



INSTITUTT FOR ELEKTRONISKE SYSTEMER

TEKNISK RAPPORT

Ω THEMIS LIBRA

Forfattere:
Gruppe 2
Abasin Essa,
Hans Henrik Heiberg,
Henry Grindheim Hogstad,
Gustav Kollstrøm,
Sindre Hyldbakk Kvande,
Elias Solhaug

Trondheim, 2021

Anerkjennelser

Takk til vår faglige veileder førsteamanuensis Milica Orlandic for veiledning, opplæring og støtte under utviklingen av produktet. Uten henne ville Libra aldri kommet så langt i prosessen.

Takk til førsteamanuensis Torstein Bolstad for råd og tilrettelegging av et rikt innovasjonsklima i prosjektets begynnelse.

Takk til Anna G. Haug for verdifulle innspill om prosjektledelse og rapportskriving.

Takk til Magnus Ramsfjell for konstruktive samtaler om innovasjonsmuligheter innenfor curlingidretten og tilbakemeldinger på Libra.

Takk til professor Lars Lundheim for hans engasjement til prosjektet og initiativet han viste for å lokalisere den sterkeste kompetansen på elektriske motorer innenfor NTNU Trondheim.

Takk til førsteamanuensis Bjørn B. Larsen for innspill om finansiering av prosjektet.

Takk til avdelingsingeniør Christer Næsset og det mekaniske verkstedet for hjelp til å producere metallstenger, treplate og bestilling av kulelager.

Takk til Martin Sesaker for verdifull informasjon, engasjement og kompetanse til produsjon av Libra og for å stille opp til intervju og video-opptak.

Takk til Steffen Walstad for kritiske spørsmål og konstruktive tilbakemeldinger om ulike konsepter knyttet til innovasjon i curling.

Takk til Bjørn-Tore Ellevold for samtaler og innspill om teknologiens rolle i dagens curling.

Takk til Greger Gahre for konstruktive samtaler om designet av Libra.

Takk til Institutt for Elektroniske Systemer - NTNU for økonomisk støtte.

Takk til Entreprenørskolen for innspill om utførelse av idémyldring.

Takk til Trondheim Curlingklubb for utlån av hallen til testing og oppbevaring.

Takk til Finn G. Maurer for utlån av video-utstyr og innspill om filming.

Takk til Slide Cooper for støtte til feilsøking av motor.

Takk til Tage Grønli for innspill om implementasjon av drivere.

Takk til Norges Curlingforbund for samarbeidet.

Innhold

Innhold	I
1 Forord	1
1.1 Bakgrunn	1
1.2 Curling-begreper	1
1.3 Om curling	2
1.4 Om curling-is, preparering og vedlikehold	2
1.5 Om kosting	2
1.6 Om curlingsteiner	3
2 Problemstilling	5
2.1 Isens egenskaper	5
2.2 Steinens egenskaper	5
2.3 Kostens virkning	6
2.4 Eksisterende bruk av teknologi	6
2.4.1 Curlingsteinkastere	6
2.4.2 Sensorer i omgivelsene	7
2.4.3 Instrumentert curling-kost	7
2.5 Relevante målgrupper	7
3 Konsept	9
4 Design	10
4.1 Fysisk utforming	10
4.2 Skyvesystem	11
4.3 Rotasjonssystem	12
4.4 Sensorsystem	12
4.4.1 Turteller	13
4.4.2 Fartssensor	13
4.4.3 Vinkelsensor	14

4.5	Systemkrav	14
5	Implementasjon	16
5.1	Fysisk utforming	16
5.1.1	Materialer	16
5.2	Implementasjon av skyvesystem	17
5.2.1	Teoretiske grunnlag for valg av motor	17
5.2.2	Trinnmotor med lukket sløyfe	18
5.2.3	Overordnet skjema av skyvesystem	20
5.2.4	Implementasjon av trinnmotor	20
5.2.5	Spenningskilde	22
5.2.6	Brukergrensesnitt	23
5.2.7	Skyvesystemets metodikk	24
5.3	Implementasjon av rotasjonssystem	25
5.3.1	Motoroppkobling	25
5.3.2	Kontrollering av motor	26
5.4	Implementasjon av sensorsystem	26
5.4.1	Turteller	26
5.4.2	Fartssensor	29
5.4.3	Vinkelsensor	30
6	Verifikasjon	32
6.1	Verifikasjonsplan	32
6.2	Kort beskrivelse av planlagte tester	32
6.2.1	Fysisk utforming: Størrelsesmåling	32
6.2.2	Fysisk utforming: Test av mekanisk robusthet	32
6.2.3	Skyvesystem: Mekatronisk presisjonstest	32
6.2.4	Skyvesystem: Maksimal fartstest	33
6.2.5	Skyvesystem: Test av brukergrensesnittets funksjonalitet	33
6.2.6	Rotasjonssystem: Maksimal rotasjonstest	33
6.2.7	Turteller: Presisjonstest med videoanalyse	33

6.2.8	Turteller: Presisjonstest med signalanalyse	33
6.2.9	Fartssensor: Presisjonstest med videoanalyse	34
6.2.10	Fartssensor: Presisjonstest med signalanalyse	34
6.2.11	Presisjonstest av vinkelsensorer.	35
6.3	Tester av Fysisk utforming	35
6.3.1	Størrelsesmåling	35
6.3.2	Mekanisk robusthet	35
6.4	Tester av Skyvesystem	36
6.4.1	Maksimal hastighetstest	36
6.4.2	Mekatronisk presisjonstest	36
6.4.3	Test av brukergrensesnittets funksjonalitet	37
6.5	Tester av rotasjonssystem	37
6.5.1	Maksimal rotasjonstest	37
6.6	Tester av Sensorersystem	38
6.6.1	Fartssensor: Presisjonstest med signalanalyse	38
6.6.2	Farttssensor: Presisjonstest med videoanalyse	39
6.6.3	Turteller : Presisjonstest med signalanalyse	40
6.6.4	Turteller: Presisjonstest med videoanalyse	42
6.6.5	Vinkelsensor: Presisjonstest	43
6.7	Test- og systemkrav-vurdering	45
7	Validering	48
8	Forslag til forbedringer	49
8.1	Mekaniske forbedringer	49
8.2	Elektroniske forbedringer	49
8.2.1	Skyvesystem : Implementasjon av statussignaler	49
8.3	Sensorforbedringer	50
8.3.1	Fartssensor	50
8.3.2	Turteller	50
8.3.3	Vinkelsensor	51

9 Konklusjon	52
9.1 Innsikt oppnådd	52
9.2 Viktige resultater	52
A Skyvesystem	56
A.1 Kretsdiagram av oppkobling	56
A.2 Datablad Nema 24	56
A.3 Driver (CL57Y)	57
A.4 Spenningskildens datablad.	58
A.5 Driverens innganger	59
A.5.1 Kontrollsignal	59
A.5.2 Statussignal	59
A.5.3 Enkoder	60
A.5.4 Motor og spenningskilde	60
A.5.5 Posisjoneringstest	60
B Kode for skyvesystemets metodikk	63
C Kode for rotasjonsmotor	67
D Sensorer	68
D.1 Bilde av Rotasjonssensor	68
D.2 Bilde av vinkelsensor	69
E Kode for turteller	70
F Kode for fartssensor	72
G Kode for vinkelsensor	73

1 Forord

1.1 Bakgrunn

Institutt for Elektroniske Systemer (IES) ved NTNU har sammen med 2. års-studenter på Elektronisk Systemdesign og Innovasjon samt Norges Curlingforbund inngått et samarbeid for å løse problemstillinger innenfor curlingsporten. Potensielle problemstillinger presentert av Curlingforbundet er; automatisk tidsnøying, loggføring av fart, akselerasjon og posisjon av steinen, og metoder for analyse av is-kvalitet.

For å avdekke flere behov eller problemer i curlingsporten har det blitt gjennomførst spørsmålsrunder og intervjuer med forskjellige personer i det lokale curlingmiljøet i Trondheim. Personene ble stilt spørsmål om sporten, diverse prosesser og treningsmetoder som kan forbedres. Blant intervjuobjektene er ice-maker i Leangen Curlinghall og spillere ved toppnivå i sporten.

1.2 Curling-begreper

Tabell 1 viser en oversikt over curlingbegreper brukt i rapporten.

Tabell 1: Oversikt over curlingbegreper brukt videre i rapporten.

Begrep	Forklaring [1]
Senterlinje	En linje som strekker seg langs midten av isen i lengderetning, brukt som et visuelt hjelpemiddel.
Curl	Bevegelsen til en stein vekk fra en rett linje.
Curlingkost	Innretningen som brukes til å børste foran en bevegende stein for å få den til å bevege seg lenger og curle mindre.
Pebbles (eng)	Små vanndråper sprayet med vilje på isen som forårsaker uregelmessigheter på overflaten, slik at steinene curler. Også et verb; handlingen å avsette vanndråper på isen.
Boet	De tre konsentriske ringene der poeng blir skåret.
Steinens utgang	Steinens parametre når den forlater en curling-kaste-maskin: fart, vinkel og rotasjonshastighet.
Ice-maker	Person som er ansvarlig for å vedlikeholde isen; plikter inkluderer, men er ikke begrenset til, pebbling og skraping av isen.
Running band (eng)	Delen av steinen som er i kontakt med isen. Vist i figur 2.
Papering (eng)/Rubbing (no)	Prosessens å bruke sandpapir for å gjøre undersiden av curlingsteinen grovere, slik at den curler mer [2].
Steinens egenskaper	Hvordan steinen oppfører seg på isen sammenlignet med andre steiner under like forhold.

1.3 Om curling

Curling er en idrett der spillere skyver (slider) steiner på isen mot et målområde bestående av en sirkel, segmentert i fire, kalt boet. To firemannslag sender annenhver stein nedover banen med hensikt om å plassere flere steiner nærmere midten av boet enn motstanderenes nærmeste stein[3]. Lagene har mulighet til å feie foran bevegende steiner med koster for å påvirke Steinens bane til ønsket plassering.

Ettersom curling omtales som en av de mest ømfintlige sportene i verden [4], er rammene rundt selve spillet, som is og steiner, meget viktige.

1.4 Om curling-is, preparering og vedlikehold

Små endringer i isen gir store endringer i Steinens oppførsel. På en ubehandlet is vil friksjonen være stor nok til å stoppe steinen halvveis over banen, selv når kasteren sender steinen hardest mulig [5]. På en godt preparert bane vil det være enkelt å sende en stein over hele banen.

Ved å spille på en godt preparert bane vil man oppleve at spillerenes ferdigheter gir store utslag i spillet ved at steinen glir lettere og curler mer sammenlignet med dårlig preparerte baner. Dette muliggjør komplekse og intrikate manøvre i spillet. Ved dårlig preparerte baner vil man oppleve at spillerenes ferdigheter utgir en mindre forskjell. Isens påvirkning på steinen vil være større en kasterens.

“From the first day of competition to the last day of competition, our goal is consistency, so when they throw that first rock they have conditions [as close as possible] to the last rock they throw before they leave.”

(Shawn Olesen, Curling-is teknikker for United States Curling Association, 2014)
[5]

Det kreves mye arbeid for å lage og vedlikeholde god is. For å lage is til vinter-OL i 2018 samarbeidet åtte til tolv teknikere i fire til fem dager på rad med arbeidsdager mellom 12 og 14 timer [6].

Vitenskapen bak is-prepareringen endrer seg stadig. Ice-makere over hele verden samles for workshops, sertifiseringer, og for å utveksle tips i forkant av hendelser som OL. Det er berømte ice-makere som reiser for å trenere andre. Det er selve spillet som får berømmelse, likevel har pebbling, og strategien bak det sin egen ære, i følge Olesen [5].

1.5 Om kosting

Dyktige kostere kan øke distansen på et kast med opp mot 4 meter [7]. Det gir en god feilmargin for kasteren.

Curling-kosting, teknikker og koster er stadig i endring selv om idretten har røtter tilbake til det 16. århundret [8]. Et utvalg av kostere er vist i figur 1. I 2015 skjedde det store omveltninger

da det ble brukt nye materialer under kostene. De nye materialene skapte mer friksjon mot isen og kosteren påvirket steinens bane og lengde i større grad enn tidligere. Flere mente idretten gikk fra å være et kaste-spill til å bli et koste-spill. Fenomenet ble kalt "Broomgate" og World Curling Federation (WCF) valgte å forby enkelte materialer [9], [10].



Figur 1: Et assortert utvalg av curlingkoster.

Den nye teknologien i Broomgate brakte med seg nye teknikker som brøt med eldre konvensjoner og visdom. Selv om materialene ble luket vekk fra sporten ble teknikken kalt "directional sweeping" brakt videre [10].

"Directional sweeping, absolutely, without a doubt, does work. How you do it is everything, though. Finding the proper techniques is essential. There are so many variables that are involved. For a lot of curlers, it's about the eye test. What works to the eye? Are you making shots?"

(Colin Hodgson, Mike McEwen team Winnipeg, 2021) [11]

"Obviously we've gotten a lot better at [directional sweeping] through practice. Everybody had to figure out how to do it. I think everybody understands that it's part of the game now. It has really, quickly, engrained itself into how we play now."

(Mike Fournier, Quebec skip, 2021) [11]

1.6 Om curlingsteiner

"For perfect rocks, you want something where all of them have a consistent running surface, so they react all the same. Just consistency. [...]. Some [rocks] can be almost like people. They have their own characteristics."

(Jamie Bourassa, Ice-maker i fire tiår, 2016) [12]

Bunnen av en curlingstein er ikke flat, men mer som en smultring som kommer i kontakt med isen. Dette reduserer overflatearealet i kontakt med isen, og øker vekten per arealeenhet. Dette øker trykket og temperaturen på isen som senker friksjonskoeffisienten. Steinen går dermed lenger [13].

Før turneringer blir steinens running band, vist i figur 2, tilpasset for å oppnå ønskede egenskaper [2]. Steinene rubbes for å oppnå ønsket bredde og ruhet på running band. Ønsket bredde varierer mellom 6.4 og 12.7 mm [14].



Figur 2: Illustrasjon av curlingsteneres underside. Den innerste sirkelen viser delen av steinen som er i kontakt med isen; steinens running band.

"We have a recipe [on papering stones] and then we fine-tune it,"
(Greg Ewasko, Is-teknikker, 2019) [2]

Ewasko sier videre at hver ice-maker har sin egen teknikk når det kommer til rubbing, og at det noen ganger er nødvendig å utføre rubbing underveis i konkurranser [2].

Curlingsteiner begynner med å veie nøyaktig 42 pund. Gjennom årene går de ned i vekt grunnet rubbing og sandblåsing [2]. Ulik vekt gir ulike egenskaper:

"You're starting to see new stones used at the Scotties and the Brier. And what's happening is there's more action with them. They're 42 pounds as opposed to 38 pounds."

(Kim Tuck, Canada Curling Stone Co., 2019) [2]

2 Problemstilling

Med bakgrunn i informasjonen beskrevet i seksjon 1 er det avduket en mangel av analyseverktøy for å kartlegge hvordan forskjellige prosesser for ispreparering påvirker steinens oppførsel på banen. Det er tilsvarende mangel på analyseverktøy for curlingkosting, -koster og -steiner.

Majoriteten av curlinghaller har i 2021 ikke tilgjengelige redskaper til å kvantifisere:

- **Isens egenskaper:** Hvordan forskjellige metoder for ispreparering påvirker steinens hastighet, curl (rotasjon) og sluttposisjon.
- **Steinenes egenskaper:** Hvordan ulikt bearbeidede steiner oppføre seg på isen.
- **Kostens virkning:** Hvordan endrer steinens sluttposisjon seg ved forskjellige kosteteknikker og koster.

Ønsker man å forbedre et produkt, begynner man med å gjøre en måling, etterfulgt av en endring, før man måler igjen for å kontrollere om ønsket resultat er oppnådd. Skal man for eksempel gjøre en racerbil raskere kan man måle topphastighet før og etter endringen ble implementert. Skal det samme gjøres for curling-is, steiner og kosting mangler man verktøy for å gjøre en troverdig verifikasjon. Det som derimot er felles for racerbiler og curling er at flere mindre endringer kan gi store utslag i sluttresultatet.

2.1 Isens egenskaper

Mangelen på metoder og verktøy for å kvantifisere isens egenskaper er en hindring i arbeidet med å preparere og vedlikeholde curling-is på en god måte.

2.2 Steinens egenskaper

Uten verktøy for å kvantifisere steinens egenskaper blir det vanskelig for curlinghaller å konsekvent;

- Sikre at steinene oppfører seg likt ved likt underlag.
- Utvikle gode oppskrifter for oppbevaring, bearbeiding og vedlikehold av steinene.
- Gjennomføre tester for å se etter ujevnheter blant steinenes oppførsel.

Curlinghaller har per i dag mulighet til å veie steinene, samt måle bredden på running band. Som beskrevet i seksjon 1.6 er steinens karakterstikk mer kompleks enn det to parametere presist kan beskrive. Samtidig er det ønskelig å teste hvilke endringer i steinen som gir ønsket resultat.

2.3 Kostens virkning

Det er i dag vanskelig å presist kartlegge hvordan kosting alene påvirker sluttposisjonen til en curlingstein. Manglende verktøy til å kvantifisere kostens virkning gjør det vanskelig å sammenligne ulike;

- **Kosteteknikker:** Hvordan endres curlingsteinens bane seg med forskjellige teknikker, frekvens osv.
 - **Kostere:** Har enkelte spillere bedre grunnlag for kosting i forskjellige situasjoner på isen.
 - **Koster:** Hvilke parametre på kosten gir ønskelig resultat.

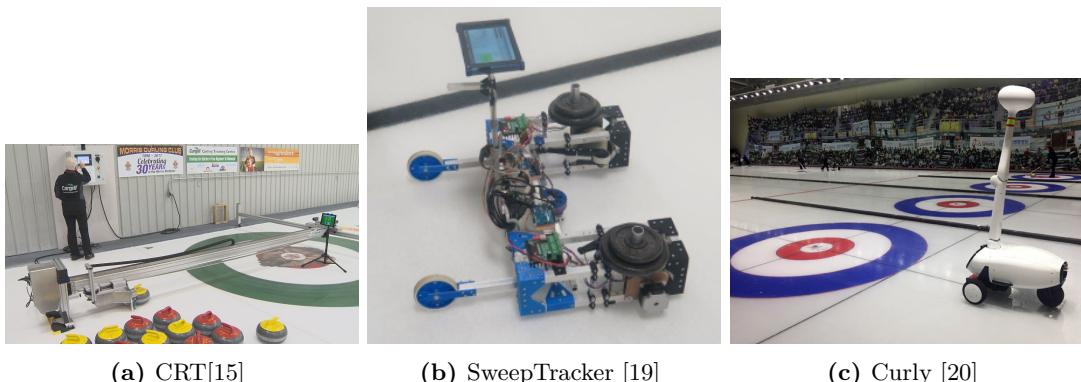
I en detaljfokusert sport som curling, stiller man med et stort konkurransesfortrinn dersom man har mulighet til å analysere, og være best på, de minste detaljene. Presis informasjon om kosting er derfor viktig.

2.4 Eksisterende bruk av teknologi

2.4.1 Curlingsteinkastere

Det finnes allerede roboter som kaster curlingsteiner. Ved å kaste steiner med samme utgangshastighet og rotasjon kan påvirkningen fra omstendighetene rundt sammenlignes. Ved å sende flere like steiner hvor man endrer på isen mellom kastene vil man kunne loggføre isens påvirkning på steinen og notere hvilken metode som gir ønskelig resultat. På samme måte kan man sammenligne ulike stein-rubbe-metoder og koste-teknikker.

I tabell 2 og figur 3 ser man en oversikt over allerede eksisterende curslingsteinkastere.



Figur 3: Bilder av allerede eksisterende curlingsteinkastere.

Dagens utfordring for curlingmiljøet er at disse robotene er ekstremt dyre. Curling har historisk vært en idrett uten større inntekter, og det virker sannsynlig at dette vil fortsette i tiden fremover. Dette betyr at disse eksisterende redskapene i praksis er utilgjengelige.

Tabell 2: Oversikt over allerede kjente curlingsteinkastere.

Navn	Beskrivelse	Pris
Canadian Rock Thrower (CRT)	Vist i figur (3a). Skinnebasert curlingsteinkaster. Per nå finnes det ett eksemplar lokalisert i Morris Curling Club i Canada [15]. Underkant av 4 meter lang.	\$75,000.00 CDN [16]
SweepTracker	Vist i figur (3b). Hjulbasert curlingsteinkaster. Krever at brukeren selv må starte rotasjon på stenen og føre roboten i ønsket retning [17]. Lik utgang kan ikke verifiseres.	Prototype. Ikke til salgs [17].
Curly	Vist i figur (3c). Avansert hjulbasert curlingstenkaster. Utviklet ved Korea-Universitetet og Technische Universität Berlin. Laget som en plattform for å demonstrere interaksjonen mellom et AI-system og den virkelige (glatte) verdenen [18].	Ukjent

“You don’t have to rely on someone saying that they throw the rock the exact same way every time because no one can.”

(Phil Bernardin, Produsent av CRT, 2017) [15]

“Every time we use [CRT], we see other opportunities for how it can help teams better their skills,”

(Chris Hamblin, tidligere landslagstrener for Kina og Sveits, 2017) [15]

2.4.2 Sensorer i omgivelsene

Det finnes sensorer som analyserer omgivelsene, både under og over isen, som temperatur og luftfuktighet [12]. Slike sensorer kan, i kombinasjon med en curlingsteinkaster, brukes for å kartlegge hvilke faktorer som påvirker curl og hastighet til steinene.

2.4.3 Instrumentert curling-kost

Koster instrumentert med sensorer brukes for å kvantifisere trykk mot isen og hastighet på kostingen. Det slike koster ikke presist kan gi mål på er kostingens virkning på curl og hastigheten til steinen.

2.5 Relevante målgrupper

“We have used [CRT] since 2017. We have found it to be very useful in teaching and developing the most effective sweeping techniques, and also in understanding

ice conditions and rock movement [...]. The World Curling Federation spent a week with us along with the National Research Council of Canada prior to covid, testing and quantifying many of the theories for sweeping and ice management.

(Lorne & Chris Hamblin, tidligere landslagstrener for Kina og Sveits, 2021) [16]

Et nytt eksemplar av CRT skal leveres til en ukjent nasjon i 2021 [16].

“Every [site] has its own little character. Every time you go somewhere you think you have it figured out but it’s still a challenge”

(Shawn Olesen, Curling-is teknikker for United States Curling Association, 2014) [5]

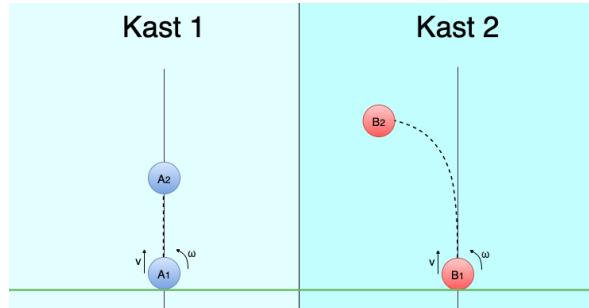
Ikke bare er alle curlinghaller forskjellig, men forholdene i hver enkelt hall vil også endre seg. Det er derfor ønskelig å kunne gjennomføre tester i alle hallene jevnlig. Derfor vil samtlige curlinghaller i Norge og utlandet være relevante målgrupper.

Det er i 2021 oppført 15 curlinghaller i Norge [21]. Det vil trolig være aktuelt for majoriteten av disse å tilskaffe seg en curlingsteinkaster om prisen er passende.

3 Konsept

Systemet, illustrert i figur 4 skal fungere som et verktøy til analyse av curlingisens egenskaper ved å sende én og samme curlingstein med lik hastighet og curl flere ganger. På denne måten kan blant annet ice-makere kartlegge hvordan forskjellige prosesser for islegging påvirker curlingsteinens bevegelsesbane. Tilsvarende skal systemet fungere som et verktøy til analyse av curlingsteiner og curlingkosting.

Det er derfor viktig at systemet sender curlingsteiner med samme utgangsfart, utgangsvinkel og curl flere ganger etter hverandre, både ved flytting av systemet og ved lengre ventetid mellom kastene. Systemet skal kunne verifisere og gi tilbakemelding om eventuelle avvik i curlingsteinens utgangshastighet, utgangsvinkel og curl. Det skal være mulig for bruker å bestemme utgangsfart og curl på curlingsteinen. Curlingsteinen skal skyves rett frem med en curl og en hastighet. Ved varsel om forskjeller i utgangsvinkel mellom to kast vet bruker at sluttposisjonene ikke kan sammenlignes.



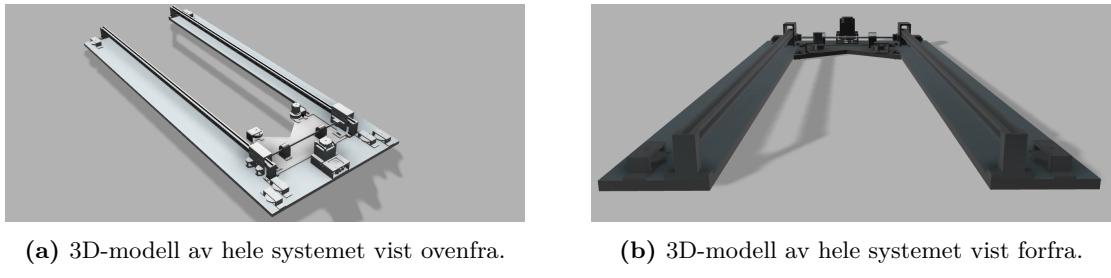
Figur 4: Konsept. Utgangen på kastene er like, men isen er ulik. Steinens oppførsel på isen brukes til å sammenligne isens kvalitet.

I figur 4 ser man to separate hendelser, ett kast til venstre og ett til høyre. sirklene A_1 og B_1 illustrerer curlingsteiner i tidspunktet de kastes. Sirklene A_2 og B_2 illustrerer steinenes avslutningsposisjon. Steinene har lik utgangshastighet V og utgangsrotasjonshastighet ω . Dermed har de lik utgang. Ved å sammenligne Stein A og Stein B sine baner kan man analysere hvilken is som gir ønsket resultat.

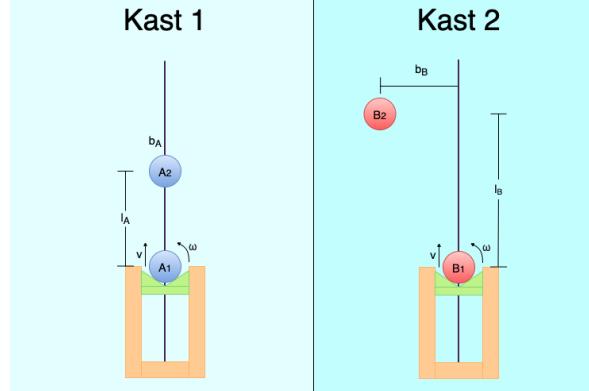
4 Design

Figur 5 illustrerer en 3D-modell av hvordan curlingkasteren, Libra, er tenkt utformet. Libra er delt opp i delsystemene skyvesystem, rotasjonsystem, sensorsystem og den fysiske utformingen.

Figur 6 illustrerer den tenkte funksjonaliteten til hele systemet. Den viser en curlingsteinkaster i fugleperspektiv vist i to separate hendelser, én til venstre og én til høyre. Sirklene A_1 og B_1 illustrerer curlingsteinene i tidspunktet de kastes. Sirklene A_2 og B_2 illustrerer steinenes avslutningsposisjon. Steinene har lik utgangshastighet V og utgangsrotasjonshastighet ω . Dermed har de lik utgang. l_A og l_B er henholdsvis stein A og stein B sin lengde-forskyvning. b_A og b_B er stein A og stein B sin bredde-forskyvning. Ved å sammenligne steinenes lengde- og breddeforskyvning kan man analysere hvilken is som gir ønsket resultat.



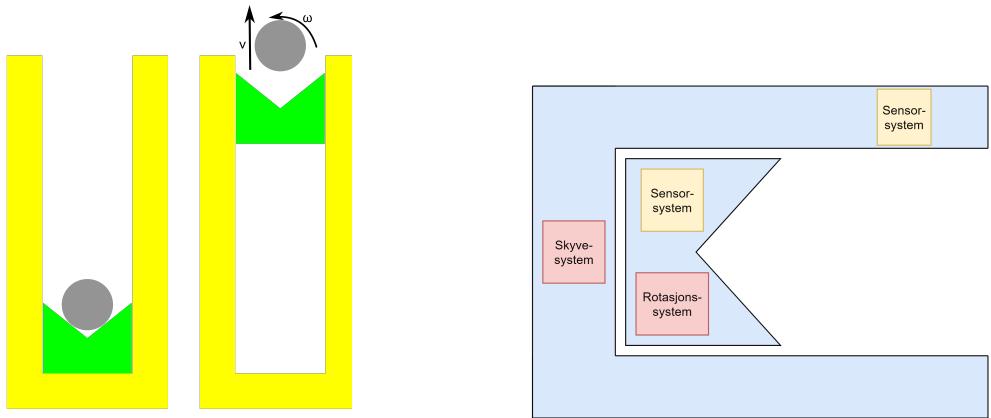
Figur 5: 3D-modell for design av hele systemet.



Figur 6: Design. Steinenes avslutningsposisjon brukes til å sammenligne isens kvalitet.

4.1 Fysisk utforming

Det fysiske produktet består av en todelt base, en statisk enhet og en dynamisk enhet. Oppdelingen av produktet er illustrert i figur 7a. Hvert delsystem skal implementeres på disse basene. Figur 7b viser hvordan skyvesystemet implementeres på den statiske enheten og rotasjonssystemet monteres på den dynamiske enheten. Sensorsystemet har moduler på begge enhetene.

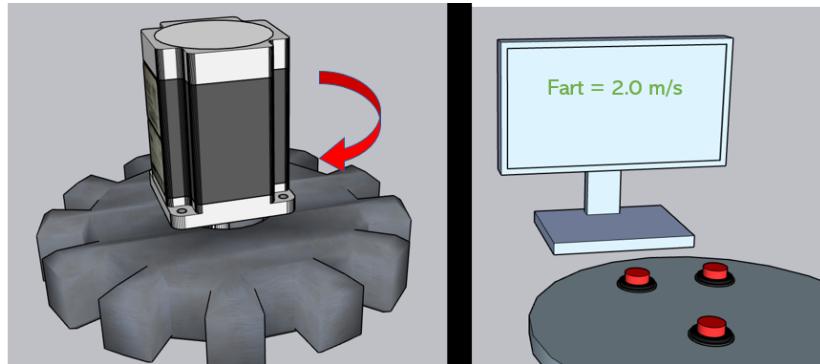


(a) Systemets to baser i to forskjellige faser av kastet. (b) Plassering av delsystemer på de ulike modulene.

Figur 7: Illustrasjon av den fysiske utforming.

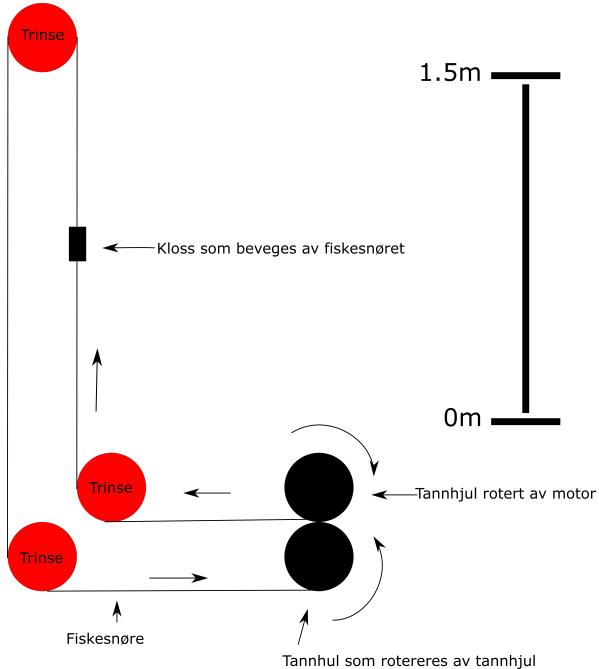
4.2 Skyvesystem

Skyvesystemets funksjon er å produsere nok mekanisk kraft til å kontrollert akselerere en last på 20kg slik at lasten når ønsket fart innen en bestemt strekning. Brukeren av systemet skal kunne regulere hvilken fart lasten skal oppnå innen den fastsatte strekningen. Skyvesystemets design vises i figur 8 der en motor koblet til et tannhjul påvirker lasten, sammen med en illustrasjon av brukergrensesnittet der bruker kan justere ønsket fart.



Figur 8: Illustrasjon av forventet konseptdesign av skyvesystemet. Det vises at lastens bevegelse er styrt av brukeren. Figuren viser ikke den totale lasten systemet akselerer.

Det mekaniske prinsippet til skyvesystemet vises i figur 9. Figuren viser hvordan skyvesystemet skyver den dynamiske basen, inkludert rotasjonssystemet, sammen med curlingsteinen. To snorer, festet til den dynamiske basen, vil ved hjelp av tannhjul koblet til motoren spoles opp og igjen for å dra den dynamiske basen frem og tilbake.



Figur 9: Viser bevegelsen skyvesystemet utfører. Her flyttes en kloss av en elektrisk motor. Klossen byttes ut med basen til rotasjonssystemet.

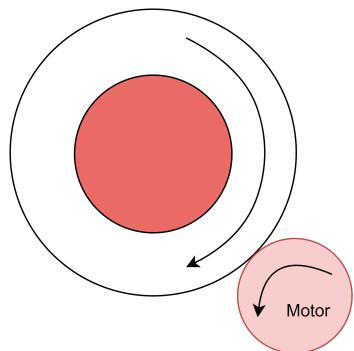
4.3 Rotasjonssystem

Hensikten med rotasjonssystemet er å ha et delsystem som kan produsere en rotasjon, eller curl, på steinen i kontrollerete former. Dette vil si at bruker skal ha mulighet til å regulere hastigheten på rotasjonen. Motorens kontakt med steinen er illustrert i figur 10.

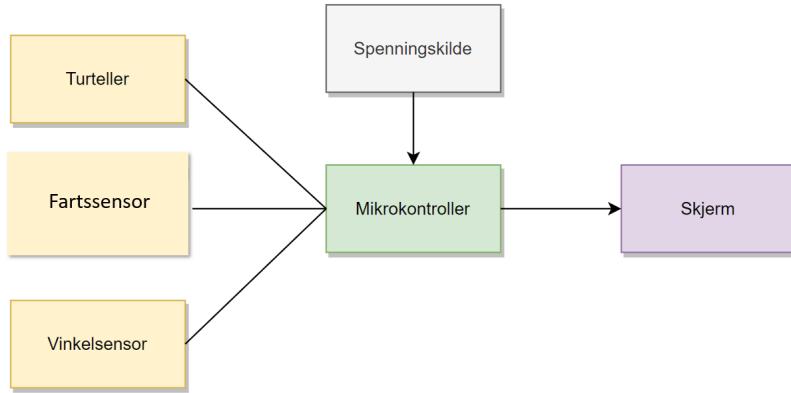
4.4 Sensorsystem

Sensormoduler monteres på basen til systemet for å validere at den reelle utgangsvinkelen, rotasjonshastigheten og farten tilført steinen er lik ved hver utskytning. Dette gjør det også mulig å gi tilbakemelding til bruker i form av reell hastighet og rotasjon på steinen.

Blokkskjemaet til sensorsystemene er illustrert i figur 11. Turtelleren sørger for å sende signaler til mikrokontrolleren som gjør det mulig å beregne rotasjonshastigheten, mens fartssensoren sender signaler som gjør det mulig å beregne utgangsfarten. Mikrokontrolleren sørger for å vise beregnet rotasjon og hastighet på skjerm.



Figur 10: Viser et hjul som roterer mot curlingsteinen og overfører rotasjonsenergi steinen.



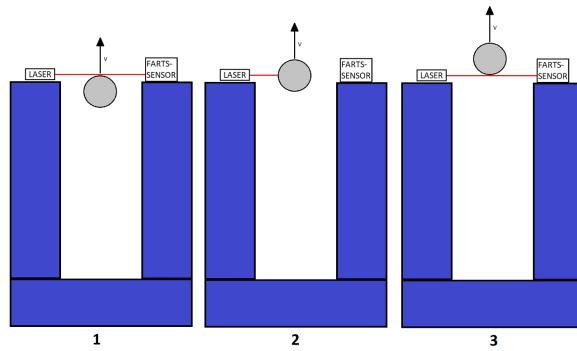
Figur 11: Sensorsystemene som brukes for å validere systemet.

4.4.1 Turteller

En turteller brukes for å måle rotasjon tilført steinen fra det mekaniske systemet. Hensikten er å validere at ønsket rotasjon i henhold til innstillinger på det mekaniske systemet stemmer overens, samt at rotasjon på steinen er lik ved hver utskyting med lik nevnt innstilling. Dette kan løses ved å benytte sensorer på rotasjonssystemet for å telle antall rotasjoner per tidsintervall.

4.4.2 Fartssensor

Et fartssensorsystem brukes for å måle utgangsfarten curlingsteinen skytes ut av det mekaniske systemet med. Dette gjøres for å kunne kontrollere og validere at farten er lik ved hver utskyting med lik fartsinnstilling. Tanken er å danne en laserbarriere som brytes når curlingsteinen passerer. Videre kan tidsintervallet barrieren er brutt samt curlingsteinens diameter brukes til å beregne steinens utgangsfart. Systemets overordnede design er illustrert i figur 12.



Figur 12: Idéen bak fartssensoren. Systemet detekterer hvor lenge laseren er brutt og beregner farten ved hjelp av curlingsteinens diameter.

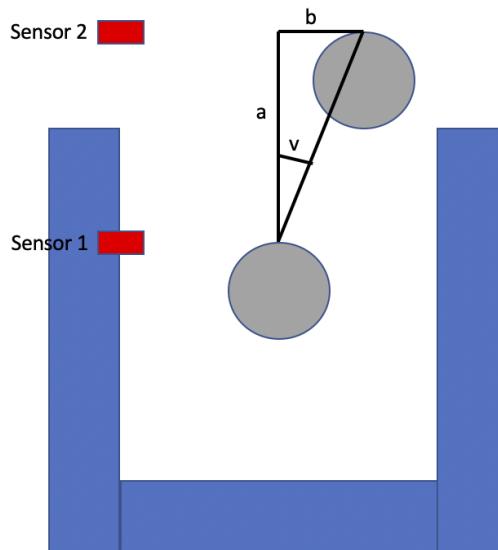
4.4.3 Vinkelsensor

For å lage en vinkelsensor og sjekke utgangsvinkelen til steinen, er to avstandssensorer implementert. Det er viktig at utgangsvinkelen er den samme ved to identiske kast, for at de skal kunne sammelignes. Dermed er kravet for denne modulen et avvik på, $\pm 3^\circ$. Alle kravene vises i tabell 3.

For å måle utgangsvinkelen settes sensorene som vist i figur 13. Sensorene kan ikke stå for langt unna hverandre, fordi isen vil etterhvert påvirke steinen sin bane. Ved å vite avstanden mellom sensorene, a, og forskjellen på avstanden til steinen ved "Sensor 1" og "Sensor 2", b, får man en tiltenkt rettvinklet trekant. Dette er illustrert i figur 13. Ved hjelp av trekanten kan man bruke formel 1 for å beregne utgangsvinkelen.

$$\tan v = \frac{b}{a} \quad (1)$$

For å gjøre utgangsvinkelen, v, fra radianer til grader, multipliseres resultatet med 57.3. Dette tilsvarer hvor mange grader en radian er.



Figur 13: Ideen bak vinkelsensoren er å bruke to avstandssensorer for å regne ut utgangsvinkelen, v. Da er a, avstanden mellom sensorene, som velges, og b er avstanden til "Sensor 2" minus "Sensor 1". Da kan vinkelen, v, finnes ved likning (1) multiplisert med 57.3.

4.5 Systemkrav

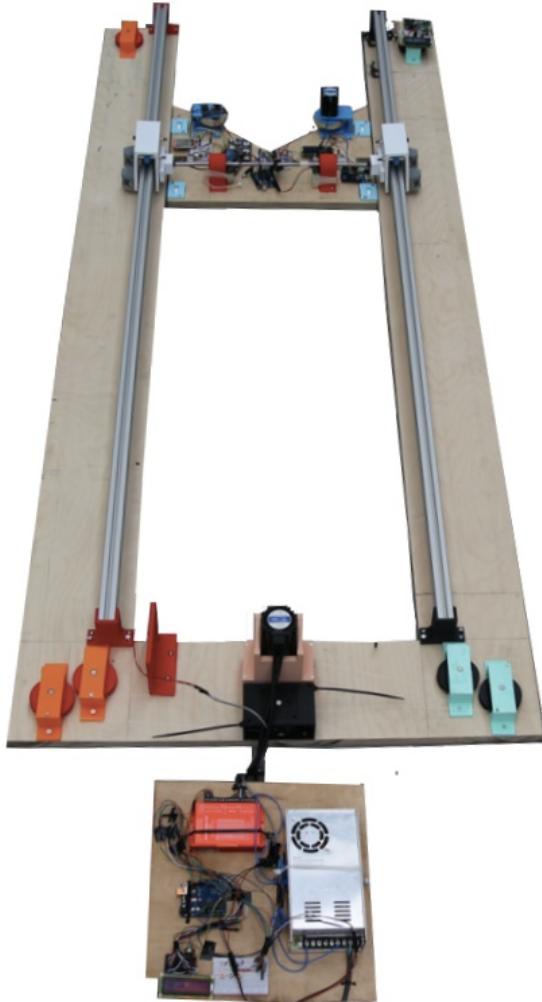
For å realisere Libra, slik at systemet er pålitelig og nøyaktig, er det satt krav til systemets oppførsel. Tabell 3 viser systemkravene til hele systemet samt de ulike delsystemene.

Tabell 3: De bestemte kravene til systemet.

Delsystem	Modul	Krav	Beskrivelse
Fysisk utforming	Rammer	Størrelse	Mindre enn $2m \cdot 1m \cdot 0.5m$.
Fysisk utforming	Matrialer	Fysisk robusthet	Produktets matrialer tåler kast.
Skyvesystem	Motor	Kraft	Oppnå fart 2.5 m s^{-1} på en curlingstein.
Skyvesystem	Motor	Presisjon	Utgangsfart oppnås etter samme strekning (1.5m).
Skyvesystem	Brukergrensesnitt	Brukervennlighet	Bruker kan velge utgangsfart.
Skyvesystem	Brukergrensesnitt	Brukervennlighet	Brukeren har tilgang på en nødstopp.
Rotasjonssystem	Motor	Vinkelhastighet	På is rotere curlingstein til 30 rpm.
Sensorsystem	Rotasjonssensor	Presisjon	Maks målingsavvik $\pm 2.5\%$.
Sensorsystem	Vinkelsensor	Presisjon	Maks målingsavvik $\pm 3^\circ$.
Sensorsystem	Fartssensor	Presisjon	Maks målingsavvik $\pm 2.5\%$.

5 Implementasjon

Det implementerte produktet, Libra, vises i figur 14. I følgende seksjon skal de tekniske løsningene forklares og diskuteres i nærmere detalj.



Figur 14: Libra ferdig montert, med alle delsystemer og moduler.

5.1 Fysisk utforming

5.1.1 Materialer

Basene ble skåret ut i treverk av 2 cm tykk kryssfiner, som er et sterkt tremateriale. Den U-formede basen ble skåret ut med omriss i dimensjonene 2 m x 0.91 m x 0.02 m for å oppfylle systemkravet for størrelse.

Trinser, tannhjul, braketter, hjul, stativ og fester er 3D printet i materialet PLA. Skinnene er

kuttet i aluminium. Roterende deler er festet med små jernstenger eller jernskruer sammen med spacere i aluminium. Spacere er brukt for å plassere hjul og trinser i riktig posisjon. For å redusere friksjon i systemet samt øke stabilitet er det brukt hjullagre mellom jernstenger og de bevegelige delene i systemet. For å holde den dynamiske og den statiske basen sammen er det brukt en jernstang.

Fiskesnøre er brukt som transmisjonsmateriale som overfører kraft fra skyvesystemet til curlingsteinen gjennom å bevege på den dynamiske basen.

5.2 Implementasjon av skyvesystem

5.2.1 Teoretiske grunnlag for valg av motor

For å skyve en curlingstein på 20 kg til en fart på 2.5 m s^{-1} med presisjon kreves det en motor med nok kraft og en metode for å presist kontrollere farten. Dette er løst ved å implementere en trinnmotor ettersom de har generelt høyt dreiemoment, som bestemmer hvor raskt den akselererer og deakselererer, og kan kontrolleres med god presisjon.

Trinnmotorer er hovedsakelig børsteløse elektriske motorer der én rotasjon er delt inn i like store steg, slik at motoren kan kontrolleres til å bevege seg til en bestemt posisjon. Ved å kontrollere motoren i slike diskrete «steg» har man fremtredende kontroll over motorens posisjoner og fart. Dette gir mulighet til å oppnå ønsket fart på kortere distanser.

Hvor mye kraft motoren må produsere er avhengig av hvor lang strekning s curlingsteinen trenger for å bli akselerert til ønsket fart v . Kraften beregnes ved å ta utgangspunkt i bevaring av mekanisk energi. Følgende likning kan løses med hensyn på akselerasjonen a :

$$as = \frac{1}{2}v^2, \quad (2)$$

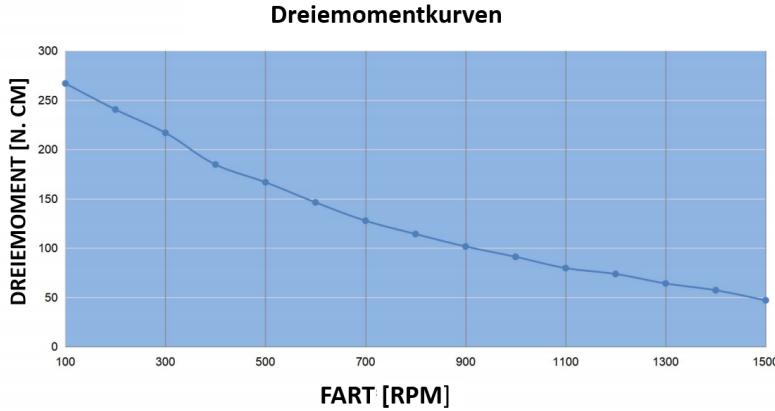
Ved å velge en strekning s på 1.5 m og sluttfart v på 2.5 m s^{-1} , blir akselerasjonen beregnet på følgende måte;

$$a = \frac{v^2}{2s} = 2.08 \text{ m s}^{-2}, \quad (3)$$

videre brukes denne akselerasjonen til å beregne nødvendig kraft F ved Newtons 2.lov. Massen m fastsettes til å være 20kg ,og følgende beregnes:

$$F = ma = 41.67 \text{ N}. \quad (4)$$

For at motoren skal skyve curlingsteinen til valgt fart trenger den et dreiemoment τ på minimum;



Figur 15: Motorenens dreiemoment som funksjon av hastigheten gitt i rpm.

$$\tau = F \cdot r = ma \cdot r = 41.67 \text{ N} \cdot r, \quad (5)$$

uten hensyn til friksjon i isen og innad i systemet.

Radiusen r til tannhjulet som overfører kraften fra motoren til systemet er 0.035 m slik at det minste dreiemomentet motoren må ha er;

$$\tau = 1.458 \text{ N m}. \quad (6)$$

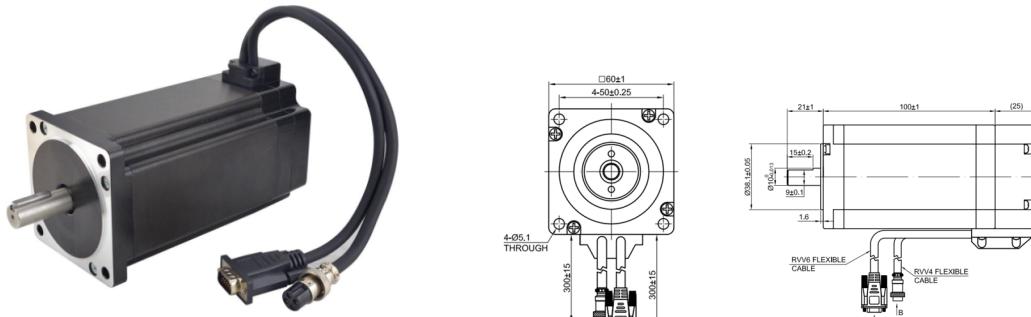
Motoren skal akselerere curlingsteinen til en fart på 2.5 m s^{-1} som tilsvarer 682 rotasjoner per minutt [rpm].

5.2.2 Trinnmotor med lukket sløyfe

En trinnmotor av typen E-series Nema 24 er valgt som kjernen i skyvesystemet. Motorenens utseende og mål vises i figur 16 og databladet finnes i vedlegg A.2. Dette med bakgrunn i motorenens momentkurve illustrert i figur 15 der man ser at motoren kan levere omtrent 1.3 N m ved en hastighet på 700rpm.

Ved å fokusere på å redusere friksjon internt i systemet skal motoren kunne produsere nok kraft til å akselerere curlingsteinen til ønsket fart. En curlingstein kan ha høy statisk friksjon, men ved å utnytte motorenes høye dreiemoment ved lavere hastigheter vil man i startfasen kunne få satt curlingsteinen i bevegelse før motoren akselerer til høyere hastigheter.

Motoren inneholder også en innebygd enkoder. Hovedfunksjonen til enkoderen i systemet er å detektere feil i posisjonering, som gir en høy systemnøyaktighet og bevegelseskontroll over lasten. Denne type motorer kalles “trinnmotor med lukket sløyfe” (closed loop stepper motor).



(a) Bilde av en Nema 24 trinnmotor.

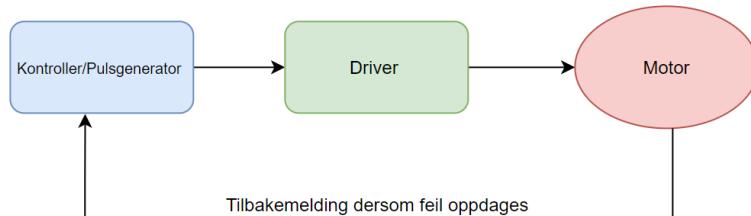
(b) Maskintegning av en Nema 24 trinnmotor. Størrelser er oppgitt i millimeter. Hentet fra databladet som ligger i seksjon A.2.

Figur 16: Et bilde av den brukte motoren for å realisere skyvesystemet. Motoren er en trinnmotor med lukket sløyfe av type E series Nema 24.

Figur 17 illustrerer forskjellen mellom en åpen sløyfe trinnmotor, der motoren ikke signaliserer dersom feil er oppstått og en lukket sløyfe trinnmotor der systemet stopper dersom en feil er detektert.

Flere fordeler med en lukket sløyfe trinnmotor er mindre energitap i form av varme, lavere strømforbruk og lite støy.

Trinnmotorsystem med lukket sløyfe



Trinnmotorsystem med åpen sløyfe

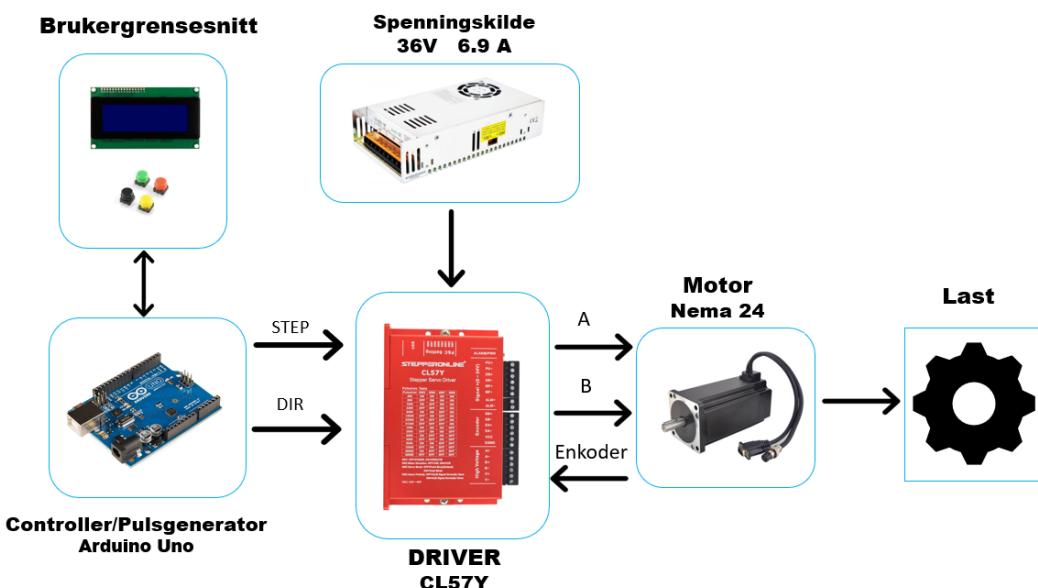


Figur 17: Forskjellen mellom lukkede systemer og åpne systemer. Ved å ha et lukket system, så kan man få en tilbakemelding på hva som har gått galt.

5.2.3 Overordnet skjema av skyvesystem

Illustrert i figur 18 er et blokkskjema av de sentrale elementene i skyvesystemet. Systemet består hovedsakelig av en mikrokontroller/pulsgenerator, en spenningskilde, og en driver som brukes til å kontrollere motoren. Det inneholder også et brukergrensesnitt bestående av en LCD-skjerm som presenterer relevante størrelser for brukeren samt knapper som gir bruker mulighet til å regulere farten før man starter kastet.

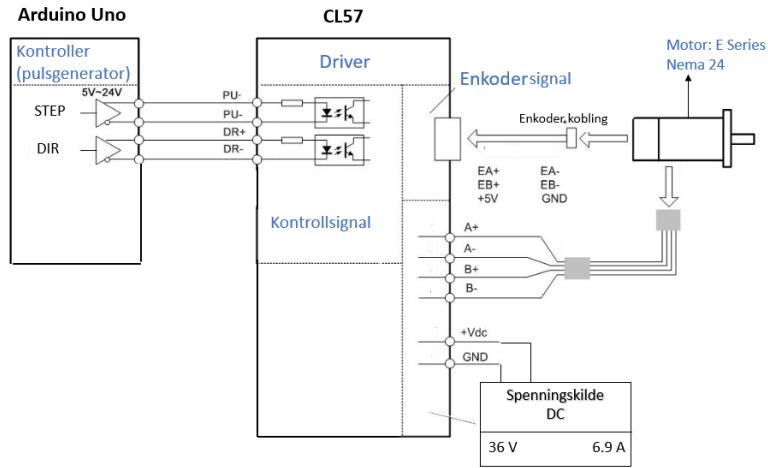
Signalene *STEP* og *DIR* fra mikrokontrolleren fungerer som instruksjoner for motorens bevegelse. *STEP*-signalet er et pulssignal som bestemmer motorens bevegelse. *DIR* er et logisk signal som bestemmer motorens retning. Enkoderen gir også signaler som oppdaterer driveren om det er oppdaget feil i posisjoneringen av motoren. Signalene *A* og *B* er koblet til motorens to faser, bestående av en gruppering av spoler.



Figur 18: Det totale skyvesystemet. Det består av en Kontroller/Pulsgenerator (Arduino Uno), en driver (CL57Y), DC-spenningskilde, LCD-skjerm og knapper.

5.2.4 Implementasjon av trinnmotor

Skyvesystemet er koblet opp som illustrert i blokdiagrammet i figur 19. Detaljert kretsdiagram er vist i figur 47 i vedlegg A.1. De sentrale komponentene i implementasjonen er mikrokontrolleren, Arduino Uno, driveren, CL57Y, og spenningskilden. Samhandlingen mellom disse komponentene får motoren til å rottere.

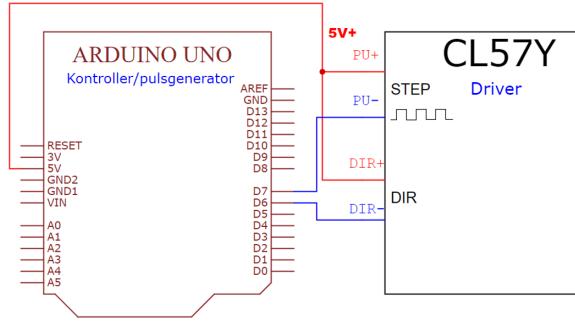


Figur 19: Et overordnet skjema av komponentene som har blitt brukt for å realisere motoren Nema 24.

Hjernen bak alle bevegelsene til motoren er mikrokontrolleren. Denne programmerte enheten sender elektriske signaler til driveren, og fungerer som instruksjoner for motoren. Signalene bestemmer hvor mange steg motoren skal gjennomføre, hvor raskt bevegelsen skjer, og i hvilken retning stegene skal tas. Disse signalene, kalt kontrollsinaler, spiller en vesentlig rolle i hvordan motoren beveger seg. Kontrollsinalene *STEP* og *DIR* kan oppsumeres på følgende måte:

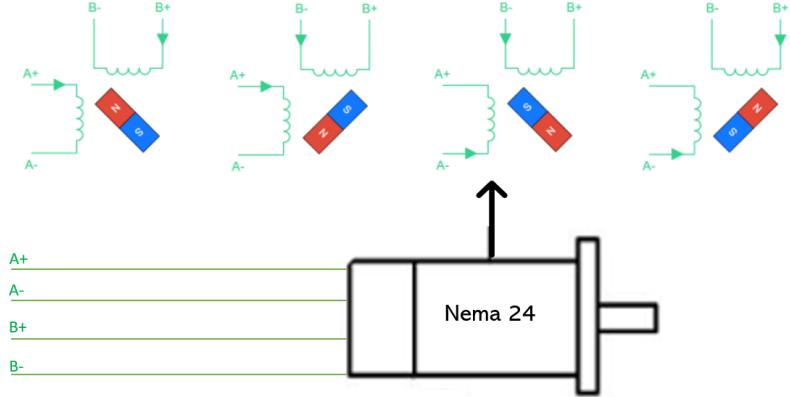
- **STEP:** Dette signalet består av firkantede pulser som bestemmer hvor mange steg motoren skal ta og hvor raskt disse stegene skal gjennomføres. *STEP*-singlet består av **PU+** og **PU-**.
- **DIR:** Dette er et logisk signal som brukes for å bestemme motorens rotasjonsretning. Det består av **DR+** og **DR-**.

Figur 20 viser et utdrag av kretskjemaet med fokus på koblingen av kontrollsinalene mellom mikrokontrolleren og driveren. Signalene kobles med felles anode, og **PU+** og **DIR+** er koblet til en felles +5V kilde fra mikrokontrolleren. Motorens bevegelse styres av driverens negative terminaler, **PU-** og **DIR-**.



Figur 20: Signalene som er koblet mellom kontrolleren (Arduino Uno) og driveren (CL57Y). Signalene er koblet med felles anode.

Hovedoppgaven til driveren er å forsterke og oversette instruksjonene fra mikrokontrolleren til motorisk bevegelse. Dette gjøres ved at driveren sender sterk strøm gjennom fasene A og B i motoren, basert på kontrollsignalene fra mikrokontrolleren. Strømmen gjennom fasene fører til at rotoren i motoren inntar en viss posisjon slik at reluktansen blir minst mulig. Ved å spenningsette fasene i en spesifikk sekvens vil det føre til en spesifikk bevegelse i motoren. Figur 21 illustrerer virkemåten til motoren. Signalene $A\pm$ og $B\pm$ viser koblingene mellom driveren og motoren. Disse signalene driver strømmen gjennom fasene i motoren, og en variasjon i strømmen fører til bevegelse i rotoren.



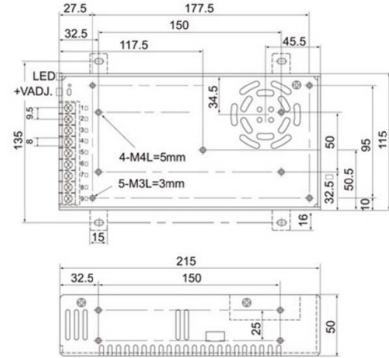
Figur 21: Figuren viser koblingen mellom motoren og driveren. I tillegg illustreres at en endring i magnetfeltet fører til at rotoren beveger seg.

5.2.5 Spenningskilde

For at motoren skal få tilført nok strøm og høy nok spenning er det brukt en spenningsforsyning som gir ut 36 V DC, og leverer strøm opptil 6.9 A. Spenningskilden kobles til enten 115 V eller 230 V nettspenning, og transformerer disse til 36 V. Databladet til spenningsforsyningen er lagt ved i vedlegg A.4. Figur 22 viser den brukte spenningsforsyningen samt måltegning.



(a) Bilde av spenningsforsyningen.



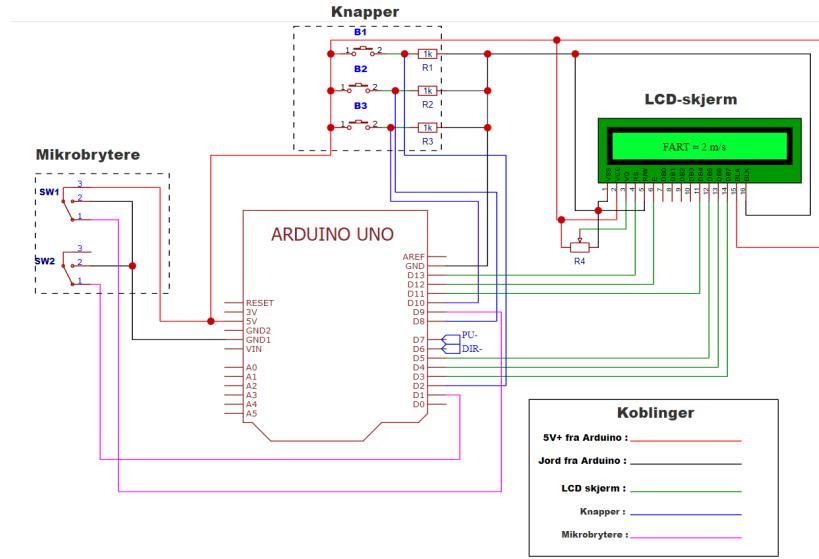
(b) Maskintegning av spenningsforsyningen. Størrelsene er oppgitt i millimeter. Hentet fra databladet som ligger i seksjon A.4.

Figur 22: Et bilde av spenningskilden som brukes til å drive motoren.

5.2.6 Brukergrensesnitt

Som brukergrensesnitt er det brukt en 16x2 linjer STN overførende LCD-skjerm, tre elektromiske trykknapper og to store mikrobrytere. Figur 23 viser kretsskjemaet for brukergrensesnittet. De sentrale komponentene i brukergrensesnittet er knappene B_1 , B_2 og B_3 , mikrobryterne SW_1 og SW_2 og LCD-skjermen. Funksjonen til komponentene oppsummeres slik:

- B_1 : Knappen brukes til å regulere farten. Hver gang brukeren trykker på knappen, økes farten med 0.1 m s^{-1} .
- B_2 : Knappen brukes til å utføre et kast. Dersom brukeren trykker på knappen vil et kast med valgt fart utføres.
- B_3 : Knappen brukes til å regulere farten. Hver gang brukeren trykker på knappen, minskes farten med 0.1 m s^{-1} .
- SW_1 : Bryteren fungerer som nødstopp. Hensikten er å stoppe prosessen/kastet dersom det kan føre til skade. Bryteren er koblet aktiv lav, dette grunnet ekstra sikkerhet ved en eventuell kortslutning i kretsen eller skade på ledningen til knappen.
- SW_2 : Bryteren definerer kastets utgangsposisjon, og er festet i den posisjonen. Bryteren spiller en stor rolle i den dynamiske basens returbevegelse. Etter et kast vil den dynamiske basen bevege seg tilbake mot utgangsposisjonen. Når den kommer i kontakt med bryteren, blir bryteren slått på. Dette medfører at returbevegelsen blir stoppet opp i samme utgangsposisjon. Slik vil neste kast klargjøres fra samme utgangsposisjon. Bryteren er koblet aktiv lav, dette grunnet ekstra sikkerhet ved en eventuell kortslutning i kretsen eller skade på ledningen til knappen.

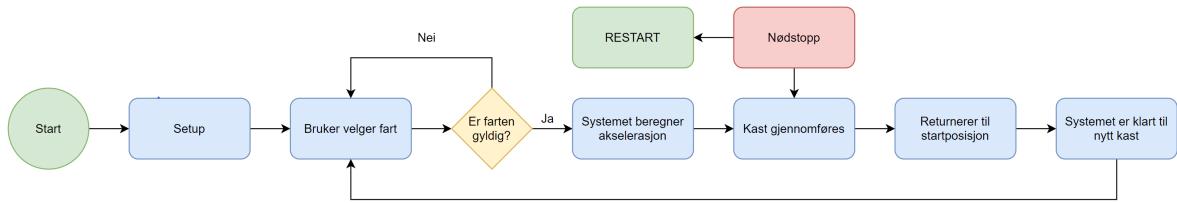


Figur 23: Oppkoblingen av brukergrensesnitt.

5.2.7 Skyvesystemets metodikk

Brukergrensesnittet vil oppføre seg som illustrert i figur 24. Systemet starter opp før bruker får mulighet til å stille inn ønsket fart. Etter ønsket fart er valgt trykker bruker på *Start* og systemet begynner å akselerere curlingsteinen til ønsket fart. Dersom valgt fart er ugyldig vil ikke systemet starte, og bruker vil igjen kunne stille inn fart. Etter kastet er begynt kan bruker, ved hjelp av nødstopknappen avbryte hele prosessen dersom dette er nødvendig. Dette resulterer i at systemet må restartes.

Koden for implementasjon av brukergrensesnittet er tilgjengelig i vedlegg B. Her ser man at bibliotekene *accelstepper*[22], *ezButton*[23] og *LiquidCrystal*[24] brukt for å kontrollere motorenens kontrollsinaler, registrere knappetrykk og bruk av LCD-skjermen.



Figur 24: Skyvesystem metodikk oppsummert.

En sentral variabel i koden er den globale variablene *fart*. *fart* endres ved at brukeren trykker på knappene *B1* og *B3* i brukergrensesnittet. Variablene lagrer ønsket fart på lasten etter input fra bruker. Deklarasjonen av *fart* er vist i figur 25.

Videre brukes variablene *fart* og omkretsen på tannhjulet til å beregne den nødvendige akselerasjonen og rotasjonshastigheten på motoren for å oppnå den ønskede farten. Funksjonene

```
29 float fart; //variable for speed
```

Figur 25: Den globale fartsvariabelen, *fart*.

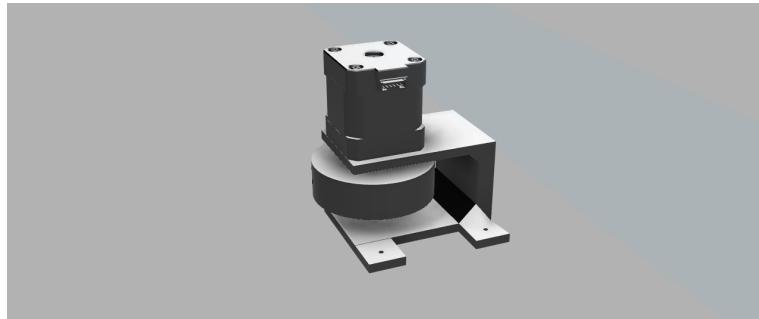
som beregner akselerasjon og rotasjonshastighet vises i figur 26. Funksjonen *calcStepSpeed* tar inn parametrene *v*, som er farten i oppgitt meter per sekund , *circumference*, som er omkretsen på tannhjulet motoren er festet til og *SPR*, som tilsvarer antall steg motoren må ta for å gjennomføre en rotasjon. Funksjonen *calcStepAcceleration* tar inn parameterne *stepSpeed* som motorens hastighet i rotasjoner per sekund og *stepDist* som er antall røtsjoner motoren skal gjøre.

```
35 // Rotasjonshastighet
36 float calcStepSpeed(float v, float circumference, unsigned short SPR){
37     float rps = v/circumference;
38     return (SPR*rps);
39 }
40 // Akselerasjon
41 float calcStepAcceleration(float stepSpeed, float stepDist){
42     return (stepSpeed*stepSpeed/(2*stepDist));
43 }
```

Figur 26: Funksjoner som beregner motorens fart og akselerasjon.

5.3 Implementasjon av rotasjonssystem

Kjernen bak rotasjonssystemet er en trinnmotor av typen Nema 17. Curlingsteinen påføres en curl ved at et hjul, festet til motorens aksel, spinner på steinen. Motoren kontrolleres ved hjelp av en L298N dual H-bridge og en Arduino Uno mikrokontroller. Figur 27 viser bildet av trinnmotoren og lasten den roterer.

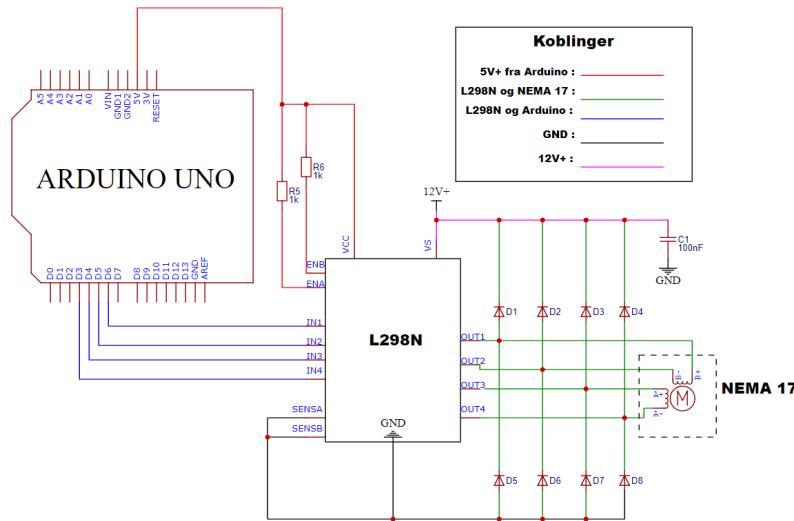


Figur 27: 3D modell av rotasjonsmotoren montert på brakett.

5.3.1 Motoroppkobling

Figur 28 viser kretsskjemaet for oppkoblingen av trinnmotoren med L298N dual-H bridge og mikrokontrolleren. Mikrokontrolleren kontrollerer trinnmotoren ved å benytte L298N som driver for å kunne levere høyere effekt til trinnmotoren. Trinnmotoren og mikrokontrolleren benytter 9 V batterier som spenningskilder.

Trinnmotoren er en bipolar trinnmotor og har fire innganger for å kontrollere rotasjonen. En Arduino Uno mikrokontroller klarer ikke levere høy nok spenning og strøm sammenlignet med det motoren krever. Det er derfor brukt en egenlagd driver med en L298N dual H-brigde som hovedkomponent. På denne måten kan mikrokontrollerens pulser åpne og lukke H-bridgen, koblet til høyere spenning, slik at motoren får levert nok spenning og strøm hver puls.



Figur 28: Kretsskjema for rotasjonssystemet, bygget opp av en L298N Dual H-Bridge.

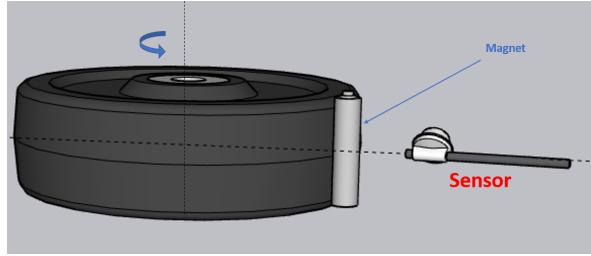
5.3.2 Kontrolling av motor

En Nema 17 trinnmotor trenger 200 steg for å rotere én gang. I motsetning til motoren brukt i skyvesystemet forklart i seksjon A inneholder det ingen innebygd enkoder. Dette gjør at man ikke får tilbakemelding dersom motoren møter mer motstand enn det den klarer å rotere. Koden, tilgjengelig i vedlegg C, er kodet slik at trinnmotoren skal prøve å rotere med en konstant hastighet, valgt i kodden. For å kontrollere motoren brukes Arduinos *Stepper.h* bibliotek [25] for kontrollering av trinnmotorer.

5.4 Implementasjon av sensorsystem

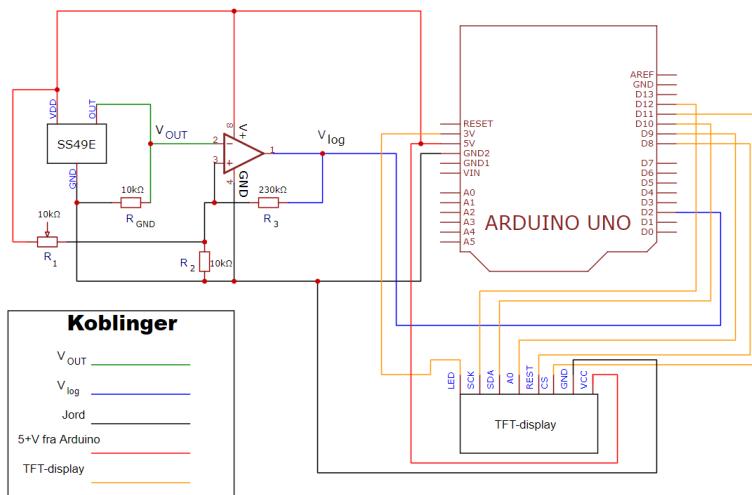
5.4.1 Turteller

For å validere utgangscurlen til curlingsteinen brukes en turteller, bygd opp av en 49E Hall-Effect sensor [26] og fire magneter plassert på et mekanisk hjul som påvirkes av steinens rotasjon. Sensoren er magnetisk sensitiv, og reagerer på endringer i magnetfeltet ettersom hjulet spinner. Denne egenskapen kan brukes til å tallfeste curlen på curlingsteinen. En illustrasjon av denne virkemåten vises i figur 29.



Figur 29: En sensor detekterer rotasjon vha. magnetfelt.

Figur 30 viser kretsskjemaet til turtelleren. Sensoren produserer et analogt signal V_{out} basert på magnetisk feltstyrke. Dersom sensoren ikke detekterer noe magnetisk felt vil $V_{out} = \frac{1}{2}VDD$. Ved $VDD = 5\text{ V}$ vil maksimal og minimal utgangsspenning henholdsvis være $V_{out\max} = 4.2\text{ V}$ og $V_{out\min} = 1.0\text{ V}$. Polarisingen av den magnetiske feltstyrken påført sensoren avgjør om V_{out} stiger eller synker. Det er derfor viktig at alle magnetene ligger med samme polarisering mot sensoren.

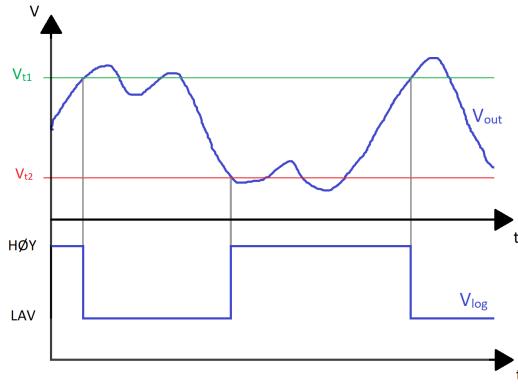


Figur 30: Kretsskjema for turtelleren.

Ved å detektere hver gang en magnet passerer sensorer, kan mengden spin i rpm kalkuleres. Ved å gjøre V_{out} digitalt basert på terskelverdier V_{t1} og V_{t2} , vil verdiene enkelt kunne behandles. Dette oppnås ved å bruke en Schmitt Trigger bygd opp av en operasjonsforsterker. I realiseringen av kretsen brukes LF353P [27]. Det logiske signalet ut av triggeren V_{log} settes LAV når $V_{out} > V_{t1}$ og $HØY$ når $V_{out} < V_{t2}$, som illustrert i figur 31. Positiv og negativ forsyningspenning på operasjonsforsterkeren kobles til henholdsvis 5 V og GND.

Grunnlaget for å bruke en Schmitt Trigger fremfor en vanlig komparator, som bare har én enkelt terskelverdi, er for å unngå at små spenningsendringer rundt terskelspenningen vil gi falske $HØY$ -/ LAV -verdier og kalte overflødige interrupts og gi en unøyaktig måling av rotasjonshastigheten.

Komponentverdiene for Schmitt Triggeren regnes ut på følgende måte [28]. Det er tatt utgangspunkt i følgende fastsatte verdier;



Figur 31: Illustrasjon av funksjonaliteten til Schmitt Triggeren. V_{log} settes HØY når $V_{out} < V_{t2}$ og V_{log} settes LAV når $V_{out} > V_{t1}$.

$$V_{LT} = V_{t2} = 2.6 \text{ V},$$

$$V_{HT} = V_{t1} = 2.7 \text{ V},$$

$$V_{ref} = VDD = 5.0 \text{ V},$$

$$V_{outH} = 5.0 \text{ V},$$

$$V_{outL} = 0 \text{ V},$$

Dette resulterer i;

$$R_1 = 8.85 \text{ k}\Omega,$$

$$R_2 = 10 \text{ k}\Omega,$$

$$R_3 = 230 \text{ k}\Omega.$$

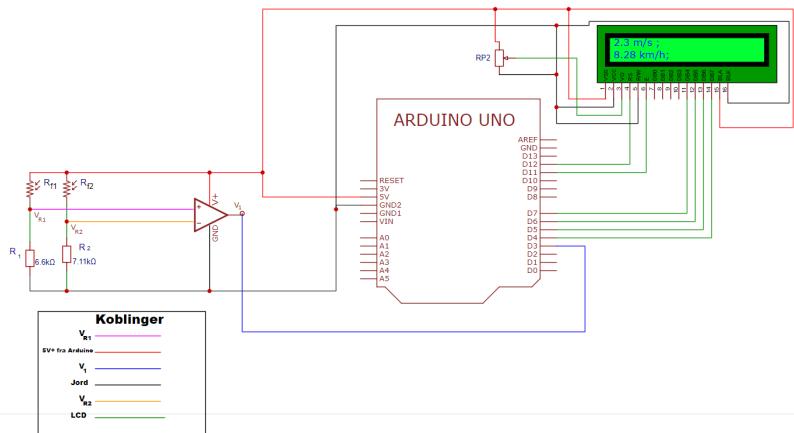
R_1 ble erstattet med et potensiometer på maksimalt $10 \text{ k}\Omega$ for å kunne justere V_{t1} og V_{t2} enkelt.

Beregninger basert på V_{log} gjøres ved hjelp av en Arduino Uno mikrokontroller. Denne leser logisk HØY dersom inngangsspenningen på pin som avleses er over 3 V og logisk LAV dersom inngangsspenningen er under 1.5 V [29]. Forsyningsspenningene 5 V og GND på operasjonsforsterkeren vil derfor være tilstrekkelig.

Koden, tilgjengelig i vedlegg E, baserer seg på interrupts som kalles hver gang en magnet passerer sensoren, og V_{log} settes LAV . Et 1.8TFT display benyttes for å vise brukeren målt og beregnet rotasjonshastighet i rpm. Skjermen kontrolleres ved hjelp av Arduino bibliotek fra Adafruit [30], [31].

5.4.2 Fartssensor

Fartssensoren er realisert ved å bruke to fotoresistorer R_{f1} og R_{f2} av typen GL55[32] sammen med laserdioden nevnt i seksjon 4.4.2. Fotoresistorene tar inn lys fra omgivelsene, og er satt i serie med motstandene R_1 og R_2 som vist i figur 32. Laserdioden er sentrert inn på R_{f1} . Motstanden R_{f1} synker når laseren er tilstede slik at spenningen over R_1 , V_{R1} øker. Når laseren er brutt vil R_{f1} øke slik at spenningen V_{R1} synker.



Figur 32: Kretskjema for fartssensoren.

Signalet digitaliseres ved å bruke en komparator basert på en LF353 operasjonsforsterker [27]. Utgangsspenningen V_1 settes $HØY$ når laseren er tilstede på R_{f1} og LAV når laseren er brutt. Som terskelspanning brukes V_{R2} da R_{f2} leser av lyset i omgivelsene. Ved å sette R_2 større enn R_1 vil $V_{R1} < V_{R2}$ når laseren er brutt, dette fordi V_{R1} stiger når laseren er tilstede på R_{f1} . Dette tilsvarer at $V_1 = LAV$ når laseren er brutt og $V_1 = HØY$ når laseren er tilstede.

Realiserte motstandsverdier i systemet er;

$$R_1 = 6.6 \text{ k}\Omega. \quad (7)$$

$$R_2 = 7.11 \text{ k}\Omega. \quad (8)$$

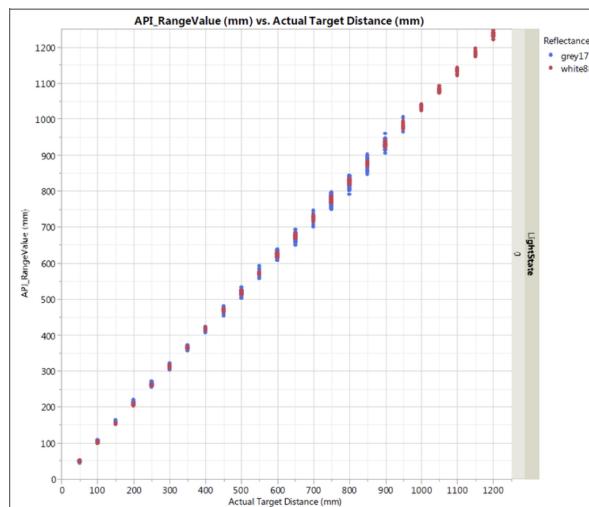
Positiv og negativ forsyningsspenning på operasjonsforsterkeren er koblet til 5 V og GND på mikrokontrolleren.

Koden tilgjengelig i vedlegg F brukes til å måle tidsintervallet der $V_1 = LAV$. Mikrokontrolleren beregner da hvor lenge curlingsteinen bryter laserlyset på R_{f1} . Ved å bruke curlingsteinens diameter regner mikrokontrolleren ut steinens fart.

Koden baserer seg på interrupts som kalles hver gang V_1 skifter mellom $HØY/LAV$. Bruker-grensesnittet består av en LCD1602 LCD-skjerm for å vise beregnet utgangsfart. Her benyttes Arduinos *LiquidCrystal* bibliotek [33] for å oppdatere skjermen.

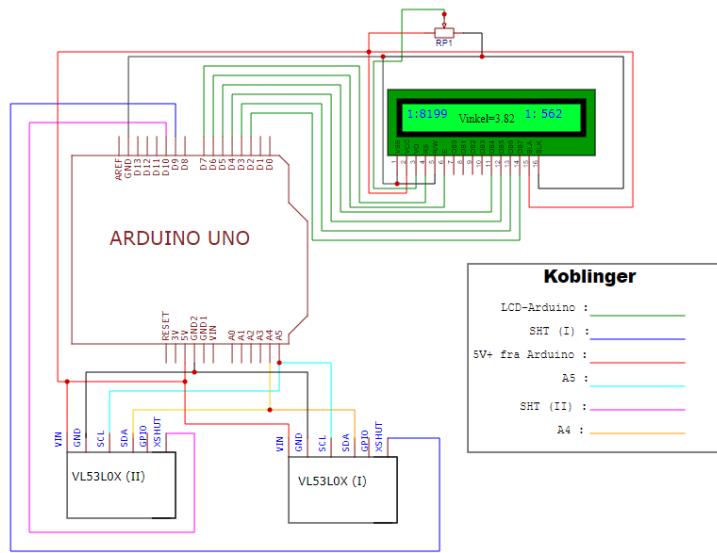
5.4.3 Vinkelsensor

For å sikre at utgangsvinkelen er lik ved hvert kast brukes to VL53L0X Time of Flight avstandssensorer [34] som vist i vedlegg D.2. Sensorene bruker en laserkilde for å måle avstanden ved å beregne hvor lang tid laseren bruker på å reflekteres tilbake til sensoren. VL53L0X ble valgt da den oppfyller systemkravet for vinkelsensor beskrevet i tabell 3. Dette ved å sørge for at fargen på objektet laseren lyser på, og avstanden mellom sensoren og objektet er innenfor avviket presentert i figur 33. Utgangsvinkelen presenteres for bruker på en LCD-skjerm.



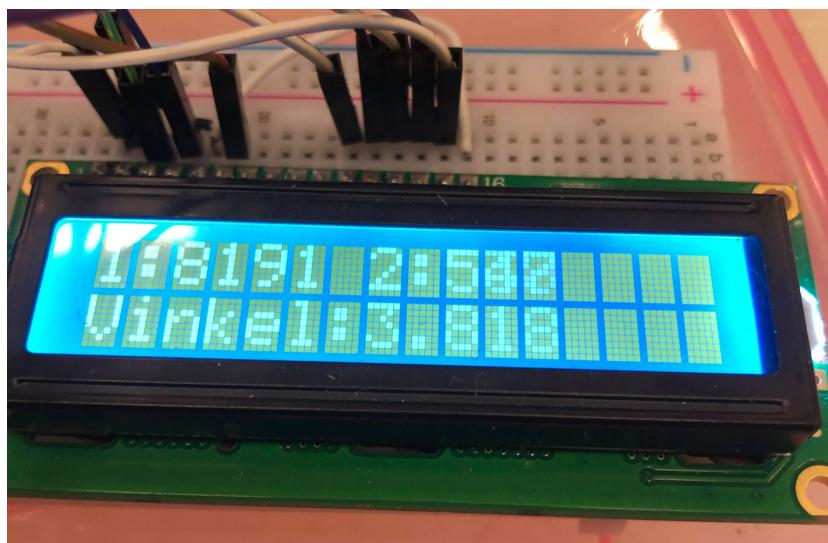
Figur 33: Avviket til sensorene sine målinger ved ulike avstander og farge [34].

Sensorene kobles til en Arduino Uno mikrokontroller som vist i figur 34. Ettersom sensorene kommuniserer med mikrokontrolleren gjennom en I^2C buss må man sørge for at sensorene ikke sender til samme minneadresse i mikrokontrolleren. Dette gjøres ved å koble sensorene til mikrokontrolleren for å så sette $XSHUT$ høy på én og én sensor mens man setter minneadressen til den andre.



Figur 34: Kretskjema for vinkelsensor, bestående av to ToF avstandssensorer.

Koden tilgjengelig i vedlegg F tar kontinuerlige målinger, frem til den første sensoren detekterer curlingsteinen ved at den gjør en måling under 400 mm. Etter steinen detekteres gjør den første sensoren et gitt antall målinger som lagres i en liste. Like etter gjør den andre sensoren det samme. Utgangsvinkelen beregnes ut ifra den korteste målingen fra begge listene. Slik vet man at det er samme punkt på steinen som er målt. Målingen vises deretter på en LCD-skjerm til bruker, som vist i figur 35. Koden er basert på eksempelkoden “vl53l0x_dual” [35], som er hentet fra “Adafruit Industries” sitt Arduinobilbibliotek for avstandssensorene.



Figur 35: Målingene til avstandssensorene og utgangsvinkelen regnet ut av mikrokontrolleren, vist på display.

6 Verifikasjon

6.1 Verifikasjonsplan

Systemet er verifisert med utgangspunkt i systemkravene presentert i seksjon 4.5. Tabell 4 viser verifikasjonsplanen for test av systemet og delsystemene.

Tabell 4: Plan for verifikasjonstester.

Nr.	Delsystem	Test	Modul
6.2.1	Fysisk utforming	Størrelsesmåling	Rammer
6.2.2	Fysisk utforming	Mekanisk robusthet	Alle bevegelige deler
6.2.3	Skyvesystem	Mekatronisk presisjon	Trinnmotor (Nema 24)
6.2.4	Skyvesystem	Maksimal test	Trinnmotor (Nema 24)
6.2.5	Skyvesystem	Brukergrensesnitt	Brukergrensesnitt
6.2.6	Rotasjonssystem	Mekatronisk styrke	Trinnmotor (Nema 17)
6.2.7	Sensorsystem	Presisjon med videoanalyse	Turteller
6.2.8	Sensorsystem	Presisjon med signalanalyse	Turteller
6.2.9	Sensorsystem	Presisjon med videoanalyse	Fartssensor
6.2.10	Sensorsystem	Presisjon med signalanalyse	Fartssensor
6.2.11	Sensorsystem	Presisjonstest	Vinkelsensor

6.2 Kort beskrivelse av planlagte tester

6.2.1 Fysisk utforming: Størrelsesmåling

Tester om kravet for størrelse presentert i tabell 3 oppfylles. Testen er gjennomført ved å måle den fysiske utformingen av systemet som en helhet.

6.2.2 Fysisk utforming: Test av mekanisk robusthet

Tester om kravet for fysisk robusthet presentert i tabell 3 oppfylles. Testen er gjennomført ved gjentatte gjennomføringer av systemet med realistisk motstand for å se om de mekaniske komponentene tåler påkjenningen de utsættes for.

6.2.3 Skyvesystem: Mekatronisk presisjonstest

Tester om kravet for presisjon presentert i tabell 3 oppfylles. Testen er gjennomført ved å kjøre systemet med og uten last på is, og uten last på flisunderlag. Strekningen som måles er distansen steinen har akselerert.

6.2.4 Skyvesystem: Maksimal fartstest

Tester om kravet for kraft presentert i tabell 3 oppfylles. Testen er gjennomført ved å utføre et kast og loggføre farten.

6.2.5 Skyvesystem: Test av brukergrensesnittets funksjonalitet

Tester om kravet for brukervennlighet i brukergrensesnittet presentert i tabell 3 oppfylles. Testen er gjennomført ved å justere farten og starte systemet ved hjelp av brukergrensesnittet.

6.2.6 Rotasjonssystem: Maksimal rotasjonstest

Tester om kravet for kraft i rotasjonsmotoren presentert i tabell 3 oppfylles. Testen er gjennomført ved å rotere objekter med varierende masse til ønsket rotasjonshastighet.

6.2.7 Turteller: Presisjonstest med videoanalyse

Tester om kravet for presisjon i turtelleren presentert i tabell 3 oppfylles. Testen er gjennomført ved å ta et videooppdrag av hjulet med magnetene når det roterer rundt Hall-Effect sensoren. Rotasjonshastigheten rpm_{reell} beregnes ved å finne tidsintervallet Δt_{video} mellom to påfølgende frames der samme magnet på hjulet befinner seg under Hall-Effect sensoren. Dette tilsvarer en hel rotasjon. Basert på Δt_{video} og bildefrekvensen, fps, på videoen beregnes det et intervall rpm_{reell} må ligge innenfor:

$$\frac{60}{\Delta t_{video} - \frac{1}{fps}} < RPM_{reell} < \frac{60}{\Delta t_{video} + \frac{1}{fps}} \quad (9)$$

Det stilles krav til at $RPM_{turteller}$ befinner seg innenfor intervallet til RPM_{reell} når $fps = 30$. Dette gir også et maksimale avvik for $RPM_{turteller}$.

6.2.8 Turteller: Presisjonstest med signalanalyse

Tester om kravet for presisjon i turtelleren presentert i tabell 3 oppfylles. Testen er gjennomført ved å måle forsinkelsestid over spenningene V_{log} , V_{out} , V_{t1} og V_{t2} gjennom et oscilloskop.

For at kretsen skal være logisk riktig må følgende krav oppfylles:

1. $V_{t1} > V_{t2}$.
2. V_{out} fluktuerer komfortabelt over V_{t1} og under V_{t2} når det mekaniske hjulet spinner og det magnetiske feltet påført sensoren endres.

3. V_{log} settes HØY når $V_{out} < V_{t2}$
4. V_{log} settes LAV når $V_{out} > V_{t1}$
5. V_{log} oppnår spenningsnivå som oppfyller arduinoens krav for å lese HØY/LAV [29]. Det vil si:
 - $V_{log} > 3.0 \text{ V} \Rightarrow V_{log} = \text{HØY}$
 - $V_{log} < 1.5 \text{ V} \Rightarrow V_{log} = \text{LAV}$

Forsinkelsestiden i kretsen Δt_f måles som den maksimale forskjellen i forsinkelse fra V_{out} stiger over V_{t1} til V_{log} går LAV (som utløser interrupts på arduinoen) basert på 10 målinger. Fra systemkravene settes det som krav at målt Δt_f påvirker målt rotasjonshastighet $rpm_{målt}$ med mindre enn $\pm 2.5\%$. For å regne ut avviket brukes følgende formler for å regne ut den antatt reelle rotasjonshastighet rpm_{reell} og $rpm_{målt}$. Δt er den reelle tiden hjulet bruker på en rotasjon:

$$RPM_{reell} = \frac{60}{\Delta t} \quad (10)$$

$$RPM_{målt} = \frac{60}{\Delta t + \Delta t_f} \quad (11)$$

6.2.9 Fartssensor: Presisjonstest med videoanalyse

Tester om kravet for presisjon i fartssensoren presentert i tabell 3 oppfylles. Testen er gjennomført ved å analysere et videooppdrag av curlingsteinen i utgangsøyeblikket. Tidsintervallet Δt_{video} måles fra første frame laseren er brutt til første frame laseren ikke er brutt. Basert på Δt_{video} , diameteren på steinen, D, og bildefrekvensen på videoen, fps, vil man finne et intervall den reelle farten v_{reell} må ligge innenfor:

$$\frac{D}{\Delta t_{video} + \frac{1}{fps}} < v_{reell} < \frac{D}{\Delta t_{video} - \frac{1}{fps}} \quad (12)$$

Det settes som krav at farten målt av sensoren v_{sensor} ligger innenfor dette intervallet når videoen har $fps = 30$ og steinen passerer sensoren med en fart på 1 m s^{-1} . Intervallet for v_{reell} vil også gi maksimalt avvik for v_{sensor} ved denne farten basert på videoens fps.

6.2.10 Fartssensor: Presisjonstest med signalanalyse

Tester om kravet for presisjon i fartssensoren presentert i tabell 3 er oppfylt. Testen er gjennomført ved å måle forsinkelsestid av spenningene V_1 , V_{R1} og V_{R2} gjennom et oscilloskop.

Følgende krav må være oppfylt for at kretsen skal oppføre seg logisk riktig:

1. $V_{R1} > V_{R2}$ når laserlyset er tilstede på R_{f1} .
2. $V_{R1} < V_{R2}$ når laserlyset ikke er tilstede på R_{f1} .
3. $V_1 = HØY$ når $V_{R1} > V_{R2}$
4. $V_1 = LAV$ når $V_{R1} < V_{R2}$
5. V_1 oppnår spenningsnivå som oppfyller arduinoens krav for å lese HØY/LAV [29]. Det vil si:
 - $V_1 > 3.0 \text{ V} \Rightarrow V_1 = HØY$
 - $V_1 < 1.5 \text{ V} \Rightarrow V_1 = LAV$

Forsinkelsestiden t_f i kretsen måles i form av tiden mellom spenningsfallet/stigningen over R_1 skjer til V_1 når henholdsvis 1.5 V (LAV) og 3.0 V (HØY). Fra systemkravene settes det som krav at t_f påvirker beregnet målt fart $v_{målt}$ med mindre enn $\pm 2.5\%$. For å kunne regne ut avviket brukes følgende formler for å regne ut den antatt reelle farten v_{reell} og $v_{målt}$. Δt er den reelle tiden steinen bryter laseren, og D er steinens diameter:

$$v_{reell} = \frac{D}{\Delta t} \quad (13)$$

$$v_{målt} = \frac{D}{\Delta t + t_f} \quad (14)$$

6.2.11 Presisjonstest av vinkelsensorer.

Tester om kravet for presisjon i vinkelsensoren presentert i tabell 3 oppfylles. Testen er gjennomført ved å sende en curlingstein med fart 1.4 m s^{-1} , uten curl, gjentatte ganger.

6.3 Tester av Fysisk utforming

6.3.1 Størrelsesmåling

Systemet etter måling passer innenfor størrelseskravene, og måler dimensjoner på $2 \text{ m} \cdot 0.91 \text{ m} \cdot 0.2 \text{ m}$.

6.3.2 Mekanisk robusthet

Etter gjentatte gjennomføringer av systemet ble de 3D-printede tannhjulene ødelagt. Slitasjen på tannhjulet skyldes manglende robusthet i materialet den er laget av, slik at det høye dreiemomentet til motoren slet gjennom festemekanismen mellom motor og tannhjul ved

større last. Dette fører til ustabile og manglede resultater i mange av testene der funksjonalitet og krav til skyvesystemet testes.

Testen konkluderer med at materialet PLA ikke er passende for produksjon av tannhjulene i systemet.

6.4 Tester av Skyvesystem

6.4.1 Maksimal hastighetstest

Testen er gjennomført til tross for *ikke bestått* for kravene om mekanisk robusthet.

Etter gjentatte forsøk på å akselerere curlingsteinen opp til farten 2.5 m s^{-1} er kravet ikke bestått. Dette grunnet høyere statisk friksjon mellom curlingsteinen og isen enn antatt i den teoretiske beregningen i seksjon 5.2.1, og kortere akselerasjonsdistanse grunnet det ødelagte tannhjulet.

6.4.2 Mekatronisk presisjonstest

Etter gjentatte gjennomføringer viser tabell 5 gjennomsnittlig avvik og typisk avvik fra den ønskelige distansen på 1.5 m. Resultatene fra testen under de ulike forholdene nevnt i seksjon 6.2.4 vises i vedlegg A.5.5.

Tabell 5: Typetall og gjennomsnitt ble beregnet basert på målingene fra testene som ble gjennomført. Disse målingene ligger i seksjon A.5.5.

<i>Metode Verdi</i>	Flisgolv uten last	Curlingbane uten Last	Curlingbane med last (5kg)	Curlingbane med last (20kg)
Gj.snitt avvik [cm]	0.078 ± 0.022	7.141 ± 1.619	14.81 ± 0.6266	102.27 ± 7.28
Typetall av avvik målinger [cm]	0.09	6.5	14.9	100.3

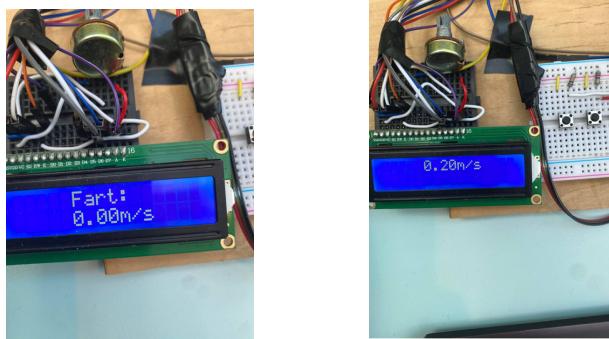
Fra tabell 5 ser man at systemet har lite avvik under testene på flisgolv. Dette er antatt grunnet at tannhjulet var inntakt under testingen. Avviket man observerer fremkommer fra det elastiske fiskesnøret som trekker den dynamiske basen. Avvik fra andre årsaker som dårlige måleinstrumenter og egenskaper ved observatør må også tas med i resultatet.

Målingene gjennomført på curlingbanen har større gjennomsnittlig avvik. Dette kommer av det ødelagte tannhjulet som forklart i seksjon 6.3.2.

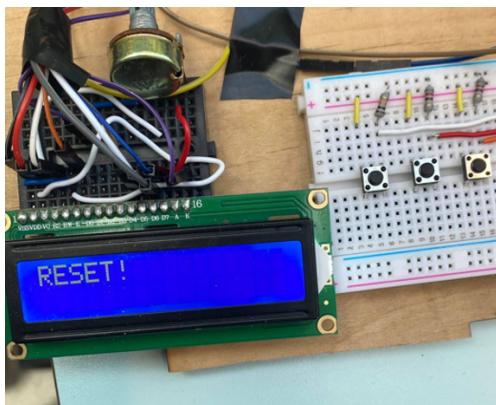
Videre observeres det at større last i systemet gir større avvik fra ønsket verdi. En forklaring på dette kan være det ødelagte tannhjulet. Et dårlig festet mellom tannhjulet og motoren fører til at motoren ikke roterer like mange ganger som tannhjulet. Dette gir en uønsket bevegelsebane. Testen erklares *ikke bestått*.

6.4.3 Test av brukergrensesnittets funksjonalitet

Alle komponentene fungerte optimalt og oppfylte sin funksjon. Brukeren har muligheten til å regulere farten på systemet, og det er mulig å stoppe gjennomføringen av kastet ved hjelp av nødstoppbryteren. Figur 36 viser LCD-skjermens respons når forskjellige knapper har blitt trykket på.



(a) Det skjermen viser før et kast.
(b) Brukeren har trykket på knappen $B1$ to ganger. Dermed har farten økt med 0.2 m s^{-1} .



(c) Nødstoppbryteren $SW1$ har blitt brukt. Systemet har blitt stoppet.

Figur 36: Virkemåten av skjermen.

6.5 Tester av rotasjonssystem

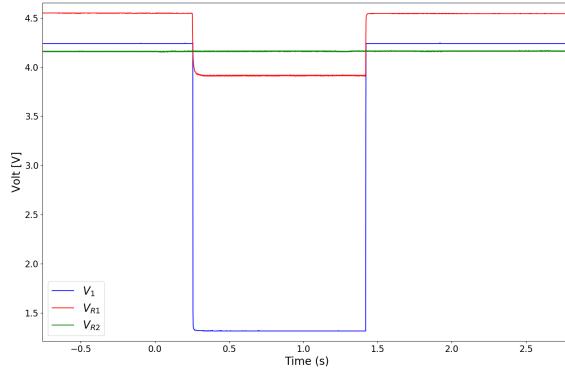
6.5.1 Maksimal rotasjonstest

Rotasjonsmotoren kan, med initiell hjelp, rotere objekter av mindre masse. Motoren er derimot for svak til å både starte og opprettholde rotasjon i en curlingstein.

6.6 Tester av Sensorersystem

6.6.1 Fartssensor: Presisjonstest med signalanalyse

Målingen i figur 37 viser signalene i kretsen i det et objekt med lengde lik curlingsteinens diameter ($D = 29$ cm) sendes forbi laserbarriieren.

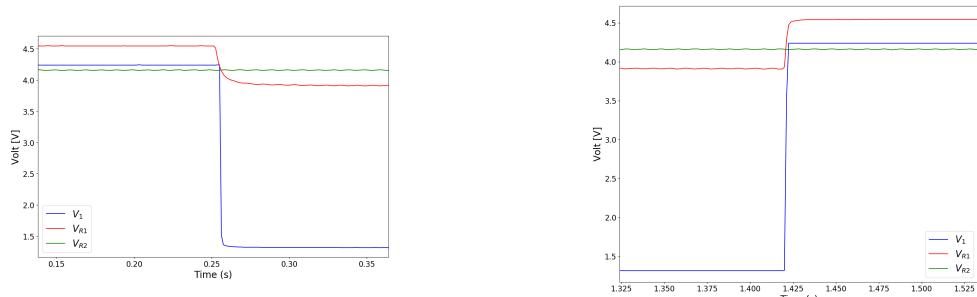


Figur 37: Måling av V_1 , V_{R1} og V_{R2} . V_{R1} går under V_{R2} når laseren ikke er tilstede på R_{f1} og V_1 settes lav.

Når laserlyset er tilstede på R_{f1} er $V_{R1} > V_{R2}$ og når laserlyset er blokkert er $V_{R1} < V_{R2}$. Videre er $V_1 = HØY$ når $V_{R1} > V_{R2}$ og $V_1 = LAV$ når $V_{R1} < V_{R2}$. De fire første kriteriene er dermed oppfylt.

Ved $V_1 = LAV$ er spenningen rundt 1.32 V og ved $V_1 = HØY$ er spenningen rundt 4.24 V. Det vil si at testen oppfyller kravet om henholdsvis <1.5 V og >3 V.

En nærmere illustrasjon av signalene i det V_1 går HØY/LAV illustreres i figur 38.



(a) Det er noe forsinkelse fra V_{R1} går under V_{R2} til V_1 går LAV.
(b) Det er tilnærmet ingen forsinkelse fra V_{R1} begynner å stige til V_1 settes HØY.

Figur 38: Nærmere illustrasjon av signalene i det V_1 går HØY/LAV.

Målingen viser at det er en liten forsinkelse fra V_{R1} starter å minke til V_1 settes LAV på omkring 5 ms. På stigende flanke er det en forsinkelse på omkring 2 ms fra V_{R1} begynner å

øke til V_1 settes HØY. Totalt vil dette påvirke perioden mikrokontrolleren leser LAV Δt med $t_f = -3$ ms. Dette vil påvirke resultatet i størst grad når farten på curlingsteinen er høy. Det er lite sannsynlig at en utgangsfart på mer enn 2.5 m s^{-1} er relevant for systemet ettersom denne farten medfører at curlingsteinen vil krysse tilnærmet hele banen. Ved denne farten vil $t_f = -0.003$ s føre til følgende v_{reell} og $v_{målt}$ ved hjelp av likning 13 og 14:

$$v_{reell} = \frac{0.29 \text{ m}}{0.116 \text{ s}} = 2.5 \text{ m s}^{-1}$$

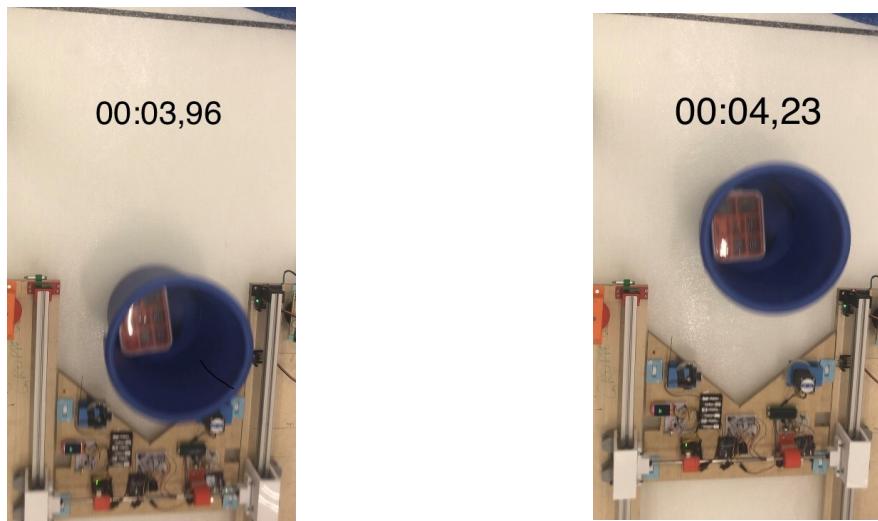
$$v_{målt} = \frac{0.29 \text{ m}}{0.116 \text{ s} - 0.003 \text{ s}} = 2.57 \text{ m s}^{-1}$$

Dette resulterer i et avvik på 0.7 m s^{-1} ved $v_{reell} = 2.5 \text{ m s}^{-1}$. Dette tilsvarer et avvik på 2.8% som er så vidt utenfor kravet på $\pm 2.5\%$. Mesteparten av forsinkelsen og dermed avviket forårsakes trolig av fotoresistoren R_{f1} som har en maksimal forsinkelse på 10ms fra lysendringene skjer, til motstanden i komponenten endres [36].

Verdt å nevne er at denne sensoren i hovedsak er designet for å validere at steinen oppnår lik utgangsfart ved hver utskyting med samme fartsinnstilling. Ved lik fartsinnstilling og dermed tilnærmet lik reell utgangsfart vil også avviket være tilnærmet det samme. I målingene som skal sammenlignes vil derfor avviket være så godt som ubetydelig for hensikten sensoren har i systemet.

6.6.2 Farttssensor: Presisjonstest med videoanalyse

Videomålingen ble tatt med bildefrekvens på 30 fps og er illustrert i figur 39.



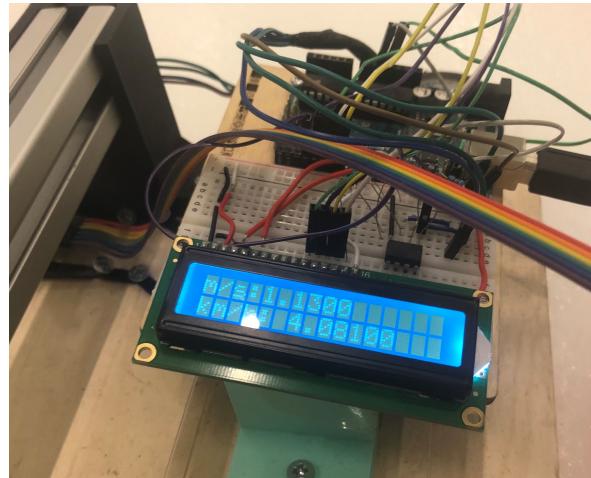
(a) Første frame etter laseren er brutt. Δt_{video} beregnes fra 00:03,96. (b) Første frame etter laseren ikke lenger er brutt. Δt_{video} beregnes frem til 00:04,23.

Figur 39: Første frame etter laseren er brutt og første frame etter laseren ikke lenger er brutt. Objektet har tilnærmet lik radius som steinen uten at dette er av stor betydning.

Videoen viste at $\Delta t_{video} = 0.27$ s. Diameteren for en curlingstein brukes, $D = 0.29$ m. Dette førte til følgende intervall for v_{reell} ved hjelp av likning 12.

$$0.957 \text{ m s}^{-1} < v_{reell} < 1.224 \text{ m s}^{-1}$$

Figur 40 viser målt v_{sensor} .

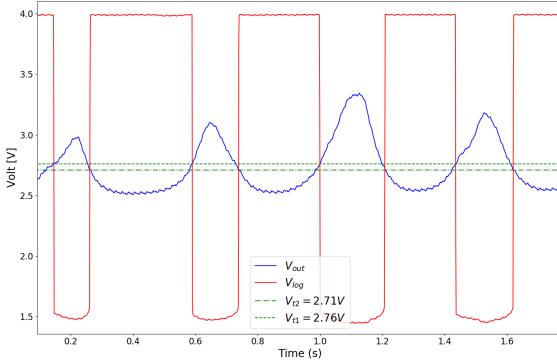


Figur 40: Farten regnet ut av mikrokontrolleren vist på display.

Displayet viser at $v_{sensor} = 1.13 \text{ m s}^{-1}$, noe som er godt innenfor intervallet til v_{reell} og indikerer at v_{sensor} er tilnærmet lik v_{reell} . Avviket til v_{sensor} basert på intervallet til v_{reell} er derfor minimalt 0 m s^{-1} og maksimalt 0.173 m s^{-1} ved denne farten. 0.173 m s^{-1} tilsvarer et avvik på 18%. Sannsynligvis er dette avviket mye mindre ettersom dette er basert på et videoklipp med 30fps som gir et nokså brent intervall for v_{reell} . Dette underbygges også av signaltesten. Denne testen blir derfor en grov test for å sjekke om v_{sensor} kan stemme overens med v_{reell} , noe den gjør.

6.6.3 Turteller : Presisjonstest med signalanalyse

Målinger av signalene i systemet er illustrert i figur 41.



Figur 41: Måling av signalene V_{out} og V_{log} . Basert på signalenes oppførsel kan det fastslås at $V_{t2} \approx 2.71$ og $V_{t1} \approx 2.76$.

Målingene viste at $V_{t1} > V_{t2}$ og V_{out} fluktuerer komfortabelt over/under V_{t1} og V_{t2} .

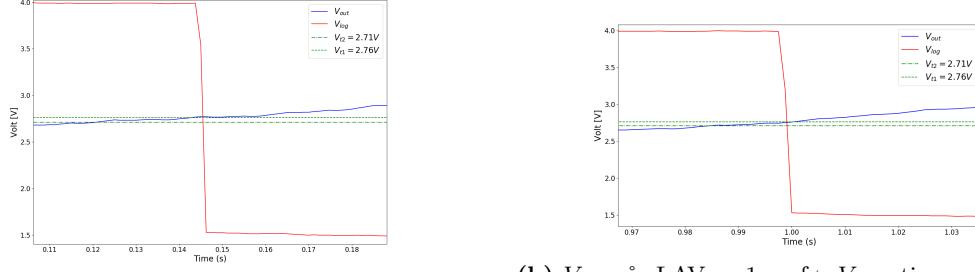
V_{log} settes HØY når $V_{out} < V_{t2}$ og LAV når $V_{out} > V_{t1}$.

Ved $V_{log} = HØY$ er spenningen ≈ 4.0 V og ved $V_{log} = LAV$ er spenningen ≈ 1.5 V. Det ser ut som nivået hvor arduinoen leser LAV reelt er litt høyere enn 1.5 V ettersom interrupts ble kalt hver gang V_{log} gikk LAV. Ideelt kunne spenningen på V_{log} nådd lavere, men da måtte negativ spenning vært brukt på negativ forsyningsspenning på opampen.

Basert på 10 prøver viser det seg at maksimal forskjell i forsinkelse fra V_{out} stiger over V_{t1} til V_{log} går LAV (som utløser interrupts på arduinoen) er $\Delta t_f = 5$ ms. Dette er illustrert i figur 42 og vil påvirke resultatet i størst grad når rotasjonshastigheten er høy. I curling vil stort sett alle kast føre til en rotasjonshastighet på mellom 2rpm og 30rpm på steinen. En rotasjonshastighet på 30rpm på steinen tilsvarer 124rpm på det mekaniske hjulet som tilfører steinen rotasjon. Dette gir følgende RPM_{reell} og $RPM_{målt}$ ved bruk av likning 10 og 11:

$$RPM_{reell} = \frac{60}{0.484\text{ s}} = 123.97\text{ rpm}$$

$$RPM_{målt} = \frac{60}{0.484\text{ s} + 0.005\text{ s}} = 122.70\text{ rpm}$$



(a) V_{log} går LAV ca 4 ms etter V_{out} stiger over V_{t1} .

(b) V_{log} går LAV ca 1 ms før V_{out} stiger over V_{t1} .

Figur 42: Nærmore illustrasjon av signalene i det V_{log} går LAV. Basert på ti tilfeller hvor V_{log} gikk LAV var dette de to tilfellene som ga størst Δt_f .

Dette fører til et avvik på -1.27rpm som tilvarer -1.0% . Dette oppfyller kravet på $\pm 2.5\%$.

6.6.4 Turteller: Presisjonstest med videoanalyse

Videomålingen ble tatt med bildefrekvens på 30 fps og er illustrert i figur 43.



(a) Målingen av Δt_{video} starter (b) Samme magnet befinner seg (c) Skjermen oppdateres før en i dette bildet hvor en magnet er under hall-effekt-sensoren og hju- ny magnet passerer og skjermen lett har fullført tilnærmet en rota- viser systemets utregnede rotasjon. Målingen av Δt_{video} slutter sjonshastighet i rpm. her.

Figur 43: Videotest av turteller. Hjulet fullfører en hel rotasjon og tidsintervallet på filmen brukes til å sammenligne opp mot systemets utregninger.

Basert på videoen er $\Delta t_{video} = 1.30$ s. Dette gir følgende intervall for rpm_{reell} ved bruk av likning 9:

$$45.11 < RPM_{rell} < 47.24.$$

Fra figur 43 ser vi at $rpm_{arduino} = 45.73$ som er innenfor det nokså lille intervallet for rpm_{reell} . Dette tilsier at $rpm_{arduino}$ er nokså treffsikker med et avvik på mellom minimum 0rpm og maksimum -1.51rpm basert på denne testen. -1.51rpm tilsvarer et avvik på -3.2% . Sannsynligvis er avviket enda mindre ettersom testen baserer seg på et videoklipp med 30fps. Dette betyr at avviket høyst sannsynlig er innenfor systemkravet på $\pm 2.5\%$.

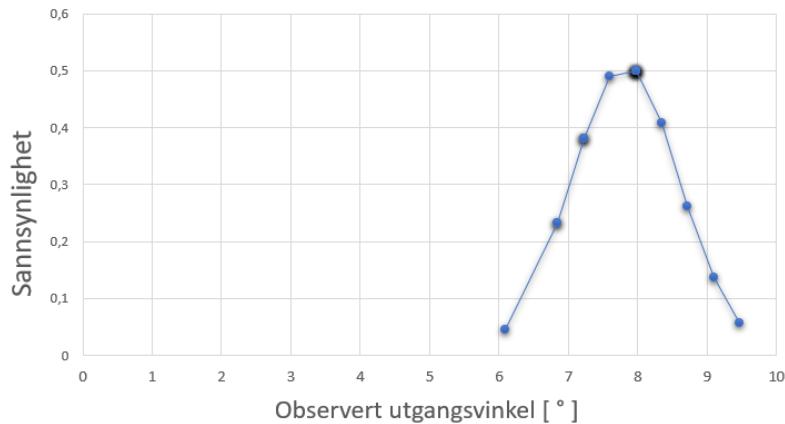
6.6.5 Vinkelsensor: Presisjonstest

Etter gjentatte forsøk med fart 1.4 m s^{-1} er det observert at den fremste sensoren, sensor 1, bare gir nøyaktige målinger på avstander mellom 50 mm og 150 mm. Den siste sensoren, sensor 2, er defekt etter transport av systemet, som fører til feilberegninger. Dette resulterer i at det ikke blir gjort like mange målinger som ønsket etter at sensor 1 detekterer steinen, som forklart i seksjon 5.4.3.

Målet med testen er å sjekke at vinkelsensoren er presis nok og holder kravet, som er et målingsavvik på $\pm 3^\circ$. Hensikten er at utgangsvinkelen skal være den samme ved to like kast. Derfor er det viktig at kravet blir oppfylt for å kunne sammenligne kastene. I teorien skulle utgangsvinkelen bli 0° , men muligheten for at systemet ikke lå helt rett og at sensorene ikke ble montert helt rett i forhold til hverandre kan ha påvirket målingene. Dette har egentlig ikke noe si, siden poenget er at utgangsvinkelen skal bli lik ved hver måling, hvertfall innenfor kravet. Målingene som ble gjort vises i tabell 6. I tillegg ble disse målingene plottet med utgangspunkt i sannsynligheten for de ulike verdiene. Dette vises i figur 44.

Tabell 6: Resultatene fra verifikasjonstesten til vinkelsensoren.

Måling	Utgangsvinkel [°]
1	6.09
2	7.22
3	8.72
4	7.6
5	7.97
6	7.97
7	6.84
8	7.97
9	7.22
10	8.34
11	7.97
12	6.84
13	7.97
14	7.97
15	9.46
16	7.97
17	7.22
18	9.09
19	7.97
20	7.97



Figur 44: Illustrasjon av sannsynligheten av de ulike målingene fra tabell 6.

Under målingene ble det sjekket hvilke avstandsverdier sensorene ga. Når det ble oppdaget at den ødelagte sensoren ga feil verdier, ble ikke denne målingen brukt.

For å finne ut hvor god vinkelsensoren er i forhold til kravet antas det at den reelle utgangsvinkelen er typetallet på 7.97° . Dette med god grunn siden avstanden er den samme hver gang, og differansen hos sensorene, som brukes til utregning, vil oftest være riktig. Dermed

brukes største og minste måling for å finne målingsavviket. Disse målingene sjekkes opp mot typetallet for å finne målingsavviket. Målingsavviket er oppgitt i tabell 7.

Tabell 7: Typetall observert fra test, som er den reelle utgangsvinkelen, med målingsavvik.

Verdi	Utgangsvinkel med avvik [°]
Typetall av målingene og målingsavvik	7.97 ± 1.88

Ut ifra disse målingene kan det konkluderes med at vinkelsensoren er innenfor presisjonskravet. Differansen mellom den største målingen og typetallet er bare 1.12. Likevel må den største differansen brukes, siden målingsavviket kan bli like stort begge veier. Målingsavviket på $\pm 1.88^\circ$ kan også forbedres ytterligere med flere målinger ved hver sensor.

6.7 Test- og systemkrav-vurdering

Tabell 8 viser testvurderingen av testene som ble gjennomført i seksjon 6.3-6.6.

Tabell 8: Vurdering av tester og systemkrav.

Test Nr.	Delsystem	Modul	Systemkrav	Vurdering	Kommentar
6.2.1	Fysisk utforming	Rammer	Størrelse	Bestått	Størrelsen på systemet passet innenfor kravene.
6.2.2	Fysisk utforming	Matrialer	Fysisk robusthet	Ikke bestått	Tannhjul som er koblet til motoren er i materialelet PLA, og tålte ikke påkjenningene da lasten og akselerasjonen ble for stor.
6.2.3	Skyvesystem	Motor	Kraft	Ikke bestått	Motoren greide ikke å akselerere en curlingstein opptil en fart på 2.5 m s^{-1} . Årsakene er det ødelagte tannhjulet og underestimering av den statiske friksjonen en curlingstein opplever.

6.2.4	Skyvesystem	Motor	Presisjon	Delvis bestått	Utgangsfarten ble ikke oppnådd etter 1.5 m når systemet ble belastet med en curlingstein og en last på 5 kg. Tabell 5 viser gjennomsnittlig avvik med forskjellige type last og underlag.
6.2.5	Skyvesystem	Brukergrensesnitt	Brukervennlighet	Bestått	Brukeren har muligheten til å regulere farten, og brukeren har tilgang på nødstopp.
6.2.6	Rotasjonssystem	Motor	Kraft	Ikke bestått	Klarte ved lav rotasjonshastighet å rotere curlingsteinen hvis den ble dyttet i bevegelse, og fikk litt hjelp. Statisk friksjon var for stor.
6.2.9	Sensorsystem	Fartssensor	Presisjon	Ingen konklusjon	Videoklippet viste at v_{sensor} lå innenfor intervallet til v_{reell} . Basert på dette kan det fastslås at avviket på v_{sensor} er et sted mellom 0% og $\pm 18\%$. Testen kan derfor ikke brukes til å avkrefte eller bekrefte systemkravet på maksimalt $\pm 2.5\%$ avvik.
6.2.10	Sensorsystem	Fartssensor	Presisjon	Ikke bestått	Basert på målinger gjennomført med oscilloskop av signalene i kretsen kan det fastslås at kretsen fungerer logisk som den skal. Forsinkelsen i kretsen t_f ble målt til -3 ms . Dette tilsvarer et avvik på 2.8% ved en reell fart på 2.5 m s^{-1} . Dette er så vidt utenfor systemkravet på $\pm 2.5\%$.

6.2.7	Sensorsystem	Turteller	Presisjon	Ingen konklusjon	Videoklippet viste at $RPM_{arduino}$ lå innenfor intervallet til RPM_{reell} . Basert på dette kan det fastslås at avviket på $RPM_{arduino}$ er et sted mellom 0% og $\pm 3.2\%$. Testen kan derfor ikke brukes til å avkrefte eller bekrefte systemkravet på maksimalt $\pm 2.5\%$ avvik.
6.2.8	Sensorsystem	Turteller	Presisjon	Bestått	Basert på målinger gjennomført med oscilloskop av signalene i kretsen kan det fastslås at kretsen fungerer logisk som den skal. Forsinkelse i kretsen førte til at Δt_f ble målt til 5 ms. Dette tilsvarer et avvik på -1.0% ved en reell rotasjonshastighet på 123.97 rpm. Dette oppfyller systemkravet på $\pm 2.5\%$.
6.2.11	Sensorsystem	Vinkelsensor	Presisjon	Bestått	20 målinger med hastighet på 1.4 m s^{-1} og uten curl ble gjennomført. Gir målingsavvik på $\pm 1.8^\circ$ som er innenfor kravet på $\pm 3^\circ$.

7 Validering

Etter integrasjon av det fulle systemet ble produktet vurdert av en interessent. Følgende har ice-maker Martin Sesaker sagt om produktet:

"Libra kan hjelpe meg som en ice-maker på det å få testet steinen veldig grundig ved å lage statistikk og data på hvordan hver enkelt Stein oppfører seg. Jeg kan være frekk å bruke det til min egen fordel, men kan også dele det med alle klubbens medlemmer sånn at de også kan bli bedre curling-spillere. Samtidig kan jeg teste [og] forbedre isen som en ice-maker. Libra kan være med å hjelpe livet til curlingspillere i en veldig stor grad, ved at curlingen og curlingsteinene blir helt på nivå med de beste i verden [...] til en hver tid."

(Martin Sesaker, Ice-maker i Trondheim Curlingklubb, 2021) [37]

Dette viser at konseptet og en forbedring av produktet kan brukes til oppfylle interessentenes behov.

8 Forslag til forbedringer

8.1 Mekaniske forbedringer

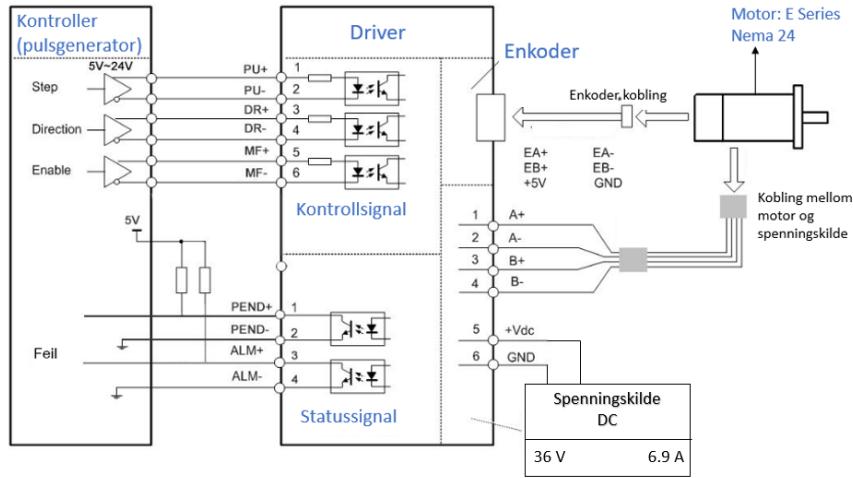
- Uelastisk snøre eller belte slik at trinnmotoren ikke vil flytte den dynamiske basen for kort eller for langt ved variasjon i last og akselerasjon.
- En mekanisk arm eller liknende som kan holde på plass curlingsteinen når den skal roteres.
- Flere kulelager, for mindre friksjon ved transport av rotasjonssystemet.
- Tannhjul til motor i høyere kvalitet, med høyere slitestyrke. f.eks. i metall, gjerne mer effektive tannhjul for reduksjon av friksjon.

8.2 Elektroniske forbedringer

- Sterkere skyvemotor. Gjerne en Nema 34 med 12 N m dreiemoment.
- Sterkere rotasjonsmotor som kan overvinne curlingsteinens statiske friksjon.
- Mer sammensatt system, slik at en ikke trenger så mange arduinoer og batterier/spenningskilder.
- Lage PCB kretser for å redusere fysisk størrelse på kretser, og gjøre de mindre ustabile for å kunne løsne, og redusere støy fra kapasitans mellom ledninger osv.

8.2.1 Skyvesystem : Implementasjon av statussignaler

Som det ble nevnt i seksjon 5.2.2, så er det mulig å konstruere et fullstendig lukket sløyfesystem med den valgte motoren for skyvesystemet. I et slikt system vil statussignalene implementeres. Figur 45 viser et slikt system hvor man implementerer alle inngangene til driveren. En kort og oversiktlig forklaring på funksjonen til disse inngangene finnes i seksjonen A.5. Et slikt system ville gitt muligheten til å programmere mikrokontrolleren slik at systemet hadde reagert på typiske feil. Dette hadde gitt et mer automatisert system. Eksempel på typiske feil er underspenning, overspenning eller posisjoneringsfeil.



Figur 45: Et fullstendig lukket sløyfe hvor alle inngangene til driveren er koblet opp.

8.3 Sensorforbedringer

8.3.1 Fartssensor

Benytte raskere komponenter. Eksempelvis kan fotoresistorene som har opp mot 10ms forsinkelsestid [36] erstattes med en fotodiode. En annen løsning vil være å ta hånd om forsinkelstiden i koden og justere tidsintervallet deretter.

8.3.2 Turteller

En svakhet ved sensoren er at den er avhengig av at rotasjonshjulet roterer en hel runde for å gjøre beregninger. Ved lave rotasjonshastigheter er den derfor avhengig av et større tidsintervall for å kunne gjøre målingen.

Dersom rotasjonssystemet programmeres til å rotere hjulet en hel runde med konstant hastighet før skyvesystemet settes i gang vil ikke dette være et problem.

Rotasjonssensor brukte 4 målinger for å kunne avgjøre rpm, noe som gjør den dårlig for lave hastigheter. ca ± 5 rpm avvik.

Eventuelt kan antallet målinger per rotasjon økes ved å implementere flere og mindre magneter som skal registreres.

8.3.3 Vinkelsensor

Endre koden slik at begge avstandssensorene starter sine målinger ved deteksjon av stein. Dette gir rom for flere målinger som også vil være en forbedring.

9 Konklusjon

9.1 Innsikt oppnådd

Det er funnet et behov for billigere curlingsteinkastere enn de som er tilgjengelige på dagens marked for å gi flere curlingspillere muligheten til å spille og trenre på is av høyeste kvalitet. Libra er en mulig løsning på problemet og et bevis på konsept som anerkjennes av ice-maker Martin Sesaker.

Libra oppfyller ikke alle systemkrav og er derfor ikke klar for markedet. Sannsynligvis vil produksjonskostnadene på et nytt tilsvarende eksemplar være under 10 000 NOK. Sannsynligvis vil det være mulig å utvikle Libra slik at det både oppfyller systemkravene og er billigere enn CRT som koster \$75,000.00 CDN.

9.2 Viktige resultater

Gode tester har gitt kunnskap om hvilke deler som ikke tåler nåværende belastning. Dermed er det klart hvilke deler som må skiftes ut.

Systemet har et brukergrensesnitt hvor brukeren enkelt kan styre skyvesystemet. Brukeren har mulighet til å regulere fart og tilgang til nødstopp. Dette er satt opp med få, oversiktlige og enkle knapper.

Det trengs en sterkere rotasjonsmotor fordi statisk friksjon var for stor. Det må også lages et design som kan holde steinen inntil rotasjonsmotorene, for å så enkelt slippe steinen.

Systemet har en hastighetssensor som må forbedres for å kunne oppfylle systemkravene. Det er implementert en turteller som fungerer riktig logisk. Den har liten forsinkelse med avvik på -1.0% i presisjon, som oppfyller systemkravet på $\pm 2.5\%$. Den implementerte vinkelsensoren er også innenfor systemkravet, med et målingsavvik på $\pm 1.88^\circ$.

Basert på resultatene er det flere områder som må forbedres for å oppnå et produkt som er klart for markedet. Produktet kan nå i en viss grad sende en curlingstein med lik rotasjon, utgangsfart og utgangsvinkel, med mulighet for verifisering. Dette gir et godt utgangspunkt for å skape identiske kast i fremtiden, gitt at nødvendige komponenter blir oppgradert. Systemet er brukervennlig for brukeren hvor få knapper og automatisering gjør at brukeren ikke trenger å bruke mye tid på å lære seg systemet. Testingen og resultatene har gitt god innsikt i hva som må gjøres og hvordan Libra kan forbedres. For det mekaniske er det viktigste å få på plass uelastisk snøre, tannhjul som tåler belastningen og mekanisk arm som gjør rotasjon mulig. Når det gjelder det elektroniske er det spesielt viktig å få sterkere motorer. Det blir også viktig å sørge for at sensorene oppfyller systemkravene. Med et litt større budsjett er det gode mulighetet for å realisere konseptet til et produkt som kan selges på markedet.

Referanser

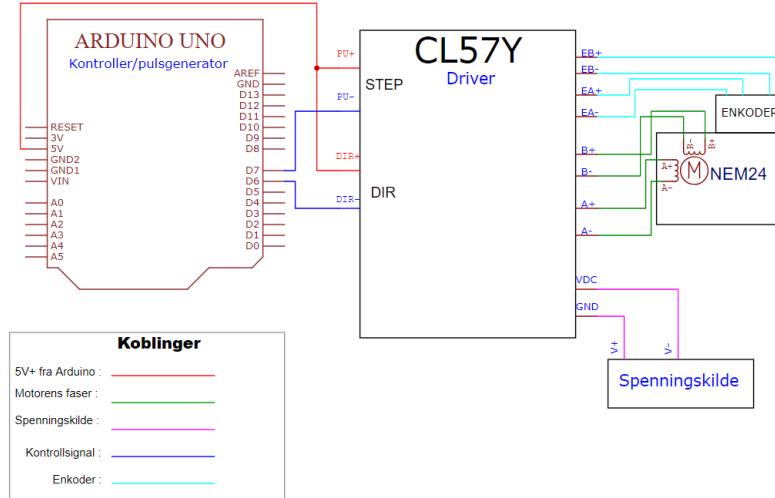
- [1] (ingen dato). «Glossary of curling,» Wikipedia, adresse: https://en.wikipedia.org/wiki/Glossary_of_curling (sjekket 2021).
- [2] D. Heroux. (2019). «The secret of the stone, or how 'papering' gets the best out of 42 lbs. of granite,» adresse: <https://www.cbc.ca/sports/olympics/winter/curling/the-secret-of-the-stone-or-how-papering-gets-the-best-out-of-48-lbs-of-granite-1.5061970> (sjekket 2021).
- [3] (ingen dato). «Om Curling,» Norges Curling Forbund, adresse: <https://www.curling.no/index.php/om-curling> (sjekket 2021).
- [4] (2020). «Meet Curly, the Curling Robot That Beats the Pros,» Wired, adresse: <https://www.wired.com/story/meet-curly-the-curling-robot-that-beats-the-pros/> (sjekket 2021).
- [5] E. R. Hendry. (2014). «Why Curling Ice is Different Than Other Ice,» adresse: <https://www.smithsonianmag.com/innovation/why-curling-ice-is-different-than-other-ice-180949752/> (sjekket 2021).
- [6] R. Huehmer. (ingen dato). «Making Championship Curling Ice - Water Treatment,» adresse: <https://www.linkedin.com/pulse/making-championship-curling-ice-water-treatment-robert-huehmer> (sjekket 2021).
- [7] D. V. Winkle. (2014). «Learn Why Curling Has So Much Sweeping, Because Science Is the Best Part of Sports,» adresse: <https://www.themarysue.com/science-of-curling-sweeping/> (sjekket 2021).
- [8] (ingen dato). «History of curling,» World Curling Federation, adresse: <https://worldcurling.org/about/history/> (sjekket 2021).
- [9] D. V. Brenk. (2020). «Curling researchers dig deep into sweep science,» adresse: <https://news.westernu.ca/2020/09/curling-researchers-dig-deep-into-sweep-science/> (sjekket 2021).
- [10] (2016). «Here's what you need to know about sweeping in 2016,» Curling Canada, adresse: <https://www.curling.ca/blog/2016/02/09/heres-what-you-need-to-know-about-sweeping-in-2016/> (sjekket 2021).
- [11] T. Wyman. (2021). «Directional sweeping has changed curling and it's here to stay,» adresse: <https://torontosun.com/sports/curling/directional-sweeping-has-changed-curling-and-its-here-to-stay> (sjekket 2021).
- [12] S. Fitz-Gerald. (2016). «The key to perfect curling ice,» adresse: <https://www.thestar.com/sports/curling/2016/02/24/the-key-to-perfect-curling-ice.html> (sjekket 2021).
- [13] (2010). «Curling Rocks,» Science World, adresse: <https://www.scienceworld.ca/stories/curling-rocks/> (sjekket 2021).
- [14] (ingen dato). «Curling,» Wikipedia, adresse: <https://en.wikipedia.org/wiki/Curling> (sjekket 2021).
- [15] M. Rosset. (2017). «Manitoba curling club builds world's first rock throwing machine,» adresse: <https://globalnews.ca/news/3816318/manitoba-curling-club-builds-worlds-first-rock-throwing-machine/> (sjekket 2021).

- [16] (2021). «Mail-korrespondanse mellom "Lorne og Chris Hamblin" og Hans Henrik Heiberg.»
- [17] (2021). «Mail-korrespondanse mellom Lightguide Systems Inc. og Hans Henrik Heiberg.»
- [18] W. Dong-Ok, K. Byung-Do, K. Ho-Jung, E. Tae-San, M. Klaus-Robert og L. Seong-Whan. (2017). «Curly: An AI-based Curling Robot Successfully Competing in the Olympic Discipline of Curling,» adresse: <https://www.ijcai.org/Proceedings/2018/0870.pdf> (sjekket 2021).
- [19] (ingen dato). «SweepTracker,» Lightguide Systems Inc., adresse: <http://lightguidesystems.com/index.php/curling-technology/sweeptracker> (sjekket 2021).
- [20] (ingen dato). «Curly: de robot die een expert is in curling,» Fresh Gadgets, adresse: <https://freshgadgets.nl/curly-de-robot-die-een-expert-is-in-curling> (sjekket 2021).
- [21] (ingen dato). «Klubber,» Norges Curling Forbund, adresse: <https://www.curling.no/index.php/klubber> (sjekket 2021).
- [22] M. McCauley. (2020). «AccelStepper library for Arduino,» adresse: <http://www.airspayce.com/mikem/arduino/AccelStepper/> (sjekket 2021).
- [23] (2020). «Arduino-Button Library,» Arduinogetstarted, adresse: <https://arduinogetstarted.com/tutorials/arduino-button-library> (sjekket 2021).
- [24] (2019). «Arduino-LiquidCrystall,» Arduino, adresse: <https://www.arduino.cc/en/Reference/LiquidCrystal> (sjekket 2021).
- [25] (ingen dato). «Stepper Library,» Arduino, adresse: <https://www.arduino.cc/en/reference/stepper> (sjekket 2021).
- [26] (ingen dato). «49E Hall-Effect Linear Position Sensor.» v1.1, YANGZHOU POSITIONING TECH. CO., LTD, adresse: <https://p.globalsources.com/IMAGES/PDT/SPEC/440/K1139513440.pdf> (sjekket 2021).
- [27] (ingen dato). «LF353 Wide-Bandwidth JFET-Input Dual Operational Amplifier,» Texas Instruments, adresse: <https://www.ti.com/lit/ds/symlink/lf353.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1619844187132> (sjekket 2021).
- [28] (2017). «Inverting Schmitt Trigger Theory,» random-science-tools, adresse: <https://www.random-science-tools.com/electronics/inverting-schmitt-trigger-theory.htm> (sjekket 2021).
- [29] (ingen dato). «constants,» Arduino, adresse: <https://www.arduino.cc/reference/en/language/variables/constants/> (sjekket 2021).
- [30] (ingen dato). «Adafruit GFX Library,» Adafruit Industries, adresse: <https://github.com/adafruit/Adafruit-GFX-Library> (sjekket 2021).
- [31] (ingen dato). «Adafruit ST7745 Library,» Adafruit Industries, adresse: <https://github.com/adafruit/Adafruit-ST7735-Library> (sjekket 2021).
- [32] (ingen dato). «GL55 Series Photo Resistor (LDR),» Handson Technology, adresse: <https://www.handsontec.com/dataspecs/sensor/GL55-LDR.pdf> (sjekket 2021).
- [33] (ingen dato). «LiquidCrytal Library,» Arduino, adresse: <https://www.arduino.cc/en/Reference/LiquidCrystal> (sjekket 2021).

- [34] (2020). «Adafruit VL53L0X Time of Flight Micro-LIDAR Distance Sensor Breakout,» Adafruit Industries, adresse: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout.pdf> (sjekket 2021).
- [35] (2020). «Adafruit VL53L0X Library,» Adafruit Industries, adresse: https://github.com/adafruit/Adafruit_VL53L0X/blob/master/examples/vl53l0x_dual/vl53l0x_dual.ino (sjekket 2021).
- [36] (ingen dato). «Photoresistor,» Wikipedia, adresse: <https://en.wikipedia.org/wiki/Photoresistor> (sjekket 2021).
- [37] (2021). «Intervju av Martin Sesaker gjort av Hans Henrik Heiberg,» adresse: <https://youtu.be/stWDMJNyR4w>.
- [38] (2019). «Nema24 datasheet,» StepperOnline, adresse: <https://www.omc-stepperonline.com/download/24HE40-5004D-E1000.pdf> (sjekket 2021).
- [39] (2019). «250W 36V 6.9A 115/230V Switching Power Supply Stepper Motor CNC Router Kits,» StepperOnline, adresse: <https://www.omc-stepperonline.com/250w-36v-70a-115230v-switching-power-supply-stepper-motor-cnc-router-kits-s-250-36.html?search=power%5C%20supply> (sjekket 2021).

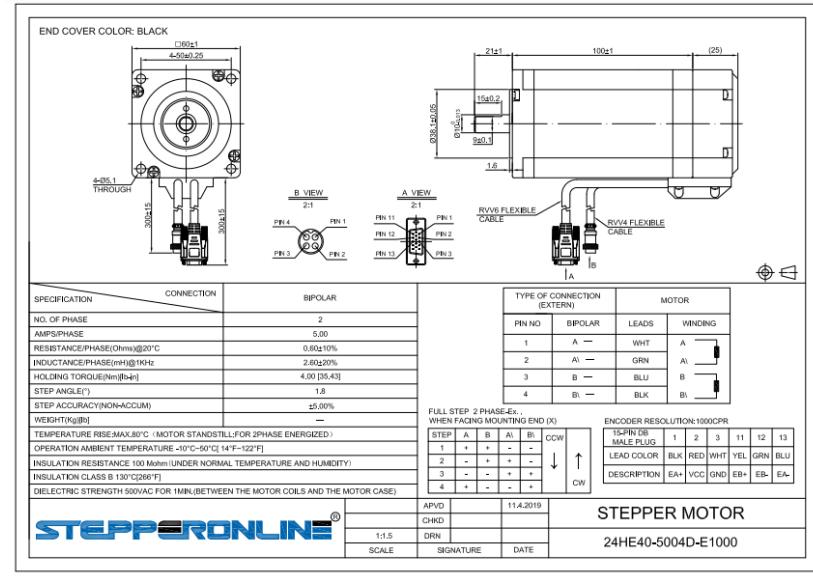
A Skyvesystem

A.1 Kretsdiagram av oppkobling



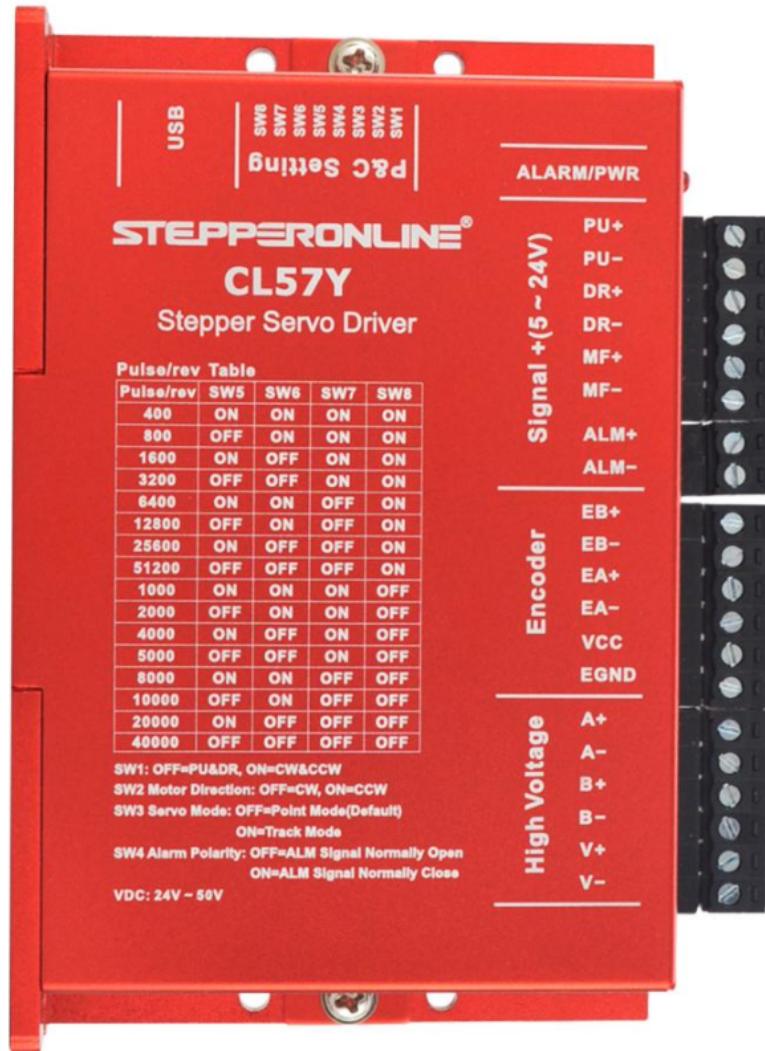
Figur 46: Kretsskjema av implementasjonen av trinnmotoren.

A.2 Datablad Nema 24



Figur 47: Motoren bak skyvemekanismen [38].

A.3 Driver (CL57Y)



Figur 48: Driveren som har blitt brukt til å realisere motoren Nema 24.

A.4 Spenningskildens datablad.

250W Power Supply S-250-36

Features:

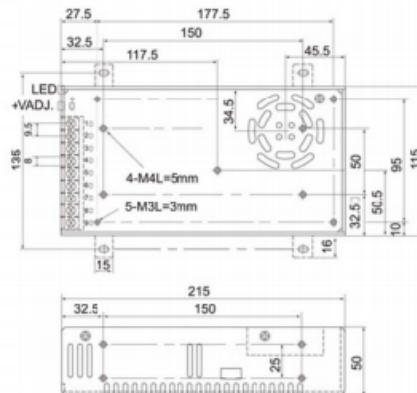
- 36V DC 7A output
- AC input voltage range: 90~132V/180~264VAC
- 115V/230V AC selected by switch
- High efficiency low cost
- Forced air cooling by built-in DC fan
- Low output ripple and yawp
- Over current, over voltage, short circuit and overheat protections
- 215*115*50mm (L*W*H)



General Specification:

Model	S-250-36
DC Output	36V 7A
Wave and Noise	150mVp-p
Inlet Stability	±0.5%
Load Stability	±0.5%
Efficiency	86%
Adjustable range for DC Voltage	10%
AC Input Voltage	90~132V/180~264VAC Selected by Switch
AC Input Current	5.0A/115VAC 2.5A/230VAC
Working Temperature	-10~50°C
Safety Standards	GB4943, UL60950, EN60950
EMC Standards	GB9254, 55022, ClassB
Weight	1.1kg

Dimensions:



StepperOnline Co.,Ltd / Sales@stepperonline.com / www.omc-stepperonline.com

Figur 49: Databladet til spenningkilden som har blitt til å realiserer skyvemekanismen [39].

A.5 Driverens innganger

A.5.1 Kontrollsignal

Tabell 9: Beskrivelse og funksjonalitet av kontrollsignal innganger.

Inngang	Beskrivelse
$PU+$	Koblet til signalets spenningskilde. Et signal mellom +5V til +24V kan brukes.
$PU-$	Fallende flanke er gyldig og pulsen fører til motoren tar et steg når pulsen går fra høy til lav. Det kreves et lav nivå mellom 0 til +0.5 V, et høy nivå mellom +5 V til +24 og en puls bredde større enn $2.5\mu s$.
$DR +$	Koblet til signalets spenningskilde. Et signal fra +5V til +24V kan brukes.
$DR -$	Brukes til å skifte rotasjonsretningen på motoren. Det kreves et lav nivå mellom 0 til +0.5 V, et høy nivå mellom +5 V til +24 og en puls bredde større enn $2.5\mu s$
$MF +$	Koblet til signalets spenningskilde. En verdi fra +5V til +24V kan brukes.
$MF -$	Når aktiv (lav nivå) vil det føre til at motorspolen slås av, driveren stopper og motoren vil være i fri tilstand.

A.5.2 Statussignal

Tabell 10: Beskrivelse og funksjonalitet av statussignal innganger.

Inngang	Beskrivelse
$PEND+, PEND-$	Etter at driveren har gjennomført en gitt pulse, så vil IMP signalet (in-position singal) være gyldig. $PEND-$ er den negative enden for IMP signalet, og den er jordet. Den maksimale strømmen gjennom denne inngangen er 50mA.
$ALM+, ALM-$	Ved overspenning, underspenning, overstrøm eller posisjonsfeil vil ALM være gyldig. $ALM-$ er jordet, og den maksimale strømmen i inngangen vil være 50mA.

A.5.3 Enkoder

Tabell 11: Beskrivelse og funksjonalitet av enkoder innganger.

Inngang	Beskrivelse
$EB-$	Enkoder B- inngang.
$EB+$	Enkoder B+ innang.
$EA-$	Enkoder A- inngang.
$EA+$	Enkoder A+ inngang.

A.5.4 Motor og spenningskilde

Tabell 12: Beskrivelse og funksjonalitet av innganger som er i kontakt med motoren og spenningskilden.

Inngang	Beskrivelse
GND	Spenningskildens kobling som er jordet.
$+Vdc$	Spenningskildens positive kobling. 24-48VDC anbefales.
$A+, A-$	Motor fase A kobling.
$B+, B-$	Motor fase B kobling.

A.5.5 Posisjoneringstest

Tabell 13: Målinger som ble observert på gulvflis uten last.

Måling	Avvik [cm]
1	0.04
2	0.06
3	0.09
4	0.08
5	0.09
6	0.1
7	0.09

Tabell 14: Gjennomsnitt og typetall av målinger som ble gjennomført på gulvflis uten last.

Verdi	Avvik [cm]
Gjennomsnitt av målinger	0.078 ± 0.022
Typetall av målinger	0.09

Tabell 15: Målinger som ble observert på is uten last.

Måling	Avvik [cm]
1	6.5
2	6.5
3	10
4	7
5	5
6	6
7	9

Tabell 16: Gjennomsnitt og typetall fra målinger på is uten last.

Verdi	Avvik [cm]
Gjennomsnitt av målinger	7.142 ± 1.619
Typetall av målinger	6.5

Tabell 17: Målinger som ble observert på is og med en last på 5kg.

Måling	Avvik [cm]
1	15.4
2	15.2
3	15.4
4	13.5
5	14.9
6	14.9
7	14.4

Tabell 18: Gjennomsnittlig og typetall observert fra tester utført på is og en last på 5kg.

Verdi	Avvik [cm]
Gjennomsnitt av målinger	14.81 ± 0.6266
Typetall av målinger	14.9

Tabell 19: Målinger som ble observert på is med en curlingstein på 20 kg.

Måling	Avvik [cm]
1	100
2	90.8
3	115.2
4	100.3
5	99.7
6	109.6
7	100.3

Tabell 20: Gjennomsnitt og typetall observert fra målinger som ble utført på is og med en curlingstein på 20kg.

	Verdi	Avvik [cm]
Gjennomsnitt av målinger	102.27 ± 7.289	
Typetall av målinger	100.3	

B Kode for skyvesystemets metodikk

Listing 1: Arduinokode for skyvesystemet

```
#include <AccelStepper.h>
#include <ezButton.h>
#include <LiquidCrystal.h>

//----- DEFINING PERMANENT VALUES -----
#define killSwitch 1
#define microPin 9
const unsigned short StepsPerRev = 400;
const float circ = 0.2199; //In meters
const float dist = 1.5; //In meters
const float calibrationFactor = 1.031; //Used to compensate for some mechanical
elasticity
const float extraAcceleration = 500; //This variable is sett to compensate for
lower torque at high rpm.
const float backwardSpeed = 150;
const float maxSpeedVal = 2.5; //2.5 m/s
const float incr = 0.1; //increasing value

//----- SETTING PINS -----
AccelStepper stepper(1,7,6); //Defines stepper pins
ezButton interButGo(8); //Sends stone

LiquidCrystal lcd(12, 11, 5, 4, 3, 13); //LCD-display
ezButton interButUp(10); //Fart++
ezButton interButDown(2); //Fart--

//----- DEFINING GLOBAL VARIABLES -----
bool forward = false;
bool forwardSet = false;
bool backward = true;
float fart; //variable for speed

//----- DEFINING FUNCTIONS -----
float calcStepDist(float l, unsigned short SPR, float circumference){ //l in meters
    return (l*SPR/circumference);
}

float calcStepSpeed(float v, float circumference, unsigned short SPR){
    float rps = v/circumference;
    return (SPR*rps);
}

float calcStepAcceleration(float stepSpeed, float stepDist){
    return (stepSpeed*stepSpeed/(2*stepDist));
}

const float distInSteps = calcStepDist(dist, StepsPerRev, circ*calibrationFactor);
//defining number of steps to reach a distance
```

```

float stepVelocity = calcStepSpeed(1, circ*calibrationFactor, StepsPerRev);
float stepAcceleration = calcStepAcceleration(stepVelocity, distInSteps);

void setThrowAcc(float aStep, float vStep, float lStep){ //setting acceleration for
    the stone
interButGo.loop();
if(interButGo.isPressed()){
    stepper.setAcceleration(aStep+extraAcceleration);
    stepper.setMaxSpeed(vStep);
    stepper.moveTo(lStep);
    forwardSet = true;
}
}

void returnStone(float backSpeed){
    stepper.setSpeed(-backSpeed);
    stepper.runSpeed();
}

//Fart++ , fart-- funksjon
float adjustSpeed(){
    interButUp.loop();
    interButDown.loop();

    if(interButUp.isPressed()){ //up
        fart+=incr;
        if(fart<=maxSpeedVal){
            lcd.clear();
            lcd.setCursor(5,0);
            lcd.print(fart);
            lcd.print("m/s");
            return fart;
        }
        fart-=incr;
        lcd.clear();
        lcd.setCursor(5,0);
        lcd.print(fart);
        lcd.print("m/s");
        return fart;
    }
    if(interButDown.isPressed()){ //down
        fart-=incr;
        if(fart>0){
            lcd.clear();
            lcd.setCursor(5,0);
            lcd.print(fart);
            lcd.print("m/s");
            return fart;
        }
        fart+=incr;
        lcd.clear();
        lcd.setCursor(5,0);
        lcd.print(fart);
    }
}

```

```

        lcd.print("m/s");
        return fart;
    }
    return fart;
}
//----- SETUP FUNCTION -----

void setup() {
    //button setup
    pinMode(microPin, INPUT);
    pinMode(killSwitch, INPUT);
    interButGo.setDebounceTime(50);

    //Motor setup
    stepper.setCurrentPosition(0);
    stepper.setMaxSpeed(1000);

    ///LCD Screen setup
    lcd.begin(16,2);
    lcd.setCursor(5, 0);
    lcd.print("Fart: ");
    lcd.setCursor(5,1);
    lcd.print(fart);
    lcd.print("m/s");

}

//----- MAIN LOOP -----

void loop() {
    if(digitalRead(killSwitch)== HIGH){
        if(forward){
            if(!forwardSet){
                stepVelocity = calcStepSpeed(adjustSpeed(), circ*calibrationFactor,
                    StepsPerRev);
                stepAcceleration = calcStepAcceleration(stepVelocity, distInSteps);
                setThrowAcc(stepAcceleration, stepVelocity, distInSteps);
            }else{
                stepper.run();
            }
            if(stepper.distanceToGo()==1){
                backward = true;
                stepper.setCurrentPosition(distInSteps);
                forward = false;
                delay(10000);
            }
        }
        if (backward){
            if(digitalRead(microPin) == HIGH){
                returnStone(backwardSpeed);
            }else{
                stepper.setCurrentPosition(0);
            }
        }
    }
}

```

```
    backward = false;
    forward = true;
    forwardSet = false;
}
}
} else {
    backward = false;
    forward = false;
    forwardSet = false;
    lcd.clear();
    lcd.print("RESET!");
}

```

C Kode for rotasjonsmotor

Listing 2: Arduinokode for kontrollering av rotasjonsmotoren [25].

```
#include <Stepper.h>

const int stepsPerRevolution = 200; // change this to fit the number of steps per
// revolution
// for your motor

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 2, 3, 4, 5);

int stepCount = 0; // number of steps the motor has taken

void setup() {
    // nothing to do inside the setup
}

void loop() {
    // read the sensor value:
    // map it to a range from 0 to 100:
    int motorSpeed = map(1023, 0, 1023, 0, 100);
    // set the motor speed:
    if (motorSpeed > 0) {
        myStepper.setSpeed(50);
        // step 1/100 of a revolution:
        myStepper.step(stepsPerRevolution / 100);
    }
}
```

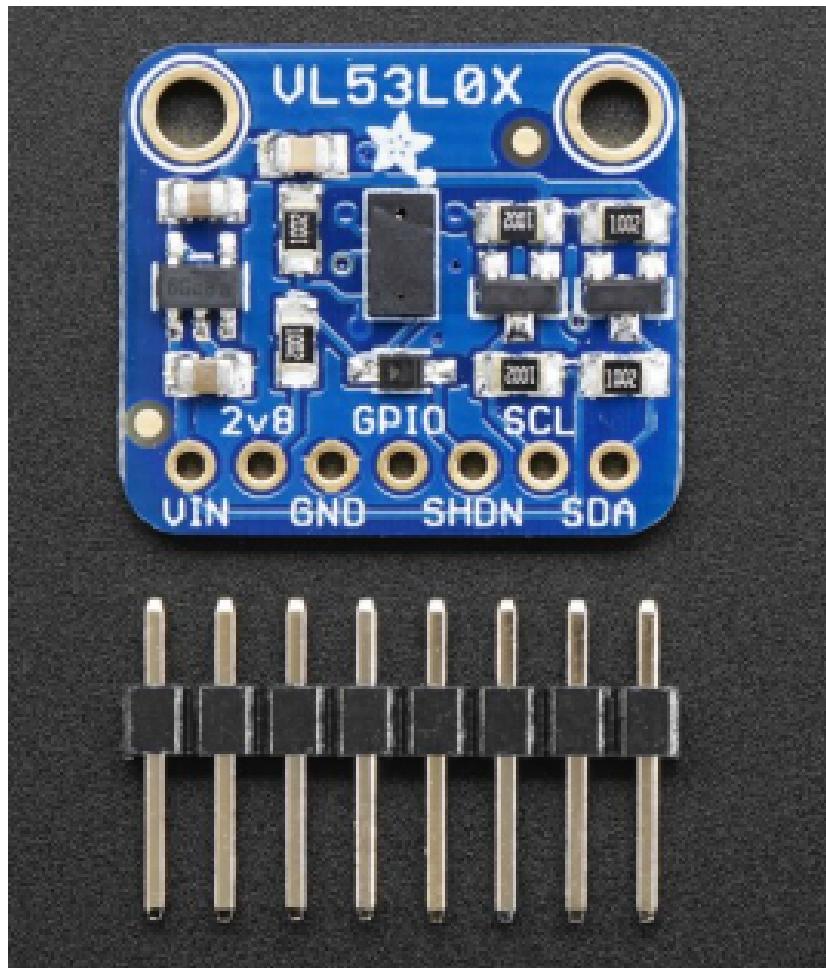
D Sensorer

D.1 Bilde av Rotasjonssensor



Figur 50: Bilde av brukt sensor [28]. Dette er en 49E Hall-effect sensor som registrerer magnetfelt.

D.2 Bilde av vinkelsensor



Figur 51: Bilde av brukt sensor [30].

E Kode for turteller

Listing 3: Arduinokode for turteller

```
#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>
#include <SPI.h>

#define CS      10
#define RST     8
#define DC      9

Adafruit_ST7735 ada = Adafruit_ST7735(CS, DC, RST); //LCD object

void setup() {

    // Screen setup
    ada.initR(INITR_BLACKTAB);

    // RPM setup
    pinMode(2, INPUT_PULLUP);
    attachInterrupt(digitalPinToInterrupt(2), interruptFunctionRPM, FALLING);

}

unsigned long meas[] = {0,0,0,0};
long elapsedTime = 0;
float rpm = 0;

void loop() {
    rpm = 0;
    delay(1000);                                // calculations and screen updates
    every 1 sec

    elapsedTime = meas[0] - meas[sizeof(meas)-1]; // time between first and last
    pulse in list

    rpm = 15000.0 / elapsedTime;                // calculates rpm (Remember to
    adjust to number of magnets on wheel)

    ada.fillScreen(ST77XX_BLACK);
    ada.setCursor(40, 50);
    ada.setTextColor(ST77XX_GREEN);
    ada.setTextSize(2);
    ada.setTextWrap(true);
    ada.print(String("RPM: \n ") + rpm);
}

void interruptFunctionRPM() //interrupt called every time magnetic field detected
{
    for (int i = 1; i < 4; i++) {
```

```
    meas[i] = meas[i-1];
}
meas[0] = millis();
}
```

F Kode for fartssensor

Listing 4: Arduinokode for fartssensor

```
#include <LiquidCrystal.h>

// initializing pins
const int d4 = 5, d5 = 4, d6 = 7, d7 = 6, rs = 12, en = 11;
LiquidCrystal lcd = LiquidCrystal(rs, en, d4, d5, d6, d7);

unsigned long meas1 = 0;
unsigned long meas2 = 0;
double elapsedTime = 0;
double velocity = 0;
double velocityKmh = 0;
double distance = 0.29; // Curlingsteinens diameter
int interruptCount = 0;

void setup() {
    pinMode(3, INPUT_PULLUP);

    lcd.begin(16, 2);
    attachInterrupt(digitalPinToInterrupt(3), interruptFunction, CHANGE);
}

void loop() {
    elapsedTime = (meas2 - meas1)/1000.0; // elapsed time in seconds
    velocity = distance/elapsedTime;
    velocityKmh = velocity * 3.6;

    lcd.setCursor(0,0);
    lcd.print("m/s: " + String(velocity));
    lcd.setCursor(0,1);
    lcd.print("km/h: " + String(velocityKmh));

    delay(1000);
}

void interruptFunction()
{
    if(digitalRead(3) == LOW){
        meas1 = millis();
        interruptCount = 1;
    }
    else{
        meas2 = millis();
        interruptCount++;
    }
}
```

G Kode for vinkelsensor

Listing 5: Arduinokode for vinkelsensor

```
#include "Adafruit_VL53L0X.h"

//DISPLAY
#include <LiquidCrystal.h>
//DISPLAY

// address we will assign if dual sensor is present
#define LOX1_ADDRESS 0x30
#define LOX2_ADDRESS 0x31

// set the pins to shutdown
#define SHT_LOX1 10
#define SHT_LOX2 9

// objects for the vl53l0x
Adafruit_VL53L0X lox1 = Adafruit_VL53L0X();
Adafruit_VL53L0X lox2 = Adafruit_VL53L0X();

// this holds the measurement
VL53L0X_RangingMeasurementData_t measure1;
VL53L0X_RangingMeasurementData_t measure2;

double degree;
int absolute;

float value1 = 0.0;
float value2 = 0.0;

int lst1[20];
int lst2[20];
int index1 = 0;
int index2 = 0;

int iterations = 4;
int low_limit = 100;
int high_limit = 400;

/*
    Reset all sensors by setting all of their XSHUT pins low for delay(10), then
    set all XSHUT high to bring out of reset
    Keep sensor #1 awake by keeping XSHUT pin high
    Put all other sensors into shutdown by pulling XSHUT pins low
    Initialize sensor #1 with lox.begin(new_i2c_address) Pick any number but 0x29
    and it must be under 0x7F. Going with 0x30 to 0x3F is probably OK.
    Keep sensor #1 awake, and now bring sensor #2 out of reset by setting its XSHUT
    pin high.
    Initialize sensor #2 with lox.begin(new_i2c_address) Pick any number but 0x29
    and whatever you set the first sensor to

```

```

/*
void setID() {
    // all reset
    digitalWrite(SHT_LOX1, LOW);
    digitalWrite(SHT_LOX2, LOW);
    delay(10);
    // all unreset
    digitalWrite(SHT_LOX1, HIGH);
    digitalWrite(SHT_LOX2, HIGH);
    delay(10);

    // activating LOX1 and reseting LOX2
    digitalWrite(SHT_LOX1, HIGH);
    digitalWrite(SHT_LOX2, LOW);

    // initing LOX1
    if(!lox1.begin(LOX1_ADDRESS)) {
        Serial.println(F("Failed to boot first VL53L0X"));
        while(1);
    }
    delay(10);

    // activating LOX2
    digitalWrite(SHT_LOX2, HIGH);
    delay(10);

    //initing LOX2
    if(!lox2.begin(LOX2_ADDRESS)) {
        Serial.println(F("Failed to boot second VL53L0X"));
        while(1);
    }
}

void read_dual_sensors() {

    lox1.rangingTest(&measure1, false); // pass in 'true' to get debug data printout!
    lox2.rangingTest(&measure2, false); // pass in 'true' to get debug data printout!

    // print sensor one reading
    Serial.print(F("1: "));
    if(measure2.RangeStatus != 4) { // if not out of range
        Serial.print(measure2.RangeMilliMeter);
    } else {
        Serial.print(F("Out of range"));
    }

    Serial.print(F(" "));

    // print sensor two reading
    Serial.print(F("2: "));
    if(measure1.RangeStatus != 4) {
        Serial.print(measure1.RangeMilliMeter);
    } else {
}
}

```

```

    Serial.print(F("Out of range"));
}

Serial.print(F(" "));

Serial.println();

get_value();
}

void get_value(){

if(measure2.RangeMilliMeter >= low_limit and measure2.RangeMilliMeter <=
high_limit){
for (int i = 0; i <= iterations; i++) {
lox2.rangingTest(&measure2, false);
lst1[i]=measure2.RangeMilliMeter;

Serial.print(F("1: "));
if(measure2.RangeStatus != 4) { // if not out of range
Serial.print(measure2.RangeMilliMeter);
} else {
Serial.print(F("Out of range"));
}

Serial.print(F(" "));
}

for (int i = 0; i <= iterations; i++){
if ((lst1[i] <= lst1[index1]) && (lst1[i] < high_limit) && (lst1[i] >
low_limit)) {
index1 = i;
value1 = lst1[index1];
}
}

for (int i = 0; i <= iterations; i++) {
lox1.rangingTest(&measure1, false);
lst2[i]=measure1.RangeMilliMeter;

Serial.print(F("2: "));
if(measure1.RangeStatus != 4) {
Serial.print(measure1.RangeMilliMeter);
} else {
Serial.print(F("Out of range"));
}

Serial.print(F(" "));
}

for (int i = 0; i <= iterations; i++){
if ((lst2[i] <= lst2[index2]) && (lst2[i] < high_limit) && (lst2[i] >
low_limit)) {
index2 = i;
value2 = lst2[index2];
}
}
}

```

```

        }

        absolute = abs(value2-value1);
        degree = atan(absolute/150.0)*57.3;

        Serial.println();

        Serial.print(F("S1: "));
        Serial.print(value1);

        Serial.print(F(" "));

        Serial.print(F("S2: "));
        Serial.print(value2);

        Serial.print(F(" "));

        Serial.print(F("3: "));
        Serial.print(degree);

        Serial.print(F(" "));

        Serial.println();

        delay(10000);

        index1 = 0;
        index2 = 0;
        value1 = 0.0;
        value2 = 0.0;
    }
}

//DISPLAY
LiquidCrystal lcd(2,3,4,5,6,7);
//DISPLAY

void setup() {
    Serial.begin(115200);

    // wait until serial port opens for native USB devices
    while (! Serial) { delay(1); }

    pinMode(SHT_LOX1, OUTPUT);
    pinMode(SHT_LOX2, OUTPUT);

    Serial.println(F("Shutdown pins initied..."));

    digitalWrite(SHT_LOX1, LOW);
    digitalWrite(SHT_LOX2, LOW);

    Serial.println(F("Both in reset mode...(pins are low)"));
}

```

```
Serial.println(F("Starting..."));
setID();

//DISPLAY
lcd.begin(16,2);
lcd.clear();
//DISPLAY
}

void loop() {

    read_dual_sensors();
    delay(1);

    //DISPLAY
    lcd.setCursor(0,1);
    lcd.print("Vinkel:");
    lcd.print(degree);
    lcd.setCursor(0,0);
    lcd.print("1:");
    lcd.print(measure1.RangeMilliMeter);
    lcd.print(" 2:");
    lcd.print(measure2.RangeMilliMeter);
    lcd.print("  ");
    //DISPLAY
}
```
