

Forbedret treningsmaskin for fotballspillere

Torbjørn Tveit Alvestrand

Master i teknisk kybernetikk og robotikk

Innlevert: juni 2014

Hovedveileder: Øyvind Stavdahl, ITK

Norges teknisk-naturvitenskapelige universitet
Institutt for teknisk kybernetikk

Sammendrag

Oppgaven tar for seg konstruksjonen av en fotballskytmaskin-prototype. Prototypen skal kunne ta inn et gitt punkt og deretter finne den rette retningen og hastigheten ballen må ha når den forlater maskinen for å treffe det gitte punktet. Oppgaven er en videreutvikling av en tidligere prosjektoppgave [1].

Det ble utformet en funksjonspesifikasjon for maskinen i samråd med oppgavens veiledere. Deretter ble det funnet hvilke komponenter som skal utgjøre maskinen og gjort innkjøp eller innhenting av disse komponentene. Det ble brukt en eksisterende fotballskytmaskin som basis for oppgaven. Det var nødvendig å identifisere motorene på denne maskinen for å vite hvilke reguleringskonstanter som skulle settes inn. Optimaliseringsrutinen ble redusert til å ta for seg to variabler for å vise at minimumskravene for et skudd kunne bli møtt. Disse to var vinkelen opp fra bakken og farten til ballen. Det ble formulert en hypotese for hvordan maskinen gir hastighet til ballen og testet hvordan dette fungerte. Ballen bremset ned hjulene på maskinen slik at hastigheten ble lavere. Dette ble kompensert for. Hastigheten var fortsatt lavere. Dette tilsier at hypotesen var feilaktig og må etterforskes nærmere. Det ble gjort grep for å kompensere for hastighetstapet og maskinen klarte å treffe det ønskede punktet. I videre arbeid bør maskinens utskytningsmodul gjøres mer rede for eller byttes ut fullstendig for å kunne ha mer kontroll på ballens hastighet. Det bør også ses på måter å vite ballens hastighet og når den forlater maskinen.



Masteroppgave

Studentens navn: Torbjørn Alvestrand
Fag: Teknisk kybernetikk
Tittel (norsk): Forbedret treningsmaskin for fotballspillere
Tittel (English): An improved soccer ball machine
Description:

Soccer ball machines (SBM) are available that are designed to “shoot” balls at relatively high speeds to be used as training tools. Typically, these machines can have a designated ball feeder delivering a certain number of balls at specific intervals.

Currently available SBMs fail to exploit their full potential as flexible tools for assessment and exercise of individual soccer skills. A recent project identified the theoretical basis for a more general device, established a functional requirements specification and explored possible ways to realize such a machine. In this assignment you will take this work further towards a physical, operational prototype device.

1. Present a complete, concise and realistic set of functional requirements to be met by your first prototype, based on results from the previous project.
2. Select or specify mechanical, electronic and software modules for all parts of the system. Commercially available components should be used when appropriate, but custom-made modules may be included if justified. Re-use of parts from an existing, available SBM is encouraged when this is compatible with the system requirements.
3. Coordinate the acquisition and manufacturing of the modules and the assembly of the system.
4. As far as time permits, implement the control software necessary to operate the machine in at least one relevant use case, and assess the result in light of the specified requirements.

Veileder(e): Øyvind Stavdahl, Department of Engineering Cybernetics, NTNU
Håvard Lorås, Sør-Trøndelag University College (HIST)
Arve. V. Pedersen, Sør-Trøndelag University College (HIST)

Trondheim, August 2013
Øyvind Stavdahl
Faglærer

Nomenclature

γ	Vinkelen som beskriver spinnaksens stilling i x,y-planet [<i>rad.</i>]
λ	Vinkelen som beskriver spinnaksens stilling i ballens z-retning i forhold til ballens x,y, plan [<i>rad.</i>]
ω	Ballens rotasjonshastighet rundt en akse fast i ballen [$\frac{\text{rad.}}{\text{s}}$]
ω_m	Rotasjonshastigheten til motoren [<i>rad./s</i>]
ψ	Vinkelen mellom z-retning og x,y-planet [<i>rad.</i>]
θ	Vinkelen mellom x- og y-retning [<i>rad.</i>]
B	Luftfriksjonskonstanten [$\frac{V \cdot A}{(\frac{\text{rad.}}{\text{s}})^2}$]
i_a	Strømmen igjennom motoren [<i>Ampere</i>]
J	Treghetsmomentet til motoren [$Kg \cdot m^2$]
k	Counter EMF og fartskonstant [$\frac{V}{\frac{\text{rad.}}{\text{s}}}$]
L	Induktansen i motorkretsen [<i>Henry</i>]
R	Motstanden i motorkretsen [Ω]
V	Påført spenning [Volt]
v	Ballens hastighet [$\frac{m}{s}$]

Innhold

Nomenclature	v
1 Introduksjon	1
2 Bakgrunn	3
2.1 Modell av fotballens bane	3
2.1.1 Utredning av modellen	3
2.1.2 Diskusjon	6
2.2 Implementering	7
2.2.1 Optimaliseringen	7
2.2.2 Optimeringsvariabler	7
2.2.3 Minimeringsfunksjonen (<i>minfunction</i>)	7
2.2.4 Valg av begrensinger	9
2.2.5 Valg av initialbetingelser	10
3 Funksjonsspesifikasjon	11
4 Valg av komponenter	15
4.1 Ballmaskinen	15
4.2 Instrumentering	18
4.2.1 Rotasjonsenkoder	18
4.2.2 Motordriver	19
4.2.3 Strømforsyninger	21
5 Konstruksjon av systemet	23
5.1 Hardware	23
5.2 Systemidentifikasjon	28
5.2.1 Analyse av DC-motorligningene	28
5.2.2 Gjennomføring av sprangrespons-tester	29
5.2.3 Sprangrespons med rotasjonshastighet	30
5.2.4 Sprangrespons uten rotasjon	35
5.2.5 Diskusjon	38
5.3 Regulering av motorene	39
5.3.1 Reguleringen	39
5.3.2 Diskusjon	43

5.4	Programvaren	44
5.4.1	Matlab-modul	45
5.4.2	GUI-modul	48
5.4.3	Kontroll-modul	49
5.4.4	Collect_Data-tråd	50
5.4.5	Roboclaw-modul	54
6	Testing av prototypen	55
6.1	Testkrav	55
6.2	Test 1	56
6.2.1	Gjennomføring	56
6.2.2	Analyse	58
6.2.3	Justeringer før neste test	59
6.3	Test 2	59
6.3.1	Gjennomføring	59
6.3.2	Analyse	60
7	Diskusjon	63
8	Konklusjon	65
9	Fremtidig arbeid	67
A	Testresultater	71
A.1	Testskjema for Test 1	71
A.2	Testskjema for Test 2	72
B	Tilstandstabell	73

1 Introduksjon

Denne oppgaven tar for seg prosessen å lage en fotballskytmaskin hvor maskinen skal skyte ballen til et gitt punkt. Tanken er at det skal til slutt bli en prototype som klarer å bruke en mekanisme til å gi en gitt fart til en ball i en gitt vinkel med bakken slik at den kommer til et gitt punkt.

Rammeverket er å gjøre en kvalifisert nok gjetning og utføre gjetningen, med presisjon, på ballen. For at dette skal fungere må gjetningen basere seg på fysikken. Utvelgelsen av gjetningen må være god nok og det rent maskinelle må gi en såpass rett hastighet og vinkel at dette ikke avviker fra gjetningen. Dette gir da en oppgave å jobbe med.

Siden oppgaven går ut på å bygge en maskin fra grunnen av vil også rapporten være satt opp på det samme prinsippet. Oppgaven begynner med å vise til den prosjektoppgaven som denne masteroppgaven er en videreføring av, og går videre til å etablere en funksjonspesifikasjon for prototypen. Videre blir det sett på hvilke deler som må være med og sammensetningen av disse. Deretter følger en dokumentasjon av hvordan man skal få maskinen til å fungere. Avslutningsvis blir det testing av maskinen og å se fremover på hva som bør og kan gjøres for å lage en enda bedre maskin.

2 Bakgrunn

Denne oppgaven er basert på en tidligere prosjektoppgave ved NTNU. Det er visse aspekter i den oppgaven som er såpass relevant at det er nødvendig å ta med i denne oppgaven også. Det som blir tatt med er utredningen av de matematiske modellene for ballens bane og optimaliseringskoden. Optimaliseringskoden er basert på modellen av ballens bane og optimaliseringskoden brukes som basis for utformingen av prototypen. Disse to aspektene blir gjengitt i dette kapittelet.

2.1 Modell av fotballens bane

For å kunne vite hvordan ballens bane blir, og å kunne påvirke hvor den går, må man ha en matematisk modell hvor man ser på det som påvirker ballen etter den har forlatt maskinen. I dette avsnittet regner vi ut en modell for fotballens bane gjennom lufta. Til slutt diskuteres modellen i et eget avsnitt.

2.1.1 Utredning av modellen

Store deler av modellen er basert på utregninger funnet i andre artikler [2]. Det som har blitt lagt til eller utbedret er å legge til vindpåvirkningen, en ekstra dimensjon i spinnaksen, og en annen, ideell, formel for løftkraften (Magnus force) som inneholder ballens spinn[3]. Det er også hentet inn en formel for utviklingen av ballens spinn[4] hvor vi antar en ideel turbulent flyt.

Vi antar at ballen skytes ut i y -retning. Dette gir en vektor med størrelse v . For å beskrive avvik fra retning y må vi innføre vinkelen ψ for å beskrive avvik i z -retning og vinkelen θ for avvik i x -retning. Dette gir retningsvektoren

$$\tau = \cos\psi \sin\theta i + \cos\psi \cos\theta j + \sin\psi k \quad (1)$$

Ved å legge til størrelsen av vektoren kan man finne beskrivelser for

farten i x, y og z retning.

$$\dot{x} = v \cos \psi \sin \theta \quad (2)$$

$$\dot{y} = v \cos \psi \cos \theta \quad (3)$$

$$\dot{z} = v \sin \psi \quad (4)$$

Med disse funksjonene får man definisjonen for v .

$$v = |v| = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \quad (5)$$

Ballen antas også å ha spinn rundt en gitt akse som er fast i ballen. For å finne denne utnyttet jeg sfæriske koordinater og at ballen har en fast radius. Dette gir vektoren

$$\sigma = \sin \lambda \cos \gamma \mathbf{i} + \sin \lambda \sin \gamma \mathbf{j} + \cos \lambda \mathbf{k} \quad (6)$$

Her er λ vinkelen mellom z_{ball} og planet (x_{ball}, y_{ball}) og γ er vinkelen som beskriver forholdet mellom x_{ball} og y_{ball} . Vi bruker ikke de kartesiske koordinatene videre.

Vi innfører også vektoren for vind μ siden dette kan påvirke ballen. Denne har bare en retning i x - og y -retning.

$$\mu = \cos \beta \mathbf{i} + \sin \beta \mathbf{j} \quad (7)$$

Dette antas som målt verdi og tas ikke inn som en tilstand i systemet.

Videre finner vi kreftene som påvirker ballen mens den er i luften. Det består av tyngdekraft, vindkraft F_w , friksjonskraft som kommer av bevegelse i et fluid F_d , kjent som drag force, og løftkraft fra spinnet til ballen F_l , som er kjent som Magnus force.

$$F_{sum} = m\mathbf{g} + F_w + F_d + F_l \quad (8)$$

Hvor

$$\mathbf{g} = -g\mathbf{k} \quad (9)$$

som er tyngdekraftakselerasjonen. Påvirkningen av vinden blir

$$F_w = \rho A v_w^2 \mu \quad (10)$$

v_w er her vindhastigheten.

Friksjonskraften fra bevegelsen gjennom vinden blir

$$F_d = -\frac{1}{2}\rho AC_d(v^2\tau - v_w\mu) \quad (11)$$

og løftkraften fra spinnnet blir

$$F_l = \frac{8}{3}\pi^2 r^3 \rho \omega (v\sigma \times \tau - v_w\sigma \times \mu) \quad (12)$$

Her er ρ lufttettheten. r og A er henholdsvis radiusen og arealet til ballen og C_d er friksjonskoeffisienten. ω er ballens vinkelhastighet rundt en akse fast i ballen. Denne starter med en initialhastighet ω_0 og akselerasjonen er gitt av

$$\dot{\omega} = -8\pi\eta r^3 \omega^2 \quad (13)$$

Dette tilsvarer turbulent flyt rundt objektet, og retningen til denne defineres ut fra retningen til σ , som er akse den går rundt, og vil gi spinn mot klokka når verdiene er positiv og med klokka når verdiene er negativ. Konstantene er hentet fra [5]

Vi forbedrer lesbarheten av ligningene ved å innføre konstantene

$$k_l = \frac{8\rho\pi^2 r^3}{3m} \quad (14)$$

$$k_d = \frac{\rho AC_d}{2m} \quad (15)$$

$$k_w = \frac{\rho A}{m} \quad (16)$$

$$k_\omega = -8\pi\eta r^3 \quad (17)$$

Ved å sette inn disse i (8) i tillegg til kreftene, og å løse opp for de

forskjellige retningene får vi ligningene

$$\begin{aligned} \ddot{x} = & k_w v_w \dot{x}_w + k_l \omega [(\sin \lambda \sin \gamma \dot{z} - \cos \lambda \dot{y}) + \dot{y}_w \cos \lambda] \\ & - k_d (v^2 \cos \psi \sin \theta - v_w^2 \cos \beta) \end{aligned} \quad (18a)$$

$$\begin{aligned} \ddot{y} = & k_w v_w \dot{y}_w + k_l \omega [(\sin \lambda \cos \gamma \dot{z} - \cos \lambda \dot{x}) - \dot{x}_w \cos \lambda] \\ & - k_d (v^2 \cos \psi \cos \theta - v_w^2 \sin \beta) \end{aligned} \quad (18b)$$

$$\begin{aligned} \ddot{z} = & k_l \omega [(\sin \lambda \cos \gamma \dot{y} - \sin \lambda \sin \gamma \dot{x}) - (\sin \lambda \cos \gamma \dot{x}_w - \sin \lambda \sin \gamma \dot{y}_w)] \\ & - (k_d v^2 \sin \psi + g) \end{aligned} \quad (18c)$$

$$\dot{v} = \frac{\ddot{x} + \ddot{y} + \ddot{z}}{v} \quad (18d)$$

$$\dot{\omega} = -k_w \omega^2 \quad (18e)$$

Dette beskriver hele ballmodellen og ser også på vindens påvirkning. Det er verdt å merke seg at vinden påvirker både løftkraften og dragkraften i tillegg til å påvirke ballen direkte. Vi antar ideelle tilstander (vindstille) og setter $v_w = 0$. Dette gir ligningsettet som blir brukt videre.

$$\ddot{x} = k_l \omega (\sin \lambda \sin \gamma \dot{z} - \cos \lambda \dot{y}) - k_d (v \dot{x}) \quad (19a)$$

$$\ddot{y} = k_l \omega (\sin \lambda \cos \gamma \dot{z} - \cos \lambda \dot{x}) - k_d (v \dot{y}) \quad (19b)$$

$$\ddot{z} = k_l \omega (\sin \lambda \cos \gamma \dot{y} - \sin \lambda \sin \gamma \dot{x}) - (k_d v \dot{z} + g) \quad (19c)$$

$$\dot{v} = \frac{\ddot{x} + \ddot{y} + \ddot{z}}{v} \quad (19d)$$

$$\dot{\omega} = -k_w \omega^2 \quad (19e)$$

2.1.2 Diskusjon

Dette ligningsettet er vurdert for ideelle verdier. Det betyr at ballen har en uniform overflate, noe som ikke er forenlig med de ballene som brukes i fotball i dag, som nevnt i [1]. Vi anser det likevel som mulig å få gode resultater med dette settet siden det omfavner mye av det som påvirker ballen. Det er tydelig at de initielle verdiene til retning, hastighet, spinn og spinnakse er de vi kan påvirke ballen med. Derfor fokuserer prosjektet videre på hvordan man skal finne de rette initielle verdiene for å gjøre det vi ønsker maskinen skal gjøre.

2.2 Implementering

I dette kapitlet tar vi for oss hvordan de ønskede initialbetingelsene ble funnet. Vi ser først på hvordan systemet ble løst i matlabkode og viktige aspekter ved den koden. Deretter ser vi på parametrene og hvorfor de gitte verdiene ble satt slik de ble satt. Til slutt kommer et diskusjonsavsnitt.

2.2.1 Optimaliseringen

For å skrive optimaliseringskoden ble Matlab brukt. Dette er fordi det er et verktøy med gode funksjoner for simulering, optimalisering og plotting av data i tre dimensjoner. Det ble skrevet to egne funksjoner. *ballpathfunction* er en funksjon som beskriver ballbanen og brukes med *ode45* [6] for å gjøre en numerisk simulering. *minfunction* beskriver den funksjonen *fmincon* [7] bruker til å optimalisere initialverdiene for å oppnå ønsket resultat. Videre i denne delen vil det bli tatt for seg valg av optimeringsvariabler og startpunkt, og valg av begrensninger for optimaliseringen. Funksjonen *ballpathfunction* er basert på ligningssettet 19 og vil derfor ikke bli sett nærmere på i denne delen.

2.2.2 Optimeringsvariabler

For å kunne påvirke alle de forskjellige parametrene som går inn i systemet var det mest optimalt å se på vinklene θ, ψ og hastigheten v . Dette er fordi de forskjellige hastighetene $(\dot{x}, \dot{y}, \dot{z})$ er relatert til hverandre gjennom vektor τ (1). Dermed kan man variere dem uten å få verdier som ikke er mulige å få til med den dynamikken vi har beskrevet tidligere. For å ha alle de påvirkende mengdene som kan settes til en initiell verdi i (19) tar vi også inn ω_0, λ , og γ , som dukker direkte opp i ligningssettet. Dermed er det mulig å endre alle parametrene for å finne det ønskede resultatet.

2.2.3 Minimeringsfunksjonen (*minfunction*)

For å løse optimaliseringsproblemet definerer vi en funksjon som simulerer banen med de gitte initialbetingelser og finner den minste avstanden fra det ønskede, brukerdefinerte punktet og banen. Som vi så i tabell 7 så inngår posisjonen i alle scenarier man kan ha med de gitte tilstandene.

Dette er dermed grunnformen til den ønskede funksjonen.

$$f(x) = g_p(\min((x_i - x_f)^2) + \min((y_i - y_f)^2) + \min((z_i - z_f)^2)) \quad (20)$$

g_p er her en designkonstant for å oppnå ønskelig resultat innenfor de toleranser som er satt som standard i funksjonen. For å realisere denne funksjonen simulerer vi skuddet med den numeriske løseren *ode45*[6], som er en funksjon i Matlab. Vi går igjennom den simulerte banen iterativt og finner det punktet som ligger nærmest. Dette tar seg av posisjonsoptimeringen.

For å påvirke farten til ballen endrer vi basisfunksjonen vår. Ved å legge til hastigheten i det gitte punktet, som er det punktet som er nærmest der man ønsker å plassere skuddet i funksjonen, har man mulighet til å enten maksimere eller minimere hastigheten. Leddet blir seende slik ut

$$g_v v(i_{final})$$

g_v er her en designkonstant som enten kan settes større eller mindre enn null alt ettersom om man trenger å maksimere eller minimere hastigheten i tillegg til å kunne brukes som vektning. For eksempel når man skal gi en pasning er det ønskelig at ballen ikke har for høy fart. Da kan det være lurt å minimere hastigheten til forskjell fra skudd som ofte skal være så harde som mulig. Maskinen forholder seg til de gitte maksverdier for fart, noe som gjør at vi kan begrense verdiene for hastighet til et område som er vanlig for fotballspillere å kunne klare, noe som er oppgitt i funksjonspesifikasjonen.

Spinnet påvirkes på samme måte som farten. Fra ligningssettet (19) ser vi at *omega* er ikke-økende og bare påvirket av ω . Dette gjør at uansett hvilket punkt man tar inn vinkelhastigheten for i funksjonen $f(x)$ så får man det maksimale spinnet. For å holde på enkelheten i koden bruker vi det samme punktet som i hastighetsleddet. Hele funksjonen blir da.

$$f(x) = g_p(\min((x_i - x_f)^2) + \min((y_i - y_f)^2) + \min((z_i - z_f)^2)) + g_v v(i_{final}) + g_\omega \omega(i_{final}) \quad (21)$$

For å variere hva vi tar hensyn til endrer man på de forskjellige designkonstantene (g_i) og setter dem til null om man ønsker å ekskludere den ene parameteren.

Funksjonen kan da oppsummeres i algoritmen:

1. Simuler systemet med x_k
2. finn minimum avstand fra banen til punkt p_f og legg dette til $f(x)$
3. finn hastigheten i punktet fra forrige aksjon
 - (a) om $g_v > 0$ legg farten til $f(x)$
 - (b) ellers trekk farten fra $f(x)$
4. finn spinnet i punktet fra aksjon 2
 - (a) om $g_\omega > 0$ legg farten til $f(x)$
 - (b) ellers trekk farten fra $f(x)$
5. om $f(x) < \infty$ returner $f(x)$

2.2.4 Valg av begrensinger

I tabell 7 ser man at 'Lav pasning' og 'Direkteskudd' har krav om at vi bestemmer mer direkte hvor høyt ballen skal kunne gå; altså hvor stor vinkelen ψ kan være. *fmincon* gir muligheten til å bestemme høyeste og laveste verdi for begge disse og gjør det trivielt å bestemme når systemet designes. Det kan være det er fysiske parametre ved maskinen som også begrenser de optimeringsvariablene. Vi antar forholdsvis ideelle forhold for vinkelpåvirkningene θ , ψ , λ og γ og har satt grensene mellom 0 og 2π , som tilsvarende alle vinkler.

For å oppfylle kravet om å sende baller uten spinn må man bruke begrensninger. Om man setter $\omega = 0$ som en av likhetsbegrensningene så vil dette gjøre at ω forblir lik null og vi vil derfor ikke få spinn. For å forenkle gjør vi det samme for λ og γ siden disse ikke har noen effekt på ligningene når vi ikke har spinn.

Det er nevnt i funksjonspesifikasjonen at det skal være maksimumsverdier som er forenlige med det en vanlig fotballspiller kan klare å få til. Det vi fant var at den maksimale hastigheten oppnådd i [8] er 29.28 m/s, men er mindre (25.83 m/s) når man har spinn med på ballen på grunn av mindre kontaktflate og at fotens kontaktflate med ballen er lengre unna midten. Disse verdiene bruker vi som maksverdier i tillegg til spinnet funnet i samme artikkel. Vi endrer også på begrensningene om det er med eller uten spinn for å få en mer naturtro avlevering av ballen.

2.2.5 Valg av initialbetingelser

For å velge gode initialbetingelser bør vi se på minimeringsfunksjonens behov og de gitte områdene. For det første er det nødvendig å ha en hastighet for å få en bane som kommer seg noe sted. Ballen må også skytes opp i fra bakken; altså må vinkel ψ være større enn 0, men mindre enn π . Det er også lurt å gå ut fra at man retter seg mot det ønskede punktet fra starten av. Ved å bruke den rette linjen fra startpunktet til det ønskede punktet kan vi bestemme θ ved å finne vinkelen i x,y -planet som utgangspunkt for θ . ψ setter vi som basis til å være $\frac{\pi}{4}$ så lenge punktet har en z -verdi lik 0. Ellers gjør vi det samme som for θ . Dette gjør at den første gjetningen er bedre. Spinnet kan vi velge vilkårlig så lenge det er et mulig punkt i de gitte begrensningene.

3 Funksjonsspesifikasjon

For å kunne lage en prototype er det behov for å vite hva prototypen skal gjøre. I dette kapitlet blir det sett på hva som er funksjonsspesifikasjonen til maskinen. Det blir først oppsummert i en liste med hva som skal være med og hva som bør være med. Det vil da videre bli sett på hvorfor det er disse som tas med i funksjonsspesifikasjonen og hvorfor visse ting ble utelatt.

Listen er bestemt i samråd med veilederene og tar utgangspunkt i funksjonsspesifikasjonen som ble presentert i [1]. Listen ble laget da det var bestemt at ballmaskinen beskrevet i 4.1 som blir brukt videre skulle være en del av prototypen. Dette gir føringer for funksjonsspesifikasjonen.

Maskinen skal:

- Kunne skyte en fotball som oppfyller kravene til FIFA til et ønsket punkt. Et godkjent skudd er at ballen havner innenfor en sirkel med radius på en meter hvor sentrum er det ønskede punktet.
- Kunne sette hastigheten som ballen skal forlate maskinen med til alle verdier i området $0.0 < v \leq 27.0$ avrundet til en desimal (z - og y -retning).
- Kunne sette vinkelen skuddet går i i forhold til et horisontalt plan, manuelt med en oppløsning på 1 grad.
- Ha en kalibreringsrutine for å optimalisere maskinen for en gitt balltype og lagre den til senere bruk.

Maskinen bør:

- Kunne sette spinnhastigheten til ballen så nøyaktig som nødvendig.
- Kunne legge inn flere forskjellige avleveringsverdier (fart, spinn, retning) for ballen til flere posisjoner.
- Kunne laste flere baller i maskinen.
- Kunne sende flere baller etter hverandre.

Det blir her fokusert på å definere minimumskravet for at man skal kunne oppnå en fotballskytmaskin som klarer å treffe et ønsket punkt. Vi antar vinden er lik null for å forenkle ligningene, og fordi det ikke blir fokusert på å vite hva vinden er. Dette gjør at det er mest naturlig å gjennomføre tester innendørs.

Maskinen som blir brukt som basis for prototypen har ikke muligheten til å variere på spinnaksen. Dette ble ikke sett på som essensielt for å gjennomføre skudd innenfor en mindre sirkel og ble derfor ikke vurdert som krav. Med tanke på at spinnaksevinkelen er satt konstant får vi fastsatt flere av de variable som funksjonene definerte i (2.1). Det nye ligningsettet blir

$$\ddot{x} = -k_l \omega \cos \lambda \dot{y} - k_d (v \dot{x}) \quad (22a)$$

$$\ddot{y} = -k_l \omega \cos \lambda \dot{x} - k_d (v \dot{y}) \quad (22b)$$

$$\ddot{z} = -(k_d v \dot{z} + g) \quad (22c)$$

$$\dot{v} = \frac{\ddot{x} + \ddot{y} + \ddot{z}}{v} \quad (22d)$$

$$\dot{\omega} = -k_\omega \omega^2 \quad (22e)$$

Dette gjør at man ikke påvirker z-retning med spinn og bare påvirker x- og y- dynamikken. Siden prototypen skal vise et prinsipp i det at man skal kunne finne et punkt man kan treffe med ballen. Ved å kunne gjøre det i en dimensjon (y -retning) vil det tilsa at det er mulig i flere dimensjoner. For at ballen skal forflytte seg i y -retning må ballen også ha fart z -retning ut fra maskinen. For å redusere antall frihetsgrader vi kan påvirke er det logisk å bare forholde seg til ψ og farten til ballen. Dette gjør at ballmaskinen ikke har en variasjon i x, y -planet utenom det man får av spinn. Når man i tillegg setter funksjonen til å ha null spinn ($\omega = 0$) så blir det ingen påvirkning i x -retning, som vist i ligning (23).

$$\ddot{x} = -k_d v \dot{x} \quad (23a)$$

$$\ddot{y} = -k_d v \dot{y} \quad (23b)$$

$$\ddot{z} = -k_d v \dot{z} - g \quad (23c)$$

$$\dot{v} = -\frac{(k_d v (\dot{x} + \dot{y} + \dot{z}) + g)}{v} \quad (23d)$$

$$\dot{\omega} = -k_\omega \omega^2 \quad (23e)$$

Ved å kunne bevise at det er mulig å treffe det punktet med verdier hentet fra dette ligningssettet vil det si det er mulig å treffe et punkt vi ønsker. Dette er minimumskravet for at det skal være en akseptabel prototype. Fra ligningssettet (23) ser man at det er en konstant som er ukjent. Dette er bakgrunnen for kalibreringsbehovet. Hver ball har sin egen konstant og det er derfor behov for å få et anslag av denne verdien. For å unngå å lagre alle verdier for alle baller fokuseres det på å lagre verdier for gitte ballmodeller da disse har mye av de samme egenskapene og derfor vil få mye likhet i denne konstanten.

Det å laste flere baller i maskinen kommer gratis fra at maskinen har dette implementert fra før av. Dette er dog ikke essensielt for at maskinen fungerer som den skal og faller derfor under bør-kategorien. Maskinen har i dag også en manuell metode for å variere utskytningsvinkel opp fra det horisontale planet. For å være sikker på hvilken vinkel man stiller inn på er det ønskelig å kunne ha en metode for å vite hvilken vinkel man får. Ved å gjøre en avveing mellom brukervennlighet og nødvendig vinkelnøyaktighet stiller man inn vinkelen med en sikkerhet som er toppen én grad. Dette gjør at det er lettere å stille inn den ønskede utgangsvinkelen uten at dette skal gå utover sannsynligheten

Om det blir tid til det så er det mulig å se på spinn. Det antas at om man kan sette hastigheten til ballen med høy nøyaktighet så er det mulig å lage programvar for å lage spinn på ballen også. Siden det ikke er et skal-krav blir det ikke prioritert å teste for dette.

Dette blir grunnlaget for maskinen. Videre blir det sett på hvordan denne funksjonsspesifikasjonen skal gjennomføres. Først dokumenteres komponentene.

4 Valg av komponenter

For å kunne tilfredstille funksjonspesifikasjonen må det lages en maskin. Siden dette er en prototype blir det fokusert på deler som allerede eksisterer og fungerer og derifra sette dem sammen til et endelig produkt. I denne delen blir det sett nærmere på ballmaskinen som ligger til grunn for prototypen, de valgte rotasjonsenkoderne og motordriveren.

4.1 Ballmaskinen

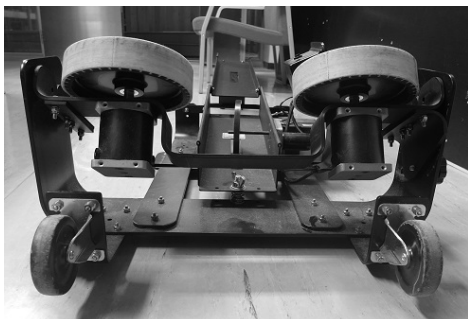
Det brukes en allerede konstruert ballmaskin . Ballmaskinen er av modellen Pro Trainer fra det amerikanske firmaet Sports Tutor. De produserer ballskytemaskiner for tennis, baseball, softball, volleyball og fotball med størst fokus på baseball og tennis.

Maskinen er laget for å kunne ha flere baller på en renne som mater, ved hjelp av en dispensermotor, en utskytningsmekanisme bestående av to hjul. Denne rennen har en liten vinkelforskjell fra underlaget, som gjør at ballene naturlig ruller mot dispensermotoren. Dette gjør at ballen er i bevegelse når den kommer til maskinen. Det må vurderes om det er en feilkilde for startverdiene til ballen senere.

Dispensermotoren er en liten elektrisk motor som har festet en plastskive hvor en tredjedel av skiven er fjernet (se figur (1(b))). Dette er for å kunne kontrollere om en ball skal inn i maskinen eller ikke. For å bestemme frekvensen på hvor mange baller som skal leveres i minuttet er motoren koblet til et hjul hvor man kan øke og senke farten på motoren. Det er ikke spesifisert hvor fort motoren snurrer ved de forskjellige hastighetene og det er ikke mulig å sette en gitt frekvens på balleveringene med dagens oppsett.

Batteriene som forsyner systemet med strøm har 12 volt og 7 ampere timer hver. De er parallellkoblet, noe som betyr at maksimumspenningen til motorene er satt til 12 volt. Dette kommer til å bli holdt som maksimumspenning videre også for å unngå skader på motorene. Batteriene var foreldet da maskinen ble hentet på Høyskolen i Sør-Trøndelag (HiST) og må derfor byttes ut.

Utskytningsmodulen er to motorer som har et hjul hver med en diameter på 10 cm festet til akselen. Rundt dette hjulet er det et lag med gummi som er i kontakt med ballen. Motstanden i motorene blir målt og funnet til å være 0.5Ω .



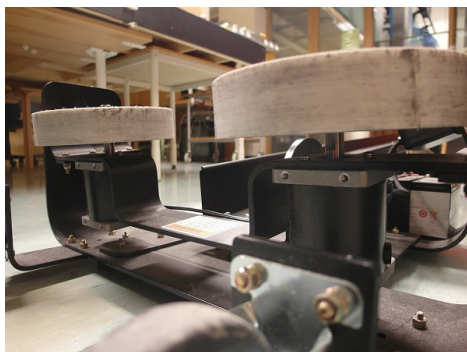
(a) Ballmaskin



(b) Dispenser



(c) Kontrollpanel



(d) Utskytningsmekanisme

Maskinens elektriske komponenter er:

- To motorer som er koblet til hvert sitt hjul med et gummibånd rundt hjulet. Disse står for aksereelasjonen av ballen.
- Et brukerpanel som inneholder:
 - En Av/På bryter for strømmen
 - En Reset knapp
 - Et hjul som stiller inn hvilken fart du vil ha på en skala fra 1 til 10 hvorav 1 er lavest fart og 10 er høyest fart.
 - Et hjul som stiller inn hvilket intervall du vil at ballene skal komme på. Dette har en ukjent oppløsning og minimum- og maksimumintervall.
 - Strøminngang for ekstra batteripakke og lader.

Kretskortet bak kontrollpanelet (se figur (1(c))) virker å koble strømkilden videre til motorene gjennom de dedikerte instillingshjulene. Avfyrimekanismen (se figur (1(d))) har begge motorene koblet til samme signal fra maskinen. Dette signalet er regulert ved hjelp av kontrollhjulene. Dette gjør at denne maskinen ikke kan kontrollere eller variere spinnen. Om det skapes spinn med dette oppsettet er det på grunn av forskjeller i motorene eller forskjeller i overflaten på hjulene.

De funksjonskravene som påvirker denne maskinen er å kunne styre hastigheten til ballen til en ønsket hastighet og å kunne stille vinkelen til det horisontale planet mellom 0 og 90 grader. Maskinen kan stille vinkelen i det ønskede området uten å gjøre noen modifikasjoner utenom benevning av de forskjellige vinkelposisjonene. Maskinen har også mulighet til å stille farten til ballen på en enkel måte så fort det er mulig å sette hastigheten til motoren. Dette krever en bedre motordriver og noe å måle hastigheten til de forskjellige motorene med. Om hastigheten til motorene er kjent er det også mulig å finne ut hvordan hastigheten til ballen påvirkes. Dette gjør at det er mulig å kontrollere hastigheten til ballen, som ønsket i funksjonsspesifikasjonen. Det at det er lagt opp til en automatisk mating av baller til maskinen er en bonus som gjør det enklere å få like startverdier for ballene det skal testes på. Maskinen er eid av HiST og dermed ikke kjøpt inn til denne oppgaven.

4.2 Instrumentering

For å kunne styre hastigheten til motorene på ballmaskinen må hastigheten måles og det må være mulig å bruke denne informasjonen til gi rett spenning på motorene. Dette gjør at det er behov for to komponenter for hver motor. En motordriver og en rotasjonsenkoder.

4.2.1 Rotasjonsenkoder

Det ble først fokusert på å finne en enkoder som har plass imellom hjulene og selve motoren. Dette krevde en rotasjonsenkoder som det var mulig å plassere rundt motorens aksel på 16 millimeter og som samtidig var lavere enn 25 millimeter. Dette viste seg å være en umulig oppgave da de fleste med plass til en såpass stor aksling var mye høyere enn det var plass til. Den neste muligheten ble da å feste rotasjonsenkoderne under motorene ved å feste en tynnere metallpinne inn i akslingen, som var synlig under motoren. Dette gjorde at den nødvendige diameteren på rotasjonsenkoderen ble mye mindre. Prisen ble da også lavere enn de som hadde plass til akslingen ved den forrige løsningen og det var flere modeller som kunne fylle kravene. For å begrense antallet muligheter ble det sett nærmere på hva som kreves av funksjonspesifikasjonen. For å vite hastigheten til ballen må vi vite hva hastigheten til motoren blir gjennom enkoderen. Hastigheten til enkoderen måles ved å telle antall pulser enkoderen gir per sekund. Ved å vite hvor mange pulser det er per rotasjon for enkoderen er det enkelt å finne hastigheten. Hvor mange pulser det er per rotasjon sier noe om hvor god oppløsning det er. I funksjonspesifikasjonen er det et krav at hastigheten kan leses av til en oppløsning på 0.1 m/s. Dette gjør at det er en minimumsverdi på antall pulser per rotasjon enkoderen må gi. Denne finner vi ved å finne hvor høy vinkelhastighet 0.1 m/s tilsvarer. Deretter omgjøres benevningen til rotasjon per sekund før man finne hvor mange pulser per rotasjon dette tilsvarer. Utregningen blir da til slutt:



Figur 1: Valgt rotasjonsenkoder

$$v = 0.1[m/s] \quad (24)$$

$$r = 0.1[m] \quad (25)$$

$$n \geq \frac{4\pi r}{v} \quad (26)$$

$$n \geq 40 \quad (27)$$

$$(28)$$

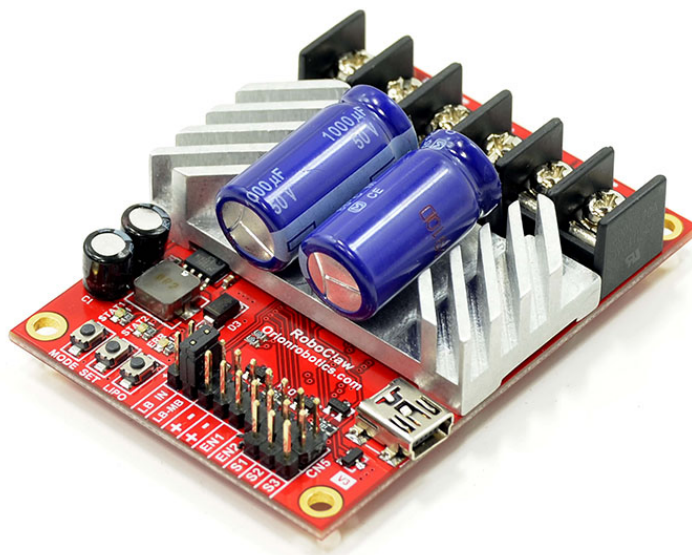
som betyr at enkoderen må ha en oppløsning som er høyere enn 40. Dette ble avrundet oppover til nærmeste heltall. Dette er et veldig lavt tall og minsket ikke antallet kandidater mye. Det betyr at kostnader og tilgjengelighet ble utslagsgivende. I tillegg er det ønskelig å ha mulighet til å fjerne enkoderene om maskinen skal brukes i sin originale form. Derfor var bare enkodere av typen 'hollow shaft' aktuelle og valget falt på en modell fra Kübler. Denne ble solgt av Elfa som hadde to på lager og til en forholdsvis lav pris med kort leveringstid.

4.2.2 Motordriver

For å kunne bruke fartsmålingen fra enkoderene til å styre motorene kreves det en motordriver. Motorene er likestrømsmotorer med børster, så driveren må være laget for denne typen motorer. Ut i fra spenningen til maskinbatteriet på 12 Volt og motstanden i motoren på 0.5 Ohm kreves det at driveren tåler en spenning på 24 Ampere. For å unngå tidsforsinkelser,

som kan skape problemer for reguleringen, er det ønskelig å ha mulighet til å kjøre reguleringen så nær motordriveren som mulig. Minimumskravet til reguleringen er å regulere motorene hver for seg ved å manipulere spenningen. Mest versatilitet får man da ved å ha mulighet for en PID-regulator. Dette betyr også at enkoderne sender signalet direkte til kortet som gjør at kortet må ha muligheten til dette. Kortet bør kunne motta beskjeder fra en PC og sende informasjon, for eksempel hastigheten til motorene, tilbake. Det er også ønskelig at prisen ikke er for høy. Om det er mulig å få regulering av begge motorene på samme kort kan dette spare mye penger og skape mindre problemer med flere kommunikasjonslinjer.

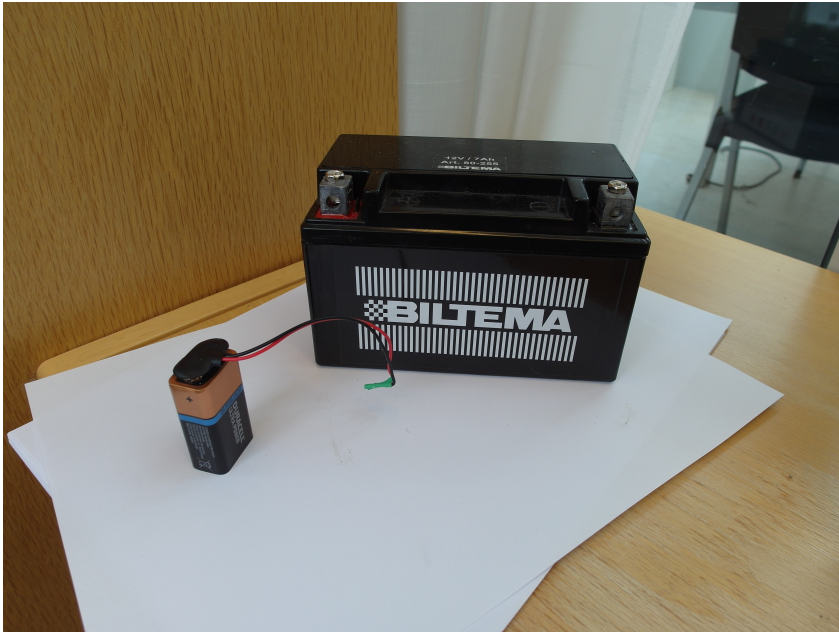
Det ble først sett på en motordriver fra Maxxon Motors. Denne møt- te de fleste av kravene som er blitt beskrevet tidligere. Denne var dog en smule for dyr med tanke på at den bare hadde mulighet til å styre en motor om gangen og en prislapp på 425 USD per stk. Det ble videre sett på motordrivere fra hobbymarkedet siden dette er et marked som baserer seg på store kvanta og ikke har like mange krav til å unngå for mye støy. Roboshop.com er en nettbutikk som spesialiserer seg på elektronikkdeler for akkurat dette markedet. Siden mye av hobbymarkedet baserer seg på konstruksjon av roboter på to hjul var det lett å finne motordrivere som var laget for å styre to motorer samtidig. Her ble den motordriveren som blir brukt videre kjøpt. 'Roboclaw 2x30A Motor Controller with USB' har mulighet for å styre to motorer på opptil 30 ampere hver. Den har integrert PID-reguleringsløyfer, seriell kontaktmulighet, enkoderinnganger og koster bare 125 USD. Det at den er på størrelsen med et kredittkort er også et pluss.



Figur 2: Motordriveren

4.2.3 Strømforsyninger

Det som krever strømforsyning i det tenkte oppsettet er motorene på ballmaskinen (både dispensermotoren og utskytningsmotorene) og logikken til motordriveren. Basert på det batteriet som brukes på den originale maskinen trengs det en spenning på 12 volt og en god batterilevetid. Ballmaskinen hadde opprinnelig et konstant pådrag på motorene. Motorene i protypen trenger ikke dette siden de skal reguleres og kan settes til null hastighet når de ikke brukes. Dette gjør at batterilevetiden forbedres og det er ikke behov for å ha samme kapasitet som det batteriet som var der



Figur 3: Batteriene som ble brukt

fra før. Kortet legger opp til muligheten for å bruke motorbatteriet som strømforsyning til logikken. Dette kan skape støyp problemer i logikken og oppkoblingen til PC'en, men for enkelhet skyld er det dette som velges å bruke. Om det ikke fungerer er det mulig å koble inn et mindre batteri på to headerpinner uten større modifikasjoner. Det må være mulig å koble ut batteriet fra motordriveren om motorene ikke responderer på vanlig kontroll. En lampebryter brukes for å kontrollere strømforsyningens oppkobling til kortet. Batteriet som velges er et motorsykelbatteri fra Biltema med 12 volt og 7 Amperetimer. Dette skal være nok for dette formålet.

5 Konstruksjon av systemet

For at den tenkte maskinen skal bli en realitet er det behov for å sette delene sammen. Det er også behov for identifisering og regulering av motorene til maskinen. I tillegg må programvaren som binder det hele sammen bli programmert. I dette kapitlet ser vi på selve sammensetningen av de forskjellige modulene, systemidentifikasjonen av motorene, reguleringen av motorene og til slutt programvaren til systemet.

5.1 Hardware

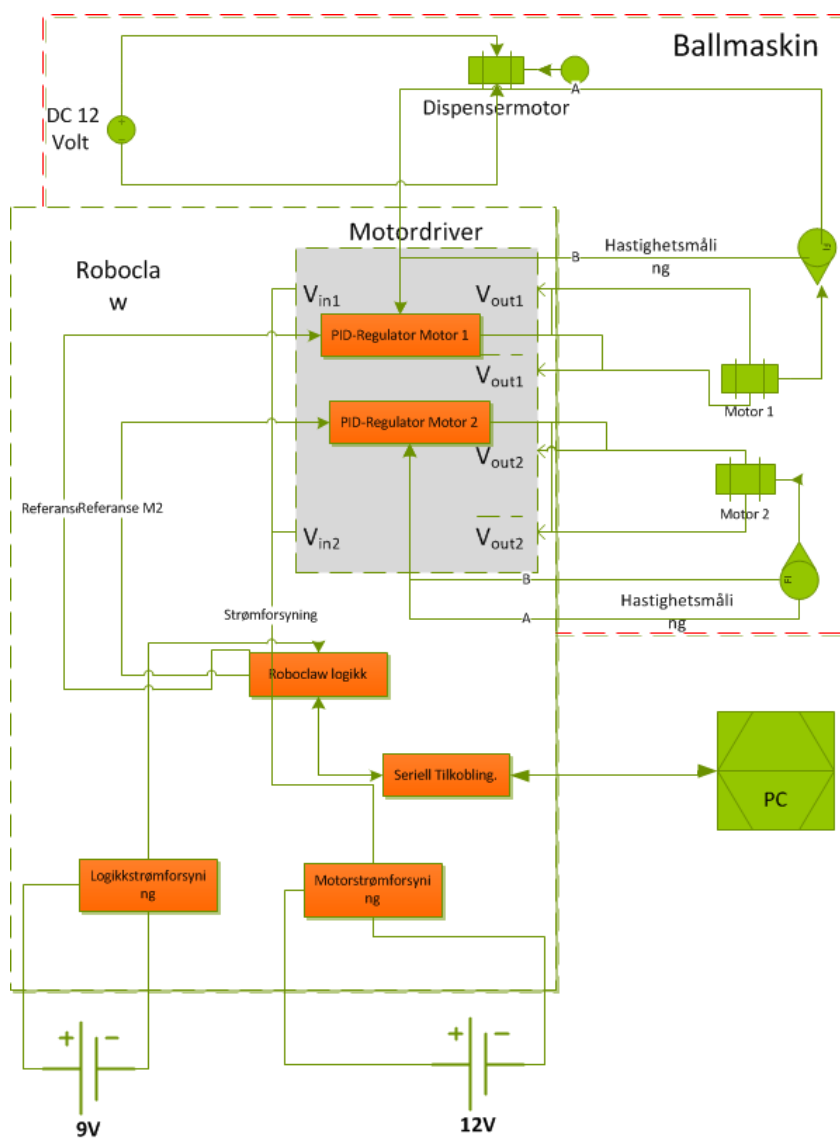
Det rent hardware-relaterte omhandler oppkoblingen av de forskjellige komponentene (2x motorer, 2x Enkodere, Motordriver-kortet, og strømfor-syninger). I tillegg måtte dette kobles opp mot en PC. Siden kortet hadde en USB-tilgang på kortet ble denne antatt til å være god nok til å bruke. I bruk ble denne utsatt for mye støyproblemer. Dette skapte problemer for kommunikasjonen med kortet da den serielle koblingen fikk timeout-problemer og man mister muligheten til å snakke med kortet. Etter en mailkorrespondanse med produsenten av kortet ble en USB til TTL seriell tilkobling kjøpt inn. Dette fjernet alle disse problemene. Kabelen kan sees i figur (7(b)).

For å minske annen eventuell støy ble det koblet til et 9-volts batteri som ga strøm til logikken. Dette gjorde at eventuelle sprang i motorstrømmene som resulterer fra last ikke påvirket logikken. Av sikkerhetsgrunner var det nødvendig å implementere en strømbryter på motorbatteriet. På grunn av rask variasjon i strømmen i motoren er det ønskelig at strømkablene til motorene blir så kort som mulig. Kabelen er en ujordet lampeledning. Dispensermotoren og dens oppkobling er en strømforsyning som bruker strømmettet. Strømforsyningen er laderen til den originale ballmaskinen. Denne gir nok strøm til motoren for å oppfylle funksjonen den har. Strømkablene til motoren og logikken er vist i figur (7(a)).

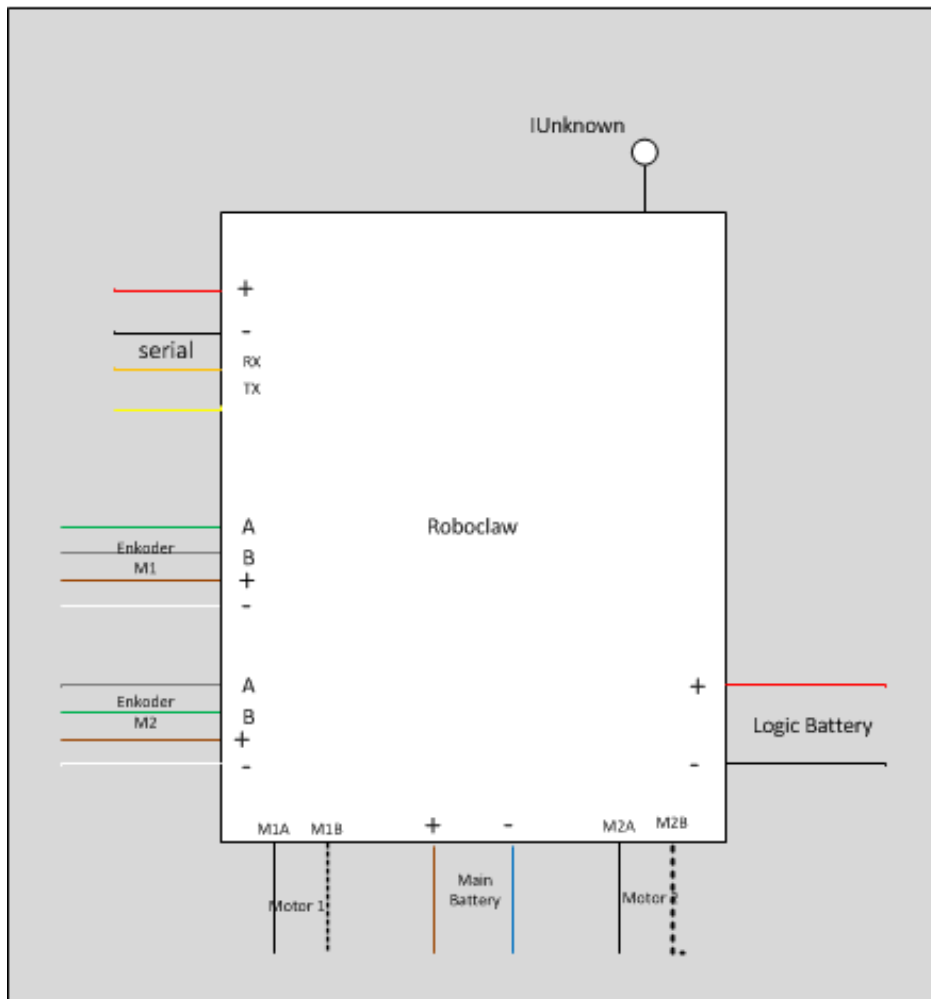
Enkoderene blir koblet til hver sin motor ved hjelp av et gummistrikk og to hjul som sett i figur (6). Kablene ble lagt langs motorkablene og festet til disse. Kablene ble terminert med headerhylser siden dette er kontakten som er på motordriverkortet. Enkoderen har fire kabler som ikke skal brukes. Disse blir tvinnnet rundt hverandre og hold ubrukt. Uisolerte deler av de ubrukte kabelene ble også fjernet. Ingenting tilsa at disse skapte problemer

for avlesningen av enkoderene eller andre deler av systemet. Det som kunne ha skapt problemer for avlesningen er at enkoderen er koblet til motoren med en gummistrikk. På grunn av strikkens elastisitet kunne dette ha skapt svingninger i målingene noe som ville gitt forstyrrelser for systemet. I målinger har det ikke vært mulig å se slike svinginger. Dette kan enten være på grunn av at strikken ikke er veldig elastisk eller at dempingen i strikken er stor nok til å fjerne eventuelle svinginger det før det merkes.

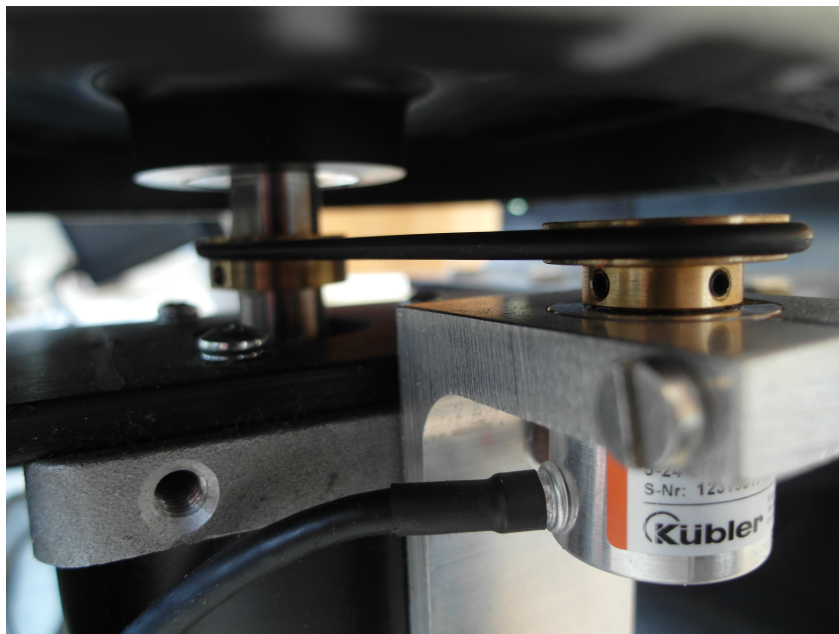
Prototypens oversikt er vist i (4). Dette gir en grov oversikt over hvordan komponentene kobles sammen. Oversikten er symbolsk og viser ikke til faktiske koblinger. Koblingen med andre systemer på motordriverkortet er oppsummert i figur (5).



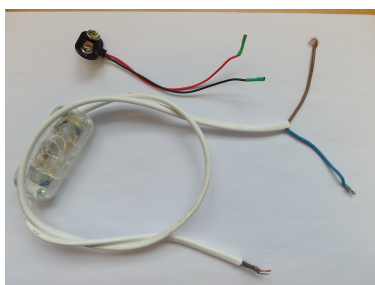
Figur 4: Oversikt over systemet



Figur 5: kontaktskjema for Motordriver-kortet



Figur 6: en av enkoderene festet til motoren



(a) strømkablene



(b) USB-kabel

Figur 7: Kablene brukt til strøm og seriell tilkobling

5.2 Systemidentifikasjon

For å kunne kontrollere motorene godt er det nødvendig å vite parametrene deres. For å finne disse ble motorligningene for en DC-motor analysert og det ble funnet fram til hva som trengs for å finne de forskjellige konstantene. Videre ble det gjennomført de nødvendige testene og systemet ble identifisert. Dette gjengis i samme rekkefølge i de etterfølgende delkapitlene.

5.2.1 Analyse av DC-motorligningene

Siden motoren har to innganger tilsier det at det er en børstet DC-motor. DC-motorligningene for denne blir da

$$\dot{x} = Ax + bu \quad (29)$$

hvor x , A , b og u er gitt av

$$x = \begin{bmatrix} i_a \\ \omega_m \end{bmatrix} \quad (30)$$

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{k}{L} \\ \frac{k}{J} & -\frac{B}{J} \end{bmatrix} \quad (31)$$

$$b = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} \quad (32)$$

$$u = V \quad (33)$$

Lasten på motoren er utelatt av denne modellen og er ansett som en forstyrrelse. Siden det er rotasjonshastigheten som skal styres er det nødvendig å se på transferfunksjonen fra fart til spenning. Transferfunksjonen er

$$\omega_m = \frac{k}{JL(s^2 + \frac{RJ+BL}{LJ}s + \frac{BR+k^2}{LJ})} \quad (34)$$

Det er tydelig at systemet er et to-pol system. For å definere systemet spesifikt må polene bli identifisert. Uten kunnskap om konstantene til systemet er dette umulig. Derfor er det nødvendig å se nærmere på måter å

finne disse verdiene. Det enkleste er å finne motstanden R . Ved hjelp av et multimeter kan vi måle motstanden på motoren og den ble funnet til å være 0.5Ω . Videre ser vi på stasjonærverdiene til systemet. De er

$$V = Ri_a + k\omega_m \quad (35)$$

$$B\omega_m = ki_a \quad (36)$$

Dette gir to ligninger og to ukjente (B og k) om vi vet V og i_a ved en stasjonær tilstand. R er, som tidligere nevnt, allerede kjent. B og k ble funnet ved å kjøre stegresponser på motorene. Om systemet er overdempet og dermed har to reelle og ikke-sammenfallende poler kan det bli nødvendig å ha kontroll på hvilken tidskonstant som er dominerende. Fra [9] oppgis det at de to tidskonstantene er

$$T_a = \frac{L}{R} \quad (37)$$

$$T_m = \frac{JR}{K^2} \quad (38)$$

$$(39)$$

hvor T_a kalles den elektriske konstanten og T_m kalles den mekaniske konstanten. For å finne den elektriske konstanten må farten holdes til 0. Dette gir en transferfunksjon fra spenning til strøm som er

$$i_a = \frac{1}{R(\frac{L}{R}s + 1)} \quad (40)$$

Herfra er det mulig å finne tidskonstanten T_a . Dette gir igjen muligheten til å finne den ukjente L . Er denne anderledes en den tidskonstanten som man finner for (34) vil det også være mulig å finne J . Dette vil da gi full oversikt over systemet og dermed vil systemidentifikasjonen være komplett.

5.2.2 Gjennomføring av sprangrespons-tester

Sprangrespons-testene er todelt. Siden man skal innsamle sprangresponser for både med rotasjon og uten rotasjon vil det være hensiktsmessig å gjøre dette som to separate tester. Derfor blir også gjennomføringen og forutsetningene tatt opp i to separate deler i dette delkapitlet.

5.2.3 Sprangrespons med rotasjonshastighet

Innhenting av data ble gjort ved hjelp av inneliggende programvare på motordriveren som kan måle strømmen til motorene og logge hastigheten til motorene igjennom eksterne enkodere. For å få et rikt datasett blir det gjennomført 9 sprangresponser med forskjellig sluttverdi på sprangene (sett i tabell 3). Tiden sprangene går er lagt til 1 sekund etter loggingen av data starter for å unngå å miste data og for å kunne se at motorene er i ro når spranget går. For å sikre at sprangresponsen er kommet til en stasjonær verdi holdes volten til sluttverdien i 19 sekunder slik at hele loggingen skjer over 20 sekunder. Samplingsraten ble satt til 5 samples per sekund noe som tilsvarer en sample per andre tidel. Dette var nok for å få god oversikt over alle sprangene. Etter et sprang er gjennomført lagres strøm- og hastighetsmålingene i separate *.mat* filer for å kunne bli tatt inn i Matlab for å finne tidskonstanter og forsterkninger. Spenningen holdes lav i 30 sekunder i mellom sprangene for at motorene skal nå null hastighet. Ved høyere spenningsverdier var det nødvendig å bremse hjulet for at hastigheten skulle rekke å bli 0 før neste sprang. På grunn av at funksjonen på kortet som kjører motorene ikke har mulighet til å sette inn verdier med desimaler ble det nødvendig å måle hvilken spenning som faktisk ble satt på de forskjellige motorene. Dette ble gjort ved hjelp av et multimeter. Målingene er i tabell 3. Sprangresponsene er gjengitt i figurene (8) og (9).

Fra figurene er det lett å se at systemet er overdempet. Dette gjør at det har to unike og reelle poler. Sprangresponsene blir analysert til å ha verdiene gjengitt i tabell 1 og 2. Det er da mulig å finne B og k ved å sette inn stasjonærverdiene for de forskjellige sprangene i ligningene (36) og ta gjennomsnittet av disse. Det gir verdiene

$$B_{m1} = 2.6453 \cdot 10^{-7} \quad \sigma_{B_{m1}} = 6.7 \cdot 10^{-15} \quad (41)$$

$$K_{m1} = 0.0405 \quad \sigma_{K_{m1}} = 1.29 \cdot 10^{-5} \quad (42)$$

$$B_{m2} = 5.9780 \cdot 10^{-7} \quad \sigma_{B_{m2}} = 5.4 \cdot 10^{-11} \quad (43)$$

$$K_{m2} = 0.0435 \quad \sigma_{K_{m2}} = 1.02 \cdot 10^{-4} \quad (44)$$

$$(45)$$

Tidskonstanten blir funnet ved bruk av *interp1* funksjonen i Matlab.

Testnummer	Motor 1		
	$\omega_m(max)[\frac{rad}{s}]$	$\omega_m(63.2\%)[\frac{rad}{s}]$	$i_{stasj.}[mA]$
1	18.14	11.46	0.10
2	28.21	17.83	0.25
3	43.77	27.66	0.31
4	59.07	37.33	0.55
5	69.69	44.04	0.37
6	85.98	53.71	0.59
7	100.69	63.64	0.43
8	110.52	69.85	0.67
9	126.20	79.76	0.56

Tabell 1: avleste verdier fra fart og strømmåling til motor 1

Det ble bare sett på området hvor kurven ikke hadde gjentatte verdier. Det ga tidskonstanten for motorene.

$$T_{m1} = 1.67 \quad \sigma_{T_{m1}} = 1.4 \cdot 10^{-3} \quad (46)$$

$$T_{m2} = 2.15 \quad \sigma_{T_{m2}} = 0.18 \quad (47)$$

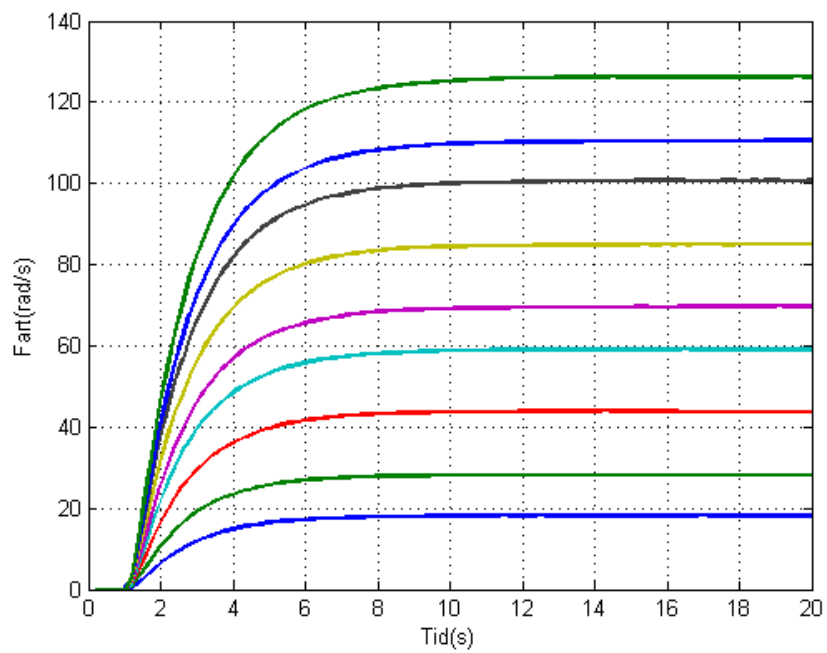
Siden det ikke er kjent om denne verdien er den mekaniske eller elektriske konstanten er det ikke mulig å finne J eller L ennå. Det må gjøres flere sprangresponser for å finne den elektriske konstanten.

Testnummer	Motor 2		$i_{stas.}$ [mA]
	$\omega_m(max)$ [$\frac{rad}{s}$]	$\omega_m(63.2\%)$ [$\frac{rad}{s}$]	
1	12.89	8.14	4.3
2	24.26	15.33	0.69
3	40.47	25.57	1.3
4	57.39	36.27	0.74
5	68.35	43.20	0.69
6	85.41	53.97	0.71
7	102.57	64.83	0.62
8	113.62	71.81	0.81
9	129.25	81.69	0.86

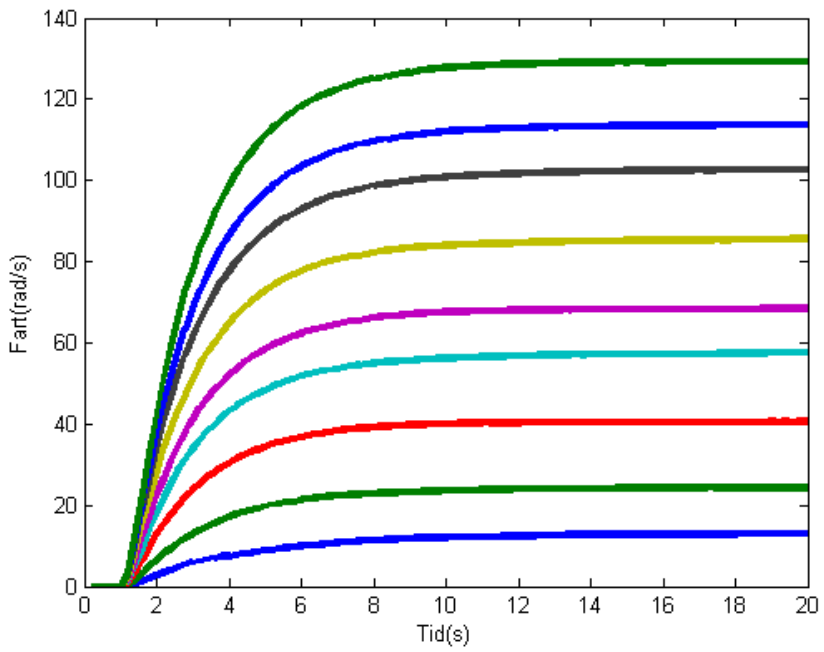
Tabell 2: avleste verdier fra fart og strømmåling til motor 2

Testnummer	Spenning Motor 1 [V]	Spenning Motor 2 [V]
1	0.87	0.87
2	1.25	1.23
3	1.82	1.78
4	2.36	2.36
5	2.73	2.70
6	3.26	3.26
7	3.84	3.84
8	4.19	4.19
9	4.70	4.69

Tabell 3: inputspenningene for sprangresponer med rotasjon



Figur 8: Fartskurven til motor 1



Figur 9: fartskurven til motor 2

5.2.4 Sprangrespons uten rotasjon

Innhenting av data skjer igjennom motordriveren. Samplingsraten er satt til 10 samples per sekund. Denne har økt i forhold til innhenting av hastigheten fordi det antas å være et raskere sprang og man trenger derfor høyere oppløsning. Det ble gjort 10 tester på hver av motorene med et steg til 2.06 Volt. Det er ikke behov for å se på sprangresponser med forskjellige verdier fordi det er tidskonstanten til systemet som skal finnes og ikke forsterkningen. For å unngå rotasjon ble hjulene på maskinen holdt fast. Test nummer 9 ble ikke vurdert i estimeringen av tidskonstanten fordi motorene hadde rotasjon på grunn av menneskelig svikt. Kurvene til testene kan ses i figur (10) og (11). I figurene ser man at det er en varians på sluttverdiene. Dette kommer av støy, men siden variansen er så lav som 2.342 μA for motor 1 og 6.286 μA for motor 2 så er dette ansett som neglisjerbart. Maksverdien blir funnet ved å ta middelverdien av de 30 siste verdiene i målingen. Dette, i tillegg til den utregnede tidskonstantsverdien, er gjengitt i tabellene 4 og 5. Tidskonstanten og konstanten L ble funnet til å være

$$Ta_{m1} = 0.46 \quad \sigma_{Ta_{m1}} = 6 \cdot 10^{-3} \quad (48)$$

$$L_{m1} = 0.23 \quad \sigma_{L_{m1}} = 1.5 \cdot 10^{-3} \quad (49)$$

$$Ta_{m2} = 0.52 \quad \sigma_{Ta_{m2}} = 11.5 \cdot 10^{-3} \quad (50)$$

$$L_{m2} = 0.26 \quad \sigma_{L_{m2}} = 2.9 \cdot 10^{-3} \quad (51)$$

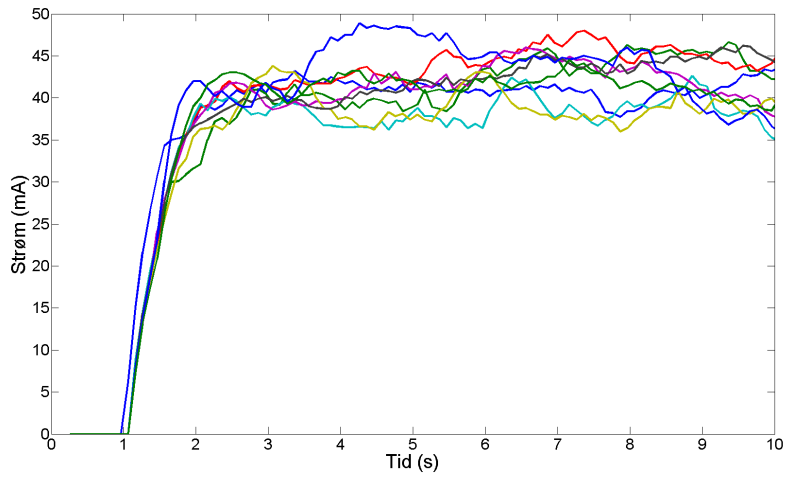
$$(52)$$

Tidskonstanten er anderledes enn den som ble funnet i forrige steg. Dette tilsier at forrige tidskonstant tilsvarer den mekaniske tidskonstanten. J blir da regnet til å være

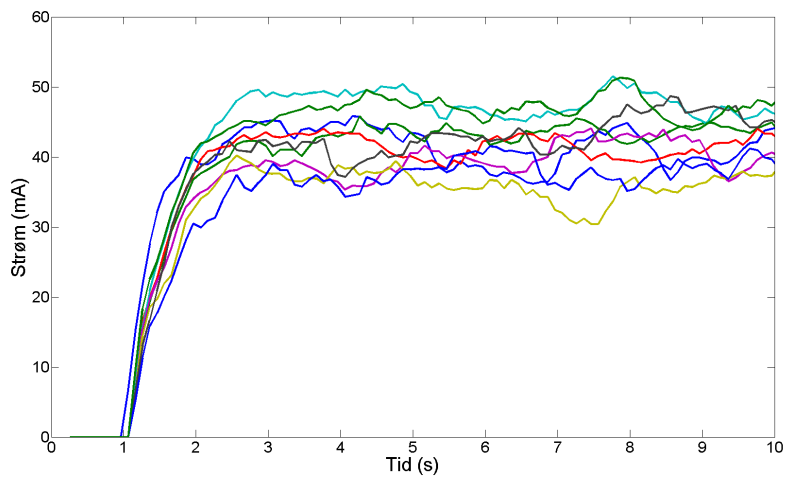
$$J_{m1} = 0.0062 \quad \sigma_{J_{m1}} = 1.3 \cdot 10^{-6} \quad (53)$$

$$J_{m2} = 0.0101 \quad \sigma_{J_{m2}} = 6.8 \cdot 10^{-5} \quad (54)$$

Dermed er altså alle konstantene identifisert.



Figur 10: strømkurven til motor 1 uten rotasjon



Figur 11: strømkurven til motor 2 uten rotasjon

Testnummer	Motor 1	
	Strøm _{max} [mA]	Strøm _{0.632} [mA]
1	40.9	25.8
2	44.5	28.1
3	45.5	28.7
4	39.0	24.6
5	42.7	27.0
6	38.5	24.4
7	44.3	28.0
8	42.0	26.6
10	41.3	26.1

Tabell 4: avleste strømverdier til motor 1 uten rotasjon

Testnummer	Motor 2	
	Strøm _{max} [mA]	Strøm _{0.632} [mA]
1	40.5	25.6
2	43.6	27.6
3	41.5	26.2
4	47.1	29.8
5	40.9	25.9
6	35.2	22.2
7	44.7	28.2
8	38.3	24.2
10	47.2	29.8

Tabell 5: avleste strømverdier til motor 2 uten rotasjon

5.2.5 Diskusjon

Det er tydelig at luftfriksjonen for motoren B er så nær null at den kan neglisjeres fra det endelige systemet. Dette gir transferligningen (55).

$$\omega_m = \frac{k}{(JLs^2 + RJs + k^2)} \quad (55)$$

Eigenfrekvensen til systemet er da

$$\omega_0 = \frac{k}{\sqrt{JL}} \quad (56)$$

Dette gir en dempningsgrad som er lik

$$\zeta = \frac{R\sqrt{JL}}{kL} \quad (57)$$

Tallverdiene for de to motorene er da

Motor	ζ	ω_0
1	2.0	1.067
2	2.26	0.85

Tabell 6: Dempningsfaktor og egenfrekvens

At dempingen var stor var tydelig i fartsgrafene 8 og 9, så dette tilsier at systemet tilsvarer de faktiske motorene. Variansen til de utregnede konstantene er også veldig lav, noe som videre tilsier at de er reelle. Systemet blir brukt videre for å bestemme reguleringskonstantene for å kontrollere farten til motorene og derav kontrollere farten til ballen. Størst usikkerhet ligger hos målingene av strømverdiene uten hastighet på hjulene. Dette er fordi det er en del støy på målingene av strømmen og at det kan forekomme små bevegelser i motorene, som ville ha påvirket målingene.

5.3 Regulering av motorene

I dette kapitlet dokumenteres gjennomføringen av reguleringen av de to motorene. I tillegg blir gjennomføringen og resultatet diskutert.

5.3.1 Reguleringen

For å kunne bestemme en godkjent regulator må visse krav til regulatoren settes. Motorene skal sette hastigheten til ballen ved å ha samme hastighet som den ønskede hastigheten. Motoren skal ikke ha overstyring med mer enn $0.1 \frac{m}{s}$ ved en gitt hastighet. På maksimalverdien ($27 \frac{m}{s}$) tilsier det 0.37%.

Ballens påvirkning på motoren er en økt last. Dette vil si at hastigheten vil synke når ballen kommer i kontakt med motorene. Med tanke på at ballen er i utskytningsmekanismens handlingsområde i veldig kort tid vil regulatoren måtte ha muligheten til å oppnå hastigheten den hadde før ballen påvirket motorene på veldig kort tid. På grunn av metningen til systemet som følge av batterikapasiteten blir det satt som øvre grense at stigningshastigheten er mindre enn ett sekund. Oppsummert er kravene:

1. Unngå overstyring med mer enn 0.37%
2. ha en stigningstid lavere enn 1 sekund.

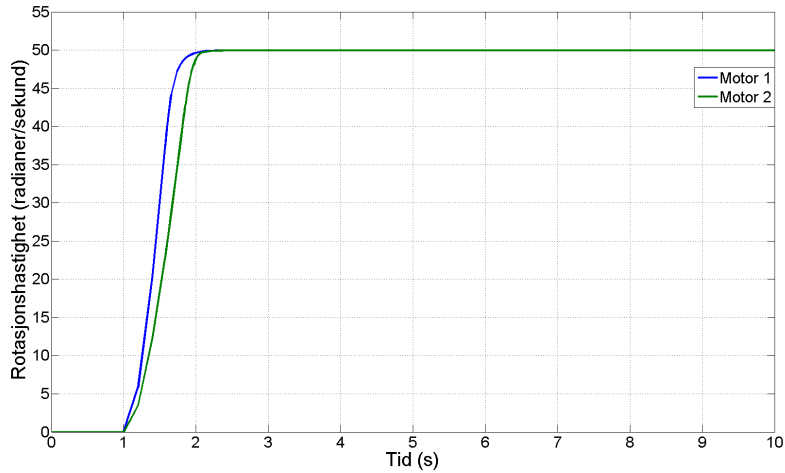
For å finne de ønskede konstantene lages en Simulink-modell av en PID-regulator koblet med en ideell motor med de verdiene som ble funnet fra systemidentifikasjonen. Dette brukes istedenfor PIDtuner-funksjonen til MatLab for å legge inn metning på grunn av batteriets kapasitet. Dette ga konstantene

$$K_p = 20 \tag{58}$$

$$K_i = 0.1 \tag{59}$$

$$K_d = 2.5 \tag{60}$$

Konstantene er like for begge motorene. Dette gir litt forskjellig stigningskurve men skal tilfredstille de kravene som ble satt for regulatoren, som man ser i figur (12). For å se om motoren responderer likt som i



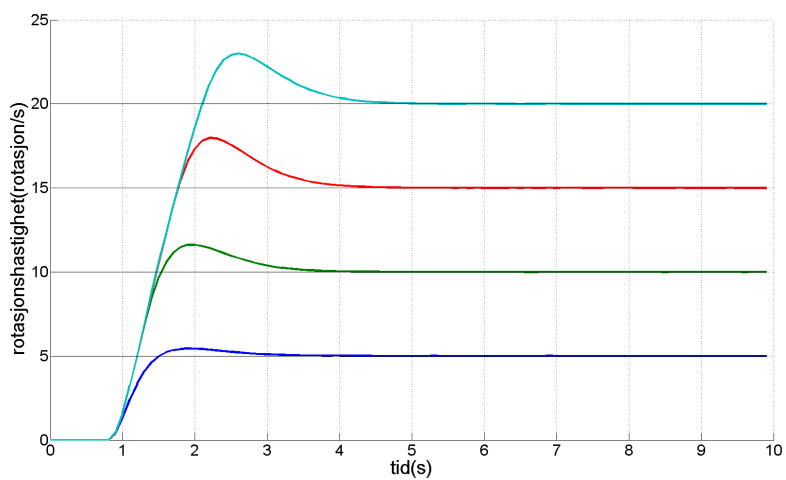
Figur 12: Stegrespons til begge motorene hentet fra SimuLink

Simulink-modellen blir det kjørt stegresponser til 5,10,15 og 20 rotasjoner per sekund på maskinen. Dette ble også plottet og kan ses i figur (13) og (14).

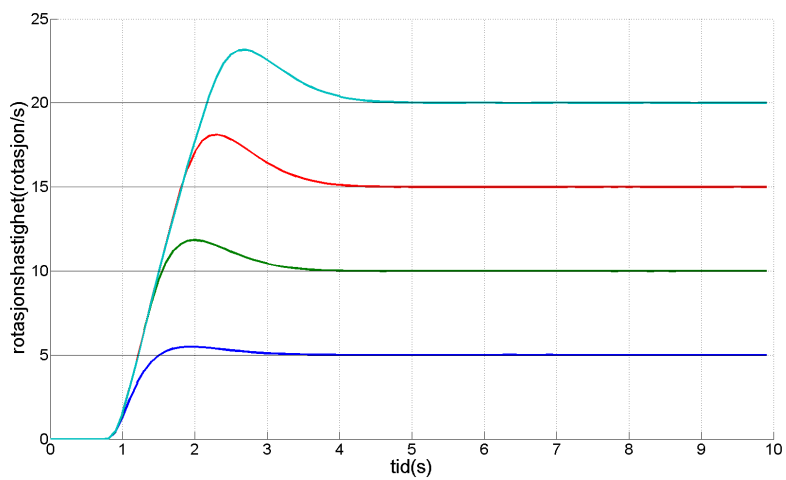
Regulatorkonstantene ville, med tanke på den modellen som ble funnet i forrige del, gi en veldig annen respons. På grunn av mangel på tid ble det valgt å bruke denne reguleringen uansett.

Det ble gjort en test der ballen ble ført inn i systemet og det ble sett på hvordan ballen påvirket hastigheten til motorene. Farten sank med forholdsvis mye da ballen gikk inn i utskytningsmodulen, som er mulig å se i figur (15). Den sank såpass mye at det ble behov for å gjennomføre en foroverkobling på maskinen. For å finne ut hvor mye som trengs for at motorene skal ha den hastigheten som ballen skal ha når den kommer ut av maskinen ble det produsert en kalibreringsalgoritme. Denne algoritmen gikk som følger

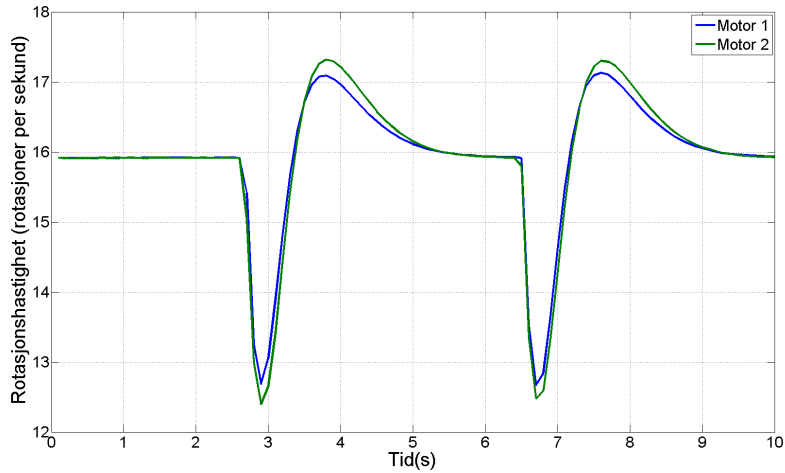
1. Maskinen setter motorene til den hastigheten brukeren ønsker og starter logging av hastigheten etter 5 sekunder
2. Brukeren gjennomfører et skudd



Figur 13: stegresponsene til Motor 1



Figur 14: stegresponsene til Motor 2



Figur 15: To skudd

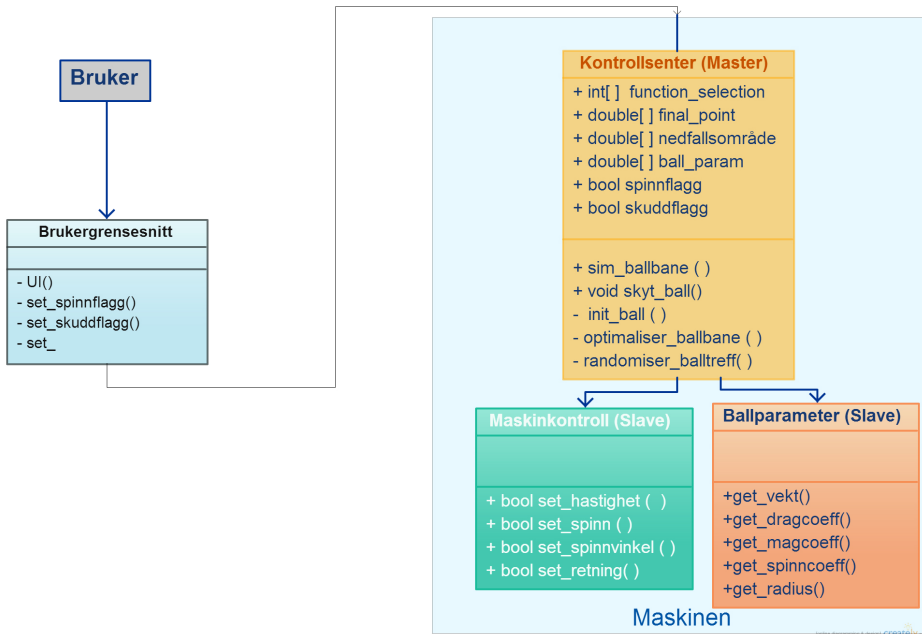
3. Den laveste hastigheten motorene har blir da hentet ut og ballens ønskede hastighet blir delt på minimalhastigheten
4. Dette forholdstallet blir videre ganget med den hastigheten maskinen setter som referansepunkt til maskinen.

Dette ville trolig gi den hastigheten som var ønsket og hastighetsproblematikken var derfor løst. Motorene ble også behandlet hver for seg og ganges ikke med det samme tallet siden de har forskjellige systemkonstanter og derfor vil motorene reagere ulikt på ballen. Dette kan utjevne eventuelle fysiske feilkilder på hvordan ballen går ut. Det vil si at om ballen har en høyere friksjonskonstant i forhold til den ene motoren enn til den andre vil dette skape en lavere hastighet på det ene hjulet. Ved å kompensere for hver motor separat kan dette neglisjeres, noe som er positivt for farten til ballen og retningen til ballen.

Dette blir den reguleringen som vil bli brukt når testene gjennomføres.

5.3.2 **Diskusjon**

Reguleringen har en del mangler som kommer av manglende informasjon om reguleringskretsen på kortet. Det kan virke som det er en lavere metning enn det som er brukt for å simulere maskinen siden det tar lengre tid å oppnå en høyere verdi. Det kan også virke som om kortets oppdaterings-hastighet er forholdsvis lav siden det er mye overstyring i stegresponsen. Regulatoren bør tunes bedre og det bør ses etter en bedre reguleringskrets.



Figur 16: Klassediagram som utgangspunkt for programvarestruktur

5.4 Programvaren

Programvaren blir selvprodusert foruten noen få aspekter. Kodingen av selve motordriverkortet er allerede gitt på kortet. Det ble også funnet eksempler på produsentens hjemmeside, skrevet i Python, for å kunne kommunisere med kortet. Alt utenom dette er tenkt å bli programmert i konstruksjonen av systemet.

For å ha et startpunkt for programvaren blir det først gått utifra det klassediagrammet som ble presentert i 16.

Dette er en programvarestruktur som må sees nærmere på for at det skal tilsvare det prototypen skal baseres på. Klassediagrammet fra prosjektoppgaven ble basert på den funksjonspesifikasjonen som ble satt i den oppgaven. Siden funksjonspesifikasjonen har blitt endret må endringer i klassediagrammet også vurderes. Det av funksjonspesifikasjonen som programvaren skal gjøre er

- Kunne skyte en fotball som oppfyller kravene til FIFA til et ønsket

punkt. Et godkjent skudd er at ballen havner innenfor en sirkel med radius på en meter hvor sentrum er det ønskede punktet.

- Kunne sette hastigheten, som ballen skal forlate maskinen med, til alle verdier i området $0.0 < v \leq 27.0$ avrundet til ett desimaltall.
- Ha en kalibreringsrutine for å optimalisere maskinen for en gitt type ball og lagre den til senere bruk.

For dokumentasjon og utprøving av maskinen var det også ønskelig å ha muligheten til å loggføre motorens fart og strøm. Dette har derfor blitt fokusert på i programvaren.

Kalibreringen i henhold til ball blir ansett som todelt. I optimaliseringen er det noen konstanter som er ukjente. Disse varierer i henhold til hvilken ball som brukes. Dette gjør at maskinen må ha en måte å finne ut hva disse er. I tillegg vil baller av forskjellig tyngde, størrelse og elastisitet oppføre seg forskjellig i maskinen. Dette gjør at det er behov for å se på kalibrering av selve utskytningsmodulen. Den kalibreringsrutinen som ble valgt i 5.3 ble brukt.

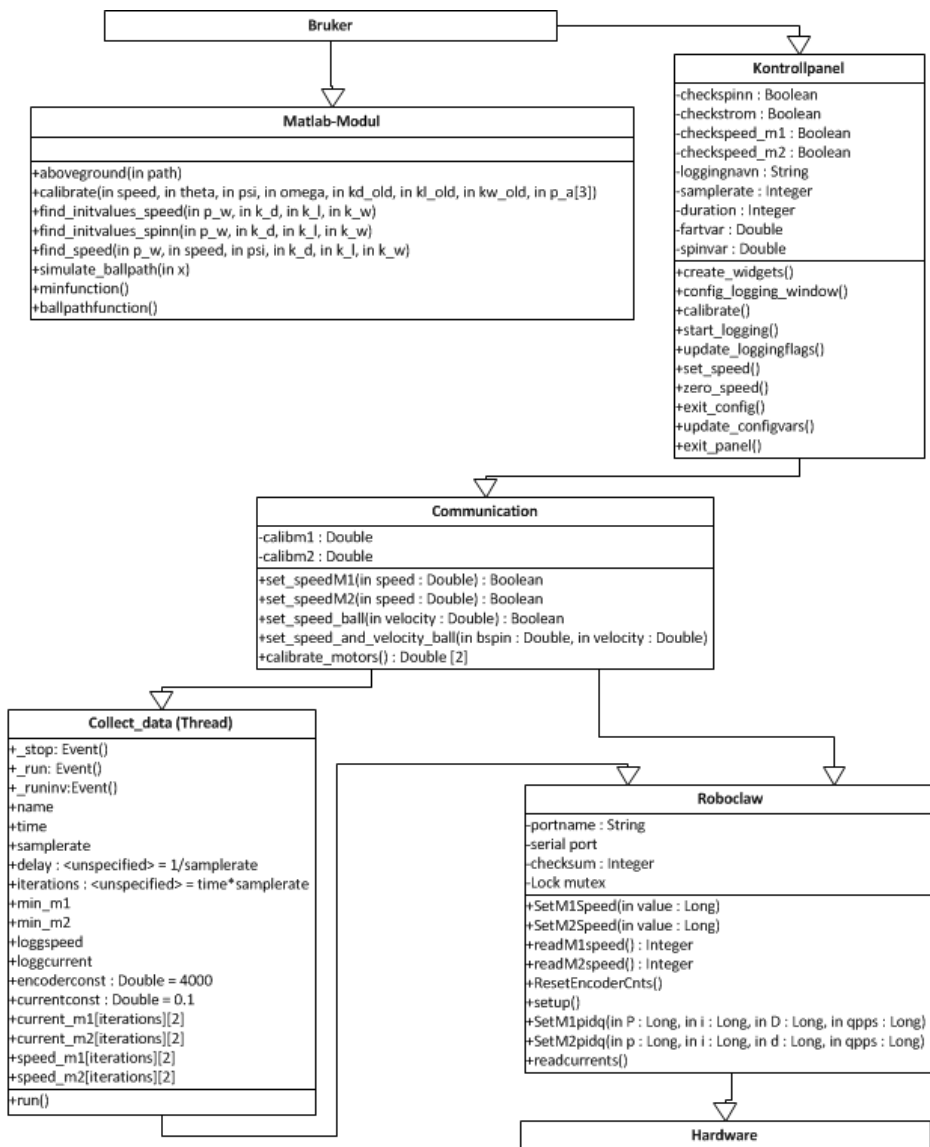
For å spare tid blir ikke optimaliseringsfunksjonene i Matlab overført til selve programmet. Dette gjør at maskinen har en todelt programmeringsstruktur. Dette gjør igjen at brukeren blir den som kombinerer disse to modulene. Klassediagrammet er gjengitt i figur (17).

For å få mer oversikt over programstrukturen blir det sett nærmere på de mest signifikante funksjoner og aspekter ved de forskjellige modulene.

5.4.1 Matlab-modul

Denne modulen tar seg av optimaliseringen med tanke på ballens flygebane. Den bruker den koden beskrevet i 2.2. Funksjonene *find_speed*, *find_initvalues*, og *calibrate* er basert på optimaliseringsfunksjonen, bare med forskjellige begrensninger. Det blir gått igjennom de forskjellige funksjonene en og en.

- *find_initvalues_speed* - Finner rett fart og utgangsvinkel i z-retning for å kunne treffe et oppgitt punkt. Dette er funksjonen som kan anses som hovedfunksjonen til MatLab og skal kunne bestemme de rette initialbetingelsene. I første omgang ble det lagt inn en mulighet for å skru av og på spinn, men siden maksverdiene for fart og spinn



Figur 17: klassediagrammet for programvaren til prototypen

var anderledes enn for en ball med ren hastighet ble dette fjernet og en funksjon som finner initialbetingelser ved hjelp av spinn ble laget. *find_initvalues_speed* gir bare muligheten til å variere i z og y retning. I denne oppgaven har det bare blitt sett på variasjon i planet, altså ikke i høyde. Dette gjør at om man skal ha kontroll over variasjonen i x-retning må det introduseres spinn.

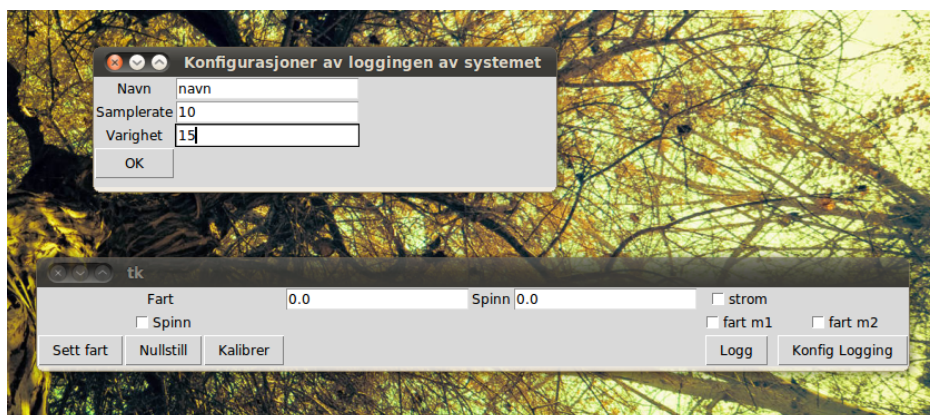
- *find_initvalues_spin* - Funksjonen gjør det samme som *find_initvalues_speed* bare med spinn også. Funksjonen kan finne posisjonen i tre dimensjoner ved hjelp av spinn. Denne funksjonen har fortsatt θ fast siden dette krever at maksinen endrer vinkel i forhold til en gitt y-retning. Siden det ikke er påtenkt å gi brukeren muligheten til å manipulere θ blir noen områder av x,y -planet fysisk umulig å lande på, på grunn av ballens fysiske ligninger. Dette gjør det mulig å beregne baner som oppfyller halvparten av bør kravet om å kunne sette spinn på ballen. Den andre halvdelene kommer i Kommunikasjonsmodulen.
- *find_speed* - Når man får et avvik i landingsposisjon fra det man ønsker er det stor sannsynlighet for at hastigheten innsatt er feil. Dette gjør at man må finne ut hva hastigheten faktisk er utifra hvor ballen lander. Dette blir gjort ved å holde fast vinkelen, siden denne er kjent, og bare variere hastigheten for å finne optimal vinkel. Her blir det antatt at konstantene i ligningsettet er reelle. For å ha minst mulig påvirkning av konstantene bør skuddene være korte. Dette er fordi tyngdekraften er generelt en større påvirkningskraft enn konstantene når det kommer til flygeavstand, så et kort skudd vil gjøre eventuelle avvik i konstantene neglisjerbare. Dette avhenger av at det ikke er noen spinn.
- *calibrate* - finner konstantene til systemet basert på landingen til et skudd ved å holde hastighet og vinkel fast og variere på konstantene. Denne funksjonen er en del av løsningen for kalibreringsbehovet. Funksjonen er også grunnen til å innføre konstantene i ligningene som optimaliseringsvariabler. Avviket kommer da i form av å gå for langt eller for kort i forhold til ønsket punkt. Det er ikke sikkert om denne funksjonen er nødvendig for å få et presist skudd. Det vil nok bli ansett som en form for fininnstilling. Dette er fordi med de verdiene som man fant i kildene for balltype [10] så vil ballen ha liten

variasjon. Utifra den koeffisienten som ble funnet i [8] vil ikke variasjonen være stor og koeffisienten beholdes dermed forholdsvis liten. Forstatt er ikke denne kjent og funksjonen holder begrensningene mellom 0 og 10 for alle de tre konstantene.

- *minfunction* - Dette er funksjonen *fmincon* bruker for å optimalisere funksjonen. Funksjonen tar inn en simulert bane med de gitte initielle verdiene og finner det punktet som har kortest avstand fra det ønskede punktet og returnerer avstanden til *fmincon*. Avstanden i *y*-retning er vektet 10 ganger mer enn de andre retningen. Dette er fordi dette er den retningen som får de største utslagene siden maskinen er rettet mot denne retningen.
- *Simulate_ballpath* - Abstraherer simuleringsfunksjonen *ode45*, som bruker *ballpathfunction*, til å finne ballens bane igjennom luften. Funksjonen endrer også formen til optimaliseringsvariablene slik at de passer med simuleringsfunksjonen.

5.4.2 GUI-modul

GUI-modulen er grensesnittet mellom bruker og selve ballmaskinen. Denne tar inn brukerens input og sender det videre til kontroll-modulen. Her har man et enkelt oppsett som gir muligheten til å styre ballens hastighet og om ballen skal ha spinn eller ikke. Det er også mulig å sette hvilken spinn ballen skal ha i tillegg til å nullstille hastigheten og kalibrere for balltypen som blir brukt av maskinen. Man kan også stille inn lengden på loggetiden og samplingraten igjennom en underliggende meny, i tillegg til å starte målingene. Det er mulig å velge hvilke motorer man vil logge fart fra. På grunn av strømmålingens underliggende funksjon er det bare mulig å skru av og på strømmålingene for begge motorene samtidig. Det ble ikke vurdert som nødvendig å gjøre det mulig å skru målingene av og på separat siden det ville i såfall være å ikke lagre en verdi. Det å kunne skru av målinger ville bare være nødvendig om man ønsker høyere samplingrate, da kallene tar ukjent antall sekunder og kan skape forsinkelser. Dermed er det ingen vits i å bare logge en av strømverdiene. Modulen er programmert med Python-pakken TKinter som er cross platform basert. Dette gjør at GUIet er forholdsvis likt på både Windows og Linux. Et bilde av systemet kan ses i figur (18)



Figur 18: Brukergrensesnittet som sett i Linux (Ubuntudistribusjon)

5.4.3 Kontroll-modul

Her ligger hoveddelen til programmet. Dette er laget som tolker det brukeren vil ha igjennom brukergrensesnittet og videreformidler det motordriverkortet trenger å vite igjennom Roboclaw-klassen. De eneste variablene maskinen har er å kalibrere motorene i forhold til balltype. Disse finner man ved å gjennomføre en kalibreringsrutine. Farten blir beregnet i *set_speed_ball* og loggføring av hastighetsverdier og lignende blir gjort ved hjelp av thread-klassen *Collect_data*. Denne tråden blir gjort mer rede for senere. De essensielle funksjonene blir nå forklart i en punktvis liste.

- *set_speed_ball* endrer farten fra meter per sekund til pulser per sekund. Dette er fordi regulatoren i motordriverkortet bruker denne benevnningen til å beregne hastigheten. Hypotesen for hastigheten er at hastigheten til ballen er lik hastigheten til hjulene når de kjører med samme hastighet. Har hjulene forskjellig hastighet oppstår det en spinn på ballen der forskjellen mellom de to motorenes hastighet er det som gir spinn. For å finne hastigheten til ballen i meter per sekund bruker man ligningen

$$v = r\omega$$

For å finne hastigheten til et punkt som går rundt en fast akse med avstand r med en gitt rotasjonshastighet, som blir det samme som

ballens hastighet. Siden det er rotasjonshastigheten funksjonen vil finne, deles hastigheten til ballen med radiusen til hjulet. Dette gir hastigheten oppgitt i radianer per sekund. For å få det i rotasjoner per sekund må man dele dette med 2π . Siden kortet teller pulser per sekund må funksjonen ganges med enkoderkonstanten som er 4000. Dette gir

$$\omega = \frac{v4000}{r2\pi}$$

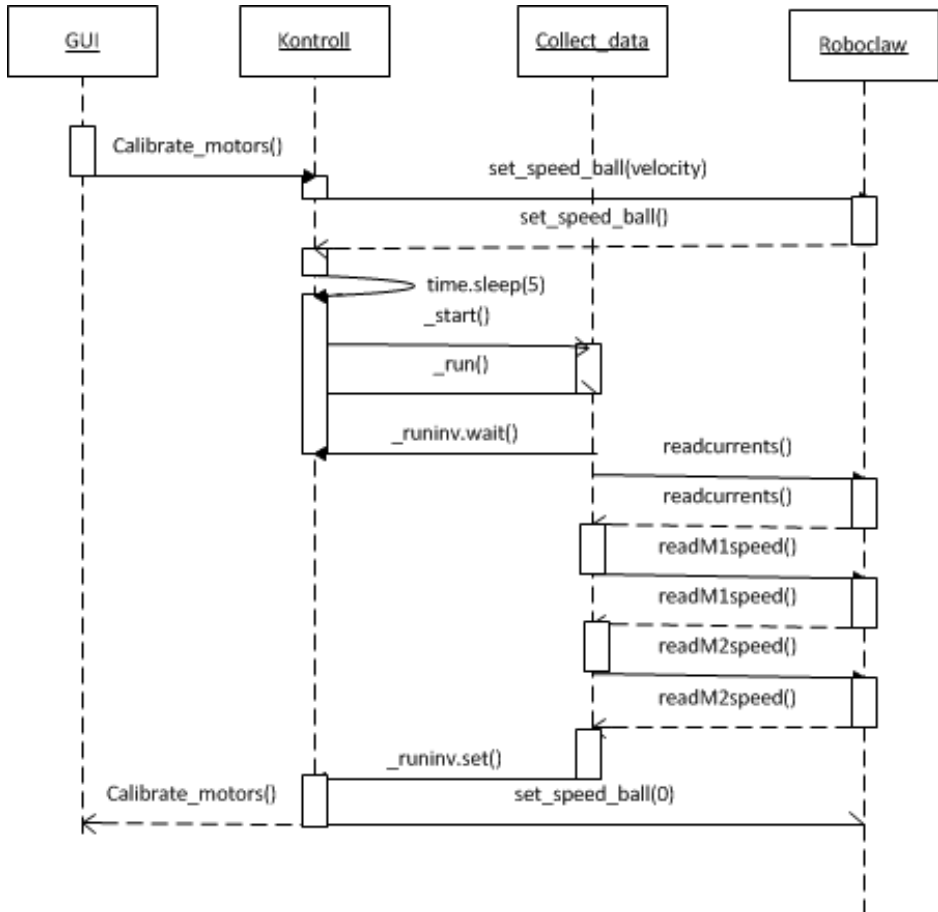
Enkoderkonstanten vil forandre seg om det er andre enkodere som blir brukt. Den samme funksjonen brukes i spinnfunksjonen. Det brukes ikke på den ønskede spinnen da denne blir oppgitt i radianer per sekund. Der ganges tallet bare med $\frac{4000}{2\pi}$.

- *set_speed_spin_ball*- bruker samme prinsippet som *set_speed_ball* for å sette farten. Det eneste forskjellige er at man tar inn både fart og spinn. Den høyre motoren blir satt til farten i tillegg til halvparten av den gitte spinnen. Den venstre motoren får trukket fra halvparten av den gitte spinnen. Dette tilsier at forskjellen mellom de to er lik den gitte spinnen man ønsker ballen skal ha mens farten er summen delt på to. Dette vil gi ballen den hastigheten og spinnen man ønsker. Det er mulig å sette inn negativ spinn med denne løsningen og.
- *calibrate_motors* Dette gjennomfører den kalibreringsrutinen som blir beskrevet i 5.3. Kalibreringskonstantene er de eneste variablene denne klassen har. Det er lagt opp kode for å kunne definere forskjellige kalibreringskonstanter for forskjellige baller. Denne koden var ikke ferdigskrevet ved den testingen som ble gjennomført. Det vil dog ikke være store forskjeller mellom å kalibrere direkte før ballen skytes ut siden konstantene vil være like, bare fra en annen kilde.

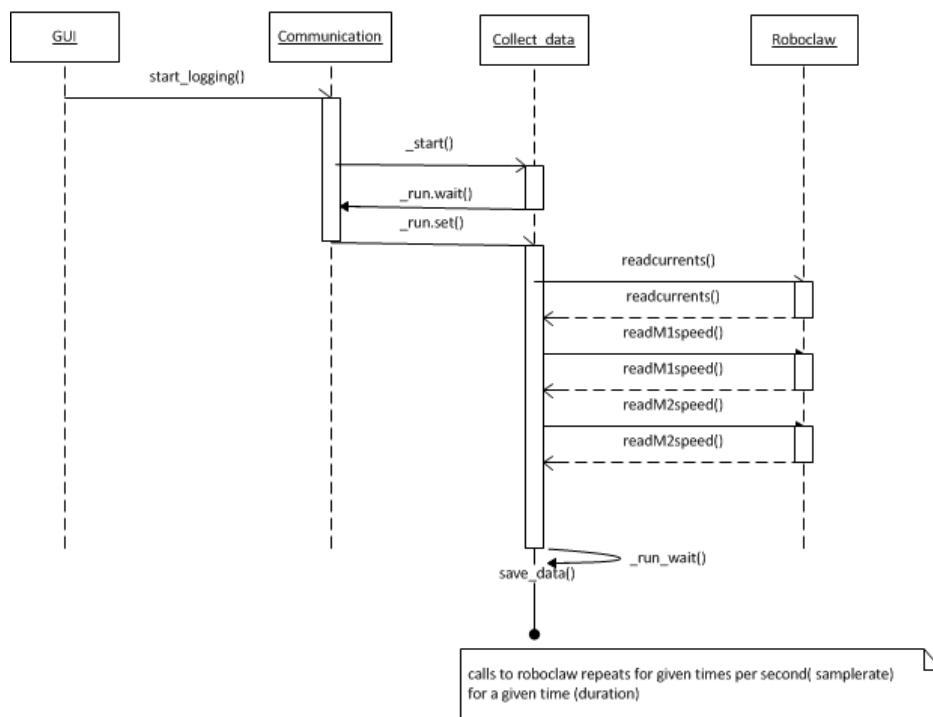
5.4.4 Collect_Data-tråd

Dette er en trådmodul som har i oppgave å logge informasjon. Det er ikke behov for å logge informasjon hele tiden derfor er det nødvendig å kontrollere når denne går. Dette gjør det nødvendig med synkronisering med

Kontroll-modulen og gjennomføres ved hjelp av Event-klassen til Python-modulen Threading. Tråden brukes til to forskjellige typer funksjoner. Den ene er standard logging som skjer synkront med andre handlinger som å sette hastighet. Den andre er kalibreringsrutinen hvor maskinen må vente på at en logging av et skudd blir gjennomført. For å kunne synkronisere med kontroll-modulen brukes `_runinv` som oppfører seg motsatt av `_run`. Kalibreringsfunksjonen kaller dermed opp en wait funksjon i Event-klassen. Denne gjør at funksjonen venter til `_runinv` har blitt satt igjen. Kalibreringsfunksjonen klarer `_runinv` før den returnerer funksjonen slik at Wait() funksjonen igjen vil sperre kalibreringen om den blir kalt igjen. Sekvensdiagram av begge disse funksjonene kan ses i figurene (20) og (19)



Figur 19: sekvensdiagram av kalibreringsrutinen



Figur 20: sekvensdiagram av logging av data fra maskinen

5.4.5 Roboclaw-modul

- **Roboclaw-modul** Denne står for kommunikasjonen med motordriverkortet. Oppgaven til modulen er å kommunisere med kortet og har derfor en seriellport. Dette er, rent softwaremessig, selve kortet og er hvor PID-konstantene er satt inn i tillegg til at det bestemmer maksverdien for batteriet. Dette setter arbeidsområdene for maskinen og er nødvendig for funksjonaliteten. Det er også lagt til en mutex-lås slik at man unngår at flere programmer aksesserer porten samtidig og dermed skaper problemer for hverandre. Funksjonene er, som nevnt tidligere, hentet fra Orion Robotics sine hjemmesider som er produsent av motordriver-kortet. Motordriver-kortet er ikke avhengig av at PC-programmet kjører for å kunne kjøre. Det er mulig å sette maskinen til en gitt hastighet, skru av programmet og koble fra pcen. Dette gjør det sikrere for maskinen at den ikke får problemer skapt av programmet. Maskinen går utifra hvilke meldinger den får og ingenting mer. I programvaren på kortet er det implementert en checksum for å unngå feile meldinger til kortet.

6 Testing av prototypen

For å vite om maskinen kan fylle de kravene som stilles i funksjonspesifikasjonen er det nødvendig å teste maskinen i bruk. Dette gjør at det først må identifiseres krav som funksjonspesifikasjonen faktisk stiller. Deretter må det gjøres rede for hvilke tester som må gjøres og deretter den faktiske gjennomføringen. Til slutt må testene analyseres og eventuelle feil eller problemer må gjøres rede for. Det er i denne rekkefølgen dette kapittelet er bygget opp.

6.1 Testkrav

Fra funksjonspesifikasjonen skal maskinen kunne skyte en ball til et gitt punkt. Det er ikke noen feilmargin oppgitt, men det blir gått utifra at om man er mindre enn en halvmeter fra i direkte luftlinje er det et godkjent skudd. Det vil bare bli sett på avvik i x - og y - retning da det blir målt når ballen lander. Dette betyr at det oppgitte punktet som maskinen sikter på har alltid 0 som z -koordinat. Skuddene blir gjort med fire forskjellige baller der to er fra samme produsent. De har forskjellig tekstur på overflaten og forskjellig sammensetning av overflate. Det blir gjort tester på lengdene 5, 10, 15 og 20 meter.

Å kunne sette hastigheten til å ha toppen et avvik på $0.1 \frac{m}{s}$ er det andre kravet som må testes. Dette innebærer å finne ut hva hastigheten til ballen er når den forlater maskinen. Dette blir gjort ved å måle punktet hvor ballen treffer bakken og deretter finne den reelle hastigheten med kjent utgangsvinkel. Dette blir gjennomført både på ukalibrerte målinger og kalibrerte målinger for å teste om kalibreringsmetoden også fungerer.

Siden utgangsvinkelen til ballen er satt manuelt er dette en måling som blir gjennomført utenom de testene som blir gjort på skudd. Det er satt opp en gradebenevning ved festemekanismen til maskinen. Denne har god nok oppløsning for funksjonspesifikasjonen. Det som må gjøres er å kalibrere viseren slik at null grader på gradebenevningen faktisk er null grader.

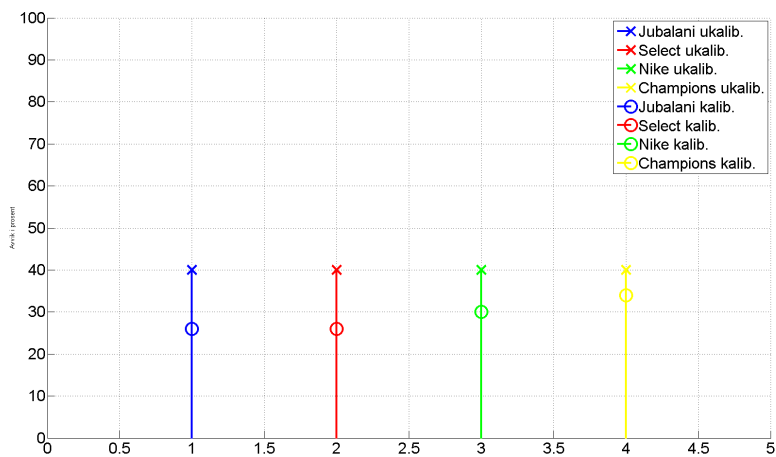
Kalibreringsrutinen blir testet ved å skyte ballene uten og med kalibreringsrutinen. Om ballene kommer innenfor feilmarginen før er ikke dette kravet nødvendig å fylle. Om ballene bare kommer innenfor feilmarginen etter kalibreringen blir gjennomført er kalibrering nødvendig og godkjent.

Siden kalibreringsrutinen er todelt må maskinen testes uten noen form for logging, bare med motorkalibreringen, og med kalibrering med hensyn til konstantene. Dette blir rekkefølgen fordi det antas at det er større avvik ved hastigheten ballen får utav maskinen enn det er med de konstantene som blir brukt i optimaliseringen av initialbetingelsene.

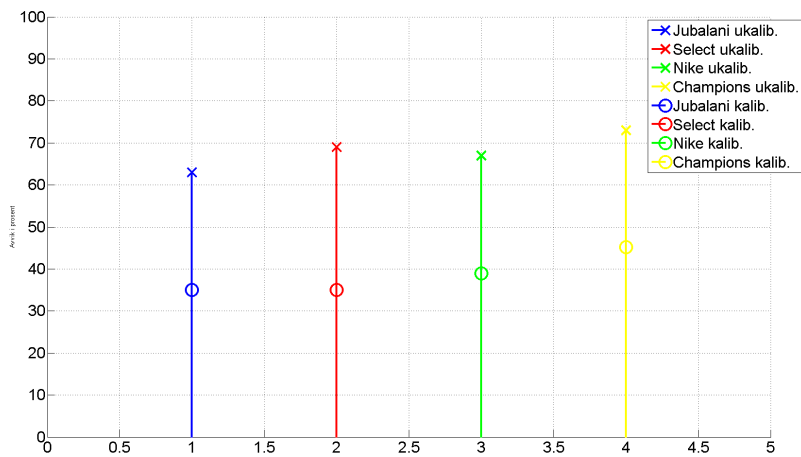
6.2 Test 1

6.2.1 Gjennomføring

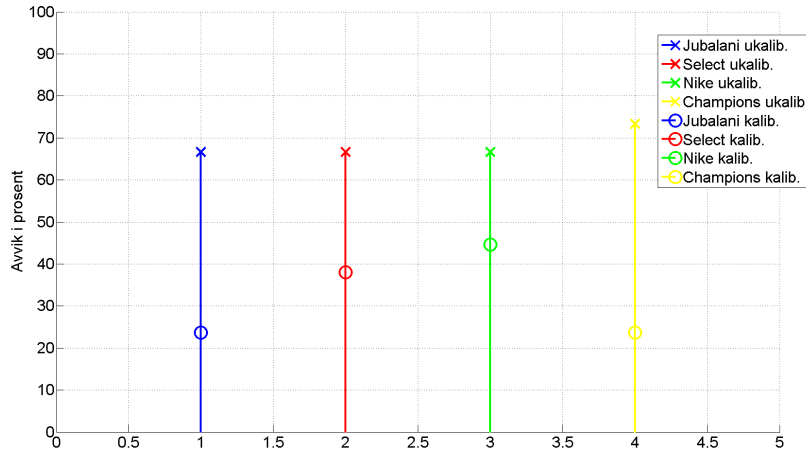
Testen ble gjennomført i gymsalen på Dronning Mauds Minne Høgskole. Den ble gjennomført med fire forskjellige baller fra tre forskjellige produsenter (Adidas(2x), Nike, og Select). De to Adidas ballene blir referert til Jubalani og Champions. Dette for å skille mellom de to. Denne testen var for å teste om den gitte kalibreringsrutinen virket og at maskinen kunne treffe det ønskede punktet. Det ble ryddet et område på 20×5 meter og markert opp 5, 10 og 15 meter med kjebler. Avstandene ble målt i forhold til disse kjeblene og bare i y -retning som maskinen skulle skyte i. Dette ble gjort fordi avvikene i x -retning var små i forhold til lengden til ballskuddet. Det er også vanskelig å måle dette korrekt med det utstyret som ble brukt da testen ble gjennomført. Det ble først gjort skudd uten å kalibrere maskinen for systemet. Deretter ble motorene kalibrert for ballen kalt 'Jubalani'. Testene ble da gjennomført igjen med de samme verdiene som den ukalibrerte versjonen. Resultatene av den testen kan man se i A.1.



Figur 21: 5-meter testen.



Figur 22: 10-meter testen.



Figur 23: 15-meter testen.

6.2.2 Analyse

Testen viser at det er behov for kalibrering av maskinen. Den viser også at kalibreringen ikke er god nok og at maskinen ikke gir rett hastighet til ballen. Ved å bruke *find_speed* funksjonen for å finne hvor mye av hastigheten som forsvinner på et skudd blir det identifisert hvor mye fart ballen egentlig hadde. Hva som forårsaker hastighetstapet er usikkert, men det er tydelig at ballen ikke oppnår den hastigheten hjulene har når ballen forlater utskytningsmodulen. Det er tydelig at kalibreringen gir en bedre effekt på hvor den faktisk lander. Om man ser på avvikene på 10 og 15 meter i figurene (21),(22) og (23) hvor hastigheten er lik både kalibrert og ukalibrert er det en signifikant økning i forbedret plassering. Det er også lett å se at Jubalani-ballen er den med størst forbedring fra ukalibrert til kalibrert skudd. Dette gir bakgrunn for at det hjelper mer å kalibrere for hver enkelt ball. Dette har nok med størrelse på ballen, lufttrykket inni ballen, og hvilket materiale ballen er laget av. Ved forskjellig høyde på ballene så er det også mulig at hjulene ikke treffer ballen med midtpunktet i ballen. Dette kan medføre at maskinen ikke får maksimal fart på ballen. For å stille på hvor hjulene treffer ballen er det installert en skrue med fjær på

som hever og senker den rennen ballen kommer inn i utskytningsmodulen på. Det er ikke mulig å stille denne mellom hvert skudd da den er lokalisert under ballen. Det er også en langtrekkelig prosess basert på øyemål. Skruen ble ikke justert på som en del av kalibreringen.

6.2.3 Justeringer før neste test

Fordi Test 1 ikke ga gode nok resultater ble det bestemt å gjøre noen justeringer.

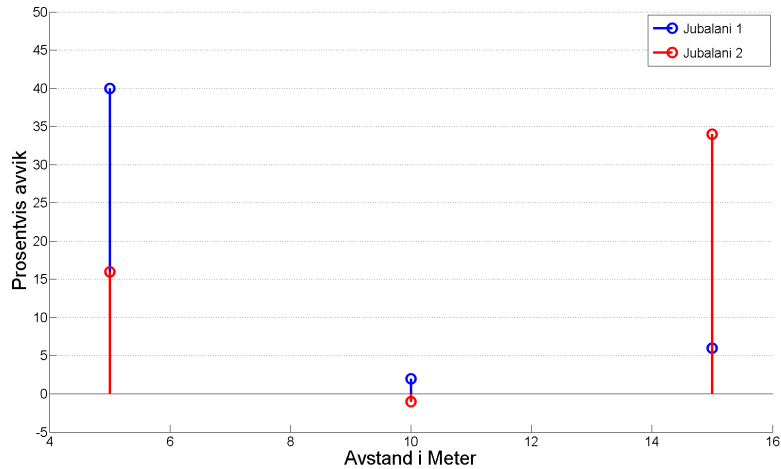
- Det blir brukt bare en type ball
- Det blir gjort testskudd på 10 meter per sekund og 45 grader og funnet den faktiske hastigheten.
- Hastigheten som ble funnet i forrige punkt blir kompensert for i *set_speed_ball*
- Kalibreringsrutinen blir igjen kjørt med ny *set_speed_ball* og for hver instilling med ønsket vinkel.

Siden avviket fra ønsket og faktisk landingspunkt var såpass mye forbedret ved å kalibrere maskinen antas det at det fortsatt mangler på kalibreringen. For å få en mer korrekt hastighet vil neste test følge den listen som blir nevnt over. Dette vil si at hastigheten til ballen blir tatt mer hensyn til som dermed, forhåpentligvis, vil gi bedre resultater. Valgt ball blir 'Jubalani'-ballen, da denne var den eneste tilgjengelige i flere eksemplarer. Det blir altså ikke sett på konstantene da det ikke virket som dette hadde det største utfallet på endelig verdi.

6.3 Test 2

6.3.1 Gjennomføring

Testen ble gjennomført i Glassgården i El-bygget på NTNU Gløshaugen. Det ble avsperrert et 20 meter langt område ved hjelp av bord for å unngå at ballene trillet for langt. Målingene ble gjort av en lasermåler som Institutt for Teknisk Kybernetikk disponerer. Der ballen landet ble markert med oransje briketter og deretter ble avstanden fra motoren målt i



Figur 24: Landingen til testene vist som prosent av ønsket landing

luftlinje. Det ble, igjennom tester funnet at farten var omtrent halvparten av forventet verdi. Hastigheten motorene ble satt til ble derfor doblet. Kalibreringsrutinen ga en minimal minking av hastighet på motorene.

Resultatet ble at 4 av 6 baller havnet innenfor den ønskede plasseringen. Mest imponerende er at ballene som skulle treffe et punkt 10 meter unna hadde 0.1 og 0.2 meter avvik fra det ønskede punktet.

6.3.2 Analyse

Resultatene i A.2 viser at landingene helst havnet for langt og ikke for kort. Dette tilsier at det å doble verdien kan være en overkompensasjon. I figur (24) er skuddene vist fram som prosentvis avvik fra ønsket verdi. Dette er for å vise hvor mye av ønsket verdi avviket er. På grunn av Funksjonspesifikasjonens krav til et godkjent skudd (innenfor en sirkel med radius 1 meter og sentrum i det ønskede punktet) blir det vanskeligere å få godkjente skudd ved større avstander. Allerede ved 10 meter er det ikke tillatt med et avvik på 10%. Utifra de verdiene som blir framstilt i figur (24) kan dette være vanskelig å oppnå da det er store avvik ved 2 av skuddene. Maskinen klarer å oppnå funksjonspesifikasjonens krav på noen avstander,

men bør jobbes mer med for å få det mer presis.

7 Diskusjon

Prototypen klarer det som er ønsket i Funksjonspesifikasjonen, men ikke veldig godt.

Utskytingsmodulen måte å gi ballen fart er ikke fullstendig utredet da den funksjonen som ble brukt ikke virket som den var presis nok. Det er tydelig at motordriverens regulering ikke klarer å holde motorene ved den ønskede hastigheten. Figur (15) viser store dropp i hastigheten til ballen. Figuren viser også at det er en sub-optimal regulering med for mye overstyring etter ønsket hastighet er nådd. Det at maskinen har forskjellig oppførsel ved forskjellige hastigheter er også et tankekors. Det kan virke som det er problemer med reguleringsystemet på kortet eller at modellen ikke er reell. Med tanke på at figurene (8) (9) viser en forholdsvis uniform stegrespons med forskjellig amplitude, og det er liten varians i de konstantene som ble funnet i systemet, tilsier det at PID-regulatoren på kortet er sub-optimal. Antageligvis er det for lav oppdateringsfrekvens på kortet. I databladet til motordriverkortet er det oppgitt en funksjon som gir en hastighetsmåling som tilsier hastigheten for hver 0.125 sekund. Dette gir en frekvens på 8 Hertz, som er veldig lavt, om dette tilsier det som er reelt på kortet vil dette være en veldig lav oppdateringsfrekvens, noe som kan skape problemer ved raske forandringer (som et ballskudd er) og ved høyere hastigheter.

Siden reguleringsmodulen på kortet ikke er optimal ble ikke reguleringen som henspeilet. Dette burde blitt tatt hånd om før systemet ble testet. Et bedre tunet system ville ihvertfall ha gitt mer konsistente hastigheter. I A.2 ser man at hastighetene ikke er like for de forskjellige skuddene. Dette kan komme av at dispensermotoren er raskere enn det motorene er til å få rett hastighet, noe som gjør at ballen dermed har feil hastighet. Det kan dermed være lurt å droppe derivasjonsleddet i regulatoren, siden denne fungerer som en form for demping, og ha et minimalt integrasjonsledd som fjerner standardavviket. Ved så å øke proposjonalkonstanten vil dette gi en forbedret respons fra systemet.

Utskytningsmodulens virkning på ballen er ikke helt presis og bør testes. Det bør bli gjort målinger av hvilken hastighet ballen faktisk har når den forlater maskinen. Dette kan da brukes i en kalibreringsrutine eller det går an å finne en enklere og/eller bedre mekanisme til å gi ballen fart.

Optimaliseringsrutinen fungerer som den skal. Utskytningsmodulen er

en større feilkilde og når den var omtrentlig korrekt ble målene nådd. Alle de tre avstandene som blir prøvd å treffe i Test 2 blir truffet av minst et skudd.

Maskinen bruker mindre batteri enn tidligere. Testene ble gjennomført over flere timer og det var liten til ingen indikasjoner på at batteriet ble kritisk utladet. Den originale ballmaskinen hadde en estimert arbeidsperiode på 15 minutter med to batterier av samme effekt. Dette kommer av at motorene ikke står på hele tiden, noe som bruker mye strøm. Dette er ikke et primært behov, men noe som er positivt for en maskin av en slik type.

8 Konklusjon

Det er mulig å lage en maskin som gir en gitt hastighet og vinkel til et ballskudd for å treffe et gitt punkt. Kalibreringrutinen hjalp til å få et bedre resultat, men var ikke nok for å kompensere for tap av hastighet til hjulene. Prototypens funksjonskrav ble oppfylt i løpet av byggeprosessen. Reguleringsinstansen til motordriverkortet var ikke god nok for denne typen applikasjon med tanke på metning og oppdateringsrekvens på systemet og skapte problemer for reguleringen. Funksjonspesifikasjonens krav var innenfor det som var mulig innenfor den tiden allokert til konstruksjon og nødvendig for å få en godkjent prototype. Systemet kan videreutvikles til å ha med flere funksjonaliteter og muligheter til å variere på andre initialbetingelser.

9 Fremtidig arbeid

For å videreutvikle denne prototypen bør det legges til rette for målinger av ballens faktiske hastighet og bedre utredning av hvordan en ball kommer opp i en gitt hastighet. Siden det ikke er nok å sette motorene til ønsket hastighet er det en usikkerhet å bruke den kalibreringsrutinen som brukes. En mulig løsning for å finne ballens hastighet er å sette opp to lysmottakere med en kjent avstand fra hverandre ved munningen av utskytningsmodulen. For at dette skal fungere må ballen dekke til hele virkningsområde til lysmottakerne for å skape en puls. Når det da skapes to pulser er det enkelt å regne ut hvor mye fart ballen har. Det er også mulig å analysere nærmere hvordan skuddet faktisk skjer ved å lage en modell for skuddet. Dette kan gi en bedre forståelse for hvordan ballen får farten sin og dermed gi mer kunnskap om hvordan farten kan manipuleres.

Å gjøre maskinen mer automatisk i settingen av initielle betingelser er også noe man kan videreutvikle. Det å sette retning til skuddet, i tillegg til hastigheten og spinn, gjør at maskinen har større muligheter til å gjennomføre flere varierte skudd enn det som eksisterer i dag. Med en helautomatisk maskin kan det være mulig å gi en variert keepertrening eller forskjellige ballavleveringer i en treningsøvelse. Det gir også mer presisjon og øker bruksområde til maskinen.

Å strømlinjeforme kontroll-modulen ved å slå den sammen med matlab-modulen til ett program er også ønskelig å gjennomføre. Det kan gi muligheter til å gjøre brukergrensesnittet enklere å forstå og bruke. Det er også mulig å kunne lagre ballers aerodynamiske konstanter for bruk hver gang og det kan dermed forbedres for bruk på flere områder i tillegg til kalibreringskonstanter. Ved å samkjøre disse to modulene gis en mer helehetlig prototype. Dette vil være ønskelig framover.

I forhold til optimaliseringen er det mulig å videreutvikle dens funksjon. Det er mulig å legge til flere punkter som ballen skal treffe slik at man får en unik kurve på ballen. Dette vil gjøre det mulig for maskinen å plassere et skudd rett over hodet på en mur eller legge en ball til et gitt punkt med en gitt retning.

Det er mulig å utvikle produktet 'En forbedret fotballskytmaskin' videre ved å bygge opp en maskin mer fra grunnen av med en dedikert utskytningsmodul som er skreddersydd for å variere spinnvinkel og mer kontroll på hvilken hastighet ballen har. Siden det er mye arbeid som må

gjøres før dette bør ikke dette være første fokus for videre arbeid.

Referanser

- [1] En forbedret fotballskytmaskin, December 2013.
- [2] Ken Bray and David Kerwin. Modelling the flight of a soccer ball in a direct free kick. *Journal of Sport Sciences*, 2011.
- [3] Ideal lift of a spinning ball. <http://www.grc.nasa.gov/WWW/k-12/airplane/beach.html>, July 2008.
- [4] Manon E. Grudgel. Modeling the viscous torque acting on a rotating object. *J. Chem. Physics*, 1998.
- [5] K.C. Wu U. Lei, C. Y. Yang. Viscous torque on a sphere under arbitrary rotation. *Applied Physics Letters*, 2006.
- [6] ode23. <http://www.mathworks.se/help/matlab/ref/ode23.html>, November 2013.
- [7] fmincon. <http://www.mathworks.se/help/optim/ug/fmincon.html>, November 2013.
- [8] Duarte Araújo Thomas Reilly, Jan Cabri. *Science and Football V*. Routledge, 2005.
- [9] Bjarne A. Foss Jens G. Balchen, Trond Andresen. *Reguleringsteknikk*. Insitutt fo teknisk kybernetikk, 2004.
- [10] Fédération Internationale de Football Association (FIFA). Rules of the game 2013/2014, 2013.

A Testresultater

A.1 Testskjema for Test 1

Navn	Type	fart	vinkel	Y	avvik i prosent%	Godkjent
Jubalani 1	5 m ukalib.	19.1	0	3	40	nei
Select	5 m ukalib.	19.1	0	3	40	nei
Nike	5 m ukalib.	19.1	0	3	40	nei
Champions	5 m ukalib.	19.1	0	3	40	nei
Jubalani 1	10 m ukalib.	10	41	3,7	63	nei
Select	10 m ukalib.	10	41	3,1	69	nei
Nike	10 m ukalib.	10	41	3,3	67	nei
Champions	10 m ukalib.	10	41	2,7	73	nei
Jubalani 1	15 m ukalib.	15	20.5	5	66.67	nei
Select	15 m ukalib.	15	20.5	5	66.67	nei
Nike	15 m ukalib.	15	20.5	5	66.67	nei
Champions	15 m ukalib.	15	20.5	4	73.33	nei
Jubalani 1	5 m kalib.	8.5	21.6	3,7	26	nei
Select	5 m kalib.	8.5	21.6	3,7	26	nei
Nike	5 m kalib.	8.5	21.6	3,5	30	nei
Champions	5 m kalib.	8.5	21.6	3,3	34	nei
Jubalani 1	10 m kalib.	10.2	40	6,50	35	nei
Select	10 m kalib.	10.2	40	6,50	35	nei
Nike	10 m kalib.	10.2	40	6,10	39	nei
Champions	10 m kalib.	10.2	40	5,48	45.2	nei
Jubalani 1	15 m kalib.	15	22.4	11,45	23.7	nei
Select	15 m kalib.	15	22.4	9,30	38	nei
Nike	15 m kalib.	15	22.4	8,30	44.7	nei
Champions	15 m kalib.	15	22.4	11,45	23.7	nei

Figur 25: Testskjema fra Test 1

A.2 Testskjema for Test 2

Ballnavn	Mål	Fart (m/s)	Vinkel(grader)	Y(meter)	Godkjent?	Prosent avvik (%)
Jubalani 1	5 meter	8.5	21	7.0	NEI	40%
Jubalani 2	5 meter	8.5	21	5.8	JA	16%
Jubalani 1	10 meter	10.6	32	10.2	JA	2%
Jubalani 2	10 meter	10.6	32	9.9	JA	1%
Jubalani 1	15 meter	17.2	15	15.9	JA	6%
Jubalani 2	15 meter	17,2	15	20.1	NEI	34%

Figur 26: Testskjema fra Test 2

B Tilstandstabell

Denne tabellen er referert til i delkapitlet 2.2 og er derfor tatt med som en appendiks. Tabellen ble produsert i [1]

Tilstand	med spinn		uten spinn	
	Optimalisere	Kontrollere	Optimalisere	Kontrollere
Pasning	posisjon, max(spinn)	λ, γ	posisjon	ω
Lav Pas- ning	posisjon, max(spinn)	$\psi_{max}, \lambda, \gamma$	posisjon	$\psi_{max} \omega$
Skudd	posisjon, max(spinn) og fart	λ, γ	posisjon, fart	ω
Direkteskudd	posisjon, max(spinn) og fart	$\psi_{max}, \lambda, \gamma$	posisjon, fart	$\psi_{max} \omega$

Tabell 7: De forskjellige tilstandene for maskinen og hva de må ta hensyn til