



DET TEKNISK-NATURVITENSKAPELIGE FAKULTET

BACHELOROPPGAVE

Studieprogram/spesialisering:	Vårsemesteret 2021
Bachelor i ingeniørfag / Automatisering og elektronikkdesign	Åpen
Forfatter: Magnus Dahle	
Fagansvarlig: Sven Ole Aase	
Veiledere: Sven Ole Aase og Ståle Freyer	
Tittel på bacheloroppgaven: Treghtsnavigasjon av skøyteløper på bane	
Engelsk tittel: Inertial navigation of a speedskater on track	
Studiepoeng: 20	
Emneord: Treghtsnavigasjon, posisjon, fart, IMU	Sidetall: 47 Stavanger 5. juni 2022

Innhold

Innhold	i
Forord	v
Sammendrag	v
1 Introduksjon	1
1.1 Problemstilling	1
1.2 Oppsett av oppgaven	2
2 Bakgrunn	3
2.1 Skøytebanen	3
2.2 Treghetsnavigasjon	4
2.3 Iphone 13 Pro	5
2.3.1 Akselerometer	5
2.3.2 Gyroskop	7
2.4 Globalt og lokalt aksesystem	8
2.5 Passeringspunktene fra tidtakersystem	9

INNHold

2.6	Euler vinkler	11
2.7	Kalmanfilter	12
2.7.1	Prediksjon	13
2.7.2	Oppdatering/korreksjon	13
3	Konstruksjon	14
3.1	Definisjon av aksene og loggingen	14
3.2	Test med kasse	16
3.3	Konstruksjon av posisjonsmålingen	17
3.3.1	Nedlegging av sløyfene	17
3.3.2	Logging av passeringene	18
3.4	Modellering av banen	19
3.4.1	Det globale og lokale aksesystemet	19
3.4.2	Tolkning av passeringene	19
3.5	Klokkesynkronisering	20
3.6	Sensorfusjon og posisjonsestimat	21
3.6.1	Prisippet for aksesystemet	21
3.6.2	Skjema over veien frem til posisjonsestimatet	22
3.6.3	Finne starttilstanden	22
3.7	Kalmanfilter design- håndteringen av data	27
3.7.1	Modellering av integrasjonen	27
3.7.2	Utledning av matrisene i filteret	28

INNHold

4	Resultat	32
4.1	Starttilstanden	32
4.2	Gyroskopdataene	34
4.3	Gyroskopkorrigeringen av starttilstanden	36
4.4	Implementering av data i kalmanfilteret	38
5	Diskusjon/konklusjon	41
5.1	Sensorfusjonering	41
5.1.1	Magnetometer	42
5.1.2	Ny type kalmanfilter for sensorfusjonering	42
5.2	Kvaternioner fremfor Eulervinkler	42
5.3	Posisjonssystemet	43
5.4	Kalmanfilteret	43
5.5	Wifi signal som støttestystem	44
5.6	En alternativ måte som ble prøvd	45
5.7	Alternativ sensor	46
5.8	Alternative måter å estimere fart og posisjon	46
5.9	Alternativt bruksområde	47
5.10	Miljøaspekt	47
	Bibliografi	49
	Vedlegg	49

Sammendrag

Oppgaven tar for seg fart- og posisjonsanalyse ved treghetsnavigasjon på en hurtigløpsbane for skøyter. Treghetssensorene som brukes er akselerometeret og gyroskopet. Dataene fra treghetssensorene hentes ut av en app på en Iphone 13Pro hvor også sensorene ligger internt. For å ha kontroll på hva som skjer med sensorene, legges Iphonen flatt på en kasse som plasseres på isen. En skøyteløper dytter kassen langs markeringslinjen i isen som følger indre løpsbane rundt banen. Et tidtakersystem registrerer passeringstiden på 5 ulike punkter langs linjen.

Fremgangsmåten baserer seg på å kalkulere en starttilstand med x- og y-aksen horisontalt for aksesystemet til akselerometeret. Starttilstanden korrigeres av gyroskopet når systemet blir satt i bevegelse slik at aksesystemet holdes likt. På denne måten kan gravitasjonen som påvirker systemet fjernes og det blir enklere å se hva systemet blir utsatt for i testen. Et kalmanfilter modellert på dobbelintegrasjon tar videre inn de korrigerede målingene samt passeringspunktene fra tidtakersystemet for å estimere farten og posisjonen.

Resultatet fra undersøkelsene gav store avvik fra virkelig fart og posisjon. Likevel kan det konkluderes med at dette er en fremgangsmåte som det kan bygges videre på. Utfordringene relaterer seg til kalibrering og skalering av sensorene samt en svakhet i få målinger av passeringspunkt.

Forord

Denne oppgaven markerer avslutningen på mitt bachelorstudie i Automatisering og elektrodesign ved Universitetet i Stavanger. Oppgaven tar utgangspunkt i farts- og posisjonsestimering på skøytebanen. Som skøyteløper selv har det vært meget interessant å se på mulighetene dette kan gi for meg. Et slikt ferdigutviklet system vil være nyttig når man skal forbedre seg som skøyteløper.

Jeg vil spesielt takke Håvard Myklebust, førsteamanuensis på Institutt for grunnskolelærerutdanning, idrett og spesialpedagogikk, som hadde ideen til oppgaven. Han bidro med gode innspill og forslag i oppgaven. Videre vil jeg takke for fine innspill fra Damiano Rotondo og Morten Tengesdal angående kalmanfilteret som brukes. Ståle Freyer og Sven Ole Aase sto for god veiledning gjennom oppgaven. Det var god motivasjon i å forsøke å overbevise Sven Ole om at intergrasjon av akselerasjon faktisk kan fungere.

Det har vært en lærerik og spennende prosess. Dette er en innledende undersøkelse for å se på mulighetene innen farts- og posisjoneringsestimering. Mye gjenstår, men det er ingen tvil om at det ligger muligheter her som kan bli et nyttig verktøy for idrettsutøvere.

Kapittel 1

Introduksjon

Hurtigløp på skøyter er en idrett som dessverre har hatt synkende interesse etter suksessen under Lillehammer OL i 1994. Liten interesse fører med seg lite mediadekning og sponsorflykt. Som et resultat av dårligere økonomi, må det gjøres harde prioriteringer med tanke på hva pengene skal bli brukt til. Forskning og teknologisk utvikling er kanskje det som blir tatt av budsjettet først. Idag finnes det ikke noe tilgjengelig system som viser sammenhengen mellom akselerasjon, fart og posisjon på isen. Et slikt system ville vært et nyttig verktøy med tanke på forbedring og perfektionering av skøyteteknikk. Formålet med oppgaven er derfor å undersøke om det er mulig å utvikle et system som kan bidra med verdifull informasjon i teknikkanalyse av skøyteløperen og således bidra til utvikling av skøytesporten.

1.1 Problemstilling

Opgaven undersøker hvordan treghetsnavigasjon kan fungere med en skøyteløper i en skøytehall. Etter som GPS signaler er tilnærmet helt fraværende for å få posisjonen i hallen, vil treghetssensorene stå for de eneste estimate-
ne. Sensorene festes til en plastkasse som dyttes rundt banen for å simulere skøyteløperen under kontrollerte forhold. Fordelen ved et fungerende system vil være å kunne fortelle skøyteløperen hvor farten økes og reduseres samt hvordan kurvene settes opp.

Problemstillingen er som følger:

Hvordan kan treghetssensorer bidra til å estimere fart og posisjon til en skøyteløper?

1.2 Oppsett av oppgaven

1.2 Oppsett av oppgaven

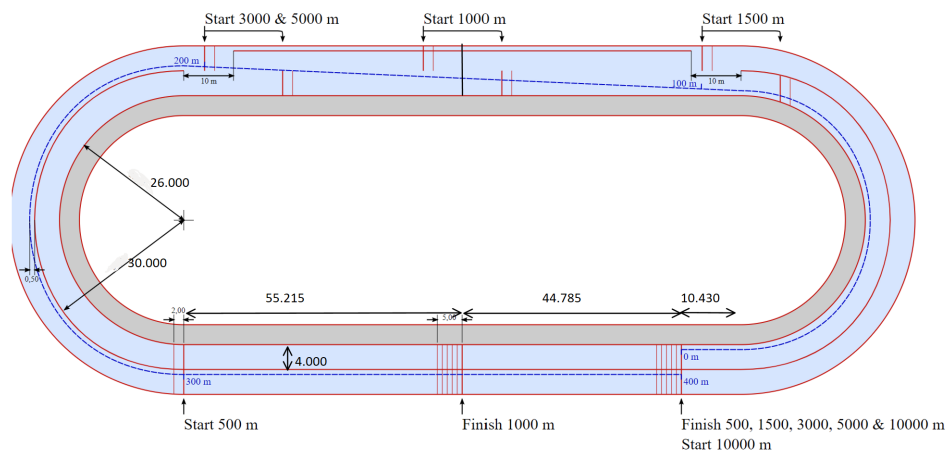
Oppgaven går videre over 4 kapitler. Kapittel 2 tar for seg bakgrunnen for oppgaven. Delkapitlene viser til relevant teori. I kapittel 3 blir fremgangsmåten for testen lagt frem. Her forklares hva som er gjort og hvorfor. Resultatene av fremgangsmåten blir presentert i kapittel 4. Kapittel 5 tar for seg drøfting av resultatene, tanker om videre arbeid samt alternative fremgangsmåter. En mappe med Pythonscript og tilhørende datasett brukt i oppgaven er vedlagt og kjøres enkelt etter nedlastingen. Skriptene har enkle forklaringer for å kunne leses dersom Pythonspråket er ukjent.

Kapittel 2

Bakgrunn

2.1 Skøytebanen

Skøytebanen er vist under i figur 2.1 og er basert på det offisielle målebrevet til Sørmarka Arena i Stavanger. Hvert år leies et akkreditert selskap inn for måle opp banen med laser. Dette kreves for å kunne arrangere større løp for toppidretten. I kapittel 3 figur 3.4 er banen modellert i Python for å vise hvilke linjer og markeringer som er viktigst for oppgaven.



Figur 2.1: Tegningen[4] er tilpasset målebrevet og viser alle oppmålinger og markeringer. Legg merke til at langsiden er tilnærmet 110 meter og at stiplet blå linje viser hva som er 400 meter

2.2 Treghetsnavigasjon

2.2 Treghetsnavigasjon

Treghetsnavigasjon er et navigasjonssystem som er uavhengig av andre ytre støttesystemer og signaler[12]. Systemet er svært viktig i blant annet fly, romskip og missiler for orientering, men systemet brukes også i forbrukerelektronikk. Et slikt treghetsnavigasjonssystem brukes sammen med GPS signaler i bilnavigasjonssystem. Dersom bilen kjører inn i en tunnel, vil GPS-signalet som forteller hvor bilen befinner seg mistes. Da må bilen kun støtte seg på treghetssystemet som kan estimere fart og posisjon mens den kjører gjennom tunnelen. Ved mottatt GPS signal på andre siden av tunnelen, fortsetter systemene å jobbe sammen om et best mulig estimat for posisjonen og farten til bilen.

Prinsippet bak treghetsteknologien baserer seg på 3 treghetssensorer,- akselerometer, gyroskop og magnetometer. Akselerometeret måler akselerasjon, altså endring i hastigheten. Gyroskopet måler vinkelhastighet og sier noe om hastigheten aksene roterer. Disse to sensorene kan brukes til navigasjon alene gitt at retningen det startes i kan defineres. Da dette kan være vanskelig, brukes det ofte et magnetometer som et digitalt kompass. Magnetometeret kan finne magnetisk nord og på den måten bidra med å si noe om hvordan systemet er orientert til enhver tid. Alle sensorene har tre akser for å kunne orientere seg i rommet. Disse tre sensorene vil komplementere hverandre. Der en sensor har en svakhet, har en annen sin styrke. Gyroskopet er svakt over lengre tid på grunn av drift, men reagerer raskt på kjappe bevegelser. Akselerometeret er mer stabilt over tid og drifter ikke, men reagerer ikke så raskt ved hurtige bevegelser. Magnetometeret bidrar med en referanseorientering til systemet sammen med akselerometeret, samtidig som det kan fortelle noe om rotasjonen sammen med gyroskopet.

Ved matematiske utregninger basert på rådata fra sensorene og integrasjon i en datamaskin, kan man estimere orientering, fart og posisjonen til objektet gitt en starttilstand. Systemet fungerer like godt på land som i luften og under vann på grunn av sin uavhengighet av ytre signaler. Derfor er systemet en viktig del av militære navigasjon- og våpensystemer da det er svært vanskelig å forstyrre systemet utenfra ved eksempelvis jamming.

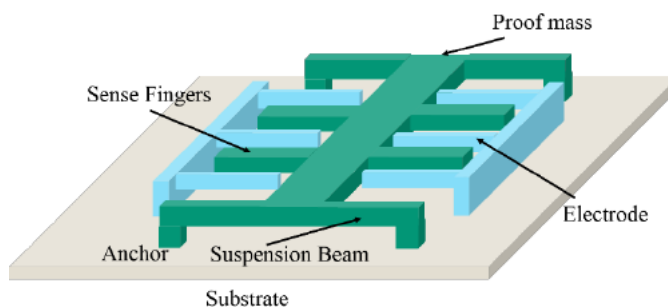
2.3 Iphone 13 Pro

2.3 Iphone 13 Pro

Oppgaven bruker enheten Iphone 13 Pro som datalogger ettersom den har tregthetsensorene inkludert. Offisielle datablad om hvilke typer sensorer som brukes og deres egenskaper har vist seg å være vanskelig å få tak i. Sensorene har alle tre akser: x, y og z og en maksimal samplingsfrekvens på 100Hz. Under følger en beskrivelse av de to sensorene i Iphone 13 Pro som inngår i denne studien.

2.3.1 Akselerometer

Et akselerometer måler endring i hastighet[23]. Akselerometer brukes i industrier verden over og de finnes i alle prisklasser. Dette gjør dem aktuelle for bruk i bl.a. forbrukerelektronikk, men brukes også til medisinske, militære og navigasjons formål. Sensoren måler krefter i g hvor 1 g er lik gravitasjonskonstanten i det området sensoren befinner seg. Gjennomsnittlig ligger konstanten på 9.81m/s^2 [23]. I sensoren er det den totale akselerasjonen for hver akse som blir målt. Det vil si at dersom den ligger i ro, vil den måle gravitasjonskraften som trekker på sensoraksene. Skulle sensoren ligge 100 prosent flatt, med x og y horisontalt, ville sensoren eksempelvis gi et utslag på g eller -g i z-aksen og null på x og y aksene. Dessverre er det sjeldent at sensoren ligger helt flatt og derfor vil gravitasjonen trekke på de andre aksene også. Ved bevegelse i en eller flere retninger, vil akselerometeret måle bevegelsen samt gravitasjonens påvirkning av aksene. Dette kan bli sett på som støy eller forstyrrelser i målingene og kan gi et uklart bilde av hva som skjer med sensoren.



Figur 2.2: MEMS akselerometer[19]. 'Sense Fingers' kan tolkes som massen som flytter seg til eller fra ene eller andre siden av 'Electrode'

Den vanligste typen akselerometer i telefoner og annen forbrukerelektronikk er MEMS-basert (Micro Electro-Mechanical System). Prinsippet bak virkemåten kan illustreres ved å se for seg en masse festet i en fjær. Ved akselerasjon vil

2.3 Iphone 13 Pro

fjæren presses bakover og man kan måle hvor kraftig den akselerer etter hvor mye fjæren er trykket bakover. Ved retardasjon, mer kjent som negativ akselerasjon, vil fjæren strekkes utover og man kan måle den motsatte virkningen på massen. Figur 2.2 viser virkemåten for en akse for den brikkebaserte teknologien som ofte brukes i telefoner. Dersom man skal ha 3 akser, må man ha tre slike montert vinkelrett på hverandre.

Kalibrering

Kalibrering av akselerometeret er en viktig del av håndteringen av sensoren[23]. Dette er for å sikre at dataene blir mest mulig nøyaktige og for å få reduserte feil i målingene. Ved montering av de tre aksene sammen for å gi x,y og z, vil man ha vanskelig for å montere de nøyaktig vinkelrett på hverandre. Samtidig vil konstruksjonen av hvert enkelt akse ha en usikkerhet. Temperatur og trykk er andre faktorer som er med på å påvirke målingene. Forholdene som akselerometeret ble konstruert, tilpasset og kalibrert i, vil ikke være like under bruk av sensoren. En forholdsvis enkel kalibrering av akselerometeret, men med gode resultater, kan gjøres ved hjelp av noen utregninger. Kalibreringen må korrigere hver enkelt registrerte måling. Først må man finne ut om målingen er korrekt. Det kan gjøres ved å bruke gravitasjonskonstanten som referanse da denne er kjent. Akselerometeret legges flat og hvis sensoren er ideell, skal man kunne måle -g eller g alt etter hvor den positive akse peker. Snus sensoren rundt, skal denne vise motsatt verdi. Dette vil dessverre sjeldent være tilfelle. Derfor må maksimal- og minimalverdien for hver akse måles der maksimalverdien er verdien rundt +1g og minimalverdien er verdien rundt -1g. Vet man disse målingene, kan offseten, a_o , for hver akse regnes ut ved hjelp av likning 2.2. a_{mo} er halve målingsområdet.

$$a_{mo} = \frac{a_{max} - a_{min}}{2} \quad (2.1)$$

$$a_o = a_{max} - a_{mo} \quad (2.2)$$

Videre kan man finne skaleringen. Denne verdien brukes for å definere målingsområdet til å være 2g, +1g til -1g. Det skal være like lange steg oppover som nedover på aksene. Skaleringen, a_s finner man med likning 2.3:

$$a_s = \frac{2}{a_{max} - a_{min}} \quad (2.3)$$

Neste steg er å kalibrere hver enkelt måling, a_m , fra hver akse med likning 2.4. a_k er den kalibrerte målingen for akselerasjonen.

$$a_k = (a_m - a_o)a_s \quad (2.4)$$

2.3 Iphone 13 Pro

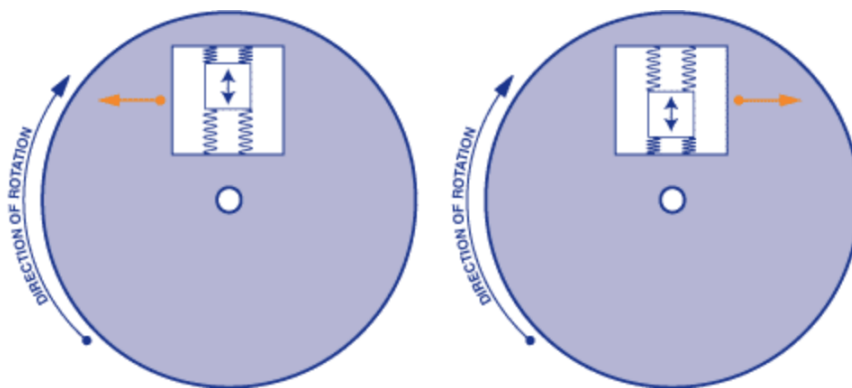
I tillegg brukes den enklere likingen for offset i 2.5 rett før testen starter i hver akse. Dette er bare snittet i hver akse og det forutsetter også at systemet er i ro. Kalibreringen av målingen ved en slik offset er vist i 2.6.

$$a_o = \bar{a} = \frac{1}{N} \sum_{i=0}^{N-1} a_i \quad (2.5)$$

$$a_k = a_m - a_o \quad (2.6)$$

2.3.2 Gyroskop

Gyroskopet måler vinkelhastigheten ω i radianer per sekund eller grader per sekund[23]. Et typisk gyroskopet måler vinkelhastigheten i tre akser,- x,y og z. Dette vil da si at ved en bevegelse som følger en av aksene helt rett, skal det ikke gi noe utslag i gyroskopet. Ved hjelp av denne informasjonen kan det finnes en orienteringen for objektet gitt en startorientering.



Figur 2.3: MEMS gyroskop[24]. Rotasjonen får den lille massen til å forflytte seg. Dersom det ikke er rotasjon på akse vil heller ikke massen flytte seg

Figur 2.3 illustrerer virkemåten til gyroskopet. Prinsippet bak gyroskopet er den lille massen som flytter seg ved rotasjon. Forflytningen av massen den ene eller den andre veien gir et estimat av vinkelhastigheten. Ved spinn på objektet, kan gyroskopet fort møte sine begrensninger da massen ikke kan flyttes mer i den gitte retningen. Gyroskopet egner seg best i navigasjonssystemer da det navigerende objektet bare roterer rundt aksene. Disse bevegelsene kan registreres og kompenseres for dersom det er nødvendig.

2.4 Globalt og lokalt aksesystem

Kalibrering

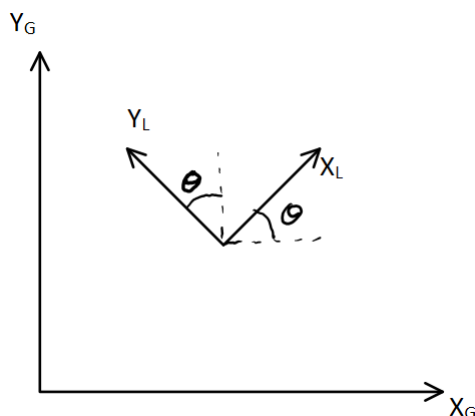
Gyroskopet trenger også å kalibreres. Oppgaven benytter en enkel kalibreringen av gyroskopet. Likevel er det en kraftfull kalibrering med gode resultater. Kalibreringen tar et gjennomsnitt av hver akse i et gitt intervall hvor gyroskopet ligger i ro. Ligning 2.8 viser kalibreringen ω_k for hver målte verdi ω_m ved offseten ω_o i likning 2.7.

$$\omega_o = \bar{\omega} = \frac{1}{N} \sum_{i=0}^{N-1} \omega_i \quad (2.7)$$

$$\omega_k = \omega_m - \omega_o \quad (2.8)$$

Offsetverdien i ligning 2.7 finnes ved å summere alle målte verdier i intervallet og dele på antall målinger.[16] Da vil man hindre en del drift i dataene ved videre prosessering. Akselereometeret og gyroskopet fungerer veldig bra sammen. Likevel vil man gjerne også ta med styrkene i et magnetometeret for å optimalisere systemet og ha en referanse til orienteringen.

2.4 Globalt og lokalt aksesystem



Figur 2.4: Aksesystemene

Det er viktig å kunne skille på hva som er det lokale og globale aksesystemet[23]. Figur 2.4 viser aksesystemene. Det globale aksesystemet er et fast system som det lokale systemet orienterer seg innenfor. X_L og Y_L er det bevegelige lokalet

2.5 Passeringspunktene fra tidtakersystem

systemet som følger objektet innenfor det faste aksesystemet notert X_G og Y_G i figuren. θ viser vinkelen som det lokale systemet er rotert i forhold til det globale. Figur 2.4 illustrer bare prinsippet i to dimensjoner da det her i oppgaven bare er x og y posisjon som plottes på banen.

2.5 Passeringspunktene fra tidtakersystem

Passeringspunktene hentes ut ved hjelp av et Mylaps-system.[7] Systemet er et tidtakersystem som er svært utbredt i veldig mange idretter og konkurranseformer. Mylaps brukes i alt fra sykkel-, skøyte- og maratonløp til radiostyrte biler og formel 1. Mylaps fører en del forskjellige typer systemer alt etter bruksområdet. Deres ProChip-system[7] brukes i oppgaven her. Under følger en kort forklaring av systemet og hvordan det brukes i oppgaven.

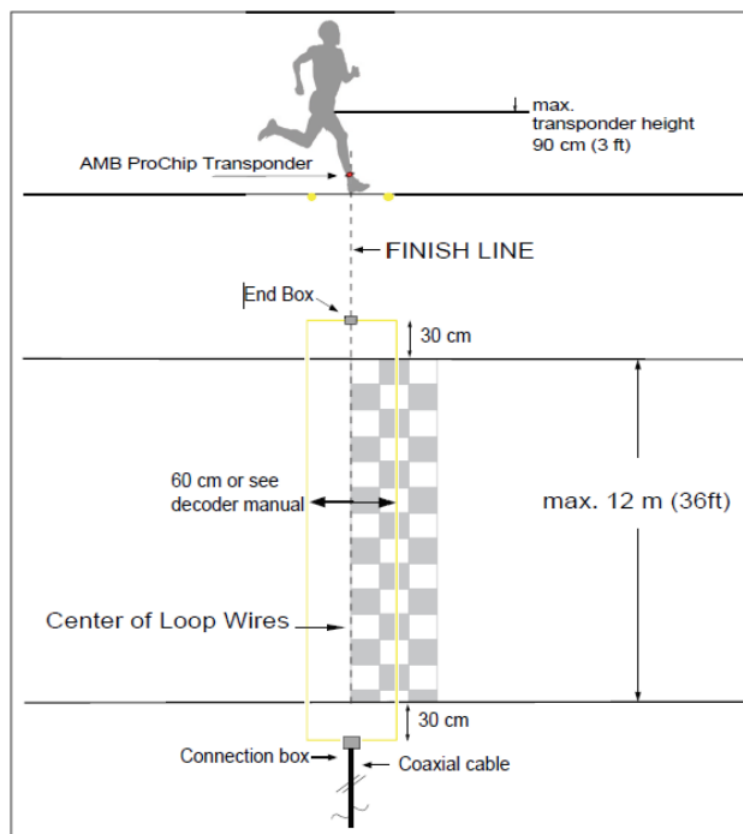


Figur 2.5: Figuren illustrerer prinsippet bak mylaps systemet sett fra skøyteperspektiv. Passeringen leses av ved hjelp av en sløyfe i isen tilkoblet en dekodeer og datamaskin[7].

Figur 2.5 viser hvordan systemet fungerer på en skøytebane. Skøyteløperen har en brikke på ankelene som en passeringssløyfe i isen registrerer. Sløyfen er tilkoblet en liten mottaker som sender passeringen videre til en dekodeer som lagrer og behandler passeringstiden før den vises i ønsket programvare. Plasseringen av sløyfen i isen er særdeles viktig med tanke på nøyaktigheten av passeringen. Figur 2.6 tar for seg detaljene om hvordan sløyfen må plasseres for å få et best mulig resultat.

Som vist i figur 2.6 kommer det her frem hvordan passeringssløyfen må legges

2.5 Passeringspunktene fra tidtakersystem

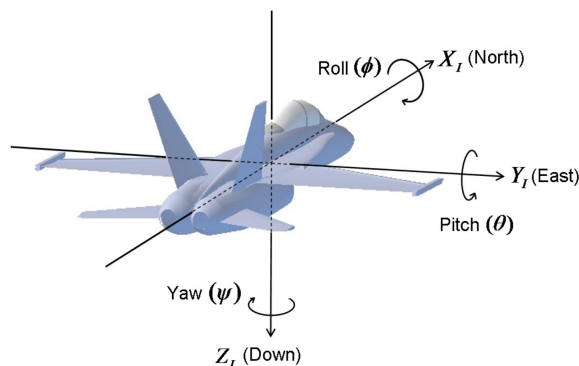


Figur 2.6: Det er svært viktig å legge midten av sløyfen midt på målstreken. [7]

og hvor brikken må plasseres på skøyteløperen for å få riktige passeringer. Brikken registreres litt før den passerer sløyfen, hele tiden frem til selve passeringen og litt etter at den har passert. En algoritme regner ut når brikken passerte midt mellom det gule sløyferektangelet i figuren. Plasseringene av disse sløyfene, kan si noe om distansen skøyteløperen har beveget seg. Det vil derfor være en stor fordel å ha flest mulig passeringssløyfer rundt banen.

2.6 Euler vinkler

Eulervinklene ble introdusert av Leonhard Euler. Teoremet hans sier at alle rotasjoner kan beskrives med tre vinkler. Vinklene representerer en tredimensjonal rotasjon ved vinklene; roll, pitch og yaw(ϕ, θ, ψ)[16].



Figur 2.7: Eulervinklene illustrert fra fly perspektiv [16]

Figur 2.7 følger høyrehåndsregelen som er den vanligste måten å definere aksene og rotasjonene på[20]. For å definere retning på rotasjonen om aksene, ses det utover aksene fra origo. Det vil si at positiv yaw(ψ) er definert ved rotasjon med klokken om z-aksen. Positiv pitch(θ) er definert med klokken rundt y-aksen og roll(ϕ) med klokken rundt x-aksen[16]. Hodet er en fin analogi til vinklene. Hodet lent fra side til side symboliserer roll, nikke hodet opp og ned kan ses på som pitch og riste på hodet kan tolkes som yaw.

Disse vinklene kan brukes i rotasjonsmatriser for å rotere x,y og z systemet i en ønsket retning. De tredimensjonale rotasjonsmatrisene for hver akse er gitt i matrisene 2.9, 2.10 og 2.11 med theta(θ) som en generell vinkel som ikke er relatert til pitch[23][16].

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (2.9)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.10)$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

2.7 Kalmanfilter

Likning 2.9 kan rotere systemet om x-aksen med en gitt vinkel. Samme for likning 2.10 rundt y-aksen og 2.11 rundt z-aksen. For å rotere et aksesystemet i et akselerometer om x kreves en 3×1 matrise R_a med henholdsvis akselerasjonene a_x, a_y og a_z et gitt tidspunkt. Rotasjonen med vinkel θ utføres ved å multiplisere rotasjonsmatrisen R_x med R_a .

Det bør noteres at rekkefølgen vinklene (ϕ, θ, ψ) er presentert ikke er så viktig, men hvordan rotasjonene er ordnet er sentral. I oppgaven her brukes rotasjonsordningen YXZ hvor roll er rundt y-aksen, pitch om x-aksen og yaw om z-aksen. Under er rekkefølgen på matrisene i likning 2.12. Her er cos og sin forkortet til c og s.

$$R_{\phi, \theta, \psi} = \begin{bmatrix} c(\phi) & 0 & s(\phi) \\ 0 & 1 & 0 \\ s(\phi) & 0 & c(\phi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) \\ 0 & s(\theta) & c(\theta) \end{bmatrix} \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

2.7 Kalmanfilter

Rudolf E. Kalman utviklet og la frem kalmanfilteret i 1960[6]. Filteret er en algoritme som i dag er veldig utbredt i sammenslåingen av sensorer og data. Det har blitt utviklet og tilpasset mange systemer og flere industrier siden det ble fremlagt. Filteret har mange bruksområder og blir for eksempel brukt i navigasjonssammenheng for å kombinere forskjellige data for slik å få et best mulig estimat av posisjon og orientering. Andre bruksområder er prosessstyring av ventiler og regulering av tanker. Det finnes flere utgaver av kalmanfilteret slik som utvidet kalmanfilter(EKF). Her i oppgaven fokuseres det på et mer vanlig kalmanfilter som tilstandsestimator for en prosess påvirket av stokastiske prosessforstyrrelser og stokastisk støy. Støyen blir antatt å være normalfordelt hvit støy og prosessen lineær. Ønskede tilstandsvariabler estimeres på bakgrunn av innkommende målinger, variansen til støyen i målingene og en matematisk modell av prosessen. Filteret jobber med data over tid og predikerer et estimat for de ønskede tilstandene før det tar inn målinger. Målingene og predikert estimat vektet mot hverandre gitt hvor stor usikkerhet det er i de innkommende målingene. Er usikkerheten i målingene store vektlegges estimatene mer. Enkelt forklart deler man gjerne kalmanfilteret inn i to steg som kjører i en løkke; prediksjon og korreksjon/oppdatering. En enkel modell for filtert brukt i oppgaven følger[1]. Full utledning av filteret gjøres i delkapitel 3.7

2.7 Kalmanfilter

2.7.1 Prediksjon

Prediksjonsfasen har som oppgave å predikere eller estimere tilstandsvariablene og kovariansmatrisen for prosesstøyen, som er den tilhørende usikkerheten i modellen, basert på de forrige estimatene.

$$x_{kp} = Ax_{k-1} + Bu_k \quad (2.13)$$

$$P_{kp} = AP_{k-1}A^T + BQ_kB^T \quad (2.14)$$

Ligning 2.13 tar for seg x_{kp} som er estimatet av tilstanden ved tidsteg k . Tilstanden er en $n \times 1$ matrise av de n ønskede estimatene. Her er det posisjon i x og y samt fart i x og y . A er overgangsmatrisen for tilstanden, u_k er de innkommende akselerasjonsmålingene og B er matrisen som kobler målingene til A . A og B utgjør sammen en dobbelintegrering av akselerometerdataen. Videre predikerer likning 2.14 prosesstøyen som tar inn usikkerheten i prosessen og kovariansmatrisen Q for akselerometermålingene. A^T og B^T viser til de transponerte matrisene til A og B . Kolonnene blir rader og radene blir kolonner.

2.7.2 Oppdatering/korreksjon

Ettersom som tilstanden og usikkerheten er predikert i forrige fase, vil det nå være nødvendig å ta inn en ny måling. Det må bestemmes hvor tungt den nye målingen skal vektet når den skal oppdatere tilstanden[6]. Den innkommende målingen er her passeringsdataen.

$$K = \frac{P_{kp}H^T}{HP_{kp}H^T + R} \quad (2.15)$$

Kalman vektingen(K) indikerer hvilke data som skal vektlegges. Den verdien regnes ut ved likning 2.15 hver runde i løkken basert på usikkerheten til prosessen P_{kp} og kovariansen R i målingen. H er en matrise som forteller hvilke estimater i tilstanden x som er dekket målingsmatrisen Y . Oppdateringen i tilstandsvektoren og usikkerhetsmatrisen følger i likning 2.16 og 2.17.

$$x_k = x_{kp} + K(Y - Hx_{kp}) \quad (2.16)$$

$$P_k = (I - KH)P_{kp} \quad (2.17)$$

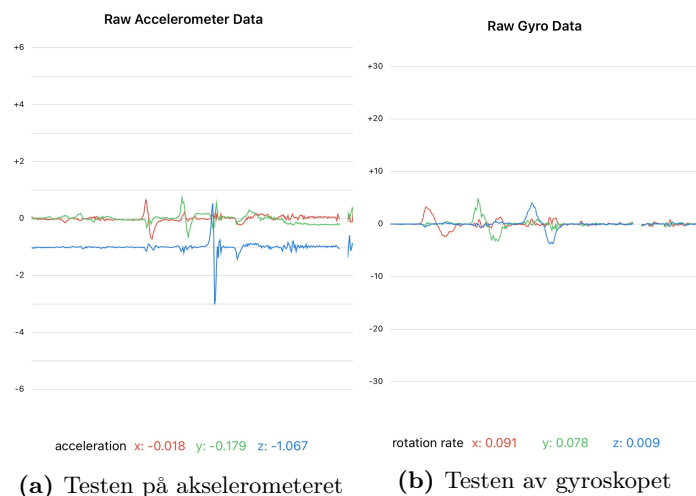
x_k oppdateres ved å ta stilling til den forrige predikerte tilstanden og de nye målingene Y . Her styrer K ved å si hva som skal vektlegges. K -verdiene i matrisen ligger mellom 0 og 1. Ved $K=0$ vil man bare gå videre med den predikerte tilstanden. Hvis $K=1$ vil all vekt legges på den innkommende målingen og den plusses i sin helhet på prediksjonen. P_k bruker en identitetsmatrise I i tillegg til den predikerte prosesstøymatrisen P_{kp} . Her hjelper også K med å fortelle hvor mye filteret skal stole på den nye usikkerheten.

Kapittel 3

Konstruksjon

3.1 Definisjon av aksene og loggingen

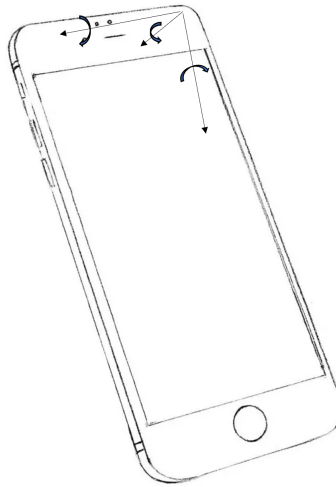
Retningen og rotasjonen på aksene må defineres for å forstå hva som skjer med sensoren. De er det best å kjøre noen tester på akselerometeret og gyroskopet for å se hva som er positiv og negativ retning. Sensorene blir logget ved hjelp av en app'en SensorLog på Iphonen. På appen vises en graf av alle aksene og hva som skjer med sensoren i nåtid. På bakgrunn av dette kan man definere aksene og rotasjonen om dem. Under er skjermbildet av testene for positiv retning på akselerometeret og gyroskopet.



Figur 3.1: Testene sjekker hva som er positiv retning for sensorene.

3.1 Definisjon av aksene og loggingen

I figur 3.1(a) holdes telefonen hele tiden flatt i hånden med skjermen oppover. Ved å akselerere telefonen mot venstre gir det et positivt utslag i x . Telefonen akselerert inn mot brystet gir positiv y og z er positiv når telefonen akselererer mot gulvet med skjermen opp. Videre kan de positive gyroskop retningene defineres ved å holde telefonen holdes likt. Positiv rotasjon rundt x -aksen oppstår ved å vende skjermen på mobilen mot brystet. Rotasjonen rundt y -aksen blir positiv ved å vri skjermen på telefonen mot høyre og positiv rotasjon rundt z når telefonen flatt vrir mot venstre. Ved å se ut de positive aksene fra akselerometeret fra origo, som nevnt i 2.6, gir det positive rotasjoner mot klokken om alle aksene. Alle de positive retningene er definert i figur 3.2.



Figur 3.2: Rotasjonene rundt aksene er mot klokken for positivt utslag på gyroskopet[5]

Et høyrehåndssystem er et system hvor man kan legge pekefingeren langs x -aksen og bøye langefingeren 90 grader ut fra pekefingeren for å treffe y -aksen[20]. Z -aksen reflekteres i tommelen når den peker ut. Dette er det mest brukte aksesystemet. Et slikt system er ikke tilfellet for de definerte aksene og rotasjonene i Iphonen. Her blir det da et venstrehåndssystem.

3.2 Test med kasse

Iphonen har en maksimal samplefrekvens på 100Hz via Sensor Log app'en på telefonen. Det vil si et sample per hundredel av et sekund. Dette bør være mer enn nok ved bruk av kassen det testes med, men er også nok for en faktisk skøyteløper. Samplingsteoremet sier at samplingsraten må være dobbelt så høy som den høyeste frekvenskomponenten i signalet[11]. Menneskelig frekvens ved bevegelse er mindre enn 10Hz[17]. På skøyter vil man antakelig ha maksimalt 3 skyv per sekund. Det vil da gjerne være i startfasen der det bygges fart. Ved toppfart regnes antatte skyv per sekund til 1. Etersom man har et samplingsintervall på 100Hz, vil dette tilsvare rundt 33 målinger per skyv i startfasen og 100 målinger per skyv i toppfart. I toppfart, antatt rundt 60km/t eller 16.6 m/s, vil løperen få en måling per $\frac{16.67}{100} = 0.1667$ m. Dette gir et sample per 16,67cm. Muligheten for å interpolere dataene er også tilstede. For kassen som er dyttet rundt av en skøyteløper og oppnår en toppfart på rundt 5m/s gir det et sample per 5cm.

3.2 Test med kasse



Figur 3.3: Kassen som ble brukt i testen.

Figur 3.3 viser kassen som ble brukt i testen. Kassen er en enkel IKEA kasse snudd på hode for minst mulig friksjon. På toppen ligger en vektskive på 20 kg plassert på en myk skumplate. Denne lille skumplaten sammen med vekten demper de små bevegelsene og restingene som kan oppstå relatert til hakk og andre ujevnheter i isen mens kassen blir dyttet rundt. Det er lettere å ha

3.3 Konstruksjon av posisjonsmålingen

kontroll på hva som skjer med sensoren til en hver tid dersom man benytter en kasse. I selve testen ble kassen dyttet manuelt av en skøyteløper for å få litt fart på kassen.

Iphonen ble lagt på kassen med telefonens skjerm oppover som figur 3.3 viser. I forhold til aksene ble den lagt slik at $y+$ peker bakover og $x+$ peker til venstre. Dette er særdeles viktig å være klar over når rådataene fra sensorene skal behandles. Denne måten å legge telefonen på føles naturlig, men vil føre til et negativt utslag i akselerasjonen i y -aksen ettersom kassen beveger seg fremover.

3.3 Konstruksjon av posisjonsmålingen

Mylaps-systemet brukes på alle internasjonale skøyteløp rundt om i verden[8]. Sørmarka Arena, hvor testingen for oppgaven gjøres, har arrangert verdens-cup på skøyter flere ganger. Hallen har derfor anskaffet Mylaps systemet. Dette brukes som et reservesystem dersom den laserbaserte tidtakingen skulle feile samt til estimering av farten til skøyteløperen ved $\frac{Distanse}{Tid}$. Hallen har muligheten for 9 passeringspunkter i isen. Dette medfører at et fartsestimat blir utregnet tilsvarende ganger rundt banen. Tre av disse sløyfene ligger fast i betongen under isen. Disse punktene er ved 1000m start, 1000m mål og 500m mål, se 2.1. De andre sløyfene må legges ned manuelt hvert år hvis det er behov for det.

3.3.1 Nedlegging av sløyfene

Sløyfene har hvert sitt tilkoblingspunkt rundt banen og det er viktig å sjekke at alle disse fungerer før isen skjæres opp for å legge sløyfen ned. Testen er enkel i den forstand at det kobles til en sløyfe på punktet og drar så en passeringsbrikke over sløyfen. Dette skulle vise seg å være viktig da 4 av 9 punkter ikke fungerte av ukjente årsaker. Hvilke punkter som fungerte, illustreres senere i delkapittel 3.4 figur 3.4.

Etter testingen av tilkoblingspunktene, var det tydelig at 2 sløyfer utenom de faste 3 sløyfene måtte legges ned i isen. Det ble tatt utgangspunkt i de allerede laseroppmålte passeringsstrekene som lå i isen og målebrevet. Fra streken måtte det måles opp et rektangel som vist i figur 2.6,- 30 cm over streken og 30 cm under streken. Det ble brukt en krysslaser for å få en mest mulig rett markering på begge sider langs streken. En enkel tommestokk ble brukt for å plassere laseren 30 cm fra streken. Videre ble isen skåret opp ved hjelp av en stikksag

3.3 Konstruksjon av posisjonsmålingen

med et 1 cm lagt blad. Dette ga et 1 cm dypt kutt hvor ledning ble lagt ned. Her var det viktig å få lagt midten av sløyfen, se 'end box' i 2.6, riktig. Isvann ble brukt til å fylle igjen hullet og skape en glatt isoverflate igjen.

3.3.2 Logging av passeringene

Alle tilkoblingspunktene til sløyfene går via en coax-kabel til sekretariatet. Her står det en dekodeer til det respektive punktet som tar imot alle passeringene. Dekoderen registrerer og behandler de innkommende passeringene. Alle dekodeerne er igjen tilkoblet et internt trådløst nettverk i hallen. En PC tilkoblet samme nettverk kan dermed nå dekodeeren trådløst via dekodeerens gitte IP-adresse.

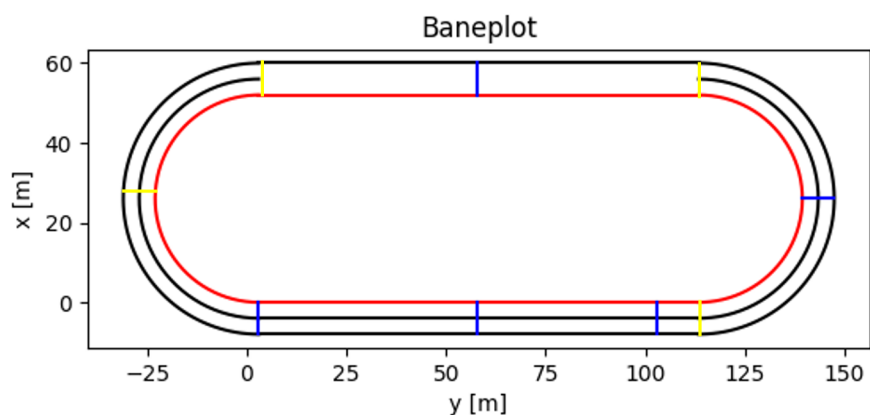
Mylaps hadde ikke mulighet til å bistå med hjelp da deres dataprogram ikke ville lese og koble til de 2 nye aktiverte dekodeerene. Serviceavtalen mellom Mylaps og Sørmarka Arena var utløpt. Disse utfordringene ble løst ved å ta i bruk et tredjeparts dataprogram. Dette er et uoffisielt dataprogram designet av Ladislav Skokan[21]. Programmet leser også passeringene via det interne nettet og rapporterer det i JSON format som så blir skrevet over til en tekst fil. Nedlastningen av programmet er gratis dersom databaselagring ikke er ønskelig. Det var det ikke i dette test tilfellet. Windows PowerShell ble brukt til å kjøre programmet med gode resultater. Det finnes lite dokumentasjon om selve programmet, men antar at funksjonen er tilnærmet lik programmet utviklet av Mylaps. Loggingen av passeringsdata ble manuelt aktivert rett før en test startet og avsluttet rett etter for sortere dataene etter hvilken test som ble gjort

3.4 Modellering av banen

3.4 Modellering av banen

3.4.1 Det globale og lokale aksesystemet

Den lokale orienteringen til telefonen må kunne settes i forhold til det globale aksesystemet. Det globale aksesystemet er banen inne i hallen. Banen blir illustrert ved bruk av Python biblioteket Matplotlib som vist i figur 3.4 under.



Figur 3.4: Det er viktig å legge merke til at x er på den loddrette aksene og y på den horisontale aksene. Legg også merke til at nullpunktet(0,0) hvor kassen startet er 3 meter bak 500m start, se 2.1. Den manglende streken i midten av den ene langsiden viser at det er vekslingsiden. Det finnes ingen markering i banen her.

Som vist i figuren over er x på den loddrette aksene og y på den horisontale aksene. Bakgrunnen for det er å lettere illustrere bevegelsen samtidig som det er en konsekvens av måten telefonen ble lagt på kassen. Telefonen følger sin lokale y-akse i starten. Derfor ble det konkludert med at det ble naturlig å vise y-aksene på den horisontale aksene for det globale systemet også. Derav har det lokale og globale aksesystemet likt utgangspunkt før kassen begynner å bevege seg. Etterhvert som kassen når svingen på banen vil det lokale aksesystemet dreie mot venstre iforhold til det faste globale aksesystemet. Midt i svingen vil lokal y-akse peke langs global x-akse men i motsatt retning gitt positiv y peker bakover på kassen.

3.4.2 Tolkning av passeringene

De gule strekene i figur 3.4 viser de 4 passeringene som dessverre ikke fungerte. Årsaken til at disse ikke fungerte, er ukjent, men hallen er gjort kjent

3.5 Klokkesyknronisering

med problemet og skal utrede manglende funksjon iløpet av sommeren 2022. De 5 blå brukes videre i oppgaven. Linjen kassen skulle følge, er markert i rødt. Fordelen med å følge denne, er at det gir en fin referanselinje når behandlede sensordata skal plottes. Derav vil det være klart hvor mye sensorene avviker fra den virkelige posisjonen kassen hadde.

De blå strekene er passeringslinjene på tvers av løpsbanen. Mylaps systemet registrerer bare at brikken har passert streken en gitt tid, men ikke hvor den har passert på streken. Utfordringen løses når kassen følger den rød linjen i testen. Hver passering kan da få tildelt et punkt i x og y retning på den røde streken. En tabell er satt opp med passeringspunktene, distanse mellom hvert passeringspunkt og den totale distansen kassen har beveget seg.

Nr.	Passeringsrefreanse	(y,x)	Distanse fra siste passering	Total distanse
1	500m start	(0,3)	3m	3m
2	1000m mål	(0,58)	55,215m	58,215m
3	500m mål	(0,103)	44,785m	103m
4	Midt sving nord	(26,139.4)	51,271m	154,271m
5	1000m start	(52,58)	104,995m	259,266m
6	500m start	(0,3)	127,957m	387,223m
7	1000m mål	(0,58)	55,215m	442,438m
8	500m mål	(0,103)	44,785m	487,223m

Tabell 3.1: Dataene er regnet ut med utgangspunkt i målebrevet og er derav nøyaktig med millimeter presisjon. De første 3 meterne er målet opp med tommestokk for plassering av kassen bak streken.

På passeringspunkt nummer 8 i tabell 3.1 ser man at kassen har beveget seg 487,223 meter. Dette reflekterer hvilken betydning vekslingen mellom indre bane og ytre bane har. En skøyteløper går en indre sving og en ytre sving per runde. Distansen kassen har beveget seg ved siste passeringspunkt, forteller her at kassen ikke gikk en ytre sving og derav heller ikke helt 500m. Rundt 13 meter kortere går kassen ved to indre svinger.

3.5 Klokkesyknronisering

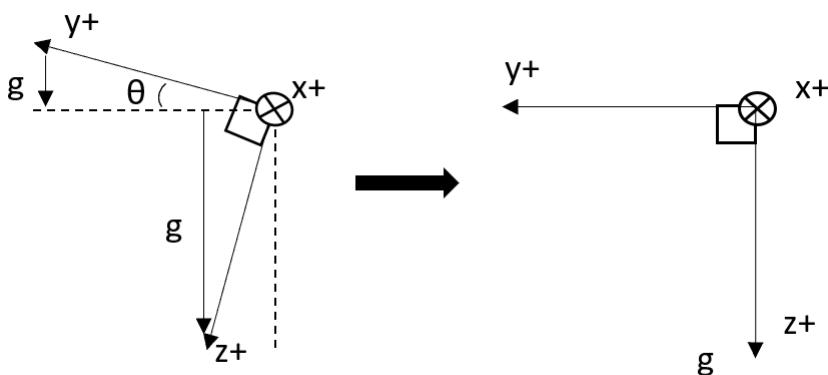
For at de telefonen og posisjonssystemet skal kunne kobles skikkelig sammen forutsetter det at klokkene i systemene er synkroniserte. Posisjonssystemet bruker PC klokken under registrering. PCen er en Microsoft Surface Pro8 med Windows 11 som følger Microsofts klokkeserver Windows.time. Iphonen følger Apple sin klokkeserver, time.apple.com. For å teste de to klokkene ble et enkelt

3.6 Sensorfusjon og posisjonsestimat

Pythonskript satt opp. Skriptet returnerte PC klokken med nøyaktighet ned til millisekunder ved nedtrykket mellomrom-tast. På denne måten kunne telefonen legges på mellomrom-tasten, men uten at den ble trykket ned. En logging av sensordataene på telefonen ble aktivert og et kjapt trykk på telefonen ville trykke ned mellomromstasten samtidig. Ved å se hvor i akselerometerdataene trykke ble registrert kunne tilhørende tid sammenlignes med tiden på PCen. Resultatet av 20 tester ga et maksimalt avvik på ± 0.05 sekund. Konklusjonen ble at klokken var like nok og derfor ble det sett bort fra korrigeringer av dette.

3.6 Sensorfusjon og posisjonsestimat

3.6.1 Prinsippet for aksesystemet



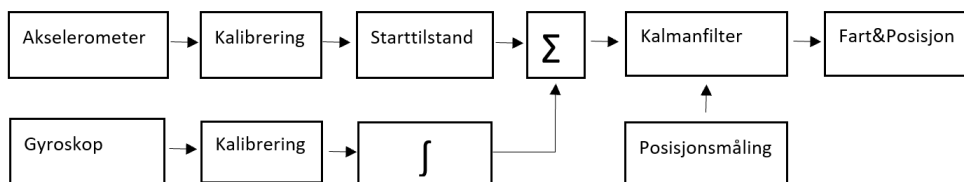
Figur 3.5: Systemet til venstre illustrerer en pitch-rotasjon θ om x-aksen for telefonen. Ønsker å hele tiden ha aksene i akselerometeret horisontalt som vist til høyre.

Prinsippet for fremgangsmåten er å rotere x- og y-aksene slik at de ligger horisontalt[23]. Ved at de ligger horisontalt kan gravitasjonen fjernet fra systemet. Figur 3.5 illustrer en pitch-rotasjon om x-aksen. Sett opp mot aksedefinisjonene i figur 3.2 trenger systemet en rotasjon mot klokken om x-aksen for å rotere y-aksen ned horisontalt. Dette er definert som en positiv rotasjon om x-aksen. I praksis for telefonen er det en gitt rotasjon om y-aksen også slik at x heller ikke er helt horisontal. Ulempen ved at x- og y-aksen ikke ligger helt horisontalt er at gravitasjonen trekker litt i begge akser og skaper forstyrrelser i systemet. Derfor er formålet med å rotere x- og y-aksen horisontalt at all gravitasjonen da kun trekker i z-aksen. På denne måten kan gravitasjonen bare trekkes fra z-aksen og derfor hele systemet. Dette vil gjøre det enklere å se den virkelige akselerasjonen aksene i sensoren blir utsatt for.

3.6 Sensorfusjon og posisjonsestimat

De ønskede vinklene roll og pitch, for å rotere aksene horisontalt, kan finnes ved noen matematiske beregninger på akselerometeret i ro. Ved bevegelse i systemet vil den tilførte akselerasjonen forstyrre utregningene av vinklene og skape dårlige estimater. Etter som akselerometeret ligger i ro i starten av testen kan aksesystemet legges horisontalt på x- og y-aksen ved bruk av rotasjonsmatrisene i likningene 2.9 og 2.10. Rotasjonsmatrisene tar inn de utregnede vinkelen. Den totale orienteringen til systemet kan bestemmes ved å regne ut en yaw vinkel basert på hvordan telefonen beveger seg. Utgangspunktet er at det lokale systemet skal hele tiden ha x-aksen langs den globale x-aksen og likt for lokal y langs global y. Etter som systemet settes i bevegelse rundt banen kan vinkler, som resultat av gyroskopintegrasjon, bidra med å holde denne starttilstanden lik hele tiden. På denne måten skal aksesystemet til akselerometeret holdes likt uavhengig av hvordan telefonen fysisk roterer. Hvilke akser det er akselerasjon i og hvor mye vil avgjøre hvordan systemet beveger seg. Disse korrigerede akselerasjonsmålingene kan videre dobbelintegreres og fusjoneres sammen med passeringmålingene i et kalmanfilter. Slik kan farten og posisjonen til kassen estimeres.

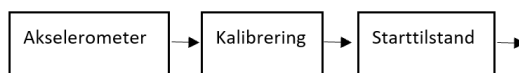
3.6.2 Skjema over veien frem til posisjonsestimatet



Figur 3.6: Skjema for fremgangsmåten av sammenkoblingen

Figur 3.6 viser et skjema for hvordan sensorene blir koblet sammen og behandlingene av dataene for å oppnå fart og posisjonsestimatet. Videre nedover i delkapitlene her vil skjemaet stegvis bli gjennomgått for å tydeliggjøre fremgangsmåten og prosessen.

3.6.3 Finne starttilstanden



Figur 3.7: Fremgangsmåte for starttilstanden

3.6 Sensorfusjon og posisjonsestimat

Akselerometeret og kalibrering

Akselerometerdatene som kommer inn er rådata. Disse må behandles i henhold til kalibreringen fra likning 2.4 i delkapittel 2.3.1. Før starten av testen innhentes a_{max} og a_{min} for alle aksene. For målingene i z legges telefonen flatt med skjermen oppover og et lite datasett logges i 15 sekund før telefonen flippes 180grader rundt for å gjøre det samme. Målingene i y hentes ved å stille telefonen rett opp for så å snu den på hode. X målingene hentes ved å stille telefonen på venstrekant og så på høyrekant. Fordelen med en Iphone 13 Pro er at kantene er helt flate som gjøre det lett å stille den opp. De enkeltvise gjennomsnittene av de 6 målingene som er logget brukes i likningene 2.1-2.3 som fører frem til kalibreringslikningen 2.4. Denne kalibreringen skal gjøres på akselerasjonsdataene i selve testen i hver akse for hvert tidssteg. De kalibrerte dataene kan videre brukes til å finne roll, pitch og yaw(ϕ, θ, ψ). Roll er rotasjonen om y-aksen og pitch er rundt x-aksen i telefonen sitt tilfelle. Yaw kan ikke direkte regnes ut ved kun bruk av akselerometeret, men med kunnskap om hvordan telefonen er orientert i det globale aksesystemet på banen kan det regnes ut en yaw-vinkel for systemet.

Finne roll og pitch

Trigeometri står sentralt når akselerometeraksene skal finne roll og pitch[22]. Rekkefølgen på rotasjonen er viktig og etter som roll er rundt y-aksen og pitch rundt x-aksen blir rotasjonen R_{yxz} . Likningene for vinklene er vist under i 3.1 og 3.2. Utregningen er med utgangspunkt i gjennomsnittet av alle aksene det første sekundet av opptaket hvor systemet ligger i ro. Gravitasjonskonstanten g er 1 eller 9.81m/s.

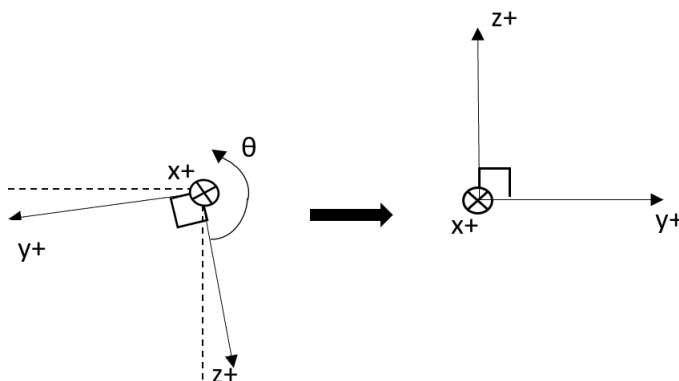
$$\tan \phi_{yxz} = \frac{\frac{\bar{a}_x}{g}}{\frac{\bar{a}_z}{g}} \quad (3.1)$$

$$\tan \theta_{yxz} = \frac{\frac{\bar{a}_y}{g}}{\sqrt{\left(\frac{\bar{a}_x}{g}\right)^2 + \left(\frac{\bar{a}_z}{g}\right)^2}} \quad (3.2)$$

Merk at dette er gitt et høyrehåndssystem og ikke venstre som i telefonens tilfelle. Det er derfor en fordel å jobbe i radianer ved utregning av invers tangens for å finne vinklene. Bruken av `arctan2` i Python er å foretrekke over `arctan`. `Arctan2` tar inn to argument og regner ut vinkelen i riktig kvadrant. `Arctan` vil ha to speilede løsninger om en akse i hver sin kvadrat. I Matlab brukes `atan2`. I oppgaven viser det spesielt fordelene ved å bruke `arctan2`. Likningene 3.1

3.6 Sensorfusjon og posisjonsestimat

og 3.2 er høyrehåndsbaserte likninger og $\arctan2$ funksjonen roterer systemet til kvadranten det ville vært i dersom det var et høyrehåndssystem. Merk at systemet ikke endres fra venstre til høyrehåndssystem, men det roteres om x- og y-aksen som om det var det.



Figur 3.8: Pitchrotasjon rundt x-aksen. Fordelen er at positiv lokal y-akse nå peker fremover i samme retning som global y-akse.

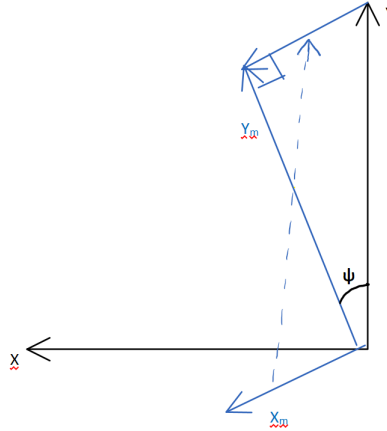
Figur 3.8 under illustrerer hva som skjer i telefonens tilfelle. Her roteres y-aksen rundt til å følge y-aksen globalt tillegg til å bli lagt horisontalt. Den utregnede vinkelen for pitch er positiv. Det vil si at Det skjer en positiv rotasjon mot klokken om x-aksen slik som figuren viser.

Roll og pitch bruker rotasjonsmatrisene i likning 2.9 og 2.10 for å rotere horisontalt med den utregnede vinkelen. Først multipliseres rotasjonsmatrisene R_x og R_y sammen for så ta inn en 3×1 matrise med akselerometerdataene i x, y og z for hvert tidssteg. Det er viktig å notere at alle akselerometerdataene skal roteres med disse vinklene, ikke bare det første sekundet. Videre gjenstår det å definere orienteringen til de horisontale aksene. For å fortelle hvilken retning det er regnes yaw ut med utgangspunkt i at all akselerasjon når kassen begynner å bevege seg er i positiv y-akse.

Finne yaw

Yaw må estimeres på bakgrunn av hva som er kjent om systemet og hvordan det er tenkt å bevege seg. Ettersom y lokalt skal bevege seg langs y globalt kan vinkelen matematisk regnes ut ved å sette et krav til hvor akselerasjonen skal skje. Telefonen ligger allerede med sin lokale y-akse langs den globale yaksen. All akselerasjon i starten, når systemet sette i bevegelse, skal i utgangspunktet skje i den lokale y-aksen.

3.6 Sensorfusjon og posisjonsestimat



Figur 3.9: Illustrasjon av hvordan yaw kan finnes basert på kunnskap om retningen kassen beveger seg. Det blå reflekterer målingen.

På bakgrunn av figur 3.9 og ved bruk av enkel trigonometri kan likning 3.3 utledes.

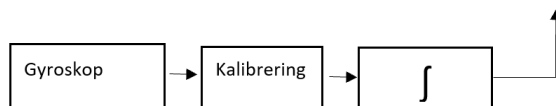
$$\tan \psi_{yxz} = \frac{\bar{a}_x}{\bar{a}_y} \quad (3.3)$$

\bar{a}_x og \bar{a}_y settes til å være gjennomsnittet av de første 10 sekundene av bevegelse i de nye horisontalkorrigerede dataene. Denne vinkelen roterer systemet hvor x og y allerede er horisontalt til at lokal y-akse ligger tilnæremet likt med global y-akse. Roteringen av systemet gjøres med matrisen i likning 2.11.

Etttersom roll, pitch og yaw nå er definert og korrigert slik at man oppnår riktig starttilstand, kan gravitasjonen trekkes fra systemet. Optimalt sett skal bare z-aksen bli påvirket av gravitasjonen. Likvel blir et gjennomsnitt av alle aksene, mens systemet er i ro, trukket fra alle målingene i sin respektive akse. Likningene 2.5 og 2.6 blir brukt til trekke fra denne nye offseten i alle aksene.

3.6 Sensorfusjon og posisjonsestimat

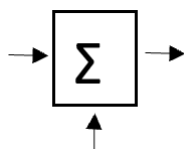
Gyroskopkorrigeringen



Figur 3.10: Fremgangsmåten for gyroskopintegreringen

Kalibreringen av gyroskopet blir gjort ved å finne offseten til gyroskopet over ett minutt i en annen test[23]. Denne testen forutsetter at systemet ligger helt i ro hele tiden. Bakgrunnen er at gyroskopet kan drifte mye over tid. Med det menes at gyroskopmålingene flytter seg sakte bort fra null selv om det ikke skjer noe med sensoren. Ved integrasjon av denne driften blir feilmarginene på vinklene store over tid. Derfor er det spesielt viktig å få fjernet offset over minst like lang tid som testen. Drift må regnes med uansett i denne fremgangsmåten her.

Gyroskopet måler som kjent vinkelhastigheten i rads/s, men det er selve vinkelen som er mest interessant. Vinkelen kan finnes ved å integrere vinkelhastigheten til vinkelposisjon. Disse vinkelene vil fortelle hvilken vei og hvor mange radianer aksene roterer til en hver tid. Derfor kan disse vinklene også tolkes som roll, pitch og yaw vinkler. Disse vinklene sammen med starttilstanden gå inn i figur 3.11



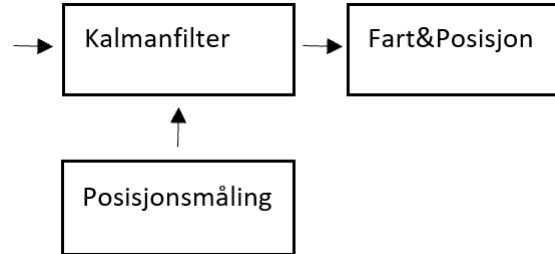
Figur 3.11: Selve gyroskopkorrigeringen av starttilstanden

Selvekorrigeringen skjer ved at de korrigerte akselerasjonsdatene føres inn som gitt under i figur 3.4:

$$\begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = \begin{bmatrix} c(\phi) & 0 & s(\phi) \\ 0 & 1 & 0 \\ s(\phi) & 0 & c(\phi) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) \\ 0 & s(\theta) & c(\theta) \end{bmatrix} \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{xk} \\ a_{yk} \\ a_{zk} \end{bmatrix} \quad (3.4)$$

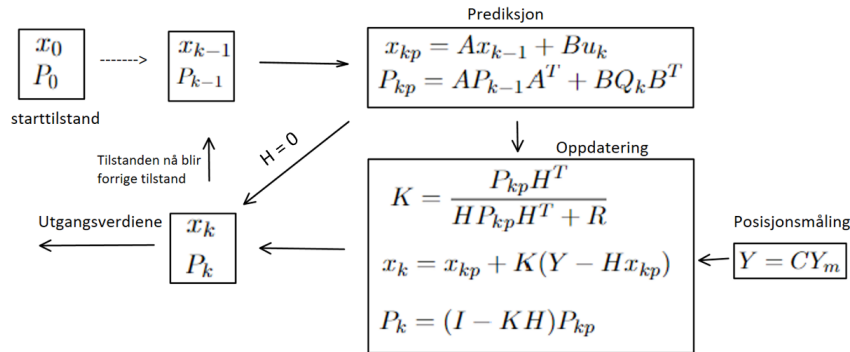
3.7 Kalmanfilter design- håndteringen av data

3.7 Kalmanfilter design- håndteringen av data



Figur 3.12: De korrigerede dataene føres inn i kalmanfilteret som produserer de ønskede estimatene

Kalmanfilteret sin oppgave er å koble sammen behandlet sensordata og posisjonsdata for å gi et best mulig estimat av hvor løperen befinner seg. Et skjema for et enkelt kalmanfilter er vist i figur 3.13 og bruker ligningene fra delkapittelet om kalmanfilter i kapittel 2[1][6].



Figur 3.13: Verdt å notere seg at dersom $H = 0$, hopper filteret over oppdateringsfasen. Konsekvensen blir at bare integreringen av akselerometeret står for oppdateringen av tilstanden.

3.7.1 Modellering av integrasjonen

Det blir tatt utgangspunkt i en matematisk beskrivelse av prosessen som skal håndteres av kalmanfilteret. Her er det enkel- og dobbelintegrasjon av akselerometerdataen som skal modelleres. Fart og posisjon i x og y er av interesse og blir brukt ved konstruksjonen av modellen. Likningen 3.5,3.6,3.7 og 3.8 under viser modellen for integrasjonen.

3.7 Kalmanfilter design- håndteringen av data

$$x_{k+1} = x_k + \dot{x}_k \Delta t + \frac{\ddot{x}_k \Delta t^2}{2} \quad (3.5)$$

$$y_{k+1} = y_k + \dot{y}_k \Delta t + \frac{\ddot{y}_k \Delta t^2}{2} \quad (3.6)$$

$$\dot{x}_{k+1} = \dot{x}_k + \ddot{x}_k \Delta t \quad (3.7)$$

$$\dot{y}_{k+1} = \dot{y}_k + \ddot{y}_k \Delta t \quad (3.8)$$

Likningene ovenfor er basert på trapesmetoden for integrering[18]. Det er en numerisk metode for beregning av verdien til et bestemt integral av en funksjon. Posisjonen er representert av x og y , \dot{x} og \dot{y} er fart og \ddot{x} og \ddot{y} er akselerasjon ved gitte tidssteg.

3.7.2 Utledning av matrisene i filteret

Kalmanfilteret estimerer fire tilstander: posisjon og fart i x og y . Tilstandene blir lagret i en tilstandsvektor eller en 4×1 matrise vist i x_k i likning 3.9. Ettersom kassen ligger i ro på et gitt punkt vi kaller $(0,0)$, kan vi også sette opp starttilstanden for systemet som vist i x_0 i likning 3.9.

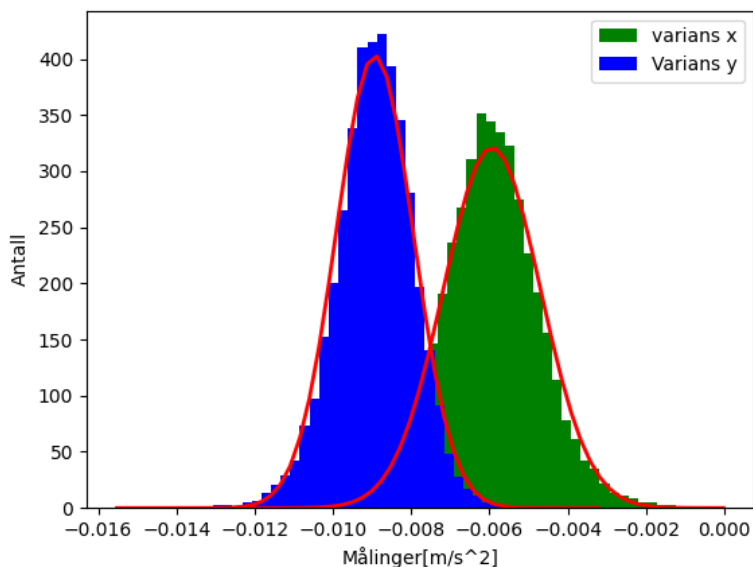
$$x_k = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, x_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.9)$$

Videre defineres prosesstøymatrisen P_k som en 4×4 matrise hvor usikkerheten i prosessen er i form av en kovariansmatrise gitt i likning 3.11.

$$P_k = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 & \sigma_{x\dot{x}}^2 & \sigma_{x\dot{y}}^2 \\ \sigma_{yx}^2 & \sigma_y^2 & \sigma_{y\dot{x}}^2 & \sigma_{y\dot{y}}^2 \\ \sigma_{\dot{x}x}^2 & \sigma_{\dot{x}y}^2 & \sigma_{\dot{x}}^2 & \sigma_{\dot{x}\dot{y}}^2 \\ \sigma_{\dot{y}x}^2 & \sigma_{\dot{y}y}^2 & \sigma_{\dot{y}\dot{x}}^2 & \sigma_{\dot{y}}^2 \end{bmatrix}, P_0 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix} \quad (3.10)$$

Startverdiene for tilstandsusikkerheten settes til en antatt usikkerhet i tilstandene hvor usikkerheten er variansen i estimatet. Ettersom tilstanden er i ro og har et gitt startpunkt, kan det velges lave verdier for usikkerheten som utgangspunkt. Antar også at det ikke er korrelasjon i de forskjellige tilstandsestimaterne for å gjøre det enklere. Herav kan P_0 utledes ved at det bare ses på diagonalen i matrisen og setter en antatt usikkerhet til 10 for $\sigma_x^2, \sigma_y^2, \sigma_{\dot{x}}^2$ og $\sigma_{\dot{y}}^2$. Uansett hvilke verdier som blir satt her, vil systemet regne ut nye estimater som setter nye verdier for usikkerheten. Systemet svinger seg raskt inn til riktige estimat hvis verdien er alt for stor eller lav i starten.

3.7 Kalmanfilter design- håndteringen av data



Figur 3.14: Variansen i x og y når systemet ligger i ro i 60 sekund

Den matematiske modellen kalmanfilteret her baserer seg på, må også modelleres i matriser. Likningene 3.5 til 3.8 illustrerer matrisene A og B under:

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \quad (3.11)$$

Det er matrisene A og B som tar for seg integreringen for hvert tidssteg. Korrigerte akselerasjonsmålinger legges inn i målingsmatrisen u_k da det er akselerasjonen som skal integreres. u_k er vist i likningen 3.12 under.

$$u_k = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}, \quad (3.12)$$

Den predikerte usikkerheten for estimatene er også avhengig av kovariansmatrisen Q_k for akselerasjonsmålingen. Det vil si variansen i målingen. Det antas her at det ikke er noen avhengighet mellom \ddot{x} og \ddot{y} . Dette kalmanfilteret er definert for å gjelde et lineært system med hvit og normalfordelt støy. Dataene fra sensoren i ro over en gitt periode må studeres.

3.7 Kalmanfilter design- håndteringen av data

Figur 3.14 viser at x- og y-aksen er normaltdelt i støyen. Samtidig er variansen mindre i y(σ_y^2) enn i x(σ_x^2). Denne informasjonen brukes videre i matrisen Q_k vist i likningen 3.14 under.

$$Q_k = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad (3.13)$$

H er en matrise som representerer hvilke av tilstandsestimatene som er dekket av en måling. I dette tilfellet er det posisjonssestimatene som dekkes av en måling da akselerasjonsmålingen er en måling i prediksjonsfasen. Ettersom x_k er en 4x1 matrise med x og y som element (1,1) og (1,2)c, må H bli en 4x2 matrise som vist under.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.14)$$

Videre skal målingen tas inn og matrise C i likning 3.15 er her en 2x2 identitetsmatrise. y_m er selve posisjonsmålingen i x- og y-koordinater.

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, y_m = \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.15)$$

Målingen av posisjon har også et usikkerhet i seg som blir reflektert i matrisen R i likning 3.16. Det antas ingen korrelasjon mellom x og y.

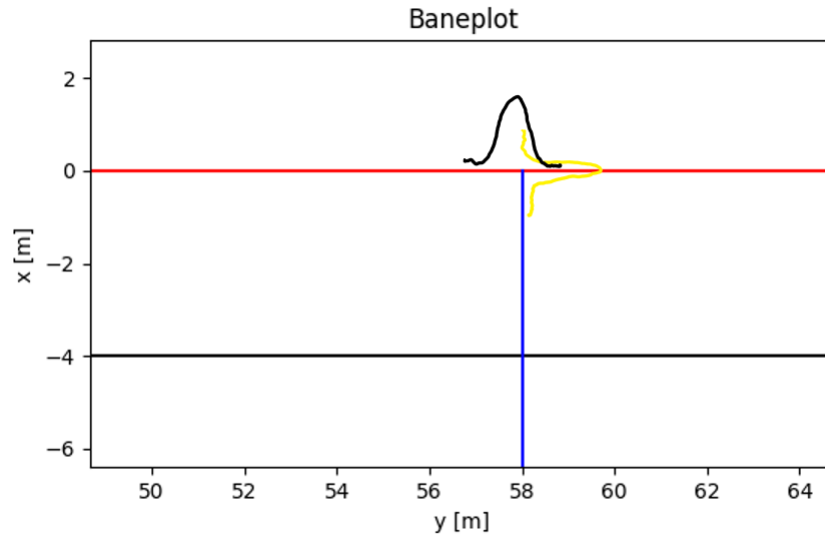
$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (3.16)$$

Variansen her blir bare en kvalifisert antakelse ettersom kassen passerer et gitt forhåndsdefinert punkt. Likevel er det ukjent om kassen og derav akselerometeret passerte akkurat på punktet. Figuren under viser hvordan man ser for seg denne usikkerheten for passeringen. Som illustrert i figur 3.15 er den svarte fordelingen bredere enn den gule og derav en større usikkerhet for når kassen faktisk passerte linjen. Den gule fordelingen er det lettere å ha kontroll på da kassen tilnærmet hele tiden følger linjen mens den dyttes. Nøyaktig når Mylapsbrikken ble registrert vet man ikke helt sikkert basert på den uoffisielle programvaren som blir brukt for å registrere passeringen. σ^2 settes til 0.5 for y og henholdsvis blir σ^2 satt til 0.25 for x på bakgrunn av denne informasjonen. Det er viktig å notere at dette endrer seg i passeringen i svingen. Da blir det motsatt for x og y ettersom kassen da peker rett i den definerte globale x-aksen.

Siste matrise som må gjøres rede for er I. Dette er bare en enkel 4x4 identitetsmatrise som vist i likning 3.17.

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

3.7 Kalmanfilter design- håndteringen av data



Figur 3.15: Gul antatt fordeling for x og svart for y. Mindre varians i y enn x.

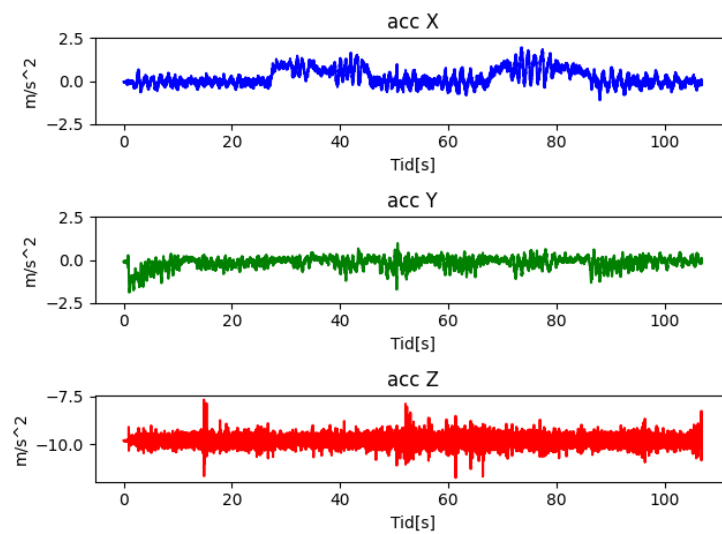
Innlesning av målte data skjer kolonne for kolonne i en $2 \times n$ matrise for akselerasjonen i x- og y-aksen samt posisjonen i x og y. n er her totalt antall målinger. Alt ligger nå tilrette for å legge filteret inn i et digital hjelpemiddel. Python er brukt for å kjøre filteret i denne oppgaven. Tilstandsmatrisene x_k lagres for være runde for senere behandling og plotting før den går videre til å bli forrige steg. Prosesstøymatrisen P_k oppdateres for hver runde, men hentes bare ut tilslutt.

Kapittel 4

Resultat

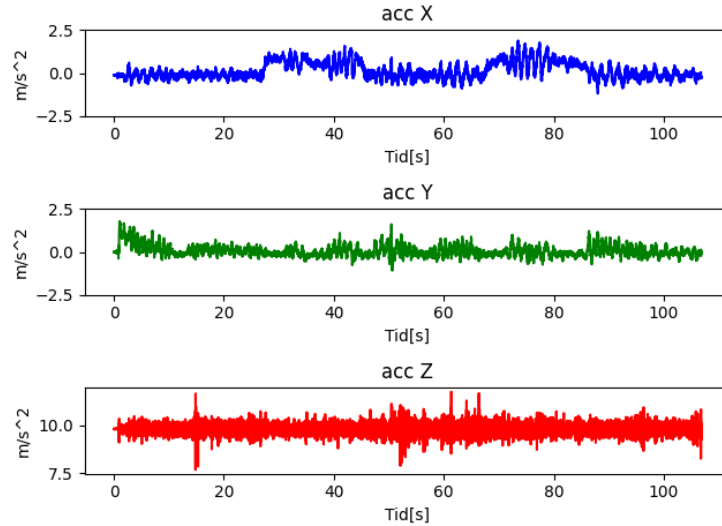
Resultatene fokuserer i hovedsak på x- og y-aksene og plottet av disse. Z må likevel være med for at systemet skal bli riktig orientert samtidig som det reflekterer en del unøyaktigheter i fremgangsmåten.

4.1 Starttilstanden



Figur 4.1: Rådata fra akselerometeret. Merk at loddrett akse i acc z er forskjellig fra x og y

4.1 Starttilstanden



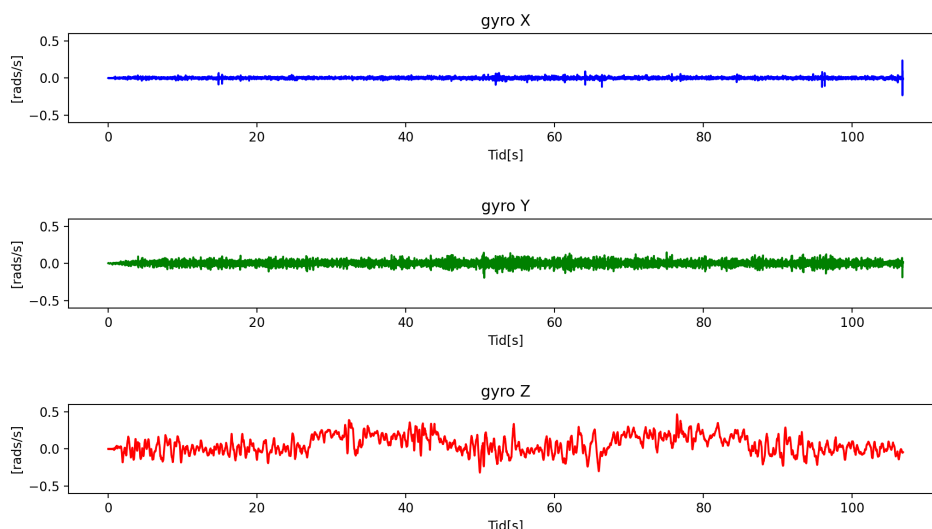
Figur 4.2: Roterte akselerasjonsdata til starttilstanden. Merk at loddrett akse for acc z er forskjellig fra x og y og motsatt av acc z i figur4.1.

Figur 4.1 viser rådataen som ble logget av akselerometeret under testen. Dataene her bekrefter aksedefinisjonene fra tidligere. Det er tydelig at akselerasjonen i y er negativ da positiv y-akse peker bakover. Videre viser figur 4.2 de snudde akselerasjonsdataene for å oppnå den ønskede starttilstanden. Her ses det tydelig virkningen av $\arctan2$ og de høyrehåndsbaserte likningene 3.1 og 3.2 i delkapittel 3.6.3. Det som figur 3.8 i samme delkapittel viser kan tydelig ses i grafene til y og z. Her er akselerasjonen i begge grafene blitt positiv. Legg merke til den loddrette aksene i figur 4.1 er z rundt -9.81 mens i figur4.2 er den rundt $+9.81$.

4.2 Gyroskopdataene

4.2 Gyroskopdataene

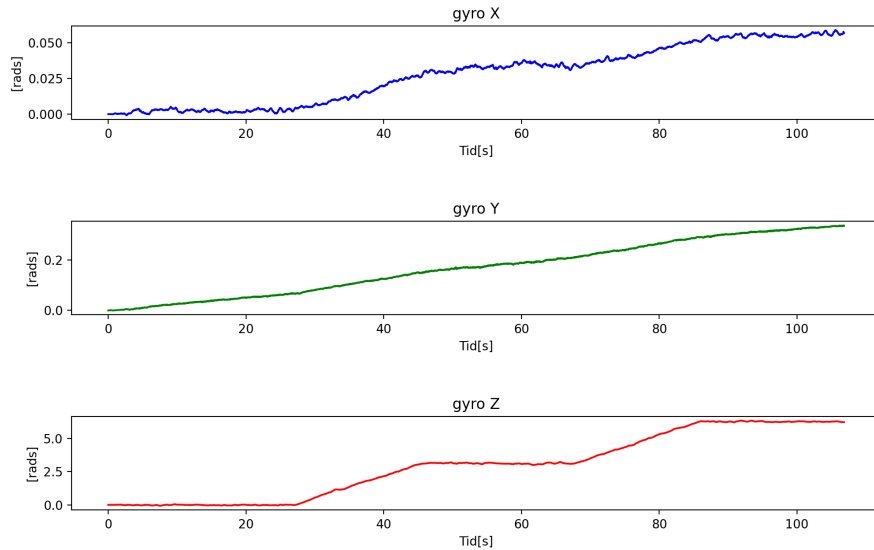
Gyroskopdataene er plottet under og viser rotasjonshastigheten tilkassen mens den går rundt banen. Som vist i grafene er det klart at det er mer bevegelse rundt z-aksen enn x- og y-aksen. Likevel kan grafen fortelle at telefonen rullet litt mer fra side til side rundt y-aksen enn den vippet opp og ned rundt x-aksen.



Figur 4.3: Målingene fra gyroskopet med offseten trukket fra.

Videre er gyroskopdataen fra figur 4.3 integrert i figur 4.4. Her kommer det frem radianene systemet har rotert. Rundt z-aksen roteres det 2π som tilsvarer hen hel runde rundt banen. Der hvor grafen er flat er det tilnærmet ingen rotasjon rundt z-aksen. Etter rundt 27sekunder begynner grafen å stige oppover. Ved å se på den definerte positive rotasjonsretningen som er mot klokken i rundt z-aksen viser grafen at systemet roterer mot venstre. Dette stemmer med retningen rundt banen. Først roterer den π radianer ut på den andre langsiden og så π radianer tilbake til første langsiden. Derav 2π radianer tilsvarende en hel runde.

4.2 Gyroskopdataene

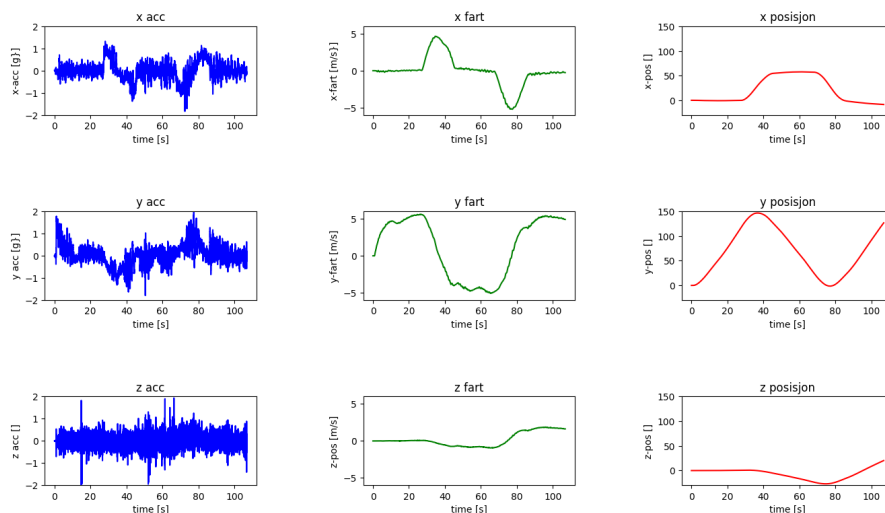


Figur 4.4: De integrert gyroskopdataene fra figur4.3. For å vise problemene med drift er ikke den loddrette aksene for radianene like for alle plottene.

Legg merke til at loddrett akse i figur 4.4 ikke er lik for alle plottene. Dette er gjort for å illustrere drift i integreringen. Selv om gyroskopet har fratrukket offseten vil det forekomme drift. Grafene til y tilsier eksempelvis at telefonen har rullet 0.3 radianer mot venstre. Omregnet til grader blir det tilnærmet 17 grader som helt klart ikke stemmer da telefonen ligger flatt under hele testen. Hadde driften og testen fortsatt i drøyt en 15 minutt hadde driften fortalt systemet at telefonen hadde rotert 180 grader slik at skjermen ligger nedover. Drift i z forekommer også, men er ikke like tydelig å lese det ut av grafen.

4.3 Gyroskopkorrigeringen av starttilstanden

4.3 Gyroskopkorrigeringen av starttilstanden

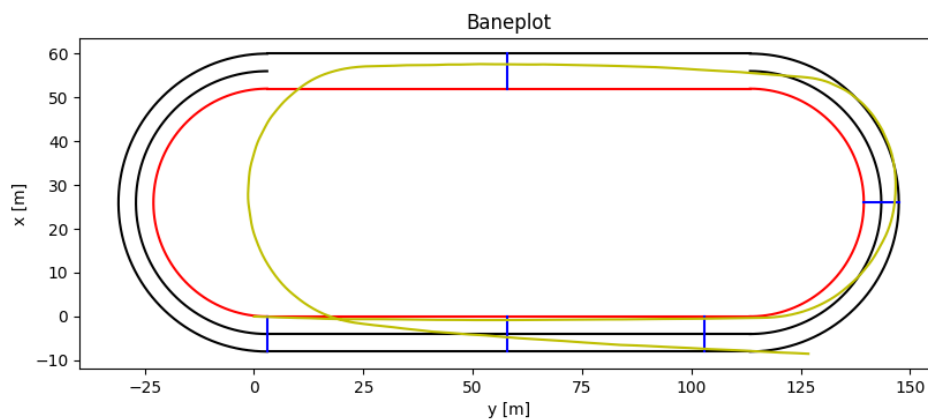


Figur 4.5: Akselerasjon, fart og posisjon av gyrokorrigert data.

Resultatet av gyrokorrigeringen av starttilstanden er som vist i figur 4.5. Her er dataene også trapesintegrert til fart og posisjon i alle aksene. Farten i x og y stemmer fint med virkeligheten basert på den bestemte starttilstanden aksesystemet står under bevegelsen. Farten i x er tilnærmet null frem til svingen og når toppfarten sin midt i svingen ettersom kassen da peker rett opp global x-akse da blir også all akselerasjon målt i lokal x-akse. Ut på langsiden er farten null igjen ettersom kassen følger y-aksen. For farten i y-aksen øker den mye i starten før den holdes stabil frem til svingen. I svingen reduseres farten mens farten i x øker. Ut på langsiden igjen er farten for y negativ da kassen går motsatt vei av definert positiv y-akse i starttilstanden. Slik virker det for x og y aksene i siste svingen igjen, med med motsatt fortegn. Det som klart skiller seg ut iforhold til virkeligheten er farten i z. Etter som telefonen ligger flatt på en kasse skal det være tilnærmet null fart i z. Posisjonen til z forteller at telefonen etter 75 sekund har sunket 30m ned i isen. Ved slutten av testen flyr telefonen 15m over isen. Dette er helt klart feil og det er et resultat av den nevnte integrerte driften i gyroskopet. Driften snur systemet litt ut av starttilstanden slik at z ikke lenger peker rett opp og derfor mottar z litt av akselerasjonen som x og y aksene egentlig skulle hatt. Det fører også til at posisjonen i x og y heller ikke blir helt som ønsket. Posisjonen i y skulle eksempelvis beveget seg ned til rundt -22 etter 75 sekund på grunn av at kassen er bak startpunktet for testen gitt den positive y aksene fremover fra punktet.

Under i figur 4.6 plottes posisjonen for x og y. Det er tydelige store avvik fra

4.3 Gyroskopkorrigeringen av starttilstanden



Figur 4.6: Posisjonsdataene i x og y fra 4.5 plottet i banemodellen.

den virkelige røde linjen som kassen fulgte. Videre med utgangspunkt i likning 4.1[14] kan den totale distansen for kassens bevegelse i x og y vises ved likning 4.2.

$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (4.1)$$

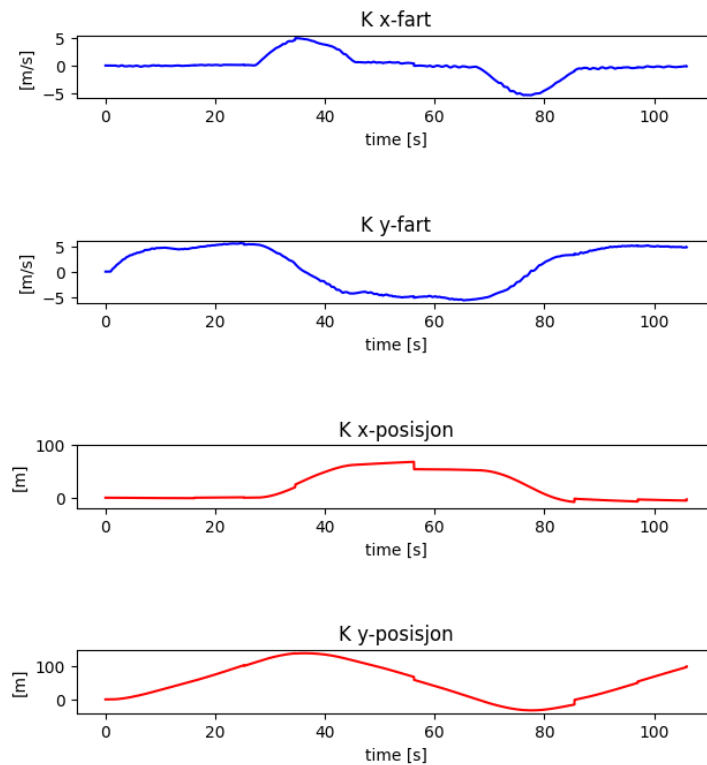
$$d = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (4.2)$$

Den totale distansen d blir 489,557 meter i dette tilfellet her. Sammenlignet med den virkelige linjen som er 487,223meter vis i tabell 3.1 viser det at selve distansen kassen går er ikke så langt fra virkeligheten. Det som ikke er medregnet her er distansen som den har beveget seg i z-aksen. Denne distansen stemmer som kjent ikke, men regnes den med blir $d = 499,391$

4.4 Implementering av data i kalmanfilteret

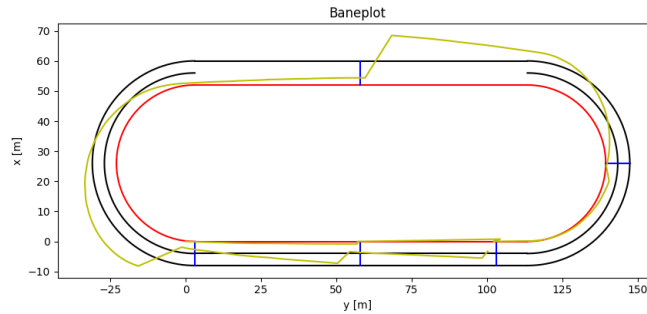
4.4 Implementering av data i kalmanfilteret

Kalmanfilteret skal bidra med å gi et bedre estimat for fart og posisjon. Alle matriser er definert som i delkapittel 2.7. De gyroskopkorrigerede akselerasjonsdatene blir tatt inn i u_k for å trapesintegreres til fart- og posisjonsestimater. Akselerasjonsdataene legges inn som en $2 \times n$ matrise og itereres gjennom en kolonne per tidssteg. n er her antall målinger. Filteret sjekker hele tiden for passeringssdata fra Mylapssystemet og korrigerer for dette dersom det finnes for tidssteget.



Figur 4.7: Estimaterne av fart og posisjon

4.4 Implementering av data i kalmanfilteret



Figur 4.8: Posisjonene x og y i 4.7 plottet på banen.

Posisjonsgrafene for x i figur 4.7 viser et tydelige hakk i grafene på rundt 55 sekund. Det reflekteres godt i baneplottet i figur 4.8 også ved posisjonen mellom midten av svingen og ut til 1000m start. Flere mindre hakk i grafene finnes også. Bakgrunne for disse hakkene er at posisjonen fra den dobbelintegreerte akselerasjonen ønsker å være en annen plass en hvor kassen faktisk var. Hver gang resultatet av akselerasjonsmålingen blir korrigert inn til faktisk posisjon så fortsetter integreringen bare videre fra det nye punktet i den retningen den hadde tenkt seg før den ble korrigert. Plottet på banen illustrerer godt utfordringene med få posisjonskorrigeringet systemet.

4.4 Implementering av data i kalmanfilteret

$$P_{res} = \begin{bmatrix} 1.02e^{-1} & 0 & 1.44e^{-3} & 0 \\ 0 & 2.55e^{-2} & 0 & 3.63e^{-4} \\ 1.44e^{-3} & 0 & 3.16e^{-5} & 0 \\ 0 & 3.63e^{-4} & 0 & 8.05e^{-6} \end{bmatrix} \quad (4.3)$$

Resultatet av prosessstøymatrisen i likning 4.3 gir videre en indikasjon på usikkerheten i målingene. Et raskt tilbakeblikk i den definerte matrisen i likning 3.11 forteller hva de forskjellige tallene representerer. Fra resultatmatrisen ser man at variansen i posisjonen (σ_x^2, σ_y^2) er rundt 10000 ganger større enn i farten. Dette er også illustrert på grafene i 4.7. Sammenlignes x-fart og x-posisjon ses hakkene i grafen mye tydeligere i posisjonen enn farten. Det er altså mye større usikkerhet knyttet til posisjonen enn farten. I tillegg viser det seg at det er korrelasjon mellom fart og posisjon i matrisen. $\sigma_{x\dot{x}}, \sigma_{\dot{x}x}, \sigma_{y\dot{y}}$ og $\sigma_{\dot{y}y}$ har her fått utregnede verdier. Det er naturlig at det er korrelasjon mellom fart og posisjon da disse i utgangspunktet kommer fra ren integrasjon av akselerometerdataene med noen få korreksjoner fra posisjonssystemet. $\sigma_{x\dot{x}}$ og $\sigma_{\dot{x}x}$ har lik verdi etter som rekkefølgen av \dot{x} og x ikke betyr noe. Det samme gjelder y . Det er bare registrert korrelasjon mellom x og \dot{x} og mellom y og \dot{y} . Ikke noen registrert korrelasjon mellom x og y eller \dot{x} og \dot{y} .

Ved bruk av likning 4.2 kan den totale distansen etter kalmanfilteret har estimert posisjonen også regnes ut. Kalmanfilteret gir ut $d = 545,242$ meter. Det gir rundt 50 meter mer enn virkeligheten.

Kapittel 5

Diskusjon/konklusjon

Det ble støtt på en del utfordringer under arbeidet med denne oppgaven. Usikkerhetene og potensielle fremtidige forbedringer drøftes her. Alternative løsninger blir også vurdert.

5.1 Sensorfusjonering

Det er en utfordring å få sensorene skikkelig kalibrert. Det er grunn til å tro, basert på plottene av posisjonene fra resultatene, at både akselerometeret og gyroskopet ikke er kalibrert godt nok. Dette viser seg i form av at den integrerte gyroskopdriften roterer systemet uten at det faktisk roterer. Dette medfører at akselerasjonene i akselerometeret blir oppfattet i feil akse. Det vises tydelig ettersom z-posisjonen i figur 4.5 har store avvik iforhold til virkeligheten. For en skøyteløper vil det bety at skøyteløperen flyr over isen eller befinner seg under isen. Dataene fra sensorene på kassen er langt fra nøyaktige nok til å kunne gi verdi for skøyteløperen. Likevel indikerer målingene at dersom sensorene er håndtert og kalibrert riktig kan det gi nyttig informasjon om fart og posisjon på isen. Det er nok en lang vei å gå til et slikt system kan brukes i praksis.

Mer avanserte fusjoneringsmetoder kan vurderes for videre arbeid. En metode hvor gyroskopet og akselerometeret hele tiden regner ut estimater sammen hvert tidssteg kan være en fin vei videre. Her vil det også være mulighet for å trekke bort gravitasjonen. Det vil kreve noen mer avanserte matematiske utregninger, men skal til gjengjeld gi gode resultater. Det er sannsynligvis noe slikt som blir brukt i de større kommersielle og militære systemene.

5.2 Kvaternioner fremfor Eulervinkler

5.1.1 Magnetometer

Et fint supplement til akselerometeret og gyroskopet er et magnetometer. Iphonen har et magnetometer tilgjengelig som kan bidra ved en referanseorientering til systemet. Magnetometeret måler magnetisk feltstyrke, magnetisk flukstetthet og magnetiske egenskaper i materialer [9]. Styrken det kan bidra med i dette systemet, er at det kan finne magnetisk nord. Slik kan det orientere systemet og benyttes som et digitalt kompass i denne sammenhengen[13]. Det er da yaw-vinkelen i dette systemet som kan estimeres ved det digitale kompasset. På denne måten er det ikke behov for å regne ut yaw-vinkelen med en forhåndsbestemt retningen som systemet skal bevege seg i.

5.1.2 Ny type kalmanfilter for sensorfusjonering

Et noe mer avansert kalmanfilter kan brukes for å slå sammen de tre treghets-sensorene, akselerometer, gyroskop og magnetometer. På denne måten kan usikkerhetene kontrolleres og modelleres. Dette vil styrke estimatene som de sammen gir av systemets orientering og posisjon. Selve prosessmodellen kan være noe utfordrende å utvikle, men denne skal kunne gi det beste resultatet ved riktig oppsett. Kalmanfilteret er det som kanskje er mest vanlig å bruke i navigasjon-, industri- og militærsystemer på grunn av sine modellerte usikkerheter. Kalmanfilteret kan hjelpe systemet til å være uavhengig av ytre signaler. I denne studien innebærer det at det ikke er nødvendig med posisjonskorrigeringer fra passeringssløyfene i isen.

5.2 Kvaternioner fremfor Eulervinkler

Kvaternioner er et alternativ til Euler-vinklene og egentlig en bedre løsning. Det er riktig nok litt tyngre å jobbe med og forstå seg på kvaternioner[2]. Kvaternioner er komplekse tall på formen $w + xi + yj + zk$, hvor w , x , y og z er reelle tall mens i , j og k er imaginære tall kalt Hamilton-identiteten etter fysikeren som definerte kvaternionene. Under er enhetskvaternionet definert i likning 5.1

$$w^2 + x^2 + y^2 + z^2 = 1 \quad (5.1)$$

Enhetskvaternionet kan betraktes som en kvaternionkule. Multiplikasjon av to enhetskvaternioner gir et nytt enhetskvaternion. I et fullverdig system for

5.3 Posisjonssystemet

orientering på banen kan det være hensiktsmessig å bruke kvaternioner for orienteringen.

5.3 Posisjonssystemet

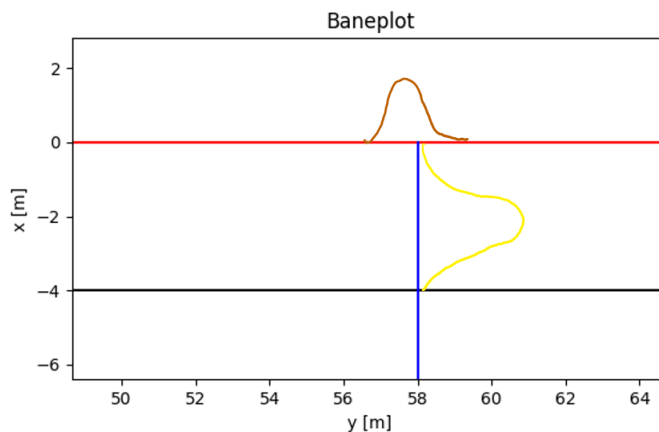
Det vil være ønskelig å få flere posisjonsmålinger. De 4 passeringspunktene som ikke fungerte ville sannsynligvis styrket estimatene og korreksjonen i integrasjonen av akselerometerdataene. I søndre sving er det ikke noen sløyfer og det blir derfor over 150 meter uten noen korreksjon. Det vil tillate estimatene å gå godt utenfor virkelig posisjon. Dataene blir unøyaktige slik at skøyteløperen vanskelig vil kunne nyttiggjøre seg av disse. Sørmarka Arena er gjort kjent med problemet og skal se på mulighetene for å utbedre ledningsnettets samt oppgradere systemet. Systemet som brukes nå er egentlig utdatert og vanskelig å få service på. Likevel har det fungert relativt tilfredsstillende for oppgaven med de sløyfene som fungerte og den uoffisielle programvaren.

For posisjonsmålingen som kommer inn i filteret kan det være en utfordring med tanke på når brikken egentlig ble registrert. For eksempel kan brikken ha blitt registrert en meter før punktet som er definert i kalmanfilteret for passeringen. Dette vil naturlig nok skape problemer for filteret da det blir korrigert til en plass som ikke stemmer. Samtidig blir variansen også feil da denne er sentrert rundt punktet. Manglende nøyaktighet for når brikken blir lest er ulempen med å bruke det uoffisielle systemet som ble brukt her da dokumentasjonen er dårlig. Dette kan løses ved å få tilgang til det offisielle systemet til Mylaps. Her har de kontroll og prosesserer passeringene på riktig måte. Ettersom det brukes som tidtaking i alt fra skøyteløp til formel 1, er de nødt til å få måleresultatet midt på sløyfen. Variansen ved bruk av det offisielle systemet vil da være markant lavere i filteret ettersom passeringen er tilnærmet absolutt.

5.4 Kalmanfilteret

Kalmanfilteret som ble konstruert gjør det lettere å legge til nye sensorer og målinger. Eksempelvis kan en fartsmåling enkelt legges inn der posisjonsmålingene kommer inn. Kalmanfilteret i oppgaven her fungerte som tiltenkt, men ble veldig farget av de innkommende målingene. Usikkerheten i akselerometerdataene samt posisjonsdataene fører til avvikene som ble sett i resultatene til kalmanfilteret i figur 4.7 og 4.8. Dersom kalmanfilteret i oppgaven skal brukes på en skøyteløper, må dette videreutvikles. Integrasjonsmodellen kan eksempel-

5.5 Wifi signal som støttesystem



Figur 5.1: Illustrerer passeringslinjen og hvor det er størst sannsynlig å passere

vis endres til simpson-integrasjon fremfor trapesmetoden. Simpson-integrasjon er en annengradsintegrasjon som går over like store intervaller. Funksjonen erstattes med parabelbuer som går gjennom intervallens midt og endepunkt [3]. Dette kan gi et bedre integrasjonsresultat.

Figur 5.1 over viser også en oppgradering i filteret som må gjøres før en skøyte-løper kan benytte systemet. I en løpssituasjon vil det være mest sannsynlig at løperen befinner seg midt i banen ved passering. Da er det viktig at filteret får kunnskap om dette. Problemet løses ved å si at passeringen er en plass på en linje mellom to punkter og en varians må defineres slik som den gule normalfordeling i figuren viser. Dersom dette er definert, vil kalmanfilteret ha kunnskap om at det er mest sannsynlig at løperen passerer midt i banen. Dersom posisjonen for sensorintegrasjonen er på vei ut av banen, vil kalmanfilteret justere posisjonen noe mer inn mot midten av banen før det integreres videre. Det vil være en fordel å også ha alle de 9 sløyfene i isen for flest mulig korrigeringer.

5.5 Wifi signal som støttesystem

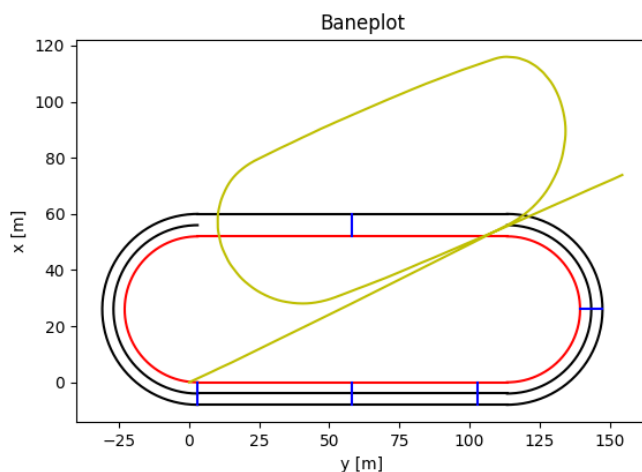
Det kan vurderes å implementere et støttesystem av wifi-sendere. Hallen har flere wifi-sendere for å gi god dekning på det trådløse nettverket i hele hallen. Med kunnskap om plasseringen til senderne kan man orientere seg i forhold til plasseringen ved bruk av triangulering. Eksempel på triangulering er vist i Figur 5.3. Dette forutsetter at sensoren eller enheten har en mottaker for wifi-signalet. Dette er en vurdering som må gjøres dersom Iphonen skal byttes ut til fordel for en mindre sensor som er lettere å ha på kroppen. Wifi-signalet og trianguleringen om senderne kan også implementeres i kalmanfilteret for å styrke

5.6 En alternativ måte som ble prøvd

estimatene der.

5.6 En alternativ måte som ble prøvd

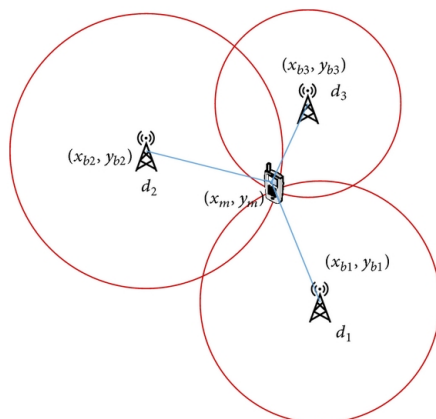
En alternativ måte som også ble kort utprøvd i oppgaven baserte seg på Apple sine egne behandlede data kalt 'Core Motion'[10]. Disse datene har tatt i bruk alle treghetssensorene for å gi det Apple kaller bevegelsesdata. Teorien var at disse dataene burde gi et fint utgangspunkt for å finne orienteringen og posisjoneringsen til telefonen. Roll, pitch og yaw var allerede regnet ut for hvert tidssteg. Disse dataene ble tatt inn i rotasjonsmatriser sammen med akselerometerdatene. De roterte datene ble dobbelintegret for å få posisjonen. Under er de plottede resultatene på banen i figur ?? vist. Dette er da uten bruk av kalmanfilter. Problemet med Core Motion dataene er at det er ukjent hva som



Figur 5.2: Baneplot av Core Motion dataene

er gjort og hva som burde gjøres for å forbedre resultatet. Dette gjorde at valget falt på å behandle rådataen selv for å ha kontroll på alt. Rådataene som ble behandlet i denne oppgaven ga et litt bedre resultat enn de ferdig behandlede dataene til Apple. For videre arbeid kan det vurderes å studere dokumentasjonen for Core Motion og se om det er mulig å hente ut fremgangsmåten for behandlingen av dataene. Det er mulig at dette kan gi bedre resultater når det blir håndtert riktig.

5.7 Alternativ sensor



Figur 5.3: Illustrerer prinsippet bak triangulering[25]

5.7 Alternativ sensor

Oppgaven baserte seg i utgangspunktet på sensoren Polar Verity Sense[15]. Sensoren brukes i hovedsak som pulsmåler, men innehar også de tre treghets-sensorene. Utfordringer knyttet til unøyaktighet og klokkesykronisering gjorde at Iphone 13 pro ble valgt som enhet for oppgaven. Polarsensoren kunne likevel vært spennende å benytte i en fremtidig studie. Polarsensoren er en liten brikke som festes på hodet eller brystet. Denne brikken inneholder også en pulsmåler og mange idrettsutøver har allerede en slik sensorbrikke. Tilgangen på dataene fra Polar Verity Sense kan hentes ut gjennom en SDK(Software Development Kit) Polar har laget for sensoren. En app kan for eksempel utvikles med SDK'en hvor løperen kan få all nødvendig informasjon på treningen. Dette ville gitt en skøyteløper et flott verktøy på veien mot raskere rundetider.

5.8 Alternative måter å estimere fart og posisjon

Radiotårn

Triangulering ved bruk av radiotårn er også en kjent måte fart og posisjon kan estimeres på. Prinsippet baserer seg på avstanden man er fra hvert tårn og ved hjelp av matematiske formler kan man regne seg ut til hvor man er. I praksis på skøytebanen ville det gjerne bli brukt 4 tårn rundt banen. Et tårn ved inngangen og et tårn ved utgangen i begge svingene. Et slik oppsett er kostbart og egner seg ikke så godt dersom man skal gå på en annen bane som

5.9 Alternativt bruksområde

ikke har disse tårnene.

5.9 Alternativt bruksområde

Et slikt utviklet system kan være interessant for flere idretter. Et håndballag vil kunne nyte godt av et slikt system. Dynamikken i spillet og hver spillers posisjon kan monitoreres. Når kampen eller treningen skal analyseres i ettertid, vil det være god informasjon å ha for å forbedre laget frem mot neste kamp. Kanskje det viser seg at en spiller er mer offensiv enn de andre og kan plasseres på banen deretter. Videre brukes allerede slike system i flyindustrien og militæret, men det kunne vært ønskelig å få teknologien tilgjengeliggjort for alle. Da er det opp til den enkelte å finne passende bruksområder for systemet. For bruk utenfor skøytehallen, forutsetter det selvfølgelig at systemet er blitt gjort uavhengig av Mylaps-systemet for posisjonskorrigeringer. Alternative andre korrigerende målinger kan være det nevnte wifi-signalet. Dette hadde sannsynligvis vært det mest optimale i en håndballhall forutsatt at det er tilgjengelig og at hallen har nok sendere.

5.10 Miljøaspekt

Dersom farts- og posisjoneringssystemet skal utvikles, vil det være en stor fordel om dette åpner for alle typer enheter og sensorer. Dette forutsetter at enhetene har treghetssensorene. Slik kan systemutviklerne bidra til å redusere forbruk. I dag finnes det flere produsenter av ulike sensorer. Det vil ikke være hensiktsmessig med tanke på miljøet å ekskludere enkelte produsenter og slik tvinge forbrukerne til å kjøpe nytt utstyr for å kunne koble seg til systemet.

Bibliografi

- [1] Utviklet med innspill fra Damiano Rotondo, Førsteamanuensis, Institutt for data- og elektroteknologi, Univeristetet i Stavanger.
- [2]
- [3]
- [4] Speedskating rink 400 meters with dimensions. https://no.m.wikipedia.org/wiki/Fil:Speedskating_rink_400_meters_with_dimensions.svg. Bilde hentet: 05.05.22.
- [5] Stepan Ayvazyan. How to draw an iphone. <https://www.drawingforall.net/how-to-draw-an-iphone/>. Bilde hentet: 25.05.22.
- [6] Alex Becker. Kalmanfilter. <https://www.kalmanfilter.net/default.aspx>. Åpnet: 14.04.22.
- [7] Mylaps B.V. Manual prochip timingsystem. <https://mylaps.my.salesforce.com/sfc/p/#D0000000oHhG/a/2p000000YjF9/T6vHDgKS55h0ebeeSBPydRaxvY0VF6rYt6gM31QAn7I>. Åpnet: 29.04.22.
- [8] Mylaps B.V. Mylaps. <https://www.mylaps.com/>. Åpnet: 22.03.22.
- [9] Jan A. Holtet. magnetometer. <https://snl.no/magnetometer>. Åpnet: 16.05.22.
- [10] Apple Inc. Core motion. <https://developer.apple.com/documentation/coremotion>. Åpnet: 16.05.22.
- [11] Ragnar Johnsen. Sampling. <https://snl.no/sampling>. Åpnet: 01.06.22.
- [12] Norvald Kjerstad. Treghetsnavigasjon. <https://snl.no/treghetsnavigasjon>. Åpnet: 22.04.22.
- [13] Anssi Kostiainen and Rijubrata Bhaumik. Magnetometer. <https://www.w3.org/TR/magnetometer/>. Åpnet: 16.05.22.

BIBLIOGRAFI

- [14] Jim Lambers. Three-dimensional coordinate systems. <https://www.math.usm.edu/lambers/mat169/fall09/lecture17.pdf>. Åpnet: 02.06.22.
- [15] Polar. Polar verity sense. https://www.polar.com/nb/products/accessories/polar-verity-sense/?gclid=Cj0KCQjwheyUBhD-ARIsAHJNM-Mfkue7t0y1R1sNwfCOJajL_12YIOT6N77TV0iIiBNEXheEsG7wqtoaAgcQEALw_wcB. Åpnet: 01.06.22.
- [16] Alvaro Lopez Revuelta. Orientation estimation and movement recognition using low cost sensors, June 2017.
- [17] Ifeyinwa E. Achumba Rinat Khusainov, Djamel Azzi and Sebastian D. Bersch. Real-time human ambulation, activity, and physiological monitoring: Taxonomy of issues, techniques, applications, challenges and limitations. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3859040/>. Åpnet: 01.06.22.
- [18] H.J. Rivertz. Nummerisk integrasjon – trapesmetoden. https://wiki.math.ntnu.no/_media/tma4100/2010h/tmc/slides/2010.10.21.pdf. Åpnet: 22.03.22.
- [19] Soumendu Sinha. Capacitive accelerometer with lateral sensing. https://www.researchgate.net/figure/Capacitive-accelerometer-with-lateral-sensing_fig2_264833557. Bilde hentet: 22.04.22.
- [20] Johannes Skaar. høyrehåndsregelen.
- [21] Ladislav Skokan. Amm-converter. <https://ammconverter.eu/>. Åpnet og nedlastet: 27.02.22.
- [22] Morten Tengesdal. Ele340 datamaskinkonstruksjon, litt om koordinattransformasjonar, 31.juli 2015.
- [23] David Titterton and John Weston. *Strapdown Inertial Navigation Technology*. The Institution of Engineering and Technology, London, UK and The American Institute of Aeronautics, Reston, Virginia, USA, 2 edition, 2004.
- [24] Ukjent. Gyroscope. <https://learn.sparkfun.com/tutorials/gyroscope/a1>. Bilde hentet: 22.04.22.
- [25] Yunzhou Zhang. A sample of triangulation positioning. https://www.researchgate.net/figure/A-sample-of-triangulation-positioning_fig2_281753273. Åpnet: 27.05.22.