

**UNIVERSITETET I OSLO**  
**Fysisk institutt**

**MEMS - basert  
nordsøker**

Masteroppgave

Anders Berge

28. mai 2012





# Forord

Denne oppgaven er skrevet som et avsluttende mastergradsoppgave i elektronikk og datateknologi ved Fysisk Institutt på Universitetet i Oslo. Jeg har underveis i studiet spesialisert meg innenfor Kybernetikk. Oppgaven er gjort i samarbeid mellom Forsvarets forskningsinstitutt (FFI) og Universitetsenteret på Kjeller (UNIK). Arbeidsted for skrivingen har vært UNIK mens labforsøk er gjennomført hos FFI. Jeg ønsker herved å takke min oppdragsgiver og veileder Ørnulf Kandola for samarbeidet. En takk rettes også til forsker Anders Rødningsby ved FFI for god hjelp under gjennomføringen av labforsøk.

Til slutt vil jeg takke min foreleser Oddvar Hallingstad ved UNIK for gode forelesninger og hjelp under gjennomføringen av denne oppgaven.

Kjeller, 28. mai 2012

---

Anders Berge





# Sammendrag

Denne oppgaven er gjort for å undersøke om det er mulig å lage en MEMS-basert nordsøker. De siste årene har det vært en veldig utvikling på denne fronten og jeg har derfor undersøkt om MEMS sensorene er blitt gode nok til å lage en nordsøker som har god nok nøyaktighet til å brukes til militære formål. I oppgaven har jeg tatt utgangspunkt i gyroene STIM 210 og STIM202, fra Sensoror. Jeg har utviklet en matematisk modell for å beskrive støykomponentene i denne vha. Allanvarians. Denne modellen har jeg brukt til å gjøre simmuleringer av en nordsøker basert på måling av jordrotasjonen i to retninger, orientert i horisontalplanet. Videre er det implementert en modell av et akselerometer for å måle skjevstillingen til gyroen i forhold til horisontalplanet. Dette er gjort for å spare tid med tanke på horisontaloppretting av kompasset.

Til slutt har jeg brukt Monte Carlo simmuleringer for å finne ut hvilke parametere som har størst innvirkning på resultatet. Basert på dette kommer jeg frem til en konklusjon for hvordan kompasset bør stilles opp, hvor lenge man må måle og hvilke gyroparametere som har størst innvirkning på det endelige resultatet.



# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
1.1	Inndeling av oppgaven . . . . .	2
1.2	Nomenklatur . . . . .	3
<b>2</b>	<b>Matematisk bakgrunnstoff</b>	<b>5</b>
2.1	Normalfordeling, forventning og varians . . . . .	5
2.1.1	Allanvarians . . . . .	7
2.2	Støykilder . . . . .	9
2.2.1	Hvitstøy . . . . .	10
2.2.2	Farget støy . . . . .	11
2.3	Monte Carlo simuleringer . . . . .	12
2.4	Vektorer og Retningskosinmatriser . . . . .	13
2.4.1	Vektorer . . . . .	13
2.4.2	Euler vinkler . . . . .	14
<b>3</b>	<b>MEMS - teknologi</b>	<b>15</b>
3.1	Hva er MEMS? . . . . .	15
3.2	MEMS Gyroskop . . . . .	16
3.2.1	Coriolis effekten . . . . .	16
3.2.2	Måleprinsipp . . . . .	17
3.3	MEMS kontra ordinære gyroskop. . . . .	19
3.4	STIM210 . . . . .	20
<b>4</b>	<b>Modell av sensorer</b>	<b>23</b>
4.1	Støykilder i Gyroskopet . . . . .	23
4.1.1	Støytyper i STIM gyroene . . . . .	23
4.1.2	Angular Random Walk - ARW . . . . .	25
4.1.3	Rate Random Walk - RRW . . . . .	27
4.1.4	Sammenheng mellom ARW, RRW og parameterne i datatablaget . . . . .	29
4.2	Støytyper i Akselrometeret . . . . .	31

4.3	Matematisk modell av sensorene . . . . .	32
4.3.1	Modell av Gyroskopet . . . . .	32
4.3.2	Modell av Akselerometeret . . . . .	36
4.4	Forventning og varians for gyroen . . . . .	37
<b>5</b>	<b>Simuleringsmodell av gyrokompass.</b>	<b>39</b>
5.1	Innføring av koordinatsystemer . . . . .	40
5.1.1	Rotasjonsmatriser . . . . .	42
5.2	Simulering av målinger . . . . .	44
5.2.1	Gravitasjonsmålinger . . . . .	44
5.2.2	Gyromålinger . . . . .	45
5.3	Estimering av $\hat{\theta}$ , $\hat{\phi}$ og $\hat{\psi}$ . . . . .	46
5.4	Implementering av modellen i MatLab . . . . .	48
5.5	Sammenligning av simulatoren med forsøk på ratebord . . . . .	49
<b>6</b>	<b>Simuleringsresultater</b>	<b>53</b>
6.1	Optimal måletid i hver posisjon. . . . .	53
6.2	Optimal initiell innstilling av kompasset . . . . .	54
6.2.1	Initiell $\psi$ retning . . . . .	54
6.2.2	Betydningen av skjevstilling . . . . .	58
6.2.3	Nøyaktighet som funksjon av breddegrad . . . . .	59
6.2.4	Feilbudsjett for sensorene . . . . .	60
<b>7</b>	<b>Konklusjon</b>	<b>63</b>
<b>8</b>	<b>Videre arbeid</b>	<b>65</b>
<b>A</b>	<b>Matlab kode</b>	<b>69</b>
A.1	Allanvariens.m . . . . .	69
A.2	gra2rad.m . . . . .	70
A.3	rad2gra.m . . . . .	70
A.4	gyromodel.m . . . . .	70
A.5	gyrosim.m . . . . .	71
A.6	akssim.m . . . . .	72
A.7	mccompass.m . . . . .	73
A.8	init.m . . . . .	75
A.9	mctid.m . . . . .	76
A.10	minpsi.m . . . . .	79
A.11	simintvinkel.m . . . . .	80
A.12	simbreddegrad.m . . . . .	82
A.13	simskråstilling.m . . . . .	84

<i>INNHOLD</i>	vii
A.14 feilbudsjett.m . . . . .	86
<b>B Sammendrag av datablader for STIM202 og STIM210</b>	<b>89</b>



# Figurer

2.1	Normalfordelingen med standardavvik.[6]	6
2.2	Eksempel på allanvarians-plot av en prosess bestående av hvit støy og Random Walk.	9
2.3	En hvitstøy simulert i 100s ved 50Hz, med tilhørende autokorrelasjonsfunksjon.	10
2.4	1. ordens markov prosess simulert i 100s, med tilhørende autokorrelasjon. Variansen til hvitstøyen er 1 og korrelasjonstiden er 2s.	12
3.1	A: Kasting av ball på roterende karusell sett fra system festet til den roterende karusellen. B: Samme forsøket sett fra jordfast referanseramme.	16
3.2	Illustrasjon av gyroens virkemåte. Figur A) To masser settes i svingninger med $180^\circ$ faseforskjell langs 1. Man måler kapasitansen mellom M og L (2). Når gyroen står i ro vil denne være konstant. Figur B) Når gyroen roteres om P vil Coriolis effekten skape vibrasjoner i M langs 3. Dette fører til at kapasitansen mellom M og L (2) vil følge en sinusfunksjon. Amplituden til funksjonen er direkte funksjon av vinkelfarten.	18
3.3	Venstre: Gyro basert på roterende masse. Hjulet i midten vil, når det roterer hurtig, holde seg stabilt i treghetsrommet når aksene rundt roteres. Vinkel kan da måles relativt til dette. Høyre: Gyro basert på laser. En laserstråle sendes i hver retning i trekanten. Når trekanten roteres i planet vil lengden laserlyset må gå bli lenger med rotasjonsretning enn mot.	19
3.4	STIM210 er den nye høypresisjonsgyroen fra det norske selskapet Sensoror.[12]	21
4.1	De 7 forskjellige støykildenes representasjon i et Allanvariansplot[1]	24
4.2	Allanvarians for Stim 202. Asymptoter for ARW og RRW er inntegnet.	25

4.3	Allanvarians-plot for de forskjellige støybidragene i Gyromodellen . . . . .	26
4.4	Random walk prosess simulert 10 ganger med $q_\beta=1$ . Utviklingen i standardaviket er også vist. . . . .	28
4.5	Utdrag fra databladet til STIM 210. . . . .	29
4.6	Sentrale parametre fra databladet representert i et Allanvarians plot. . . . .	30
4.7	Målt og simulert signal for STIM202, simulert i 5 min. . . . .	34
4.8	Allanvarians plot for simulert og målt gyrosignal basert på en måleserie over 10000s, samlet ved 1000 Hz . . . . .	35
5.1	Illustrasjon av måleprinsipp for gyrokompass med vippe plassert i horisontal stilling. Jordrotasjonskomponentene $\omega_1$ og $\omega_2$ måles og $\psi$ estimeres basert på disse. . . . .	40
5.2	Skisse av et tenkt oppsett av kompasset sett fra forskjellige sider. Vinklene $\theta$ , $\phi$ og $\psi$ , samt de forskjellige koordinatsystemene er tegnet inn. Merk: Koordinatsystemene definerer egentlig vektorrom og vil derfor ikke ha forskjellig origo, de er imidlertid tegnet med forskjellig plassering for å illustrere bedre. . . . .	41
5.3	Illustrasjon av projeksjonen av $\Omega$ ned i $\mathcal{F}^r$ . . . . .	45
5.4	GUIen til simulatoren. . . . .	49
5.5	Bildet viser oppsettet av gyroen plassert på ratebordet. Til venstre ses styreenheten for ratebordet der nordvinkelen kan stilles inn. . . . .	50
5.6	Nordsøk gjort på lab sammenlignet med simulerte målinger. . . . .	51
6.1	Allanstandardavik for simulert STIM210 basert på måleserie på 100000s. . . . .	54
6.2	Monte-Carlo simulering av gyrokompass med varierende midlingstid fra 20-1000 sek. . . . .	55
6.3	Figuren viser forventet jordrotasjonskomponent som funksjon av $\psi$ . Løsning av ulikheten over er merket inn og angir største initielle $\psi$ som er gunstig for å få gode målinger. . . . .	56
6.4	Øverst: forventningsverdi og standardavik for kompasset som funksjon av initiell vinkel. Hver 1000 montecarlo simulering er plottet for illustrasjon. Nederst: Kurvetilpasset standardavik hvor minimum er plottet inn, i tillegg til beregningspunktene. . . . .	57
6.5	Simulering av $\psi$ som funksjon av skjevstilling, både med å uten akselerometer. . . . .	58
6.6	Simulering av $\psi$ som funksjon av breddegrad. . . . .	59



# Tabeller

3.1	Sammenligning av vesentlige data for de to modellene. . . . .	20
4.1	Støytyper representert i gyroer. . . . .	24
6.1	Parametere brukt ved simulering av $\psi$ som funksjon av $\theta$ . . . . .	59
6.2	Parametere brukt ved simulering av $\psi$ som funksjon av breddegrad. . . . .	60
6.3	Parametere brukt ved feilbudsjett. . . . .	61
6.4	Resultat av feilbudsjett. . . . .	61



# Kapittel 1

## Innledning

Å bestemme sann nordretning raskt, effektiv og med god nøyaktighet er et stort problem i militære sammenhenger. Til dette har det lenge vært brukt kompass som baserer seg på målinger av jordrotasjonen vha. et gyroskop. De siste årene har det vært en rivende utvikling av gyroskop basert på MEMS teknologi. I denne oppgaven har jeg derfor undersøkt om et slikt MEMS gyroskop vil være i stand til å brukes i et gyrokompass. Fordelene ved en måling basert på en MEMS-gyro vil være mange som feks. fysisk størrelse, robusthet og pris for å nevne noen.

Denne oppgaven vil i grove trekk omhandle følgende temaer:

- Innføring i MEMS teknologi spesielt mot treghetsensorer
- Modell av MEMS treghetsensorer
- Simulering av gyrokompass basert på jordrotasjonsmålinger i to antipodale posisjoner i ca. øst-vest retning.
- Analyse av faktorer som påvirker målenøyaktigheten for dette kompasset.

I forbindelse med oppgaven har jeg utviklet et Matlabprogram som kan brukes til å beregne hvordan MEMS sensorer med gitte spesifikasjoner vil prestere i et gyrokompass. Dette programmet ser jeg for meg kan brukes i fremtiden for å raskt få et inntrykk av om nye sensorer på markedet vil være aktuelle for Gyro-kompassering.

## 1.1 Inndeling av oppgaven

Oppgaven er delt opp i kapitler som følger en relativt logisk inndeling basert på de forskjellige punktene i oppgaveteksten. Under følger en liten oversikt over innholdet i de forskjellige delene.

**Kapittel 2: Matematisk bakgrunnstoff** Denne delen består av en del matematisk bakgrunnstoff som kan være nyttig å kunne for å lese rapporten.

**Kapittel 3: MEMS - teknologi** Denne delen består av en kort innføring i MEMS-teknologi. Måleprinsippet for MEMS-gyroer og akselerometere blir belyst, samt grunner for at det er ønskelig å bruke disse i nordsøking.

**Kapittel 4: Modell av sensorer** I dette kapitlet har jeg analysert forskjellige støybidrag og satt opp en modell for målingene til gyroen og akselerometeret. Disse blir også sammenlignet med målinger gjort i et forsøk med en fysisk sensor.

**Kapittel 5: Simuleringsmodell av Nordsøker:** Modellene som blir satt opp i kapittel 4 har jeg her brukt til å lage en simuleringsmodell av gyrokompasset. Jeg lager rutiner for å generere målinger og estimere nord fra disse.

**Kapittel 6: Usikkerhetsanalyse** Simuleringsmodellen blir her brukt til å simulere kompasset med forskjellige feilbidrag for å se på hvilke faktorer som har størst betydning for nøyaktigheten.

**Kapittel 7: Konklusjon** Jeg vil i dette kapitlet trekke en konklusjon på hvordan en MEMS sensor vil fungere i et gyrokompass.

**Kapittel 8: Videre arbeid** Til sist har jeg tatt med en del ideer om hva som kan undersøkes videre på området MEMS-gyrokompass.

## 1.2 Nomenklatur

Jeg bruker følgende enheter og benevninger i oppgaven:

$g$	Gravitasjonskonstanten = $9,81^m/s^2$
$\Omega$	Jordas rotsjonshastighet om hovedaksen = $15,04^\circ/h$
$\phi, \psi, \theta$	Euler vinklene Pitch, Roll og Yaw
$R_{a(i)}^b$	Retningskosinmatrise fra $\mathcal{F}^a$ til $\mathcal{F}^b$ . $i$ indikerer målingsnr. dersom matrisen har forskjellig verdi i måling 1 og 2.
$\underline{a}^b$	Vektoren $a$ representert i $\mathcal{F}^b$
$\beta$	Gyroens Rate Random Walk støy
$\gamma$	Gyroens Angular Random Walk støy
$v_i$	Hvitstøyen tilhørende prosess $i$
$Q, q_i$	Kovariansmatrise, Varians til prosess $i$
$Qd, q_{id}$	Kovariansmatrise, Varians til prosess $i$ , diskret versjon
$\omega_m$	Rotasjonsmåling fra gyroen
$\omega_s$	Sann rotasjonshastighet
$a_m^i$	Målt akselerasjon langs akse $i$
$a_s^i$	Sann akselerasjon retning $i$
$b$	Bias/Nullpunktsfeil
$\bar{\omega}_m$	Gjennomsnitt av måling $\omega_m$
$\hat{\psi}$	Estimert verdi
$c_\theta, s_\theta$	Forkortelse for $\cos(\theta)/\sin(\theta)$



# Kapittel 2

## Matematisk bakgrunnstoff

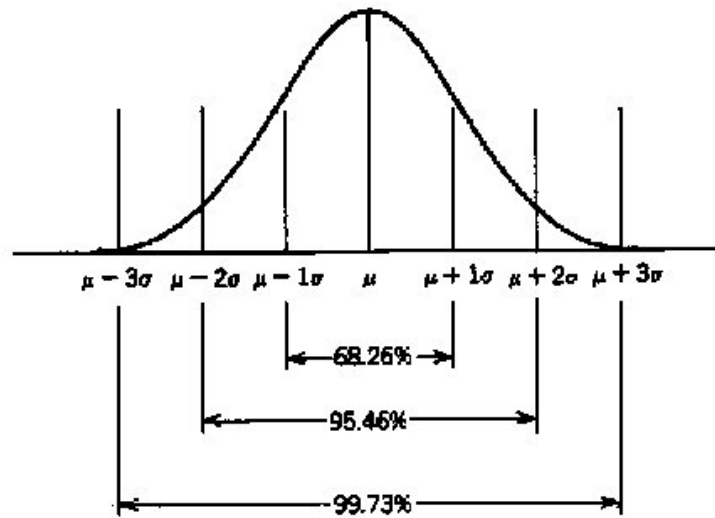
Dette kapittelet tar for seg en del av den matematikken som ligger til grunn for resten av rapporten. Her blir begreper og hvordan disse blir regnet ut forklart nærmere.

### 2.1 Normalfordeling, forventning og varians

I denne rapporten vil store deler av verdiene vi regner med være stokastiske. Dette er verdier som ikke kan bestemmes nøyaktig, men må beskrives med en stokastisk variabel,  $X$ .  $X$  er en variabel som kan inneholde verdier trukket fra sannsynlighetsfordelingen  $p(x)$ . Denne fordelingen er en funksjon som forteller noe om sannsynligheten for utfallet til  $X$ . I denne oppgaven vil det bli brukt stokastiske variabler som er trukket fra normalfordelingen. Dette er den mest utbredte fordelingen og er en god beskrivelse av mange av de tilfeldige prosessene som opptrer i naturen. Normalfordelingen er gitt ved formelen:

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}} \quad (2.1)$$

Fordelingen kjennetegnes av den karakteristiske klokkeformen. Forventningsverdien,  $\mu$ , beskriver her toppunktet til  $p(x)$ . Variansen gir spredningen. Roten av variansen er standardavviket. I normalfordelingen angir  $\pm 1 \cdot \sigma$  intervallet rundt  $\mu$  som vi med 68% sannsynlighet vil havne innefor når vi trekker fra fordelingen. Utvides intervallet til  $\pm 2 \cdot \sigma$  økes denne sannsynligheten til 95%. For denne fordelingen bruker vi ofte en forkortet notasjon når vi skal skrive den. Dersom  $v(t)$  er en støy beskrevet med normalfordelingen skrives dette ofte  $v(t) \sim N\{\mu, \sigma^2\}$ . De to parameterne  $\mu = E[X]$  og  $\sigma^2$  som omtales her er sentrale parametere her, som sier noe om forventet verdi og spredning. Disse parameterne er gitt ved:



Figur 2.1: Normalfordelingen med standardavvik.[6]

### Forventningsverdi, $\mu$

Forventningsverdien er den verdien vi kan vente oss som middelvei dersom vi trekker mange verdier fra fordelingen  $p(x)$ . Dersom  $x$  trekkes fra en kontinuerlig fordeling, kan forventningsverdien beregnes som integralet:

$$E[X] = \int_{-\infty}^{\infty} x \cdot p(x) dx \quad (2.2)$$

Dersom verdien vi regner med er en diskret variabel vil forventningsverdien være gitt ved:

$$E[X] = \sum_{i=1}^n x_i \cdot p(x_i) \quad (2.3)$$

der  $p(x_i)$  er sannsynligheten for at  $x_i$  skal inntreffe.

For en måleserie  $Z$ , bestående av  $k$  målinger kan forventningsverdien til målingen beregnes som:

$$\hat{E}[Z] = \frac{1}{k} \sum_{i=1}^k z_i. \quad (2.4)$$

Dersom  $k \rightarrow \infty$  vil  $\hat{E}[Z] \rightarrow E[Z]$ .



### Varians $\sigma^2$

Variansen forteller noe om spredningen til den stokastiske variabelen  $X$ . Dersom  $x$  trekkes fra den kontinuerlige fordelingen  $p(x)$  vil variansen være gitt ved integralet:

$$\text{Var}(X) = \int_{-\infty}^{\infty} (x - E[X])^2 \cdot p(x) dx \quad (2.5)$$

Dersom  $X$  er en diskret stokastisk variabel med  $n$  mulige utfall vil variansen være gitt ved:

$$\text{Var}(X) = \sum_{i=1}^n p_i \cdot (x_i - E[X])^2 \quad (2.6)$$

For en måleserie  $Z$ , bestående av  $k$  målinger der hvert sample er uavhengig av hverandre kan variansen estimeres ved formelen:

$$\hat{\sigma}(Z)^2 = \frac{1}{k-1} \sum_{i=1}^k (z_i - E[Z])^2 \quad (2.7)$$

Dersom  $k \rightarrow \infty$  vil  $\hat{\sigma}[Z] \rightarrow E[Z]$ .

#### 2.1.1 Allanvarians

Dersom måleserien vår består av flere forskjellige støytyper vil det være vanskelig å skille disse fra hverandre ved hjelp av ordinær variansberegning. Hvis for eksempel målingene inneholder støy med effektspekter:

$$\phi(\omega) = \frac{1}{\omega^\alpha} \quad \forall \alpha < -1 \quad (2.8)$$

vil den ordinære variansen divergere ettersom måleserien blir lengre. Det betyr at disse komponentene fullstendig vil dominere variansen når måleserien blir av en viss lengde. For å kunne identifisere andre støykomponenter i måleserien brukes en metode som kalles Allanvarians.

Allanvarians ble opprinnelig utviklet av David Allan under hans arbeid med å undersøke stabiliteten i Atomklokker[2]. Stabiliteten i disse klokkene viste seg å ikke kun være gitt av hvit støy, og dermed fikk han divergerende varians. For å finne de underliggende støykomponentene ble løsningen å dele måleserien opp i mindre grupper og se på den gjennomsnittlige spredningen til middelveiden i to etterfølgende grupper. Man vil da ved å endre på gruppelengden få frem forskjellige støykomponenter avhengig av gruppelengde. For eksempel, dersom disse gruppene er veldig små vil variasjonen i middelveidi mellom to og to grupper være dominert av høyfrekvent støy ettersom lavfrekvente støybidrag ikke vil føre til nevneverdig variasjon i middelveiden

til to etterfølgende grupper. Dersom gruppelengden er stor vil man derimot middle vekk høyfrekvent støy internt i hver gruppe, og variasjonen mellom to etterfølgende grupper vil være dominert av lavfrekvent støy.

Denne metoden for å identifisere støykomponenter kalles Allanvarians etter oppfinneren. Beregningen er relativt enkel og vises her med det stokastiske signalet  $x(t \rightarrow t + T)$ . Dette signalet deles opp i  $N$  grupper, hver med lengde  $\tau$ , der gruppe  $y_k$  inneholder dataene  $x(t_k) \rightarrow x(t_k + \tau)$ .  $t_k = t_{k-1} + \tau$ , dvs. det er ingen dødtid mellom settene. Vi beregner så middelverdien i hver gruppe som  $\bar{y}_k = E\{y_k\}$  Variansen mellom  $\bar{y}_k$  og  $\bar{y}_{k+1}$  er da gitt som:

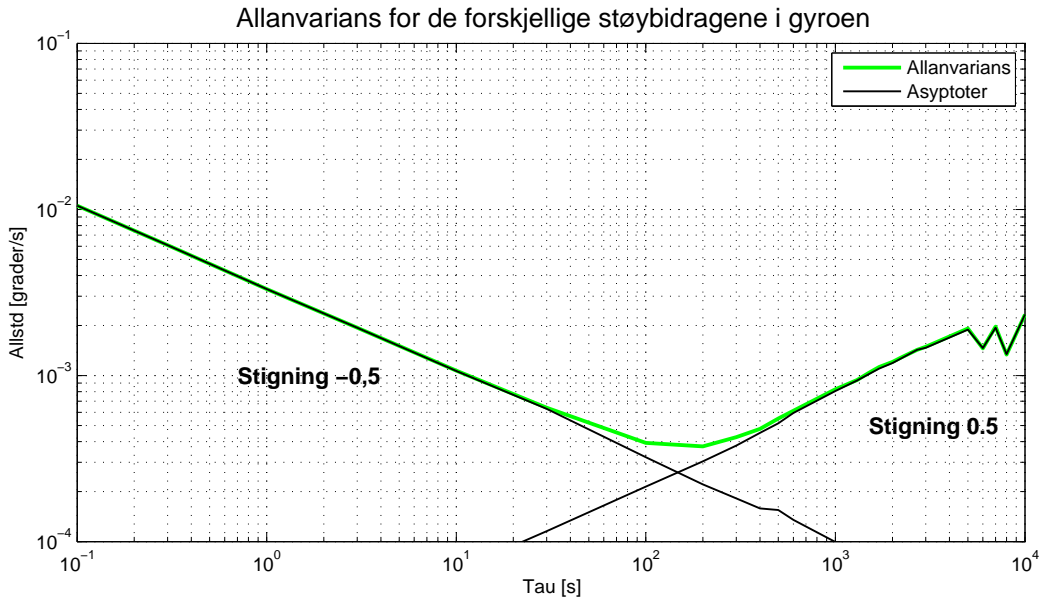
$$\begin{aligned}\sigma_k(y)^2 &= \sum_{i=k}^{k+1} (y_i - \frac{1}{2} \sum_{j=k}^{k+1} y_j)^2 \\ \sigma_k(y)^2 &= (y_k - \frac{1}{2}(y_k + y_{k+1}))^2 + (y_{k+1} - \frac{1}{2}(y_k + y_{k+1}))^2 \\ &= \frac{1}{2}(y_{k+1}^2 - 2y_{k+1}y_k + y_k^2) \\ &= \frac{1}{2}(y_{k+1} - y_k)^2\end{aligned}\quad (2.9)$$

Allanvariansen fremkommer ved å middle over disse  $N-1$  2 samplevariansene:

$$\begin{aligned}\sigma(x)_{AVAR}^2 &= \frac{1}{N-1} \sum_{k=1}^{N-1} \sigma_k(y)^2 \\ &= \frac{1}{N-1} \sum_{k=1}^{N-1} \frac{1}{2} (y_{k+1} - y_k)^2 \\ &= \frac{1}{2(N-1)} \sum_{k=1}^{N-1} (y_{k+1} - y_k)^2\end{aligned}\quad (2.10)$$

Dersom vi beregner Allanvariansen som funksjon av  $\tau$  kan vi plote  $\sigma(\tau)_{AVAR}$ . Vi vil da får en graf som kan se ut som figur 2.2. Dette er allanstandardaviket til en prosess bestående av hvit målestøy og en Random Walk prosess. Hvitstøyen har stigning  $-0,5$  i plottet og vi ser den dominerer for små  $\tau$ . Random Walk har stigning  $+0,5$  og vi ser denne dominerer for store verdier av  $\tau$ .

**Beregning av Allanvarians i Matlab** Når man skal beregne Allanvariansen i Matlab har man måleserien som en vektor  $x$  med samplingsfrekvens  $f = \frac{1}{T_s}$  der  $T_s$  er samplingstiden. Vi splitter dette datasettet opp i grupper  $y$



Figur 2.2: Eksempel på allanvarians-plot av en prosess bestående av hvit støy og Random Walk.

med størrelse  $\frac{\tau}{T_s}$ , og gjør beregningene beskrevet ovenfor med disse gruppene. Jeg har utviklet et program for å beregne Allanvariansen, matlabkoden for dette programmet finnes i Vedlegg A.1

Ved store datasett kan beregningene bli tidkrevende. For å forenkle beregningene kan man velge tauverdier som multipler av den forrige verdien, for eksempel  $\tau = [2, 4, 8, 16, 32, 64, \dots]$ .  $\bar{y}(\tau[i])$  kan da beregnes som:

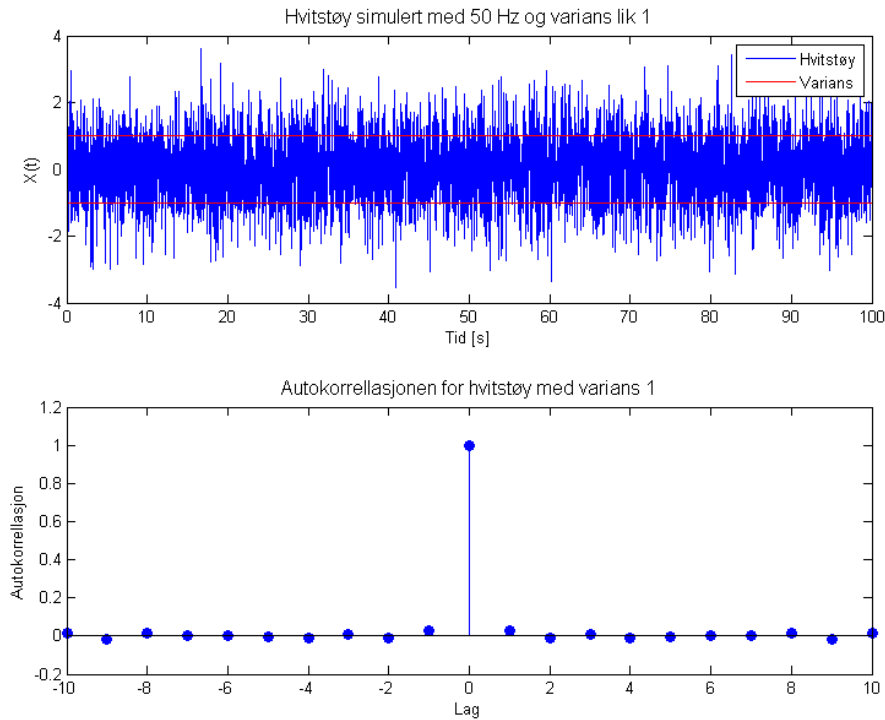
$$\begin{aligned}\bar{y}_k(\tau[1]) &= E\{y_k(\tau[1])\} \\ \bar{y}_k(\tau[i]) &= \frac{1}{2}(\bar{y}_k(\tau[i-1]) + \bar{y}_{k+1}(\tau[i-1]))\end{aligned}\quad (2.11)$$

Vi trenger da kun å middle over alle samplene i  $x$  en gang, og bruke disse middelveidene som grunnlag ved neste gruppering. Dette er ikke tatt hensyn til i programmet beskrevet i Vedlegg A.1.

## 2.2 Støykilder

Målinger fra sensorer som gyroer og akselerometre inneholder støy, både i form av måle- og prosess-støy. Denne støyen beskrives matematisk ved hjelp av to støytyper, hvit og farget støy. Måle- og prosess-støyen til sensorene blir da en funksjon av disse støytypene.

### 2.2.1 Hvitstøy



Figur 2.3: En hvitstøy simulert i 100s ved 50Hz, med tilhørende autokorrelasjonsfunksjon.

Dersom prosessen  $v(t)$  er totalt ukorrelert i tid kalles den for en hvitstøy prosess. Dette betyr at om vi kjenner verdien til  $v(t_1)$  vil vi ikke kunne si noe om  $v(t)$  der  $t \neq t_1$ , da disse er helt uavhengige av hverandre. Hvitstøy har derfor autokorrelasjonen[14]:

$$R_w(t, s) = Q\delta(t - s) \quad (2.12)$$

der  $Q$  er en skaleringskonstant som representerer variansen til  $v(t)$  og  $\delta(\tau)$  er dirac deltafunksjonen der  $\delta(\tau) = \infty$  for  $\tau = 0$  og 0 ellers.  $v(t)$  trekkes fra fordelingen  $F(v(t))$  med varians  $Q$ . Dersom  $F(v(t))$  er normalfordelt kalles  $v(t)$  for en gausisk hvitstøy.

Hvitstøy kjennetegnes også av at den er likt representert i hele frekvensspekteret, båndbredden er derfor uendelig. I praksis vil uendelig båndbredde være umulig å implementere, hvitstøy er derfor en teoretisk størrelse. Effekt-

spektert er gitt ved:

$$\begin{aligned}\phi(\omega) &= \int_{-\omega}^{\omega} R_{\omega}(\tau, s) e^{-j\omega\tau} d\tau \\ &= \int_{-\omega}^{\omega} Q\delta(\tau) e^{-j\omega\tau} d\tau \\ &= Q\end{aligned}\tag{2.13}$$

En illustrasjon av en hvitstøy med tilhørende autokorrelasjonen for en hvitstøyprosess med  $Q = 1$  er vist i figur 2.3. Den kontinuerlige hvitstøyen  $v(t)$  kan også beskrives diskret som  $v_k$ . Den diskrete prosessen har da autokorrelasjonen:

$$R_{k,s} = Q_d \delta_{k,s}\tag{2.14}$$

der  $Q_d$  er spektraltettheten til  $v_k$ , og  $\delta_{k,s}$  er Kroenecker-deltaet der  $\delta_{k,s} = 1$  for  $k = s$  og 0 ellers.

### 2.2.2 Farget støy

En annen type støy, som brukes for å karakterisere støyen i sensorer, er farget støy. Mens hvitstøy er totalt uavhengig av tidligere og fremtidige verdier, så er farget støy korrelert i tid og avhengig av tidligere verdier. Denne støytypen er ikke likt representert i alle frekvenser. Den gir oss derfor ikke et uniformt effekt spekter. Den fargede støyen kan modelleres med en 1. ordens Markov prosess. En 1. ordens Markov prosess  $x(t)$  kjennetegnes ved at tetthetsfunksjonen oppfyller følgende[5]:

$$F[x(t_k) | x(t_{k-1}, \dots, x(t_1))] = F[x(t_k) | x(t_{k-1})]\tag{2.15}$$

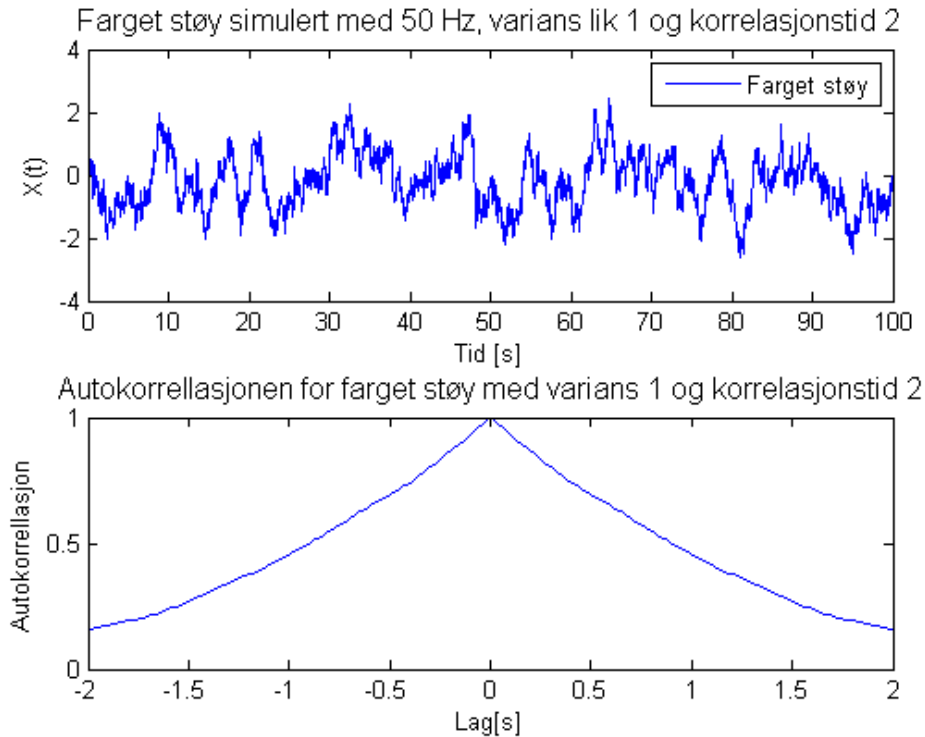
når  $t_k > t_{k-1}$  for alle k.

En 1. ordens markov prosess kan beskrives som en stoastisk differensiallikning drevet av hvit støy:

$$\dot{x}(t) = -\frac{1}{T}x(t) + v(t)\tag{2.16}$$

der  $v(t)$  er hvitstøy med varians  $Q$ .  $T^{-1}$  er korrelasjonstiden. Et eksempel på farget støy generert med en 1. ordens markov prosess er vist i figur 2.4 Når prosessen er normalfordelt kalles den en Gauss-Markov prosess og har autokorrelasjonen[3]:

$$R_x(\tau) = \sigma^2 e^{-T|\tau|}\tag{2.17}$$



Figur 2.4: 1. ordens markov prosess simulert i 100s, med tilhørende autokorrelasjon. Variansen til hvitstøyen er 1 og korrelasjonstiden er 2s.

hvor  $\sigma^2$  er variansen til prosessen og  $T^{-1}$  er korrelasjonstiden. Effektspekteret er da gitt som den fouriertransformerte av (2.17):

$$\begin{aligned}
 \phi(\omega) &= \int_{-\infty}^{\infty} R_x(\tau) e^{-j\omega\tau} d\tau \\
 &= \int_{-\infty}^{\infty} \sigma^2 e^{-T|\tau|} e^{-j\omega\tau} d\tau \\
 &= \frac{2\sigma^2 T}{\omega^2 + T^2}
 \end{aligned} \tag{2.18}$$

## 2.3 Monte Carlo simuleringer

Dersom man skal beregne forventningsverdi og varians for ulinneære systemer der det vil være vanskelig å sette opp ligninger for forventningsverdi og varians, kan Monte Carlo simuleringer benyttes. Prosessen simuleres da med samme initialbetingelser  $n$  ganger. Dette gir oss et sett med  $n$  monte

carlo kjøring,  $\{Z_k^1, \dots, Z_k^i, \dots, Z_k^n\}$ , der  $k$  angir måling nr.  $k$  i måleserie  $i$ . Et estimat for forventningsverdien i sample  $k$  kan da beregnes basert på (2.4)

$$\hat{E}[Z_k] = \frac{1}{n} \sum_{j=1}^n z_k^j \quad (2.19)$$

Et estimat for variansen kan beregnes basert på (2.7)

$$\hat{\sigma}(Z_k)^2 = \frac{1}{n-1} \sum_{j=1}^n \left( z_k^j - \hat{E}[Z_k] \right)^2 \quad (2.20)$$

Dersom det skal gjøres mange simuleringer og måleseriene er lange vil det bli svært store datamengder å lagre unna. Det er derfor en fordel å gjøre formlene for forventning og varians rekursive. Forventningsverdi og varians beregnes rekursivt vha. følgende formler [18]

$$\begin{aligned} \delta_k^N &= Z_k^i - \hat{E}[Z_k]^{N-1} \\ \hat{E}[Z_k]^N &= \hat{E}[Z_k]^{N-1} + \frac{1}{N} \delta_k^N, & \hat{E}[Z_k]^0 &= 0 \\ \hat{\sigma}^2(Z_k)^N &= \frac{N-2}{N-1} \hat{\sigma}^2(Z_k)^{N-1} + \frac{1}{N} (\delta_k^N)^2, & \hat{\sigma}^2(Z_k)^1 &= 0, \quad \hat{\sigma}^2(Z_k)^1 = 0 \end{aligned}$$

Når  $n \rightarrow \infty$  vil  $\hat{\sigma}(Z_k)^2$  og  $\hat{E}[Z_k]$  gå mot verdien for den sanne forventningen og variansen for sample  $k$ . Under simuleringer vil regnekapasiteten begrense hvor mange kjøring vi kan gjøre. Et veldig godt estimat oppnås dersom  $n$  ligger i størrelsesorden 1000 - 10000 kjøring.

## 2.4 Vektorer og Retningskosinmatriser

Dette er en kort oversikt over noen av de begrepene som senere blir introdusert i forbindelse med vektorer og koordinattransformasjoner. For mer utfyllende stoff om dette temaet henvises det til [17] hvor også mye av dette stoffet er hentet fra.

### 2.4.1 Vektorer

For å beskrive målinger i denne oppgaven brukes det vektorer. En vektor er en størrelse som har både retning og lengde. Geometriske vektorer defineres:

$$\vec{a} = \sum_{i=1}^3 a_i b_i \quad (2.21)$$

Hvor  $a_i$  er den algebraiske vektoren:

$$\underline{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (2.22)$$

$\{b_1, b_2, b_3\}$  definerer basisvektorene i rammen  $\mathcal{F}^b$ . I denne oppgaven brukes bare orthogonale basisvektorsett, og  $\{b_1, b_2, b_3\}$  beskriver da henholdsvis x, y og z-aksen i ramme  $\mathcal{F}^b$ . Alle rammene definerer vektorrom og superscriptet  $\mathcal{V}$  er derfor droppet.

### 2.4.2 Euler vinkler

Vektorene i denne oppgaven er definert i flere forskjellige rammer. For å transformere mellom rammene inføres linære operatorer. Dette er operatorer som transformerer vektorer fra  $\mathcal{F}^b$  til  $\mathcal{F}^a$  og benevnes  $C_b^a$ . En operator som foretar en rotasjon om en av basisvektorene benevnes med  $R_b^a(\theta)$  og kalles ofte retningskosinmatrise. Vi har 3 elementære retningskosinmatriser som definerer rotasjonen  $\theta$  om basisvektor 1, 2 og 3:

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix} \quad R_2(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \quad R_3(\theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

Ved rotasjoner om alle 3 basisvektorene (aksene) vil de to rammene kunne oppnå enhver stilling rommet i forhold til hverandre. Retningskosinmatrisen blir da:

$$R_b^a(\psi, \theta, \phi) = R_3(\psi) \cdot R_2(\theta) \cdot R_1(\phi) \quad (2.24)$$

Vinklene har her fått benevnningen  $\psi, \theta$  og  $\phi$ . Jeg vil i denne oppgaven kun benytte rotasjoner om nye akser og vinklene er da eulervinkler. Rotasjonen blir da følgende: Gitt systemet  $b$ . Vi roterer det en vinkel  $\psi$  rundt  $x^b$ . Dette gir oss et nytt system  $b'$ . Vi roterer deretter en vinkel  $\theta$  om  $y^{b'}$ . Dette gir oss systemet  $b''$ . Til sist roterer vi  $b''$  en vinkel,  $\phi$  om  $x^{b''}$  og systemet  $a$  fremkommer. Vi har da transformert fra  $b$  systemet via systemene  $b'$  og  $b''$  til systemet  $a$ . Denne rotasjons sekvensen kalles ofte 3-2-1 eulervinkler og brukes i denne oppgaven fordi den gir enkle uttrykk å regne med i dette problemet.



# Kapittel 3

## MEMS - teknologi

Dette kapitlet skal forsøke å belyse teknologien bak sensorene som er brukt i denne oppgaven. Kapitlet tar først for seg hva som ligger i begrepet MEMS. Videre vil måleprinsippet bak MEMS-gyroer bli belyst før det blir gitt en kort innføring i utviklingen av STIM210, gyroen tenkt brukt i denne oppgaven. Til sist er det litt teori om virkemåten for MEMS akselerometer som skal brukes til å estimere stilling i denne oppgaven.

### 3.1 Hva er MEMS?

MEMS er forkortelse for Mikro ElektroMekaniske Systemer (Micro-Electro-Mechanical Systems) og er en teknologien bak veldig små enheter som består av både elektriske kretser og mekaniske komponenter. Enheter som karakteriseres av begrepet MEMS er systemer som har komponenter i størrelsesorden  $1-100\mu\text{m}$ , og samlet størrelse for hele komponenten ligger i størrelsesorden  $20\text{-}\mu\text{m} - 1\text{mm}$ [7]. MEMS består av en del som er mekanisk. Denne kan inneholde sensorer, motorer og aktuatorer av forskjellig slag for å måle og kontrollere forskjellige ting. Den andre delen av MEMS systemet består av Mikro-elektronikk med mikroprosessorer og kretser for å styre de forskjellige pådragene samt hente inn og behandle data fra sensorene.

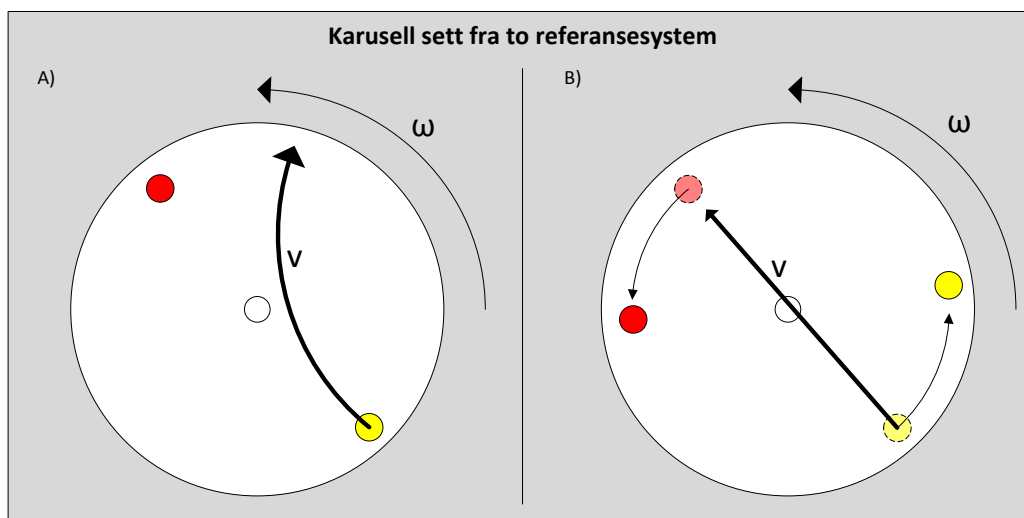
Produksjonen av slike små systemer er avansert og krever høyteknologisk utstyr. Produktene blir laget på forskjellige måter, både ved hjelp av forskjellige etse-teknikker og med svært nøyaktige kuttemaskiner. Mer om fremstilling av MEMS-chiper kan finnes i [7].

Teknologien regnes som et av de største fremskritt som er gjort på denne fronten og det ventes at teknologien i kombinasjon med nanoteknologi vil gi store muligheter i årene som kommer, feks. innenfor medisin og legemiddel-industrien.

## 3.2 MEMS Gyroskop

For å detektere endring i stillingen til ett objekt relativt til treghetsrommet kan et gyroskop benyttes. Slike gyroskop finnes i mange forskjellige utgaver, størrelser og nøyaktighetsklasser. I denne oppgaven blir det som nevnt brukt et MEMS gyroskop. Denne typen gyroer er utformet på flere forskjellige måter, men felles for alle er at de utnytter Coriolis effekten for å beregne vinkelfarten om gyroens sensitivitetsakse.

### 3.2.1 Coriolis effekten



Figur 3.1: A: Kasting av ball på roterende karusell sett fra system festet til den roterende karusellen. B: Samme forsøket sett fra jordfast referanseramme.

For å illustrere Coriolis effekten starter jeg med et eksempel. (Se illustrasjon i fig 3.1.) Tenk deg at du (Gul prikk) sitter på en karusell som snurrer mot klokken og kaster en ball til en venn (Rød prikk), som sitter på andre siden i karusellen. Du sikter rett på personen og kaster, men bommer grovt. Sett fra ditt ståsted sitter vennen din fortsatt rett foran deg og det vil se ut som om ballen ikke fulgte en rett bane, men svingte av mot høyre slik fig. 3.1A viser. I følge Newtons lover skulle vel ballen ha gått rett frem i den retningen du gav den fart?

Ser man på forsøket ovenfra vil det se ut som om ballen fulgte en rett linje, mens det var du og vennen din som flyttet seg som følge av at karusellen roterte mens ballen var i lufta, fig. 3.1B. For å beskrive bevegelsen til ballen sett fra ditt ståsted er det nødvendig å innføre en kraft som drar ballen mot

høyre for at Newtons lover skal være oppfylt i det roterende karusellsystemet. Denne kraften kalles Coriolis-kraften.

Coriolis kraften kommer av at når du kaster ballen mot andre siden i karusellen vil avstanden fra ballen til karusellens rotasjonsakse først bli mindre, for deretter å øke igjen. Når massen til et roterende legeme beveger seg nærmere rotasjonsenteret vil banefarten øke, og derfor vil det se ut som om ballen beveger seg i en bue mot venstre. Når ballen har passert rotasjonsaksen begynner radiusen å øke og ballen vil sakke farten. Nå har vi krysset rotasjonsaksen, og banefarten til karusellen får motsatt retning. Derfor vil en sakking av farten på denne siden føre til at ballen fortsetter ut mot venstre relativt til karusellsystemet.

Matematisk kan akselrasjonen som følge av Coriolis-effekten uttrykt på vektor form skrives:

$$\underline{a}_c = -2\underline{\Omega} \times \underline{v} \quad (3.1)$$

der  $\underline{a}_c$  er akselasjonen som forårsakes av Coriolis-kraften,  $\underline{\Omega}$  er rotasjonshastigheten til det roterende systemet relativt til treghetsrommet og  $\underline{v}$  er farten til objektet relativt til det akselrerte systemet. Coriolis-kraften kan dermed uttrykkes som:

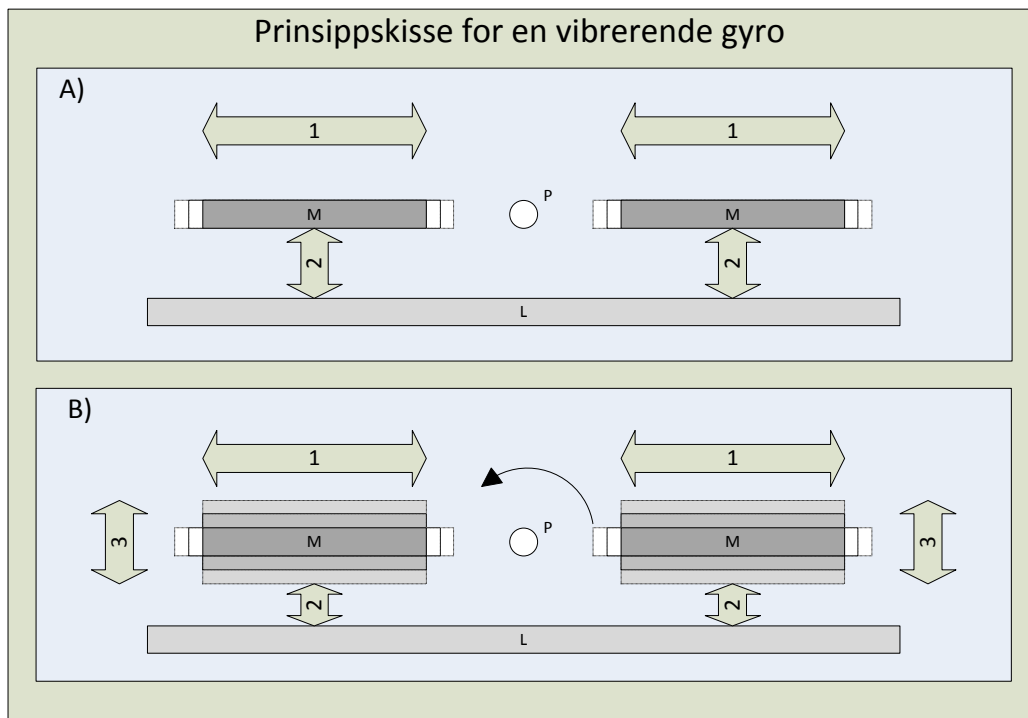
$$\underline{F}_c = -2 \cdot m \cdot \underline{\Omega} \times \underline{v} \quad (3.2)$$

Som vi ser er dette et vektorprodukt og retningen til kraften kan bestemmes ved hjelp av høyrehåndsregelen. Vi må imidlertid være påpasselige med å ta hensyn til minustegnet, som fører til en resulterende kraft i motsatt retning. Hvis vi for eksempel står på ekvator og kaster et legeme langs ekvator i retning øst, vil høyrehåndsregelen gi oss at Coriolis kraften virker normalt på jordoverflaten og være rettet motsatt vei av gravitasjonen.

I MEMS-gyroskop utnyttes denne kraften ved at en masse settes i svingninger i et plan. Massene er montert over en annen plate, og kapasitansen mellom denne platen og den vibrerende massen måles. Når systemet settes i rotasjon vil vibrasjonene føre til at det virker en Corioliskraft på massene vinkelrett på vibrasjonsplanet. Massene vil derfor begynne å vibrere også i denne retningen. Denne vibrasjonen fører til en endring i kapasitansen mellom massene og platen som en direkte funksjon av rotasjonshastigheten.

### 3.2.2 Måleprinsipp

Det finnes flere typer MEMS-gyroskop. Felles for alle er at de består av vibrerende masser og utnytter Coriolis-kraften for å beregne rotasjon. Det er hovedsaklig 3 metoder for å måle corioliskraften som benyttes. De tre metodene er:



Figur 3.2: Illustrasjon av gyroens virkemåte. Figur A) To masser settes i svingninger med  $180^\circ$  faseforskjell langs 1. Man måler kapasitansen mellom  $M$  og  $L$  (2). Når gyroen står i ro vil denne være konstant. Figur B) Når gyroen roteres om  $P$  vil Coriolis effekten skape vibrasjoner i  $M$  langs 3. Dette fører til at kapasitansen mellom  $M$  og  $L$  (2) vil følge en sinusfunksjon. Amplituden til funksjonen er direkte funksjon av vinkelfarten.

- Enkel oscillator
- Balansert oscillator
- Ring oscillator

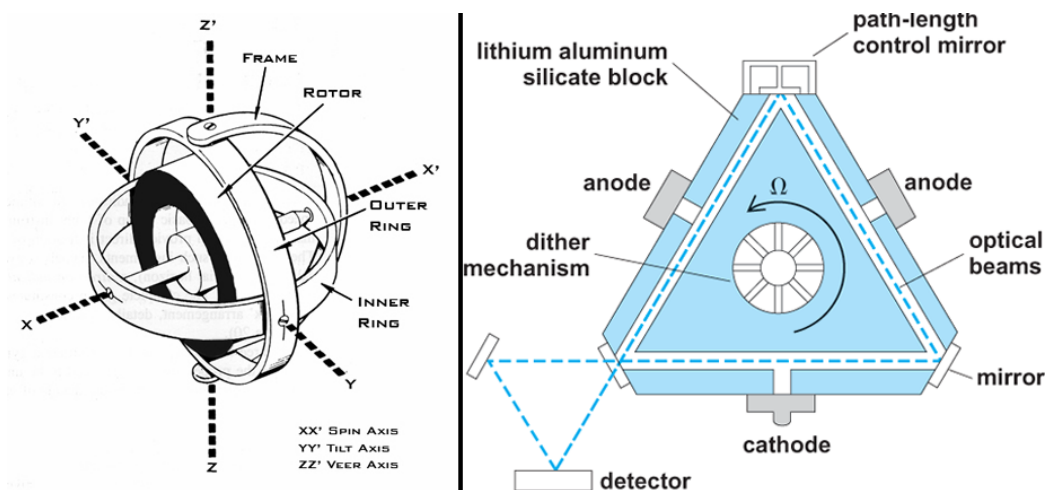
For en mer detaljert beskrivelse av disse tre målemetodene og forskjellen mellom dem henvises det til [16]. I denne oppgaven skal vi bruke en gyro basert på en balansert oscillator, og vil derfor kun gå inn på denne typen.

En balansert oscillator er en oscillator som består av to vibrerende masser som vist i fig. 3.2. Man måler så kapasitansen mellom de to massene og en fast plate. Massene settes i svingninger med forskjellig fase. Dette gjør at når den ene massen beveger seg i en retning, så vil den andre bevege seg i motsatt retning. Når systemet roteres vil derfor Corioliskraften på de to

massene hele tiden ha motsatt fortegn og kapasitansen som måles mellom massene og platen vil svinge i motfase. Eventuell støy fra vibrasjoner og lineær akselerasjon vil virke inn likt på begge massene og kan enkelt filtreres vekk ved å fjerne den delen av signalet som er likt i de to målingene. Dette gir et måleoppsett som er mye mer robust i forhold til støy enn feks. ett oppsett basert på en enkel oscillator. Signalet som fås fra disse oscillatorene forsterkes så opp og gir et målesignal som gjenspeiler rotasjons hastigheten til gyroen relativt til treghetsrommet.

### 3.3 MEMS kontra ordinære gyroskoper.

På markedet i dag finnes det flere andre metoder å utforme et gyroskoper på. Mange av disse gir en mye bedre nøyaktighet enn hva et MEMS gyroskoper vil klare å oppnå i overskuelig fremtid. Disse gyroene baserer seg blant annet på måling av tregheten i roterende masser, lasere og fiberoptikk. Et bilde av to forskjellige andre gyroer er vist i figur 3.3. Så hva er grunnen til at man ønsker å benytte MEMS gyroer til nordsøking i stedet for andre og mer nøyaktige gyroer? Et av de store problemene med gyroer basert på roterende



Figur 3.3: Venstre: Gyro basert på roterende masse. Hjølet i midten vil, når det roterer hurtig, holde seg stabilt i treghetsrommet når aksene rundt roteres. Vinkel kan da måles relativt til dette. Høyre: Gyro basert på laser. En laserstråle sendes i hver retning i trekanten. Når trekanten roteres i planet vil lengden laserlyset må gå bli lenger med rotasjonsretning enn mot.

masser er at de er veldig fintfølene for mekanisk slitasje siden der er mange roterende deler. Disse gyroene er også veldig dyre dersom stor nøyaktighet

kreves. En nordsøker som skal brukes til militære formål er avhengig av å være robust og i tillegg er det ønskelig at den skal være relativt billig å lage. Gyroer basert på roterende masser er derfor lite egnet til dette formålet.

Fiberoptiske gyroer brukes i dag til nordsøking. Disse er montert på en mekanisk dreieinnretning, og måler jordrotasjonskomponenten automatisk i flere retninger. Den beregner så nord med stor nøyaktighet basert på dette. Disse gyroene er også relativt store og den mekaniske dreieinnretningen gjør at også denne målemetoden har svakheter i forhold til mekanisk slitasje. Basert på disse utfordringene er det derfor av stor interesse å lage en nordsøker basert på MEMS og vippeprinsippet. En slik måleinnretning vil være veldig lett og liten, i tillegg til at måleprinsippet kun baserer seg på en enkel hengsel og derfor vil være lite sårbar for mekanisk slitasje. Antakelig vil et MEMS gyroskop også bli vesentlig billigere.

Nøyaktigheten er derimot veldig mye dårligere for MEMS gyroene enn den er for andre typer gyroer, og inntil nylig har det ikke vært aktuelt å bruke MEMS gyroer til nordsøking. Det har vært gjort forsøk på nordsøking med MEMS gyroer [9], men foreløpig har dette gitt en nøyaktighet og en måletid som ligger langt unna det som er ønskelig til militære formål.

Det er imidlertid en rivende utvikling på dette området. Denne oppgaven tar utgangspunkt i et helt nytt gyroskop fra den norske produsenten Sensoror, som i følge produsenten skal være kapabel til å brukes i nordsøkings enheter.

### 3.4 STIM210

STIM210 er siste tilskudd til gyro-serien fra det norske selskapet Sensoror. Sensoror har i en årrekke drevet forskning og utvikling av sensorer av forskjellig slag, men har de siste årene blitt ledene innefor utvikling av høytytende gyroskop basert på MEMS teknologi. Forløperen til STIM210, STIM202, var banebrytende når den kom og vant bla. prisen “European High Precision MEMS Gyroscopes New Product Innovation Award”[8]. STIM210 bygger på mye av det samme som STIM202, men den er kraftig forbedret i forhold til støy. Sensoren benytter måleprinsippet med en balansert oscilator som

	STIM202	STIM210
Angular Random Walk $[\circ/\sqrt{h}]$	0,2	0,15
Bias instability $[\circ/h]$	0,5	0,5

Tabell 3.1: Sammenligning av vesentlige data for de to modellene.



Figur 3.4: STIM210 er den nye høypresisjonsgyroen fra det norske selskapet Sensoror.[12]

er beskrevet tidligere i kapitlet. Den leveres i en robust boks på størrelse med en fyrstikkeske, og har et RS422 grensesnitt mot PCen. Sensoren har imidlertid ikke vært tilgjengelig for testing i denne oppgaven. Derfor er forgjengeren STIM202 brukt i forsøk for å støtte opp om simulatoren som er laget. De to sensorene har imidlertid tilsynelatende ganske like karakteristikk, og beregningene gjort i denne oppgaven er ventet og også holde for STIM210.





# Kapittel 4

## Modell av sensorer

For å lage mest mulig reelle målinger som kan brukes i simuleringen av kompasset må det lages en modell av sensorene, som kan gjenskape forstyrrelsene i målingene på en mest mulig reell måte. I dette kapitlet vil det bli utarbeidet en matematisk modell for gyroen og akselrometeret. Modellen for sensorene blir på formen:

$$\omega_m(t) = \omega_s(t) + v(t) + b \quad (4.1)$$

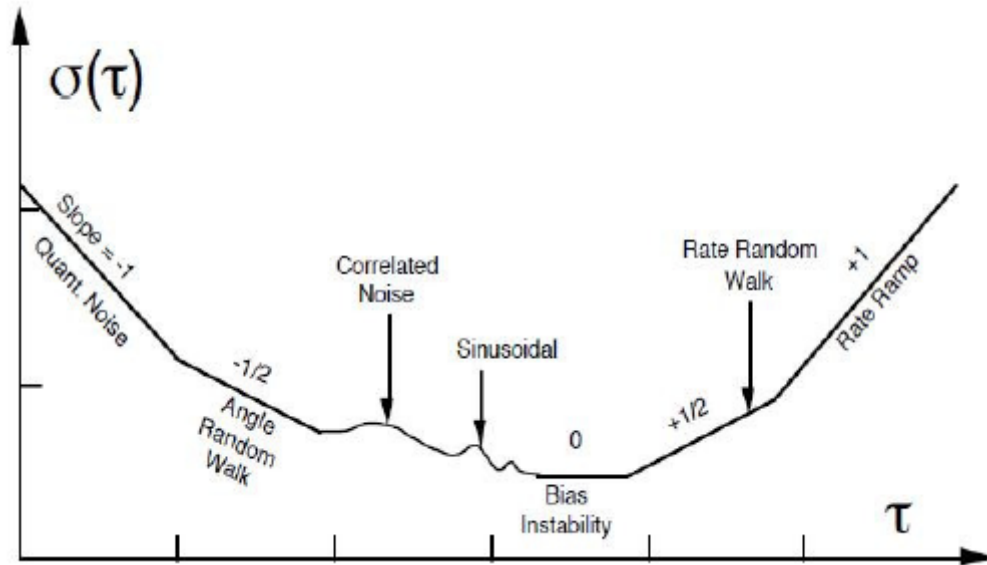
der  $\omega_s$  betegner den virkelige målingen,  $v$  betegner bidraget fra støykomponentene og  $b$  betegner en konstant bias som kan ligge på målingene. Vi skal på de neste sidene analysere hver av disse komponentene, og sette opp en modell basert på data fra gyroens datablad og forsøk gjort hos FFI. Til slutt vil vi se på hvordan forventningsvedien og variansen til gyrosignalet ser ut. Dette kan brukes for å finne hvilken måletid som er optimal for å undertrykke støyen best mulig. Først ser vi på modellen for støyen i gyroen,  $v$ .

### 4.1 Støykilder i Gyroskopet

I gyroer er det hovedsaklig 7 støykilder som er fremtredende[1]. Disse kan gjenkjennes på et Allanvarians-plot som rette linjer med forskjellig helning. De 7 forskjellige støykildene med helning i Allanvariansplottet er illustrert i figur 4.1.

#### 4.1.1 Støytyper i STIM gyroene

I tabell 4.1 har jeg nå vist en oversikt over de støytyper som er mest vanlige i MEMS treghetssensorer. For å få nøyaktig inntrykk av hvilke støytyper som er fremtredende i gyroer fra STIM serien har jeg gjort et forsøk med



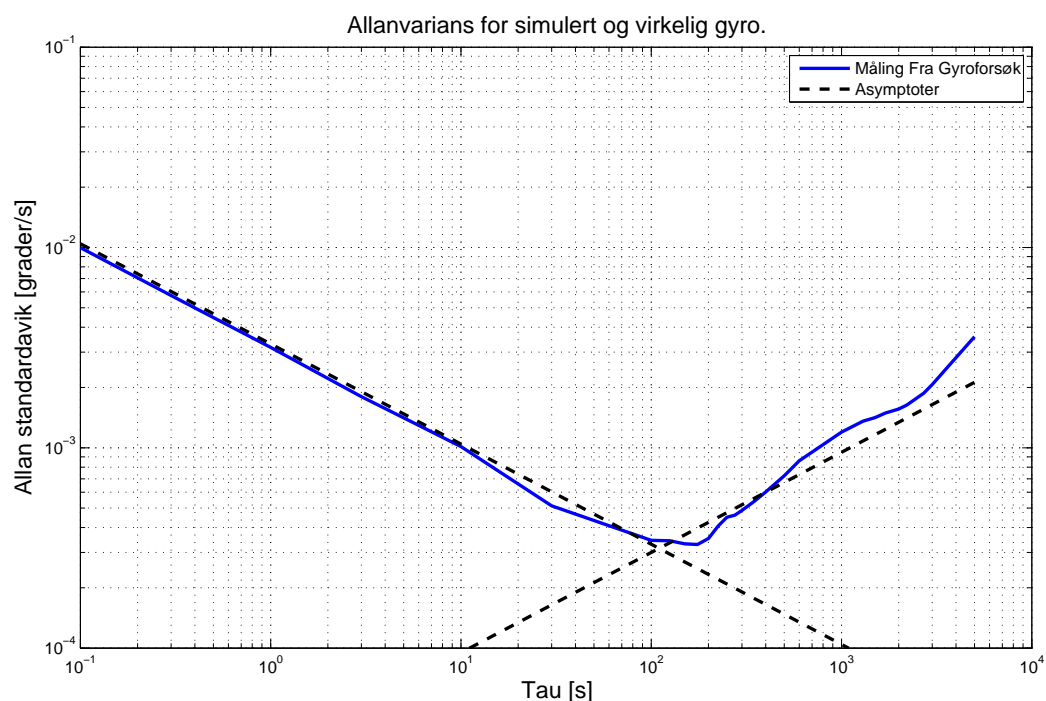
Figur 4.1: De 7 forskjellige støykildenes representasjon i et Allanvariansplot[1]

Støytype:	Stigningstall
Kvantiserings støy	-1
Angular Random Walk	$-\frac{1}{2}$
Korrelert Støy	-
Sinus støy	-
Bias Instability	0
Rate Random Walk	$+\frac{1}{2}$
Rate Ramp	+1

Tabell 4.1: Støytyper representert i gyroer.

STIM202 siden jeg ikke har hatt tilgang til STIM210. Disse bygger imidlertid på samme prinsipper og det er rimelig å anta at støytypene som oppstår er omtrent de samme. For å finne hvilke støytyper som er mest fremtredene, har jeg derfor montert opp STIM202 på et bord og tatt opp en lang måleserie. Gyroen har ligget helt i ro over en periode på 10000 sekunder eller ca 3 timer. Denne lengden på serien er helt i minste laget for å kunne karakterisere alle støykildene, men måleutstyret som var tilgjengelig hadde en begrensning på 10mill. sampler ( $10000s \cdot 1000Hz$ ), så dette var det beste som lot seg gjøre.

Etter måling ble Allanvariansen til måleserien beregnet i Matlab, og resultatet er vist i figur 4.1.1. Jeg har i figuren også lagt inn to asymptoter for ARW- og RRW-støy. Som vi ser følger helningen til måleserien disse kurvene



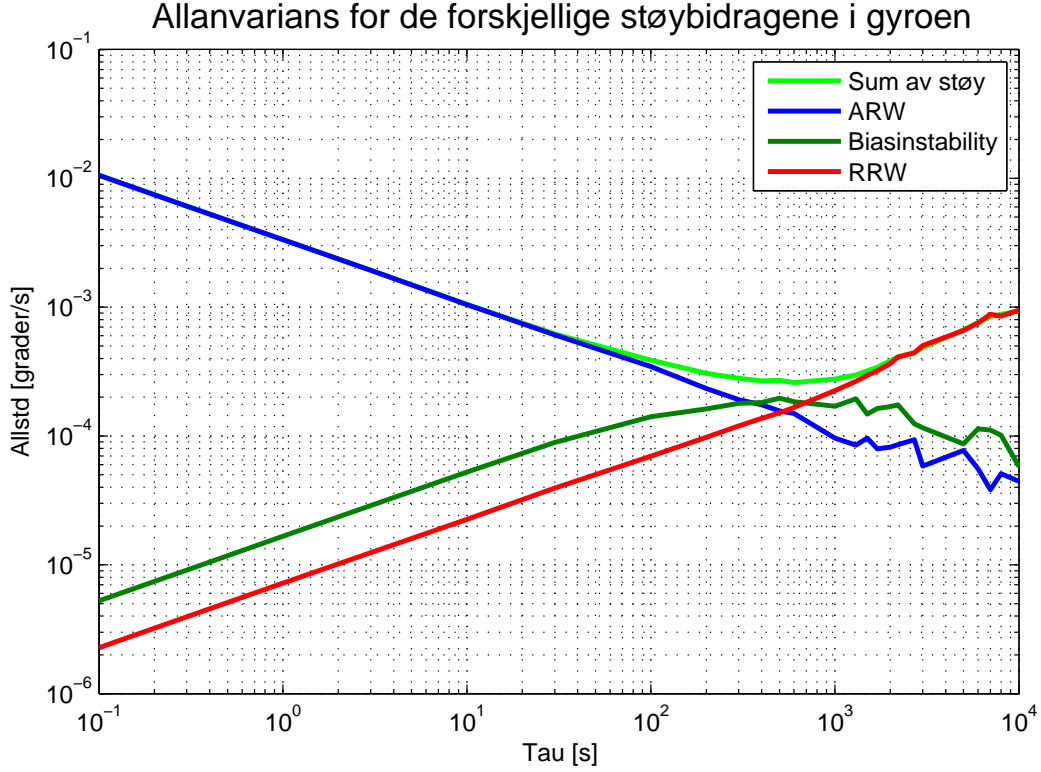
Figur 4.2: Allanvarians for Stim 202. Asymptoter for ARW og RRW er inn-tegnet.

ganske godt. Også i bunnpunktet er støyen godt beskrevet av disse to asymptotene. Det betyr at en modell av gyrosignalet basert på ARW og RRW vil være tilstrekkelig for å gjenskape signalene til vårt formål.

NB: Asymptotene i figur 4.1.1 er kun tegnet inn for å gi et visuelt inntrykk og kan ikke brukes til å estimere gyroparametre. For å estimere parameterne til gyroen referer jeg til [16]. Estimeringsresultatene i [16] fremstår imidlertid ikke helt troverdige fra mitt ståsted. Jeg har senere i oppgaven vist hvordan man kan bruke dataene oppgitt i databladet, ARW- og Biasinstabilitetskoeffisient, for å beskrive støykomponentene ARW og RRW.

### 4.1.2 Angular Random Walk - ARW

Angular Random Walk oppstår som følge av hvitstøy på hastighetsmålingen. Integrasjon av hastigheten fører til at feilen opptrer som en Random Walk feil på stillingen, derav navnet ARW. I Gyromodellen er vi interessert i hastigheten, derfor vil denne feilen opptre som en hvitstøy med varians  $q_\gamma$  direkte på målingen. For å finne denne variansen som funksjon av ARW konstanten i databladet, må vi se nærmere ARW prosessen. Denne er gitt med



Figur 4.3: Allanvarians-plot for de forskjellige støybidragene i Gyromodellen

differensiallikningen:

$$\gamma(t) = \int_0^T v_\gamma(t), dt \quad \text{der : } v_\gamma(t) \sim N\{0, q_\gamma \delta(t)\} \quad (4.2)$$

Siden gyromålingene gjøres diskrete med samplingstiden  $\delta t$ , vil vi kunne se på dette integralet som en sum av rektangler med bredde  $\delta t$  og høyde trukket fra fordelingen  $N\{0, q_\gamma\}$ . Dette gir oss at integralet fra 0 til T blir en sum av arealet av  $n$  rektangler der  $n = T/\delta t$  og kan skrives[4]:

$$\int_0^T v_\gamma(t), dt = \delta t \sum_{k=0}^n v_{\gamma k} \quad (4.3)$$

Fordelingen til hvitstøyen er tidsinvariant,  $E[v_{\gamma k}] = 0$  og  $Var[v_{\gamma k}] = Var[v_\gamma] = \sigma^2$ . Dette gir oss følgende forventning og varians for det integrerte signalet:

$$\begin{aligned} E \left[ \int_0^{\Delta t} v_\gamma(t), dt \right] &= \delta t \sum_{k=0}^n E[v_{\gamma k}] = 0 \\ Var \left[ \int_0^{\Delta t} v_\gamma(t), dt \right] &= \delta t^2 \sum_{k=0}^n Var[v_{\gamma k}] = \delta t^2 n \sigma_{v_\gamma}^2 \end{aligned} \quad (4.4)$$

Innsatt  $n\delta t = T$  gir det oss følgende uttrykk for variansen i det integrerte signalet etter tiden T:

$$\sigma(T)_\gamma^2 = \delta t \cdot T \cdot \sigma_{v_\gamma}^2 \quad (4.5)$$

I databladet for gyroer er Angular Random Walk konstanten (ARW) oppgitt med enheten  $\left[\frac{\circ}{\sqrt{hr}}\right]$ . Denne beskriver roten av variansen til det integrerte signalet, avlest etter 1 time. Verdien for ARW kan leses av i et Allanvariansplot ved  $\tau = 1h$ . Ved å snu på (4.5) og sette inn ARW konstanten for  $\sigma(1[h])_\gamma$  og T = 1 time, gir dette oss følgende sammenheng mellom  $\sigma_{v_\gamma}$ ,  $\delta t$  og ARW konstanten:

$$(\sigma_{v_\gamma} \left[\frac{\circ}{h}\right])^2 = (ARW \left[\frac{\circ}{\sqrt{hr}}\right])^2 \cdot \frac{1}{\delta t [hr]} \quad (4.6)$$

Under våre simuleringer vil vi påtrykke hvitstøyen diskret med samplingstiden  $Ts[s]$ . Hvitstøyen vi får på våre målinger er derfor karakterisert ved:

$$v_{\gamma k} \sim N\{0, q_{\gamma k} \delta_{k,s}\} \quad (4.7)$$

der:

$$q_{\gamma k} = (\sigma_\gamma \left[\frac{\circ}{s}\right])^2 = \left(\frac{ARW}{\sqrt{3600}} \cdot \frac{1}{\sqrt{Ts}}\right)^2 = \left(\frac{ARW}{\sqrt{3600}} \cdot \sqrt{Hz}\right)^2 \quad (4.8)$$

hvor  $Hz$  er gyroens samplingsfrekvens.

I et Allan-varians plot vil denne støyen opptre som en linje med stigning -0,5. Et eksempel på denne er vist som blå linje i figur 4.3.

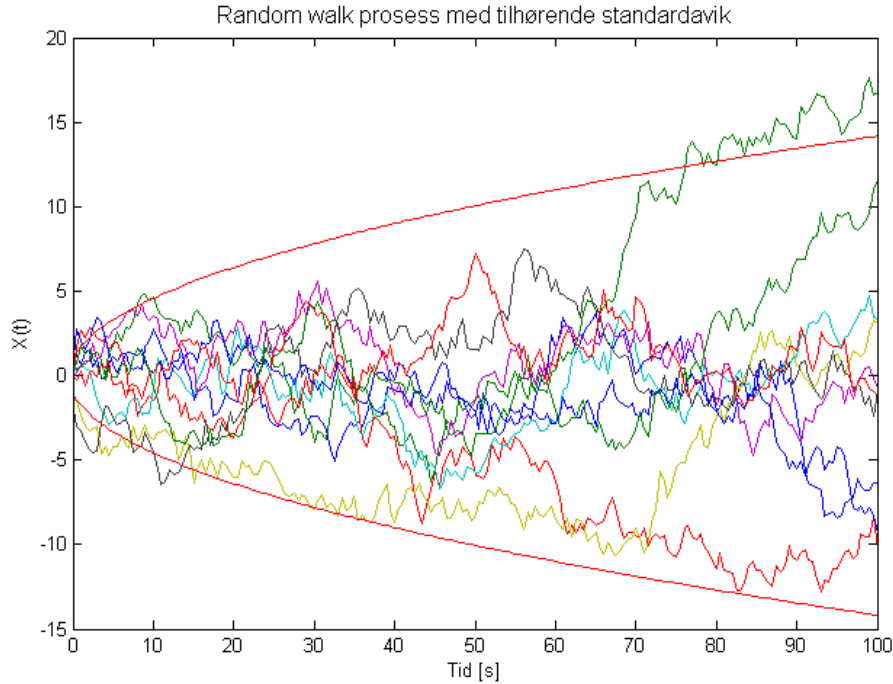
### 4.1.3 Rate Random Walk - RRW

Rate random walk er en feil av samme type som Angular Random Walk, men her snakker vi om en hvitstøy som gjør seg gjeldende på vinkelakselrasjonsnivå. Når vi måler hastigheten blir denne hvitstøyen integrert opp og fører til en random walk prosess som legges på hastighetsmålingene. Matematisk kan denne støyen ses på som prosesstøy. Utviklingen til denne støyprosessen på hastighetsnivå kan beskrives med differensiallikningen:

$$\dot{\beta} = v_\beta(t) \quad (4.9)$$

Der  $v_\beta(t)$  er hvitstøy med varians  $q_\beta$ . En slik prosess vil opptre på lave frekvenser, og gjøre at hastighetsmålingene vil drifte ut over tid. Et eksempel på dette er gitt i figur 4.4.

I et Allanvariansplot vil RRW fremstå langs en asymptote med stignings-tall  $+\frac{1}{2}$ . RRW beskrives i databladet med en parameter som har benevnning  $\circ/h/\sqrt{h}$ . Denne angir standardavviket til den integrerte prosessen avlest etter 1



Figur 4.4: Random walk prosess simulert 10 ganger med  $q_\beta=1$ . Utviklingen i standardaviket er også vist.

time. Basert på denne verdien kan man regne ut variansen  $q_\beta$ . Denne utregningen har mange likhetstrekk med utregningen av  $q_\gamma$ , men i dette tilfellet er prosessen kontinuerlig, og derfor må det gjøres litt annerledes. Variansen til  $\beta$  har følgende utvikling:

$$\dot{P}_\beta = q_\beta \quad (4.10)$$

Som gir:

$$P_\beta(T) = T \cdot q_\beta \quad (4.11)$$

Standardaviket er da gitt ved ligningen:

$$\sigma_\beta(T) = \sqrt{T} \cdot \sigma_{v_\beta} \quad (4.12)$$

RRW konstanten oppgitt i databladet er standard aviket til prosessen avlest etter 1 time. RRW konstanten i databladet kan derfor settes inn for  $\sigma_{v_\beta}$  og standard aviket for prosessen etter T timer vil derfor utvikle seg som:

$$\sigma_\beta(T)[^\circ/h] = \sqrt{T[h]} \cdot RRW[^\circ/h/\sqrt{h}] \quad (4.13)$$

Gjøres dette om til sekunder blir funksjonen:

$$\sigma_\beta(T[h])[^\circ/s] = \sqrt{T[h]} \cdot \frac{RRW[^\circ/h/\sqrt{h}][^\circ/s/\sqrt{h}]}{3600} \quad (4.14)$$

Vi ønsker også at T kan oppgis i sekunder, derfor blir uttrykket for  $\sigma_\beta(T[s])$ :

$$\begin{aligned}\sigma_\beta(T[s])[\text{°}/s] &= \frac{\sqrt{T[s]}}{\sqrt{3600}} \cdot \frac{RRW[\text{°}/h/\sqrt{h}]}{3600} [\text{°}/s/\sqrt{h}] \\ &= \sqrt{T[s]} \cdot \underbrace{\frac{RRW[\text{°}/h/\sqrt{h}]}{\sqrt{3600^3}}}_{\sqrt{q_\beta}} [\text{°}/s/\sqrt{s}]\end{aligned}\quad (4.15)$$

Dette gir da følgende uttrykk for  $q_\beta$  i den kontinuerlige prosessmodellen:

$$q_\beta = \frac{(RRW[\text{°}/h/\sqrt{h}])^2}{3600^3} [\text{°}^2/s^3] \quad (4.16)$$

Og den komplette beskrivelsen av Rate Random Walk feilen er gitt ved:

$$\dot{\beta} = v_\beta(t) \quad \text{der: } v_\beta \sim N\{0, q_\beta \delta(t-s)\} \quad (4.17)$$

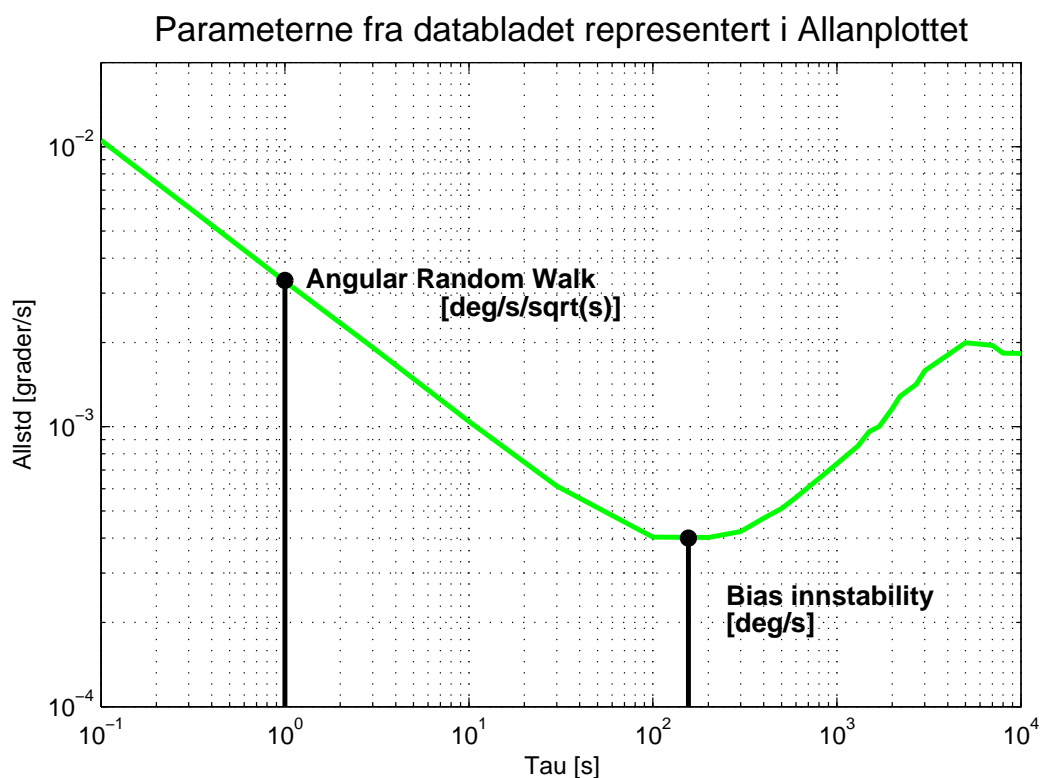
#### 4.1.4 Sammenheng mellom ARW, RRW og parameterne i databladet

Sampling frequency	2000	SPS
Mechanical shock	1500	g
In-run bias instability	0.5	°/h
Angular random walk	0.15	°/√h
Bias error over temperature gradients	±10	°/h rms
Bandwidth (-3dB)	262	Hz

Figur 4.5: Utdrag fra databladet til STIM 210.

Støyen i MEMS-gyroer karakteriseres hovedsaklig av to parametere i databladet. Disse er oppgitt som Angular Random Walk og In-run Bias instability. I figur 4.6 har jeg tegnet inn de 2 parameterne i et Allanvarians-plot.

**Angular Random Walk:** Parameteren forteller om hvor mye vinkelmålingen drifter i løpet av 1 time, som følge av målestøy på hastighetsmålingen. Denne parameteren kan leses av på Allanvariansplottet der  $\tau$  er 1 time. (Evt. 1 sekund dersom vi ønsker driften per sekund.) Verdien beskriver variansen til målestøyen og kan implementeres direkte i (4.8).



Figur 4.6: Sentrale parametre fra databladet representert i et Allanvarians plot.

**Bias Instability:** Parameteren forteller oss den absolutt beste nøyaktigheten man kan få på hastighetsmålingen dersom man kompenserer for feil og midler over flere målinger. Denne parameteren bestemmes som bunnpunktet i Allanvariansplottet.

Det er, som vi ser, ikke gitt noen eksplisitt verdi for Rate Random Walk i databladet. Man kan selvfølgelig, dersom man har gyroen tilgjengelig, foreta målinger og prøve å estimere denne slik som det er gjort i [16], men dersom dette ikke er en mulighet vil man kunne beregne en RRW konstant som gir riktig allanvarians basert på de to verdiene i databladet samt antagelsen vi har gjort om at støyen i hovedsak består av de to støykildene Angular Random Walk og Rate Random Walk.

Allanstandardavviket til støykildene Angular Random Walk og Rate Random Walk som funksjon av  $\tau$  er gitt som rette linjer i et loglog plot. Linjene



kan beskrives ved ligningene:

$$\text{ARW: } \sigma_{ARW}(\tau) = N \cdot \tau^{-\frac{1}{2}} \quad (4.18)$$

$$\text{RRW: } \sigma_{RRW}(\tau) = \frac{K}{3} \cdot \tau^{\frac{1}{2}} \quad (4.19)$$

Der N benevner ARW konstanten [ $^{\circ}/\sqrt{hr}$ ] og er gitt i databladet, K benevner RRW konstanten og består av [ $^{\circ}/h/\sqrt{h}$ ], og er ukjent. Det totale allanstandardaviket er gitt som en RMS sum av disse to bidragene.

$$\sigma_{tot}(\tau)^2 = \sigma_{ARW}(\tau)^2 + \sigma_{RRW}(\tau)^2 \quad (4.20)$$

$$= N^2 \cdot \tau^{-1} + \left(\frac{K}{3}\right)^2 \cdot \tau \quad (4.21)$$

Bunnpunktet faller der linjene krysser hverandre i  $\tau_1$ , her er  $\sigma_{tot}(\tau_1)$  lik Bias instability fra databladet. Dette gir:

$$\sigma_{tot}(\tau_1)^2 = 2 \cdot \left(N \cdot \tau_1^{-\frac{1}{2}}\right)^2 \quad (4.22)$$

$$\Downarrow \quad (4.23)$$

$$\tau_1 = \frac{2 \cdot N^2}{\text{Biasinst.}^2} \quad (4.24)$$

Settes dette inn i (4.21) sammen med Bias instability og N kan K løses ut og vi får:

$$K = \sqrt{3} \cdot \frac{\text{Biasinstability}^2}{2 \cdot N} \quad (4.25)$$

En rask dimensjonsanalyse av dette svaret viser at det stemmer overens med dimensjonen til  $q_{\beta}$  i likning (4.16). Når ikke annen informasjon om sensoren finnes enn dataene som er oppgitt i databladet, er dette en grei måte å finne verdien for Rate Random Walk som gir oss bunnpunkt på riktig plass. Dette forutsetter som nevnt at gyroen ikke har andre dominerende støykilder enn de to overnevnte.

## 4.2 Støytyper i Akselerometeret

Det er ikke gitt noen preferanser i oppgaveteksten når det gjelder valg av akselerometer. Jeg har derfor heller ikke hatt muligheten til å gjøre målinger på akselerometeret. Støymodellen for dette baserer seg derfor kun på data fra akselerometerets datablad, samt hvordan akselerometeret er modellert i andre lignende oppgaver[15].

Støyen i akselerometeret lar jeg derfor bestå av målestøy, samt en nullpunktsfeil.

**Målestøy** Siden akselerometermålingene måler akselerasjon vil denne målestøyen tilsvare hvitstøyen som ligger til grunn for Rate Random Walk støyen i gyroen. Dette er en støy som vil føre til en Random walk prosess på et eventuelt fartsestimat basert på akselerometer målingene. I denne oppgaven er jeg imidlertid interessert i å bruke akselerometeret til å måle den statiske akselerasjonen forårsaket av gravitasjonskraften,  $g$ . Støyen modelleres derfor som en hvitstøy på hver akse. Denne beskrives ved:

$$v_{ak}^i \sim N\{0, q_{ak}^i \delta_{k,s}\} \quad (4.26)$$

$q_{\gamma k}^i$  betegner her variansen til målestøyen på akse  $i$ . Denne er oppgitt i datablader med enheten  $[mg/\sqrt{Hz}]$  eller  $[g/\sqrt{h}]$ , og må evt. skaleres til  $[mg/\sqrt{Hz}]$  før den settes inn for  $q_{\gamma k}^i$ , tilsvarende det jeg har gjort for ARW verdien for gyroen.

**Nullpunktsfeil** Nullpunktsfeil er en feil som kommer av at akselerometeret ikke alltid går tilbake til nullpunktet etter å ha vært utsatt for en akselerasjon. Siden målinger av stillingen foregår under statiske forhold er det rimelig å anta at denne kan modelleres som en stokastisk konstant som trekkes fra en normalfordeling ved starten av hver måleserie. Feilen kan beskrives:

$$b^i \sim N\{0, q_b^i\} \quad (4.27)$$

Der  $b^i$  er nullpunktsfeilen for akse  $i$ .  $q_b^i$  beskriver spredningen til nullpunktsfeilen. Etter å ha undersøkt en del forskjellige datablad fra forskjellige produsenter har jeg funnet at standardavviket til denne ligger i størrelsesorden 30 - 150  $mg$ .

## 4.3 Matematisk modell av sensorene

Jeg har nå identifisert prosessene som er grunnlaget for den mest vesentlige delen av støyen. Derfor er jeg nå istand til å sette opp en modell av sensorene.

### 4.3.1 Modell av Gyroskopet

Modellen av gyroen kan nå settes opp på formen vist først i kapittelet:

$$\omega_{mk} = \omega_{sk} + v_k + b \quad (4.28)$$

Hvor  $v_k$  er en samling av støykomponentene. For gyroen blir dette ARW- og RRW-støybidragene som vi har beskrevet i (4.7) og (4.17). Modellen for

støyen  $v_k$  blir derfor:

$$\begin{aligned} \dot{\beta}(t) &= v_\beta(t) & \text{der : } v_\beta(t) &\sim \mathcal{N}(0, q_\beta \delta(t - \tau)) \\ v_k &= \beta_k + v_{\gamma k} & \text{der : } v_{\gamma k} &\sim \mathcal{N}(0, q_\gamma \delta_{kl}) \end{aligned} \quad (4.29)$$

Settes dette inn i måleligningen (4.28), fås en modell for gyro målingene som funksjon av virkelig rotasjonsfart:

$$\begin{aligned} \dot{\beta}(t) &= v_\beta(t) \\ \omega_{mk} &= \omega_{sk} + \beta_k + v_{\gamma k} + b \end{aligned} \quad (4.30)$$

$$v_\beta(t) \sim \mathcal{N}(0, q_\beta \delta(t - \tau)) \quad v_{\gamma k} \sim \mathcal{N}(0, q_\gamma \delta_{kl})$$

Verdiene for  $q_\beta$  og  $q_\gamma$  er funnet i (4.16) og (4.8):

$$q_\beta = \frac{(RRW[^\circ/h/\sqrt{h}])^2}{3600^3} [\text{o}^2/\text{s}^3] \quad q_\gamma = \left( \frac{ARW[^\circ/\sqrt{h}]}{\sqrt{3600}} \cdot \sqrt{Hz} \right)^2 \quad (4.31)$$

### Diskretisering av støymodellen:

Modellen for RRW støyen er kontinuerlig, mens resten av målemodellen er diskret. Under implementering i Matlab må alt være på diskret form. Jeg må derfor diskretisere den kontinuerlige delen av støymodellen og få den over på den diskrete formen:

$$\begin{aligned} \beta_{k+1} &= \phi \beta_k + v_{\beta k} \\ \omega_{mk} &= \omega_{sk} + \beta_k + v_{\gamma k} + b \end{aligned} \quad (4.32)$$

$$v_{\beta k} \sim \mathcal{N}(0, q_{\beta d} \delta_{kl}) \quad v_{\gamma k} \sim \mathcal{N}(0, q_\gamma \delta_{kl})$$

For å finne det diskrete systemet må vi beregne verdiene  $\phi$  og  $q_{\beta d}$ .

Diskretiseringen utføres slik at  $\beta_k = \beta(t_k)$  og  $P_\beta(t_k) = P_{\beta k}$ . For å oppfylle dette brukes diskretiseringen gitt i [19], her for det skalare tilfellet:

$$\phi = \phi(t_{k+1}, t_k) = e^{F \cdot (t_{k+1} - t_k)} = e^{F \cdot \Delta t} \approx \underbrace{I + F \cdot \Delta t}_{\text{1. ordens tilnærming}} \quad (4.33)$$

$$q_{\beta d} = \int_{t_k}^{t_{k+1}} \phi(t_{k+1}, \tau) G q_\beta G^T \phi(t_{k+1}, \tau)^T d\tau \quad (4.34)$$

F blir her 0 siden det ikke er noen tilbakekobling på tilstanden  $\beta$  og  $G = 1$ . Innsatt dette gir følgende verdier i den diskrete modellen:

$$\Phi = \Phi(t_{k+1}, t_k) = e^{0 \cdot (t_{k+1} - t_k)} = 1 \quad (4.35)$$

$$\begin{aligned}
q_{\beta_d} &= \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau) G Q_x G^T \Phi(t_{k+1}, \tau)^T d\tau \\
&= \int_{t_k}^{t_{k+1}} Q_x d\tau = q_{\beta} \cdot \Delta t
\end{aligned} \tag{4.36}$$

hvor  $\Delta t$  er samplingstiden.

Vi har dermed fått den skalare gyromodellen:

$$\begin{aligned}
\beta_{k+1} &= \beta_k + v_{\beta k} \\
\omega_{mk} &= \omega_{sk} + \beta_k + v_{\gamma k} + b
\end{aligned} \tag{4.37}$$

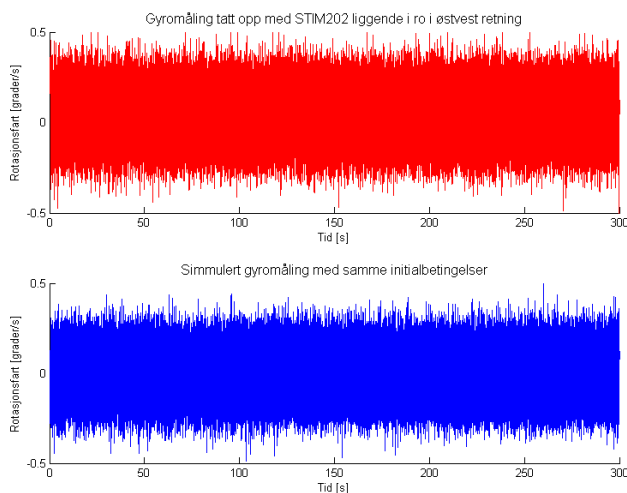
$$v_{\beta} \sim \mathcal{N}(0, q_{\beta d} \delta_{kl}) \quad v_{\gamma k} \sim \mathcal{N}(0, q_{\gamma} \delta_{kl})$$

Med verdiene:

$$q_{\beta d} = \frac{(RRW[\frac{rad}{h}/\sqrt{h}])^2}{3600^3} \cdot \Delta t[s] \quad q_{\gamma k} = \left( \frac{ARW[\frac{rad}{\sqrt{h}}]}{\sqrt{3600 \cdot \Delta t[s]}} \right)^2 \tag{4.38}$$

Modellen gir da ut en målt vinkelfart med benevnning [rad/s]. Dette forutsetter at også  $\omega_{sk}$  er gitt i [rad/s].

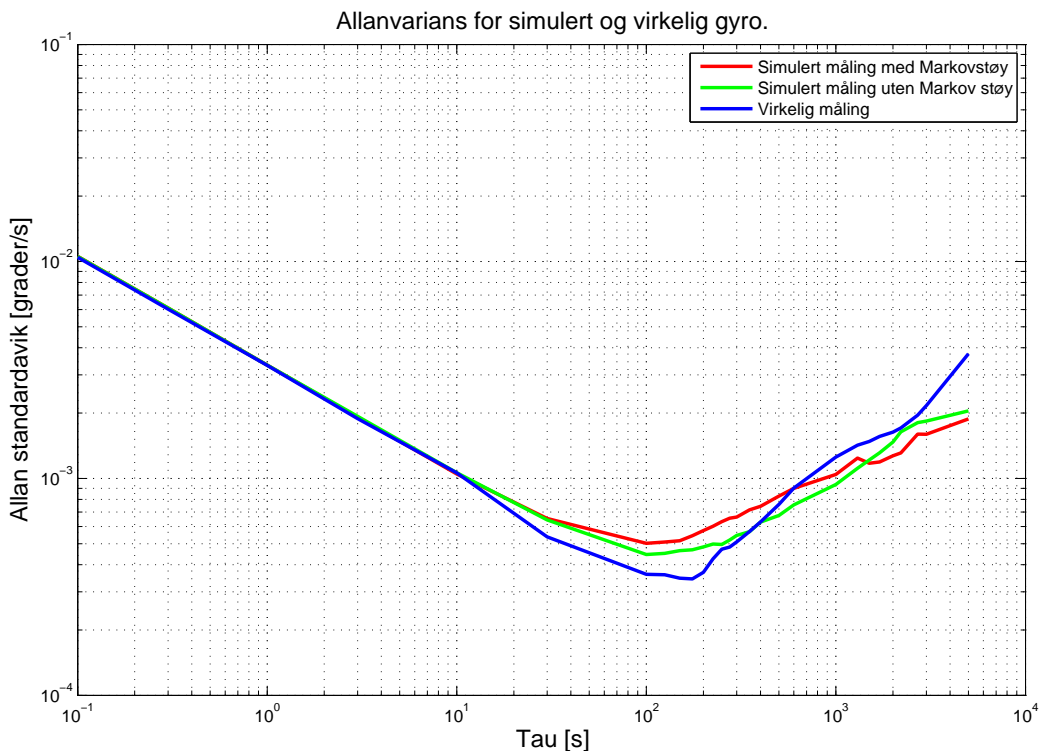
### Sammenligning av modell og virkelige målinger



Figur 4.7: Målt og simulert signal for STIM202, simulert i 5 min.

For å verifisere at modellen er riktig, og gir oss målinger med tilfredstillende likhet i forhold til virkelige gyromåling, har jeg her sammenlignet simulerte

data fra modellen med målinger gjort på gyroskopet STIM 202. Målingene som det sammenlignes med er gjort på et ratebord i horisontalplanet med gyroens sensitivitetsakse pekende i øst- vest retning. Denne oppstillingen gjør at målingene ikke blir påvirket av jordrotasjonen, og vi får en måling som kun inneholder støyen. I figur 4.7 ser vi øverst et simulert gyrosignal, og nederst ser vi det målte gyrosignalet. Som vi ser er standardavviket til de to signalene tilsynelatende like.



Figur 4.8: Allanvarians plot for simulert og målt gyrosignal basert på en måleserie over 10000s, samlet ved 1000 Hz

For en mer nøyaktig sammenligning av signalene, sammenligner vi Allanvariansen for det målte og det simulerte signalet. Disse er vist i figur 4.8, og er beregnet basert på måleserier av 10000 sekunders lengde, med en samplingsfrekvens på 1000Hz. Den grønne linjen angir simulerte målinger basert på modellen utledet over. Vi ser at denne vil følge signalet ganske bra. Den røde linjen beskriver en modell der det også er tatt med korrellert målestøy modellert som en 1. ordens markov. Dette har vært brukt i tidligere modeller av de samme gyroene, men som vist her gir dette en dårligere tilpassning til det målte signalet enn en prosess utelukkende basert på ARW og RRW støy.

En viktig ting å bemerke her er at RRW konstanten brukt under simuleringene her ikke er beregnet ut fra formlene gitt i kap 4.1.4, men funnet ved kurvetilpasning av de målte dataene. Rate Random Walk koefisienten er her  $8 [^\circ/h/\sqrt{h}]$ . Dette er ca. 8 ganger høyere enn hva som er oppgitt fra produsenten.

**Avik fra databladet:** Målingene jeg har gjort av Allan standardaviket viser en støykarakterestikk som er litt annerledes enn den som er oppgitt i databladet[10]. Det kan se ut til at labforsøket over gir målinger som er beheftet med mye mer Rate Random Walk støy enn det som er oppgitt i databladet. Dette resulterer i at et nordsøk basert på målingene blir mye dårligere enn det teoretisk burde vært. Simuleringene av STIM210 er imidlertid gjort basert på databladet og det er derfor muligheter for at disse simuleringsresultatene er en del bedre enn det som er mulig å fremkalle i praksis. Lab resultatene mine er imidlertid ganske like de som er gjort i [16].

### 4.3.2 Modell av Akselerometeret

Jeg skal bruke et akselerometer som måler gravitasjonskomponenten i 3 retninger. Disse 3 retningene står vinkelrett på hverandre og sammenstilles med kordinataksene til akselerometer-referanse-rammen omtalt senere. Jeg skal derfor modellere et akselerometer bestående av 3 måleakser. Modellen for akselerasjonsmålingen langs akse  $i$  består av den virkelige akselerasjonen langs denne, akselen pluss de to støykomponentene beskrevet i (4.27) og (4.26).

$$\begin{aligned} a_{mk}^i &= a_s^i + v_k^i + b^i \\ b^i &\sim N\{0, q_b^i\} \quad v_{ak}^i \sim N\{0, q_{ak}^i \delta_{k,s}\} \end{aligned} \quad (4.39)$$

For et 3-akset akselerometer kan dette settes opp på vektor form:

$$\begin{bmatrix} a_{mk}^x \\ a_{mk}^y \\ a_{mk}^z \end{bmatrix} = \begin{bmatrix} a_s^x \\ a_s^y \\ a_s^z \end{bmatrix} + \begin{bmatrix} v_k^x \\ v_k^y \\ v_k^z \end{bmatrix} + \begin{bmatrix} b^x \\ b^y \\ b^z \end{bmatrix} \quad (4.40)$$

↓

$$\underline{a}_{mk} = \underline{a}_s + \underline{v}_k + \underline{b} \quad (4.41)$$

Støyvektorene vil da være beskrevet ved:

$$\underline{b} \sim N\{0, Q_b\}, \quad \underline{v}_{ak}^i \sim N\{0, Q_{ak} \delta_{k,s}\} \quad (4.42)$$

$$Q_b = \begin{bmatrix} q_b^x & 0 & 0 \\ 0 & q_b^y & 0 \\ 0 & 0 & q_b^z \end{bmatrix} \quad Q_{ak} = \begin{bmatrix} q_{ak}^x & 0 & 0 \\ 0 & q_{ak}^y & 0 \\ 0 & 0 & q_{ak}^z \end{bmatrix} \quad (4.43)$$

Jeg har her satt alle korrelasjonsleddene lik 0 ettersom det ikke foreligger noe informasjon om sammenheng mellom støyen langs de forskjellige aksene.

Denne modellen beregner den målte akselerasjonen i de 3 retningene x,y og z som funksjon av den virkelige akselerasjonen i denne retningen. I denne oppgaven vil akselerometeret være i ro og akselerasjonen som måles vil være gravitasjonskraften målt langs de forskjellige aksene. Stillingen til målesystemet i forhold til horisontalplanet kan da beregnes.

## 4.4 Forventning og varians for gyroen

Det kan nå beregnes forventning og varians for målingene som er gitt av gyroen. Disse uttrykkene skal senere brukes til å beregne minste initielle øst-vest avik som er gunstig for nordsøking. Se kap. 6.2.1.

**Forventningsverdi:** Forventningsverdien til sample k i målingen  $\omega_{mk}$  er gitt ved:

$$E\{\omega_{mk}\} = E\{\omega_{sk}\} + E\{\beta_k\} + E\{v_{\gamma k}\} + E\{b\} \quad (4.44)$$

der:

$$\begin{aligned} E\{\omega_{sk}\} &= E\{\omega_s\} && \text{Forventning til sann vinkelfart} \\ E\{v_{\gamma k}\} &= 0 && \text{På grunn av egenskapene til hvitstøy} \\ E\{b\} &= b && \text{Siden biasen er en deterministisk størrelse} \\ E\{\beta_k\} &= 0 && \text{Siden dette er en Random Walk prosess.} \end{aligned}$$

Innsatt dette blir  $E\{\omega_{mk}\}$ :

$$E\{\omega_{mk}\} = E\{\omega_s\} + b \quad (4.45)$$

Som vist her er  $E\{\omega_{mk}\}$  uavhengig av k. En midling av k samples vil derfor ikke endre forventningsverdien og forventningen for middelet blir:

$$E\{\bar{\omega}_{mk}\} = E\{\omega_s\} + b \quad (4.46)$$

Forventningsverdien er derimot avhengig av biasen, denne kanselerer imidlertid i denne oppgaven ved å bruke vippe prinsippet, og den totale målingen blir derfor forventningsrett.

**Varians:** Variansen i simulatoren beregnes på bakgrunn av formelene og regnereglene for varians gitt i kap. 2.1. Variansen til målingen  $\omega_{mk}$  er da gitt ved:

$$Var\{\omega_{mk}\} = Var\{\omega_{sk}\} + Var\{\beta_k\} + Var\{v_{\gamma k}\} + Var\{b\} \quad (4.47)$$

der:

$$\begin{aligned}
 Var\{\omega_{mk}\} &= P_{\omega_{mk}} && \text{Variansen til målingen} \\
 Var\{\omega_{sk}\} &= 0 && \text{Variansen til sann vinkelfart} \\
 Var\{v_{\gamma k}\} &= q_{\gamma} && \text{Variansen til ARW} \\
 Var\{b\} &= 0 && \text{Siden dette i vårt tilfelle regnes som en deterministisk størrelse} \\
 Var\{\beta_k\} &= P_k && \text{Variansen til Random Walk prosessen}
 \end{aligned}$$

$P_k$  er gitt ved variansen til støymodellen:

$$P_k = P_{k-1} + Var\{v_{k-1}\} = P_{\beta(0)} + k \cdot q_{\beta d} \quad (4.48)$$

$P_{\beta(0)} = 0$  siden dette er en random walkprosess med kjent startverdi. Settes dette inn i (4.47) vil variansen for sample  $k$  være gitt ved:

$$P_{\omega_{mk}} = P_{\omega_s} + k \cdot q_{\beta d} + q_{\gamma} \quad (4.49)$$

For å undertrykke støyen midler vi alle målingene fra  $\omega_{m1}$  til  $\omega_{mk}$ . Variansen til den midlede målingen er da gitt ved:

$$\begin{aligned}
 Var\{\bar{\omega}_{mk}\} &= \frac{1}{k^2} \cdot \sum_{i=1}^k Var\{\omega_{mi}\} \\
 &= P_{\omega_s} + \frac{1}{k^2} \left( q_{\gamma} \cdot k + q_{\beta d} \cdot \sum_{i=1}^k k^2 \right)
 \end{aligned} \quad (4.50)$$



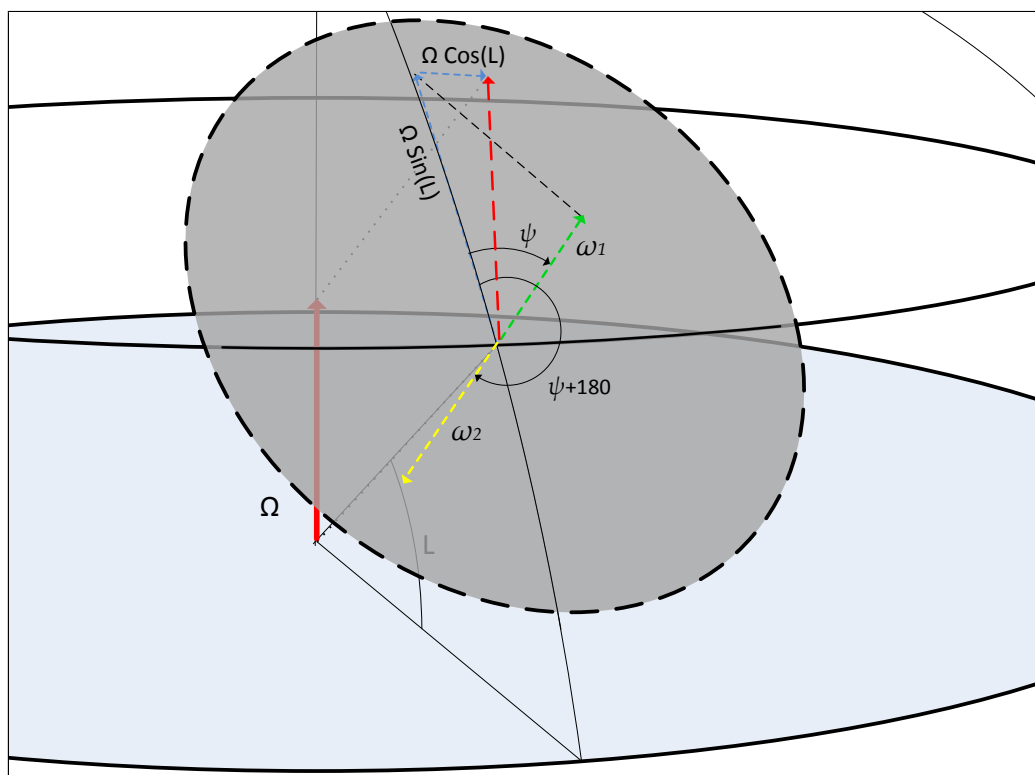
# Kapittel 5

## Simuleringsmodell av gyrokompass.

Sensormodellene jeg har laget i kapittelet foran, skal nå brukes til å lage en simuleringsmodell av et gyrokompass. Kompasset skal basere seg på et såkalt vippeprinsipp, dvs. at man først legger gyroen i en posisjon og foretar en måling av jordrotasjonskomponenten,  $\omega_1$ . Deretter vippes kompasset  $180^\circ$  vha. en hengsel-anordning for så å ta opp en tilsvarende måling,  $\omega_2$ , i denne posisjonen. Jordrotasjonskomponenten som tas opp i de to målingene vil da ha forskjellig fortegn mens eventuelle biaser i gyroen vil ha likt fortegn. Trekk disse målingene fra hverandre vil biasen kanselleres mens jordrotasjonskomponenten vil dobles. Målingen vil da være uavhengig av biasen og kan brukes til å estimere vinkelen mellom gyroens måleakse og sann nord. Måleprinsippet er forsøkt vist i figur 5.1 for en nordsøking der gyroen er oppstilt i horisontalplanet.

For å kunne bestemme nordvinkelen ved kun 2 posisjoneringer må breddegraden være kjent. Denne kan for eksempel bestemmes med en GPS eller manuelt ved avlesning på kart. Det er ikke ventet at nøyaktigheten i breddegrad vil være mer kritisk enn at en GPS, med nøyaktighet på ca. 7 meter, eller en manuell avlesning på kart vil gi mer enn god nok nøyaktighet.

Målingene av  $\omega_1$  og  $\omega_2$  som er vist i figur 5.1 forutsetter også at gyroens måleakse er stilt inn slik at den ligger i horisontalplanet i forhold til jorda. Dette kan være tidkrevende å justere inn, og jeg utstyrer derfor kompasset med et 3 akset akselerometer for å kunne bestemme stillingen til gyroen relativt til jordens horisontalplan. Når denne stillingen er bestemt kan jordrotasjonskomponenten i horisontalplanet bestemmes som funksjon av skjevstillingen. Jeg vil i kapitlet 6 se på hvor mye denne skjevstillingen har å bety for nøyaktigheten til nordvinkel-estimatet. Først vil jeg innføre noen forskjellige rammer/koordinatsystemer for å bedre kunne beskrive sammenhengen



Figur 5.1: Illustrasjon av måleprinsipp for gyrokompass med vippe plassert i horisontal stilling. Jordrotasjonskomponentene  $\omega_1$  og  $\omega_2$  måles og  $\psi$  estimeres basert på disse.

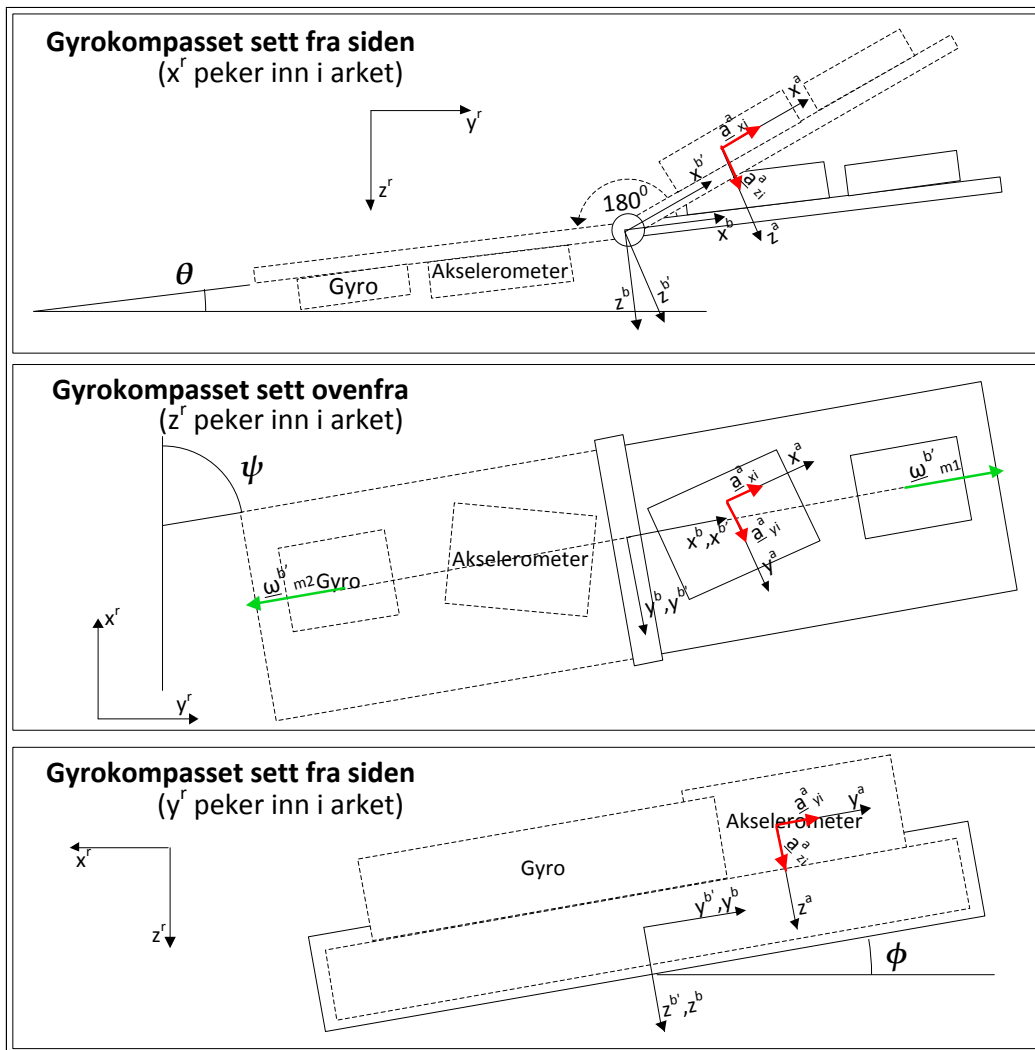
mellom de forskjellige vinkler og målinger.

## 5.1 Innføring av koordinatsystemer

For å systematisere målinger, vinkler og forholdet mellom disse, innfører jeg flere forskjellige rammer. Disse blir referanser for de forskjellige målingene. Koordinatsystemene jeg bruker i denne oppgaven og sammenhengen mellom dem blir presentert under.

### Jordfast-koordinatsystem, $R$

Jeg definerer et koordinatsystem,  $R$ , som referansesystem. Dette koordinatsystemet ligger fast i forhold til jordoverflaten med origo i gyroens sentrum. Aksene  $x$ ,  $y$  og  $z$  velges slik at  $x$  peker mot sann nord og  $y$  peker mot øst.



Figur 5.2: Skisse av et tenkt oppsett av kompasset sett fra forskjellige sider. Vinklene  $\theta$ ,  $\phi$  og  $\psi$ , samt de forskjellige koordinatsystemene er tegnet inn. Merk: Koordinatsystemene definerer egentlig vektorrom og vil derfor ikke ha forskjellig origo, de er imidlertid tegnet med forskjellig plassering for å illustrere bedre.

Høyrehåndsregelen gir da at  $z$  blir pekende mot jordens sentrum. Denne orienteringen kalles ofte NED (North, East, Down).

### Kompass-koordinatsystemet B

Jeg definerer et koordinatsystem som ligger fast i kompasssystemet. Systemet påvirkes ikke av vippingen av kompasset men ligger fast i kompassets posisjon

1. Dette systemet har også origo i gyroens sentrum. Aksene  $x$ ,  $y$  og  $z$  velges slik at  $x$  akse står vinkelrett på hengselens rotasjonsakse, pekende mot øst når  $y$  peker mot sør.  $Y$ -aksen liggende parallellt med hengselens rotasjonsakse. Høyrehåndsregelen gir da at  $Z$  akse står ut fra planet pekende ned når  $y$  peker sør og  $x$  peker øst.

### Vippe-koordinatsystemet $B'$

Jeg definerer et koordinatsystem til som ligger fast i kompassystemet. Dette systemet sammenfaller med  $\mathcal{F}^b$  når målingene er i posisjon 1. Systemet er festet til hengselen og følger med rundt når vi vipper gyroen. Dette systemet har også origo i gyroens sentrum. Aksene  $x, y$  og  $z$  legges slik at de sammenfaller med aksene for  $B$  systemet i posisjon 1.

### Akselerometer-koordinatsystemet $A$

Jeg definerer et koordinatsystem for akselerometeret som ligger fast til akselerometeret. Dette systemet legges med origo i gyroens sentrum. Aksene legges slik at de sammenfaller med akselerometerets sensitivitetsakser med  $z$  pekende ned når  $x$  peker mot nord og  $y$  peker mot øst.

For å forenkle regningen velger jeg å ikke innføre et eget koordinatsystem for gyroen. Denne består av en 1-akset gyro og jeg forutsetter derfor at denne aksene kan monteres parallellt med  $\mathcal{F}^{b'}$  sin  $x$ -akse. Gyromålingen kan derfor presenteres langs  $x$ -aksen i  $B'$  systemet.

**Merknad til valg av origo:** Under regningen velger jeg å legge origo til alle koordinatsystemene i sentrum av gyroen. Denne forenklingen gjør jeg for å forenkle transisjonsmatrisene ved å velge felles origo. Dette kan jeg gjøre siden det i kompasset ikke er noen translatoriske bevegelser. (Unntatt forflytningen når gyroen vipper, men den er så liten i forhold til jordradien at vi kan se bort fra den). Dersom systemet hadde inneholdt translatoriske bevegelser ville jeg lagt koordinatsystemene omtrent som på figur 5.3. Da måtte vi også målt avstanden mellom origoene og utvidet transisjonsmatrisen til en  $4 \times 4$  matrise.

#### 5.1.1 Rotasjonsmatriser

Jeg vil her vise de forskjellige retningskosinmatrisene som inngår i systemet, beskrive hvilke vinkler som opptrer og hvilke forutsetninger som er tatt for

at matrisene blir som de blir. Alle rotasjonsmatrisene beskriver rotasjoner i Eulervinkler, der rotasjoner skjer rundt nye akser mot klokken.

**Matrisen  $R_b^r(\psi, \theta, \phi)$ :**

Rotasjonsmatrisen som viser hvordan en vektor fra  $\mathcal{F}^b$  kan representeres i  $\mathcal{F}^r$ . Denne består av rotasjoner i 3-2-1 eulervinkler. Jeg velger denne rotasjonssekvensen da dette vil gi gunstige utfall under estimeringen av stillingen senere i oppgaven.  $R_r^b$  blir da sammensatt av de 3 elementære rotasjonsmatrisene  $C_x(\phi), C_y(\theta)$  og  $C_z(\psi)$ . Rotasjonene følger rotasjonssekvensen  $R_b^r(\psi, \theta, \phi)$  og blir:

$$R_b^r(\psi, \theta, \phi) = C_z(\psi) \cdot C_y(\theta) \cdot C_x(\phi) \quad (5.1)$$

$$= \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (5.2)$$

**Matrisen  $R_{bi}^{b'}$**

Denne matrisen vil inneha forskjellig verdi avhengig av om vi måler i posisjon 1 eller 2. Jeg benevner derfor denne med subscript  $i$ , der  $i$  angir måling 1 eller 2. For posisjon  $i = 1$  vil de to systemene sammenfalle og matrisen blir da identitetsmatrisen:

$$R_{b1}^{b'} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

For posisjon  $i = 2$  vil system  $b'$  være rotert  $180^\circ$  om  $y$  akse i forhold til  $b$  og matrisen ser da slik ut:

$$R_{b2}^{b'} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (5.4)$$

**Matrisen  $R_{b'}^a$ :**

Rotasjonsmatrisen som viser hvordan en vektor fra  $\mathcal{F}^{b'}$  kan representeres i  $\mathcal{F}^a$ .  $\mathcal{F}^a$  legges langs Akselerometerets sensitivitetsakser. Aksene er montert på hengselen, orientert slik at  $\epsilon_1, \epsilon_2$  og  $\epsilon_3$  er eulervinklene mellom  $\mathcal{F}^{b'}$  og  $\mathcal{F}^a$  sine  $x, y$  og  $z$  akser. Dette gjør at  $R_{b'}^a$  består av rotasjonene  $\epsilon_1, \epsilon_2$  og  $\epsilon_3$  og

matrisen blir som følger:

$$R_{b'}^a(\epsilon_3, \epsilon_2, \epsilon_1) = C_z(\epsilon_3) \cdot C_y(\epsilon_2) \cdot C_x(\epsilon_1) \quad (5.5)$$

$$= \begin{bmatrix} c_{\epsilon_3} c_{\epsilon_2} & c_{\epsilon_3} s_{\epsilon_2} s_{\epsilon_1} - s_{\epsilon_3} c_{\epsilon_1} & s_{\epsilon_3} s_{\epsilon_1} + c_{\epsilon_3} s_{\epsilon_2} c_{\epsilon_1} \\ s_{\epsilon_3} c_{\epsilon_2} & c_{\epsilon_3} c_{\epsilon_1} + s_{\epsilon_3} s_{\epsilon_2} s_{\epsilon_1} & s_{\epsilon_3} s_{\epsilon_2} c_{\epsilon_1} - c_{\epsilon_3} s_{\epsilon_1} \\ -s_{\epsilon_2} & c_{\epsilon_2} s_{\epsilon_1} & c_{\epsilon_2} c_{\epsilon_1} \end{bmatrix} \quad (5.6)$$

$\epsilon_1, \epsilon_2$  og  $\epsilon_3$  er oppgitt som eulervinkler, og matrisen følger sekvensen 3-2-1 eulervinkler.

Jeg har nå funnet transformasjonene mellom de forskjellige rammene. De transformasjonene som ikke er funnet over kan gjøres ved å transformere via en av de andre rammene og egenskapen ved at  $R$  er en orthogonal matrise som fører til:

$$R_r^b = (R_b^r)^{-1} = (R_b^r)^T \quad (5.7)$$

Jeg skal nå bruke de overnevnte rammene og transformasjonene til to ting. Først vil jeg generere målinger fra sensorene beskrevet i kap. 4.3 basert på en simulert nordvinkel og skjevstilling. Deretter vil jeg lage en algoritme som estimerer nordvinkel basert på målinger av jordrotasjonskomponent og gravitasjonskomponenter. Denne estimatoren vil jeg i neste kapittel teste med de simulerte målingene.

## 5.2 Simulering av målinger

I denne delen vil det bli gjennomgått hvordan jeg simulerer målingene fra sensorene. Jeg har utviklet støymodeller for sensorene tidligere i oppgaven og disse vil bli brukt for å lage målinger av jordrotasjon og statisk akselasjon.

### 5.2.1 Gravitasjonsmålinger

Jeg bruker akselerometeret til å måle jordas gravitasjonskomponent langs de forskjellige aksene i  $\mathcal{F}^a$ . Jordas gravitasjon er en akselerasjon med størrelse  $1g = 9.81^m/s^2$ . Denne er representert i  $\mathcal{F}^r$  langs  $z^r$  akse. Akselerasjonen tatt opp i de 3 akselerometerne representert i  $\mathcal{F}^a$ , for posisjon  $i$ , med enheten  $[g]$ er da gitt ved:

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}_i^a = R_{b'}^a(\epsilon_3, \epsilon_2, \epsilon_1) \cdot R_{bi}^{b'} \cdot R_r^b(\psi, \theta, \phi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^r \quad (5.8)$$

$$\Downarrow$$

$$\underline{a}_i^a = R_{b'}^a(\epsilon_a) \cdot R_{bi}^{b'} \cdot R_r^b(\psi, \theta, \phi) \cdot \underline{a}^r \quad (5.9)$$

Settes dette inn for den sanne akselerasjonen i (4.41), får man følgende modell for målingene til akselerometeret i posisjon  $i$  som funksjon av vinklene  $\psi, \theta, \phi$  og  $\epsilon_i$ :

$$\underline{a}_{mki}^a = R_{b'}^a(\epsilon_3, \epsilon_2, \epsilon_1) \cdot R_{bi}^{b'} \cdot R_r^b(\psi, \theta, \phi) \cdot \underline{a}^r + \underline{v}_{ak}^a + \underline{b}^a \quad (5.10)$$

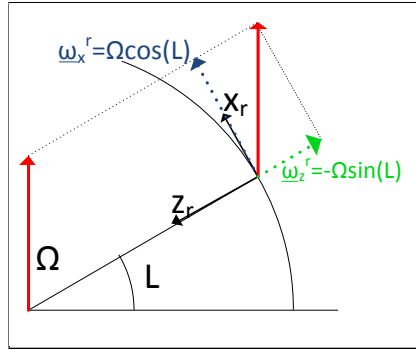
Med matrisene:

$$\underline{b} \sim N\{\underline{0}, Q_b^a\}, \quad \underline{v}_{ak}^a \sim N\{\underline{0}, Q_{ak}^a \delta_{k,s}\} \quad (5.11)$$

$$Q_b^a = \begin{bmatrix} q_b^x & 0 & 0 \\ 0 & q_b^y & 0 \\ 0 & 0 & q_b^z \end{bmatrix}^a, \quad Q_{aki} = \begin{bmatrix} q_{ak}^x & 0 & 0 \\ 0 & q_{ak}^y & 0 \\ 0 & 0 & q_{ak}^z \end{bmatrix}^a \quad (5.12)$$

Gjør oppmerksom på at  $b$  her benevner bias og ikke ramme  $b$  slik som i likningene over.

### 5.2.2 Gyromålinger



Figur 5.3: Illustrasjon av projeksjonen av  $\Omega$  ned i  $\mathcal{F}^r$ .

Gyroen skal brukes til å generere målinger av  $\omega_i$  langs  $x^{b'}$ -aksen. Når breddegraden er kjent, kan jordrotasjonen om jordas hovedakse dekomponeres i  $\mathcal{F}^r$  som en komponent langs  $x^r$  og en langs  $z^r$ . Jordrotasjonen representert i  $\mathcal{F}^r$  er dermed:

$$\omega^r = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}^r = \begin{bmatrix} \Omega \cos(L) \\ 0 \\ -\Omega \sin(L) \end{bmatrix}^r \quad (5.13)$$

$$(5.14)$$

$\omega^r$  tatt opp av gyroen er da representert langs  $x$ -aksen i  $\mathcal{F}^{b'}$ , ved måling  $i$ , som:

$$\underline{\omega}_i^{b'} = R_{bi}^{b'} \cdot R_r^b(\psi, \theta, \phi) \cdot \underline{\omega}^r \quad (5.15)$$

$$(5.16)$$

Den 1-aksede gyroen jeg har modellert i (4.37) er orientert langs aksen  $y^{b'}$  og målingen vi registrer på denne er da gitt som:

$$\omega_{mki} = H \cdot R_{bi}^{b'} \cdot R_r^b(\psi, \theta, \phi) \cdot \underline{\omega}^r + \beta_k + v_{\gamma k} + b \quad (5.17)$$

der  $H = [1 \ 0 \ 0]$  er en målematrise for kun å måle  $y$ -aksen.  $b$  i siste ledd benevner her bias.

Jeg har nå en god simulering av både akselerometer og gyromålingene og kan fortsette med å lage en estimator for vinklene basert på målingene.

### 5.3 Estimering av $\hat{\theta}$ , $\hat{\phi}$ og $\hat{\psi}$

Jeg skal nå lage en simuleringsmodell som først estimerer skjevstillingen til kompasset,  $\hat{\phi}$  og  $\hat{\theta}$ , ved hjelp av målingene fra akselerometerne. Disse vinklene skal brukes til å finne et estimat for vinkelen  $\hat{\psi}$ .  $\hat{\psi}$  kan da brukes til å finne nord da denne angir vinkel mellom kompasset og sann nord.

Estimatene baserer seg på de midlede målingene,  $\bar{\omega}_{mi}$  og  $\bar{a}_{mi}^a$  fra posisjon  $i$ :

$$\bar{\omega}_{mi} = \frac{1}{k} \cdot \sum_{k=1}^n \omega_{mki} \quad (5.18)$$

$$\bar{a}_{mi}^a = \frac{1}{N} \cdot \sum_{k=1}^N \underline{a}_{mki}^a \quad (5.19)$$

#### Estimering av stillingen representert ved $\hat{\theta}$ og $\hat{\phi}$ :

Jeg skal først se på estimeringen av  $\theta$  og  $\phi$ . Disse skal beregnes basert på akselerometermålingene av gravitasjonen. Store deler av fremgangsmåten er hentet fra [13] med noen modifikasjoner. Jeg estimerer stillingen i både posisjon 1 og 2 og beregner endelig stilling som et middel av disse for å øke nøyaktigheten. Målingene gjøres i  $\mathcal{F}^a$  men jeg velger å representere dem i  $\mathcal{F}^b$  siden det er stillingen til  $b$  som er interessant og vinklene  $\epsilon_1, \epsilon_2$  og  $\epsilon_3$  er kjente. Sammenhengen mellom  $\bar{a}_{mi}^a$  og vinklene  $\theta$  og  $\phi$  er da gitt ved:

$$R_{b'i}^b \cdot R_a^{b'}(\epsilon_3, \epsilon_2, \epsilon_1) \cdot \bar{a}_{mi}^a = R_r^b(\psi, \theta, \phi) \cdot \underline{a}^r \quad (5.20)$$

Valget av 3-2-1 eulervinkler gjør at  $\psi$  forsvinner når vi multipliserer ut venstre siden over og vi står igjen med:

$$\underline{a}_i^b = R_{b'i}^b \cdot \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}^{b'} = \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \cdot \sin(\phi) \\ \cos(\theta) \cdot \cos(\phi) \end{bmatrix} \quad (5.21)$$



$\underline{a}_i^b$  er gitt ved:

$$\begin{bmatrix} a_{x_i}^b \\ a_{y_i}^b \\ a_{z_i}^b \end{bmatrix} = R_{b'i}^b \cdot \begin{bmatrix} c_{\epsilon_3} c_{\epsilon_2} \bar{a}_x^a + s_{\epsilon_3} c_{\epsilon_2} \bar{a}_y^a - s_{\epsilon_2} \bar{a}_z^a \\ (c_{\epsilon_3} s_{\epsilon_2} s_{\epsilon_1} - s_{\epsilon_3} c_{\epsilon_1}) \bar{a}_x^a + (c_{\epsilon_3} c_{\epsilon_1} + s_{\epsilon_3} s_{\epsilon_2} s_{\epsilon_1}) \bar{a}_y^a + c_{\epsilon_2} s_{\epsilon_1} \bar{a}_z^a \\ (s_{\epsilon_3} s_{\epsilon_1} + c_{\epsilon_3} s_{\epsilon_2} c_{\epsilon_1}) \bar{a}_x^a + (s_{\epsilon_3} s_{\epsilon_2} c_{\epsilon_1} - c_{\epsilon_3} s_{\epsilon_1}) \bar{a}_y^a + c_{\epsilon_2} c_{\epsilon_1} \bar{a}_z^a \end{bmatrix} \quad (5.22)$$

Her er alle verdiene kjent når vi kjenner målingen  $a_i^a$  og måleposisjonen  $i$ .

Stillingen til kompasset er nå uttrykt som funksjon av alle 3 målingene i (5.21) og med litt regning finner man  $\theta$  og  $\phi$  som:

$$\theta_i = \tan^{-1} \left( \frac{a_{y_i}^b}{a_{z_i}^b} \right) \quad (5.23)$$

$$\phi_i = \tan^{-1} \left( \frac{-a_{x_i}^b}{\sqrt{a_{y_i}^b{}^2 + a_{z_i}^b{}^2}} \right) \quad (5.24)$$

Jeg har nå funnet et uttrykk for stillingen til kompasset i de 2 stillingene  $i = 1, 2$ . Ved å middle estimatene fra posisjon 1 og 2 vil jeg få et bedre estimat av vinklene  $\theta$  og  $\phi$ . Dette gir følgende estimat for  $\theta$  og  $\phi$ :

$$\hat{\theta} = \frac{1}{2} (\theta_1 + \theta_2) \quad (5.25)$$

$$\hat{\phi} = \frac{1}{2} (\psi_1 - \psi_2) \quad (5.26)$$

### Estimering av $\hat{\psi}$

Estimatoren for  $\hat{\phi}$  og  $\hat{\theta}$  kan nå brukes til å estimere nordvinkelen  $\hat{\psi}$  ved å se på målingene fra gyroen som er gitt langs x-aksen i  $\mathcal{F}^b$ . Sammenhengen mellom gyromålingen  $\bar{\omega}_{mi}^b$ , skjjevstillingen  $\hat{\phi}$ ,  $\hat{\theta}$  og  $\hat{\psi}$  er da gitt ved:

$$\bar{\omega}_{mi} = [1 \ 0 \ 0] \cdot R_{bi}^b \cdot R_r^b(\psi, \theta, \phi) \cdot \begin{bmatrix} \Omega \cos(L) \\ 0 \\ -\Omega \sin(L) \end{bmatrix}^r \quad (5.27)$$

Ved å sette inn for  $R_{b1}^b$  og  $R_r^b(\psi, \theta, \phi)$  og regne sammen blir sammenhengen mellom  $\bar{\omega}_{m1}$  i posisjon 1 og  $\psi$ :

$$\bar{\omega}_{m1} = [c_\psi c_\theta \quad s_\psi c_\theta \quad -s_\theta] \cdot \begin{bmatrix} \Omega \cos(L) \\ 0 \\ -\Omega \sin(L) \end{bmatrix}^r \quad (5.28)$$

$$= \cos(\psi) \cdot \cos(\theta) \cdot \cos(L) \cdot \Omega + \sin(\theta) \cdot \sin(L) \cdot \Omega \quad (5.29)$$

Tilsvarende kan man sette inn  $R_{b2}^{b'}$  og  $R_r^b(\psi, \theta, \phi)$  for å finne  $\bar{\omega}_{mk2}$  uttrykt med  $\psi$ :

$$\bar{\omega}_{m2} = \begin{bmatrix} -c_\psi c_\theta & -s_\psi c_\theta & s_\theta \end{bmatrix} \cdot \begin{bmatrix} \Omega \cos(L) \\ 0 \\ -\Omega \sin(L) \end{bmatrix}^r \quad (5.30)$$

$$= -\cos(\psi) \cdot \cos(\theta) \cdot \cos(L) \cdot \Omega - \sin(\theta) \cdot \sin(L) \cdot \Omega \quad (5.31)$$

Som vi ser har målingene forskjellig fortegn. Ved trekke de to målingene fra hverandre vil jeg fjerne biasen i målingene. Sammenhengen kan da settes opp:

$$\bar{\omega}_{m1} - \bar{\omega}_{m2} = 2 \cdot \cos(\psi) \cdot \cos(\theta) \cdot \cos(L) \cdot \Omega + 2 \cdot \sin(\theta) \cdot \sin(L) \cdot \Omega \quad (5.32)$$

og en estimator for  $\psi$  blir da:

$$\hat{\psi} = \cos^{-1} \left( \frac{\bar{\omega}_{m1} - \bar{\omega}_{m2} - 2 \cdot \sin(\hat{\theta}) \cdot \sin(L) \cdot \Omega}{2 \cdot \cos(\hat{\theta}) \cdot \cos(L) \cdot \Omega} \right) \quad (5.33)$$

Som vi ser avhenger ikke estimatet av  $\hat{\phi}$ . Dette kommer av at  $\hat{\phi}$  er rotasjon om gyroens sensitivitetsakse og har ingen innvirkning på målingen.

Jeg har nå utviklet formler for å estimere nordvinkelen basert på målinger av jordrotasjonen med en gyro, og skjevstillingen til denne målt med akselerometer. Disse formlene kan nå brukes til å simulere gyrokompasset.

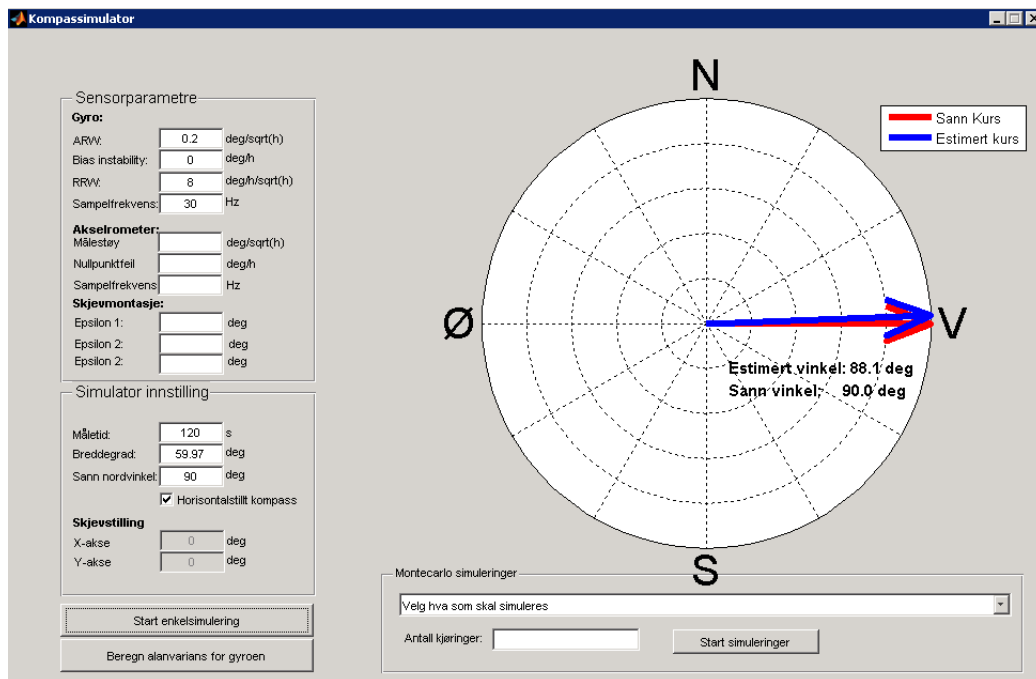
## 5.4 Implementering av modellen i MatLab

For å gjøre simuleringer av systemet implementerer jeg likningene jeg har funnet over i MatLab. På grunn av kompliserte ulinneære likninger vil en utregning av variansen for  $\hat{\psi}$  som funksjon av de forskjellige parameterne som inngår være vanskelig. Jeg velger derfor å gjøre Monte-Carlo simuleringer av kompasset i forskjellige posisjoner for å finne hvilke usikkerhetsmomenter som har størst innvirkning på estimatet.

For å enkelt å kunne endre parameterne som inngår har jeg laget en GUI til simulatoren. Dette gjør at simulatoren får et mer brukervennlig grensesnitt og vil i fremtiden kunne brukes for å undersøke hvordan andre og bedre typer gyroer vil fungere i et gyrokompass basert på vippeprinsippet.

Jeg vil ikke gå gjennom Matlab programmet i detalj ettersom det i grove trekk følger formelverket som er gitt tidligere i oppgaven. GUIen for programmet kan ses i figur. 5.5. Den har et relativt enkelt grensesnitt der man legger inn data om gyroen og akselerometeret samt kompassets orientering i

## 5.5. SAMMENLIGNING AV SIMULATOREN MED FORSØK PÅ RATEBORD49

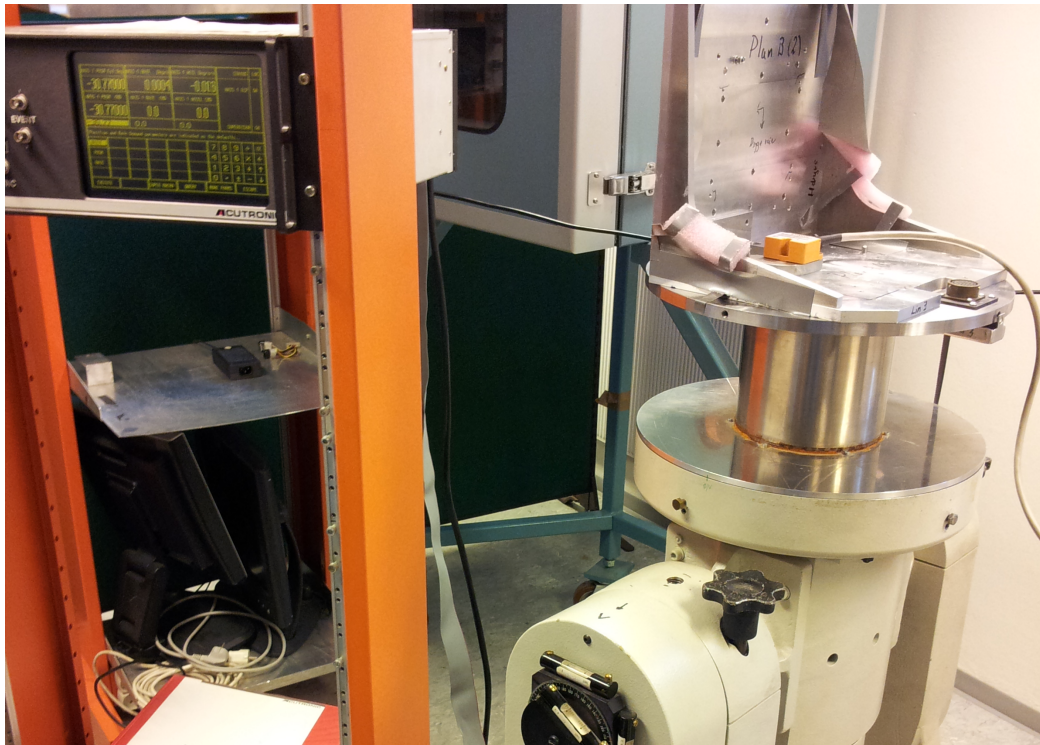


Figur 5.4: GUIen til simmulatoren.

forhold til nord. Når man trykker start vil det kjøres 1 simulering av kompasset i den innlagte posisjonen, og resultatet vises i kompassrosen på siden. Å kjøre en simulering er imidlertid ikke veldig interessant, ettersom dette ikke gir noe inntrykk av hvor godt kompasset virkelig presterer. Mulighet for forskjellige Monte-Carlo simuleringer er derfor implementert nederst. Her har jeg laget et sett med forskjellige simuleringer som samlet vil gi et godt bilde av hvilke feilbidrag som vil ha størst innvirkning på nøyaktigheten. Brukeren velger selv hvor mange simuleringer han vil kjøre. Men for at simulatoren skal gi et godt bilde av den sanne usikkerheten anbefaler jeg å bruke over 1000 simuleringer. Når man trykker start vil en status linje fortelle hvor langt man er kommet og når simuleringene er ferdig vil resultatet vises i en figur der kompassrosen ligger.

## 5.5 Sammenligning av simulatoren med forsøk på ratebord

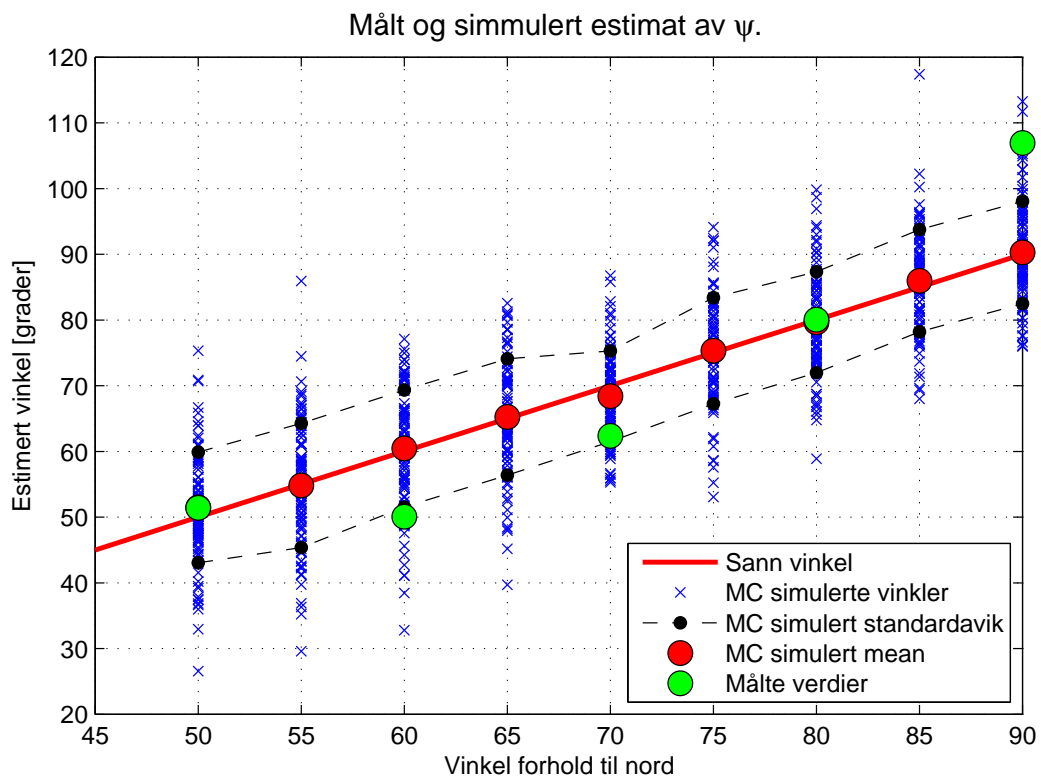
Jeg har gjort et forsøk på å estimere nordvinkelen basert på målinger som jeg har gjort på et ratebord hos FFI. Dette ratebordet ble stilt opp i horisontalplanet. På ratebordet ble gyroen montert slik at måleaksen lå parallelt med



Figur 5.5: Bildet viser oppsettet av gyroen plassert på ratebordet. Til venstre ses styreenheten for ratebordet der nordvinkelen kan stilles inn.

rotasjonsplanet. Deretter ble det tatt opp målinger der gyroen ble plassert i vinklene  $45^\circ$ ,  $50^\circ$ ,  $60^\circ$ ,  $70^\circ$ ,  $80^\circ$  og  $90^\circ$  i forhold til sann nord. I hver posisjon ble det tatt opp en måling på 5 minutter. Mellom hver vinkel ble ratebordet rotert  $180^\circ$  og en tilsvarende måling ble tatt opp i denne posisjonen. Dette tilsvarer å vippe gyroen  $180^\circ$ . Deretter ble målingene prosessert i Matlab og nordvinkelen estimert basert på de 156 første sekundene av hver måling. Resultatet fra de 6 forsøkene er plottet i figur 5.6 sammen med en Monte-Carlo simulering av de samme dataene og viser at simulatoren fungerer brukbart i forhold til målinger gjort på det virkelige utstyret.

5.5. SAMMENLIGNING AV SIMULATOREN MED FORSØK PÅ RATEBORD51



Figur 5.6: Nordsøk gjort på lab sammenlignet med simulerte målinger.



# Kapittel 6

## Simuleringsresultater

I dette kapitlet vil det bli presentert simuleringsresultater, basert på gyroen STIM210, for å vise hvordan variansen til nordvinkelen påvirkes av de forskjellige usikkerhetsmomentene. Simuleringene baserer seg på data fra gyroens datablad siden det ikke er gjort noen målinger på STIM210.

Det viste seg imidlertid i kap 4.3.1 at målingene vi fikk til å ta opp av gyroen var langt dårligere enn det databladet skulle tilsi. Estimaten i dette kapitlet er derfor muligens noe bedre enn hva som er praktisk mulig å få til.

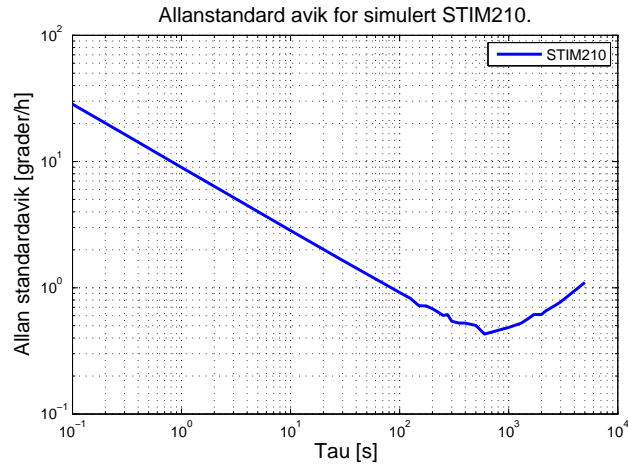
Siden det ikke er gjort målinger på STIM210 kan det heller ikke fastslås om antakelsen at støyen kun består av ARW og RRW støy holder. Når man ser på Allanvariansen som Sensoror har beregnet for STIM210 gitt i [11] kan det se ut til at denne også inneholder noe korrellert målestøy, siden. Dette blir imidlertid bare synsing og kan ikke fastslås uten å gjøre målinger på gyroen, eventuelt innhente beregningsgrunlaget til produsenten.

### 6.1 Optimal måletid i hver posisjon.

Først har jeg sett på hva som er optimal midlingstid av sensorene. Antakelig vil optimal midlingstid sammenfalle med bunnpunktet i Allanvariansplottet for gyroen, som kan ses i figur 6.1. Bunnpunktet ligger her rundt 600s. Jeg velger derfor å kjøre 1000 Monte-Carlo simuleringer av prosessen der jeg estimerer  $\psi$  for forskjellige midlingstider. Jeg velger å se på hvert 20 sekund fra 20 til 1000 sekunder. Den optimale midlingstiden vil da være den som gir minst spredning i estimatet. Gyroen er stillt i horisontalplanet, og innstilt på  $\psi = 80^\circ$ .

Resultatet er vist i figur 6.2 og viser at for middlingstider over 300s, eller 5 min, vil vi ikke oppnå særlig mye bedre estimater. Dette vil gi en total kompasseringstid på 10 min. Resultatet stemmer godt overens med plottet

av Allanstandardaviket. Som vi også ser er det ikke mye å tjene på å øke midlingstiden utover 300s. Kompasseringstiden kan også settes lavere, men da vil det ganske raskt gå ut over nøyaktigheten.



Figur 6.1: Allanstandardavik for simulert STIM210 basert på måleserie på 100000s.

## 6.2 Optimal initiell innstilling av kompasset

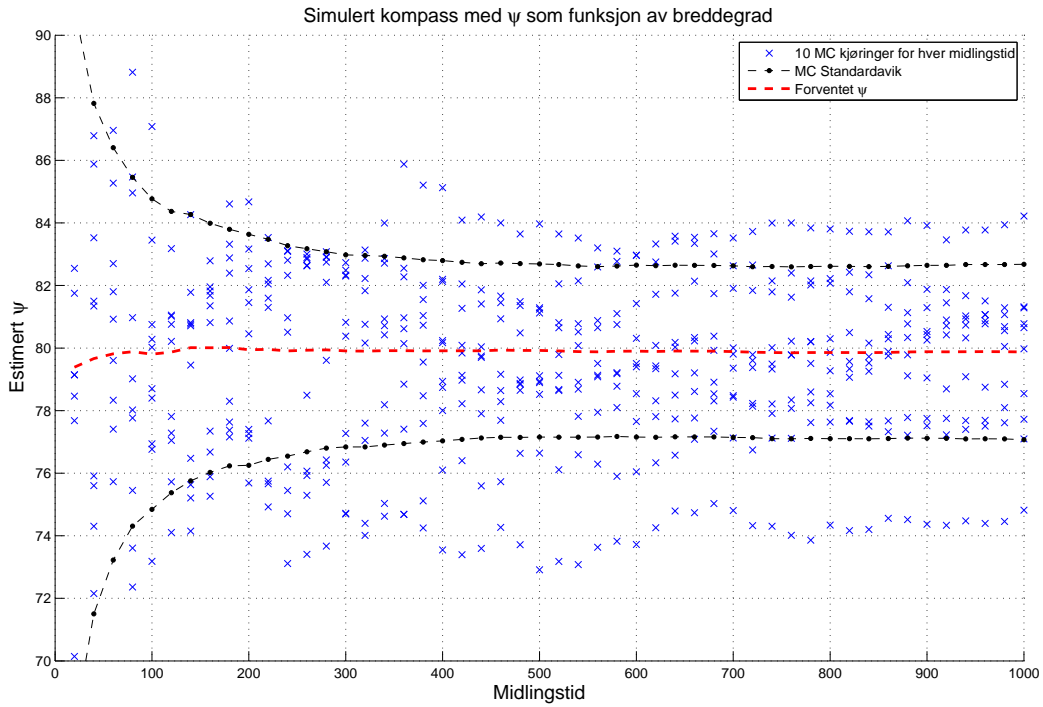
Ved å bruke midlingstiden funnet over vil jeg undersøke hvordan initiell oppstilling av kompasset påvirker målingene. De parameterne som vil bli undersøkt er:

- Initiell innstilling av kompasset i horisontalplanet.
- Betydningen av skrå oppstilling.
- Breddegrad

### 6.2.1 Initiell $\psi$ retning

I oppgaveteksten er det oppgitt at initiell innstilling antakelig er i nærheten av øst-vest retning. En måling for nær øst-vest retning bør sannsynligvis også unngås ettersom  $\bar{\omega}_1$  og  $\bar{\omega}_2$  her vil kunne få likt fortegn og dermed kansellerer hverandre. Jeg ønsker derfor å bestemme et intervall,  $\langle \Delta_{min}, \Delta_{maks} \rangle$  for aviket,  $\Delta$ , fra øst-vest retning som man bør prøve å stille systemet i med feks. et magnetkompass.





Figur 6.2: Monte-Carlo simulering av gyrokompass med varierende midlingstid fra 20-1000 sek.

$\Delta_{min}$ : For å finne en minste verdi for  $\Delta$  må vi finne en vinkel som sannsynliggjør at  $\bar{\omega}_1$  og  $\bar{\omega}_2$  får forskjellig fortegn. Ved å sette opp ulikheten:

$$E\{\bar{\omega}(\psi)_{1k}\} + \sqrt{Var\{\bar{\omega}(\psi)_{1k}\}} > 0 \quad (6.1)$$

Løses denne mhp.  $\psi$ , vil vi finne en  $\psi$  som gjør at  $\bar{\omega}_{1k}$  med 68% sannsynlighet har positivt fortegn. Denne prosenten kan økes ved å øke antall standard avik i ulikheten. Siden  $\bar{\omega}(\psi)_{2k} = \bar{\omega}(\psi + \pi)_{1k}$  vil da  $\bar{\omega}(\psi)_{1k}$  og  $\bar{\omega}(\psi)_{2k}$  ha forskjellig fortegn. Gitt et kompass opstilt i horisontalplanet vil forventning og varians for gyromodellen vær egitt ved 4.45 og 4.50:

$$Var\{\bar{\omega}_{1k}\} = P_{\omega_s} + \frac{1}{k^2} \left( q_\gamma \cdot k + q_{\beta d} \cdot \sum_{i=1}^k k^2 \right) \quad (6.2)$$

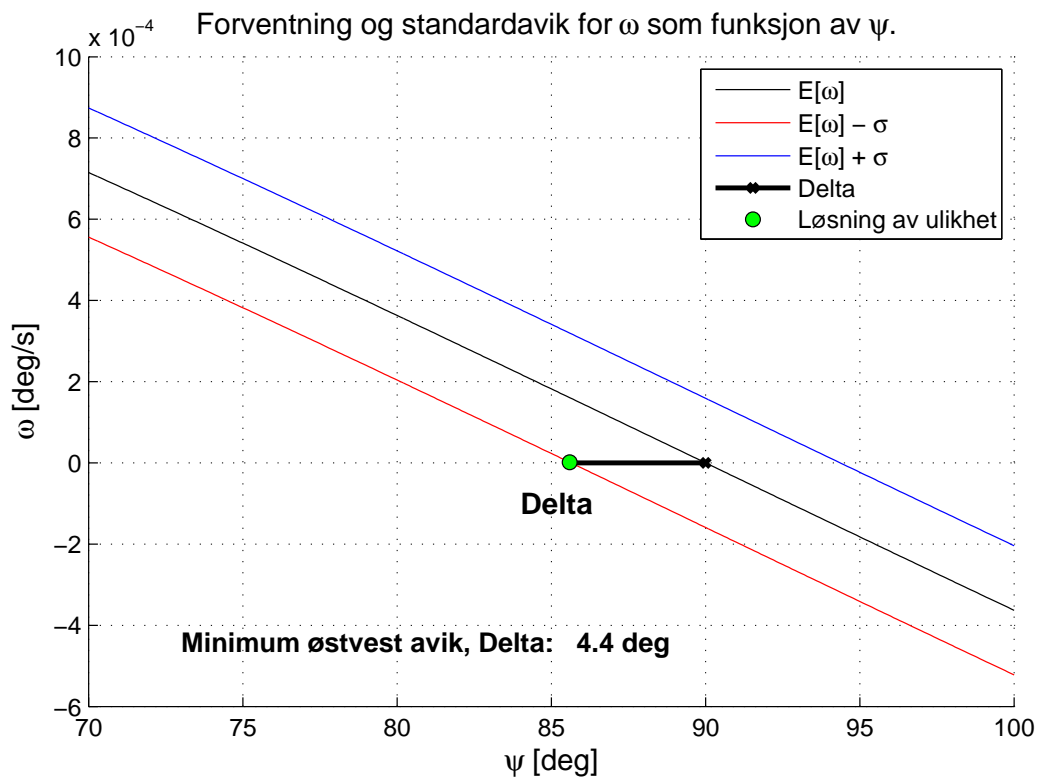
og forventningsverdi:

$$E\{\bar{\omega}_{1k}\} = E\{\omega_s\} + b \quad (6.3)$$

Vi kan i dette tilfellet se bort fra biasen, siden denne kanselleres når vi slår sammen målingene. Settes disse inn i ulikheten fås:

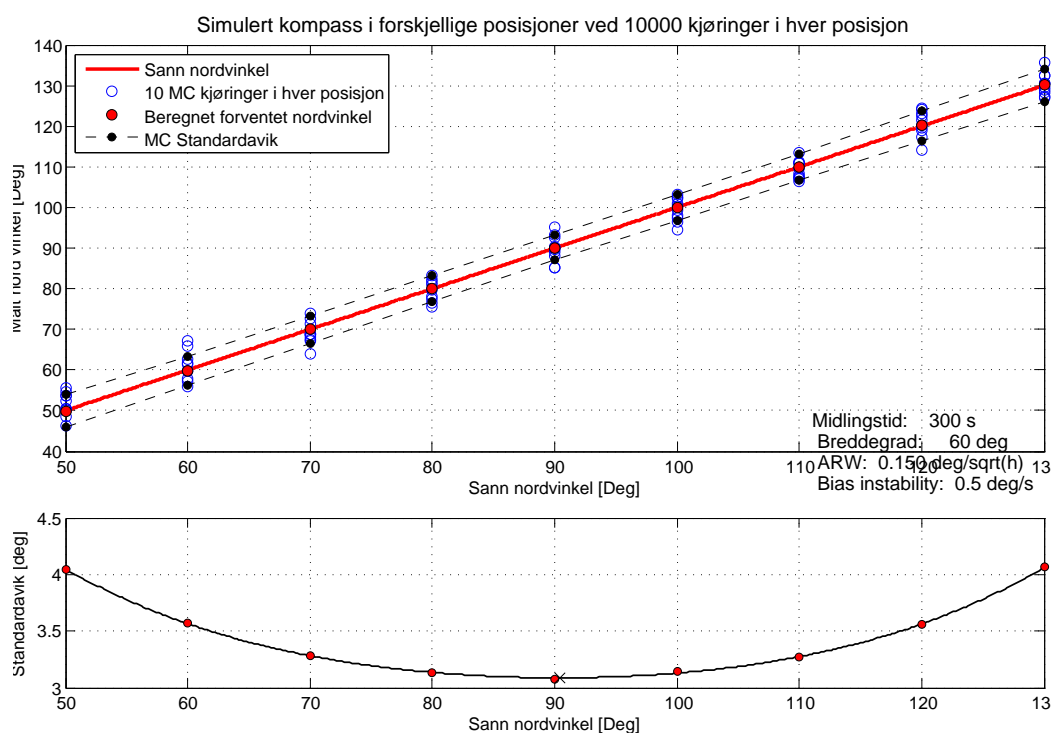
$$\Omega \cos(L) \cos(\Psi) + \sqrt{P_{\omega_s} + \frac{1}{k^2} \left( q_\gamma \cdot k + q_{\beta d} \cdot \sum_{i=1}^k k^2 \right)} > 0 \quad (6.4)$$

Løser likningen med  $T = 300s$  og samplingsfrekvens på  $1000Hz$ . Løsningen er vist i figur 6.3 og som vi ser må  $\psi$  være mindre enn  $85,6^\circ$ , som betyr at  $\Delta$  bør være minst  $4,4^\circ$ .



Figur 6.3: Figuren viser forventet jordrotasjonskomponent som funksjon av  $\psi$ . Løsning av ulikheten over er merket inn og angir største initielle  $\psi$  som er gunstig for å få gode målinger.

**Største gunstige  $\Delta$**  Dersom  $\Delta$  blir for stor vil dette gå kraftig ut over nøyaktigheten på målingene. Dette kommer av at estimeringsformelen inneholder arcos funksjonen. Denne funksjonen er definert på intervallet  $x = \langle -1, 1 \rangle$  og er veldig ulineær. Rundt 0 er den imidlertid ganske lineær. Og det vil derfor



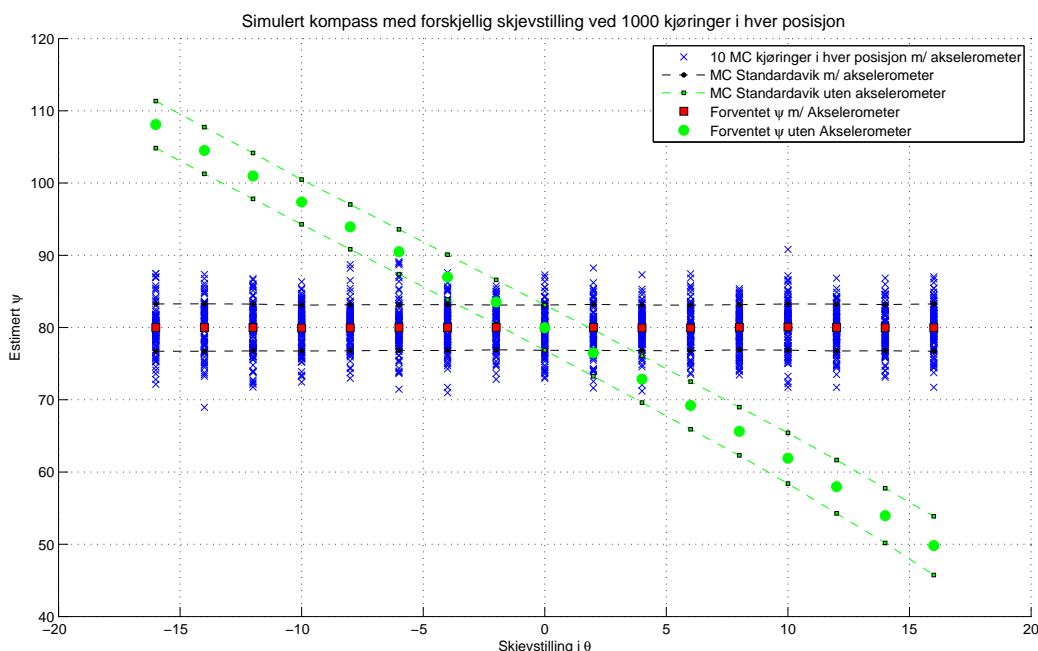
Figur 6.4: Øverst: forventningsverdi og standardavik for kompasset som funksjon av initiell vinkel. Hver 1000 montecarlo simulering er plottet for illustrasjon. Nederst: Kurvetilpasset standardavik hvor minimum er plottet inn, i tillegg til beregningspunktene.

være gunstig for variansen til  $\psi$  at målingene ligger nærmest mulig  $90^\circ$  slik at funksjonen havner rundt 0. For å finne hvor stor innvirkning forskjellige initielle vinkler vil ha på variansen i estimatet har jeg derfor kjørt Montecarlo simuleringer av kompasset med forskjellig initilell  $\psi$ . Forsøket er gjennomført ved å stille kompasset i en vinkel, kjøre 10000 simuleringer for så å beregne forventning og varians i denne posisjonen basert på disse målingene. Deretter er initiell  $\psi$  økt med  $10^\circ$  for så å gjenta forsøket. Når så dette er gjort for vinkler fra  $50^\circ$  til  $130^\circ$  er deet foretatt en kurvetilpassning for variansen. Resultatet er vist i fig. 6.4. Som vi ser vil variansen øke ganske kraftig dersom initielt øst-vest avik,  $\Delta$ , blir større enn ca: 15-20°.

Resultatene viser at det i sum beste resultatet for nordsøker basert på STIM210 får man dersom man initielt stiller kompasset i en retning på mellom  $75^\circ$  og  $85^\circ$  i forhold til nord. Å få plassert utstyret i et intervall på 10 grader skal være mulig å få til uten store vanskeligheter med et standard magnetkompass.

## 6.2.2 Betydningen av skjevstilling

Jeg har undersøkt hvordan en skjev oppstilling av kompasset vil virke inn på estimatet av  $\psi$ . Skjevstillingen er gitt av de to vinklene  $\phi$  og  $\theta$ .  $\phi$  er en rotasjon om gyroens sensitivitetsakse, og vil derfor ikke ha noen innvirkning på estimatet av  $\psi$ . Jeg har derfor ikke sett variasjoner i  $\phi$ . Simuleringene er gjort ved å gjøre 10000 Monte Carlo simuleringer ved forskjellige innstillinger av  $\theta$ . Jeg har simulert for annenhver grad i intervallet  $[-16^\circ, 16^\circ]$ . Resultatet er vist i figur: 6.5 Som vi kan se vil det innenfor dette intervallet være likegyldig hvordan kompasset stilles opp dersom stillingen estimeres med et akselerometer. Simuleringen er gjort med parametere gitt i tabell 6.1.



Figur 6.5: Simulering av  $\psi$  som funksjon av skjevstilling, både med å uten akselerometer.

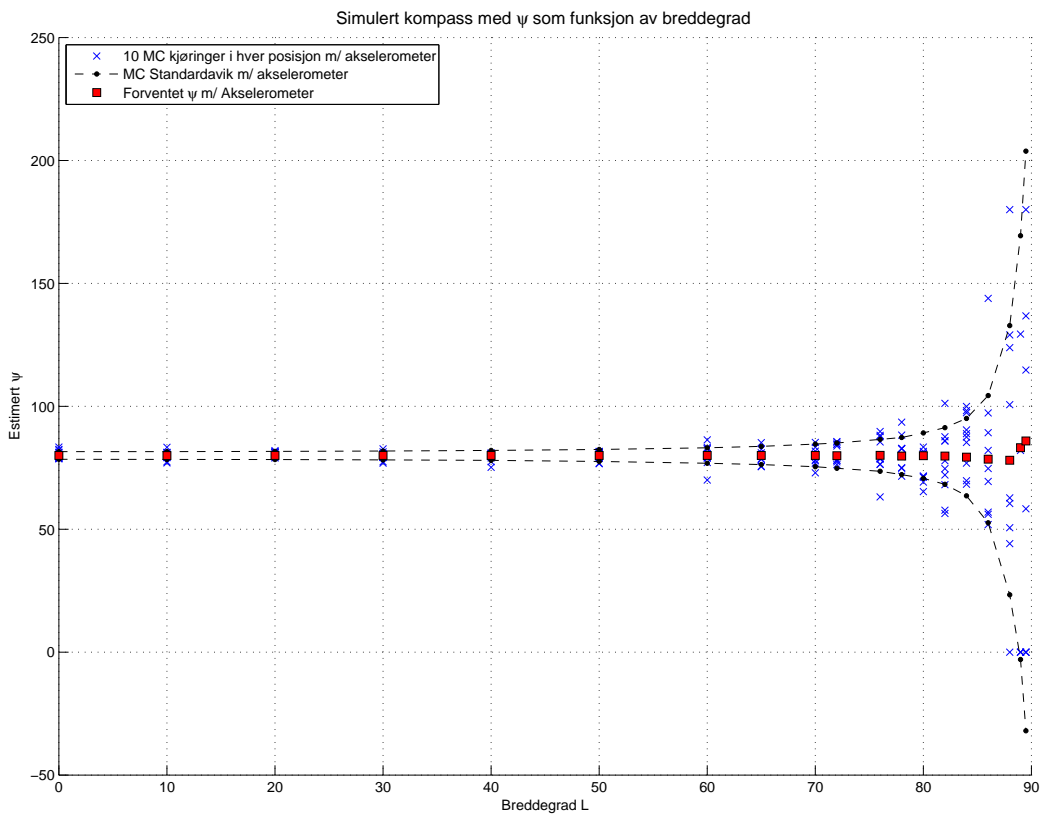
Jeg har også sett på hvordan en eventuell skjevstilling vil virke inn dersom det ikke brukes et akselerometer for å beregne skjevstillingen, men kompasset likevel settes opp skjevt som følge av unøyaktig oppsetning. Dette er simulert på samme måte som over men her er  $\hat{\psi}$  beregnet gitt kompass i horisontalstilling. Standardaviket for denne oppsetningen er gitt er vist som grønn kurve i figur 6.5 og viser at dersom det ikke implementeres akselerometer i kompasset vil dette ganske raskt gå ut over forventningsverdien dersom kompasset settes skjevt. Det betyr at et akselerometer må implementeres i systemet dersom det skal fungere uten at tiden til horisontaloppretting blir

Parameter:	Verdi:
Breddegrad	$59.97^\circ$
Nordvinkel	$80^\circ$
Akselerometer målestøy	$30 \mu g / \sqrt{(h)}$
Akselerometer nullpunktfeil	$3mg$
Gyro ARW	$0.15^\circ / \sqrt{(h)}$
Gyro RRW	$8^\circ / s / \sqrt{(h)}$

Tabell 6.1: Parametere brukt ved simulering av  $\psi$  som funksjon av  $\theta$ .

alt for stor.

### 6.2.3 Nøyaktighet som funksjon av breddegrad

Figur 6.6: Simulering av  $\psi$  som funksjon av breddegrad.

Breddegraden må kjennes på forhånd og jeg regner i dette tilfellet denne som en deterministisk størrelse. Det jeg ønsker å undersøke her er om bred-

de graden har noe å si for målingene. Dess nærmere man kommer polene, dess mindre blir den komponenten vi skal måle, og man skulle tro støyen i gyroen fikk større virkning på høye breddegrader siden den da blir større del av den totale målingen. Dette har jeg testet ved å sette systemet opp med parameterne gitt i tabell 6.2 og simulert målinger ved hver tiende breddegrad fra ekvator og med økende frekvens opp mot 90 grader. Resultatet er vist i figur 6.6. Som vi ser vil et slik gyrokompass oppstilt i horisontalplanet ikke fungere godt på høye breddegrader. Dette kommer av at jordrotasjonen i horisontalplanet her blir så liten at den blir begravd i støyen. Hvor langt nord/sør vi vil få brukbare resultater avhenger av hvor god gyroen er og hvor stort krav som stilles til nøyaktighet. Generelt vil det være risikabelt å bruke et slikt kompass når breddegraden overstiger 75-80 grader nord/sør. Det vil bety at det stort sett vil fungere på fastlandet, men f problemer i arktis.

Parameter:	Verdi:
Nordvinkel	$80^\circ$
Skjevstilling $\phi, \theta$	$0^\circ, 0^\circ$
Akselerometer målestøy	$30 \mu g / \sqrt{(h)}$
Akselerometer nullpunktfeil	$3mg$
Gyro ARW	$0.15^\circ / \sqrt{(h)}$
Gyro RRW	$8^\circ / s / \sqrt{(h)}$

Tabell 6.2: Parametere brukt ved simulering av  $\psi$  som funksjon av breddegrad.

### 6.2.4 Feilbudsjett for sensorene

Jeg har nå sett på hvilke ytre forhold som påvirker målingene og hvordan disse forholdene bør være for at nordsøkeren skal fungere best mulig. Nå ønsker jeg å se på hvordan usikkerhetsmomentene i målingene fra sensorene bidrar til den totale feilen i estimatet av  $\psi$ . Dette har jeg gjort ved å sette opp ett feilbudsjett for systemet som inneholder alle feilkildene i gyroen og akselerometeret. Feilbudsjettet er gjort ved først å simulere systemet med alle feilene inne. Deretter er systemet simulert med en og en feil. Det vil da fremkomme hvor mye av usikkerheten som kommer fra hver enkelt feilkilde. Jeg har undersøkt alle de modellerte feilene i både gyroen og akselerometeret. Feilene som er undersøkt er:

- Gyroens Angular Random Walk.

- Gyroens Rate random Walk.
- Akselerometerets målestøy.
- Akselerometerets nullpunktsfeil.

Parameterne som er brukt i simmuleringen er gitt i tabell 6.3. Resultatet av simuleringene er gitt i tabell: 6.4. Som vi kan se her er det gyroparameterne

Parameter:	Verdi:
Nordvinkel	$80^\circ$
Skjevstilling $\phi, \theta$	$0^\circ, 0^\circ$
Akselerometer målestøy	$30 \mu g / \sqrt{(h)}$
Akselerometer nullpunktfeil	$3mg$
Gyro ARW	$0.2^\circ / \sqrt{(h)}$
Gyro RRW	$8^\circ / s / \sqrt{(h)}$
Monte-carlo simuleringer	10000

Tabell 6.3: Parametere brukt ved feilbudsjett.

som har størst innvirkning på hvor godt resultat man får. Akselerometeret gir kun et bidrag på under 10% av den totale feilen. Det er derfor ikke kritisk å få tak i et veldig nøyaktig akselerometer. Vi kan også se at ARW bidrar med noe mer enn RRW. Dette ville jevnet seg ut om vi hadde økt midlingstiden til 600s som er bunnpunktet i Allanstandardaviket.

Parameter:	Verdi:
Angular Random Walk	$2.84^\circ$
Rate Random Walk	$1.28^\circ$
Akselerometer målestøy	$0.0000034^\circ$
Akselerometer nullpunktfeil	$0.21^\circ$
Alle bidrag innlagt	$3.12^\circ$

Tabell 6.4: Resultat av feilbudsjett.





# Kapittel 7

## Konklusjon

I denne oppgaven er det undersøkt hvordan en MEMS gyro vil fungere i et gyrokompass. Det har også blitt undersøkt hvilke faktorer som er viktigst for å få et godt resultat. Jeg har utviklet en simulator som kan brukes til å undersøke hvordan forskjellige MEMS sensorer vil prestere i en nordsøking applikasjon. Basert på denne simulatoren har jeg undersøkt STIM210 og hvordan denne fungerer i nordsøking sammenheng. Gyroen er en av de beste på dagens marked og bør gi et godt bilde av hvor gode estimater det er mulig å oppnå.

Resultatet viser at dagens MEMS gyroer kan, under optimale forhold, oppnå en nordvinkeløyaktighet på i overkant av 3 grader. Dette er et resultat som ligger et godt stykke unna hva man trenger til militære formål og foreløpig vil et magnetkompass antakelig gi oss et vel så godt estimat av nordvinkelen som en MEMS basert Nordsøker. Resultatet fordrer også at man klarer å oppnå like gode målinger som llevrandøren hevder skal være mulig. Målinger gjort undervis i denne oppgaven tyder imidlertid på at dette kan være vanskelig å oppnå. Et gyrokompass basert andre teknologier vil derfor fortsatt være å foretrekke.

Et annet aspekt som er funnet i oppgaven er at horisontaloppretting av kompasset er helt essensielt for å få et godt estimat av nordvinkelen. Et akselerometer for å bestemme stillingen vil derfor være svært nyttig ekstra utstyr på nordsøkeren for å slippe tidkrevende horisontaloppretting. Feilbudsjettet viser også at dagens MEMS akselerometer vil være mer en gode nok til å gjøre denne jobben og at det kritiske for målingen er nøyaktigheten til gyroen.

Nordvinklestimatet er også til en viss grad breddegradsavhengig. Man skal imidlertid veldig langt nord før dette vil ha stor innvirkning.



# Kapittel 8

## Videre arbeid

Resultatene i denne oppgaven viser at nordsøking basert på MEMS gyroer fortsatt har et stort forbedringspotensiale. Det er derfor en rekke områder innefor emnet som kan være interessant å undersøke nærmere. Noen forslag kan være:

- Gjøre en undersøkelse av hvilke faktorer som gir denne oppgaven mye dårligere resultat enn leverandørens spesifikasjoner
- Bruk av MEMS-gyromatriser til nordsøking
- Nordsøk basert på gyro som roterer med konstant hastighet i horisontalplanet
- Undersøke om det er noe periodisitet i bias ustabiliteten og evt. lage algoritmer for å kompensere for dette.



# Bibliografi

- [1] Allan variance analysis on error characters of mems inertial sensors for an fpga-based gps/ins system. [http://www.xbow.com/pdf/Bias\\_Stability\\_Measurement.pdf](http://www.xbow.com/pdf/Bias_Stability_Measurement.pdf), Februar 2012.
- [2] Bias stability measurement: Allan variance. [http://www.xbow.com/pdf/Bias\\_Stability\\_Measurement.pdf](http://www.xbow.com/pdf/Bias_Stability_Measurement.pdf), Februar 2012.
- [3] Gauss-markov process - wikipedia. [http://en.wikipedia.org/wiki/Gauss-Markov\\_process](http://en.wikipedia.org/wiki/Gauss-Markov_process), Mars 2012.
- [4] An introduction to inertial navigation. <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>, Februar 2012.
- [5] Markov process - wikipedia. [http://en.wikipedia.org/wiki/Markov\\_process](http://en.wikipedia.org/wiki/Markov_process), Mars 2012.
- [6] Measuring spread. [http://syque.com/quality\\_tools/toolbook/Variation/measuring\\_spread.htm](http://syque.com/quality_tools/toolbook/Variation/measuring_spread.htm), Mai 2012.
- [7] Memes. [http://en.wikipedia.org/wiki/Microelectromechanical\\_systems](http://en.wikipedia.org/wiki/Microelectromechanical_systems), Februar 2012.
- [8] Micronews - sensoror gyro awarded the best high precision mems gyro in the market. <http://www.i-micronews.com/lectureArticle.asp?id=5902>, Mai 2012.
- [9] North finding system using a mems gyroscope. [http://www.tkt.cs.tut.fi/research/nappo\\_files/Iozan\\_ENC\\_2010\\_web.pdf](http://www.tkt.cs.tut.fi/research/nappo_files/Iozan_ENC_2010_web.pdf), Mai 2012.
- [10] Stim202-datasheet. <http://www.sensoror.com/media/84231/datasheet-stim202-miniature-gyro-module-ts1439r9.pdf>, Mai 2012.

- [11] Stim210-datasheet. <http://www.sensor.com/media/83029/datasheet-stim210-miniature-gyro-module-ts1545r7.pdf>, Mai 2012.
- [12] Supergyro i lommeformat. <http://www.tu.no/bragd/2011/10/31/kandidat-4-supergyro-i-lommeformat>, Mai 2012.
- [13] Tilt sensing using linear accelerometers. [http://www.freescale.com/files/sensors/doc/app\\_note/AN3461.pdf](http://www.freescale.com/files/sensors/doc/app_note/AN3461.pdf), Mai 2012.
- [14] White noise - wikipedia. [http://en.wikipedia.org/wiki/White\\_noise](http://en.wikipedia.org/wiki/White_noise), Mars 2012.
- [15] Odd Johan Blomdahl. Statisk gyrokompass med fiberoptisk gyro. *Masteroppgave ved Fysisk institutt, UIO*, 1998.
- [16] Jørn Grahn. Estimering av mems-gyro parametre. *Masteroppgave ved Fysisk institutt, UIO*, 2011.
- [17] Oddvar Hallingstad. Matematisk modellering av dynamiske systemer. *Kompendie i faget UNIK4540, UIO*, Oktober 2008.
- [18] Oddvar Hallingstad. Forelesning 13. *Kompendie i faget UNIK4540, UIO*, November 2009.
- [19] Oddvar Hallingstad. Forelesningsnotater stokastiske systemer, Oktober 2010.

# Tillegg A

## Matlab kode

### A.1 Allanvarians.m

Program for å beregne allanvarians:

```
1 function [ tau,Avar] = allanvar( X_in, Ts,tau )
2 %ALLVAR Summary of this function goes here
3 %Beregner Allanvariansen i punktene gitt i tau,
4 %for datasettet X_in tatt
5 %opp med samplingsintervall Ts.
6 % Detailed explanation goes here
7 % c: Antallsamples i hver subgruppe
8 % H: dXc matrise med en rad for hver subgruppe
9 % A: Snittet av hver rad i H.
10 % F: A(k+1)-A(k)
11
12 a= length(X_in);
13 i=1;
14
15 %Beregner Allanvarians for alle tauverdier som gir
16 %mer enn 2 grupper
17 while i ≤length(tau) && tau(i) ≤ a*Ts/2
18     %Samples i hver gruppe
19     c=tau(i)/Ts;
20
21     %Deler settet opp i grupper pa storrelse c.
22     [H,pad] = vec2mat(X_in,c);
23
24     %Beregner snittet av subgruppene.
25     %Bruker bare hele subgrupper
26     if pad ≠ 0
27         A = mean(H(1:end-1,:),2)';
28     else
```

```

29     A = mean(H, 2)';
30     end
31
32     %Trekker snittet i en gruppe fra
33     %snittet i den forige
34     F = diff(A);
35
36     %Allanvarians for gruppestr c
37     Avar(i) = 1/(2*(length(A)-1))*(F*F');
38     i = i+1;
39 end
40 %Returnerer kunn de tau verdier det
41 %er beregnet varians for
42 tau = tau(1:i-1);
43 end

```

## A.2 gra2rad.m

Program for å regne om fra grader til radianer:

```

1 function [rad] = gra2rad( grader )
2 % Konverterer grader til radianer
3 rad = grader * pi /180;
4 end

```

## A.3 rad2gra.m

Program for å regne om fra radianer til grader:

```

1 function [ grader ] = rad2gra( rad )
2 %Konverterer radianer til grader
3 grader = rad *180 /pi;
4 end

```

## A.4 gyromodel.m

Program for å generere gyromodell:

```

1 function [ gyromodell ] = gyromodel( gyrodata)
2 %Funksjon som lager en kontinuerlig gyromodell basert pa ...
   inputs og

```



```

3  %diskretiserer denne. Modellen baserer seg pa konstant bias ...
   i tiden
4  %modellen skal brukes.
5
6  Ts = 1/gyrodata.sps;           %Samplingstid
7
8  %Kovertering av enheter
9  ARWs = gyrodata.ARW*sqrt(gyrodata.sps); %(rad/(s^0.5)) -> ...
   (deg/(Ts^0,5)/)
10 RRWs = gyrodata.RRW;
11 qbias = gyrodata.qbias;
12
13 %Kontinuerlig modell av gyrofeilen.
14 F = 0;   %System matrise
15 G = 1;   %Prosesstoymatrise
16 H = 1;   %Malematrise
17 %Variansen til stoyen i systemet
18 Q = RRWs^2; %Tilstandenenes varians
19 R = ARWs^2; %Malestoyens kovarians
20
21 %Trekker en bias for gyroen
22 bias = sqrt(qbias)*randn(1); %Gyrobias [rad/s]
23
24 %Modellen diskretiseres:
25 Fi = 1 + F*Ts;
26 Ga = 1;
27 Qd = Q*Ts;
28 Rd = R;
29
30 gyromodell.F = F;
31 gyromodell.G = G;
32 gyromodell.H = H;
33 gyromodell.Q = Q;
34 gyromodell.R = R;
35 gyromodell.Fi = Fi;
36 gyromodell.Ga = Ga;
37 gyromodell.Qd = Qd;
38 gyromodell.Rd = Rd;
39 gyromodell.sps = gyrodata.sps;
40 gyromodell.bias = bias;
41 end

```

## A.5 gyrosim.m

Program for simulere gyroen:

```

1 function [z, stoy] = gyrosim(virkeligvf, gyromodell, time)
2 %Simulerer en gyromaling med lengde time Basert pa
3 %virkelig fart og modell av gyroen.
4 %virkeligvf: Sann Vinkelfart, Farten gyroen skal male
5 %Gyromodell: Struct som kommer som retur av funksjonen ...
   gyromodell
6 %time:      Simuleringstid
7
8 %Henter inn data om gyromodellen
9 Fi = gyromodell.Fi;
10 Ga = gyromodell.Ga;
11 Qd = gyromodell.Qd;
12 Rd = gyromodell.Rd;
13 H = gyromodell.H;
14 bias = gyromodell.bias;
15
16 %-----Starter simuleringen-----
17 antsamp = time*gyromodell.sps;    %Antall samples som skal ...
   simuleres
18
19 %Lager stoy
20 v = sqrt(Qd)*randn(1, antsamp);
21 vg = sqrt(Rd)*randn(1, antsamp);
22
23 %Initialverdier
24 X = zeros(1, antsamp);
25 z = zeros(1, antsamp);
26 X(1) = 0;
27
28 %Simulerer gyroen
29 for k = 1:antsamp
30     X(k+1) = Fi * X(k) + Ga * v(k);
31     z(:, k) = virkeligvf + bias + H * X(k) + vg(k);
32 end
33 %Stoyen pluss virkelig maling
34 stoy = [vg(:, 1:k); X(1:k)];
35 end

```

## A.6 akssim.m

Program for å simulere akselerometeret:

```

1 function [aksmaling] = akssim(aksels, akspar, simtime)
2 %Genererer 3-akset akselerometermaling med
3 %aksels som virkelig akselasjon
4 %aksels: Virkelig akselasjon pa de tre akselerometerne

```

```

5 %akspar: Struct som beskriver akselerometeret
6 %simtime: Simuleringstid
7 Ts = 1/akspar.sps;           %Samplingstid
8
9 antsamp = simtime*akspar.sps;
10
11 %Lager stoy og bias
12 bias = sqrt(akspar.qbias)*randn(3,1);
13 vg    = akspar.noise*sqrt(akspar.sps)*randn(1,antsamp);
14
15 %Initialverdier
16 aksmaling    = zeros(3,antsamp);
17
18 %Simulerer akselrometeret
19 for k = 1:antsamp
20     aksmaling(:,k)    = aksels + vg(k) + bias;
21 end
22
23 end

```

## A.7 mcompass.m

Program for gjøre montecarlo simulering av kompass:

```

1 function [ psiest,psiestna,phiest,thetaest ] = mcompass( ...
    gyropar,posgra,pos,akspar,simtime,mcrun)
2 %Funksjon som gjør et antall ,mcrun, simmuleringer av ...
    gyrokompasset.
3 % Vinkleestimatene psi theta og phi blir
4 %lagret for hver kjoring og kan bruker i montecarlo ...
    simmuleringer
5 %gyropar:  Parametere om gyroen
6 %posgra:   Informasjon om posisjon i grader
7 %pos:      Informasjon om posisjon i SI enhetre
8 %akspar:   Parametre om akselrometeret
9 %simtime:  Simuleringstid i hver posisjon
10 %mcrun:   Mantall kjoringer
11
12 %-----Transformasjonsmatriser-----
13 R_bm_a = @(e3,e2,e1) ([cos(e3)*cos(e2) ...
    cos(e3)*sin(e2)*sin(e1)-...
14     sin(e3)*cos(e1) sin(e3)*sin(e1)+cos(e3)*sin(e2)*cos(e1);
15     sin(e3)*cos(e2) cos(e3)*cos(e1)+...
16     sin(e3)*sin(e2)*sin(e1) ...
    sin(e3)*sin(e2)*cos(e1)-cos(e3)*sin(e1);
17     -sin(e2) cos(e2)*sin(e1) cos(e2)*cos(e1)];

```

```

18 R_b_r = @(psi,theta,phi) ([cos(psi)*cos(theta) ...
    cos(psi)*sin(theta)*sin(phi)-...
19     sin(psi)*cos(phi) ...
        sin(psi)*sin(phi)+cos(psi)*sin(theta)*cos(phi);
20     sin(psi)*cos(theta) cos(psi)*cos(phi)+...
21     sin(psi)*sin(theta)*sin(phi) ...
        sin(psi)*sin(theta)*cos(phi)-...
22     cos(psi)*sin(phi);
23     -sin(theta) cos(theta)*sin(phi) cos(theta)*cos(phi)]);
24 R_b1_bm = eye(3);
25 R_b2_bm = [-1 0 0; 0 1 0; 0 0 -1];
26
27
28 for j = 1:mcrun
29     %Lager gyro modell
30     gyromodell = gyromodel( gyropar);
31
32     %Genererer sanne malinger i posisjon 1 og 2
33     aks_r = [0;0;posgra.g];
34     %Posisjon 1
35     omegas1 = pos.Omega *(cos(pos.L)*cos(pos.psi)*...
36         cos(pos.theta)+sin(pos.L)*sin(pos.theta));
37     aksels1 = ...
38         R_bm_a(akspar.epsilon3,akspar.epsilon2,akspar.epsilon1)...
39         * R_b1_bm * R_b_r(pos.psi,pos.theta,pos.phi) '*aks_r;
40
41     %Posisjon 2
42     omegas2 = -pos.Omega ...
43         *(cos(pos.L)*cos(pos.psi)*cos(pos.theta)+...
44         sin(pos.L)*sin(pos.theta));
45     aksels2 = ...
46         R_bm_a(akspar.epsilon3,akspar.epsilon2,akspar.epsilon1) ...
47         * R_b2_bm * R_b_r(pos.psi,pos.theta,pos.phi) '*aks_r;
48
49     %Simulerer Gyro og akselerasjonsmaling i posisjon 1
50     [Z1.gyro,-] = gyrosim(omegas1, gyromodell, simtime);
51     [Z1.aks] = akssim(aksels1, akspar, simtime);
52
53     %Simulerer Gyro og akselerasjonsmaling i posisjon 2
54     [Z2.gyro,-] = gyrosim(omegas2, gyromodell, simtime);
55     [Z2.aks] = akssim(aksels2, akspar, simtime);
56
57     %Midler over malingene
58     Z1.gyromean = mean(Z1.gyro);
59     Z2.gyromean = mean(Z2.gyro);
60
61     Z1.aksmean = mean(Z1.aks,2);
62     Z2.aksmean = mean(Z2.aks,2);

```

```

60
61 %Beregner Akselrometer malingene i b rammen
62 Z1.aksmean_bm = R_b1_bm * ...
        R_bm_a(akspar.epsilon3,akspar.epsilon2,...
63         akspar.epsilon1)' * Z1.aksmean;
64 Z2.aksmean_bm = R_b2_bm ...
        *R_bm_a(akspar.epsilon3,akspar.epsilon2,...
65         akspar.epsilon1)' * Z2.aksmean;
66
67 %Estimerer theta til kompasset
68 philest = atan(Z1.aksmean_bm(2)/Z1.aksmean_bm(3));
69 phi2est = atan(Z2.aksmean_bm(2)/Z2.aksmean_bm(3));
70 phiest(j) = 0.5*(phi2est + philest);
71
72 %Estimerer phi til kompasset
73 thetalest = ...
        atan(-Z1.aksmean_bm(1)/sqrt(Z1.aksmean_bm(2)^2+...
74         Z1.aksmean_bm(3)^2));
75 theta2est = ...
        atan(-Z2.aksmean_bm(1)/sqrt(Z2.aksmean_bm(2)^2+...
76         Z2.aksmean_bm(3)^2));
77 thetaest(j) = 0.5*(theta2est + thetalest);
78
79 %Estimerer psi til kompasset basert pa stillingen
80 psiest(j) = acos((Z1.gyromean - Z2.gyromean -...
81         2*sin(thetaest(j))*sin(pos.L)*pos.Omega)/...
82         (2*cos(thetaest(j))*cos(pos.L)*pos.Omega));
83 psiestna(j) = acos((Z1.gyromean - Z2.gyromean)/...
84         (2*cos(pos.L)*pos.Omega));
85 end
86 end

```

## A.8 init.m

Program for initialisere systemet:

```

1 #####Innleging av ...
        simulatordata#####
2 %-----GYRODATA-----
3 gyropargra.sps      = 30;          %Gyroens samples per sekund [Hz]
4 gyropargra.binst    = 0.5;          %Bias instabillity (deg/h)
5 gyropargra.ARW      = 0.15;        %Angular Random Walk ...
        (Deg/ (h) ^0.5)
6 %gyropargra.RRW     = 8;           %Rate Random Walk ...
        (Deg/h/ (h) ^0.5)

```

```

7 gyropargra.qbias = 0.0;           %Variansen til statistisk bias ...
   pa gyroen (Deg/s)^2
8
9 %Regner rate random Walk
10 gyropargra.RRW = ...
   sqrt(3)*gyropargra.binst^2/(2*gyropargra.ARW);
11
12 %-----AKSELEROMETERDATA-----
13 akspargra.sps = 30;           %Akselrometerets samples per ...
   sekund [Hz]
14 akspargra.noise = 50e-6;       %Varians til ...
   Akselrometerets malestoy [g/(h)^0.5]
15 akspargra.biasacc = 3e-3;      %Akselrometerets ...
   nulpunktfeil standardavik [g]
16
17 akspargra.epsilon1 = 0.0;      %Montasjevinkler for ...
   Akselrometer[gra]
18 akspargra.epsilon2 = 0.0;      %Montasjevinkler for ...
   Akselrometer[gra]
19 akspargra.epsilon3 = 0.0;      %Montasjevinkler for ...
   Akselrometer[gra]
20
21 %-----Lokasjon og stilling-----
22 posgra.Omega = 15.04;          %Jordens rotasjonshastighet ...
   [deg/h]
23 posgra.g = 9.81;              %Gravitasjonskonstant [m/s^2]
24 posgra.L = 59.97;             %Breddegrad [deg]
25
26 posgra.theta = 0;             %Euler rotasjonsvinkel om Y ...
   aksen [deg]
27 posgra.phi = 0;               %Euler rotasjonsvinkel om X ...
   aksen [deg]
28 posgra.psi = 80;              %Euler rotasjonsvinkel om Z ...
   aksen [deg]
29
30 #####
31
32 %-----Omregning av data til ...
   SI-enheter-----
33 [gyropar,akspar,pos ] = converterdata( ...
   gyropargra,akspargra,posgra );

```

## A.9 mctid.m

Program for gjøre montecarlo simulering midlingstid:

```

1  clc;
2  clear;
3  %Program som simulerer gyromalinger 10 ganger for hver 10 ...
   grad. og plotter
4  %disse sammen med den teoretisk riktige farten
5  %Legger inn data og Gyro og geografisk plassering
6  run('init');
7  simtime = 1000;
8
9  R_bm_a = @(e3,e2,e1) (...
10     [cos(e3)*cos(e2)  cos(e3)*sin(e2)*sin(e1)-...
11     sin(e3)*cos(e1)  sin(e3)*sin(e1)+cos(e3)*sin(e2)*cos(e1);
12     sin(e3)*cos(e2)  cos(e3)*cos(e1)+...
13     sin(e3)*sin(e2)*sin(e1)  ...
       sin(e3)*sin(e2)*cos(e1)-cos(e3)*sin(e1);
14     -sin(e2)  cos(e2)*sin(e1)  cos(e2)*cos(e1)]);
15  R_b_r = @(psi,theta,phi) (...
16     [cos(psi)*cos(theta)  cos(psi)*sin(theta)*sin(phi)-...
17     sin(psi)*cos(phi)  ...
       sin(psi)*sin(phi)+cos(psi)*sin(theta)*cos(phi);
18     sin(psi)*cos(theta)  cos(psi)*cos(phi)+...
19     sin(psi)*sin(theta)*sin(phi)  ...
       sin(psi)*sin(theta)*cos(phi)-cos(psi)*sin(phi);
20     -sin(theta)  cos(theta)*sin(phi)  ...
       cos(theta)*cos(phi)]);
21  R_b1_bm = eye(3);
22  R_b2_bm = [-1 0 0; 0 1 0; 0 0 -1];
23
24
25  for j = 1:1000
26  %Lager gyro modell
27     gyromodell = gyromodel( gyropar);
28  %Genererer sanne malinger i posisjon 1 og 2
29  %Posisjon 1
30     omegas1 = pos.Omega ...
       *(cos(pos.L)*cos(pos.psi)*cos(pos.theta)+...
31     sin(pos.L)*sin(pos.theta));
32  %Posisjon 2
33     omegas2 = -pos.Omega ...
       *(cos(pos.L)*cos(pos.psi)*cos(pos.theta)+...
34     sin(pos.L)*sin(pos.theta));
35  %Simulerer Gyro og akselerasjonsmaling i posisjon 1
36     [Z1.gyro(j,:),-] = gyrosim(omegas1, gyromodell, simtime);
37  %Simulerer Gyro og akselerasjonsmaling i posisjon 2
38     [Z2.gyro(j,:),-] = gyrosim(omegas2, gyromodell, simtime);
39  end
40     %Midler over malingene hver 20 sekund
41  for i = 20:20:simtime
42     Z1.gyromean(j,:) = mean(Z1.gyro(:,1:gyropargra.sps*i),2);

```

```

43     Z2.gyromean(j,:) = mean(Z2.gyro(:,1:gyropargra.sps*i),2);
44     j=j+1;
45 end
46     meantimevector = [20:20:simtime];
47 %Beregner psi estimat for MC runn hvert 20 sekund
48     fun = @(omega2,omega1)acos((omega1 - ...
49         omega2)/(2*cos(pos.L)*pos.Omega));
49     psiest = arrayfun(fun,Z2.gyromean,Z1.gyromean);
50 %Beregner Snitt og varians
51     simpsimean = mean(psiest,2);
52     simpsivar = var(psiest,0,2);
53     % regner om til grader og bruker dette i beregningene
54     simpsistd = rad2gra(simpsivar'.^0.5);
55 %-----Plotter data-----
56 %-----MC simulering av optimal k-----
57 figure(1)
58     clf
59     hold on
60     MCrungroup = hggroup; %Grupperer alle Montecarlo run
61     STDgroup = hggroup; %Grupperer alle Montecarlo run
62     STD2group = hggroup; %Grupperer alle Montecarlo run
63
64 %plotter montecarlo run
65 for i = 1:length(meantimevector)
66     plot(meantimevector(i)*ones(length(psiest(i,1:100:end)),1),...
67         rad2gra(psiest(i,1:100:end)), 'x', 'MarkerSize',7,...
68         'Parent',MCrungroup);
69 end
70 set(get(get(MCrungroup, 'Annotation'), 'LegendInformation'),...
71     'IconDisplayStyle', 'on');
72
73 plot(meantimevector, rad2gra(simpsimean), '--r', 'LineWidth',2,...
74     'MarkerFaceColor', 'r', 'MarkerSize',7);
75 plot(meantimevector, rad2gra(simpsimean + ...
76     sqrt(simpsivar)), '--ok',...
77     'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize',3,...
78     'Parent', STDgroup);
79 plot(meantimevector, rad2gra(simpsimean - ...
80     sqrt(simpsivar)), '--ok',...
81     'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize',3,...
82     'Parent', STDgroup);
83 set(get(get(STDgroup, 'Annotation'), 'LegendInformation'),...
84     'IconDisplayStyle', 'on');
85 set(get(get(STD2group, 'Annotation'), 'LegendInformation'),...
86     'IconDisplayStyle', 'on');
87
88 legend('10 MC kjoringer for hver midlingstid', 'MC ...
89     Standardavik',...
90     'Forventet \psi', 'Location', 'NorthEast');

```



```

88 xlabel('Sann nordvinkel [Deg]');
89 ylabel('Malt nord vinkel [Deg]');
90 title('Simulert kompass med \psi som funksjon av ...
      breddegrad ',...
91       'FontSize',14);
92 grid on;
93 xlabel('Midlingstid', 'FontSize',14);
94 ylabel('Estimert \psi', 'FontSize',14);
95 grid on;
96 v=axis;
97 axis([v(1) v(2) 70 90]);

```

## A.10 minpsi.m

Program for å løse ulikheten for minste østvest avik:

```

1  %Program for a beregne minimum ostvest avik
2  clear
3  clc
4  run('init');
5
6  simtime = 300;
7
8  %Legger inn de diskrete varians matrisene for male og prosesstoy
9  gyromodell = gyromodel(gyropar);
10
11 Qd = gyromodell.Q(1,1)/gyromodell.sps;
12 Rd = gyromodell.R;
13
14 Omega = gra2rad(15.04/3600);           %Rotasjon om hovedakse ...
      i gra/s
15 L = gra2rad(59.97);                   %Breddegrad omtrent til ...
      kjeller
16
17 varians = ((simtime*gyromodell.sps)*Rd+...
18           sum([1:(simtime*gyromodell.sps)].^2*Qd))/((simtime*gyromodell.sps)^2);
19 psival = gra2rad(70:0.1:100);
20
21 for i = 1:length(psival)
22     yval(i) = Omega*cos(L)*cos(psival(i));
23 end
24
25 minstd = rad2gra(yval - sqrt(varians));
26 maxstd = rad2gra(yval + sqrt(varians));
27
28 %Finner vinkel som gir 0

```

```

29 [maxstd0,maxstd0index] = min(abs(minstd));
30 %-----Plotter dataene-----
31 figure(1)
32 clf;
33 hold on;
34 plot(rad2gra(psival),rad2gra(yval),'k');
35 plot(rad2gra(psival),minstd,'-r');
36 plot(rad2gra(psival),maxstd,'-b');
37 line([rad2gra(psival(maxstd0index)),90],[0,0],'Marker','x',...
38      'MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',5,...
39      'color','k','linewidth',2);
40 plot(rad2gra(psival(maxstd0index)),maxstd0,'o','MarkerEdgeColor','k',...
41      'MarkerFaceColor','g','MarkerSize',6);
42 grid on;
43 xlabel('\psi [deg'],'FontSize',12);
44 ylabel('\omega [deg/s'],'FontSize',12);
45 title('Forventning og standardavik for \omega som ...
46      funksjon av \psi. ',...
47      'FontSize',12);
48 legend('E[\omega]','E[\omega] - \sigma','E[\omega] + ...
49      \sigma','Delta',...
50      'Losning av ulikhet');
51 %Setter pa tekst
52 s1=sprintf('Minimum ostvest avik, Delta: %5.1f deg',...
53      90-rad2gra(psival(maxstd0index))); %
54 text(73,-0.00044,s1,'FontName','Arial','Color','k','FontSize',11,...
55      'FontWeight','Bold');
56 text(84,-0.0001,'Delta','FontName','Arial','Color','k','FontSize',12,...
57      'FontWeight','Bold');

```

## A.11 simintvinkel.m

Program for å finne beste initielle vinkel:

```

1 clc;
2 clear;
3 %Program som simulerer gyromalinger 10 ganger for hver 10 ...
4   grad. og plotter
5 %disse sammen med den teoretisk riktige farten
6 %Legger inn data og Gyro og geografisk plassering
7 run('init');
8 simtime = 300;
9
10 %-----Kjoring av initiell vinkel
11 %Data som skal undersokes
12 psiarray = [50,60,70,80,90,100,110,120,130];

```

```

12 for i = 1:length(psiarray)
13     pos.psi = gra2rad(psiarray(i));
14     [ allpsiest(i,:),~ ,~ ] = mcompass( gyropar,posgra,pos,...
15         akspar,simtime,10000);
16
17 end
18
19     simpsimean = mean(allpsiest,2);
20     simpsivar = var(allpsiest,0,2);
21 % regner om til grader og bruker dette i beregningene
22     simpsistd = rad2gra(simpsivar'.^0.5);
23
24     %Beregner en kurve som passer til punktene.
25     psistdpoly = polyfit(psiarray,simpsistd,4);
26     polyx      = [50:0.2:130];
27     polyy      = polyval(psistdpoly,polyx);
28
29     %Finner minimumspunktet
30     [minstd,minindex]= min(polyy)
31     vinkel = [50:0.1:130];
32
33 %-----Plotter data
34 %-----MC simulering av initiell vinkel-----
35     figure(1)
36     clf
37     subplot(3,1,1:2);
38     plot(vinkel,vinkel,'linewidth',2,'color','r');
39     hold on
40
41     MCrungroup = hggroup; %Gruperer alle Montecarlo run
42     for i = 1:length(psiarray)
43         plot(psiarray(i)*ones(length(allpsiest(i,1:1000:end)),1),...
44             rad2gra(allpsiest(i,1:1000:end)),'o','MarkerSize',7,...
45             'Parent',MCrungroup);
46     end
47     set(get(MCrungroup,'Annotation'),'LegendInformation',...
48         'IconDisplayStyle','on');
49
50     plot(psiarray,rad2gra(simpsimean),'o','MarkerEdgeColor','k',...
51         'MarkerFaceColor','r','MarkerSize',7);
52     plot(psiarray,rad2gra(simpsimean + ...
53         sqrt(simpsivar)),'-ok',...
54         'MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',5);
55     plot(psiarray,rad2gra(simpsimean - ...
56         sqrt(simpsivar)),'-ok',...
57         'MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',5);
58
59     legend('Sann nordvinkel','10 MC kjoringer i hver ...
60         posisjon',...

```

```

58     'Beregnet forventet nordvinkel', 'MC ...
        Standardavik', 'Location', ...
59     'NorthWest');
60 xlabel('Sann nordvinkel [Deg]');
61 ylabel('Malt nord vinkel [Deg]');
62 title('Simulert kompass i forskjellige posisjoner ved ...
        10000 kjoringer i hver posisjon ', 'FontSize', 12);
63
64     s3=sprintf('Midlingstid: %5.0f s\n\rBreddegrad: %5.0f ...
        deg\n\rARW: %5.3f deg/sqrt(h)\n\rBias instability: ...
        0.5 deg/s', ...
65         simtime, rad2gra(pos.L), gyropargra.ARW); %
66     text(111, 40, s3, 'FontName', 'Arial', 'Color', 'k', 'FontSize', 11);
67
68     grid on;
69     subplot(3,1,3);
70     plot(polyx, polyy, 'linewidth', 1, 'color', 'k');
71     hold on
72     plot(psiarray, simpstsd, 'o', 'MarkerEdgeColor', 'k', ...
73         'MarkerFaceColor', 'r', 'MarkerSize', 5);
74     plot(polyx(minindex), minstd, 'x', 'MarkerEdgeColor', 'k', ...
75         'MarkerFaceColor', 'k', 'MarkerSize', 10);
76
77     xlabel('Sann nordvinkel [Deg]');
78     ylabel('Standardavik [deg]');
79     grid on;
80
81     %Setter pa tekst
82     s1=sprintf('Minste standardavik : %5.5f deg', minstd); %
83     s2=sprintf('Optimal initiell vinkel: %3.1f ...
        deg', polyx(minindex)); %
84
85     v=axis;
86     text(60, 10.2, s1, 'FontName', 'Arial', 'Color', 'k', 'FontSize', 11);
87     text(60, 9.7, s2, 'FontName', 'Arial', 'Color', 'k', 'FontSize', 11);

```

## A.12 simbreddegrad.m

program for å gjøre MonteCarlo simulering av breddegrad:

```

1  clc;
2  clear;
3
4  %Program som simulerer gyromalinger 10 ganger for hver ...
        breddegradene i
5  %Larray og plotter disse sammen med den teoretisk riktige farten

```

```

6 %Legger inn data og Gyro og geografisk plassering
7 run('init');
8 simtime = 300;
9 %-----Kjoring av initiell vinkel
10 %Data som skal undersokes
11 Larray = [0, 10, 20, 30, 40, 50, 60, 65, 70,
12          72, 76, 78, 80, 82, 84, 86, 88, 89, 89.5];
13 %phiarray = ...
14          [-16,-14,-12,-10,-8,-6,-4,-2,0,2,4,6,8,10,12,14,16];
14 for i = 1:length(Larray)
15     pos.L = gra2rad(Larray(i));
16     [allpsiest(i,:),allpsiestna(i,:),thetaest(i,:),~) = ...
17       mcompass(gyropar,...
18       posgra,pos,akspar,simtime,10000);
18 end
19 %Beregner Forventning og varians
20     simpsimean = mean(allpsiest,2);
21     simpsivar = var(allpsiest,0,2);
22
23 %-----Plotter data
24 %-----MC simulering av initiell vinkel-----
25     figure(1)
26     clf
27     hold on
28
29     MCrungroup = hggroup; %Gruperer alle Montecarlo run
30     STDgroup = hggroup; %Gruperer alle Montecarlo run
31     STD2group = hggroup; %Gruperer alle Montecarlo run
32
33     for i = 1:length(Larray)
34         plot(Larray(i)*ones(length(allpsiest(i,1:1000:end)),1),...
35              rad2gra(allpsiest(i,1:1000:end)),'x','MarkerSize',7,...
36              'Parent',MCrungroup);
37     end
38     set(get(get(MCrungroup,'Annotation'),'LegendInformation'),...
39         'IconDisplayStyle','on');
40
41     plot(Larray,rad2gra(simpsimean),'s','MarkerEdgeColor','k',...
42          'MarkerFaceColor','r','MarkerSize',7);
43     plot(Larray,rad2gra(simpsimean + sqrt(simpsivar)),'-ok',...
44          'MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',3,...
45          'Parent',STDgroup);
46     plot(Larray,rad2gra(simpsimean - sqrt(simpsivar)),'-ok',...
47          'MarkerEdgeColor','k','MarkerFaceColor','k','MarkerSize',3,...
48          'Parent',STDgroup);
49     set(get(get(STDgroup,'Annotation'),'LegendInformation'),...
50         'IconDisplayStyle','on');
51     set(get(get(STD2group,'Annotation'),'LegendInformation'),...
52         'IconDisplayStyle','on');

```

```

53     legend('10 MC kjoringer i hver posisjon m/ akselerometer',...
54           'MC Standardavik m/ akselerometer','Forventet \psi ...
           m/ Akselerometer',...
55           'Location','NorthWest');
56     xlabel('Sann nordvinkel [Deg]');
57     ylabel('Malt nord vinkel [Deg]');
58     title('Simulert kompass med \psi som funksjon av ...
           breddegrad ',...
59           'FontSize',12);
60     grid on;
61     xlabel('Breddegrad L');
62     ylabel('Estimert \psi');
63     grid on;

```

### A.13 simskråstilling.m

Program for å beregne virkningen av skråstilling:

```

1  clc;
2  clear;
3  %Program som simulerer gyromalinger 10000 ganger for hver 10 ...
   grad.
4  %og plotter disse sammen med den teoretisk riktige farten
5  %Legger inn data og Gyro og geografisk plassering
6  run('init');
7  simtime = 300;
8
9  %-----Kjoring av initiell vinkel
10 %Data som skal undersokes
11 thetaarray = ...
    [-16,-14,-12,-10,-8,-6,-4,-2,0,2,4,6,8,10,12,14,16];
12 for i = 1:length(thetaarray)
13     pos.theta = gra2rad(thetaarray(i));
14     [allpsiest(i,:),allpsiestna(i,:),thetaest(i,:),-1] = ...
        mcompass(gyropar,posgra,pos,akspar,simtime,10);
15 end
16 %Estimerer Montecarlo Snitt ig standard avik
17 simpstimean = mean(allpsiest,2);
18 simpstimeanna = mean(allpsiestna,2);
19 simpstivar = var(allpsiest,0,2);
20 simpstinavar = var(allpsiestna,0,2);
21
22 %-----Plotter data
23 %-----MC simulering av initiell vinkel-----
24 figure(1)
25 clf

```

```

26 hold on
27
28 MCrungroup = hggroup; %Gruperer alle Montecarlo run
29 STDgroup = hggroup; %Gruperer alle Montecarlo run
30 STD2group = hggroup; %Gruperer alle Montecarlo run
31
32 for i = 1:length(thetaarray)
33     plot(thetaarray(i)*ones(length(allpsiest(i,1:100:end)),1),...
34          rad2gra(allpsiest(i,1:100:end)), 'x', 'MarkerSize',7, 'Parent',MCrungroup);
35 end
36 set(get(get(MCrungroup, 'Annotation'), 'LegendInformation'), 'IconDisplayStyle', 'on')
37
38 plot(thetaarray, rad2gra(simpsimean), ...
39      's', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'MarkerSize',7);
40 plot(thetaarray, rad2gra(simpsimeanna), ...
41      'o', 'MarkerEdgeColor', 'g', 'MarkerFaceColor', 'g', 'MarkerSize',7);
42 plot(thetaarray, rad2gra(simpsimean + sqrt(simpsivar)), ...
43      '—ok', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize',3, ...
44      'Parent', STDgroup);
45 plot(thetaarray, rad2gra(simpsimean - sqrt(simpsivar)), ...
46      '—ok', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'k', 'MarkerSize',3, ...
47      'Parent', STDgroup);
48 plot(thetaarray, rad2gra(simpsimeanna + sqrt(simpsinavar)), ...
49      '—sg', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize',3, ...
50      'Parent', STD2group);
51 plot(thetaarray, rad2gra(simpsimeanna - sqrt(simpsinavar)), ...
52      '—sg', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize',3, ...
53      'Parent', STD2group);
54 set(get(get(STDgroup, 'Annotation'), 'LegendInformation'), ...
55      'IconDisplayStyle', 'on');
56 set(get(get(STD2group, 'Annotation'), 'LegendInformation'), ...
57      'IconDisplayStyle', 'on');
58
59 legend('10 MC kjoringer i hver posisjon m/ akselerometer', ...
60       'MC Standardavik m/ akselerometer', 'MC Standardavik uten ...
61       akselerometer', ...
62       'Forventet \psi m/ Akselerometer', 'Forventet \psi uten ...
63       Akselerometer', ...
64       'Location', 'NorthEast');
65 xlabel('Sann nordvinkel [Deg]');
66 ylabel('Malt nord vinkel [Deg]');
67 title('Simulert kompass med forskjellig skjevstilling ved ...
68       1000 kjoringer i hver posisjon ', ...
69       'FontSize',12);
70 grid on;
71 xlabel('Skjevstilling i \theta');
72 ylabel('Estimert \psi');
73 grid on;

```

## A.14 feilbudsjett.m

Program for å beregne feilbudsjett:

```

1  clc;
2  clear;
3
4  %Program som beregner feilbudsjett for nordsokeren
5  %Legger inn data om Gyro og geografisk plassering
6  run('init');
7  simtime = 300;
8  %-----Kjorer først 10000 ganger med alle ...
   feilene inne
9  [allpsiest(1,:),~,thetaest(1,:),~] = mcompass(...
10     gyropar,posgra,pos,akspar,simtime,10000);
11 %-----Kjorer 10000 ganger med bare ARW ...
   feilen-----
12 %-----GYRODATA-----
13 gyropargra.ARW      = 0.15;      %Angular Random Walk ...
   (Deg/(h)^0.5)
14 gyropargra.RRW      = 0;%8;      %Rate Random Walk ...
   (Deg/h/(h)^0.5)
15
16 %-----AKSELEROMETERDATA-----
17 akspargra.noise     = 0; %Varians til Akselerometerets ...
   malestoy [g/(h)^0.5]
18 akspargra.biasacc   = 0; %Akselerometerets nulpunktfeil ...
   standardavik [g]
19
20 [gyropar,akspar,pos ] = converterdata( ...
21     gyropargra,akspargra,posgra );
22 [allpsiest(2,:),~,thetaest(2,:),~] = mcompass(...
23     gyropar,posgra,pos,akspar,simtime,10000);
24
25 %-----Kjorer 10000 ganger med bare RRW ...
   feilen-----
26 %-----GYRODATA-----
27 gyropargra.ARW      = 0;%Angular Random Walk (Deg/(h)^0.5)
28 gyropargra.binst     = 0.5;%Bias instabilitet (deg/h)
29 gyropargra.RRW      = sqrt(3)*gyropargra.binst^2/...
30     (2*gyropargra.ARW);      %Rate Random Walk ...
   (Deg/h/(h)^0.5)
31
32 %-----AKSELEROMETERDATA-----
33 akspargra.noise     = 0;%Varians til Akselerometerets ...
   malestoy [g/(h)^0.5]
34 akspargra.biasacc   = 0;%Akselerometerets nulpunktfeil ...
   standardavik [g]

```



```

35
36 [gyropar,akspar,pos ] = converterdata( ...
    gyropargra,akspargra,posgra );
37 [allpsiest(3,:),~,thetaest(3,:),~] = ...
    mcompass(gyropar,posgra,pos,akspar,simtime,10000);
38
39 %-----Kjorer 10000 ganger med bare RRW ...
    feilen-----
40 %-----GYRODATA-----
41 gyropargra.ARW      = 0;%Angular Random Walk (Deg/(h)^0.5)
42 gyropargra.RRW      = 0;%Rate Random Walk (Deg/h/(h)^0.5)
43
44 %-----AKSELEROMETERDATA-----
45 akspargra.noise      = 50e-6;%Varians til Akselerometerets ...
    malestoy [g/(h)^0.5]
46 akspargra.biasacc   = 0;%Akselerometerets nulpunktfeil ...
    standardavik [g]
47
48 [gyropar,akspar,pos ] = converterdata( ...
    gyropargra,akspargra,posgra );
49 [allpsiest(4,:),~,thetaest(4,:),~] = mcompass(gyropar,...
    posgra,pos,akspar,simtime,10000);
50
51
52 %-----Kjorer 10000 ganger med bare akselerometer ...
    nullpunktsfeilen-----
53 %-----GYRODATA-----
54 gyropargra.ARW      = 0;%Angular Random Walk (Deg/(h)^0.5)
55 gyropargra.RRW      = 0;%Rate Random Walk (Deg/h/(h)^0.5)
56
57 %-----AKSELEROMETERDATA-----
58 akspargra.noise      = 0;%Varians til Akselerometerets ...
    malestoy [g/(h)^0.5]
59 akspargra.biasacc   = 3e-3;%Akselerometerets nulpunktfeil ...
    standardavik [g]
60
61 [gyropar,akspar,pos ] = converterdata( ...
    gyropargra,akspargra,posgra );
62 [allpsiest(5,:),~,thetaest(5,:),~] = mcompass(gyropar,...
    posgra,pos,akspar,simtime,10000);
63
64
65
66 simpsimean          = mean(allpsiest,2);
67 simpsivar           = var(allpsiest,0,2);
68 simpsistd           = rad2gra(simpsivar'.^0.5);

```



## Tillegg B

### Sammendrag av datablader for STIM202 og STIM210

# ULTRA-HIGH PERFORMANCE GYRO MODULE



STIM202

PRODUCT BRIEF

- Small size, low weight and low cost
- ITAR free
- Insensitive to magnetic fields
- 0.5°/h bias instability
- 0.2°/√h angular random walk
- ±400°/s angular rate input range
- 1500g shock capability

STIM202 is a multi-axis gyro module with up to 3 axes of highly accurate MEMS gyros. Each axis is factory calibrated for bias and sensitivity, and compensated for temperature effects over the full temperature operating range.

For many applications STIM202 replaces FOG's and improve system solutions directly with respect to robustness, reliability, size/ weight, power and cost. STIM202 industrialization is realized by combining the well proven Sensoror ButterflyGyro™ technology with full digital operation.

## Input range, orthogonality and output formats

The STIM202 full-scale angular rate input range is 400°/s and the output is capped at ±480°/s. 3-axis modules feature electronic axis alignment, improving orthogonality between axis (down to 1mrad). Selectable output formats are angular rate, increment angle, average angular rate and integrated angle, at sampling rates up to 1000 samples per second.

## Reliability and robustness

Perfect tuning of excitation and detection frequencies, as well as perfectly balanced vibrational masses, result in very low sensitivity to vibration and shock. For use in extreme environments, the STIM202 provides a vibration isolated internal assembly to avoid rectification errors.

## Power and Interface

The unit is powered by a single +5V supply and communicates via a Plug-and-Play high-level RS422 interface at bit rates up to 921600 bits/s.

## Device configurations and self diagnostics

The use of a 32-bit RISC ARM microcontroller provides flexibility in device configuration. Choices for output unit, sampling frequency, LP filter cut-off frequency, RS422 transmission bit rate, line termination ON/OFF, etc. can be done in device Service Mode. The Service Mode also provides the ability to perform single measurements on demand and access detailed diagnostics information.

## Evaluation kits



(39mm x 45mm x 20mm)

STIM202 evaluation kits for PCI and USB connectivity are available. The PCI kit is the recommended choice for thorough characterization. The PCI kit supports all STIM202 RS422 transmission bit rates (374400, 460800 and 921600 bits/s). The USB kit is the alternative solution, e.g. for smaller, portable laptop setups, providing an excellent choice for quick gyro module configurations and shorter measurement series. The USB kit supports the default STIM202 RS422 transmission bit rate (460800 bits/s).

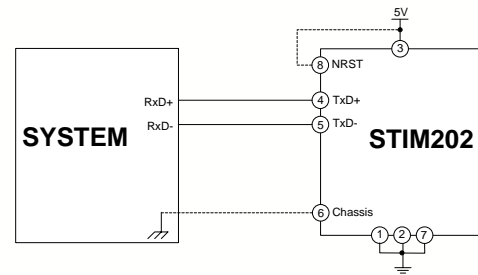
## Applications

Typical STIM202 applications are attitude heading reference systems (AHRs), platform stabilization and pointing systems (antennas, cameras, gimbals,..), high performance industrial applications, unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), automated ground vehicles (AGVs), space applications, and more. STIM202 can also open new markets, where adequate solutions previously have not yet been possible to realize.

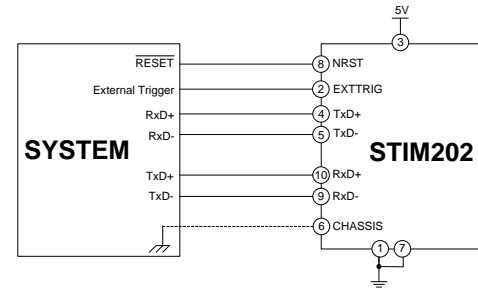
SPECIFICATIONS

Parameter	Min	Nom	Max	Unit
Weight		55		g
Input range		±400		°/s
Resolution		24		bit
Operating temperature	-40		85	°C
Storage temperature	-50		90	°C
Power supply	4.5	5.0	5.5	V
Supply current		200		mA
Start-up time		10		s
Sample Rate		1000		SPS
Dynamic overload		5000		°/s
Mechanical shock		1500		g
Bias instability (Root Allan Variance)	0.5			°/h
Angular random walk (Root Allan Variance)	0.2			°/√h
Bandwidth (-3dB)			262	Hz
Non-linearity (BSL over +/- 200 °/s)			200	ppm
Scale Factor accuracy	±0.2			%
Bias error over temperature gradients	±30			°/h rms
Linear acceleration effect			18	°/h/g
RS422 bit rate			921600	bit/s
Input resistance (termination ON)		120		Ω
Input resistance (termination OFF)				Ω
Logic levels NRST and EXT TRIGGER pins	CMOS and TTL compatible			

ELECTRICAL CONNECTIONS

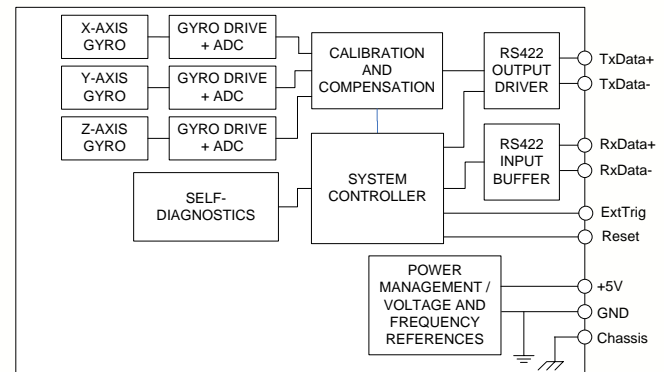


(TRANSMIT ONLY)



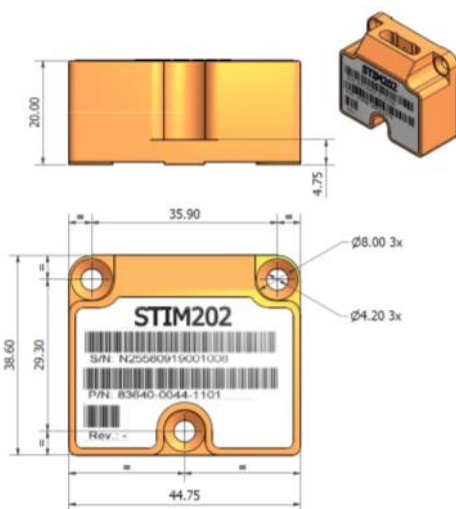
(FULL FUNCTION)

FUNCTIONAL BLOCK DIAGRAM

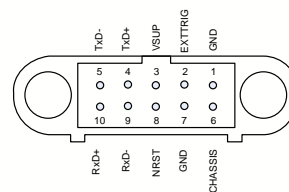


MECHANICAL DIMENSIONS

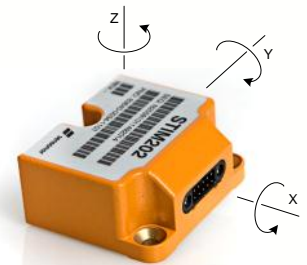
All dimensions in mm.



PIN OUT



AXIS DEFINITIONS



Information furnished by Sensoror is believed to be accurate and reliable. However, no responsibility is assumed by Sensoror for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Sensoror reserves the right to make changes without further notice to any products herein. Sensoror makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Sensoror assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. No license is granted by implication or otherwise under any patent or patent rights of Sensoror. Trademarks and registered trademarks are the property of their respective owners. Sensoror products are not intended for any application in which the failure of the Sensoror product could create a situation where personal injury or death may occur. Should Buyer purchase or use Sensoror products for any such unintended or unauthorized application, Buyer shall indemnify and hold Sensoror and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable legal fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Sensoror was negligent regarding the design or manufacture of the part.

# ULTRA-HIGH PERFORMANCE GYRO MODULE



STIM210

PRODUCT BRIEF

## FEATURES

- 10°/h bias error over temperature gradients ( $\Delta T \leq 1^\circ\text{C}/\text{min}$ )
- 0.5°/h bias instability (Root Allan Variance)
- 0.15°/√h ARW (Root Allan Variance)
- 52g module weight
- $\pm 400^\circ/\text{s}$  input range
- 2000 samples per second
- 1500g shock capability



## DESCRIPTION

STIM210 is a multi-axis gyro module with up to 3 axes of highly accurate MEMS gyros. Each axis is factory calibrated for bias and sensitivity, and compensated for temperature effects. This new design of the STIM200-series allows for improved bias error over temperature gradients (10°/h) and improved non-linearity (50ppm BSL over FS). STIM210 industrialization is realized by combining the well proven Sensoror ButterflyGyro™ technology combined with full digital operation.

### Input range, orthogonality and output formats

The STIM210 full-scale angular rate input range is 400°/s. 3-axis modules feature electronic axis alignment, improving orthogonality between axis (down to 1mrad). Selectable output formats are angular rate, increment angle, average angular rate and integrated angle, at sampling rates up to 2000 samples per second.

### Reliability and robustness

STIM210 modules have MTBF > 120k hours per axis (according to MIL-HDBK 217), which is outperforming current FOG

systems. Tuning of excitation and detection frequencies, as well as perfectly balanced vibrational masses, result in very low vibration and shock sensitivity in any direction.

### Power and interface

The unit is powered by a single +5V supply and communicates via a Plug-and-Play high-level RS422 interface at bit rates up to 1.8 Mbits/s.

### Device configuration and self diagnostics

The use of a 32-bit RISC ARM microcontroller provides flexibility in device configuration. Choices for output unit, sampling frequency, LP filter cut-off frequency, RS422 transmission bit rate, line termination ON/OFF, etc. can be done in device Service Mode. The Service Mode also provides the ability to perform single measurements on demand and access detailed diagnostics information.

### Evaluation kits

STIM210 evaluation kits for PCI and USB connectivity are available. The *PCI kit* is the recommended choice for thorough characterization. The *PCI kit* also supports all RS422 transmission bit rates

(374400, 460800, 921600 and 1843200 bits/s).

The *USB kit* provides the alternative solution, e.g. for smaller, portable laptop setups, providing an excellent choice for gyro module configuration and shorter measurement series. The *USB kit* supports the default RS422 transmission bit rate (460800 bits/s) only.

### Application areas

STIM210 applications are typically found within Industrial, Aerospace and Defence markets, for various platform stabilizations, pointing and navigation systems (e.g. antennas, cameras and gimbals), attitude heading reference systems (AHRs), inertial navigation systems (INSs) for UAVs, AUVs, AGVs, UGVs and ROVs, smart munitions, missiles, 3D mapping systems, range finders, trains, robotics, and more. For many applications STIM210 directly replaces FOGs and improves system solutions with respect to robustness, reliability, size, weight, power and cost. STIM210 can also open new markets, where adequate solutions previously have not yet been possible to realize.

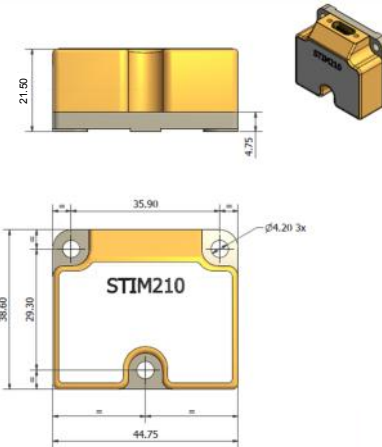


SPECIFICATIONS

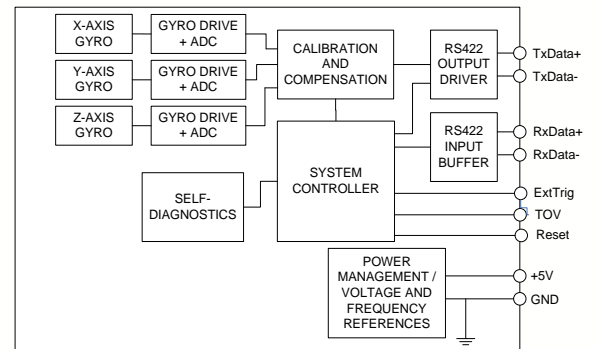
Parameter	Min	Nom	Max	Unit
Weight		52		g
Input range		±400		°/s
Resolution		24		bit
Operating temperature	-40		85	°C
Storage temperature	-50		90	°C
Power supply	4.5	5.0	5.5	V
Supply current			300	mA
Start-up time			1	s
Sampling frequency			2000	SPS
Mechanical shock			1500	g
In-run bias instability		0.5		°/h
Angular random walk		0.15		°/√h
Bias error over temperature gradients		±10		°/h rms
Bandwidth (-3dB)		262		Hz
Group delay		1.6		ms
Non-linearity (BSL over FS)		50		ppm
Scale Factor accuracy		500		ppm
RS422 bit rate			1.84	Mbit/s
Linear acceleration effect			10	°/h/g
Vibration rectification coefficient (VRC)		0.1		°/h/g rms
<b>NRST, EXT TRIGGER AND TOV PINS</b>				
Logic levels	CMOS and TTL compatible			

MECHANICAL DIMENSIONS

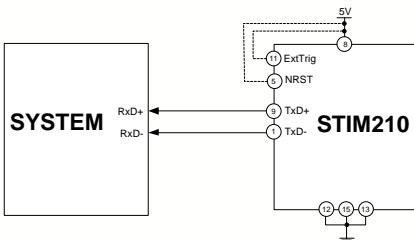
All dimensions in mm.



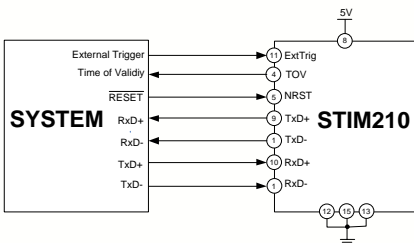
FUNCTIONAL BLOCK DIAGRAM



ELECTRICAL CONNECTIONS

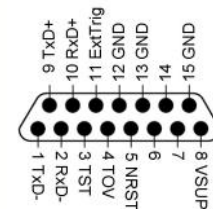


TRANSMIT ONLY



FULL FUNCTION

PIN OUT



AXIS DEFINITIONS

