

GNSS/INS-integrasjon for veiprising i urbane områder

Tore Havsø Sæstad

Elektronisk systemdesign og innovasjon

Innlevert: juni 2014

Hovedveileder: Jon Glenn Omholt Gjevestad, IET

Norges teknisk-naturvitenskapelige universitet
Institutt for elektronikk og telekommunikasjon

Sammendrag

Teknologiutvikling og stor etterspørsel etter enheter som GPS-er og smarttelefoner har redusert prisen på mikroelektromekaniske systemer som akselerometer og gyroskop og GNSS-mottakere. Q-Free har utviklet flere prototyper på bombrikker som inneholder nettopp disse sensorene. Denne oppgaven har fokusert på å utvikle algoritmer for effektiv utnyttelse av observasjoner gitt av prototyper for GNSS/INS-baserte Q-Free-brikker. Det er gjort mye forskning på GNSS/INS-integrasjon. Hovedfokuset i dette arbeidet har vært å undersøke hvilke metoder som kan brukes for å gi best mulig posisjonsestimert basert på måledataene fra forsøk utført av Q-Free.

Først ble måledataene analysert for å finne ut hvilken informasjon kunne brukes, hvor stor nøyaktighet dataene hadde og om dataene inneholdt feil. Basert på analysen ble et lineært Kalman-filter konstruert for å utføre GNSS/INS-integrasjon. Kunnskap om måledataene ble brukt for å identifisere utfordringer knyttet til posisjonering i urbane strøk og løsninger på disse ble foreslått. Kalman-filteret ble så testet i ulike situasjoner som kunne tenkes å være utfordrende. De ulike løsningene ble så evaluert.

Resultatene av måledataanalysen viste at nøyaktigheten varierer en del mellom de ulike sensorene og de ulike datasettene. En av GNSS-mottakerne hadde over en periode et gjennomsnittlig avvik på opptil 67 m i en av aksene. Gyroskopene klarte ikke å registrere høye rotasjonsrater i skarpe svinger. Dette kan sannsynligvis fikses ved å endre hvordan sensorene er konfigurert. Forsøkene viste at Kalman-filteret som ble utviklet klarer å levere bra posisjonsestimert i de fleste situasjoner.

Dette arbeidet har vist at et lineært Kalman-filter egner seg godt for GNSS/INS-integrasjon for posisjonering i urbane strøk. Med en stoppdetektor for å utføre nullhastighetsoppdateringer kan ytelsen forbedres. Resultatene viser at akselerometerne ikke klarer å holde hastigheten korrekt over lengre perioder uten GNSS-mottak. Dersom en nøyaktig hastighetsmåler implementeres i bombrikken vil systemet klare å levere nøyaktige posisjonsestimert i over 80 s.

Abstract

Development of new technology and the great demand for devices like GPS and smart phones have reduced the price of microelectromechanical systems like accelerometers and gyroscopes and GNSS receivers. Q-Free has developed several prototypes of transponders that contains just these sensors. This paper has focused on developing algorithms for efficient utilization of the observations provided by the prototypes for GNSS/INS-based Q-Free-transponders. It's done a lot of research on GNSS/INS integration. The main focus of this work was to investigate which methods can be used to provide the best possible position estimate based on measured data from experiments conducted by Q-Free.

First, the measured data was analyzed to determine what information could be used, how accurate the data was and if the data contained errors. Based on the analysis, a linear Kalman filter was designed to execute GNSS/INS integration. Knowledge of the measurement data were used to identify challenges related to positioning in urban areas and solutions were proposed. The Kalman filter was then tested in various situations that could conceivably be challenging. The various solutions were then evaluated.

The results of data analysis showed that the accuracy varies somewhat between the different sensors and the different data sets. One of GNSS receivers had over a period of time an average deviation of up to 67 m in one of the axes. Gyroscopes failed to measure high rotation rates in sharp turns. This can probably be fixed by changing how the sensors are configured. The experiments showed that the Kalman filter that was developed is able to deliver good position estimates in most situations.

This work has shown that a linear Kalman filter is suitable for GNSS/INS integration for positioning in urban areas. With a stop detector to perform zero velocity updates, the performance can be improved. The results show that the accelerometers are unable to keep the correct speed for extended periods without updates from GNSS. If an accurate speedometer would be implemented in the transponder, the system will be able to provide accurate position estimates for over 80 s.

Forord

Denne rapporten er resultatet av mitt masterarbeid ved NTNU våren 2014. Med dette avslutter jeg den fem år lange sivilingeniørutdanningen i Elektronikk ved NTNU. Jeg har alltid hatt en sterk interesse for kart, posisjonering og navigasjon. Det er bakgrunnen for at jeg har valgt å skrive min masteroppgave om nettopp dette emnet.

Jeg vil med dette takke veilederen min, Jon Glenn Omholt Gjevestad, professor II ved NTNU og professor ved NMBU, for god hjelp underveis og god veiledning når jeg har stått fast.

Trondheim, 19. juni 2014

Tore Havsø Sæstad

Innhold

Sammendrag	i
Abstract	ii
Forord	iii
Innhold	vi
Tabeller	vii
Figurer	x
Tegnforklaring	xi
1 Innledning	1
1.1 Bakgrunn	1
1.2 Problembeskrivelse og mål	2
1.3 Begrensninger	2
1.4 Rapportens oppbygging	2
2 Teori	5
2.1 Referanserammer og rotasjoner	5
2.2 INS og GNSS	7
2.3 GNSS/INS-integrasjon	8
2.4 Kalman-filteret	9
3 Metode	11
3.1 Datagrunnlag	11
3.2 Referanse-mottaker	13
3.3 Behandling av rådata	13
3.4 Design av Kalman-filter	14
3.5 Stoppdeteksjon	17

3.6	Bestemme retning fra GNSS-målt hastighet	17
3.7	Kalman-filteret i ulike situasjoner	18
3.8	Nøyaktig hastighetsmåler	19
4	Resultat	21
4.1	GNSS-mottaker	21
4.2	Gjennomgang og behandling av data	21
4.3	Stoppdeteksjon	26
4.4	Bestemme retning fra GNSS-målt hastighet	28
4.5	Kalmanfilteret i ulike situasjoner	28
4.6	Nøyaktig hastighetsmåler	32
5	Analyse	35
5.1	GNSS-mottaker	35
5.2	Analyse av måledata	35
5.3	Stoppdeteksjon	36
5.4	Kalman-filteret i ulike situasjoner	37
5.5	Nøyaktig hastighetsmåler	37
5.6	Feilkilder	37
6	Konklusjon	39
	Bibliografi	41
	Vedlegg	43
6.1	Format på GPS-rådata	43
6.2	Format på MEMS-rådata	44
6.3	Q_k	45
6.4	Gjennomgang og behandling av rådata	46
6.5	Programkode	48

Tabeller

2.1	Sammenligning av GNSS og INS.	8
4.1	Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for Rosenborg (1:1000).	21
4.2	Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for Ilsvikøra (1:1000).	22
4.3	Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for E6 (1:1000).	22
4.4	Varians for hver av gyroskopene i ulike situasjoner.	23
4.5	Varians for hver av akselerometerne i ulike situasjoner.	25
4.6	Gjennomsnittlig gyroskopmålinger i situasjoner hvor bilen står i ro.	25
4.7	Gjennomsnittlig akselerasjonsmålinger i ro med gravitasjonen samlet i hver av aksene, i ro (2:6).	26
4.8	Antall feildeteksjoner ved ulike metoder og ulike situasjoner.	26
4.9	Målestøy brukt i Kalman-filteret.	28
4.10	Prosesstøy brukt i Kalman-filteret.	29

Figurer

2.1	B-ramme.	5
2.2	E-ramme.	6
3.1	Hvordan brikke-prototypene ble plassert i bilen.	11
3.2	Bombrikke-prototype med akselerometer og gyroskop markert.	12
3.3	Eksempel på at målt posisjon fra GNSS beveger seg når bilen står i ro. IISVIKØRA (1136:1188).	18
4.1	Gyroskopmåling når bilen kjører rett fram, Rosenborg (761:765).	23
4.2	Gyroskopmåling når bilen svinger til venstre etter stopp, Rosenborg (803:807).	24
4.3	Akselerometermåling når bilen har stoppet med motoren i gang, Rosenborg (381:385).	24
4.4	Akselerometermåling når bilen svinger til venstre etter stopp, Rosenborg (803:807).	25
4.5	Resultat uten stoppdetektor, Rosenborg (371:400).	27
4.6	Resultat med stoppdetektor, Rosenborg (371:400).	27
4.7	Test av filteret med GNSS-oppdateringer hvert 5. sekund, Rosenborg (31:60).	29
4.8	Test av filteret med GNSS-oppdateringer hvert 5. sekund, Rosenborg (331:360).	30
4.9	Test av filteret med ulik GNSS-oppdateringsfrekvens, Rosenborg (31:60).	30
4.10	Test av filteret med ulik GNSS-oppdateringsfrekvens, Rosenborg (331:360).	31
4.11	Test av filteret med GNSS-oppdatering hver 5. sekund, IISVIKØRA (271:300).	31
4.12	Test av filteret med GNSS-oppdatering hver 5. sekund, IISVIKØRA (481:510).	32
4.13	Uten nøyaktig hastighetsmåler og uten GNSS-oppdateringer, Rosenborg (631:750).	33
4.14	Med nøyaktig hastighetsmåler og uten GNSS-oppdateringer, Rosenborg (631:750).	33
4.15	Avvik fra referanse ved bruk av nøyaktig fartsmåler, Rosenborg (631:750).	34
6.1	Gyroskopmåling når sensorene ligger helt stille, i ro (2:6).	46
6.2	Gyroskopmåling når bilen har stoppet med motoren i gang, Rosenborg (381:385).	47

6.3	Akselerometermåling når sensorene ligger helt stille, i ro (2:6).	47
6.4	Akselerometermåling når bilen kjører rett fram, Rosenborg (761:765). . .	48

Tegnforklaring

α	=	Roll
β	=	Pitch
γ	=	Yaw
R_b^g	=	Rotasjon fra b- til g-ramme
v_N	=	Nord-hastighet (NED)
v_E	=	Øst-hastighet (NED)
v_D	=	Ned-hastighet (NED)
λ	=	Breddegrad
ϕ	=	Lengdegrad
h	=	Høyde
X	=	X-koordinat, ECEF
Y	=	Y-koordinat, ECEF
Z	=	Z-koordinat, ECEF
a_x	=	Akselerasjon i x-aksen
a_y	=	Akselerasjon i y-aksen
a_z	=	Akselerasjon i z-aksen
ω_x	=	Rotasjonsrate rundt x-aksen
ω_y	=	Rotasjonsrate rundt y-aksen
ω_z	=	Rotasjonsrate rundt z-aksen
b_x	=	Bias, x-akse
b_y	=	Bias, y-akse
b_z	=	Bias, z-akse
w_1	=	Støy

Kapittel 1

Innledning

1.1 Bakgrunn

Veipricing er et system for innkreving av avgifter knyttet til bruk av veisystemet. Hensikten med systemet er at brukerne skal betale for kostnadene som kommer fra bruk av veiene. Slitasje, forurensing, støy, ulykker og forsinkelser er eksempler på slike kostnader. I Norge har vi bompengesystemet AutoPASS som står for innkreving av avgiftene. Dette systemet er hovedsaklig utviklet av det Trondheim-baserte teknologiselskapet Q-Free. Systemet baserer seg på bomstasjoner som står plassert langs veiene. Kjøretøy har en bombrikke plassert i frontruta slik at når en bomstasjon passerer vil dette bli registrert i systemet slik at kunden kan faktureres.

AutoPASS-systemet er basert på radiosendere og -mottakere. Bombrikkene inneholder en 5,8 GHz-radiosender som identifikator. Bomstasjonene har en tilsvarende radiomottaker som registrerer passeringen og sender informasjonen videre til et datasystem som står for fakturering. En begrensning med fysiske bomstasjoner er at det er vanskelig å få til en rettferdig plassering for alle brukerne. Eksempelvis kan en person måtte betale for å passere en bomstasjon på vei til jobb hver dag, mens naboen som bor på andre siden av bomstasjonen og jobber sammes sted, slipper å betale.

Q-Free jobber med å utvikle bombrikker som registrer hvor og hvor mye et kjøretøy har kjørt framfor å bare registrer passeringer av bomstasjoner. Dette åpner for en mer rettferdig fordeling av kostnadene på alle brukerne. Det vil for eksempel være mulig å opprette flere virtuelle bomstasjon enn dagens fysiske bomstasjoner. En annen mulighet er å ta betalt for hvor mye man bruker veien. Betaling kan da skje etter hvor langt og på hvilke veier man har kjørt.

Teknologiutvikling og en stadig voksende etterspørsel etter GPS-mottakere og små MEMS (Mikroelektromekaniske systemer) som akselerometer og gyroskop har ført til lave produksjonskostnader for slike enheter. Q-Free har utviklet flere prototyper på bombrikker som baserer seg på en GNSS-mottaker, et treakset akselerometer og et treakset gyroskop. Data fra disse sensorene i en bombrikke kan brukes for å beregne hvor et kjøretøy har vært. En utfordring med bruk av GNSS-mottakere i kjøretøy oppstår når kjøretøyet kjører

i urbane områder med høye bygninger som skjærer for signaler fra satellittene. Akselerometer og gyroskop kan brukes for å forbedre nøyaktigheten på posisjonen, samt beregne posisjonen dersom GPS-signalet faller ut.

Q-Free har utstyrt en bil med flere ulike prototyper og en high-end GPS-mottaker. De har kjørt bilen for å samle inn måledata fra ulike veisystemer i Norge.

Denne oppgaven går ut på å utvikle algoritmer for effektiv utnyttelse av observasjoner gitt av prototyper for GNSS/INS-baserte Q-Free-brikker. Arbeidet vil først og fremst rette seg mot veiprisning i urbane strøk. Det vil imidlertid være relevant for alle former for navigasjon ved hjelp av GNSS/INS-integrasjon. Mange av de samme utfordringene er aktuelle for navigasjon til fot og andre framkomstmidler som sykkel, bil, båt og fly.

Det har tidligere vært mye forskning og arbeid på området GNSS/INS-integrasjon. I blant annet fly og båt har nøyaktige treghetsnavigasjonssystemer assistert GNSS-mottakere i lang tid. Utfordringen i denne oppgaven er at sensorene er små, billige og enkle. Derfor produserer de en del støy og målingene drifter relativt mye over lengre perioder.

1.2 Problembeskrivelse og mål

Utgangspunktet for denne oppgaven har vært følgende problembeskrivelse:

Kandidaten skal utvikle algoritmer for effektiv utnyttelse av observasjoner gitt av prototyper for GNSS/INS-baserte Q-free-brikker. Algoritmene må utvikles for å generere realistiske mål for integritet og pålitelighet.

Etterhvert som arbeidet startet ble det valgt å bare fokusere på selve utviklingen av algoritmene og metodene knyttet til dette. Årsaken til dette var at oppgavens omfang ble større enn først antatt. Dette medførte at å utvikle algoritmer for å generere realistiske mål for integritet og pålitelighet ikke har blitt behandlet i dette arbeidet.

Følgende målsetninger har vært viktige i arbeidet denne oppgaven:

1. Analysere måledata fra forsøk utført av Q-Free
2. Identifisere utfordringer knyttet til GNSS/INS-integrasjon for posisjonering i urbane strøk
3. Utvikle algoritmer for posisjonering i urbane strøk
4. Foreslå metoder, teknikker og sensorer som kan forbedre ytelsen til systemet

1.3 Begrensninger

En begrensning med denne oppgaven er at måledataene bare stammer fra en bestemt bil og at måledataene kan se annerledes ut med andre kjøretøyer. Dette gjelder både med tanke på vibrasjoner i frontruta, men også hvor gode GNSS-mottaker-forhold det er i bilen.

1.4 Rapportens oppbygging

Kapittel 2 inneholder teori som er relevant for oppgaven og de matematiske regneoperasjonen som utføres. Videre i kapittel 3 beskrives metodene som er brukt i dette arbeidet.

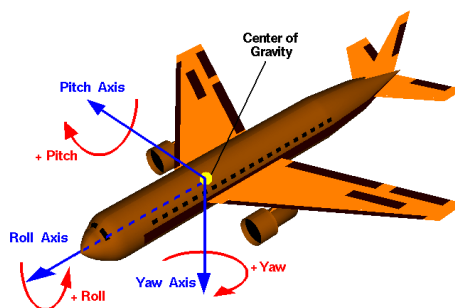
Resultatene blir presentert i kapittel 4. Disse blir analysert i kapittel 5, før det konkluderes i kapittel 6. Til slutt ligger bibliografien og alle vedleggene.

Kapittel 2

Teori

2.1 Referanserammer og rotasjoner

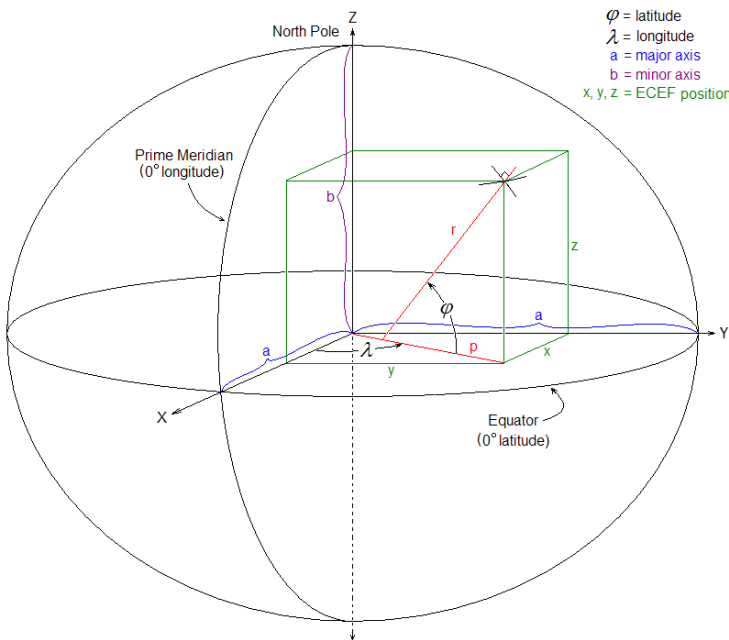
Det er vanlig å benytte seg av ulike referanserammer for å holde oversikt over ulike måleresultater og enklere kunne konvertere til ønsket format. De vanligste referanserammene er treghetsrammen (i-ramme, eng: inertial fram), sensorrammen (s-ramme, eng: sensor frame), kjøretøysrammen (b-ramme, eng: body frame), den geografiske rammen (g-ramme, eng: geographic frame) og den jordsentriske rammen (e-ramme, eng: ECEF frame).



Figur 2.1: B-ramme.

I-rammen er en referanseramme hvor Newtons bevegelseslover gjelder. Alle treghets-sensorer observerer relativt til i-rammen. S-rammen er rådataene som kommer fra f.eks. akselerometere og gyroskoper. Det er vanlig å benytte seg av denne rammen før man konverterer til b-rammen. B-rammen er knyttet til et kjøretøy. X-aksen peker i fartsretningen, y-aksen peker til høyre og z-aksen peker nedover fra kjøretøyet. Vinklene kalles ofte for roll, pitch og yaw. Se figure 2.1 for en illustrasjon. G-rammen er definert slik at x-aksen peker nordover, y-aksen peker østover og z-aksen peker nedover. E-rammen er sentrert i jordens massesenter. X-aksen peker ut mot ekvator, y-aksen peker vinkelrett på x-aksen og

z-aksen peker oppover langs jordens rotasjonsakse. Se figure 2.2 for en illustrasjon.



Figur 2.2: E-ramme.

Euler-vinkler kan brukes for å si noe om hvordan et kjøretøy er orientert. Vinklene som brukes er roll (α), pitch (β) og yaw (γ). Roll er en rotasjon rundt x-aksen til en kjøretøy og sier noe om helningsvinkelen i denne retningen. Å kjøre på en vei som heller til en av sidene vil gi utslag i roll-vinkelen. Pitch er en rotasjon rundt y-aksen til et kjøretøy og sier noe om helningsvinkelen i denne retningen. Å kjøre i en oppoverbakken vil for eksempel gi en positiv pitch-vinkel. Yaw er en rotasjon rundt z-aksen til et kjøretøy og sier noe om hvilken retning (eng: heading) kjøretøyet beveger seg. Når et kjøretøy svinger vil dette gi utslag i yaw-vinkelen.

Retningscosinusmatrisen (eng: direction cosine matrix) er en representasjon av vinklene mellom en vektor i et tre-aksen koordinatsystem. Denne matrisen kan benyttes for å utføre rotasjoner av vektorer. Innenfor navigasjon er det mest aktuelt å utføre rotasjoner rundt aksene. Matrisen får da en spesiell form. Rotasjon rundt x-aksen er gjengitt i ligning 2.1, rotasjon rundt y-aksen i ligning 2.2 og rotasjon rundt z-aksen i ligning 2.3. Rotasjonen må utføres i rekkefølgen R_z , R_y og R_x . Ligning (2.4) kan bruke for å konvertere fra g-ramme til e-ramme. Ligning (2.5) kan brukes for å konvertere fra b-ramme til g-ramme. [1]

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \quad (2.1)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \quad (2.2)$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$C_g^e = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\phi)\cos(\lambda) & -\cos(\phi)\sin(\lambda) & -\sin(\phi) \end{bmatrix} \quad (2.4)$$

$$C_b^g = R_z(\gamma)R_y(\beta)R_x(\alpha) \quad (2.5)$$

2.2 INS og GNSS

Treghetsnavigasjon (eng: INS, Inertial Navigation System) er en form for navigasjon som ofte benyttes i kjøretøy som båter, biler og fly. Teknikken går ut på å benytte seg av akselerometer- og gyroskopmålinger for å beregne hvordan kjøretøyet har beveget seg. Dette kalles bestikkregning (eng: Dead reckoning). Det vil si at man beregner posisjonen basert på forrige posisjon og kjøretøyetets hastighet og kjøreretninger. Den teknologiske utviklingen innenfor MEMS (Mikroelektromekaniske systemer) har ført til billige og små gyroskop og akselerometer med god nøyaktighet. En treghetsmåleenhet (eng: IMU, inertial measurement unit) brukes ofte i et INS. Den består for eksempel av akselerometer og gyroskop.

Treghetsnavigasjon kan fungere som en støtte til satellitnavigasjon (eng: GNSS, Global Navigasjon Satellite System). Amerikanske GPS, Russiske GLONASS og EUs kommende Galileo er eksempler på slike navigasjonssystemer. Ved å kombinere treghetsnavigasjon og satellitnavigasjon kan man oppnå flere fordeler. Kombinasjonen kan øke nøyaktigheten til systemet. Den vil gjøre systemet mindre utsatt for tap av GNSS-signal over kortere perioder. Det vil også være mulig å detektere feil på systemet dersom resultatene fra INS og GNSS er veldig ulike. Tabell 2.1, hentet fra [2], oppsummerer forskjellene på GNSS og INS.

GNSS/INS-integrasjon kan gjennomføres ved hjelp av en løs eller tett kobling. En løs kobling bruker posisjonen direkte fra GNSS og akselerasjonen og rotasjonsraten fra INS for å beregne posisjonen. En tett kobling derimot benytter seg av pseudoavstanden (eng: pseudorange) som GNSS-mottakeren beregner. En GNSS-mottaker må ha minst fire satellitter for å gi ut et posisjonsestimert. I en tett kobling kan et navigasjonssystem få informasjon fra pseudoavstandene selv om GNSS-mottakeren ikke mottar informasjon fra minst fire satellitter [1].

MEMS er utsatt for ulike typer støykilder som fører til feil observasjoner. De mest vanlige er skaleringsfeil, bias og hvit støy. Skaleringsfeil betyr at alle måleresultatene har en feil til tilsvarer sann verdi multiplisert med en konstant faktor. Bias vil si at sensoren

Tabell 2.1: Sammenligning av GNSS og INS.

GNSS	INS
- Begrenset til 1-10 Hertz	- Høy oppdateringsfrekvens tilgjengelig
- God langstidsstabilitet	- Drift oppstår over relativ korte perioder
- Stokastisk null-gjennomsnittlig støy	- Støy fra kjøretøyets dynamiske bevegelser
- Ikke bias	- Bias
- Støy	- Lite støy

måler en annen verdi enn null når den skal måle null. Feilene kan være konstante over en tidsperiode eller endre seg tilfeldig etterhvert som tiden går (eng: random walk). Bias er ofte noe som endrer seg mellom hver gang sensoren blir slått på. Akselerometermålingene blir integrert to ganger for å gi posisjon. Gyroskopmålingene blir integrert en gang for å få orientering. Dette gjør at feilene som innføres bare vil vokse etterhvert som tiden går [3].

2.3 GNSS/INS-integrasjon

Det finnes mange ulike måter å kombinere data fra GNSS og INS. Her er en oversikt over teknikker og metoder som vanligvis brukes for å utføre dette. Et treghetsnavigasjonssystem (INS) sier bare noe om hvordan et kjøretøy beveger seg. For at det skal kunne brukes til navigasjon er man avhengig av å vite noe om kjøretøyets posisjon ved start. Deretter kan man bruke informasjonen fra INS for å beregne hvordan kjøretøyet har beveget seg. Dette kalles ofte for initiering.

Man klarer sjelden å installere sensorene i et kjøretøy slik at aksene til sensorene ligger parallelt med kjøretøyets akser (b-ramme). For å få til dette må man rette opp systemet i starten. En vanlig teknikk som benyttes er å bruke målinger fra akselerometerne når man vet at bilen står i ro og er rett, det vil si horisontalt i x- og y-aksen. Da vil bare gravitasjonskraften virke på akselerometerne og man kan bruke ligning 2.6 for å måle vinklene roll og pitch. Disse vinklene kan igjen brukes for å rotere målevektorene slik at resultatet blir riktig. Ved å rette opp systemet unngår man lekkasje fra z-aksen inn i x- og y-aksen. Kjøretøy er veldig utsatt for dette fordi at gravitasjonskraften vanligvis er stor sammenlignet med andre krefter som følge av bilens bevegelser. Det sammen gjelder for gyroskopet, hvor rotasjoner rundt z-aksen (svinger) vanligvis er stor sammenlignet med rotasjon rundt x- og y-aksen.

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} = \begin{bmatrix} \arctan 2(-\bar{a}_y, -\bar{a}_z) \\ \arctan 2(\bar{a}_x, \sqrt{\bar{a}_y^2 + \bar{a}_z^2}) \end{bmatrix} \quad (2.6)$$

For å kombinere data fra GNSS og INS er det vanlig å bruke et filter av noe slag. Kalman-filteret er blant annet mye brukt. Det kan hjelpe til med å fjerne støy og gi det optimale estimatet basert på GNSS- og INS-målinger. Dersom GNSS-signalet faller ut kan INS-målinger stå for navigeringen fram til GNSS-signalet kommer tilbake.

Nullhastighetsoppdatering (eng: zero velocity update, ZUPT) er en teknikk som er mye brukt. Billige akselerometer og gyroskop vil over tid drifte en del slik at feilen bare vil bli større og større etterhvert som tiden går. En nullhastighetsoppdatering går ut på at man utfører en ny initiering når man er sikker på at bilen står i ro. På denne måten kan starte på nytt med nye verdier som man vet er riktige. Roll, pitch og hastighet er verdier som kan være aktuelle å oppdatere i slike situasjoner.

Det kan derimot være en utfordring å vite akkurat når et kjøretøy står stille. Dette gjelder spesielt dersom det er dårlige mottakerforhold for GNSS. Da vil posisjonsestimatene ofte bevege seg fram og tilbake slik at man får inntrykk av at bilen er i bevegelse. Drift i akselerometerne kan også indikere at bilen ikke står stille når den har stoppet. Hastighetsvektoren beregnet av GNSS-mottakeren brukes ofte for å beregne i hvilken retning kjøretøyet beveger seg i. Det kan være problematisk å benytte seg av denne teknikken når kjøretøyet hastighet er veldig lav eller den står helt i ro. Da vil den faktiske hastighetsvektoren være veldig liten eller null sammenlignet med støyen i GNSS-mottakeren. En konsekvens av dette er at retningen som beregnes kan bli helt feil. Det er ikke uvanlig at beregnet retning er motsatt av faktisk retning når kjøretøyet står i ro [1].

De viktigste ligningene som brukes i et INS er listet opp i ligning (2.7). For en fullstendig oversikt og forklaring henvises det til [1].

$$\omega_{eb}^b = \omega_{ib}^b - C_b^e \omega_{ie}^e \quad (2.7a)$$

$$\Omega_{eb}^b = [\omega_{eb}^b \times] \quad (2.7b)$$

$$\dot{C}_b^e = C_b^e \Omega_{eb}^b \quad (2.7c)$$

$$\hat{C}_b^e = C_b^e + \dot{C}_b^e \Delta t \quad (2.7d)$$

$$\Omega_{ie}^e = [\omega_{ie}^e \times] \quad (2.7e)$$

$$a_{eb}^e = C_b^e f_{ib}^b - 2\Omega_{ie}^e v^e + C_g^e g_e^g \quad (2.7f)$$

$$v_{eb}^e = \hat{C}_b^e v_{eb}^b + a_{eb}^e \Delta t \quad (2.7g)$$

Det finnes flere forskningsartikler om dette fagområdet. [4], [5], [6], [7], [8] og [9] omhandler alle posisjonering ved hjelp av INS og eksterne sensorer som for eksempel GNSS- og WLAN-mottakere.

2.4 Kalman-filteret

Kalman-filteret er et sett med matematiske ligninger utviklet av Rudolf E. Kalman. Filteret er en rekursiv løsning for filtrering av lineære, diskrete data ved å minimere gjennomsnittet av kvadrert feil. Kort oppsummert består ligningene av en tidsoppdatering og en måleoppdatering. Ligningene er listet opp i 2.8. De to første ligningene står for tidsoppdateringen, mens de tre siste står for måleoppdateringen. For en mer detaljert gjennomgang refereres det til [10]. \tilde{x}_k er et a priori estimat av tilstanden som beregnes ut fra systemmodellen. \tilde{P}_k er et a priori estimat av feilkovariansen. K_k blir ofte kalt for Kalman-faktoren eller Kalman-gain og beregnes for å minimere a posteriori feilkovarians. \hat{x}_k er et a posteriori estimat av tilstanden basert på en vektning mellom a priori tilstandsestimat og målingene z_k . \hat{P}_k er et a posteriori estimat av feilkovariansen.

$$\tilde{x}_k = \phi_{k-1} \hat{x}_{k-1} \tag{2.8a}$$

$$\tilde{P}_k = \phi_{k-1} \hat{P}_{k-1} \phi_{k-1}^T + Q_{k-1} \tag{2.8b}$$

$$K_k = \tilde{P}_k H_k^T (H_k \tilde{P}_k H_k^T + R_k)^{-1} \tag{2.8c}$$

$$\hat{x}_k = \tilde{x}_k + K_k (z_k - H_k \tilde{x}_k) \tag{2.8d}$$

$$\hat{P}_k = (I - K_k H_k) \tilde{P}_k \tag{2.8e}$$

Kapittel 3

Metode

3.1 Datagrunnlag

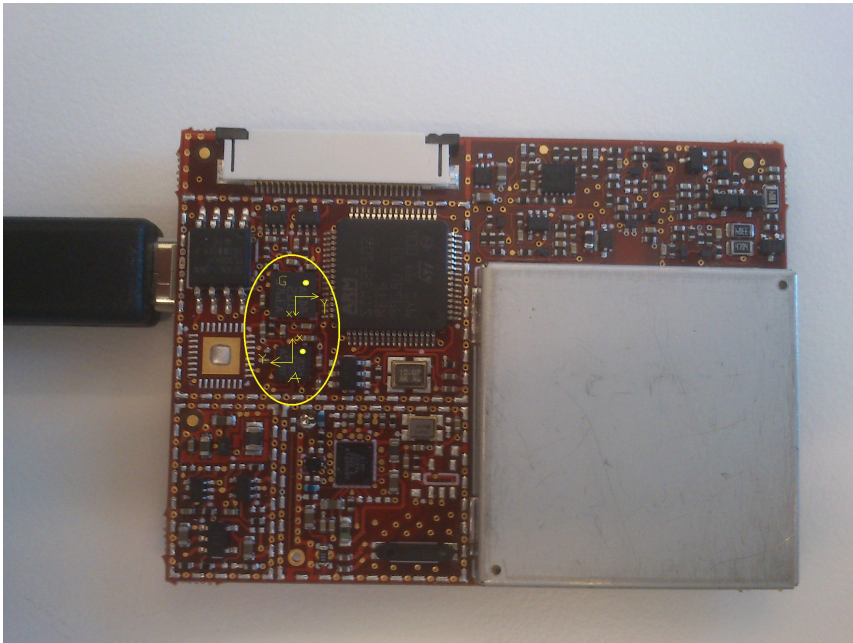
Måledataene som denne oppgaven tar utgangspunkt i stammer fra forsøk utført av Q-Free. En varebil ble utstyrt med ulike prototyper på bombrikker og en high-end GNSS-mottaker med INS. Prototypene ble plassert i frontvinduet og tilkoblet dataloggere som lagret rådataene som ble generert underveis (Se figur 3.1). High-end GPS-mottakeren ble plassert i lasterommet midt i bilen.



Figur 3.1: Hvordan brikke-prototypene ble plassert i bilen.

Prototypene består av en GNSS-mottaker, et treakset gyroskop og et treakset akselerometer. Forskjellen på prototypene er at de inneholder ulike GNSS-mottakere. Alle inneholder samme type akselerometer og gyroskop. Se figur 3.2 for et bilde av en prototype med sensorene markert. Gyroskopet som benyttes er av typen L3G4200D (MEMS motion

sensor: ultra-stable three-axis digital output gyroscope) [11]. Det måler rotasjon rundt tre akser som står ortogonalt på hverandre. Rotasjonene måles i milligrader per sekund. Akselerometeret som benyttes av er typen LIS331DLH (MEMS digital output motion sensor ultra low-power high performance 3-axes nano”accelerometer) [12]. Det måler akselerasjon i tre akser som står ortogonalt på hverandre. Akselerasjonen måles i milli-g. GNSS-mottakerne gir blant annet posisjon, hastighet i nord-, øst- og ned-retning, kurs og flere parametere. En fullstendig oversikt over hva som lagres fra de ulike sensorene finne i vedlegg 6.1 og 6.2.



Figur 3.2: Bombrikke-prototype med akselerometer og gyroskop markert.

Måledata fra følgende kjøreturer ble brukt i denne oppgaven:

1. Rosenborg: Kjøring mellom Rosenborg og NTNU Gløshaugen i Trondheim 26. august 2013
2. Ilsvikøra: Kjøring i Ilsvikøra i Trondheim 11. februar 2014
3. E6: Kjøring på E6 fra Trondheim til Oslo 4. mars 2014

Rosenborg-datasettet består av kjøring i åpne områder med gode forhold for satellittmottak. På enkelte strekninger er det høye hus langs veien. Datasettet inneholder blant annet hastigheter fra slake og krappe svinger, kjøring i rundkjøringer og stopp på rødt lys. Ilsvikøra-datasettet består av kjøring i et område med høye bygninger som gir dårlige forhold for satellittmottak. Datasettet inneholder sakte kjøring, stopp og krappe svinger. E6-datasettet består av kjøring i vekslende forhold med høye hastigheter, tunneler, bykjøring og stopp underveis.

3.2 Referanse-mottaker

Referanse-mottakeren som er brukt i bilen er av typen NovAtel Span. Nøyaktigheten på denne er på centimeter-nivå. I forsøkene blir dataene fra referanse-mottakeren sett på som den sanne verdien av målingene. Alle resultater sammenlignet mot referansen for å si noe om nøyaktigheten på målingene eller beregningene.

3.3 Behandling av rådata

For å enkelt kunne sammenligne de ulike datasettene og for å fjerne usikkerheten knyttet til hvilken GNSS-mottaker som var i bruk, ble en av mottakerne brukt til alle beregningene. Prototype nummer 1, 2 og 9 var med i alle tre datasettene. Derfor ble disse mottakerne vurdert opp mot referansen (nummer 6). En oversikt over hvilken informasjon som er tilgjengelig fra GNSS-mottakeren kan ses i vedlegg 6.1). Følgende oversikt forklarer hvilken GNSS-informasjon som ble ansett som relevant for oppgaven:

- Måletidspunkt (8) for synkronisering mellom GNSS-mottaker og IMU
- Breddegrad (11), lengdegrad (12) og høyde (14) for å måle posisjon
- Nord-hastighet (27), øst-hastighet (28) og ned-hastighet (29) for å enklere måle hvilken retning bilen beveger seg i

For å finne ut hvilken GPS-mottaker som er mest nøyaktig, ble gjennomsnittlig avvik fra referansen for hver av målingene beregnet for hver av datasettene. De 1000 første observasjonen i hver av datasettene ble sammenlignet med referansen.

En oversikt over hvilken informasjon som er tilgjengelig fra IMU-en kan ses i vedlegg 6.2). Følgende oversikt forklarer hvilken IMU-informasjon som ble ansett som relevant for oppgaven:

- Måletidspunkt (8) for synkronisering mellom GNSS-mottaker og IMU
- Akselerasjon i x-, y- og z-aksen (9-11)
- Rotasjonsrate rundt x-, y- og z-aksen (12-14)

Deretter ble måledata fra gyroskopene og akselerometerne studert for å finne ut hvor god kvalitet det er på målingene. Her ble det tatt utgangspunkt i Rosenborg-datasettet. Slik ble gyroskopene undersøkt:

- Grafisk studere gyroskopene når bilen er i ro, har stoppet, kjører rett fram og svinger skarpt, for å se etter feil
- Sammenligne varians for gyroskopene når bilen er i ro, har stopp og kjører rett fram, for å studere støyen fra vibrasjonene i bilen
- Måle gjennomsnittet ved flere stopp for å se etter bias og drift i gyroskopene

Slik ble akselerometerne undersøkt:

- Grafisk studere akselerometerne når bilen er i ro, har stoppet, kjører rett fram og svinger skarpt, for å se etter feil
- Sammenligne varians for akselerometerne når bilen er i ro, her stoppet og kjører rett fram, for å studere vibrasjonene i bilen
- Måling ved å samle g-kraften i hver av aksene når brikken ligger stille og sammenligne for å se etter bias og skaleringsfeil

Bombrikken var ikke perfekt horisontert i bilen under forsøkene. Derfor måtte måleresultatene rettes opp. Dette ble gjort som beskrevet i seksjon 2.3. Gyroskopmålingene ble også konvertert fra milligrader/s til radianer/s og akselerometermålingene ble konvertert fra milli-g til akselerasjon i $\frac{m}{s^2}$. I tillegg ble målingene rotert som beskrevet i seksjon 2.3. I tillegg måtte akselerometermålingene roteres 180° om z-aksen for at observasjonsaksene skulle falle sammen med b-rammen til bilen.

For å henvise til en posisjon i datasette brukes formatet *datasett (start:slutt)*. Eksempel: Rosenborg (31:60) betyr at det henvises til GNSS-måling nr. 31 til 60 i Rosenborg-datasettet.

Måldataene er ikke synkronisert. IMU-en leverer oppdateringer med en fast frekvens på 50 Hz. GNSS-mottakeren leverer oppdateringer omtrent hvert sekund. Måldataene har et tidsstempel som ble brukt for å synkronisere dataene. GNSS-målingene kommer litt forsinket sammenlignet med IMU-målingene. For å kompensere for dette ble IMU-målingene forsinket med 1,24 s (tilsvarende 62 IMU-oppdateringer).

3.4 Design av Kalman-filter

Kalman-filteret ble konstruert som en PV-modell. En PV-modell består av posisjon og hastighet. Dette gir et enkelt og lineært filter som egner seg godt til dette formålet. En løs kobling mellom GNSS og INS ble valgt for å unngå ikke-lineære effekter som ville krevd et komplekst filter. Tilstandsvariablene er listet opp i (3.1). Tilstanden består av posisjon i ECEF-koordinater, hastighet i ECEF-koordinater, euler-vinklene, rotasjonsrate og gyro-bias. Kalman-filteret er todelt. De seks første tilstandsvariablene er knyttet til estimering av posisjon og hastighet. De ni siste tilstandsvariablene er knyttet til estimering av roll, pitch og yaw. Kalman-filteret klarer selv å estimere bias for rotasjonsratene. Derfor er biasene tatt med som egne variable.

$$x_k = \begin{bmatrix} X & Y & Z & v_x & v_y & v_z & \alpha & \beta & \gamma & \omega_x & \omega_y & \omega_z & b_x & b_y & b_z \end{bmatrix}_k^T \quad (3.1)$$

$$X_k = X_{k-1} + v_{x,k-1}\Delta t + w_1 \quad (3.2a)$$

$$Y_k = Y_{k-1} + v_{y,k-1}\Delta t + w_1 \quad (3.2b)$$

$$Z_k = Z_{k-1} + v_{z,k-1}\Delta t + w_1 \quad (3.2c)$$

$$v_{x,k} = v_{x,k-1} + w_2 \quad (3.2d)$$

$$v_{y,k} = v_{y,k-1} + w_2 \quad (3.2e)$$

$$v_{z,k} = v_{z,k-1} + w_2 \quad (3.2f)$$

$$\alpha_k = \alpha_{k-1} + \omega_{x,k-1}\Delta t - b_x\Delta t + w_3 \quad (3.2g)$$

$$\beta_k = \beta_{k-1} + \omega_{y,k-1}\Delta t - b_y\Delta t + w_3 \quad (3.2h)$$

$$\gamma_k = \gamma_{k-1} + \omega_{z,k-1}\Delta t - b_z\Delta t + w_3 \quad (3.2i)$$

$$\omega_{x,k} = \omega_{x,k-1} + w_4 \quad (3.2j)$$

$$\omega_{y,k} = \omega_{y,k-1} + w_4 \quad (3.2k)$$

$$\omega_{z,k} = \omega_{z,k-1} + w_4 \quad (3.2l)$$

$$b_{x,k} = b_{x,k-1} + w_5 \quad (3.2m)$$

$$b_{y,k} = b_{y,k-1} + w_5 \quad (3.2n)$$

$$b_{z,k} = b_{z,k-1} + w_5 \quad (3.2o)$$

PV-modellen kan beskrives med ligningene i (3.2). Transisjonsmatrisen (3.3) tar utgangspunkt i disse ligningene. w_1 , w_2 , w_3 , w_4 og w_5 er prosessstøyen. Det ble antatt at disse er uavhengige av hverandre, er hvit og har en normal sannsynlighetsfordeling rundt null.

$$\phi = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & -dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & dt & 0 & 0 & -dt & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

GNSS-mottakeren (ca. 1 Hz) og INS-sensorene (50 Hz) har ulik oppdateringsfrekvens. Dette medfører at Kalman-filteret må utføre to forskjellige måleoppdateringer avhengig

av hvilke sensorer som gir oppdatering. Når en GNSS-oppdatering er tilgjengelig brukes målevektor (3.4). Målevektoren (3.5) brukes når en oppdatering er tilgjengelig fra INS.

$$z_1 = \begin{bmatrix} X_G & Y_G & Z_G & v_{x,G} & v_{y,G} & v_{z,G} & v_{x,I} & v_{y,I} & v_{z,I} & \alpha & \beta & \gamma & \omega_x & \omega_y & \omega_z \end{bmatrix} \quad (3.4)$$

$$z_2 = \begin{bmatrix} v_{x,I} & v_{y,I} & v_{z,I} & \alpha & \beta & \omega_x & \omega_y & \omega_z \end{bmatrix} \quad (3.5)$$

X_G , Y_G og Z_G er posisjonen målt av GNSS-mottakeren uttrykt i ECEF-koordinater. Ligning (2.4) brukes for å konvertere målingene fra g-ramme til e-ramme. Det samme gjelder for $v_{x,G}$, $v_{y,G}$ og $v_{z,G}$, bare at dette er hastighetene uttrykt i ECEF-koordinater gitt av GNSS-mottakeren. $v_{x,I}$, $v_{y,I}$ og $v_{z,I}$ er beregnet hastighet basert på målinger gjort av IMU-en. Ligningene (2.7) blir brukt for å beregne disse hastighetene. α og β blir beregnet ved å sette inn akselerometermålingene i formlene (2.6). γ blir beregnet ut fra hastighetsvektoren gitt av GNSS-mottakeren med ligningen $\hat{\gamma} = \arctan 2(v_E, v_N)$. ω_x , ω_y og ω_z er målinger av rotasjonsraten ved hjelp av gyroskopene.

Designmatrisene H_1 og H_2 er gitt i (3.6) og (3.7).

$$H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

$$H_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.7)$$

Prosesstøymatrisen (6.1) i vedlegg 6.3 ble beregnet med utgangspunkt i (3.8). Vedlegg 6.5 inneholder MATLAB-koden som ble brukt til beregningene.

$$Q_k = \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \eta) G(\eta) G(\eta)^T \phi(t_k, \eta)^T d\eta \quad (3.8)$$

Målestøymatrisene (3.9) og (3.10) er diagonalmatriser. Det ble antatt at kovariansen mellom de ulike sensorene var null.

$$R_1 = \text{diag}[\sigma_{X,G}^2, \sigma_{Y,G}^2, \sigma_{Z,G}^2, \sigma_{v_x,G}^2, \sigma_{v_y,G}^2, \sigma_{v_z,G}^2, \sigma_{v_x,I}^2, \sigma_{v_y,I}^2, \sigma_{v_z,I}^2, \sigma_{\alpha,I}^2, \sigma_{\beta,I}^2, \sigma_{\gamma,G}^2, \sigma_{\omega_x,I}^2, \sigma_{\omega_y,I}^2, \sigma_{\omega_z,I}^2] \quad (3.9)$$

$$R_2 = \text{diag}[\sigma_{v_x,I}^2, \sigma_{v_y,I}^2, \sigma_{v_z,I}^2, \sigma_{\alpha,I}^2, \sigma_{\beta,I}^2, \sigma_{\omega_x,I}^2, \sigma_{\omega_y,I}^2, \sigma_{\omega_z,I}^2] \quad (3.10)$$

3.5 Stoppdeteksjon

En utfordring ved GNSS/INS-integrasjon er å vite når bilen står i ro og når den beveger seg. Gyroskopene kan ikke si noe om bilens hastighet, bare om den endrer retning. Akselerometerne kan integrere målt akselerasjon til hastighet, men de kan ikke si nøyaktig hva hastigheten er. En bil med en akselerasjon målt til 0 m/s^2 kan bevege seg med en konstant hastighet eller stå i ro. GNSS-mottakerne gir en god indikasjon på om bilen står i ro eller ikke, men på grunn av at den stadig kommer med nye oppdateringer av posisjonen, vil det se ut som om bilen beveger seg når den står i ro. Se figur 3.3 for et eksempel. Det er også viktig å vite når bilen står i ro dersom en skal utføre nullhastighetsoppdateringer (ZUPT).

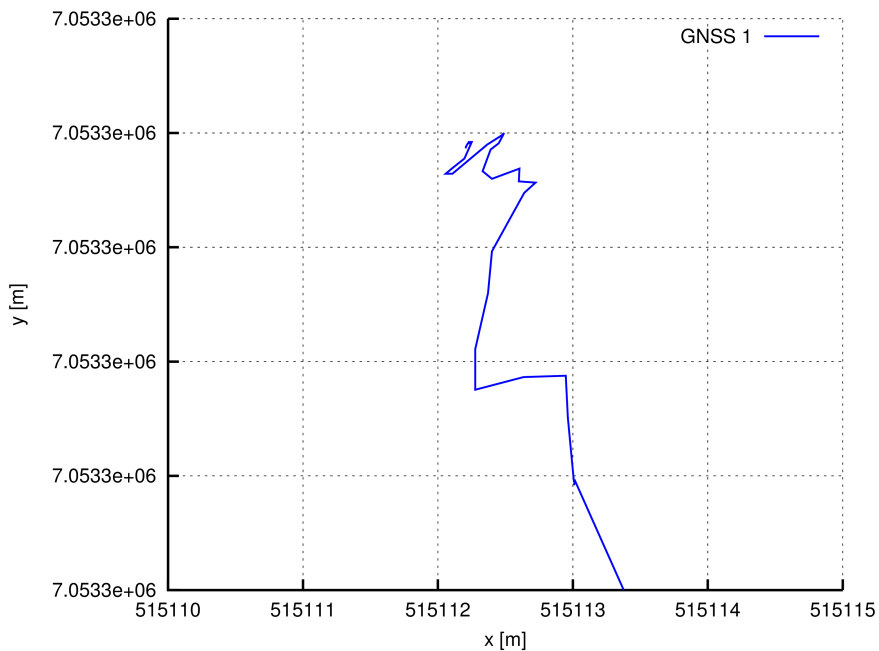
Her undersøkes det om akselerometerne likevel kan brukes for å finne ut om bilen står i ro. Siden akselerometerne registrerer vibrasjoner fra bilen som kommer fra motoren og bilens bevegelser, er tanken at mengden vibrasjoner kan indikere at bilen står i ro.

Det ble gjennomført simuleringer for å undersøke om variansen til signalet fra akselerometerne kunne brukes for å avgjøre om bilen stod i ro eller var i bevegelse. Målingene av variansen til akselerometerne i ulike situasjoner danner grunnlaget for valg av grenseverdier som avgjorde om bilen var i bevegelse eller ikke. Variansen til akselerometerne ble beregnet for tidsperioden 1,0 s for å enklere kunne sammenligne med referansemottakerens målte hastighet som kommer med en gang hvert sekund. Disse verdiene ble så sammenlignet med valgt grenseverdi for å si noe om bevegelsen. Testen ble utført over følgende perioder: Rosenberg (301:400), Rosenberg (1001:1100) og Rosenberg (1401:1500).

Deretter ble en funksjon for å sjekk om bilen står i ro implementert i filteret. Dersom bilen står i ro blir hastigheten satt til 0 m/s og roll og pitch ble målt ved hjelp av akselerometeret. Rosenberg (371:400) ble valgt for å test stoppdetektoren.

3.6 Bestemme retning fra GNSS-målt hastighet

En GNSS-mottaker kan beregne hastighetsvektoren for bevegelsen i g-rammen. Ligningen $\hat{\gamma} = \arctan 2(v_E, v_N)$ kan brukes for å estimere hvilken retning mottakeren og kjøretøyet



Figur 3.3: Eksempel på at målt posisjon fra GNSS beveger seg når bilen står i ro. Iilsvikøra (1136:1188).

beveger seg. GNSS-mottakeren vil gi en del støy på hastighetsvektoren. Ved lave hastighe-
ten eller når bilen står i ro vil denne støyen stå for en betydelig del av hastighetsvektoren.
Dette vil føre til at retningen som beregnes kan gi helt feil resultater. Det ble undersøkt
om dette var et problem. En metode for å unngå problemet ville vært å ikke utføre ret-
ningsmålinger når hastigheten er liten.

3.7 Kalman-filteret i ulike situasjoner

For å finne ut hvor godt Kalman-filteret fungerer ble det testet i ulike situasjoner. Følgende
situasjoner ble valgt for å test hvor godt systemet klarer seg med GNSS-mottak for hver
5 s, 10 s, 15 s og 30 s. Dette gir også en god indikasjon på hvor bra systemet klarer seg uten
GNS-oppdateringer.

1. Vanlig kjøring framover, Rosenborg (31:60)
2. Vanlig kjøring i høyresving, Rosenborg (331:360)

Filteret ble deretter testet i følgende to situasjoner for å sjekk hvor godt det fungerer
når målingene fra GNSS-mottakeren er dårlig. Det mottas oppdatering fra GNSS hver 5 s.

1. Sakte kjøring blant høye bygninger, Iilsvikøra (271:300)

2. Sakte kjøring med skarp sving blant høye bygninger, Ilsvikøra (480:510)

Alle testene startet med posisjon og hastighet gitt av GNSS-mottakeren sammen med korrekt yaw og bias.

3.8 Nøyaktig hastighetsmåler

Tidligere tester har vist at akselerometerne er unøyaktige og påvirkes i stor grad av støy. For å undersøke i hvor stor grad dette er tilfelle, ble filteret brukt på Rosenborg (631:750). Testen ble utført uten oppdateringer fra GNSS-mottakeren. Det vil si at bare informasjon fra IMU-en ble brukt for å beregne posisjonen. Referansen ble brukt for å gi nøyaktig hastighet til filteret en gang i sekundet.

Kapittel 4

Resultat

4.1 GNSS-mottaker

GNSS-mottaker 1, 2 og 9 ble sammenlignet med referansemottakeren for å se etter store avvik og måle nøyaktigheten. Gjennomsnittlig avvik mellom GNSS-mottakerne og referanse ble beregnet for måling av breddegrad, lengdegrad, høyde, nord-hastighet, øst-hastighet og ned-hastighet. Testen ble utført for de første 1000 målingene i datasettene Rosenberg, Ilsvikøra og E6. Se tabell 4.1, 4.2 og 4.3 for resultater. Datasettet E6 inneholder ikke hastighetsmålinger.

Tabell 4.1: Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for Rosenberg (1:1000).

	GNSS-mottaker 1	GNSS-mottaker 2	GNSS-mottaker 9
Breddegrad	25,58 m	3,61 m	10,05 m
Lengdegrad	8,57 m	2,32 m	9,15 m
Høyde	67,19 m	4,08 m	5,76 m
Nord-hastighet	0,83 m/s	0,31 m/s	0,88 m/s
Øst-hastighet	0,41 m/s	0,23 m/s	0,84 m/s
Ned-hastighet	1,83 m/s	0,19 m/s	0,19 m/s

4.2 Gjennomgang og behandling av data

Første ble dataene fra gyroskopene gjennomgått. Figur 4.1 viser utslagene på gyroskopene som kommer fra vibrasjoner og små bevegelser når bilen kjører rett fram. Figur 4.2 viser utslaget på gyroskopene når bilen akselererer etter et lyskryss og svinger skarpt til venstre.

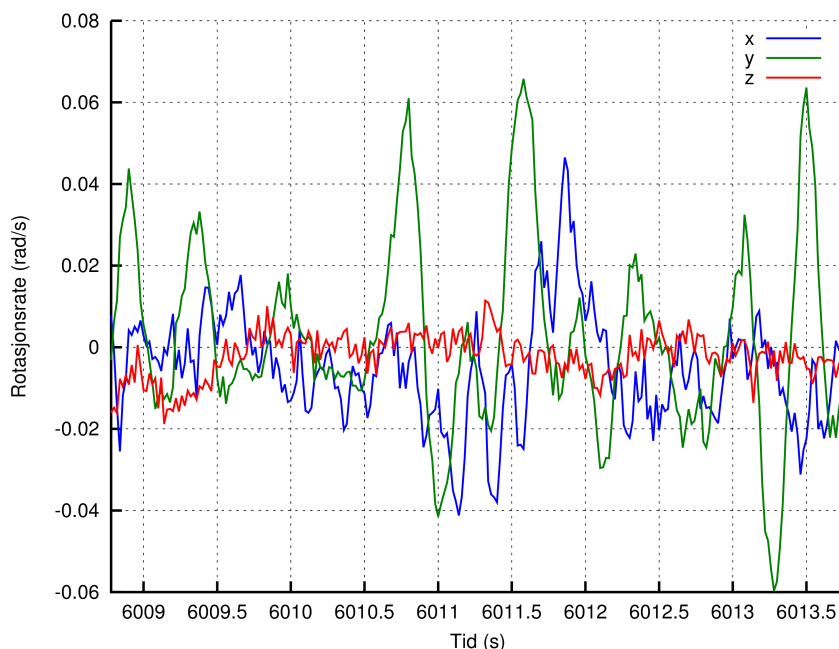
Tabell 4.2: Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for Ilsvikøra (1:1000).

	GNSS-mottaker 1	GNSS-mottaker 2	GNSS-mottaker 9
Breddegrad	5,15 m	5,50 m	5,54 m
Lengdegrad	5,20 m	4,99 m	4,92 m
Høyde	9,22 m	12,72 m	10,53 m
Nord-hastighet	0,75 m/s	0,76 m/s	0,74 m/s
Øst-hastighet	0,71 m/s	0,72 m/s	0,70 m/s
Ned-hastighet	0,05 m/s	0,05 m/s	0,05 m/s

Tabell 4.3: Gjennomsnittlig avvik fra referanse for GNSS-mottaker 1, 2 og 9 for E6 (1:1000).

	GNSS-mottaker 1	GNSS-mottaker 2	GNSS-mottaker 9
Breddegrad	1,69 m	2,03 m	1,71 m
Lengdegrad	1,86 m	2,14 m	1,73 m
Høyde	10,90 m	3,67 m	8,99 m
Nord-hastighet	–	–	–
Øst-hastighet	–	–	–
Ned-hastighet	–	–	–

Figur 6.1 og 6.2 er plassert i vedlegg 6.4. Disse viser når brikken ligger stille på et bord og når bilen står stille med motoren i gang.



Figur 4.1: Gyroskopmåling når bilen kjører rett fram, Rosenberg (761:765).

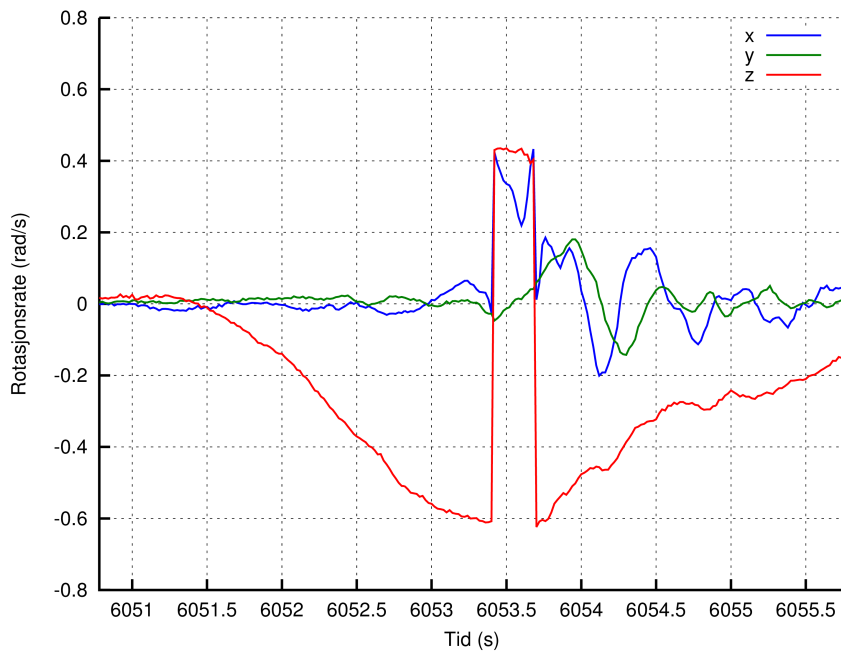
Deretter ble dataene fra akselerometerne gjennomgått. Figur 4.3 viser utslagene på akselerometerne som kommer fra vibrasjoner fram motoren når bilen står stille på rødt lys. Figur 4.4 viser utslaget på akselerometerne når bilen akselererer etter et lyskryss og svinger skarpt til venstre. Figur 6.3 og 6.4 er plassert i vedlegg 6.4. Disse viser når brikken ligger stille på et bord og når bilen står stille med motoren i gang.

Tabell 4.5 og 4.4 inneholder variansen for hver av aksene til hver av sensorene i ulike situasjoner.

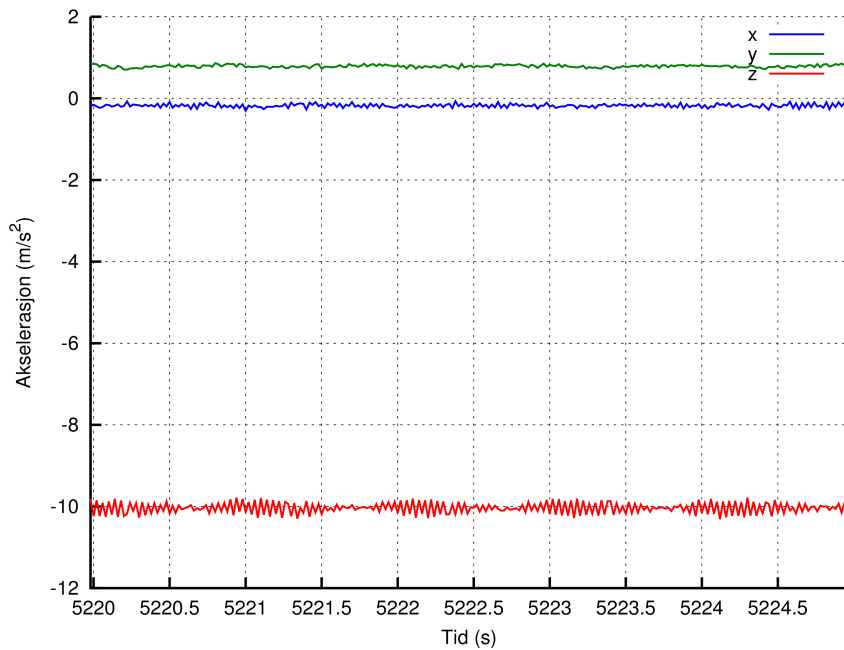
Tabell 4.4: Varians for hver av gyroskopene i ulike situasjoner.

	Gyroskop x	Gyroskop y	Gyroskop z
I ro	$7,02 \times 10^{-6}$ rad/s	$5,14 \times 10^{-6}$ rad/s	$7,36 \times 10^{-6}$ rad/s
Stopp	$5,65 \times 10^{-6}$ rad/s	$4,38 \times 10^{-6}$ rad/s	$7,22 \times 10^{-6}$ rad/s
Kjøring rett fram	$1,829 \times 10^{-4}$ rad/s	$5,503 \times 10^{-4}$ rad/s	$2,908 \times 10^{-5}$ rad/s

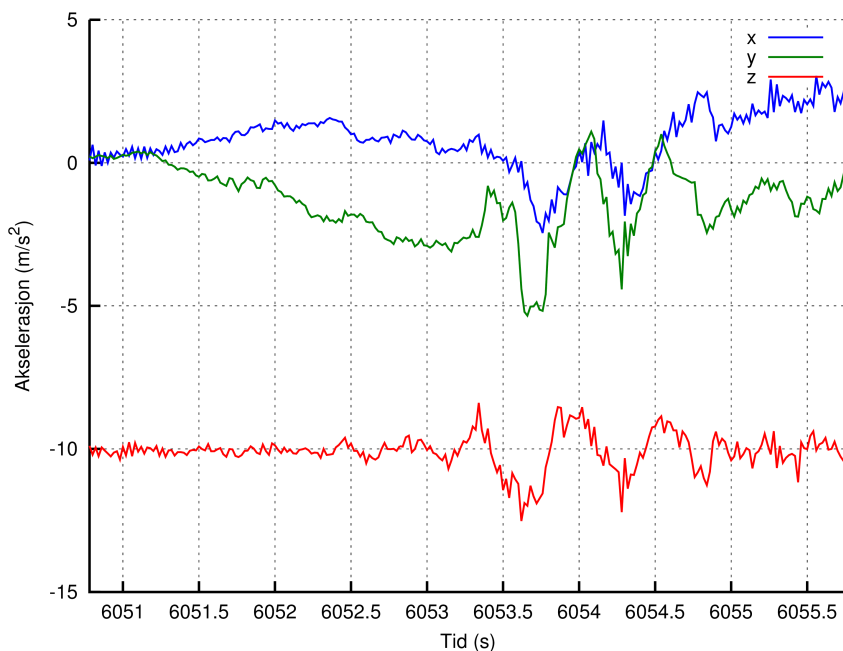
For å se etter bias i gyroskopene ble gjennomsnittet av gyroskopene beregnet i situasjoner hvor bilen står i ro. Resultatet av målingene kan leses i tabell 4.6.



Figur 4.2: Gyroskopmåling når bilen svinger til venstre etter stopp, Rosenberg (803:807).



Figur 4.3: Akselerometermåling når bilen har stoppet med motoren i gang, Rosenberg (381:385).



Figur 4.4: Akselerometermåling når bilen svinger til venstre etter stopp, Rosenberg (803:807).

Tabell 4.5: Varians for hver av akselerometerne i ulike situasjoner.

	Akselerometer x	Akselerometer y	Akselerometer z
I ro	$1,877 \times 10^{-4} \text{ m/s}^2$	$2,189 \times 10^{-4} \text{ m/s}^2$	$3,517 \times 10^{-4} \text{ m/s}^2$
Stopp	$1,986 \times 10^{-3} \text{ m/s}^2$	$1,090 \times 10^{-3} \text{ m/s}^2$	$1,930 \times 10^{-2} \text{ m/s}^2$
Kjøring rett fram	$9,949 \times 10^{-2} \text{ m/s}^2$	$4,694 \times 10^{-2} \text{ m/s}^2$	$2,0443 \times 10^{-1} \text{ m/s}^2$

Tabell 4.6: Gjennomsnittlig gyroskopmålinger i situasjoner hvor bilen står i ro.

	Gyroskop x	Gyroskop y	Gyroskop z
Rosenborg (381:385)	-0,004 23 rad/s	0,003 31 rad/s	-0,003 82 rad/s
Rosenborg (1001:1005)	-0,003 94 rad/s	0,003 19 rad/s	-0,003 24 rad/s
Rosenborg (2701:2705)	-0,001 61 rad/s	0,003 15 rad/s	-0,001 76 rad/s
Rosenborg (3061:3065)	-0,001 73 rad/s	0,003 14 rad/s	-0,001 70 rad/s

Tabell 4.7 inneholder målinger av akselerasjonen når brikken ligger i ro slik at gravitasjonskraften bare virker i en av aksene.

Tabell 4.7: Gjennomsnittlig akselerasjonsmålinger i ro med gravitasjonen samlet i hver av aksene, i ro (2:6).

	Akselerasjon
$x=-1g$	$-10.005m/s^2$
$x=1g$	$10.047m/s^2$
$y=1g$	$10.204m/s^2$
$y=-1g$	$-9.9949m/s^2$
$z=1g$	$-10.197m/s^2$
$z=-1g$	$10.163m/s^2$

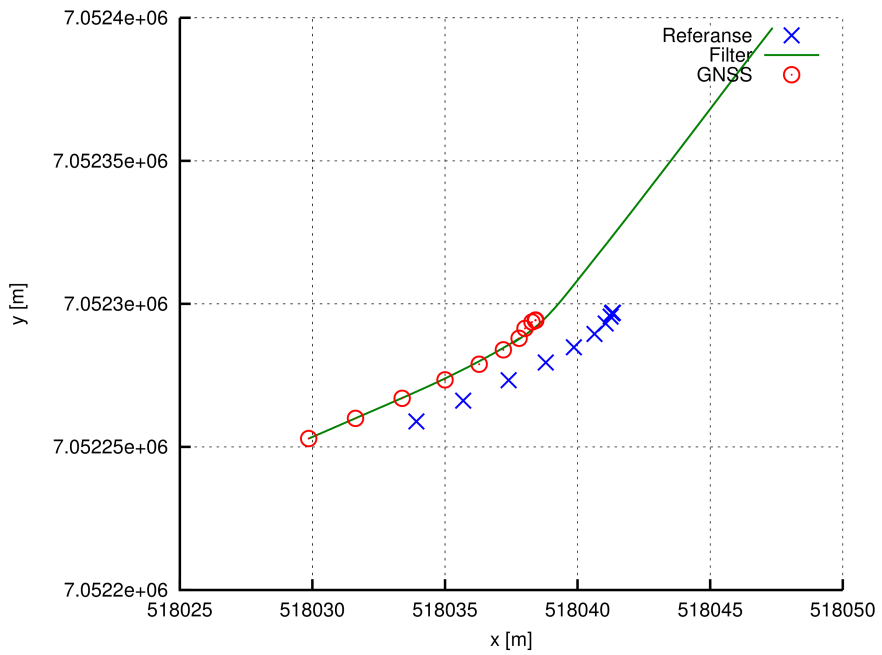
4.3 Stoppdeteksjon

Tabell 4.5 inneholder variansen for akselerometerne i ulike situasjoner. Grenseverdiene for målt varians som avgjør om bilen er i bevegelse eller ikke ble valgt ut fra disse målingene. Forholdet mellom variansen når bilen er i bevegelse og variansen når bilen står i ro er 50.1 i x-aksen, 43.1 i y-aksen og 10.6 i z-aksen. På bakgrunn av disse resultatene ble det besluttet å bare se på variansen i x- og y-aksen. Grenseverdien for akselerometer x ble satt til $0,016 m/s^2$, for akselerometer y $0,008 m/s^2$ og for akselerometer x og y $0,048 m/s^2$. Antall feildeteksjoner ble regnet ut ved å sammenligne mot en grense på $0,2 m/s$ målt av referansemottakeren. Se tabell 4.8 for antall feildeteksjoner med ulike metoder i de ulike situasjoner.

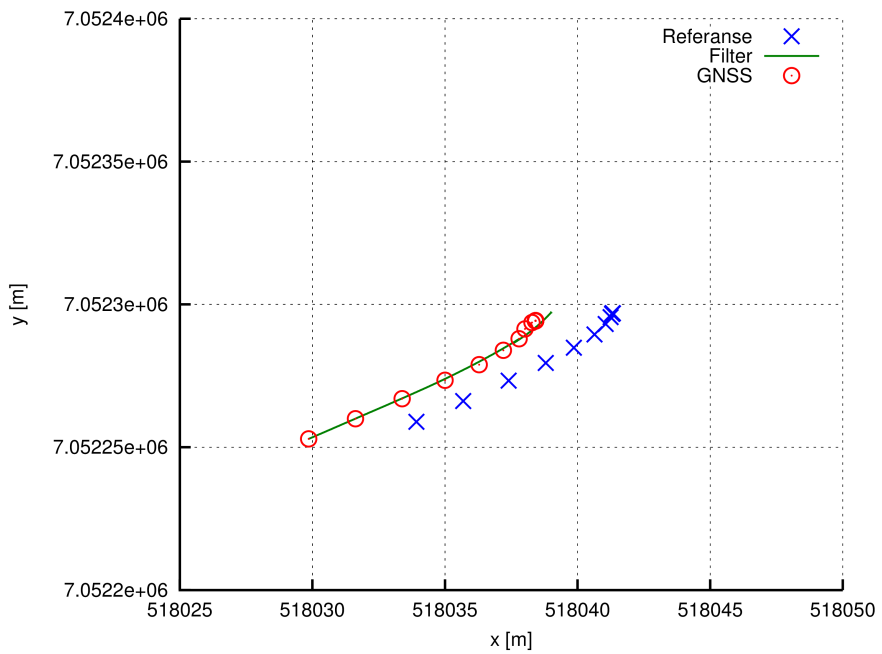
Tabell 4.8: Antall feildeteksjoner ved ulike metoder og ulike situasjoner.

	$var(a_x)$	$var(a_y)$	$var(a_x) + var(a_y)$
Rosenborg (301:400)	1	0	1
Rosenborg (1001:1100)	5	4	6
Rosenborg (1401:1500)	9	9	6

På bakgrunn av dette ble en funksjon for å sjekk om bilen står i ro implementert i filteret. Testen gikk ut på å regne ut variansen av måleresultatene fra akselerometerne x og y for det siste sekundet. Dersom variansen av x og y sammenlagt ligger under $0,048 m/s^2$, tror filteret at bilen står stille. Figur 4.5 viser resultatet uten stillesensor. Figur 4.6 viser resultatet med stillesensor aktivert.



Figur 4.5: Resultat uten stoppdetektor, Rosenberg (371:400).



Figur 4.6: Resultat med stoppdetektor, Rosenberg (371:400).

4.4 Bestemme retning fra GNSS-målt hastighet

Tabell 4.1, 4.2 og 4.3 viser at andelen støy i GNSS-mottakerne vil bli betydelig ved lave hastigheter. For å unngå feilaktige målinger av retning, ble det lagt inn en sjekk av hastigheten før det ble foretatt en GNSS-måling av retningen.

4.5 Kalmanfilteret i ulike situasjoner

Tabell 4.9 og 4.10 viser hvilke verdier som ble valgt på målestøyen og prosessstøyen. Som en generell regel ble målestøyen valgt ut fra antatt nøyaktighet på målingene og prosessstøyen ble valgt ut fra antakelser, prøving og feiling. $\sigma_{XYZ,G}^2$ $\sigma_{vxyz,G}^2$ er basert på målingene av GNSS-mottakerens nøyaktighet i tabell 4.1. $\sigma_{vxyz,I}^2$ er støy på hastigheten beregnet fra info fra IMU-en. Denne er satt litt høyere på grunn av drift i gyroene og vibrasjoner i akselerometerne. $\sigma_{\alpha,I}^2$ og $\sigma_{\beta,I}^2$ er målestøyen for roll og pitch målt med akselerometerne. $\sigma_{\gamma,G}^2$ er målestøyen for yaw beregnet fra GNSS-målt hastighet. Disse er satt såpass høyt for at vibrasjoner og støy ikke skal dominere resultatet. $\sigma_{\omega x,I}^2$, $\sigma_{\omega y,I}^2$ og $\sigma_{\omega z,I}^2$ er basert på målingene i tabell 4.4.

Tabell 4.9: Målestøy brukt i Kalman-filteret.

	Målestøy
$\sigma_{XYZ,G}^2$	(6 m) ²
$\sigma_{vxyz,G}^2$	(0,43 m/s) ²
$\sigma_{vxyz,I}^2$	(1 m/s) ²
$\sigma_{\alpha,I}^2$	(100 rad) ²
$\sigma_{\beta,I}^2$	(100 rad) ²
$\sigma_{\gamma,G}^2$	(100 rad) ²
$\sigma_{\omega x,I}^2$	(0,0134 rad/s) ²
$\sigma_{\omega y,I}^2$	(0,0234 rad/s) ²
$\sigma_{\omega z,I}^2$	(0,0053 rad/s) ²

q_1 er prosessstøyen for posisjonen, q_2 for hastigheten, $q_{3,\alpha}$, $q_{3,\beta}$, $q_{3,\gamma}$ for roll, pitch og yaw, $q_{4,\omega x}$, $q_{4,\omega y}$, $q_{4,\omega z}$ for rotasjonsratene og q_5 for bias.

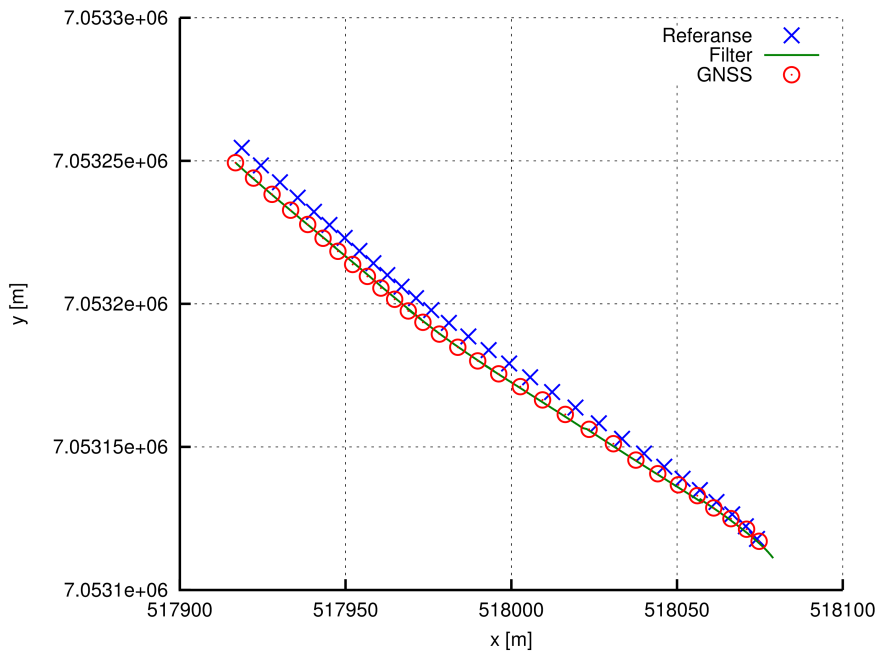
Figur 4.7 og 4.8 viser hvordan kalman-filteret yter sammenlignet med referanse og bare GNSS-mottakeren. GNSS-oppdatering gis hvert 5. sekund i to ulike situasjoner.

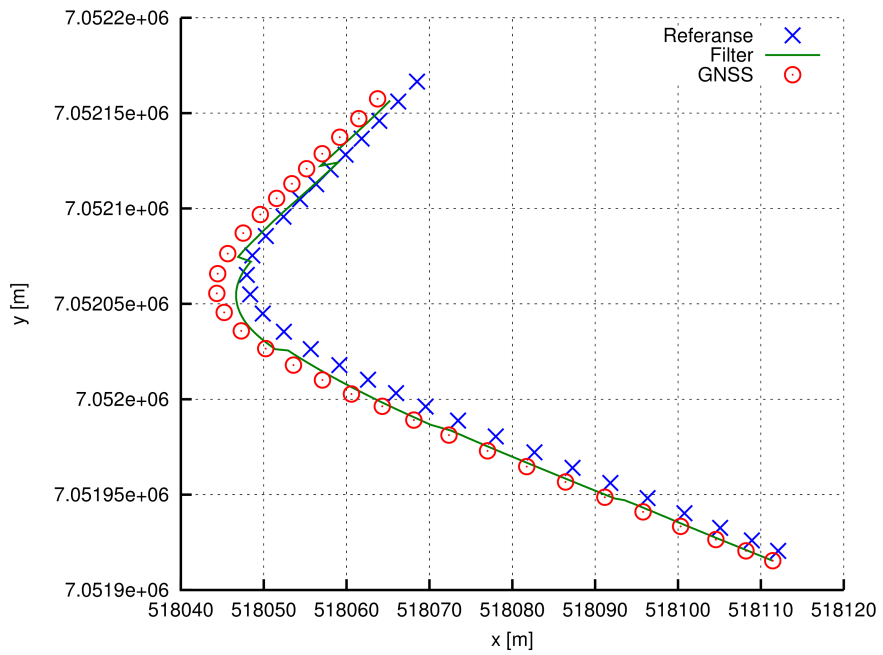
Figur 4.9 og 4.10 viser hvor stort avviket er mellom beregnet posisjon og referanse-mottakeren. Det ble gjennomført tester med en oppdateringsfrekvens på 5 s, 10 s, 15 s og 30 s.

Kalman-filteret ble også testet i to situasjoner i Ilsvikøra-datasettet. Resultatet fra disse testene kan ses i figur 4.11 og 4.12.

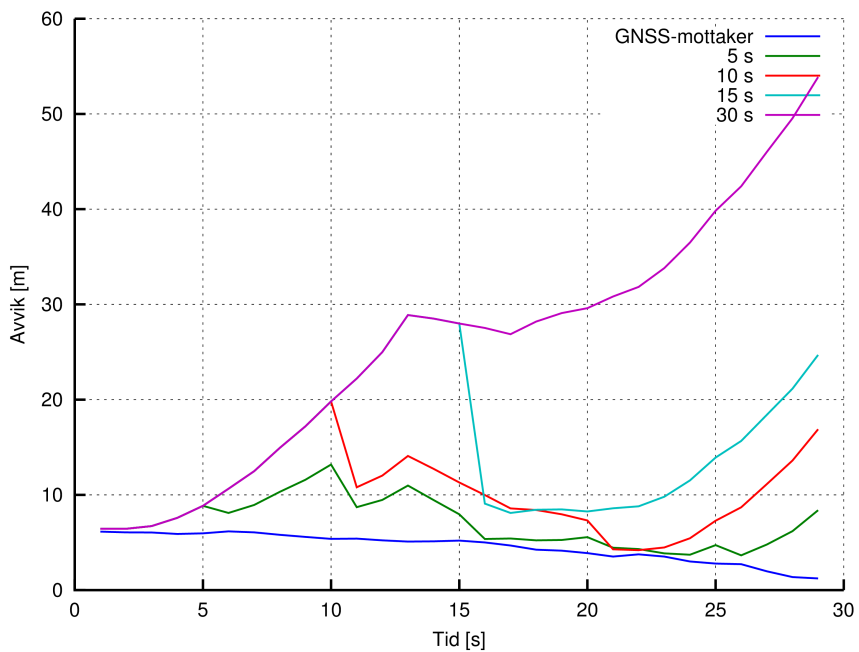
Tabell 4.10: Prosesstøy brukt i Kalman-filteret.

Prosesstøy	
q_1	$(2,2 \text{ m})^2$
q_2	$(2 \text{ m/s})^2$
$q_{3,\alpha}$	$(1 \text{ rad})^2$
$q_{3,\beta}$	$(1 \text{ rad})^2$
$q_{3,\gamma}$	$(3,16 \text{ rad})^2$
q_{4,ω_x}	$(0,0134 \text{ rad/s})^2$
q_{4,ω_y}	$(0,0234 \text{ rad/s})^2$
q_{4,ω_z}	$(0,0053 \text{ rad/s})^2$
q_5	$(0,001 \text{ rad/s})^2$

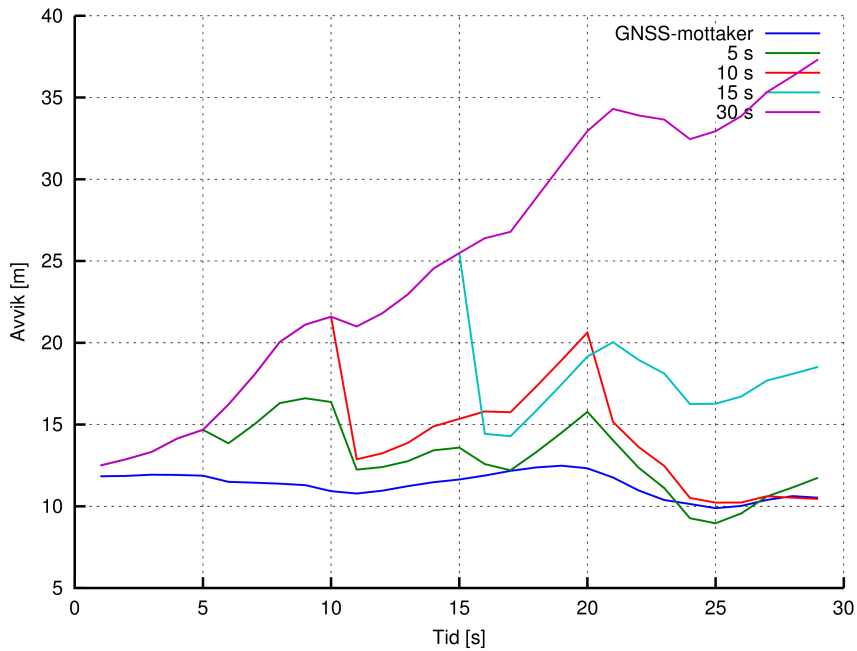
**Figur 4.7:** Test av filteret med GNSS-oppdateringer hvert 5. sekund, Rosenborg (31:60).



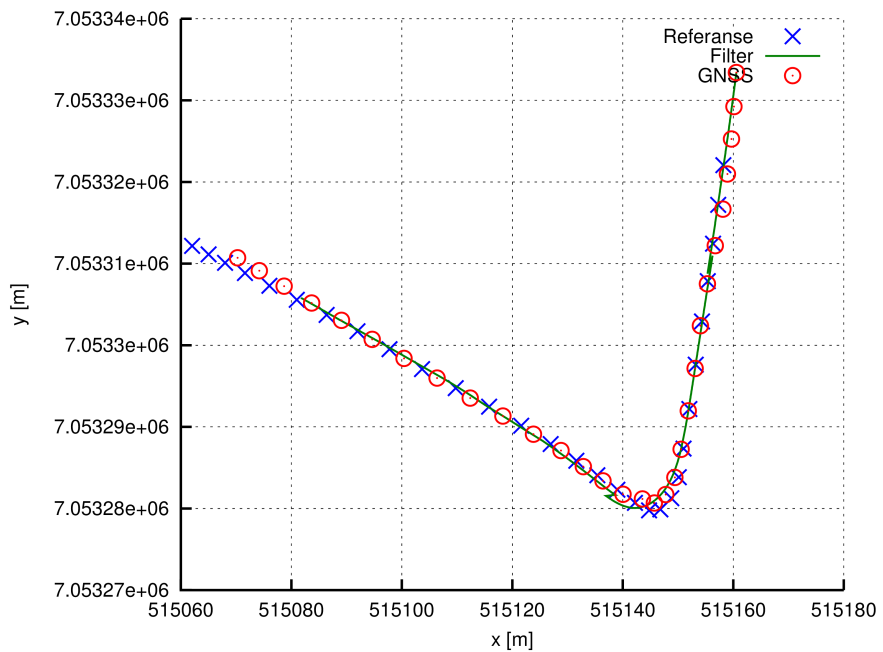
Figur 4.8: Test av filteret med GNSS-oppdateringer hvert 5. sekund, Rosenberg (331:360).



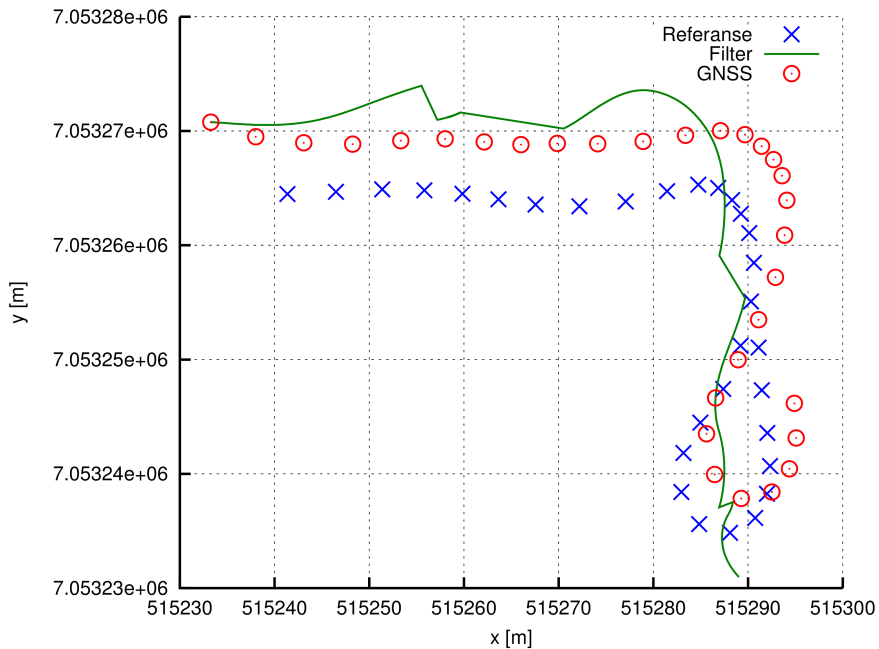
Figur 4.9: Test av filteret med ulik GNSS-oppdateringsfrekvens, Rosenberg (31:60).



Figur 4.10: Test av filteret med ulik GNSS-oppdateringsfrekvens, Rosenberg (331:360).



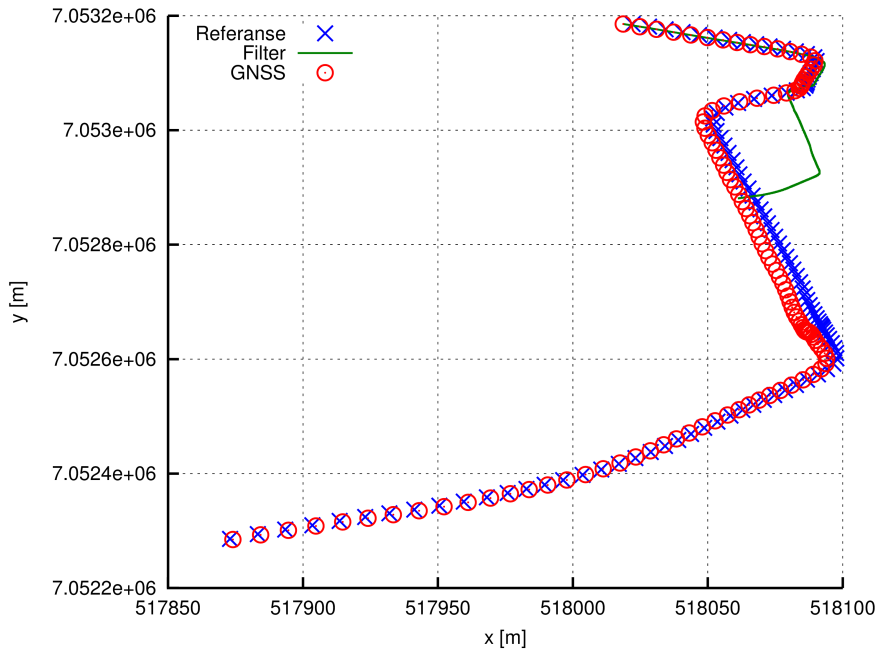
Figur 4.11: Test av filteret med GNSS-oppdatering hver 5. sekund, Ilsvikøra (271:300).



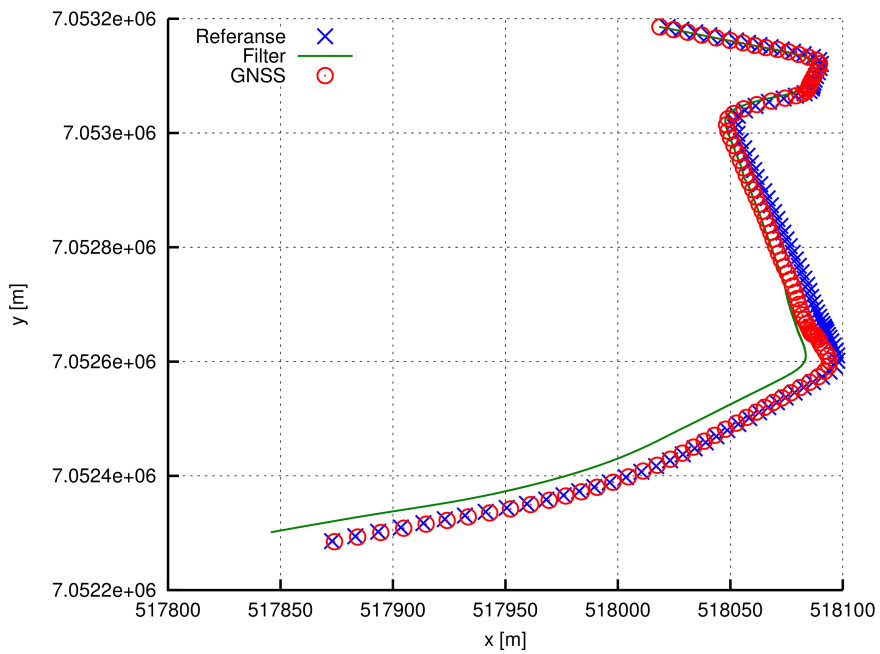
Figur 4.12: Test av filteret med GNSS-oppdatering hver 5. sekund, Ilsvikøra (481:510).

4.6 Nøyaktig hastighetsmåler

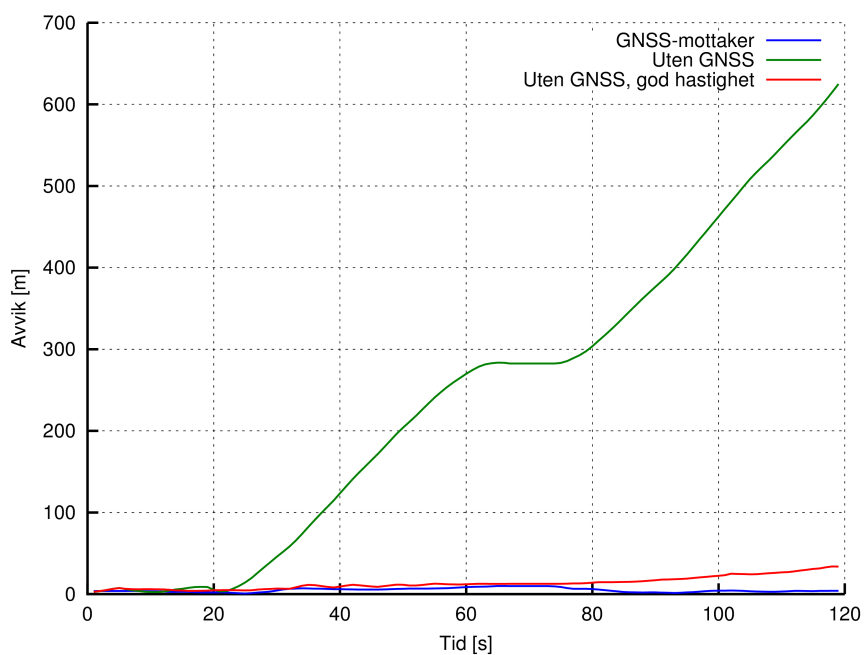
For å sjekke hvor godt akselerometerne klarer å holde korrekt hastighet og hvor godt gyroskopene klarer å holde korrekt orientering på bilen, ble det gjennomført en test hvor IMU-en brukes for å beregne posisjonen over en periode på 120 s uten oppdateringer fra GNSS-mottakeren. Resultatet fra denne testen vises i figur 4.13. Deretter ble samme test utført, men denne gangen ble nøyaktig hastighetsoppdatering hentet fra referansen hvert sekund. Resultatet fra denne testen vises i figur 4.14. I figur 4.15 vises avviket mellom posisjonene fra GNSS-mottakeren og referansen oppdatert en gang i sekundet. Avviket mellom beregnet posisjon fra bare IMU-en og referansen vises også i figuren. Det samme gjør beregnet posisjon fra IMU-en med nøyaktig hastighetsoppdatering en gang i sekundet sammenlignet med referansen.



Figur 4.13: Uten nøyaktig hastighetsmåler og uten GNSS-oppdateringer, Rosenberg (631:750).



Figur 4.14: Med nøyaktig hastighetsmåler og uten GNSS-oppdateringer, Rosenberg (631:750).



Figur 4.15: Avvik fra referanse ved bruk av nøyaktig fartsmåler, Rosenberg (631:750).

Kapittel 5

Analyse

5.1 GNSS-mottaker

Sammenligningen av de ulike GNSS-mottakerne sier noe om nøyaktigheten. Her er det mange faktorer som spiller inn. Eksempler på dette er forsinkelse i ionosfæren, antall tilgjengelige satellitter, geometrien til de tilgjengelige satellittene og støy internt i mottakeren. Beregningene i Rosenborg-datasettet i tabell 4.1 gir størst avvik. Her ser man at GNSS-mottaker 1 viser store avvik fra referansemottakeren både i posisjon og hastighet. Det er vanskelig å si akkurat hva som er årsaken til dette. Det kan være en feil i selve mottakeren, eller bare at mottakeren mister signalet fra satellittene og gir ut feil verdier fram til den kan levere riktige verdier.

De to andre datasettene, Ilsvikøra og E6, viser mindre forskjeller mellom mottakerne. Se tabell 4.2 og 4.3. E6-datasettet inneholder ikke målinger av hastigheten. For enkelthetskyld blir ikke dette datasette brukt videre i oppgaven, selv om det ville vært mulig å beregne hastighetene ut fra endringene i posisjonen. GNSS-mottaker 2 gir gir jevnt over bra målinger i alle tre datasettene, og denne ble derfor brukt i det videre arbeidet. Nøyaktigheten på denne varierer fra 2,32–12,72 m i posisjon for hver av aksene og 0,05–0,76 m/s for hastighet i hver av aksene.

5.2 Analyse av måledata

Gyroskopmålingene for kjøring rett fram (figur 4.1) viser små utslag når bilen kjører rett fram med konstant hastighet og retning. Rotasjonsraten rundt z-aksen er veldig liten, mens det er noe utslag i rotasjonsraten i x- og y-aksen. Dette skyldes at gyroskopene registrerer små ujevnheter i veien. Målingene når bilen står i ro (figur 6.2 i vedlegg 6.4) viser at målingene inneholder en bias. Det vil si at de måler en rotasjonsrate selv om bilen står helt i ro.

Gyroskopmålingene i er skarp sving (figur 4.2) viser at noe er galt. Når bilen svinger på det meste viser gyroene en rotasjonsrate i motsatt retning av det bilen svinger. Dette skjer

i når rotasjonsraten går er omtrent $0,6 \text{ rad/s}$. Omregnet til grader får vi $34,4^\circ/\text{s}$ som igjen tilsvarer $34\,400 \text{ m}^\circ/\text{s}$. Fra databladet til gyroskopet [11] står det at rotasjonsraten måles med 16-bits oppløsning. Dette tilsvarer 65536 ulike verdier. Siden gyroskopene måler både positive og negative rotasjonsrater vil det dynamiske området være halvparten av dette. Det er derfor rimelig å anta at årsaken til at rotasjonsraten viser det motsatte når den blir større enn $0,6 \text{ rad/s}$, skyldes at sensoren er konfigurert slik at dette er det høyeste den kan måle. En konsekvens av dette er at beregning gjort basert på IMU-en i krappe svinger vil gir helt feil resultater.

Figur 4.3 viser at et er små utslag i akselerometerne når bilen står i ro med motoren i gang. Avviket fra 0 i x-retning og y-retning viser at bilen ikke står helt horisontalt i disse retningene. Dette skyldes at litt av gravitasjonskraften lekker over i de andre aksene. I figur 4.4 ser man tydelig akselerasjonen i x- retning som følge av at bilen akselererer og en negativ akselerasjon i y-retningen som følge av sentripetalakselerasjonen når bilen svinger til venstre.

Gjennomsnittet av gyroskopene over en periode når bilen står i ro vises i tabell 4.6. Her vises det tydelig at gyroskopene har en bias som må kompenseres for. En feil på $-0,004\,23 \text{ rad/s}$ tilsvarer $-0,242^\circ/\text{s}$. Over en periode på 30 s vil denne feilen vokse til $-7,271^\circ/\text{s}$ dersom det ikke kompenseres for bias. Årsaken til den store forskjellen mellom de to første og de to siste målingen skyldes at Rosenborg-datasettet er satt sammen av flere kjøreturer. Sannsynligvis har sensoren blitt skrudd av og på mellom forsøkene og biasene har endret seg.

Akselerasjonsmålingene når brikken ligger i ro viser at små avvik mellom de ulike aksene. Målingene ligger likevel litt over hva man kunne forvente å måle på disse breddegrader ($9,821\,71 \text{ m/s}^2$). Dette kan løses ved å gange med en faktor som er litt mindre enn 1.

5.3 Stoppdeteksjon

Basert på variansen til akselerometerne i ulike situasjoner (tabell 4.5) ble en stoppdetektor utviklet. Tabell 4.8 viser antall feildeteksjoner for de ulike metodene. Den endelige sjekken ble implementert i filteret. Forskjellen mellom vanlig filter og filter med stoppdetektor kan ses i figur 4.5 og 4.6. Her er det tydelig at Kalman-filteret fortsetter å framskrive posisjonen etter at bilen er stoppet, men at det ikke er et problem når stillesensoren er aktivert.

Det er viktig å vite når bilen står i ro for å kunne rette opp systemet ved å gjennomføre en nullhastighetsoppdatering. Antall feildeteksjoner for løsningen som ble valgt er likevel ganske høyt (6 feil ut av 100). Alle feilene er falske negative. Det vil si at bilen tror den kjører når den egentlig står i ro. Slik feil har små konsekvenser for posisjonen. En falsk positiv vil derimot har store konsekvenser. Det vil si at bilen tror den står stille når den egentlig kjører. En nullhastighetsoppdatering i slike tilfeller vil føre til store avvik mellom faktisk posisjon og beregnet posisjon. Før å redusere antall feildeteksjonen kan f.eks. flere informasjonskilder kombineres. Eksempler på dette er variansen til gyroskopene og GNSS-målt hastighet.

5.4 Kalman-filteret i ulike situasjoner

Kalman-filteret ble testet i ulike situasjoner. I figur 4.7 ser man at filteret klarer seg veldig godt på en strekning hvor bilen kjører nesten rett fram i omtrent konstant hastighet. Ved å velge ulik oppdateringsfrekvens på GNSS-mottakeren kan man se hvor godt IMU-en klarer seg uten støtte. Resultatet for de ulike oppdateringsfrekvensene vises i figur 4.9. Det viser seg at en oppdatering hver 5 s gir omtrent samme nøyaktighet som å bare bruke en GNSS-mottaker med en oppdatering hvert sekund. Øker man til 10 s blir nøyaktigheten omtrent på 20 m, 15 s gir omtrent 30 m og 30 s gir omtrent 50 m.

Figur 4.8 viser samme forsøk utført i en sving. Her ser man tydelig hvordan IMU-en prøver å følge samme bevegelse som referansen i svingen, mens GNSS-mottakeren som avviker litt fra referansen drar posisjonen mot seg.

Filteret ble også teste på IISvikøra-datasettet hvor mottakerforholdene er litt dårligere på grunn av høye bygninger. Av figur 4.11 ser man at filteret klarer å holde posisjonen ganske bra. I den neste testen som inneholder en skarp sving (figur 4.12) er resultatet ikke like bra. Her ser man at filteret har problemer med å holde både riktig retning og hastighet. Dette skyldes trolig dårlige mottakerforhold som gir unøyaktige hastigheter og orientering på bilen. Helt mot slutten svinges det lite hvor referansen tar en U-sving. Dette skyldes at rotasjonsraten som gyroskopene skal måle er for høy. Samme effekt som oppstår i figur 4.2 gjør at bilen tror at den svinger skarpt i motsatt retning av det den egentlig svinger i en liten periode.

5.5 Nøyaktig hastighetsmåler

Av figur 4.13 ser man at filteret har problemer med å holde posisjonen særlig lenger enn 20–30 s uten GNSS-oppdateringer. Dette skyldes at akselerometerne ikke klarer å holde hastigheten riktig. Ved å hente nøyaktig hastighet fra referansemottakeren (figur 4.14), ser man at filteret nå klarer å holde posisjonen betydelig lenger. Figur 4.15 viser at avviket mellom filteret nå er på samme nivå som GNSS-mottakeren i opptil 80 s og at det ender opp på omtrent 40 m etter 120 sekunder uten GNSS-oppdatering. Dette viser at gyroskopene klarer å holde orienteringen på bilen korrekt i lange perioder uten støtte fra GNSS.

Ved å legge til en ekstra sensor i bombrikken kan man øke tidsperioden den kan klare seg uten GNSS-oppdatering betydelig. En slik sensor kan for eksempel være en sensor som står på hjulet og teller antall rotasjoner og konverterer til hastighet. Moderne biler leveres ofte med et grensesnitt for å hente ut nøyaktig hastighet fra girkassen som kan brukes til dette.

5.6 Feilkilder

Det er flere mulige feilkilder som kan påvirker måleresultatene som ble presentert. Vibrasjoner fra motoren og bilens bevegelser kan påvirker akselerometer- og gyroskopmålingene. Det at sensorene ikke er plassert i senter i bilen kan også gi utslag i målingene. Selv om det ble gjort forsøk på å rette opp sensorene fordi de ikke var perfekt horisontert i bilen, vil målingene aldri bli helt perfekte. En annen feilkilde er at sensorene som er plassert i bom-

brikken kan være loddet skjevt på kretskortet. Bare noen tidels millimeter kan gi utslag i målingene.

Kapittel 6

Konklusjon

Analysen av måledataene fra GNSS-mottakeren og IMU-en har vist at kvaliteten på målingene til tider er dårlige. I Rosenborg-datasettet er gjennomsnittlig avvikene mellom GNSS-mottaker 1 og referansemottakeren 67 m i målinger av høyde og 26 m i målinger av breddegrad. Det er vanskelig å si noe om hva de store avvikene i GNSS-mottaker 1 kommer av. Likevel er det noe man må ta høyde for at kan skje i GNSS/INS-integrasjon. Analysen av måledataene fra gyroskopene i ulike situasjoner avdekket at også av gyroskopene ikke klarer å måle rotasjonsrater som må forventes ved normal bilkjøring i urbane strøk. Gyroskopene kan omkonfigureres for å kunne registrere høyere rotasjonsrater. Egenskaper som varians, bias og gjennomsnitt i ulike situasjoner dannet grunnlaget for det videre arbeidet.

Forsøkene viste at på grunn av støy i GNSS-målingene er det vanskelig å avgjøre om bilen står i ro eller ikke. Drift og vibrasjoner i akselerometerne gjør at de har problemer med å registrere hastighetsendringer. Dette medførte at posisjonsberegninger basert på bare IMU-en vil gi store avvik etter omtrent 5–10 s. Gjennomsnittsmålingene av gyroskopene når bilen stod i ro viste at bias fører til drift i bilens orientering. Dette kan unngås ved å kompensere for bias i Kalman-filteet.

Et lineært Kalman-filter ble konstruert for å utføre GNSS/INS-integrasjon. Analysen av måledataene og bilens bevegelsesmønster dannet grunnlaget for denne prosessen. Gjennom tester av filteret ble måledata fra INS og GNSS brukt for å gi gode posisjonsestimater. Forsøkene har vist at GNSS-målingene hjalp til hvor INS hadde problemer, og motsatt at INS-målinger hjalp til på steder hvor GNSS-mottakeren hadde problemer.

En enkel stoppdetektor ble utviklet basert på kunnskap om variansen til akselerometerne i ulike situasjoner. I mange tilfeller fungerer stoppdetektoren meget bra, men i noen tilfeller detekterer den feil tilstand. Detektoren kan forbedres ved å kombinere målinger fra gyroskopene og GNSS-mottakeren. Det ble videre undersøkt at dersom en ekstra sensor hadde levert nøyaktige hastighetsoppdateringer til filteret, så er det mulig å holde posisjonen innenfor et avvik på 20–30 m i opptil 80 s. Dette vil være nyttig i for eksempel tunneler.

Det er mange retninger dette arbeidet kan videreføres i. Å se på hvilke sensorer som kan benyttes for å gi en nøyaktig måling av bilens hastighet vil kunne forbedre ytelsen til

systemet betydelig. I beregningene i dette arbeidet ble det ikke fokusert på å lage algoritmer som krever minst mulig regnekraft og strømforbruk. Det er mulig å se på hvordan algoritmene kan forenkles slik at de kan utføres på en liten mikrokontroller. Å gå fra posisjonering i tre dimensjoner til to dimensjoner vil også kunne gjøre beregningene mindre krevende.

Bibliografi

- [1] J. Farrell, *Aided navigation: GPS with high rate sensors*. New York: McGraw-Hill, 2008.
- [2] D. M. Bevly, "Global positioning system (gps): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, p. 256, jun 2004.
- [3] M. P. og Y. Gao, "Error and performance analysis of mems-based inertial sensors with a low-cost gps receiver," *Sensors*, vol. 8, pp. 2242–2245, mar 2008.
- [4] e. a. A. K. Gupta, "A kalman filtering approach for integrating mems-based ins and gps for land vehicle applications," Dept. of Electrical Engineering, Indian Institute of Technology, Madras, Tech. Rep., 2006.
- [5] e. a. Q. Ladetto, "Combining gyroscopes, magnetic compass and gps for pedestrian navigation," *Proceedings of the international symposium on kinematic systems in geodesy, geomatics, and navigation*, pp. 205–213, jun 2001.
- [6] e. a. M. Z. H. Bhuiyan, "Performance evaluation of multi-sensor fusion models in indoor navigation," *European Journal of Navigation*, vol. 11, pp. 205–213, aug 2013.
- [7] B. F.-A. og J. Vila-Carbó, "Integration of the ins and gps navigation systems using a kalman filter on a mobile phone," Universitat Politècnica de València, Tech. Rep., 2012.
- [8] e. a. p. Davidson, "Uninterrupted portable car navigation system using gps, map and inertial sensors data," in *Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium*.
- [9] —, "Using low-cost mems 3d accelerometer and one gyro to assist gps based car navigation system," in *Proceedings of 15th Saint Petersburg International Conference on Integrated Navigation Systems (May 2008)*.
- [10] G. W. og G. Bishop, "An introduction to the kalman filter," Department of Computer Science, University of North Carolina at Chapel Hill, Tech. Rep., 2006.

-
- [11] STMicroelectronics, “Mems motion sensor: three-axis digital output gyroscope,” des. 2010, rev. 3, L3G4200D. [Online]. Available: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00265057.pdf>
- [12] —, “Mems motion sensor: Ultra low power high performance 3-axis digital accelerometer,” jul. 2009, rev. 3, LIS331DLH. [Online]. Available: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00213470.pdf>

Vedlegg

6.1 Format på GPS-rådata

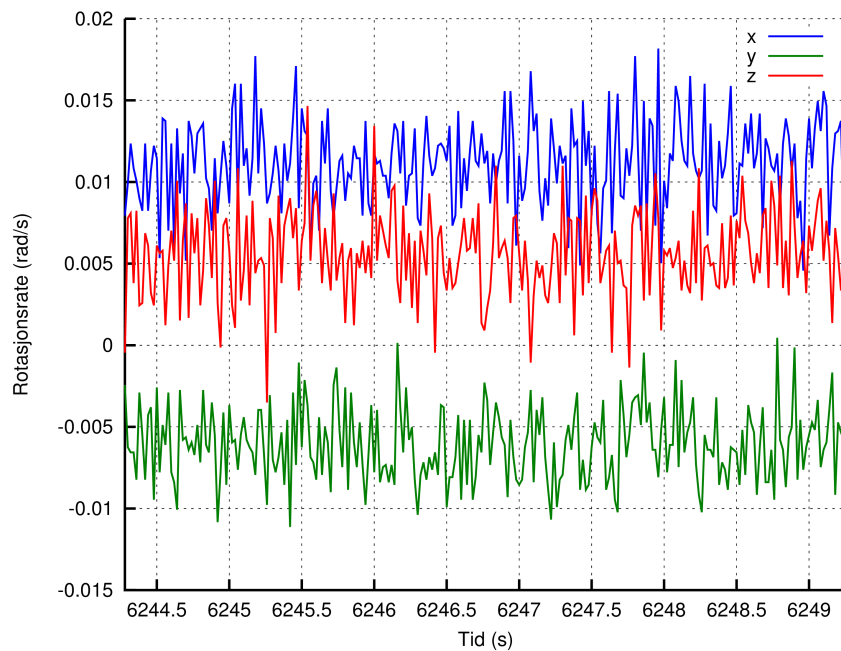
1. Equipment id (6 is reference)
2. Drive id
3. Leg id
4. Date yyyy-mm-dd (UTC)
5. Time hh:mm:ss.fff (UTC)
6. Gps week
7. Gps time of week, seconds
8. OBU Timestamp, seconds
9. GNSS Good flag (1 is good, 0 is bad)
10. GNSS Fix Type
11. Latitude
12. Longitude
13. Ellipsode height (meters)
14. Geoid height, height over sea level (meters)
15. Speed over ground (m/s)
16. Course over ground (deg)
17. Yaw (deg/second)
18. Pitch (deg/second)
19. Roll (deg/second)
20. Horizontal pos error (meter)
21. Vertical pos error (meter)

-
22. sN, st.dev of position, in north direction (meters)
 23. sE, st.dev of position, in east direction (meters)
 24. sD, st.dev of position, in down direction (meters)
 25. HDOP
 26. PDOP
 27. NED n speed (m/s)
 28. NED e speed (m/s)
 29. NED d speed (m/s)
 30. Number of satellites used in solution
 31. Average signal quality
 32. SBAS Active
 33. rN integrity, north (m)
 34. rE integrity, east (m)
 35. rD integrity, down (m)

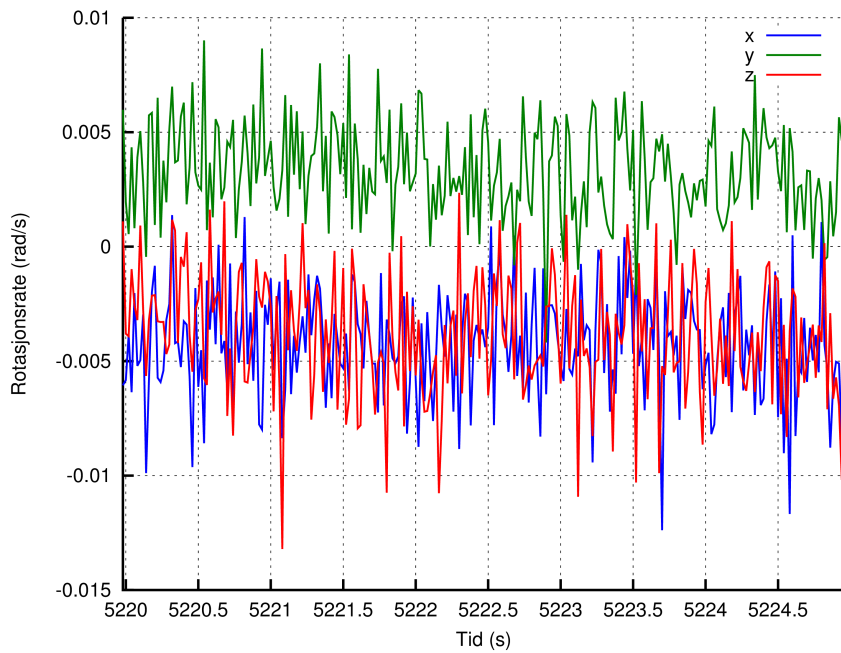
6.2 Format på MEMS-rådata

1. Equipment id (6 is reference)
2. Drive id
3. Leg id
4. Date yyyy-mm-dd (UTC)
5. Time hh:mm:ss.fff (UTC)
6. Gps week
7. Gps time of week, seconds
8. OBU Timestamp, seconds
9. Accel X
10. Accel Y
11. Accel Z
12. Gyro X
13. Gyro Y
14. Gyro Z

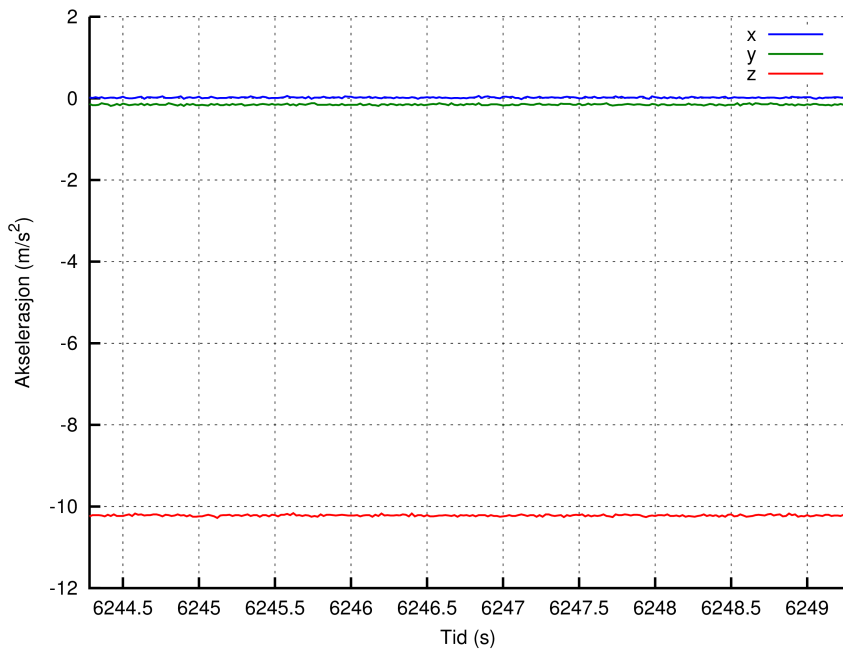
6.4 Gjennomgang og behandling av rådata



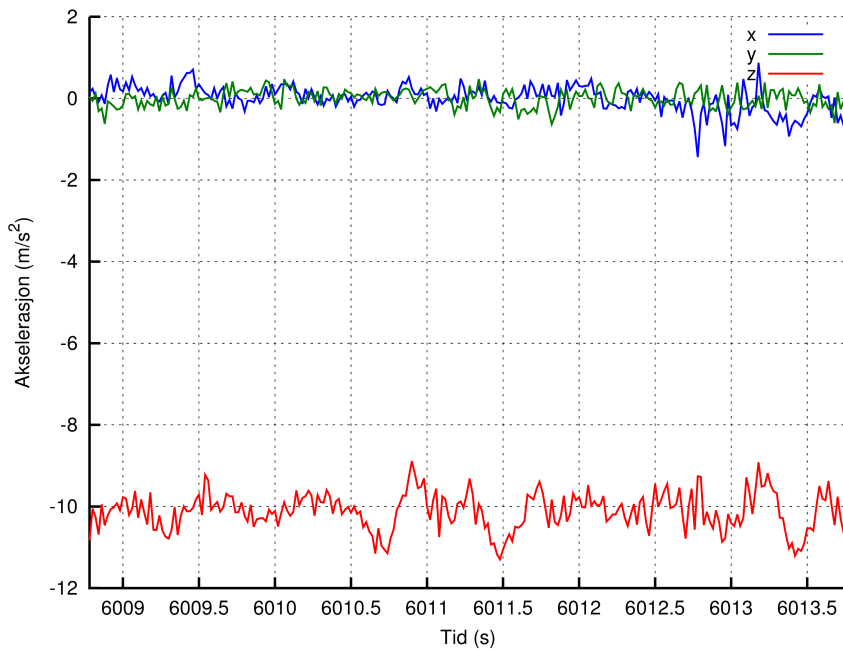
Figur 6.1: Gyroskopmåling når sensorene ligger helt stille, i ro (2:6).



Figur 6.2: Gyroskopmåling når bilen har stoppet med motoren i gang, Rosenberg (381:385).



Figur 6.3: Akselerometermåling når sensorene ligger helt stille, i ro (2:6).



Figur 6.4: Akselerometermåling når bilen kjører rett fram, Rosenberg (761:765).

6.5 Programkode

```

% External functions used by this code
% rad2deg() - convert from radians to degrees
% deg2rad() - convert from degrees to radians
% getData() - read selected data from data files
% toMems() - convert from position time to mems time
% crossm() - calculate cross product of vector
% Rz() - z-axis rotation
% Ry() - z-axis rotation
% Rz() - z-axis rotation
% geod2ECEF - convert from geographical til ECEF coordiantes
% ECEF2geod - convert from ECEF to geographical coordiantes

% Define range for calculation
posRange = 481:510;

% Define constants
oie_e = [0;0;7292115e-11];
dt = 1/50;
radius = 6371000;
geb_g = [0;0;9.82171];

% Read data files to memory
posFile = '../Data/Rosenborg-2013/Rosenborg-Jan-2013-pos-2.txt';
%posFile = '../Data/Illsvikora/Illsvikora-Pos-2.txt';

```

```

memsFile = '../Data/Rosenborg-2013/Rosenborg-Jan-2013-mems-2.txt';
%memsFile = '../Data/Ilsvikora/Ilsvikora-MEMS-2.txt';
refFile = '../Data/Rosenborg-2013/Rosenborg-Jan-2013-pos-6.txt';
%refFile = '../Data/Ilsvikora/Ilsvikora-Pos-6.txt';

memsRange = toMems(posRange);
rangeP = [posRange(1) posRange(end)];
rangeM = [memsRange(1)-250 memsRange(end)+250];

lat = deg2rad(getData(posFile,rangeP,11));
lon = deg2rad(getData(posFile,rangeP,12));
h = getData(posFile,rangeP,14);
v_ned = getData(posFile,rangeP,[27 29]);
posTime = getData(posFile,rangeP,8);

lat_ref = deg2rad(getData(refFile,rangeP,11));
lon_ref = deg2rad(getData(refFile,rangeP,12));
h_ref = getData(refFile,rangeP,14);
v_ref = getData(refFile,rangeP,[27 29]);

acc = getData(memsFile,rangeM,[9 11]);
gyro = getData(memsFile,rangeM,[12 14]);
memsTime = getData(memsFile,rangeM,8);

% Rotate accelerometers
orientation = [0.0000000000000000 0.438057067391664 -0.01816728115647];
fis_s = 0.97396*9.82171*acc/1000;
Cs_b_a = Rz(pi)*Ry(0+orientation(2))*Rx(0+orientation(3));
fis_b = Cs_b_a*fis_s;

% Rotate gyroscopes
ois_s = -gyro/1000;
Cs_b_g = Rz(0)*Ry(0+orientation(2))*Rx(0-orientation(3));
ois_b = deg2rad(Cs_b_g*ois_s);

% Kalman filter parameters
H1=[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; % Design matrix gps and
    mems
    0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
    0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0];
H2=[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0; % Design matrix mems
    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0;

```

```

0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0;
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0];
phi=[1 0 0 dt 0 0 0 0 0 0 0 0 0 0 0;           % Transition
      matrix
      0 1 0 0 dt 0 0 0 0 0 0 0 0 0 0 0;
      0 0 1 0 0 dt 0 0 0 0 0 0 0 0 0 0;
      0 0 0 1 0 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 1 0 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0;
      0 0 0 0 0 0 1 0 0 dt 0 0 -dt 0 0;
      0 0 0 0 0 0 0 1 0 0 dt 0 0 -dt 0;
      0 0 0 0 0 0 0 0 1 0 0 dt 0 0 -dt;
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 1 0 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 1 0 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 1 0 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 1 0;
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];

offset = 62;           % Offset between mems and gnss measurements

% Measurement noise
s_pos = (6)^2;
s_v = (0.43)^2;
s_vp = (1)^2;
s_acc_roll = 1e4;
s_acc_pitch = 1e5;
s_gps_yaw = 1e10;
s_gyro_r = 1.8e-4;
s_gyro_p = 5.5e-4;
s_gyro_y = 2.9e-5;
R1=diag([s_pos s_pos s_pos s_v s_v s_v s_vp s_vp s_vp s_acc_roll
         s_acc_pitch s_gps_yaw s_gyro_r s_gyro_p s_gyro_y]); % Measurement
         noise gnss and mems
R2=diag([s_vp s_vp s_vp s_acc_roll s_acc_pitch s_gyro_r s_gyro_p s_gyro_y
         ]); % Measurement noise mems only

% Process noise
q1=(0.5e1);
q2=(4);
q3r=(1e0);
q3p=(1e0);
q3y=(1e1);
q4r=(1.8e-4);
q4p=(5.5e-4);
q4y=(2.9e-5);
q5=(1e-6);
Q=[q1*dt+q2*dt^3/3 0 0 q2*dt^2/2 0 0 0 0 0 0 0 0 0 0 0;           %
   Process noise matrix
   0 q1*dt+q2*dt^3/3 0 0 q2*dt^2/2 0 0 0 0 0 0 0 0 0 0;
   0 0 q1*dt+q2*dt^3/3 0 0 q2*dt^2/2 0 0 0 0 0 0 0 0 0;
   q2*dt^2/2 0 0 q2*dt 0 0 0 0 0 0 0 0 0 0;
   0 q2*dt^2/2 0 0 q2*dt 0 0 0 0 0 0 0 0 0;
   0 0 q2*dt^2/2 0 0 q2*dt 0 0 0 0 0 0 0 0;
   0 0 0 0 0 q3r*dt+q4r*dt^3/3-q5*dt^3/3 0 0 q4r*dt^2/2 0 0 -q5*dt^2/2 0
   0;

```

```

0 0 0 0 0 0 0 q3p*dt+q4p*dt^3/3-q5*dt^3/3 0 0 q4p*dt^2/2 0 0 -q5*dt^2/2
0;
0 0 0 0 0 0 0 q3y*dt+q4y*dt^3/3-q5*dt^3/3 0 0 q4y*dt^2/2 0 0 -q5*dt
^2/2;
0 0 0 0 0 0 q4r*dt^2/2 0 0 q4r*dt 0 0 0 0 0;
0 0 0 0 0 0 0 q4p*dt^2/2 0 0 q4p*dt 0 0 0 0 0;
0 0 0 0 0 0 0 q4y*dt^2/2 0 0 q4y*dt 0 0 0 0;
0 0 0 0 0 0 -q5*dt^2/2 0 0 0 0 0 q5*dt 0 0;
0 0 0 0 0 0 -q5*dt^2/2 0 0 0 0 0 q5*dt 0;
0 0 0 0 0 0 0 -q5*dt^2/2 0 0 0 0 0 q5*dt];

% Sync mems to gnss measurements
posNum = 1;
memsNum = 1;
while (memsTime(memsNum) < posTime(posNum))
    memsNum = memsNum + 1;
end

% Initialize
pitch = atan2(-v_ref(3,posNum), sqrt(v_ref(1,posNum)^2+v_ref(2,posNum)^2));
yaw = mod(atan2(v_ref(2,posNum), v_ref(1,posNum)), 2*pi);
[x,y,z] = geod2ECEF(radius, radius, lat(posNum), lon(posNum), h(posNum));
Ce_g = [-sin(lat(posNum))*cos(lon(posNum)) -sin(lat(posNum))*sin(lon(
posNum)) cos(lat(posNum));
-sin(lon(posNum)) cos(lon(posNum)) 0;
-cos(lat(posNum))*cos(lon(posNum)) -cos(lat(posNum))*sin(lon(
posNum)) -sin(lat(posNum))];
Xr=[x;y;z;Ce_g'*v_ref(:,posNum);0;pitch;yaw;ois_b(:,memsNum)
;-0.0042254;0.0033100;-0.0038208]; % Initial state vector
P=diag([3 3 0.36 0.36 0.36 1 1 1 1e-4 1e-4 1e-4 1e-1 1e-1 1e-1]);
% Initial covariance

res = [Xr];
res_e = [];

% Filter loop
for posNum=1:(length(posRange)-1)
    posUpdated = false;

    while (memsTime(memsNum) < posTime(posNum+1)) % Sync mems and gnss

        memsNum = memsNum-offset; % offset

        %% Mems prediction
        % Compute Cb_e
        yaw = Xr(9);
        pitch = Xr(8);
        Cb_g = Rz(yaw)*Ry(pitch)*Rx(0);
        Cb_e = Ce_g'*Cb_g;

        % Velocity in e-frame
        veb_e = Xr(4:6);

        % Compute gyro rate
        bias = Xr(13:15);
        oeb_b = ois_b(:,memsNum)-bias - Cb_e'*oie_e;
        Oeb_b = crossm(oeb_b);

```

```

% Compute Cb_e_dot
Cb_e_dot = Cb_e*Oeb_b;

% Update Cb_e
Cb_e_hat = Cb_e + Cb_e_dot*dt;

% Compute Coriolis components
Oie_e = crossm(oie_e);

% Compute acceleration in e-frame
a = fis_b(:,memsNum)+Rz(0)*Ry(-Xr(8))*Rx(-Xr(7))*geb_g; %
    Compensate for gravitation
aeb_e = Cb_e_hat*a - (2*Oie_e)*veb_e;

% Compute velocity in e-frame
veb_b = [norm(veb_e); 0; 0];
veb_e = Cb_e_hat*veb_b + aeb_e*dt;

% Compute roll and pitch
acc_roll = atan2(-fis_b(2,memsNum),-fis_b(3,memsNum));
acc_pitch = atan2(fis_b(1,memsNum),sqrt(fis_b(2,memsNum)^2+fis_b(3,memsNum)^2));

% If pos available
posAvailable = mod(posNum,5)==1; % GNSS update every 5.
    second
if(posAvailable && !posUpdated && posNum~=1 && norm(v_ned
    (:,posNum))>3) % ignore when speed is < 3 m/s
    posUpdated = true;

    % Measurements
    [x,y,z] = geod2ECEF(radius,radius,lat(posNum),lon(
        posNum),h(posNum));
    v = Ce_g'*[v_ned(1,posNum); v_ned(2,posNum); v_ned
        (3,posNum)];
    gps_yaw = mod(atan2(v_ned(2,posNum),v_ned(1,posNum
        )),2*pi);
    z=[x; y; z; v; veb_e; acc_roll; acc_pitch; gps_yaw
        ; ois_b(:,memsNum)];
    H = H1;
    R = R1;

% Mems only available
else

    % Measurements
    z=[veb_e; acc_roll; acc_pitch; ois_b(:,memsNum)];
    H = H2;
    R = R2;

end

% Stop detection
variance = (var(fis_b(1,(memsNum-50):memsNum)) + var(fis_b
    (2,(memsNum-50):memsNum)));
if (variance < 0.048)

```

```

Xr(4:6) = 0; % Set speed to zero
Xr(7) = atan2(mean(-fis_b(2,memsNum-50:memsNum)),
              mean(-fis_b(3,memsNum-50:memsNum))); % Roll
Xr(8) = atan2(mean(fis_b(1,memsNum-50:memsNum)),
              sqrt(mean(fis_b(2,memsNum-50:memsNum))^2+mean(
                  fis_b(3,memsNum-50:memsNum))^2)); % Pitch
res = [res Xr];

else

    % Kalman filter
    % System size
    [n,e]=size(H);

    % Time update
    Xr=phi*Xr;
    P=phi*P*phi'+Q;

    % Kalman gain
    K=P*H'*inv(H*P*H'+R);

    % Measurement update
    Xr=Xr+K*(z-H*Xr);

    % Update covariance
    P=(eye(e)-K*H)*P;

    res = [res Xr];

    [lat_,lon_,h_]=ECEF2geod(radius,radius,res(1,end),
                           res(2,end),res(3,end));
    Ce_g = [-sin(lat_)*cos(lon_) -sin(lat_)*sin(lon_)
            cos(lat_);
            -sin(lon_) cos(lon_) 0;
            -cos(lat_)*cos(lon_) -cos(lat_)*sin(lon_)
            -sin(lat_)];

end

memsNum = memsNum+1+offset; % offset

end

end

% Convert to lat lon for presentation
for j=1:length(res)
    [lat_,lon_,h_]=ECEF2geod(radius,radius,res(1,j),res(2,j),res(3,j))
    ;
    p = [lat_; lon_; h_];
    res_e = [res_e p];
end

% Display plot and prepare for saving
set(0,'DefaultLineLineWidth',3)

la = lat(1);

```

```
he = h(1)+radius;
figure
plot(lon_ref*he*cos(la),lat_ref*he,'x',res_e(2,:)*he*cos(la),res_e(1,:)*he
      , lon*he*cos(la), lat*he, 'o')
grid on;
set(gcf,'PaperUnits','centimeters','PaperPosition',[0 0 12.6 9.45])
set(gca, 'LineWidth', 2, 'Box', 'off');
legend('Referanse','Filter','GNSS')
xlabel('xL[m]')
ylabel('yL[m]')
print('fig.png', '-dpng', '-r600', '-F:Helvetica:8')
```