

# HÅNDTERING AV INTEGRITET VED BRUK AV GNSS/INS I BYKJERNER

INTEGRITY ASSURANCE BY GNSS/INS IN URBAN AREAS

ARE JO NÆSS

UNIVERSITETET FOR MILJØ- OG BIOVITENSKAP

INSTITUTT FOR MATEMATISKE REALFAG OG TEKNOLOGI  
MASTEROPPGAVE 30 STP. 2012





## Sammendrag

I løpet av de siste 10-20 årene har måten vi navigerer på endret seg mye. GNSS-mottakere får plass på små kretskort, og i dag er det helt vanlig å bruke mobiltelefonen når man skal finne fram til et nytt sted. Ofte er det viktigere at enheten leverer en posisjon, enn at posisjonen er kvalitetssikret. Om en slik lav-kost sensor skal brukes til for eksempel veiprising er integriteten mye viktigere.

Denne oppgaven går ut på å avdekke integriteten til rimelige GNSS/INS-sensorer. En prototype sensor ble testet og analysert, samtidig som det ble simulert en tilsvarende sensor for sammenligning. Et koordinatbasert integritetsmål ble definert og testet på de aktuelle sensorene. Ved å sammenligne og teste på spesifikke områder, kunne det koordinatbaserte integritetsmålet vurderes opp mot andre metoder for beregning av integritet.

Observasjon av hastighet og satellittgeometri viste seg å være de to faktorene som bidrar på best måte for beregning av integriteten. Bykjernen viste seg å by på store utfordringer for den utprøvde sensoren når det kommer til posisjonering med integritetsmål.

## Abstract

The way we navigate have changed a lot over the last 10-20 years. Today, small GNSS receivers is quite common equipment on cell phones and it is common to use the cell phone for navigating purposes. It's often more important that the device outputs a position fix, rather than any quality assurance of the position. If an GNSS/INS based system is used in a toll road application, then, the quality control, and integrity is of great importance.

This thesis goal is to reveal the reliability of low-cost GNSS/INS-sensors. A prototype sensor was tested and analyzed. At the same time a sensor where simulatedet to perform as a similar sensor for comparison. A quality control, based on coordinates was derived and tested on the different sensors. By comparing the different results against each other, and calculate the external reliability, based on *Minimal Detectable Bias* it was possible to make an conclusion.

Observation of velocity and the satellite geometry is proved to be the two factors that contribute most for the calculation of reliability. The urban areas proved to offer great challenges to the sensor when it came to quality assurance.

## Forord

Denne masteroppgaven avslutter min utdanning i Geomatikk ved Universitetet for miljø- og biovitenskap (UMB). Treghetsnavigasjon og GNSS-systemer har fasinert meg lenge, da det finnes så utrolig mange bruksområder for systemene. Det å få lov til å prøve ut nye sensorer, og en oppgave som er fremtidsrettet gjorde at det var et lett valg å ta.

Oppgaven er et resultat av forskningsarbeide som min hovedveileder, professor Jon Glenn Gjevestad er engasjert i. Jon Glenn er ansatt ved Geomatikk-seksjonen på Institutt for Matematiske realfag og teknologi på UMB. Jeg ønsker å takke Jon Glenn for all hjelp, og for å ha vært løsningsorientert og motiverende gjennom arbeidet med oppgaven.

Jeg ønsker å takke Q-Free for lån av OBU, og for et interessant innblikk i noen av utfordringene som ligger innen veiprisning.

Til slutt ønsker jeg å takke studiekamerater, kjæreste og familie for god støtte og humor gjennom studietiden min her ved UMB.

UMB, Ås, 15. mai 2012

Are Jo Næss

# Innhold

<b>1</b>	<b>Innledning</b>	<b>2</b>
1.1	Problemstilling . . . . .	3
<b>I</b>	<b>Teori og bakgrunn</b>	<b>5</b>
<b>2</b>	<b>Bakgrunn</b>	<b>6</b>
2.1	Estimering og kvalitetskontroll . . . . .	6
2.1.1	Minste kvadraters metode . . . . .	6
2.2	Presisjon . . . . .	8
2.3	Nøyaktighet . . . . .	8
2.4	Integritet . . . . .	8
2.5	Pålitelighet . . . . .	10
<b>3</b>	<b>Koordinatsystemer</b>	<b>11</b>
3.1	Aktuelle referanserammer . . . . .	11
3.1.1	Inertialramme - ( <i>i</i> -ramme) . . . . .	12
3.1.2	Earth-centered earth-fixed - ECEF ( <i>e</i> -ramme) . . . . .	13
3.1.3	Navigasjonsramme - geografisk ramme ( <i>g</i> -ramme) . . . . .	13
3.1.4	Bodyramme ( <i>b</i> -ramme) . . . . .	15
3.1.5	Platform ( <i>p</i> -ramme) . . . . .	15
3.2	Rotasjonsmatriser . . . . .	16
3.2.1	Retnings cosinus matrisen - DCM . . . . .	17
3.2.2	Eulervinkler . . . . .	18
<b>4</b>	<b>Treghetsnavigasjon</b>	<b>20</b>
4.1	Introduksjon . . . . .	20
4.1.1	Inertial Measurement Unit - IMU . . . . .	21
4.2	Global Positioning System - GPS . . . . .	23
4.3	Navigasjonsligninger . . . . .	25
4.3.1	Generelle navigasjonsligninger . . . . .	25

4.3.2	ECEF navigasjonsligninger . . . . .	26
4.4	Kalmanfilter . . . . .	27
<b>5</b>	<b>Forberedelse</b>	<b>30</b>
5.1	Utstyr . . . . .	30
5.2	Programvare . . . . .	31
5.3	Beskrivelse av datasett . . . . .	32
5.3.1	Referansedata . . . . .	32
5.3.2	OBU - datasett . . . . .	33
5.3.3	OBU <sub>SIM</sub> - Simulert OBU . . . . .	35
<b>II</b>	<b>Resultater og analyse</b>	<b>37</b>
<b>6</b>	<b>Introduksjon</b>	<b>38</b>
6.1	Tilfeller . . . . .	40
6.1.1	Tilfelle 1 - Bogstadveien . . . . .	40
6.1.2	Tilfelle 2 - Vika . . . . .	42
<b>7</b>	<b>Integritetsmål</b>	<b>44</b>
7.1	Koordinatbasert integritetsmål . . . . .	45
7.1.1	Forutsetninger . . . . .	46
7.1.2	Optimalt eksempel - konsept . . . . .	46
7.1.3	Test av koordinatbasert integritetsmål . . . . .	49
7.2	Diskusjon . . . . .	54
<b>8</b>	<b>Konklusjon</b>	<b>58</b>
<b>A</b>		<b>62</b>
A.1	Simulering av kodemottakere . . . . .	62
A.1.1	Simulering av kodemottakere . . . . .	62
<b>B</b>	<b>Etablering av platform</b>	<b>66</b>
<b>C</b>	<b>Kalmanfilter-PV (m-kode)</b>	<b>69</b>

# Kapittel 1

## Innledning

Navigasjon og behovet for å finne posisjon og retning har vært viktig helt fra mennesket beveget seg ut på de åpne hav. Det å kunne navigere etter sol og stjerner var avgjørende for å kunne utforske verden. Riktig navigasjon kunne spare en for mange dagers reise, feil navigasjon kunne få fatalt utfall.

I dag er det ikke noe problem å bestemme posisjonen til verken fly, bil eller båt med høy *presisjon*. Det å bestemme en posisjon som er både *presis* og *nøyaktig* er derimot ikke alltid en enkel oppgave. Påliteligheten til posisjoneringssystemer er essensiell, spesielt om det skal brukes i en samfunnsmessig sammenheng.

Posisjoneringssystemer kan i mange tilfeller forbedre eller erstatte eksisterende applikasjoner og system. Veiprisning er et eksempel på en applikasjon som i dag benytter seg av fysiske bomstasjoner, som istedet kunne brukt GNSS-systemer til samme formål. SAVE (SAtelittbasert posisjonering for effektiv VEiprisning) er et forskningsprosjekt som i dag ser på hvordan et slikt, neste generasjons, veiprisningssystem kan realiseres.

Et slikt system må for sluttbruker være like enkelt i bruk som dagens OBU-enheter (on board unit), kjent som Autopassbrikker i Stor-Oslo. Systemet skal være basert på posisjonering med en GNSS/INS enhet, og enheten må møte en rekke krav til blandt annet pris, levetid og størrelse. På



grunn av begrensningene er enheten basert på en rimelig sensor som kan masseproduseres.

I veipricing tar man i dag betalt for bomplasseringer. Med posisjonsinformasjon kan man ta betalt basert på posisjonen til OBU-enheten. Det er da viktig at kvalitetskontrollen i posisjonsberegningen er god, spesielt med tanke på at kundene kun skal betale for steder de faktisk har kjørt. Om kvalitetskontrollen er dårlig kan man få noen tilfeller der kunder betaler for steder de ikke har kjørt. Dette vil si at OBUen beregner en posisjon som er langt vekk fra den sanne posisjonen uten å avdekke det selv, dette kalles falske positiver i statistikken. I veipringsprosjekter kan det være involvert flere millioner biler, det er derfor viktig å minimere antall falske positiver. I denne sammenhengen vil god kontroll på posisjonen sin pålitelighet hindre falske positiver.

I forbindelse med en test av geodetisk posisjoneringsutstyr fikk vi låne en prototype sensor tenkt til veipricing. Sensoren er en rimelig GNSS/INS enhet fra teknologiselskapet Q-Free. Q-Free leverer tjenester og produkter for veipricing og trafikkovervåkning i 17 land over hele verden.

## 1.1 Problemstilling

Denne masteroppgaven har som mål å se på hvilke muligheter og begrensinger som ligger i en rimelig GNSS/INS sensor. Sensoren skal testes i bymiljøer, da det normalt er slike områder som er mest aktuelle for veipricing. Hovedmomentet med oppgaven er å avdekke integriteten til sensoren, og hvordan den aktuelle, og tilsvarende sensorer kan kvalitetskontrollere sin egen posisjon.

Mer spesifikt:

- Lage et kalmanfilter i løs kobling for å teste og illustrere grunnprinsippene i et GNSS/INS-system.
- Prøve alternative metoder for å beregne integriteten til den estimerte

posisjonen.

- Vurdere hvor et slikt system fungerer og hvor det ikke fungerer i urbane miljøer.
- Vurdere eventuelle endringer som kan gjøres for å forbedre påliteligheten.

# Del I

## Teori og bakgrunn

# Kapittel 2

## Bakgrunn

### 2.1 Estimering og kvalitetskontroll

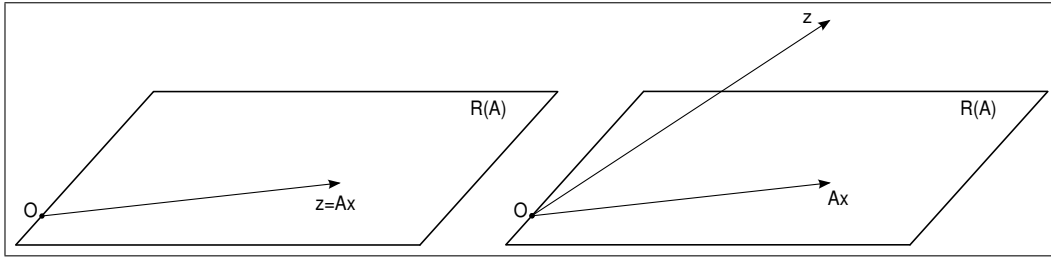
**Observasjonsligning:** Gitt et generelt tilfelle der vi ønsker vi å estimere  $n$ -antall parametere i en *statevektor*  $x$ , med tilhørende  $m$ -antall observasjoner i *observasjonsvektor*  $z$ . Om det er en kjent lineær sammenheng mellom observasjonene og parameterne vi ønsker å estimere, så kan observasjonsligningen skrives på formen [1]:

$$z = Hx \tag{2.1}$$

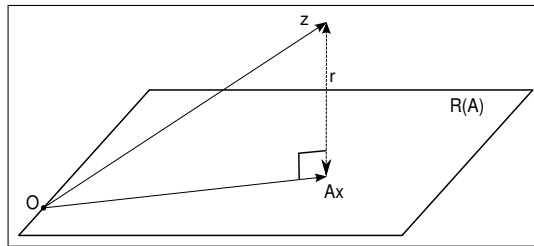
Der  $H$  er en  $m \times n$  designmatrise ( $m \geq n$ ). Løsningen av denne ligningen finnes kun om vektor  $z$  kan skrives som en lineær kombinasjon av kolonnene i matrise  $H$ . Om dette ikke er tilfellet vil systemet bli inkosistent ( $z \neq Ax$ ). Figur 2.1 på neste side viser geometrisk de to tilfellene.

#### 2.1.1 Minste kvadraters metode

Systemet til høyre i figur 2.1 inneholder en feil, enten i observasjonene eller i designmatrisa, som gjør at det blir inkonsistent. Ved å introdusere vektor  $r$



Figur 2.1: Observasjonsmodell - konsistent system til venstre og inkosistent system til høyre.  $R(A)$  er rommet utspent av kolonnevektorene til  $A$



Figur 2.2: Observasjonsmodell - konsistent system ved introdusering av feilvektor  $r$ .

i ligning (2.1), blir systemet konsistent:

$$z = Hx + r \quad (2.2)$$

Som illustrert i figur 2.2.

Konseptet i *minste kvadraters metode* er å minimere  $r$ , slik at avviket mellom  $z$  og  $Ax$  blir minst mulig.  $r$  blir kalt for *residualer*, eller *restfeil*. Videre utledning av minste kvadraters metode er forklart enkelt i [1]. De endelige løsningene til minste kvadraters metode er gitt, fra [1]:

$$\hat{x} = (H^T P_z^{-1} H)^{-1} H^T P_z^{-1} z \quad (2.3)$$

$$\hat{z} = H \hat{x} \quad (2.4)$$

$$\hat{r} = z - \hat{z} \quad (2.5)$$

Der  $P_z$  er kovariansmatrisen til observasjonene,  $\hat{x}$  er estimatet,  $\hat{z}$  er justerte observasjoner, og  $\hat{r}$  er utjevningskorreksjoner.

## 2.2 Presisjon

Presisjon forteller hvor god overenstemmelse det er mellom flere forskjellige målinger/observasjoner. Presisjonen uttrykkes vanligvis som standardavviket til normalfordelingen [2]. Om for eksempel en skarpskytter skyter på blink og har liten spredning mellom alle skuddene som treffer, så vil presisjonen være god, *uavhengig* om han treffer midt i blinken eller ikke. Om det er stor spredning mellom skuddene er presisjonen dårlig. Dette er ekvivalent i geodesi, da presisjonen blir dårlig om det er dårlig overenstemmelse mellom observasjonene, og god presisjon ved god overenstemmelse.

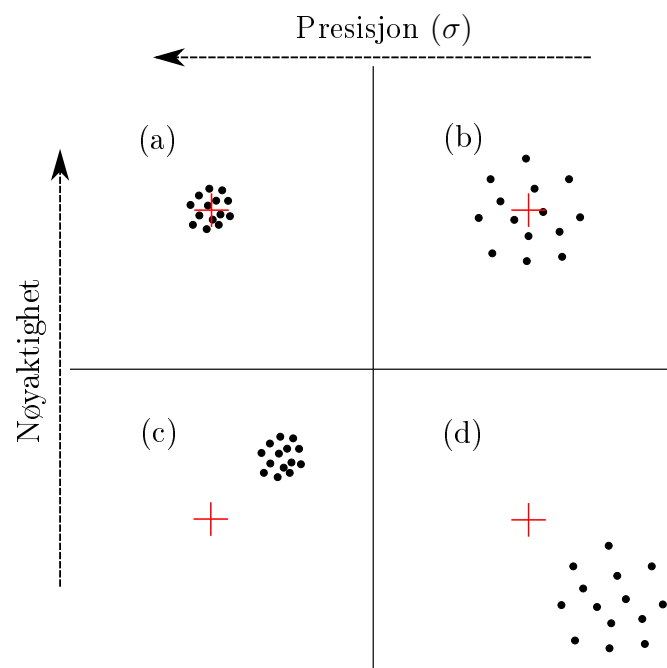
Når man måler ute i felt vet man ikke hvor den sanne posisjonen er, og figur 2.3 (c) viser at det er mulig å være langt unna den sanne posisjonen selvom presisjonen er god. Tilfellet i figur 2.3 (c) skyldes ofte menneskelige eller systematiske feil som oppstilling av basestasjon i feil punkt eller lignende.

## 2.3 Nøyaktighet

Nøyaktighet er graden av overensstemmelse mellom en estimert verdi og den sanne verdien [2]. I eksempelet med en skarpskytter vil nøyaktigheten være god om skytteren treffer *midt* i blinken, og nøyaktigheten dårlig om han treffer langt unna midten. I geodesi vil det si at en estimert posisjon har god nøyaktighet om den er nær den sanne posisjonen. Det er normalt å bruke en bedre og mer pålitelig målemetode for å kontrollere nøyaktigheten til en estimert verdi. I denne oppgaven blir referansedatasettet sett på som den til enhver tid sanne posisjonen til bilen.

## 2.4 Integritet

Integritet forteller om man kan stole på et system eller ikke. I et utdrag fra ordboka er integritet beskrevet som: «*selvstendighet, ukrenkelighet respektere*



Figur 2.3: Forskjellen mellom presisjon og nøyaktighet. De svarte prikkene er observasjoner, og det røde krysset er den sanne posisjonen: (a) God presisjon og god nøyaktighet - (b) Dårlig presisjon og god nøyaktighet - (c) God presisjon og dårlig nøyaktighet - (d) Dårlig presisjon og dårlig presisjon.

*et individs integritet...»* Selvom dette er en mer humanistisk beskrivelse, har det samme betydningen for et posisjoneringssystem.

## 2.5 Pålitelighet

I geodesiden skilles det mellom indre og ytre pålitelighet. Indre pålitelighet sier hvor godt observasjonene i et system gjensidig kontrollerer hverandre [3]. Det vil si at indre pålitelighet forteller hvordan en feil i en observasjon gjenspeiles i den tilhørende utjevningskorreksjon [3]. Ytre pålitelighet er et mål som uttrykker hvor stor virkning en feil i observasjonene har på de ukjente (koordinatene) [3].



# Kapittel 3

## Koordinatsystemer

Observasjoner fra posisjoneringssystemer er som regel gitt i forskjellige referanserammer. Det er viktig at observasjonene blir transformert til en felles og godt egnet referanseramme for prosessering. Dette kapitlet skal beskrive de forskjellige referanserammene som brukes i treghetsnavigasjon samt koblingen mellom dem.

Et treghetsnavigasjonssystem bruker normalt fire eller flere referanserammer. Aksellerometerene og gyroene observerer fra IMUen (sensoren) i forhold til et inertialsystem som beskrevet av Newton. Sensoren sitter som regel i ett fly/bil eller lignende som uttrykkes gjennom bodyramma. For å uttrykke retning og avstand osv på jorda er det behov for en lokalt geodetisk navigasjonsramme. Samt for å beregne geodetiske/kartesiske koordiner på jorda er det behov for ett system med jorda i sentrum.

### 3.1 Aktuelle referanserammer

Alle koordinatsystemer som er beskrevet i oppgaven er ortogonale høyrehåndssystemer og blir i denne delen beskrevet i enkelhet med tilhørende notasjon. Den geodetiske bruken av begrepet referanse *system*, og referanse *ramme* blir fulgt. I korte trekk inkluderer *systemet* fysiske teorier, eller til-

nærminger og modeller, som blandt annet geoider, som brukes for å definere koordinataksene; mens en *ramme* brukes om et tilgjengelig system som er realisert gjennom et sett med punkter som er observerbare. Wolfgang Torge beskriver dette inngående i Geodesy (2001) [4], en mer oversiktlig og brukervennlig tilnærming finnes i Aided Navigations [5].

### 3.1.1 Inertialramme - (*i*-ramme)

En inertial referanseramme er en referanseramme der Newtons lover gjelder. Det er derfor en ikke roterende og ikke aksellererende ramme som er i ro, eller lineær konstant bevegelse. Et celest referansesystem er den beste tilnærmingen til en inertial ramme og defineres ved konvensjoner [4]. Origo i ramma er vilkårlig og aksene står vinkelrett på hverandre, men kan peke i vilkårlig retning. Det er to konkurrerende definisjoner av et ideelt celest referansesystem, men begge fører til en ikke-roterende referanseramme [6]. Alle inertielle sensorer som aksellerometere og gyroer måler relativt til en ideel inertial ramme.

#### Earth Centered Inertial - ECI

Av praktiske årsaker blir det definert en inertielt ECI ramme som har origo i jordens massemiddelpunkt slik som ECEF i delkapittel 3.1.2 på neste side. ECI og ECEF deler Z-akse som går gjennom den midlere nordpolen (rotasjonsaksen til jorden). X-aksen peker fast mot vårjevndøgnsunktet [5]. Y-aksen kompletterer høyrehåndssystemet og står ortogonalt på de andre aksene.

ECEF og ECI har sammenfallende X- og Y-akse ved vårjevndøgn, og hver hele jordrotasjon før og etter vårjevndøgn. Det vil si at ECI ligger fast relativt stjernene, se også figur 3.1 på side 14 for relasjonen mellom ECEF og ECI. Inertial ramme blir notert « *i* ».

### 3.1.2 Earth-centered earth-fixed - ECEF (*e*-ramme)

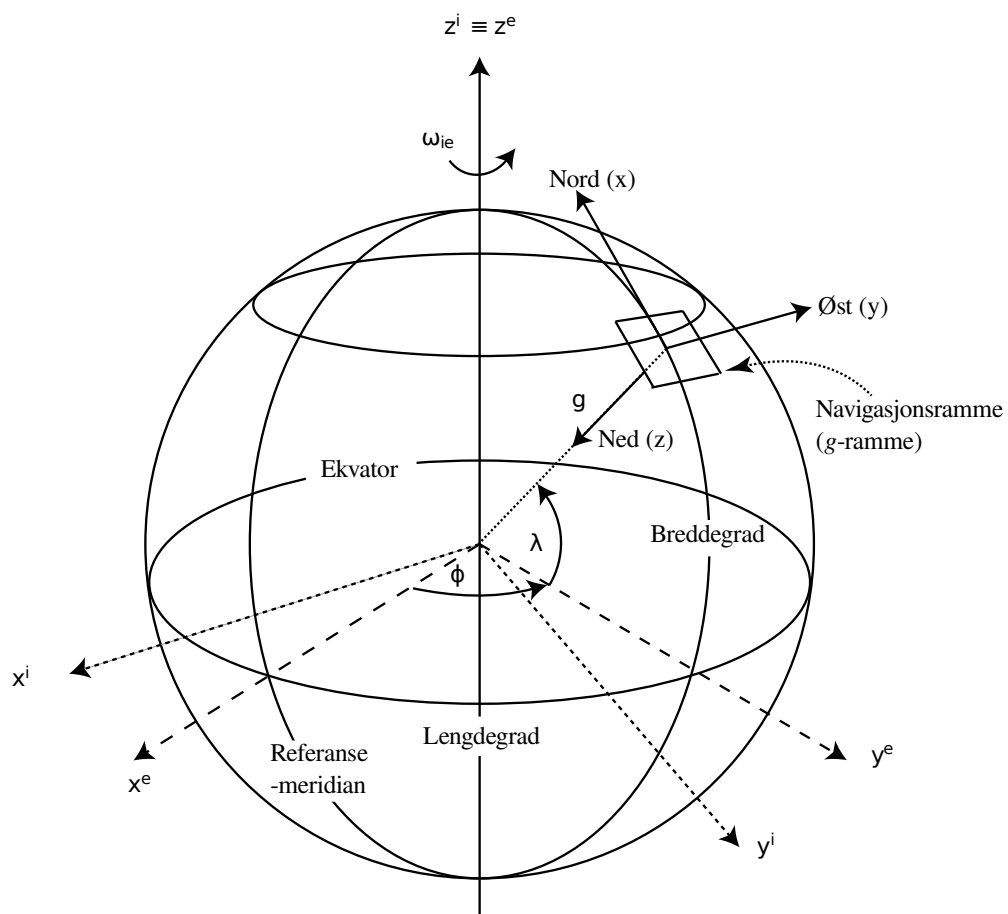
ECEF har origo i jordens masse-middelpunkt (inkludert atmosfæren), og roterer med jorden. Referansesystemet er realisert av International Earth Rotation Service (IERS) gjennom International Terrestrial Reference Frame (ITRF) som er koblet til en inertiel/celestisk referanseramme (ICRF) [4] som er ECI i delkapittel 3.1.1 på forrige side. Systemet har kartesiske koordinater  $X, Y, Z$  som i figur 3.1 på neste side.  $Z$ -aksen ligger sammen med spinnaksen til jorden og har retning opp gjennom nordpolen (middel-pol), og  $X$ - og  $Y$ -aksene ligger vinkelrett på og danner middel-ekvatorplanet.  $X$ - $Z$  -planet går gjennom Greenwich referansemeridian, og  $Y$ -aksen danner et komplett høyrehånds system [4]. Jordsentrisk ramme blir notert med «  $e$  ».

Jorda roterer 365,25 ganger på ett år i tillegg til ett omløp rundt sola. Rotasjonsfrekvensen til ECEF systemet med hensyn på ECI er  $\omega_{ie} \approx 7.292115 \times 10^{-5} \frac{rad}{sec}$ , [5]. Rotasjonen mellom ECI og ECEF, parameterisert i ECI systemet blir da

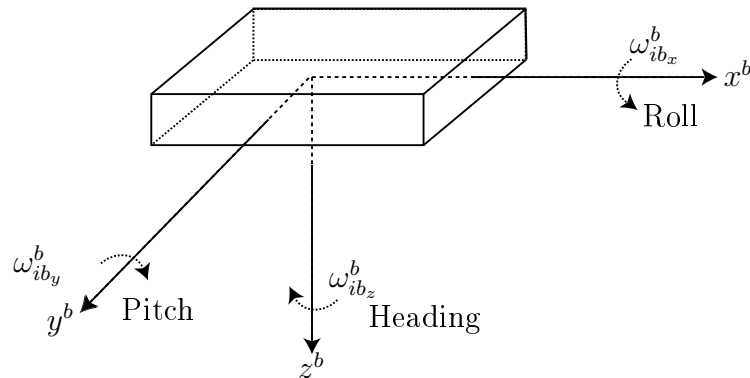
$$\omega_{ie}^i = \begin{bmatrix} 0 \\ 0 \\ \omega_{ie} \end{bmatrix} \quad (3.1)$$

### 3.1.3 Navigasjonsramme - geografisk ramme (*g*-ramme)

Navigasjonsrammen er definert lokalt relativt jordens geoide. Origo til ramma beveger seg med systemet og er projeksjonen av platformens (sensorens) origo ned på referanseellipsoiden [5].  $Z$ -aksen peker langs ellipsoidenormalen,  $X$ -aksen peker mot nordpolen (middel-pol) og  $Y$ -aksen mot øst i høyrehåndssystemet. Siden navigasjonsrammens origo flytter seg sammen med sensoren så roterer også aksene ettersom sensoren går østover og nordover.



Figur 3.1: Koblingen mellom ECEF (e-ramme), navigasjonsramme (g-ramme) og inertial (i-ramme) [7].



Figur 3.2: Bodyramme.

### 3.1.4 Bodyramme (*b*-ramme)

Bodyramma er referanseramma til kjøretøyet, der origo normalt sitter i kjøretøyets massemiddepunkt. Aksene til koordinatsystemet er ikke nødvendigvis unike, men det er vanlig å bruke et ortogonalt høyrehåndssystem der X-aksen peker i kjøretøyets fartsretning, Z-aksen rett ned, og Y-aksen til høyre. Rotasjonene om de forskjellige bodyaksene har ofte forskjellige navn i forskjellige miljøer (fly, helikopter o.l.), men notasjonen som brukes videre i denne oppgaven er: roll om X-aksen, pitch om Y-aksen, og heading om Z-aksen. Rotasjonsvektoren til bodyrammen i forhold til inertialrammen, gitt i bodyrammen blir da betegnet  $\omega_{ib}^b = \begin{bmatrix} \text{Roll} \\ \text{Pitch} \\ \text{Heading} \end{bmatrix}$

### 3.1.5 Platform (*p*-ramme)

Platformrammen blir dannet for å koble sammen forskjellige sensorer i en felles ramme. Platformrammen kan i mange tilfeller sammenfalle med *b*-rammen. For den aktuelle kjøretøreturen var det behov for en optimal platformramme for i hovedsak å gi Terrapos riktig relative koordinater mellom antenne og IMU. I tillegg til prosesseringen av selve referansen, så har Terrapos mulighet til å estimere koordinater for andre enheter koblet til platformen gjennom en monteringsfil. Figur 3.3 viser målebilen, med ferdig innmålte punkter. Platformrammen ble dannet fotogrametrisk i programmet



Figur 3.3: Plattform - målt fotogrametrisk.

Photo Modeler, med standardavvik på under 1 mm for de aktuelle sensorene. Rapporten ligger i tillegg B.

## 3.2 Rotasjonsmatriser

Det er flere måter å transformere koordinater fra en ramme til en annen. Referanserammene som betraktes i denne oppgaven er ortogonale, og for relativ orientering er det derfor nok å beskrive de med tre rotasjoner. Den relative orienteringen av koordinater er normalt formålet med transformasjonen. Det er i tillegg mulig å flytte origo ved å legge en vektor med koordinatdifferanser til koordinatene. Dette delkapittelet utleder retnings-cosinus-matrisen og euler vinkler. Funksjonen til rotasjonsmatrisene er viktig i treghetsnavigasjon da det er involvert flere forskjellige referanserammer. Christopher Jekeli [8] utleder og beskriver rotasjonsmatrisene utfyllende. Jekeli [8] utdyper også kvaternioner, som er en alternativ og mye brukt metode for å rotere koordinatrammer.

### 3.2.1 Retnings cosinus matrisen - DCM

Fremgangsmåten for å utlede retnings cosinus matrisen er basert på Jekeli [8]. Retnings cosinus matrisen forkortes normalt med DCM (Direction Cosine Matrix). Gitt to ortogonale og konsentriske koordinatrammer  $a$  og  $b$ , med forskjellig relativ orientering. Koordinatene til et punkt,  $x$ , i  $a$ -rammen blir representert med vektor  $\mathbf{x}^a$ , og, vektor  $\mathbf{x}^b$  i  $b$ -rammen. Koordinatene til et punkt er vektoren spent ut over de respektive aksene i rammen med enhetsvektorer,  $\mathbf{e}_i$ , som for henholdsvis  $a$ - og  $b$ -rammene blir:

$$\mathbf{x}^a = x_1^a \mathbf{e}_1^a + x_2^a \mathbf{e}_2^a + x_3^a \mathbf{e}_3^a, \quad (3.2)$$

$$\mathbf{x}^b = x_1^b \mathbf{e}_1^b + x_2^b \mathbf{e}_2^b + x_3^b \mathbf{e}_3^b, \quad (3.3)$$

der  $x_j^a = \mathbf{e}_j^a \cdot \mathbf{x}^a$  er koordinat  $j = 1, 2, 3$  i  $a$ -ramma og  $\mathbf{e}_j^a$  er tilsvarende enhetsvektor. Tilsvarende er koordinatene til enhetsvektor  $\mathbf{e}_k^b$ , ( $k = 1, 2, 3$ ) gitt i  $a$ -rammen

$$c_{j,k} = \mathbf{e}_j^a \cdot \mathbf{e}_k^b, \quad (3.4)$$

som gir koordinat:

$$\mathbf{e}_k^b = c_{1,k} \mathbf{e}_1^a + c_{2,k} \mathbf{e}_2^a + c_{3,k} \mathbf{e}_3^a. \quad (3.5)$$

Hvis vi nå setter (3.5) for hver  $k$  in i (3.3) og sammenligner med (3.2) har vi transformasjonen mellom koordinatene [8], som skrevet på vektorform blir:

$$\mathbf{x}^a = C_b^a \mathbf{x}^b \quad (3.6)$$

der transformasjonsmatrisen er gitt ved:

$$C_b^a = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix} \quad (3.7)$$

Fra egenskapen til prikkprodukt er koeffisienten  $c_{j,k}$ , fra ligning (3.4), cosinus til vinkelen mellom akse  $j$  i  $a$ -rammen og akse  $k$  i  $b$ -rammen.  $C_b^a$  kalles derfor

en retnings cosinus matrise.  $C_b^a$  er en ortogonal matrise, det vil si at:

$$C_b^a(C_b^a)^\top = I \quad \text{og,} \quad (C_b^a)^\top = (C_b^a)^{-1}$$

### 3.2.2 Eulervinkler

I følge Eulers rotasjonsteorem kan alle rotasjoner beskrives med tre vinkler [9]. Et alternativ til å bruke cosinus rotasjons matrise er derfor å rotere i en gitt rekkefølge om aksene. Rotasjonsmatrisene for hver akse er gitt [8]:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (3.8)$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3.9)$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

$R_x(\theta)$  er rotasjonen om  $x$ -aksen med vinkel  $\theta$ , og tilsvarende for de andre aksene. Hver rotasjonsmatrise er i prinsippet ett spesialtilfelle av en retnings cosinus matrise. Rekkefølgen på rotasjonene kan ikke være tilfeldig, da:  $R_x(\alpha)R_y(\beta) \neq R_y(\beta)R_x(\alpha)$ . Mer om rotasjonsmatrisene og egenskapen til Euler vinklene kan studeres i [9, 8].

Den endelige rotasjonsmatrisen som roterer et høyrehåndssystem:



$$\begin{aligned}
C_b^a &= R_z(\gamma)R_y(\beta)R_x(\alpha) \dots \\
&= \begin{bmatrix} \cos \beta \cos \gamma & \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma & \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma \\ -\cos \beta \sin \gamma & \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma & \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma \\ \sin \beta & -\sin \alpha \cos \beta & \cos \alpha \cos \beta \end{bmatrix}
\end{aligned} \tag{3.11}$$

For den gitte transformasjonen av en vektor må resultatet av å bruke retnings cosinuser eller Euler vinkler, til slutt bli det samme. I tilfellet med Euler rotasjonene gitt som i (3.11) er retnings cosinusene,  $c_{j,k}$ , kun elementene i matrise (3.11), [8]. Videre, om retnings cosinusene er gitt så er Euler vinklene i dette tilfellet gitt av:

$$\alpha = \tan^{-1} \left( \frac{-c_{3,2}}{c_{3,3}} \right), \quad \beta = \sin^{-1} (c_{3,1}), \quad \gamma = \tan^{-1} \left( \frac{-c_{2,1}}{c_{1,1}} \right) \tag{3.12}$$

# Kapittel 4

## Treghetsnavigasjon

Treghetsnavigasjonssystemer blir i dag brukt, i stor eller liten grad, i alt fra store fly til biler og mobiltelefoner. Treghetsnavigasjon er spesielt viktig i miljøer der sikkerhet og pålitelighet er av høyeste orden. Eksempler på dette er sivil og militær luftfart, og ubåter som ikke kan motta signaler fra GNSS eller radiosystemer under vann. Det første eksempelet på bruk av treghetsnavigasjon var av forskeren Robert Hutchings Goddard som i 1932 skjøt opp en gyrostabilisert rakett [2]. Systemet ble viderutviklet til bruk i fly av Charles Stark Draper i 1945, og senere også ubåter og romferger [2]. Treghetsnavigasjonssystemer blir normalt forkortet med den engelske forkortelsen INS, som står for Inertial Navigation System.

### 4.1 Introduksjon

Treghetsnavigasjon er i enkleste definisjon et system som måler aksellerasjon og retningsendring, og bruker det til å beregne posisjon, fart og retning på bakgrunn av Newtons lover. Newtons lover, bevegelsesmengde og bevaringslovene er viktige begreper, og de er kun gjeldende innen en inertialramme som beskrevet i delkapittel 3.1.1 på side 12. Sensoren som måler er derfor kalt «*Inertial Measurement Unit*», IMU. Det er et logisk og

Tabell 4.1: Sammenligning INS - GNSS [8]

INS	GNSS
Observerer: Akselerasjon/Vinkelhastighet God korttidsnøyaktighet Avhengig av initiell orientering Høy datarate (50-500Hz)	Observerer: Pseudoavstand God langtidsnøyaktighet Avhengig av satelittgeometri Lav rate (1-10Hz)

viktig navn da sensorene kun måler relativt inertialrammen. I motsetning til GNSS er INS uavhengig av ekstern infrastruktur, noe som blant annet gjør at det kan brukes steder GNSS-signal ikke når fram. De to systemene utfyller hverandre godt og kan sier å være komplimentære, se tabell 4.1.

#### 4.1.1 Inertial Measurement Unit - IMU

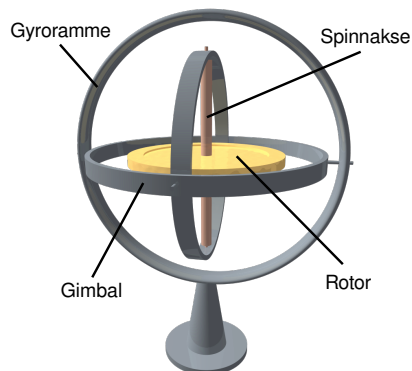
En IMU består normalt av tre akselerometere som måler lineær akselerasjon langs tre akser, og et gyroskop som måler vinkelhastighet om de samme tre aksene. Disse tre aksene er (i teorien) ortogonale, og danner instrumentrammen. Det finnes flere typer IMUer, og de graderes i klasser etter ytelse som i tabell 4.2.

#### Gyroskop

Gyroskop (gyro i kort form), er i prinsippet et svinghjul som har stort treghetsmoment om akse sin. Når gyroen spinner fort rundt vil rotoraksen beholde retningen i forhold til  $i$ -rammen [2]. Det vil se ut som at gyroskopet

Tabell 4.2: IMU klassifisering - teknologi og ytelse. Utdrag fra K.Gade [10]

Class	Position performance	Gyro bias	Acc. bias
Military grade	1 nmi/24h	$< 0.005^\circ/\text{h}$	$< 30 \mu\text{g}$
Navigation grade	1 nmi/h	$0.01^\circ/\text{h}$	$50 \mu\text{g}$
Tactical grade	$> 10$ nmi/h	$1^\circ/\text{h}$	1 mg
AHRS (Mems)	N/A	$1-10^\circ/\text{h}$	1 mg
Control system (Mems)	N/A	$10-1000^\circ/\text{h}$	10 mg



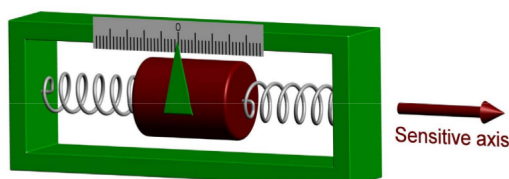
Figur 4.1: Tradisjonelt gyroskop i gimbal oppheng [11]

står stille i forhold til stjernene. Med en presis gyro kan man i prinsippet måle rotasjonen til jorda.

Dagens høypresisjonsgyroer er ofte ikke virkelige gyroskoper da de bruker lys som måleelement istedet for en spinnende masse. Lys har ikke masse, og blir derfor ikke utsatt for de samme kreftene, men de optiske gyroene har fortsatt den samme koblingen til  $i$ -rammen. Optiske gyroer er basert på *Sagnac effekten* [12], som oppstår når en lysståle går rundt i en lukket sløyfe i en ramme som roterer i forhold til inertialrammen.

### Akselerometer

Akselerometer er et utstyr som måler akselerasjon [2]. Ved å feste en testmasse til et fjæroppheng vil testmassen bevege seg om den utsettet for en kraftpåvirkning. Ved å måle bevegelsen til massen får vi et enkelt akselerometer som i figur 4.2 på neste side. Akselerometer som brukes til navigasjon er normalt tre forskjellige akselerometere som måler langs hver sin akse. Det finnes flere typer akselerometere og de har mange bruksområder. I dag brukes akselerometere i alt fra mobiltelefoner, til tyngdemåling og navigering av fly og raketter.



Figur 4.2: Akselerometer[10]

## 4.2 Global Positioning System - GPS

GPS er et Amerikansk satellittbasert navigasjonssystem. Det ble operativt i 1993, men utviklingen startet på slutten av 1960-tallet [2]. GPS ble utviklet som et rent militært system, men i dag er over 90% av brukerne sivile [2].

Satellittene går i bane 20 200 km over jordoverflaten. Ved å måle hvor lang tid signalet bruker fra satellitt til mottaker kan mottakerposisjonen bestemmes. Det trengs 4 satellitter for å bestemme en romlig posisjon samt tidsavvik i mottakeren. På grunn av relasjon til oppgaven vil observasjonsligningen for en kodemottaker beskrives. GPS eller GNSS systemer omtales ikke videre teoretisk, men for videre lesing kan Satellite Geodesy av Seeber [13] anbefales.

GPS-satellittene sender to pseudorandom-koder modulert på bærebølgene L1 og L2 [14]:

Kode	Bitlengde	$\sigma$	Bærebølge
C/A	$\approx 300m$	$\approx 3m$	L1
P	$\approx 30m$	$\approx 0.3m$	L1 og L2

Om observasjonene er gjort ved bruk av C/A- eller P-kode har ikke noen betydning for observasjonsligningene nedenfor [14]. P-koden er kryptert og forbeholdt militære formål, den er derfor normalt ikke en tilgjengelig for sivile brukere.

### Observasjonsligning

Ligninger og notasjon for GPS kodemåling er hentet fra Øvstedal [14]. Den fundamentale observasjonsligningen for en enkel pseudoavstand  $R_i^j$  mellom

mottaker  $i$  (subscript) og satellitt  $j$  (superscript) er:

$$R_i^j = \overbrace{\rho_i^j} + cdT_i \quad (4.1)$$

$$R_i^j = \sqrt{(X^j - X_i)^2 + (Y^j - Y_i)^2 + (Z^j - Z_i)^2} + cdT_i \quad (4.2)$$

### Notasjoner:

$\rho_i^j$  - den geometriske avstanden mellom mottaker og satellitt (meter).

$c$  - Lysets hastighet ( $m/s$ ).

$R_i^j$  - kode-observasjonen (pseudavstand) mellom satellitt $_j$  og mottaker $_i$ .

$X_i, Y_i, Z_i$  - ukjente koordinater til mottaker $_i$ .

$X^j, Y^j, Z^j$  - kjente koordinater til satellitt $_j$ .

$dT_i$  - differansen mellom mottaker $_i$  sin klokke og GPS-systemtid.

$dT^j$  - differansen mellom satellitt $_j$  sin klokke og GPS-systemtid.

$\Delta t_i^j = dT_i - dT^j$  - klokkeforskjell mellom mottaker og satellitt (sek).

$\Delta R_i^j_{Iono}$  - ionosfærisk korreksjon (meter).

$\Delta R_i^j_{Trop}$  - troposfærisk korreksjon (meter).

$\Delta T_i^j_{Rel}$  - Relativistisk korreksjon (sek).

$v_i^j$  - målestøy, og umodellerte effekter (meter).

Pseudoavstanden bestemmes ved å måle tidsforskjellen mellom sendt og mottatt signal [13].

$$R_i^j = c(T_i^{\text{mottatt}} - T^j_{\text{sendt}}) \quad (4.3)$$

Der  $T$  er tidspunkt for henholdsvis sendt og mottatt signal.

For en kodemåling er den endelige observasjonsligningen gitt [13, 14]:

$$R_i^j_{kode} = \rho_i^j + c\Delta t_i^j + \Delta R_i^j_{Iono} + \Delta R_i^j_{Trop} + c\Delta T_i^j_{Rel} + v_i^j \quad (4.4)$$

## 4.3 Navigasjonsligninger

Treghetsnavigasjon er basert på integrasjon av aksellerasjoner med hensyn på tid. Aksellerasjonene er målt i forhold til et inertielt system. Newtons andre lov sier at summen av kreftene som virker på et legeme er lik produktet av legemets masse og dets akselerasjon, og at akselerasjonen har samme retning som summen av kreftene [2]. Gitt newtons lover, en posisjonsvektor  $\mathbf{x}^i$ , gravitasjons aksellerasjon  $\mathbf{g}^i$  og spesifikk kraft  $\mathbf{f}^i$  kan vi avlede en ligning for aksellerasjon. Gitt for en inertial ramme [8]:

$$\ddot{\mathbf{x}}^i = \mathbf{g}^i(\mathbf{x}^i) + \mathbf{f}^i. \quad (4.5)$$

$\mathbf{g}^i$  er avhengig av posisjonsvektoren  $\mathbf{x}$ , som sammen med  $\mathbf{f}^i$  er observert av akselerometeret. Med kjent gravitasjon kan ligningen løses for posisjon  $\mathbf{x}^i$ . Tilsvarende differensialligninger kan lages for andre rammer. Integrering av posisjoner og fart ved integrering av disse ligningene er kjent som *free-inertial navigation* [8]. Utenom initielle betingelser er free-inertial navigation uavhengig av ekstern informasjon.

### 4.3.1 Generelle navigasjonsligninger

Utleddningen at navigasjonsligningene følger Jekeli [8]. Gitt posisjonsvektor  $\mathbf{x}^a$ , i en vilkårlig ramme  $a$ . Denne vektoren kan uttrykkes i inertialramma ved å rotere  $\mathbf{x}^a$  med en DCM-matrise,

$$\mathbf{x}^i = C_a^i \mathbf{x}^a. \quad (4.6)$$

Der  $C_a^i$  er DCM-matrisen (se ligning 3.6). Ved derivere ligning (4.6) to ganger med hensyn på tid, kan resultatet brukes til å substituere  $\ddot{\mathbf{x}}^i$  i ligning (4.5). Den tidsderiverte av  $C_a^i$  er gitt av:

$$\dot{C}_a^i = C_a^i \Omega_{ia}^a, \quad (4.7)$$

Der  $\Omega_{ia}^a$  er en skjevsymmetrisk matrise (4.10) med elementene fra  $\omega_{ia}^a$ . Det vil si at det er gyroen som måler endringen av  $C_a^i$ .

Utfyllende utledning kan følges i [8], der de endelige observasjonsligningene, gitt i en vilkårlig ramme  $a$  er :

$$\frac{d}{dt}\dot{\mathbf{x}}^a = -2\Omega_{ia}^a\dot{\mathbf{x}}^a - (\dot{\Omega}_{ia}^a + \Omega_{ia}^a\Omega_{ia}^a)\mathbf{x}^a + \mathbf{f}^a + \mathbf{g}^a , \quad (4.8)$$

$$\frac{d}{dt}\mathbf{x}^a = \dot{\mathbf{x}}^a . \quad (4.9)$$

Den skjevsymmetrisk matrisen er en matrise som representerer kryssproduktet, og kan for en vektor (for eksempel  $\omega$ ) skrives som:

$$\Omega = (\omega_{(3,1)} \times) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_3 & 0 \end{bmatrix} \quad (4.10)$$

Hvis sensoren er stabilisert slik at den alltid står parallelt med  $a$ -rammen er de målte aksellerasjonene  $\mathbf{f}^a$ . I skrofast (strapdown) mekanisering er disse aksellerasjonene målt i  $b$ -rammen:

$$\mathbf{f}^a = C_b^a \mathbf{f}^b . \quad (4.11)$$

$C_b^a$  blir bestemt ved integrering av vinkelratene  $\omega_{ab}^b$ , gitt fra gyroen etter ligning:

$$\omega_{ab}^b = \omega_{ib}^b - C_a^b \omega_{ia}^a \quad [8]. \quad (4.12)$$

### 4.3.2 ECEF navigasjonsligninger

Hvis vi omdefinerer  $a$ -ramma til å bli  $e$ -ramme får vi fra ligning (4.8) og (4.9):

$$\frac{d}{dt}\dot{\mathbf{x}}^e = -2\Omega_{ie}^e\dot{\mathbf{x}}^e - \Omega_{ie}^e\Omega_{ie}^e\mathbf{x}^e + \mathbf{f}^e + \mathbf{g}^e , \quad (4.13)$$

$$\frac{d}{dt}\mathbf{x}^e = \dot{\mathbf{x}}^e . \quad (4.14)$$



$\Omega_{ie}^e$  er kjent fra (3.1).  $\dot{\Omega}_{ie}^e = 0$  slik at dette leddet faller bort [8]. Den tidsderiverte,  $\dot{C}_b^e$ , av DCM matrisen kan vises [8] å bli:

$$\dot{C}_b^e = C_b^e \Omega_{eb}^b, \quad (4.15)$$

og observerte vinkelrater,  $\omega_{ib}^b$ , blir:

$$\omega_{eb}^b = \omega_{ib}^b - C_e^b \omega_{ie}^e \quad [8]. \quad (4.16)$$

Endelig observasjonsligning for akselerasjon

$$\mathbf{f}_{ib}^b - \tilde{\mathbf{f}}_{ib}^b = \tilde{C}_e^b \Delta a_{ib}^e + 2\tilde{C}_e^b \Omega_{ie}^e \Delta v_{ib}^e + (\tilde{f}_{ib}^b \times) \Delta \Psi_{ib}^e \quad (4.17)$$

Der  $\Psi$  er tiltfeil. De endelige observasjonsligningene er satt skjematisk opp i tabell 4.3.

Tabell 4.3: Observasjonsligninger

$\Delta p$	$\Delta v$	$\Delta a$	$\Delta \Psi$	$\Delta b$	$\Delta \omega$	<i>obs - foreløpig</i>
$I_{3 \times 3}$	$2\tilde{C}_e^b \Omega_{ie}^e$	$\tilde{C}_e^b$	$(\tilde{f}_{ib}^b \times)$	$I_{3 \times 3}$	$I_{3 \times 3}$	$p - \tilde{p}$ $f_{ib}^b - \tilde{f}_{ib}^b$ $\omega_{ib}^b - \tilde{\omega}_{ib}^b$

## 4.4 Kalmanfilter

Kalmanfilteret ble introdusert av R.E.Kalman i 1960, i artikkelen «A New Approach to Linear Filtering and Prediction Problems» [15]. Kalmanfilter er en rekursiv algoritme som løser minste kvadraters metode, med lineært drevet systemstøy. Kalmanfilteret kan leses mer om i [16] og [5].

I denne oppgaven ble det laget et 12-*state* filter. Observasjonene som ble inkludert i filteret var koordinater, og vinkelrater. Akselerometerene ble ikke tatt med som observasjoner da det var en monteringsfeil i OBUen (se delkapittel 5.3.2).

Observasjonene er koblet til *state vektoren* ved funksjonen

$$z = Hx + r \quad (4.18)$$

I et system som er beskrevet av en første ordens differensialligning

$$\dot{x}(t) = F(t)x(t) + G(t)u(t). \quad (4.19)$$

Med statevektor, for henholdsvis x, y og z-aksen:

$$x = [p_x, v_x, \Psi_x, \omega_x, p_y, v_y, \Psi_y, \omega_y, p_z, v_z, \Psi_z, \omega_z] \quad (4.20)$$

Matrisene videre er gitt for en akse for å det oversiktlig, da aksene i e-ramma er helt like i alle tre dimensjoner. Kronecker produktet kan brukes for å spre matrisene ut slik at de dekker tre dimensjoner. Etter tabell 4.3 blir observasjonene koblet til statevektoren med desingmatrise  $H$ , og observasjonsvektor  $z$ :

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad z = \begin{bmatrix} p_x - \tilde{p}_x \\ \omega_x - \tilde{\omega}_x \end{bmatrix} \quad (4.21)$$

## Dynamisk modell

For å beskrive den dynamiske oppførselen til bilen ble det valgt en PV-modell. Hovedforskjellen mellom en PV- og PVA-modell er at PVA-modellen kan tillatte større aksellerasjon og raskere svinger. PV-modellen ble valgt fordi den aktuelle kjøreturen ikke hadde noen hendelser som skulle tilsi at en PVA-modell var nødvendig. I tillegg kom GNSS-koordinater på 1Hz, slik at den dynamiske modellen aldri hadde behov for å drive langt uten måleoppdatering.

Tabell 4.4: Varians og støybidrag

$q_v$	$1m/s$
$q_\omega$	$0.01rad/s$
$\sigma_p^2$	$1m$
$\sigma_\omega^2$	$0.03rad/s$

Transisjonsmatrisen for en akse:

$$\Phi = \begin{matrix} & p & v & \Psi & \omega \\ \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} & & & & \end{matrix} \quad \begin{matrix} (4.22) \\ (4.23) \end{matrix}$$

### Støymodell

Det ble valgt prosesstøy bestående av kun hvitstøy som gir

$$Q_k = \begin{bmatrix} q_v \frac{\Delta t^3}{3} & q_v \frac{\Delta t^2}{2} & 0 & 0 \\ q_v \frac{\Delta t^2}{2} & q_v \Delta t & 0 & 0 \\ 0 & 0 & q_\omega \frac{\Delta t^3}{3} & q_\omega \frac{\Delta t^2}{2} \\ 0 & 0 & q_\omega \frac{\Delta t^2}{2} & q_\omega \Delta t \end{bmatrix}, \quad (4.24)$$

med målestøy

$$R = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix}. \quad (4.25)$$

Tabell4.4 viser hvilke verdier som ble valgt for å modellere støyprosessen. Det ble valgt slik at filteret fulgte observasjonene i større grad enn den dynamiske modellen. Grunnen til dette var at for oppgavens formål var det mer interessant å vise store endringer i satellitt geometri, og hvordan dette påvirker koordinatene. M-koden for PV-filteret ligger i tillegg C på side 69.

# Kapittel 5

## Forberedelse

### 5.1 Utstyr

Måleutstyret som skulle testes var kodemottakere montert i frontruta på en bil, samt geodetiske mottakere og en navigation grade IMU. De geodetiske mottakerene sammen med navigation grade IMUen skulle danne en sann referanse for kodemottakerene. I tillegg er det mange interessante aspekter å studere ved en slik høy-presisjons referanse, uten å gå nærmere inn på det.

#### Referanseutstyr

Det ble logget GPS og GLONASS på 1Hz med tre forskjellige antenner montert på platformen. De geodetiske antennene og mottakerene som ble brukt var fra Geomatikkseksjonen på UMB.

#### Antenner og mottakere:

- Topcon Legant-2 antenne: 2 stykk
- Topcon PG-A1 antenne: 1 stykk
- Topcon mottaker: 3 stykk

Grunnen til at det ble logget med tre GNSS antenner var for å hindre at datainnsamlingen skulle bli ødelagt av utstysrfeil eller menneskelige feil. Ved å gjøre dette var det også mulig å kontrollere datasettet for eventuelle feil.

IMUen som ble brukt var en navigation grade IMU, som betyr at den er laget for navigasjonsbruk.

**IMU:**

- Honeywell HG9900 IMU [17].

Mottakere og loggeutstyr for IMU ble lagret i lasterommet som i figur 5.1-b, Honeywell HG9900 IMU og de tre geodetiske antennene ble montert på en rigid stålplattform på taket.

## **Annet utstyr**

Annet utstyr som ble satt opp i bilen for testing var:

- Garmin kodemottaker m/EGNOS
- OBU, IMU og GNSS kodemottaker - lånt fra Q-Free
- Videokamera

Kodemottakerene og videokameraet ble plassert i frontruta på bilen. Videokameraet ble tatt med for å enkelt kunne se tilbake på eventuelle hendelser som ikke ble loggført.

## **5.2 Programvare**

Til beregning av referansedata, og til noen simuleringer ble Terrapos brukt. Terrapos er en *Precise Point Positioning* (PPP) programvare som etterprosesserer og integrerer INS- og GNSS-data. Ved siden av Terrapos ble det programmert egne rutiner i Matlab for å utlede resultater og tegne plott.



(a) Frontrute med videokamera, Garmin mottaker, OBU, og datamaskin for logging fra enhetene. (b) Lasterom med mottakerutstyr for antenner og HG9900 IMU.

Figur 5.1: Målebil med mottakerutstyr i lasterommet, og IMU og antenner på taket.

## 5.3 Beskrivelse av datasett

Det totale datasettet fra både referansen, OBU og Garmin mottakeren er omfattende og inneholder store mengder data. Hvilke observasjoner, fra hvilket datasett osv er beskrevet på de aktuelle stedene i oppgaven. Her følger en generell beskrivelse av datasettet for å gi et innblikk i prosessering, samt kjørerute og tidsrom.

### 5.3.1 Referansedata

Datasettet for referansen ble etterprosessert i Terrapoint i tett kobling mellom mottaker-b (rec-b) og HG9900 IMU. Kjøreturen varte i omtrent 3,5 time, og er vist i figur 5.2. Start og slutt i referansedataene er:

**Start:** 6. september 2011: kl. 08.22.29 GPS-tid

**Slutt:** 6. september 2011: kl. 12.01.30 GPS-tid

### Sammendrag av ytelsen til referansedataene

-----  
 NAVIGATION FILE SUMMARY  
 -----

```

First record      : 2011 09 06 08 22 29.0
Last record      : 2011 09 06 12 01 30.0
Average time increment : 0.003 s (300.0 Hz)
No. recs. with valid pos. : 3942373
No. recs. with valid vel. : 3942373
No. recs. with valid att. : 3942373

```

-----  
NAVIGATION PERFORMANCE SUMMARY  
-----

		Min.	1%	5%	50%	95%	99%	Max.	
Pos. hor.	std. dev.	0.026	0.026	0.027	0.098	0.277	0.658	0.709	[ m ]
Pos. vert.	std. dev.	0.020	0.020	0.021	0.096	0.321	0.642	0.759	[ m ]
Vel. hor.	std. dev.	0.001	0.001	0.001	0.001	0.006	0.008	0.023	[m/s]
Vel. vert.	std. dev.	0.000	0.000	0.000	0.001	0.004	0.007	0.013	[m/s]
Roll	std. dev.	0.000	0.001	0.001	0.001	0.002	0.003	0.004	[deg]
Pitch	std. dev.	0.000	0.001	0.001	0.001	0.002	0.003	0.004	[deg]
Heading	std. dev.	0.003	0.003	0.003	0.003	0.004	0.006	0.007	[deg]

-----

Referansedatasettet sees på som *fasit* med sanne posisjoner i sammenligning med kodemottakerene. Det kan være flere ting å studere innenfor selve referansen, men det blir ikke vurdert her. I forhold til lavkost GNSS-kodemottakere er referansen mer enn god nok. Plattformen hadde en allerede etablert koordinatramme og målebilen ble i tillegg målt opp fotogrammetrisk for å avlede relative posisjoner til kodemottakerene i frontruta på bilen.

### 5.3.2 OBU - datasett

GNSS-dataene fra OBUen blir filtrert on-board, noe som førte til at den kun logget ferdig filtrerte koordinater uten noen informasjon om designmatrise eller satelittgeometri. Posisjonsfilen ser omtrent slik ut for de tre første linjene. Dataene ble logget på 1Hz:

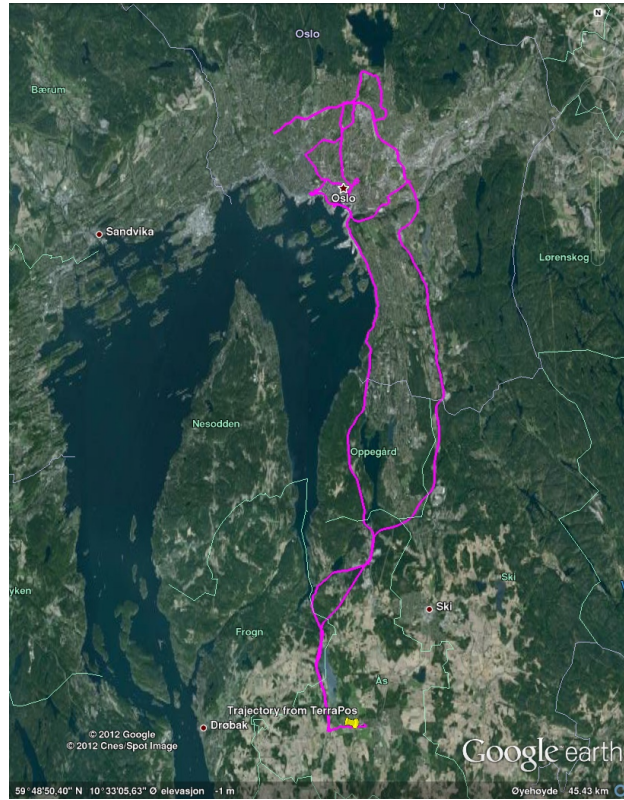
```

TSTAMP YYYY-MM-DD HH:MM:SEC lat lon alt speed heading est_hor_err est_vert_err ...
...#sat_in_fix HDOP SBAS_applied

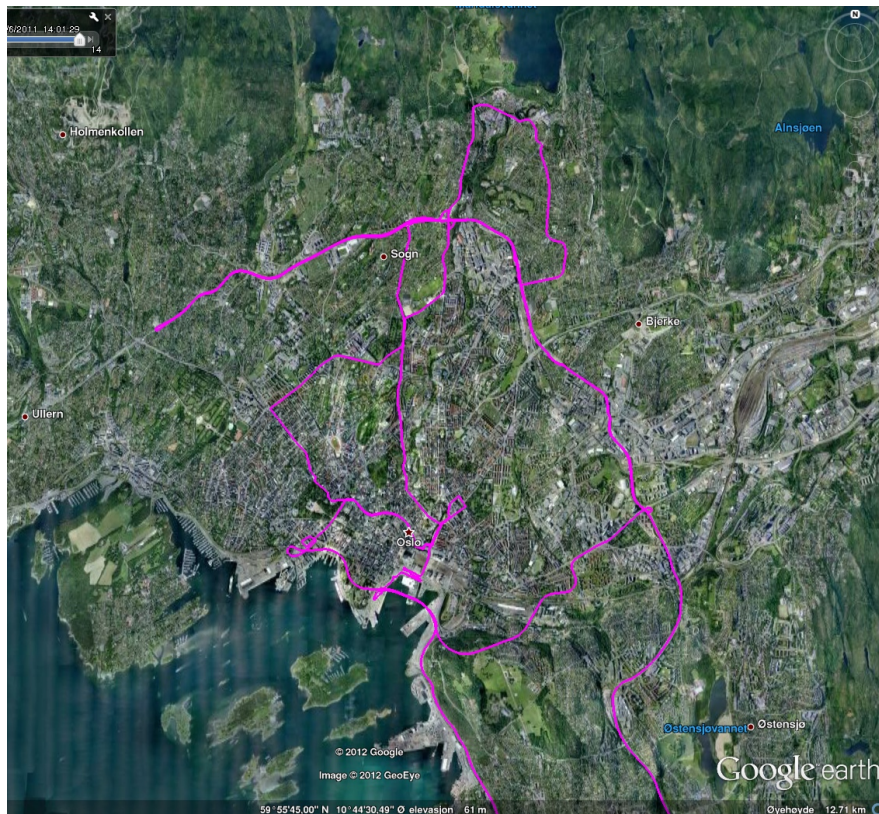
69986225 2011-9-6 8:34:26 59.665352 10.777636 94 1.79 55 0.04 0.01 8 0.80 0
69987235 2011-9-6 8:34:27 59.665363 10.777658 94 2.00 40 0.04 0.01 8 0.80 0

```

Det er verdt å merke seg at fart og heading er gitt i loggen, men det er i prinsippet bare avledet GNSS-data som ikke gir noen mer informasjon enn



(a) Hele datasettet.



(b) Utsnitt av Oslo.

Figur 5.2: Oversikt over kjøreturen. (Google Earth [18])



koordinatene.

IMU-dataene fra OBUen ble logget på 50Hz. Loggen fra IMU-dataene ser slik for de tre første epokene:

```
TSTAMP AX AY AZ GX GY GZ
69292620 -199 -2 971 166 315 -227
69292640 -207 15 977 105 -507 113
69292660 -201 -6 962 253 -577 -262
```

Aksellerometer-brikken og gyro-brikken var montert med  $X$ -aksen i motsatt retning av hverandre på kretskortet til OBUen. Det er viktig å korrigere for dette når dataene lastes inn for prosessering, ved å snu gyro observasjonene  $R_z(180^\circ)$  som i ligning (3.10).

**Feilmontert aksellerometer** Aksellerometerdataene fra OBUen virket ikke som forventet. Det var problematisk å fremskrive posisjoner, og problematisk å bruke de som observasjoner i kalmanfilteret. Det viste seg etter en tid at aksellerometerene var montert feil på selve kretskortet til OBUen. Dataene fra OBU-aksellerometerene er derfor ikke brukt som observasjoner i oppgaven.

### 5.3.3 $OBU_{SIM}$ - Simulert OBU

$OBU_{SIM}$  er ett datasett for en simulert kodemottaker, det ble laget for å se hvordan en kodemottaker oppfører seg i bymiljø uten mye støtte fra en dynamisk modell i et kalmanfilter. GNSS dataene fra OBU enheten var filtrert kraftig i et kalmanfilter på forhånd. Det vil si at posisjonsdataene som ble logget av OBUen lå relativt fint på rekke. Denne oppførselen ser naturlig og riktig ut for en generell bruker, men den store variasjonen i kodemottakeren forsvinner i den filtrerte løsningen. Posisjonsdataene fra  $OBU_{SIM}$  viser en større variasjon og posisjonene ligger ikke like fint på rekke som dataene fra OBUen.

$OBU_{SIM}$  er den samme som simuleringen av GPS/GLONASS-kodemottaker i tillegg A.1.1 på side 64.  $OBU_{SIM}$  er basert på kodeobservasjoner fra mottaker-b og etterprosessert i Terrapos. Datasettet ble prosessert uten

smoother, og med høy systemstøy slik at den dynamiske modellen i filteret skulle få lite innvirkning på det estimerte resultatet.  $OBU_{SIM}$  ligner slik sett mest på en minste kvadraters løsning.

### $OBU_{SIM}^{Koord}$ - koordinater

Koordinatene fra  $OBU_{SIM}$  er brukt i beregninger der selve koordinatene er gitt som observasjoner. Som forklart over har koordinatene fra  $OBU_{SIM}$  større variasjon, og reagerer således raskere på endring i satellittkonstellasjon.

### $OBU_{SIM}^{Pseudo}$ - pseudoavstander

Den originale OBU enheten logget som kjent kun filtrerte koordinater, som fører til at koordinatene blir beregningsgrunnlaget for resultatene i oppgaven. I  $OBU_{SIM}$ (pseudo) er pseudoavstandene beregningsgrunnlaget for resultatene.

Det vil si at integritetsmålet fra  $OBU_{SIM}^{Pseudo}$  er beregnet på den mest optimale måten, med beregning av største gjenværende grovfeil per observasjon.  $OBU_{SIM}^{Pseudo}$  kan derfor sees på som det optimale resultatet for beregning av ytre pålitelighet i en kodemottaker.

## Del II

# Resultater og analyse

# Kapittel 6

## Introduksjon

For å begrense analysen og resultatene til interessante og aktuelle områder ble det sett på to korte veistrekninger. Veistrekningene skulle representere sannsynlige *tilfeller* ved kjøring i urbane områder. Tilfellene ble valgt ut på bakgrunn av problemstillingen, for å svare denne best mulig. For å begrunne valg av tilfeller, må vi rekapitulere problemstillingen kort.

### Ekstrahert problemstilling:

- Vurdere muligheter og begrensninger i en rimelig GNSS/INS sensor.
- Sensoren skal testes i bymiljø, aktuelt for veiprising.
- Avdekke integriteten til den aktuelle, og tilsvarende sensorer.
- Vurdere endringer som kan gjøres for å forbedre påliteligheten.

For å svare problemstillingen best mulig ble det valgt å analysere sensoren der den er forventet å yte dårligst. Om sensoren fungerer bra og gir pålitelig informasjon i trange bygater med dårlig satellittgeometri, så vil den mest sannsynlig gjøre det samme på åpne landeveier. Om sensoren fungerer bra på åpne landeveier, så er det derimot *ikke* sannsynlig at den fungerer like bra i trange bygater. Det er like viktig å vise situasjoner der sensoren *ikke fungerer*, som situasjoner der den fungerer. For å forstå bedre hva som kan

Tabell 6.1: Simulering av GNSS-kodemottakere. Standardavvik beregnet i forhold til referansen.

	Std.		
	Hele datasettet	Oslo sentrum	
GPS	6,80m	8,36m	nord
	2,97m	3,19m	øst
GPS/GLONASS	5,14m	6,36m	nord
	2,83m	3,25m	øst

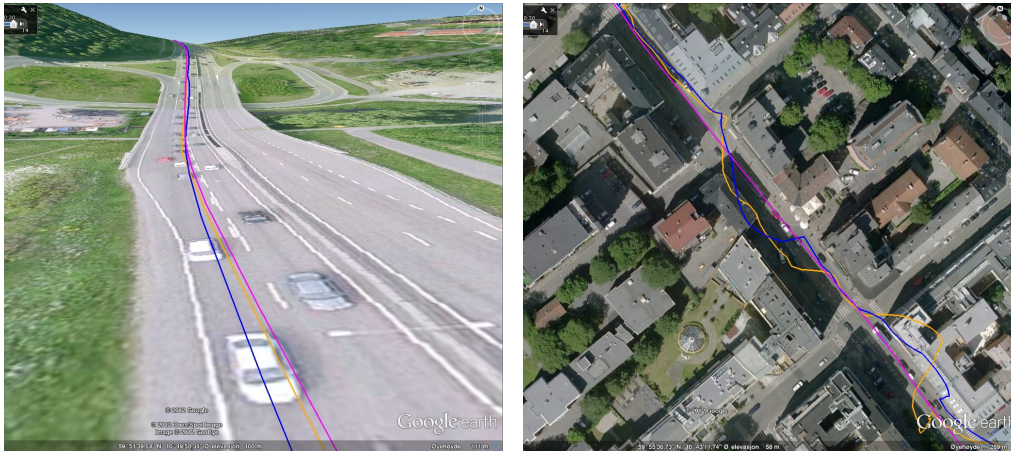
forventes av en generell posisjoneringssensor ble det laget en simulering av kodemottakere.

### Forventet ytelse - GNSS-kodemottakere

En GNSS-kodemottaker kan ved gode forhold, i åpent landskap med god satellittgeometri ha en nøyaktighet på ned mot 5 meter i grunnriss. I urbane miljø med høye bygninger, broer og generelt dårlig satellittgeometri får grove feil større innflytelse på systemet, og nøyaktigheten blir dårligere og varierer i større grad. Simuleringene ble laget for å illustrere hvordan en GPS/GLONASS og GPS-kodemottaker kan yte under bilkjøring i urbane- og åpne-miljø.

Hele simuleringen er beskrevet mer detaljert i tillegg A.1 på side 62. Det ble beregnet en kodeløsning i bymiljø, og en løsning for hele kjøreturen. Observasjonene som lå til grunn var fra mottaker-b (også brukt som referansemottaker). De forskjellige løsningene ble så sammenlignet med referansen for å se fordelingen av sanne feil. Utdrag fra resultatet av simuleringen ligger i tabell 6.1.

Figur 6.1 viser resultatet fra simuleringen plottet i Google Earth. Vi ser fra figuren, og tabell 6.1 at utfordringen til en kodemottaker ligger i urbane områder, som forventet.



(a) Motorvei (E6 ved Skullerud) god satellittgeometri.

(b) Oslo sentrum, dårlig satellittgeometri store avvik fra referansen.

Figur 6.1: Simulering av GNSS-kodemottakere: Referansekjøring:(—) GPS/GLONASS-kodemottaker:(—) GPS-kodemottaker:(—) (3Dmodell og satellittbilde: [18])

## 6.1 Tilfeller

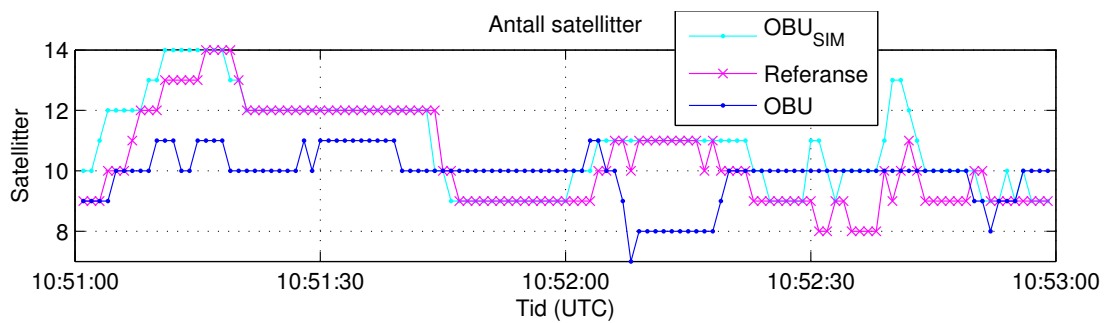
På bakgrunn av problemstillingen ble det valgt to spesifikke tilfeller. De to tilfellene ble valgt etter mye testing. Tilfellene skulle være representative for typiske urbane utfordringer, og helst være helt på grensen av hva sensorene klarte å yte.

### 6.1.1 Tilfelle 1 - Bogstadveien

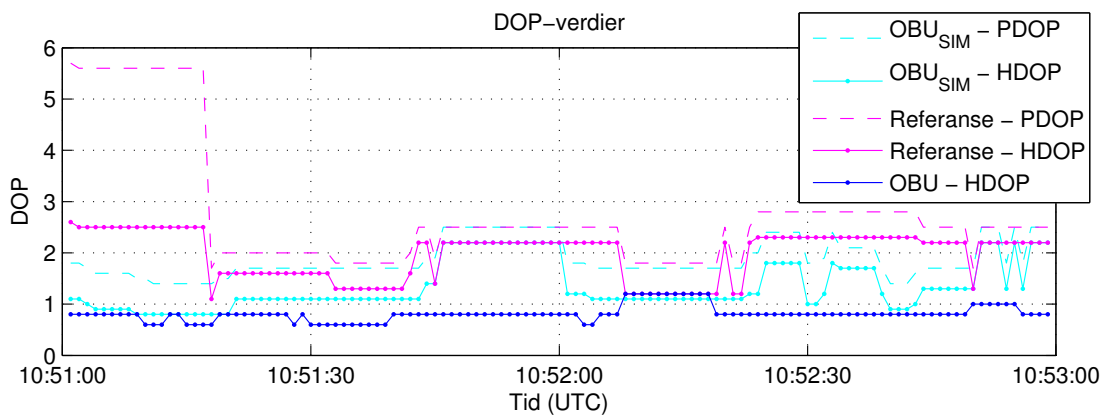
Bogstadveien, kjørende nordover ble valgt som første tilfelle. Strekningen er relativt rett frem, med svak helning oppover. Det er en vanlig tofelts bygate med flere lyskryss, og fotgjengeroverganger. Bebyggelsen er relativt representativ for Oslo sentrum med 4-5 etasjers bygårder som utfordrer satellittgeometrien. Tidsrommet for analysen i Bogstadveien var ca to minutter, fra 10:51:01 til 10:52:59 UTC. Figur 6.2 viser et stillbilde fra videoopptaket i den aktuelle gata. Figur 6.3 viser statistikk over antall satellitter og DOP-verdier for det aktuelle tidsrommet.



Figur 6.2: Stillbilde fra videoopptaket i Bogstadveien.

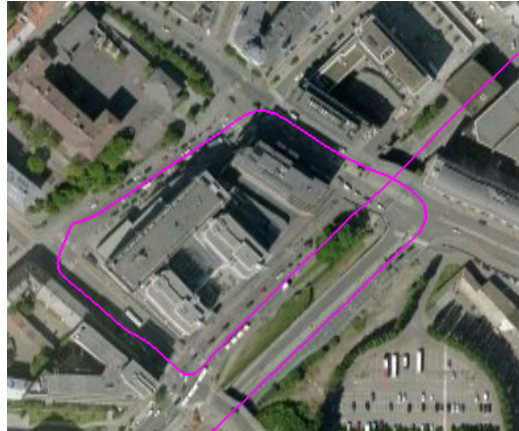


(a) Antall satellitter logget av de forskjellige sensorene.



(b) PDOP og HDOP for de forskjellige sensorene.

Figur 6.3: Tilfelle 1 - Bogstadveien - Antall satellitter og DOP-verdier.

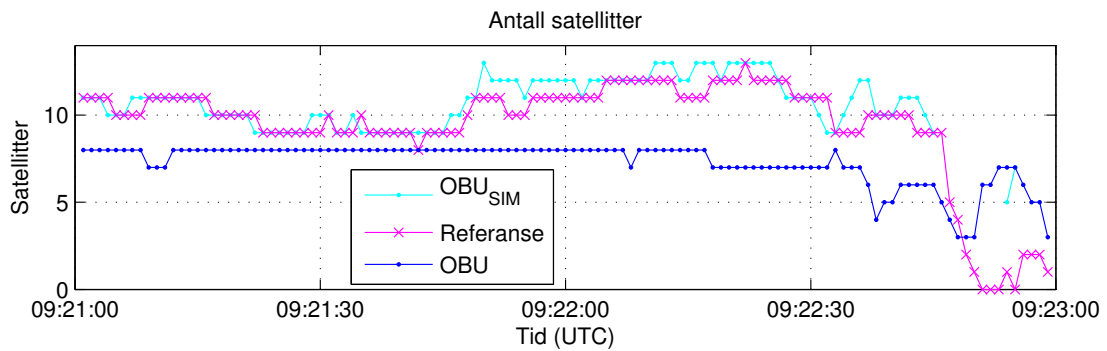


Figur 6.4: Tilfelle 2 - Vika. Referansekjøring — (Google Earth [18])

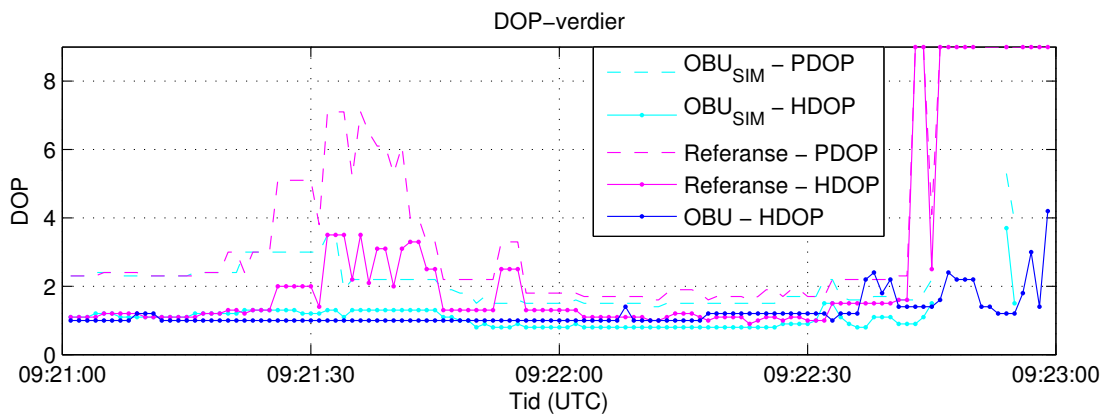
### 6.1.2 Tilfelle 2 - Vika

Munkedamsveien i Vika ble valgt som det andre tilfellet. Det ble valgt fordi det er lyskryss, 90° sving, høye bygg og overganger som utfordrer sensoren. Kjøreturen går ned Cort Adellers gate, for så å svinge inn i Munkedamsveien. Det er et typisk urbant miljø, som er representativt for indre deler av Oslo sentrum. Figur 6.4 viser referansesporet i Vika, plottet i Google Earth. Tidsrommet for analysen i Vika var ca to minutter, fra 09:21:01 til 09:22:59 UTC. Figur 6.5 viser statistikk over antall satellitter og DOP-verdier for det aktuelle tidsrommet.





(a) Antall satellitter logget fra de forskjellige sensorene.



(b) PDOP og HDOP for de forskjellige sensorene.

Figur 6.5: Tilfelle 2 - Vika - Antall satellitter og DOP-verdier.

# Kapittel 7

## Integritetsmål

Dette kapittelet vil gi forslag til beregning av integritetsmål for en lavkost GNSS/INS-sensor. Flere forslag og aspekter vil bli diskutert, men integritetsmålet må oppfylle noen krav.

### Det skal:

- Være *enkelt* å implementere.
- Kunne fungere på tilsvarende sensorer.
- *Kreve lite* prosesseringskraft.
- *Detektere* upålitelige posisjoner.

Det finnes allerede gode metoder for å detektere og fjerne grovfeil, samt beregne indre og ytre pålitelighet. Om en sensor har nok kapasitet, så vil det optimale være å beregne grovfeil per observasjon. I mange applikasjoner kan enklere løsninger fungere godt nok, spesielt om det er lite prosesseringskraft tilgjengelig.

Et integritetsmål, eller feildeteksjonsmål, er et mål på overenstemmelse mellom observasjoner. Det kan for eksempel uttrykkes som summen av alle residualer, som i [19], på generell form:

$$\sqrt{r_1^2 + r_2^2 + \dots + r_n^2} \quad (7.1)$$

I noen applikasjoner er det greit at integritetsmålet kun detekterer de største avvikene, som i smarttelefoner eller lignende der det ofte er viktigere å gi brukeren en posisjon fremfor å kvalitetssikre posisjonen. Andre applikasjoner krever et integritetsmål som heller reagerer en gang for mye, enn én gang for lite. En feil er detektert når integritetsmålet blir større enn en terskelverdi  $T$ , bestemt statistisk. Terskelverdien  $T$  bestemmer hvor store avvik som slipper igjennom.

## 7.1 Koordinatbasert integritetsmål

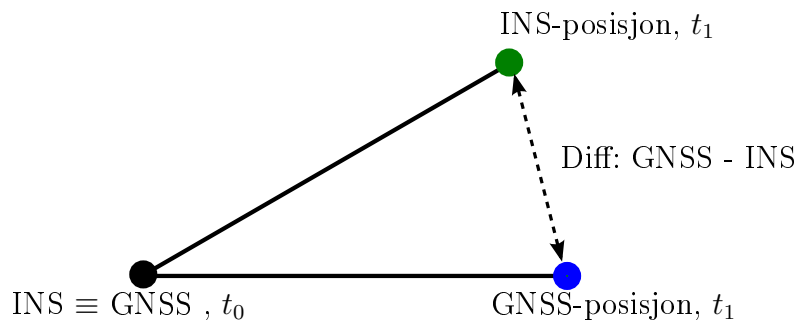
Et koordinatbasert integritetsmål er basert på å regne ut differansen mellom to koordinater ved samme tidspunkt. Siden OBUen har to, i prinsippet uavhengige sensorer som kan avlede koordinater, så kan disse også brukes til å kontrollere hverandre. Som kjent fra tidligere har GNSS-kodemottakere en relativt god nøyaktighet over lang tid, mens IMUer har god nøyaktighet over korte tidsrom.

Med utgangspunkt i figur 7.1 på neste side, og gitt at IMUen har *riktig attityde*, og *riktig fart* så vil den fremskrive posisjonen fra  $t_0$  til  $t_1$  veldig bra. Differansen mellom fremskrevet IMU-posisjon og GNSS-posisjon ved  $t_1$  vil bli stor når det ikke er overenstemmelse, og liten om systemene samsvarer med hverandre. Differansen  $GNSS - INS$ , (fig. 7.1) kan derfor brukes som et integritetsmål på samme form som i ligning (7.1), gitt i grunnriss:

$$\sqrt{(Nord_{INS} - Nord_{GNSS})^2 + (Øst_{INS} - Øst_{GNSS})^2} \quad (7.2)$$

og i 3-dimensjoner, her gitt i jordsentriske koordinater:

$$\sqrt{(X_{INS}^e - X_{GNSS}^e)^2 + (Y_{INS}^e - Y_{GNSS}^e)^2 + (Z_{INS}^e - Z_{GNSS}^e)^2} \quad (7.3)$$



Figur 7.1: Koordinatbasert integritetsmål

### 7.1.1 Forutsetninger

Selvom IMUen og GNSS-mottakeren i prinsippet er uavhengige sensorer, så er de ikke det i praksis. IMUen må stole på at posisjonen den får fra GNSS-mottakeren ved  $t_0$  er korrekt. Dette kan føre til at integritetsmålet går over terskelverdien når posisjonen går fra å være *pålitelig* til *upålitelig*, men også når posisjonen går fra å være upålitelig til pålitelig. Metoden vil derfor fange opp situasjoner der satellittgeometrien endrer seg mye.

*Attityden* til IMUen må være korrekt for at den skal kunne fremskrive posisjonen i riktig retning. Om headingen er feil (ved f.eks tidspunkt  $t_0$ ) og farten er høy kan differansen bli veldig stor, selvom GNSS-posisjonene ved  $t_0$  og  $t_1$  i seg selv har god integritet.

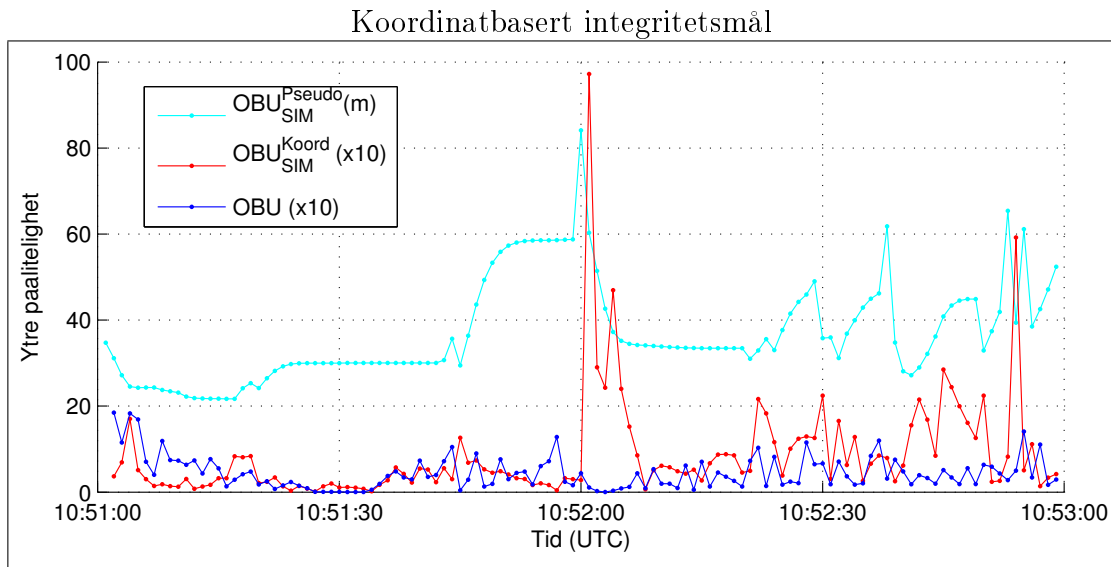
Rå, *ufiltrerte* GNSS-observasjoner vil være det beste grunnlaget for å teste metoden. GNSS-observasjonene fra OBUen er filtrert i hardwaren, noe som fører til at observasjonene ser ut til å ligge fint på rekke. Store endringer og utslag i satellittgeometrien synes derfor ikke like godt i GNSS-dataene fra OBUen. Ufiltrerte koordinater viser større variasjon, spesielt når satellittgeometrien er dårlig, og endrer seg mye.

### 7.1.2 Optimalt eksempel - konsept

Dette eksempelet illustrerer hvordan integritetsmålet i ligning (7.2)(grunnriss) fungerer i en «perfekt» situasjon. Sensoren kjenner ved hver epoke *sann at-*

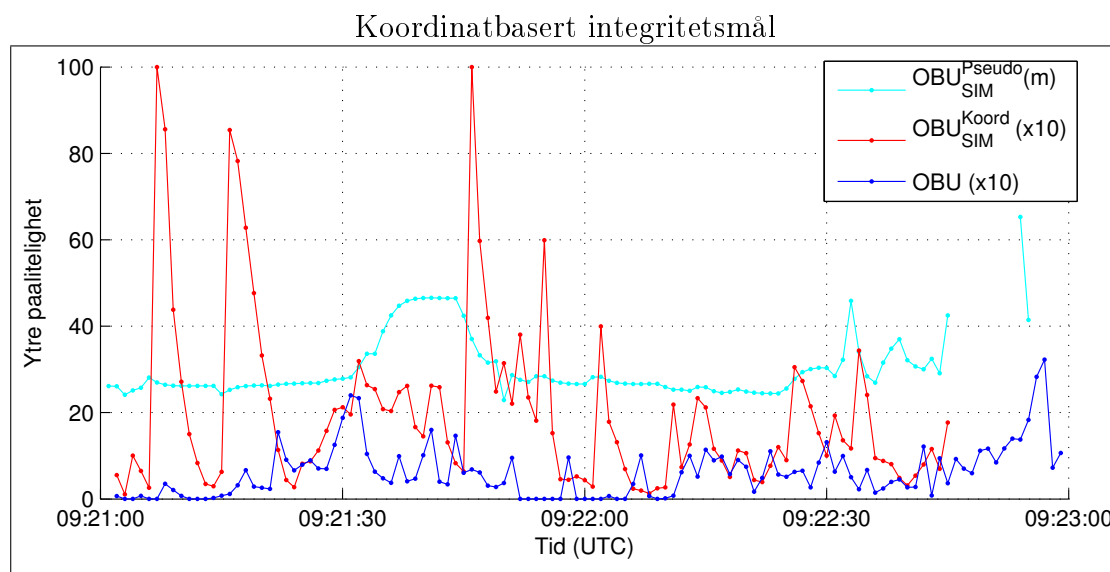
*tityde* og *sann fart*. Det vil si at ved epoke  $t_k$  får INS-systemet gitt posisjon fra kodemottaker, og sann orientering fra referansen. INS-systemet fremskrives med sann fart fra referansen gjennom gyro-observasjonene fra OBUen til  $t_{k+1}$ . Fremskrivingen sammenlignes med GNSS-posisjonen ved  $t_{k+1}$ , og så får INS-systemet igjen sann attityde og fart og kan starte ved  $t_k$ .

Integritetsmålet er testet på tilfelle 1, og tilfelle 2, og resultatet er gitt i henholdsvis figur 7.2 og figur 7.3. Integritetsmålet er sammenlignet med ytre pålitelighet fra  $OBU_{SIM}^{Pseudo}$  som her kan sees på som den ideelle oppførselen til integritetsmålet. Resultatene til  $OBU_{SIM}^{Pseudo}$  og  $OBU_{SIM}^{Koord}$  er skalert (x10) for å vise hvordan de oppfører seg relativt den ytre påliteligheten.



Figur 7.2: Optimalt eksempel (Tilfelle 1 - Bogstadveien): koordinatbasert integritetsmål sammenlignet med ytre pålitelighet fra  $OBU_{SIM}^{Pseudo}$ .

I figur 7.2, ser vi at integritetsmålet beregnet med  $OBU_{SIM}^{Koord}$  reagerer omtrent på samme måte som den ytre påliteligheten til  $OBU_{SIM}^{Pseudo}$ . Variasjonen i integritetsmålet beregnet med OBUen er derimot lavere og, generelt svakere i forhold til  $OBU_{SIM}^{Koord}$ . Grunnen til det svake utslaget fra OBUen er sannsynligvis fordi OBU-koordinatene er filtrert ganske mye som forklart i delkapittel 5.3.3. Integritetsmålet reagerer derfor tydeligere når det beregnes med  $OBU_{SIM}^{Koord}$ .



Figur 7.3: Optimalt eksempel (Tilfelle 2 - Vika): koordinatbasert integritetsmål sammenlignet med ytre pålitelighet fra  $OBU_{SIM}^{Pseudo}$ .

### Videre føringer

Eksempelet over er konseptuelt og viser således hvordan det er tenkt at integritetsmålet kan fungere. Veien mellom det optimaliserte eksempelet og et fungerende integritetsmål i for eksempel veiprisings-applikasjoner er ikke like triviell. I virkelig bruk har vi ikke all den informasjonen som er brukt i eksempelet over. Neste steg er å kun bruke den informasjonen som er gitt fra sensoren.

**Attityde:** Den informasjonen som sannsynligvis er mest optimistisk i det optimaliserte eksempelet er attityden. En rimelig sensor er ikke i nærheten av den stabiliteten som ligger i referansen sin attityde. Attityden er viktig å vite med en viss sikkerhet for å fremskrive INS-systemet i riktig retning.

**Fart:** Fremskrivingen av INS-systemet i det optimaliserte eksempelet brukte sann fart fra referansen. Verken OBUen eller  $OBU_{SIM}$  har en ren observasjon av fart. Det vil si at farten som er gitt fra de to sensorene kun

er avledet fra GNSS-observasjonene. Farten er viktig å vite for å fremskrive INS-systemet riktig. Samtidig er farten viktig for å kunne vite om bilen står stille eller ikke. Om man vet at bilen står helt stille kan man utføre en ZUPT-oppdatering som hjelper attityden.

**Geometri:** Det er ikke gitt noen direkte informasjon om geometrien til satellittene i designmatrise eller lignende. Den eneste informasjonen som er gitt fra OBU-enheten om satellittgeometri er HDOP. Det kan derfor være en mulighet å kombinere HDOP-verdien med det koordinatbaserte integritetsmålet. Til eksempel, se hvordan DOP-verdiene fra figur (b) 6.3 på side 41 varierer i forhold til integritetsmålet i figur 7.2 på side 47.

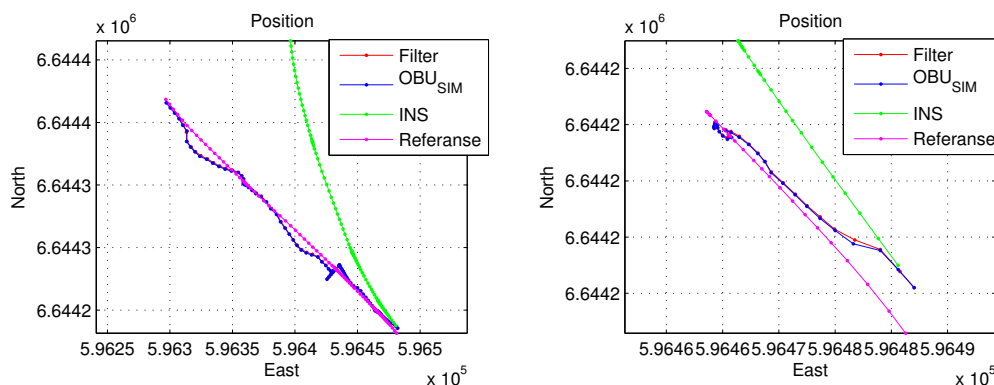
### 7.1.3 Test av koordinatbasert integritetsmål

Det koordinatbaserte integritetsmålet i ligning (7.2) og (7.3) er avhengig av å kunne fungere på sensorens premisser. Om ikke, vil det ikke fungere i sin hensikt. Dette delkapittelet vil derfor ta utgangspunkt i de videre føringene som ble gitt i slutten av delkapittel 7.1.2.

#### Attityde

Første test er å se hvordan det koordinatbaserte integritetsmålet, og sensoren fungerer når den selv må bestemme attityde. Gyroen er forventet å drifte mye, men det er interessant å vite hvor lenge den kan klare seg uten assistanse. Eksempelet er testet med kriteriene:

- Initiell attityde fra referansen.
- OBUen sin gyro styrer attityden ( $C_b^e$ -matrisen) uten videre assistanse.
- Fremskrevet med sann fart fra referansen.
- Koordinater fra OBU, og  $OBU_{SIM}^{Koord}$ .



(a) Hele tilfelle 1. 10:51:01 - 10:52:59 UTC. (b) Utsnitt av de 30 første sekundene. 10:51:01 - 10:51:31 UTC.

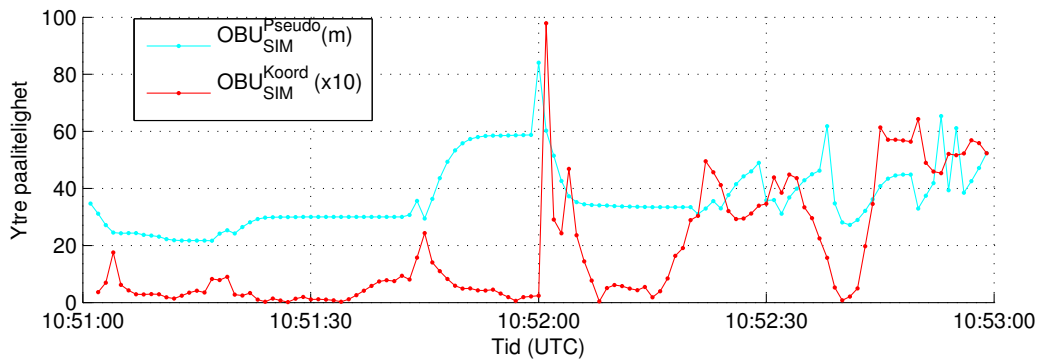
Figur 7.4: Tilfelle 1, Bogstadveien - Posisjon, referanse mot free inertial navigation.

**Tilfelle 1 - Bogstadveien:** Figur 7.4(a og b) viser posisjonsplott over strekningen. Figur 7.5 på neste side viser resultatet av integritetsmålet, fartsgrafene og estimert attityde. Ved å sammenligne resultatet i figur 7.5-a med det optimale eksempelet i figur 7.2 på side 47 er det tydelig forskjell i det siste minuttet. Toppene fra 10:52:00 og ut er tydelig større, og det kan se ut som at integritetsmålet øker med hastigheten. Dette skyldes at attityden ikke er godt nok bestemt etter en tid og med økt hastighet øker differansen  $GNSS - INS$ .

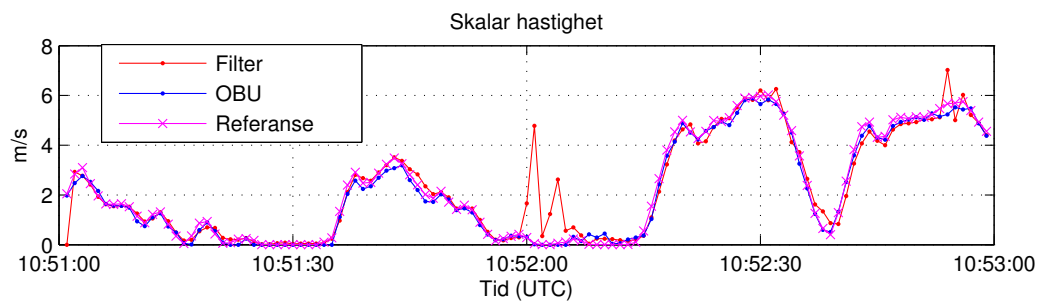
Attityden estimert i kalmanfilteret (figur 7.5-c) avviker lite fra referansen de første 30 sekundene. Ved å sammenligne attityden med posisjonsplottet i figur 7.4 er det tydelig at gyroene er relativt gode over ca 30 sekunder. Loggen fra OBUen ser ut til å gi fornuftig heading når bilen er i fart. De periodene bilen står stille (10:51:30 og 10:52:00) er headingen fra OBUen helt uforutsigbar. Dette skyldes at heading-informasjonen gitt fra OBUen er avledet fra koordinatene, og når bilen står stille forsvinner den informasjonen i GNSS-målestøy.

**Tilfelle 2 - Vika:** Den samme testen ble gjort på tilfelle 2, i Vika. Da integritetsmålet ikke fungerer optimalt under testens kriterier, er integritets-

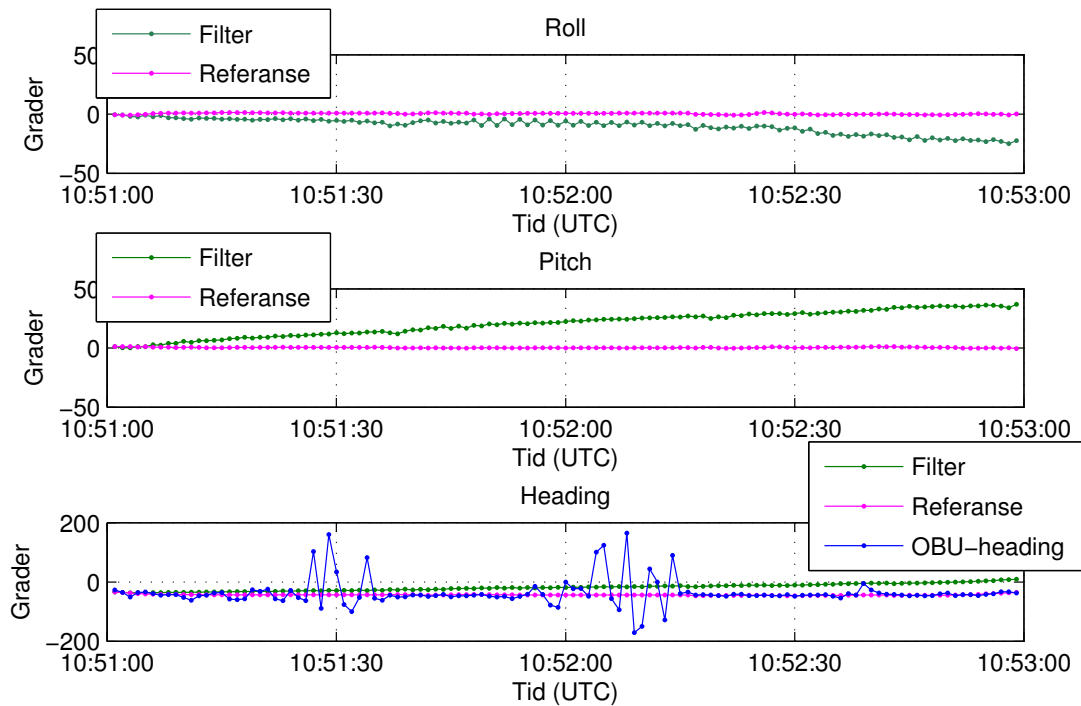




(a) Integritetsmål.



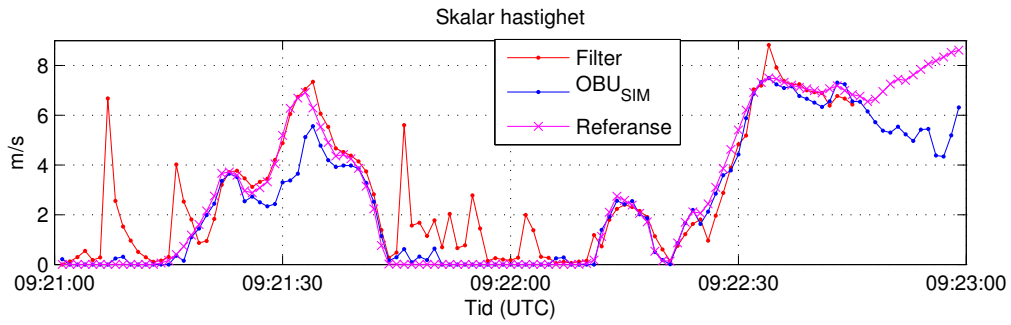
(b) Hastighet.



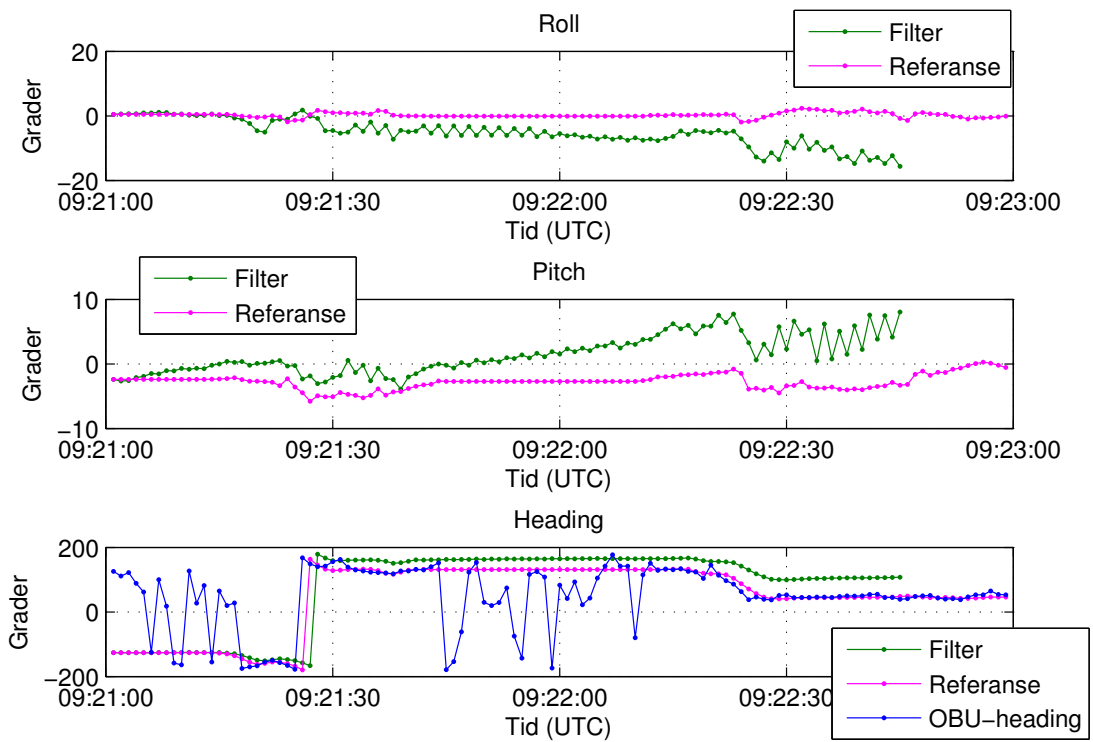
(c) Attityde (roll, pitch, heading).

Figur 7.5: Tilfelle 1, Bogstadveien.

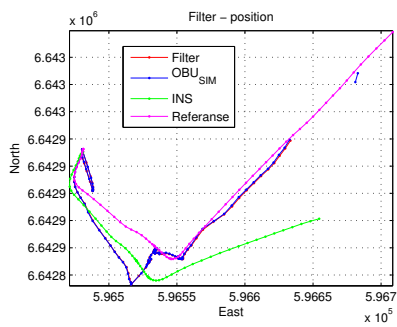
plottet ikke tatt med. Mer interessant er ytelsen til attityden, og hvordan denne oppfører seg i tilfelle 2. Resultatet, med fart og attityde vises i figur 7.6. Attityden i figur 7.6-b viser omtrent samme gode oppførselen de første 30 sekundene som i tilfelle 1. Figur 7.5-c viser tilhørende posisjonsplott.



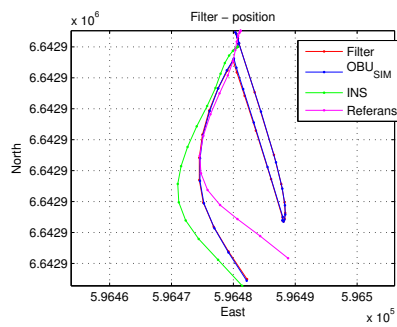
(a) Hastighet.



(b) Attityde (roll, pitch, heading).



(c) Hele tilfelle 2. 09:21:01 - 09:22:59 UTC.



(d) Utsnitt av de 30 første sekundene. 09:21:01 - 09:21:31 UTC.

Figur 7.6: Tilfelle 2, Vika.

## 7.2 Diskusjon

Det ser ut til å være flere positive sider ved sensoren, samtidig som den ikke virker å være helt selvhjulpet. OBUen som er testet ser ut til å være laget for mer generell bruk der det er viktig å gi ut en posisjon fremfor å kvalitetssikre den.

### Satellittdekning

OBUen satt i frontruten på bilen, noe som normalt ville gitt dårligere satellittmottak i forhold til OBU<sub>SIM</sub> som satt på taket. Spesielt tydelig er det at OBUen kan virke å ta inn alle signaler, selvom om den kjører inn under en bro. Figur 6.5 viser at OBU<sub>SIM</sub> og referansen mister alle satellittene samtidig som OBUen har kontakt med mellom 3 og 7 satellitter. Det er viktig å nevne at OBU<sub>SIM</sub> er basert på en geodetisk mottaker som kan skille ut multipath og andre grovfeil mens den logger. Men samtidig satt OBU<sub>SIM</sub> montert på taket, som gjør at den har større synsvidde til himmelen. OBUen virker å logge signaler helt ukritisk, noe som fort kan dra med seg store multipath-effekter eller lignende.

### Attityde

OBUen virker å estimere heading relativt fint når det er en viss hastighet på bilen. Heading er en viktig komponent for orientering av sensoren, spesielt om INS-sensoren må fremskrive posisjon uten støtte fra GNSS. Problemet med den estimerte headingen fra OBUen er at når bilen står i ro er headingen helt ukontrollert.

For å kontrollere roll og pitch vil en ZUPT-oppdatering (zero velocity update) horisontere roll/pitch-planet. Under en ZUPT-oppdatering endres designmatrisa og den dynamiske modellen slik at akselerometerene måler tyngdekomponenten, og retter opp systemet. Dette vil hjelpe roll og pitch,

men headingen står normalt på og får ikke noe hjelp fra en ZUPT-oppdatering. Det som er viktig med en ZUPT-oppdatering er at sensoren står *helt i ro* for at *kun tyngdekomponenten* skal observeres av akselerometerene.

I urbane miljøer der det blandt annet er mange lyskryss kan derfor ZUPT-oppdateringer være nyttig. Om det gjøres en ZUPT-oppdatering mens bilen er i bevegelse kan gyro-aksene få feil orientering, som igjen kan skade systemets integritet.

Et forslag til optimalisering av attityden slik sensoren er gitt i oppgaven kan være å la:

- I fart, over en viss grense:
  - Heading - estimert fra GNSS-observasjoner.
  - Roll/pitch - estimert fra gyro.
- Ved stopp, f.eks. på rødt lys:
  - Heading - estimert fra gyro.
  - Roll/pitch - estimert av akselerometere, gjennom en ZUPT.

Dette kan sansynligvis fungere i noen tilfeller, men det er flere utfordringer. Farten er ikke en kjent observasjon, og kun avledet fra GNSS-observasjoner. Det vil føre til at OBUen i noen tilfeller vil si at den står i ro, der sannheten er at bilen er i bevegelse. Dette vil ødelegge hele estimatet fra ZUPT-oppdateringen. Om bilen kjører sakte, for eksempel i kø, vil sannsynligvis ingen av forslagene fungere, da heading og fartsestimatet forsvinner i målestøyen.

Et annet alternativ er å bruke en annen sensor som kan observere heading direkte, som f.eks et elektronisk kompass. Problemet med elektroniske kompass er at de sansynligvis blir svekket ved bruk i biler på grunn av metall.

## Hastighet

Hastighetsestimater som er logget fra OBUen ser riktig ut i forhold til referansen i figur 7.5-b. Men ved å se på tilsvarende hastighetsplott i figur 7.6-a, ser man tydelig at de avviker. Dette er samme tidspunkt som OBUen rapporterer at den har flere satellitter enn hva den geodetiske mottakeren på taket hadde.

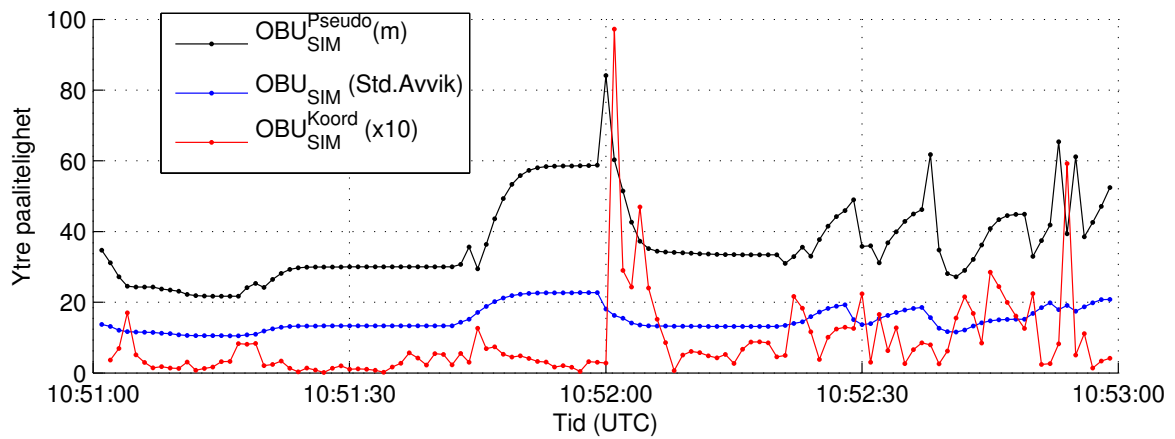
Om attityden skal estimeres på en god måte, så trenger OBUen en sensor som kan si noe om hastigheten. Enten en sensor som sier ifra om bilen står stille, eller en sensor som observerer hastighet. Den beste løsningen vil være en hastighetssensor, da den også kan brukes til å fremskrive systemet i kortere perioder uten GNSS-hjelp.

## Geometri

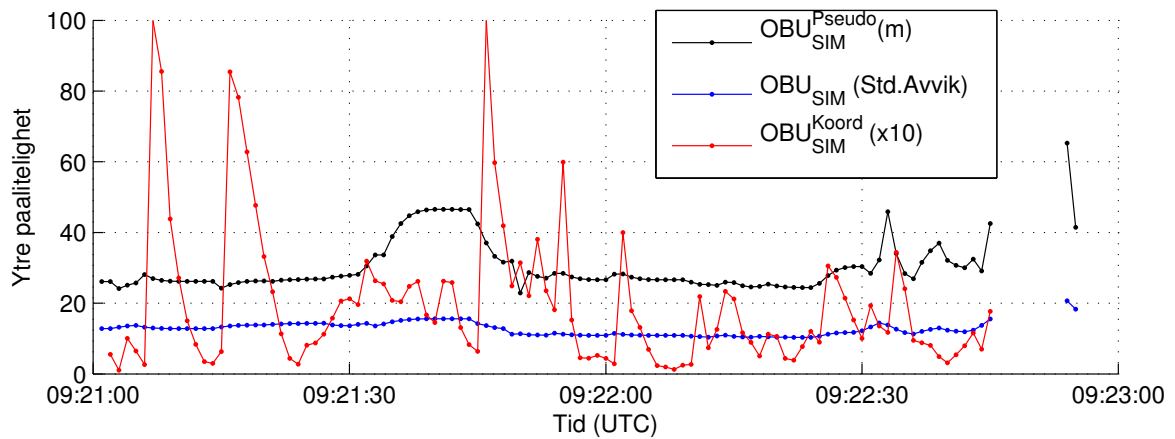
Ved å vite geometrien til satellittene er det mulig å beregne ytre pålitelighet per observasjon, og få et optimalt mål på hvor godt posisjonen er bestemt. Fra OBUen var det kun standardavvik og HDOP som gav noe informasjon om geometrien. Dette førte til at det ble simulert en  $OBU_{SIM}$ , og  $OBU_{SIM}^{Pseudo}$  som gav et optimalt integritetsmål, i form av ytre pålitelighet.

Figur 7.7 illustrerer hvordan standardavviket i grunnriss oppfører seg i forhold til ytre pålitelighet og det koordinatbaserte integritetsmålet. Det er tydelig at standardavviket følger noe av oppførselen til den ytre påliteligheten, men utslaget er veldig lite. Dette plottet sier hvor viktig det er å ikke stole blindt på standardavviket som et geometrimål.

### Ytre pålitelighet mot Standardavvik



(a) Tilfelle 1 - Bogstadveien.



(b) Tilfelle 2 - Vika.

Figur 7.7: Ytre pålitelighet mot standardavvik i grunnriss.

# Kapittel 8

## Konklusjon

Utgangspunktet for konklusjonen er problemstillingen i kapittel 1.1:

- Lage et kalmanfilter i løs kobling for å teste og illustrere grunnprinsippene i et GNSS/INS-system.
- Prøve alternative metoder for å beregne integriteten til den estimerte posisjonen.
- Vurdere hvor et slikt system fungerer og hvor det ikke fungerer i urbane miljøer.
- Vurdere eventuelle endringer som kan gjøres for å forbedre påliteligheten.

Hovedmomentet med oppgaven var å avdekke integriteten til sensoren, og hvordan den aktuelle, og tilsvarende sensorer kan kvalitetskontrollere sin egen posisjon.

Det er gode muligheter for utvikling av sensorer som kan kvalitetskontrollere sin egen posisjon i urbane miljø. Sensorer som har nok prosesseringskraft og strøm vil kunne beregne grovfeil per observasjon, og således gi et optimalt integritetsmål. I en sensor med kalmanfilter der prosesseringskraft og batteritid er mangelvare er valg av type sensorer helt kritisk for resultatet.



Integritetsmål beregnet på koordinatnivå fungerer i teorien, men det er avhengig av gode estimater på attityde. Dette gjør at vinningen ved et enkelt integritetsmål ikke kan forsvares, da testen like godt kan baseres på kalman-innovasjonen.

Det koordinatbaserte integritetsmålet fungerer der det ble testet. Problemet er at resultatet fra det samme integritetsmålet, på samme veistrekning kan endre seg fra gang til gang.

For den aktuelle OBUen som ble testen på kjøreturen vil hovedsaklig to endringer gjøre stor forskjell. Ved å la OBUen logge pseudoavstander, vil man kunne beregne ytre pålitelighet og gjøre statistiske tester på observasjonene. Den andre endringer er å observere hastighet. En hastighetsobservasjon vil kunne fremskrive INS-systemet med bedre sikkerhet. Samtidig er det å vite om sensoren står helt i ro eller ikke en viktig observasjon for å horisontere systemet med ZUPT-oppdatering.

# Bibliografi

- [1] C. de Jong, *Hydrography*. [Delft]: VSSD, 2002.
- [2] S. N. Leksikon, "Store norske leksikon," 04 2012.
- [3] S. Kartverk, N. K. Forbund, and rådet for teknisk terminologi, *Ordbok for kart og oppmåling*. Statens Kartverk, 1989.
- [4] W. Torge, *Geodesy*. Walter de Gruyter, 3rd completely rev. and extended ed. ed., 2001.
- [5] J. A. Farrell, *Aided Navigation: GPS with High Rate Sensors*. Mc Graw Hill, 2008.
- [6] B. R. Pettersen, *Forelesninger i satellittbaneteori og romgeodesi*. Institutt for kartfag, NLH, 2000.
- [7] I. Skog, "A low-cost gps aided inertial navigation system for vehicular applications," Master's thesis, Kungliga Tekniska högskolan, 2005.
- [8] C. Jekeli, *Inertial navigation systems with geodetic applications*. Berlin: Walter de Gruyter, 2001.
- [9] E. Weisstein, "Wolfram mathworld."
- [10] K. Gade, "Introduction to inertial navigation and kalman filtering." Tutorial for IAIN World Congress, Stockholm, 2009.
- [11] Wikimedia, "Wikimedia foundation," 2012.

- [12] E. J. Post, “Sagnac effect,” *Rev. Mod. Phys.*, vol. 39, pp. 475–493, April 1967.
- [13] G. Seeber, *Satellite Geodesy: completely revised and extended edition*. Walter de Gruyter, 2nd ed., 2003.
- [14] O. Øvstedal, “Absolutt posisjonsbestemmelse med kode-observasjoner fra gps.” Forelesningsnotat GMSG210, 2000.
- [15] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal Of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [16] R. G. Brown and O. Y. Hwang, *Random Signals and Applied Kalman Filtering: with MATLAB Exercises*. John Wiley & Sons, fourth ed. ed., April 2012.
- [17] Honeywell, “Datasheet: Hg9900 navigation-grade inertial measurement unit.”
- [18] G. Earth, S. Image, and G. Eye, “Google earth,” 04 2012.
- [19] “Forelesningsnotat: Delft university of technology - radio navigation - system performance and failure detection,” 2005.

# Tillegg A

## A.1 Simulering av kodemottakere

Bakgrunnen for å simulere en kodemottaker er todelt. Det ene er at GNSS dataene fra OBU enheten allerede var filtrert i et kalmanfilter, slik at posisjonene lå relativt fint på rekke. For å teste og illustrere noen konsepter var det derfor behov for å lage et datasett der de gitte posisjonene viste en mer naturlig oppførsel uten å være filtrert i stor grad på forhånd.

Det andre poenget med simuleringen var å illustrere hvordan en generell kodemottaker kan yte i forskjellige omgivelser.

### A.1.1 Simulering av kodemottakere

Simuleringen ble laget i Terrapos med utgangspunkt i referansedataene fra Oslo. For å simulere ytelsen i tilfeller med variert kjøring i by og mye motorvei, ble hele datasettet fra Oslo brukt, fra start til slutt i Ås. For å sammenligne, og simulere hvordan urbane miljøer påvirker nøyaktigheten til en kodemottaker ble det samme datasettet kuttet ned til Oslo sentrum ut til Ring 3. Tidsrommet for Oslo sentrum simuleringen går fra 9:22:15(UTC) til 11:05:45(UTC). Resultatene er utklipp fra navigasjonsdifferansen laget i Terrapos, med en kort sammenligning av resultatene i tabell 6.1 på side 39.

**Referanse:** Referansen er den etterprosesserte TC-GNSS/INS-løsningen fra Terrapos basert på mottaker-b (rec-b). Referansen sees på som komplett med sanne data slik at nøyaktigheten til de simulerte datasettene blir kjent, noe det normalt ikke er.

**Simulert kodemottaker:** «Kodemottakeren» er kun basert på kodeobservasjoner fra mottaker-b og etterprosessert i Terrapos. Datasettet er prosessert *uten smoother*, og med *høy systemstøy* slik at filteret får lite innvirkning på det estimerte resultatet, og ligner på en minste kvadraters løsning. Både referansen og kodemottakeren er basert på samme observasjonsfil, dette fører til at eksemplene er noe overoptimistiske og *ikke uavhengige*. Kodemottakeren må derfor også sees på som en simulert mottaker.

## GPS-kodemottaker

**Hele datasettet:** utklipp fra resultatet av navigasjonsdifferansen:

	Min.	1%	5%	50%	95%	99%	Max.	
North pos.	-17.003	-8.715	-5.219	0.450	9.814	38.659	57.120	[ m ]
East pos.	-24.916	-6.894	-3.877	0.003	4.536	10.247	24.934	[ m ]
Down pos.	-100.063	-50.099	-13.361	-2.824	5.623	10.960	21.049	[ m ]
	Mean	Std-dev.	RMS					
North pos.	1.386	6.798	6.938		[ m ]			
East pos.	0.163	2.974	2.978		[ m ]			
Down pos.	-3.574	9.033	9.714		[ m ]			

I grunnriss er det største avviket 57,1 m i nord, og  $\pm 24,9$  m i øst. 95% av de beregnede avvikene er innenfor 9,8m i nord og 4,5m i øst. Standardavviket blir ekvivalent med nøyaktigheten. I grunnriss er det omtrent som forventet av en GPS-kodemottaker under varierte/gode forhold. Standardavviket i høyden er omtrent det dobbelte av standardavviket i grunnriss som forventet.

**Oslo sentrum:** utklipp fra resultatet:

	Min.	1%	5%	50%	95%	99%	Max.
--	------	----	----	-----	-----	-----	------

North pos.	-16.704	-9.113	-5.014	0.749	12.855	47.912	57.121	[ m ]
East pos.	-24.919	-5.691	-3.666	-0.259	4.957	10.749	24.931	[ m ]
Down pos.	-68.935	-55.884	-14.412	-2.812	4.012	8.427	19.767	[ m ]

---

	Mean	Std-dev.	RMS	
North pos.	2.318	8.363	8.678	[ m ]
East pos.	0.210	3.192	3.198	[ m ]
Down pos.	-4.253	9.392	10.309	[ m ]

Vi ser at standardavviket blir noe dårligere, spesielt i nord. 95% av punktene avviker med mindre enn 12,8 m (mot 9,8 m i hele datasettet), noe som betyr at det er en mye større andel store avvik i Oslo sentrum.

## GPS/GLONASS-kodemottaker

For se hvordan en GPS/GLONASS-mottaker yter i forhold til en ren GPS-mottaker ble det laget en tilsvarende differanse med GPS/GLONASS-kodeobservasjoner.

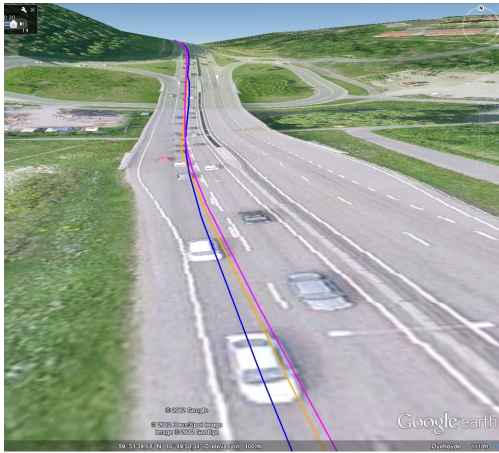
### Hele datasettet - $OBU_{SIM}$ -løsning: Utklipp fra resultatfil:

	Min.	1%	5%	50%	95%	99%	Max.	
North pos.	-41.507	-14.449	-6.387	-0.876	4.352	17.887	52.293	[ m ]
East pos.	-17.093	-7.441	-4.097	-0.188	4.019	7.943	19.998	[ m ]
Down pos.	-97.272	-53.455	-9.716	-3.125	3.543	8.854	39.980	[ m ]

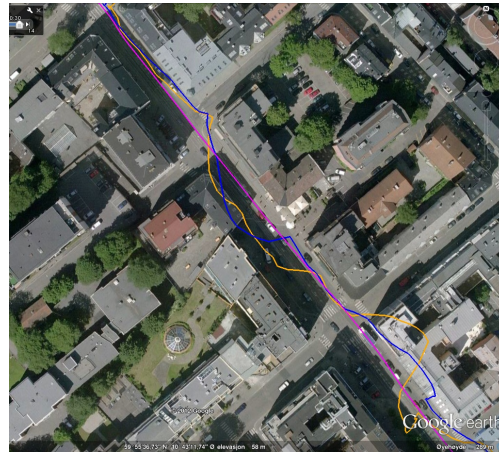
---

	Mean	Std-dev.	RMS	
North pos.	-0.872	5.142	5.216	[ m ]
East pos.	-0.136	2.832	2.836	[ m ]
Down pos.	-3.983	9.390	10.200	[ m ]

Det som er verdt å merke seg ved GPS/GLONASS-simuleringen er at det er færre store avvik enn i GPS-simuleringen. En del av dette skyldes trolig at Terrapos klarer å luke ut flere grovfeil på grunn av flere observasjoner. Et annet aspekt er også at satellittgeometrien vil være styrket, og noe mer stabil med flere satellitter tilgjengelig.



(a) Motorvei (E6 ved Skullerud) god satellittgeometri.



(b) Oslo sentrum, dårlig satellittgeometri store avvik fra referansen.

Figur A.1: Simulering av GNSS-kodemottakere: Referansekjøring:(—) GPS/GLONASS-kodemottaker:(—) GPS-kodemottaker:(—) (3Dmodell og satellittbilde [18])

#### Oslo sentrum: utklipp fra resultatet:

	Min.	1%	5%	50%	95%	99%	Max.	
North pos.	-41.394	-20.218	-7.379	-1.742	7.188	24.456	44.458	[ m ]
East pos.	-16.787	-8.671	-4.473	-0.500	5.001	10.843	21.757	[ m ]
Down pos.	-106.820	-82.590	-20.042	-3.637	1.772	7.057	39.964	[ m ]
	Mean	Std-dev.	RMS					
North pos.	-1.197	6.359	6.470		[ m ]			
East pos.	-0.160	3.252	3.256		[ m ]			
Down pos.	-5.985	12.770	14.102		[ m ]			

Standardavviket i grunnriss ved bruk av GPS- og GLONASS-observasjoner i Oslo sentrum er lavere enn med kun GPS-observasjoner.





# Tillegg B

## Etablering av platform

### Dokumentasjon og koordinater for GNSS-/INS-målebil

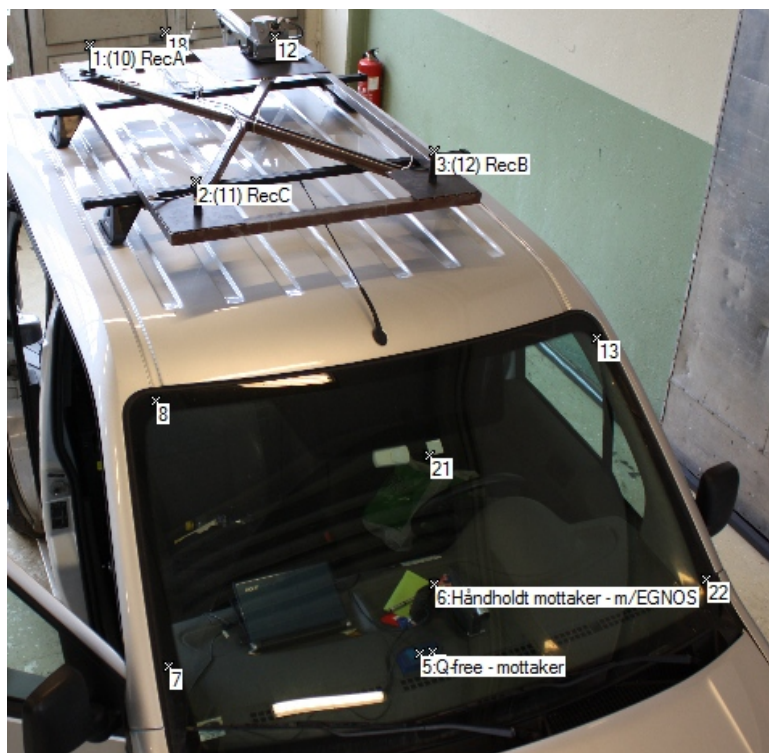
Are Jo Næss  
Universitet for Miljø og Biovitenskap  
12. oktober 2011

Programvare: PhotoModeler Pro 5.2.3  
Enhet i prosjektet: millimeter

Koordinatene er gitt i samme koordinatsystem som GNSS/INS-ramma som ble beskrevet i «Dokumentasjon og koordinater for GPS- og INS-ramme 18.03.2009» av Ivar Maalen-Johansen. Punkt-Id 1-3 er kjentpunkt fra GNSS/INS-ramma. Punkt 5 og 6 er kodemottakere. Resterende punkter er kun hjelpepunkt/sammenbindingspunkt brukt til å orientere bildene.

Tabell 1: Målte punkt

Id	Navn	Koordinater - mm			Presisjon - mm		
		X	Y	Z	X	Y	Z
1	(10) RecA	86.7	783.4	-88.8	1.7	0.5	0.5
2	(11) RecC	1380.7	783.2	-95.7	0.6	0.5	0.6
3	(12) RecB	1381.0	88.4	-89.9	0.6	0.6	0.5
5	Q-free - mottaker	2397.6	558.8	726.3	0.7	0.3	0.4
6	Håndholdt mottaker m/EGNOS	2327.7	501.1	577.0	0.6	0.3	0.4
4	-	2400.4	528.1	726.4	0.7	0.3	0.4
7	-	2308.0	1154.0	642.1	0.6	0.9	0.4
8	-	1825.8	1032.2	255.9	0.4	0.8	0.3
9	-	-163.4	1256.2	752.0	2.0	0.8	0.9
10	-	34.0	1252.6	682.5	1.8	0.8	0.7
12	-	175.7	123.0	-80.6	1.5	0.5	0.5
13	-	1864.2	-202.5	299.8	0.4	0.9	0.3
15	-	-3578.7	722.3	602.1	3.7	0.7	1.0
16	-	-3586.2	362.8	597.4	3.7	0.8	1.0
18	-	-3561.4	-464.4	988.4	4.7	1.1	1.4
19	-	-3551.0	1365.8	567.8	8.1	0.8	1.6
21	-	2084.8	412.8	389.8	0.4	0.3	0.2
22	-	2419.8	-275.4	735.5	0.8	1.0	0.5
23	-	-2541.8	-1596.9	-130.4	4.8	2.4	0.7



Figur 1: Skjermdump fra PhotoModeler

### Kamerakalibrering - konstanter beregnet:

Camera1: Canon D450 JGG	
Focal Length	K1 - radial distortion 1
Value: 18.542617 mm	Value: 5.318e-004
Deviation: Focal: 0.002 mm	Deviation: K1: 2.2e-006
Xp - principal point x	K2 - radial distortion 2
Value: 11.194518 mm	Value: -1.187e-006
Deviation: Xp: 0.003 mm	Deviation: K2: 2.0e-008
Yp - principal point y	K3 - radial distortion 3
Value: 7.313061 mm	Value: 0.000e+000
Deviation: Yp: 0.003 mm	P1 - decentering distortion 1
Fw - format width	Value: -2.998e-005
Value: 22.254661 mm	Deviation: P1: 2.2e-006
Deviation: Fw: 8.7e-004 mm	P2 - decentering distortion 2
Fh - format height	Value: -2.248e-005
Value: 14.833600 mm	Deviation: P2: 2.3e-006

Tillegg C

Kalmanfilter-PV (m-kode)

```
1 %% Are Jo Næss
2 % PV_Attitude-filter
3 % %State: Xr = [Xeb_e Veb_e_x Tilt_x bias_gyro_x ...y ...z]'
4 %
5 clear all
6 format long g
7 addpath('input data','funksjoner');
8
9 %% Start: Tilfelle 1 - Bogstadveien
10 % yy = 2011; mm=9; dd=6; hh=10; mmin=51; sec=1;
11 % % Slutt:
12 % yy_end=2011; mm_end=9; dd_end=6; hh_end=10; mmin_end=52;sec_end=59;
13
14 % Start: Tilfelle 2 - Vika
15 yy = 2011; mm=9; dd=6; hh=9; mmin=21; sec=01;
16 % Slutt:
17 yy_end=2011; mm_end=9; dd_end=6; hh_end=9; mmin_end=22; sec_end=59;
18
19
20 %% Koorigerer med 15 skuddsekund - GPS tid i terrapos-filer
21 sec_gps = sec+15;
22 sec_end_gps = sec_end+15;
23 mmin_gps = mmin;
24 mmin_end_gps = mmin_end;
25 hh_gps = hh;
26 hh_end_gps = hh_end;
27 if sec_gps >= 60
28     sec_gps = sec_gps-60;
29     mmin_gps = mmin_gps+1;
30     if mmin_gps >= 60
31         mmin_gps = mmin_gps-60;
32         hh_gps = hh_gps+1;
33     end
34 end
35 if sec_end_gps >= 60
36     sec_end_gps = sec_end_gps-60;
37     mmin_end_gps = mmin_end_gps+1;
38     if mmin_end_gps >= 60
39         mmin_end_gps = mmin_end_gps-60;
40         hh_end_gps = hh_end_gps+1;
41     end
42 end
43
44 %% ---- System data ----
45 hz = 1; % 1 hz-datafil
46 dt = 1; % tidsinterval
47 oie_e = [0; 0; 7292115e-11]; % rad/s (jordrotasjon)
48 geg_g = [0; 0; 9.808]; % tyngdekraft (m/s2)
49
50 % WGS84 ellipsoide
51 a = 6378137;
52 b = 6356752.314245;
53 % Projeksjon - UTM sone 32
54 lat0 = 0;
55 lon0 = deg2rad(9);
56 scale = 0.9996;
```

```
57 fnorth= 0;
58 feast = 500000;
59
60 %% ---- Laster inn datafiler ----
61 % Laster inn OBU-data
62 [data_1Hz] = load_1Hz_data('POS_IMU_1Hz_OBU.txt', yy, mm, dd, hh, ...
63     mmin, sec, yy_end, mm_end, dd_end, hh_end, mmin_end, sec_end);
64
65 [num_points, m] = size(data_1Hz);
66
67 % Laster inn OBU_SIM data - simulert datasett
68 [platformC1Q]=load_platformC1Q9999...
69     ('PlatformC1Q9gnssN0smoother.txt', yy, mm, dd, hh_gps, mmin_gps...
70     , sec_gps, yy_end, mm_end, dd_end, hh_end_gps, ...
71     mmin_end_gps, sec_end_gps);
72 %% ---- Organiserer observasjoner ----
73 % Akselerometer
74 g = 9.808;
75 ax = (data_1Hz(:,11)/1000)*g; % mg -> m/s2
76 ay = (data_1Hz(:,12)/1000)*g;
77 az = (data_1Hz(:,13)/1000)*g;
78
79 % Gyro
80 gx = deg2rad(data_1Hz(:,14))./1000; % mdeg/s -> Rad/s
81 gy = deg2rad(data_1Hz(:,15))./1000;
82 gz = deg2rad(data_1Hz(:,16))./1000;
83
84 % GNSS-observasjoner
85 for i = 1:num_points;
86     % Observert posisjon Lengde/Bredde/Høyde (GNSS)
87     [Lat_obs(i,1), Lon_obs(i,1)] = TM2geod(a,b,platformC1Q(i,7), ...
88         platformC1Q(i,8), lat0, lon0, scale, fnorth, feast);
89
90     % Observert ECEF posisjon (GNSS)
91     [X_e_gnss(i,1), Y_e_gnss(i,1), Z_e_gnss(i,1)] = geod2ecef(a,b, ...
92         Lat_obs(i,1), Lon_obs(i,1), platformC1Q(i,10));
93 end
94 Hght_obs = platformC1Q(:,9); % m
95
96 V_b_gnss = data_1Hz(:,6); % m/s
97 Heading_obs = deg2rad(data_1Hz(:,7)); % deg -> rad
98
99 HDOP_obu = data_1Hz(:,24);
100 num_satelitter_obu = data_1Hz(:,23);
101
102 %% Laster inn referansefil:
103 % Referansefil fra Terrapos
104 [ref_pos, start_row_pos, stop_row_pos] = load_data_terrapos...
105     ('Terrapos_reference_complete.txt', yy, mm, dd, hh_gps, mmin_gps, ...
106     sec_gps, yy_end, mm_end, dd_end, hh_end_gps, mmin_end_gps, sec_end_gps);
107
108 %% ---- Initielle data ----
109 Xeb_e_ins = X_e_gnss(1);
110 Yeb_e_ins = Y_e_gnss(1);
111 Zeb_e_ins = Z_e_gnss(1);
112
```

```
113 % Body frame -> NED
114 initiell_roll = deg2rad(ref_pos(1,15)); %deg->rad
115 initiell_pitch = deg2rad(ref_pos(1,16)); %deg->rad
116 initiell_heading = deg2rad(ref_pos(1,17)); %deg->rad
117 % Setter initiell matrise
118 Cb_g = Rz(initiell_heading)*Ry(initiell_pitch)*Rx(initiell_roll);
119
120 %% ---- Initielle filter parametre ----
121 % Initiell state vektor
122 Xr = [X_e_gnss(1) 0 initiell_roll 0 ...
123 Y_e_gnss(1) 0 initiell_pitch 0 Z_e_gnss(1) 0 initiell_heading 0]';
124
125 % Apriori covarians matrise
126 P = [10^2 0 0 0
127 0 5^2 0 0
128 0 0 3^2 0
129 0 0 0 3^2];
130 P = kron(eye(3),P);
131
132 % Varians til observasjoner
133 R = diag([1^2 0.03^2]); % Posisjon %Gyro
134 R = kron(eye(3),R);
135
136 % Prosesstøy
137 Sv = 1; % Prosesstøy fart
138 Sw = 0.01; % Prosesstøy gyro
139 % Hvit støy-prosess
140 Q_pv = Sv^2*[1/3*dt^3 1/2*dt^2
141 1/2*dt^2 dt];
142 Q_gyro = Sw^2*[1/3*dt^3 1/2*dt^2
143 1/2*dt^2 dt];
144 Q = [ [Q_pv] zeros(length(Q_pv),length(Q_gyro))
145 zeros(length(Q_gyro),length(Q_pv)) Q_gyro];
146 Q = kron(eye(3),Q);
147
148 % Design matrise
149 H = [1 0 0 0
150 0 0 0 1];
151 H = kron(eye(3),H);
152
153 % System størrelse
154 [n,e] = size(H);
155
156 % Transisjons matrise
157 phi = [1 dt 0 0
158 0 1 0 0
159 0 0 1 dt
160 0 0 0 1];
161 phi = kron(eye(3),phi);
162
163 filt=[Xr'];
164 meas=[];
165 gain=[];
166 cov=[];
167 res=[];
168 ins=[];
```

```

169 velocity_gnss=[];
170
171 k = 1;
172 i = 0;
173 for i = 1:num_points-1;
174
175     [lat,lon,hght]=ecef2geod(a,b,Xr(1),Xr(5),Xr(9));
176     % Transformasjon: ECEF til NED-frame
177     Ce_g = [-sin(lat)*cos(lon) -sin(lat)*sin(lon) cos(lat);
178            -sin(lon) cos(lon) 0;
179            -cos(lat)*cos(lon) -cos(lat)*sin(lon) -sin(lat)];
180
181     % Initiell - Bodyramme -> ECEF
182     if i == 1; % - regnes ut første gang, og så nederst i løkke
183         Cb_e = Ce_g'*Cb_g; % Initiell matrise - oppdateres av gyroene
184     end
185
186     %% ---- Observasjoner ----
187     % Gyro
188     oib_b = [gx(i);gy(i);gz(i)]; %radianer
189     oib_b = Rz(pi)*oib_b; % Roterer observasjoner om z-aksen
190     % pga. feilmontert gyro (180grader)
191
192     % Akselerometer
193     % fib_b=[accx(i);accy(i);accz(i)];
194
195     %% ---- Kalmanfilter ----
196
197     % Tidsoppdatering
198     Xr = phi*Xr;
199     P = phi*P*phi'+Q;
200
201     % Oppdaterer Cb_e
202     oeb_b = oib_b-Cb_e'*oie_e; % body_ecef korrigert for jordrotasjon
203     Oeb_b = crossm(oeb_b);
204     Cb_e_dot = Cb_e*Oeb_b; % den deriverte av Cb_e
205     Cb_e_dot = Cb_e_dot*(eye(3) - crossm([Xr(4); Xr(8); Xr(12)]));
206
207     % Beregner Cb_e ved tid t=t+1
208     Cb_e = Cb_e+Cb_e_dot*dt; % Cb_e_hat
209     Cb_e = normalize(Cb_e); % Normaliserer for å ha determinant=1
210
211
212     Veb_b_gnss = [(norm([ref_pos(i,18) ref_pos(i,19) ref_pos(i,20)])...
213                 + norm([ref_pos(i+1,18) ref_pos(i+1,19) ref_pos(i+1,20)]))/2
214                 0
215                 0];
216     Veb_e_gnss = Cb_e*Veb_b_gnss;
217
218     % Legger skalar fart fra referansen inn i state
219     Xr(2) = Veb_e_gnss(1);
220     Xr(6) = Veb_e_gnss(2);
221     Xr(10) = Veb_e_gnss(3);
222
223     % Beregner posisjon i ecef ved tid t=t+1 (free inertial)
224     reb_e = Veb_e_gnss*dt*hz;

```

```

225
226 % Beregner INS-posisjon i ecef
227 Xeb_e_ins = Xeb_e_ins+reb_e(1);
228 Yeb_e_ins = Yeb_e_ins+reb_e(2);
229 Zeb_e_ins = Zeb_e_ins+reb_e(3);
230 [lat,lon,hght_ins(i,1)] = ecef2geod...
231 (a,b,Xeb_e_ins,Yeb_e_ins,Zeb_e_ins);
232 [north_ins(i,1),east_ins(i,1)]=geod2TM...
233 (a,b,lat,lon,lat0,lon0,scale,fnorth,feast);
234
235
236 %% Koordinatbasert integritetsmål - GNSS- vs INS-posisjon
237 % Beregner INS-posisjon
238 % 3d
239 koord_fremskrevet = [X_e_gnss(i)+reb_e(1)
240                     Y_e_gnss(i)+reb_e(2)
241                     Z_e_gnss(i)+reb_e(3)];
242 Test_diff_3d = koord_fremskrevet - [X_e_gnss(i+1); ...
243                                   Y_e_gnss(i+1); Z_e_gnss(i+1)];
244 % Koordinatbasert integritet - 3D (skaleres med 10x)
245 Test_3d_sqrt(i) = sqrt(Test_diff_3d(1)^2+...
246                       Test_diff_3d(2)^2+Test_diff_3d(3)^2) *10;
247
248 % Grunnriss, 2D -
249 % GNSS
250 [lat_test_gnss,lon_test_gnss,hght_test_gnss] = ...
251 ecef2geod(a,b,X_e_gnss(i+1),Y_e_gnss(i+1),Z_e_gnss(i+1));
252 [north_test_gnss,east_test_gnss]=geod2TM...
253 (a,b,lat_test_gnss,lon_test_gnss,lat0,lon0,scale,fnorth,feast);
254 % INS
255 [lat_test_ins,lon_test_ins,hght_test_ins] = ecef2geod(a,b,...
256 koord_fremskrevet(1),koord_fremskrevet(2),koord_fremskrevet(3));
257 [north_test_ins,east_test_ins]=geod2TM...
258 (a,b,lat_test_ins,lon_test_ins,lat0,lon0,scale,fnorth,feast);
259
260 Test_diff_2d = [north_test_ins - north_test_gnss % nord
261               east_test_ins - east_test_gnss]; % øst
262 % Koordinatbasert integritet - 2D (skaleres med 10x)
263 Test_2d_sqrt(i) = sqrt(Test_diff_2d(1)^2+Test_diff_2d(2)^2) *10;
264
265 %% ---- Kalmanfilter ----
266 % ---- Foreløpige verdier ----
267 Xr0 = [Xr(1) Xr(5) Xr(9)]';
268 oib_b0 = Cb_e'*oie_e + Cb_e*crossm([Xr(3) Xr(7) Xr(11)]')*...
269                                   oie_e+[Xr(4);Xr(8);Xr(12)];
270 Xr(4) = 0; Xr(8) = 0; Xr(12) = 0;
271
272 % Målt - foreløpig
273 z = [X_e_gnss(i+1) - Xr0(1)
274      oib_b(1) - oib_b0(1)
275      Y_e_gnss(i+1) - Xr0(2)
276      oib_b(2) - oib_b0(2)
277      Z_e_gnss(i+1) - Xr0(3)
278      oib_b(3) - oib_b0(3)];
279
280 % Kalman gain





```



```
281     K = P*H'*inv(H*P*H'+R);
282
283     % Måleoppdatering
284     Xr = Xr+K*z; % Xr + dXr
285
286     % Oppdaterer covarians
287     P = (eye(e)-K*H)*P*(eye(e)-K*H)'+K*R*K';
288
289     % Lagrer resultater
290     filt = [filt;Xr'];
291     meas = [meas;z'];
292     gain = [gain;diag(K)'];
293     cov = [cov;diag(P)'];
294     res=[res;(z-H*Xr)'];
295     ins = [ins; Xeb_e_ins Yeb_e_ins Zeb_e_ins];
296     velocity_gnss = [velocity_gnss; Veb_e_gnss'];
297 end
```

# Figurer

2.1	Observasjonsmodell - konsistent system til venstre og inkonsistent system til høyre. $R(A)$ er rommet utspent av kolonnevektorene til $A$ . . . . .	7
2.2	Observasjonsmodell - konsistent system ved introdusering av feilvektor $r$ . . . . .	7
2.3	Forskjellen mellom presisjon og nøyaktighet. De svarte prikkene er observasjoner, og det røde krysset er den sanne posisjonen: (a) God presisjon og god nøyaktighet - (b) Dårlig presisjon og god nøyaktighet - (c) God presisjon og dårlig nøyaktighet - (d) Dårlig presisjon og dårlig presisjon. . . . .	9
3.1	Koblingen mellom ECEF (e-ramme), navigasjonsramme (g-ramme) og inertial (i-ramme) [7]. . . . .	14
3.2	Bodyramme. . . . .	15
3.3	Plattform - målt fotogrametrisk. . . . .	16
4.1	Tradisjonelt gyroskop i gimbal oppheng [11] . . . . .	22
4.2	Akselerometer[10] . . . . .	23
5.1	Målebil med mottakerutstyr i lasterommet, og IMU og antenner på taket. . . . .	32
5.2	Oversikt over kjøreturen. (Google Earth [18]) . . . . .	34
6.1	Simulering av GNSS-kodemottakere: Referansekjøring:(—) GPS/GLONASS-kodemottaker:(—) GPS-kodemottaker:(—) (3Dmodell og satelittbilde: [18] . . . . .	40

6.2	Stillbilde fra videoopptaket i Bogstadveien. . . . .	41
6.3	Tilfelle 1 - Bogstadveien - Antall satellitter og DOP-verdier. . .	41
6.4	Tilfelle 2 - Vika. Referansekjøring  (Google Earth [18]) . . .	42
6.5	Tilfelle 2 - Vika - Antall satellitter og DOP-verdier. . . . .	43
7.1	Koordinatbasert integritetsmål . . . . .	46
7.2	Optimalt eksempel (Tilfelle 1 - Bogstadveien): koordinatbasert integritetsmål sammenlignet med ytre pålitelighet fra $OBU_{SIM}^{Pseudo}$ . . . . .	47
7.3	Optimalt eksempel (Tilfelle 2 - Vika): koordinatbasert integritetsmål sammenlignet med ytre pålitelighet fra $OBU_{SIM}^{Pseudo}$ . . . . .	48
7.4	Tilfelle 1, Bogstadveien - Posisjon, referanse mot free inertial navigation. . . . .	50
7.5	Tilfelle 1, Bogstadveien. . . . .	51
7.6	Tilfelle 2, Vika. . . . .	53
7.7	Ytre pålitelighet mot standardavvik i grunnriss. . . . .	57
A.1	Simulering av GNSS-kodemottakere: Referansekjøring:  GPS/GLONASS-kodemottaker:  GPS-kodemottaker:  - ) (3Dmodell og satelittbilde [18]) . . . . .	65

# Tabeller

4.1	Sammenligning INS - GNSS [8] . . . . .	21
4.2	IMU klassifisering - teknologi og ytelse. Utdrag fra K.Gade [10]	21
4.3	Observasjonsligninger . . . . .	27
4.4	Varians og støybidrag . . . . .	29
6.1	Simulering av GNSS-kodemottakere. Standardavvik beregnet i forhold til referansen. . . . .	39