssystem til instituttets AGV		
Suksesskriterier: <ul style="list-style-type: none">- Karakter B eller bedre- Gjennomført prosjekt innenfor gitte tidsfrister		
Forutsetninger, risiko, hindringer: <ul style="list-style-type: none">- Tidsbegrenset av egen tidsplan- Kontinuerlig veiledning gjennom prosjektet- Tilgang til nødvendige komponenter i løpet av prosjektet		

Tidsplan for masteroppgave:
Dokkingssystem for automatisk selvstyrt vogn (AGV)



Vår 2010



* Datoen er usikker og vil avhenge av levering og produksjon av komponenter

Vedlegg 3 - KTR-ark

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP1	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Forstudierapport		
Aktiviteter: <ul style="list-style-type: none">- Presisere og avgrense masteroppgavens deloppgaver- Lage tidsplan for arbeidet med masteroppgaven		
Beskrivelse av arbeidsomfang: Forstudierapporten skal inneholde en presisering av oppgaven, nedbrytning av arbeidsoppgavene samt en tidsplan.		
Begrensninger: Avsatt tid for oppgaven		
Resultat: En ferdig forstudierapport som skal leveres veileder		
Planlagt start: 18. januar 2010	Kostnadsestimat: 0 NOK	
Planlagt slutt: 5. februar 2010	Ressursbehov: 17 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP2	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Design av AGV		
Aktiviteter: <ul style="list-style-type: none"> - Finne komponenter som skal benyttes - Tegne arbeidstegninger av deler til AGV 		
Beskrivelse av arbeidsomfang: En oppdatert versjon av instituttets AGV skal utvikles. Dette er en relativt enkel konstruksjon og skal ikke ta for mye tid og designe. Det skal være fokus på å benytte standard komponenter i størst mulig grad.		
Begrensninger: <ul style="list-style-type: none"> - Avsatt tid for oppgaven - Motorer som skal benyttes er allerede valgt - Standard komponenter skal benyttes i størst mulig grad 		
Resultat: Ferdig underlag for å bestille komponenter og produsere AGV'en skal foreligge.		
Planlagt start: 30. januar 2010	Kostnadsestimat: 0 NOK	
Planlagt slutt: 7. februar 2010	Ressursbehov: 9 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP3	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Produksjon av AGV		
Aktiviteter: <ul style="list-style-type: none"> - Bestilling av deler - Montering av deler - Igangsetting og testing 		
Beskrivelse av arbeidsomfang: Den designede AGV skal bygges. Oppgaven vil nødvendigvis være avhengig av levering og produksjon av komponenter.		
Begrensninger:		
Resultat: Ferdig montert og fungerende AGV.		
Planlagt start: 8. februar 2010	Kostnadsestimat: Komponentkostnad	
Planlagt slutt: 14. mars 2010 (muligens senere)	Ressursbehov: Usikkert	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP4	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Analyse av problemstilling		
Aktiviteter: <ul style="list-style-type: none"> - Intervjuer - Litteraturstudie 		
Beskrivelse av arbeidsomfang: Problemstillingen skal analyseres i henhold til deloppgave 1		
Begrensninger: Avsatt tid for oppgaven		
Resultat: Kapittel omkring behovene for presis posisjonering, samt nødvendige forutsetninger for utviklingsoppgaven.		
Planlagt start: 8. februar 2010	Kostnadsestimat: Deler og produksjon	
Planlagt slutt: 25. februar 2010	Ressursbehov: 18 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP5	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Litteraturstudie		
Aktiviteter: <ul style="list-style-type: none"> - Finne litteratur - Lese litteratur - Skrive om lest litteratur 		
Beskrivelse av arbeidsomfang: Litteraturstudie for å få oversikt over styresystemer samt å få innsikt i styring av AGV inn mot presist punkt. Det vil være en naturlig glidende overgang mellom AP5 og AP6.		
Begrensninger: Avsatt tid for oppgaven		
Resultat: Eget kapittel		
Planlagt start: 8. februar 2010	Kostnadsestimat: 0 NOK	
Planlagt slutt: 14. mars 2010	Ressursbehov: 29 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP6	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Konseptutvikling		
Aktiviteter: <ul style="list-style-type: none"> - Evaluering av metoder funnet i litteraturstudiet (en slags videreføring av litteraturstudiet) - Analyse av ytterligere prinsipper - Valg av konsept og verifisering av dette - Utvikling av komplett dokkingløsning 		
Beskrivelse av arbeidsomfang: Dette er den mest omfattende oppgaven i prosjektet. Den innebærer å utvikle et dokkingssystem for AGV'en. AP6 vil i en viss grad være en videreføring av litteraturstudiet, men med en større grad av evaluering av de ulike metodene i forhold til den spesifikke problemstillingen.		
Begrensninger: Avsatt tid for oppgaven. 5 - 10 dager vil gå bort i forbindelse med gjennomføring av eksamen.		
Resultat: Ferdig konsept for dokkingløsning for instituttets AGV, samt dokumentasjon for denne.		
Planlagt start: 5. mars 2010	Kostnadsestimat: 0 NOK	
Planlagt slutt: 30. april 2010	Ressursbehov: 57 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP7	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Bygging og testing av dokkingssystem		
Aktiviteter: <ul style="list-style-type: none"> - Bygge egenutviklet dokkingløsning, eller deler av denne. - Implementere og teste dokkingløsning 		
Beskrivelse av arbeidsomfang: Dersom tiden strekker til skal dokkingløsningen, eller deler av denne, bygges og testes i laboratoriet.		
Begrensninger: Avsatt tid for oppgaven		
Resultat: Ferdig implementert og testet dokkingløsning for instituttets AGV.		
Planlagt start: 1. mai 2010	Kostnadsestimat: Deler og produksjon	
Planlagt slutt: 3. juni 2010	Ressursbehov: 34 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland

Kostnader, tid, ressurser (KTR)		
Arbeidspakke nr.: AP8	Prosjekt: Dokkingssystem for automatisk selvgående vogn (AGV)	
Tittel: Klargjøring av oppgaven		
Aktiviteter: <ul style="list-style-type: none"> - Skrive ferdig tekst - Lese korrektur - Justere formatering og layout - Utskrift - Innbinding - Levering 		
Beskrivelse av arbeidsomfang: Oppgaven ferdigstilles og klargjøres for innlevering.		
Begrensninger: Avsatt tid for oppgaven		
Resultat: Ferdig korrekturlest, rettet og innbundet oppgave i to eksemplar samt et elektronisk eksemplar.		
Planlagt start: 4. juni 2010	Kostnadsestimat: 0 NOK	
Planlagt slutt: 13. juni 2010	Ressursbehov: 10 dager	
Revisjon: 0	Dato: 25. januar 2010	Utføres av: Petter Jemtland