

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

**MEMS-  
gyromatriser**

Masteroppgave

Tomas Sandmo

**30. mai 2011**





# Forord

Denne rapporten representerer den siste delen av det to-årige masterstudiet i Elektronikk og datateknologi ved Universitetet i Oslo. Jeg har spesialisert meg innen studieretningen kybernetikk, og har under hele studieperioden hatt studiested ved Universitetssenteret på Kjeller (UNIK).

Masteroppgaven er utført i samarbeid med Forsvarets forskningsinstitutt (FFI), og jeg vil takke forskerne Ørnulf Kandola og Anders Rødningby ved FFI for den positive innstillingen de har vist i forbindelse med oppgaven.

Til slutt vil jeg rette en stor takk til Professor Oddvar Hallingstad ved UNIK for all verdifull undervisning og veiledning i løpet av studieperioden.

Kjeller, Mai 2011

---

Tomas Sandmo



# Sammendrag

Denne masteroppgaven tar for seg bruk av gyromatriser for å forbedre nøyaktigheten til MEMS-gyroer. Det er utviklet en matematisk modell av vinkelhastighetsmålinger fra et enkelt gyroskop, og det er gjort forsøk på å få en høyere nøyaktighet ved å kombinere målinger fra flere gyroer. I gyromatrisene er støybidragene fra gyroene samlet og kombinert, og det er benyttet Kalmanfilter som filtrerer bort store deler av støyen. Flere scenarioer er simulert, blant annet med forskjellige antall gyroer, og tilfeller der det er korrelasjon mellom gyroene er sammenliknet med tilfeller der korrelasjon ikke er til stede.

Resultatene viser at man kan oppnå store forbedringer ved å kombinere gyromålinger i gyromatriser. Både antall gyroer som ligger til grunn for målingene, samt om det er korrelasjon til stede eller ikke, viser seg å være blant de viktigste faktorene som spiller inn.



# Innhold

<b>1</b>	<b>Innledning</b>	<b>1</b>
1.1	Om oppgaven . . . . .	1
1.2	Struktur . . . . .	2
<b>2</b>	<b>MEMS</b>	<b>3</b>
2.1	MEMS-gyroer . . . . .	3
2.1.1	Virkemåte . . . . .	4
2.2	Gyromatriser . . . . .	5
2.2.1	Korrelasjon mellom gyroene . . . . .	6
<b>3</b>	<b>Teoretisk bakgrunn</b>	<b>9</b>
3.1	Stokastiske variabler . . . . .	9
3.2	Stokastiske prosesser . . . . .	11
3.3	Allanvarians . . . . .	13
3.4	Kalmanfilter . . . . .	15
3.4.1	Kalmanfilterlikningene . . . . .	16
3.4.2	Symbolforklaring . . . . .	17
<b>4</b>	<b>Modeller</b>	<b>19</b>
4.1	Gyromodell . . . . .	19
4.1.1	Støykilder . . . . .	19
4.1.2	Feilmodell . . . . .	23
4.1.3	Implementering av feilmodell i MATLAB . . . . .	24
4.1.4	STIM202 . . . . .	26
4.2	Gyromatrise . . . . .	29
4.2.1	Modell av gyromatrise . . . . .	29
4.2.2	Implementering av gyromatrise i MATLAB . . . . .	31
<b>5</b>	<b>Resultater</b>	<b>33</b>
5.1	Generelt om simuleringene . . . . .	33
5.2	3 gyroer . . . . .	34

## Innhold

---

5.2.1	Tilfelle 1: Gyromatrise med 3 gyroer uten korrelasjon .	37
5.2.2	Tilfelle 2: Gyromatrise med 3 gyroer med korrelasjon .	39
5.3	6 gyroer . . . . .	39
5.3.1	Tilfelle 3: Gyromatrise med 6 gyroer uten korrelasjon .	42
5.3.2	Tilfelle 4: Gyromatrise med 6 gyroer med korrelasjon .	44
5.4	Oppsummering . . . . .	45
<b>6</b>	<b>Konklusjon</b>	<b>47</b>
<b>7</b>	<b>Videre arbeid</b>	<b>49</b>
<b>A</b>	<b>Datablad: STIM202</b>	<b>53</b>
<b>B</b>	<b>Matlab</b>	<b>57</b>
B.1	kp2dpGa.m . . . . .	57
B.2	kp2dpS.m . . . . .	57
B.3	Feilmodell3g.m . . . . .	57
B.4	Gyromatrise3.m . . . . .	62
B.5	Feilmodell6g.m . . . . .	64
B.6	Gyromatrise6 . . . . .	70
B.7	Allan.m . . . . .	73



# Figurer

2.1	Hvordan corioliskraften oppstår i et MEMS-gyroskop [17] . . . . .	5
2.2	Drift av et virtuelt gyroskop (bestående av 4 gyroer) i forhold til korrelasjonsfaktor [5] . . . . .	6
2.3	MEMS-gyroer plassert på en enkelt brikke [3] . . . . .	7
3.1	Normalfordelingen [2] . . . . .	12
3.2	Eksempel på Allanvarians beregnet for 3 gyroer [9] . . . . .	14
3.3	Blokkskjema av et diskret Kalmanfilter . . . . .	16
4.1	Effektspektraltetthet for hvit støy [2] . . . . .	20
4.2	Eksempel på hvit støy . . . . .	20
4.3	Effektspektraltetthet for markovprosess [2] . . . . .	22
4.4	Eksempel på random walk . . . . .	22
4.5	MEMS-gyroen STIM202 fra Sensoror . . . . .	26
4.6	Simulert utvikling av vinkelhastighet, $\omega^g$ , for STIM202 som ligger i ro (50 sekunder). . . . .	27
4.7	Simulert utvikling av støybidrag fra farget støy, $\mu^g$ , for STIM202 som ligger i ro (50 sekunder). . . . .	28
4.8	Simulert utvikling av RRW, $\beta^g$ fra farget støy for STIM202 som ligger i ro (50 sekunder). . . . .	28
4.9	Simulert utvikling av ARW, $v_\omega^g$ , for STIM202 som ligger i ro (50 sekunder). . . . .	29
5.1	Utvikling av RRW ( $\beta^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder. . . . .	35
5.2	Utvikling av ARW ( $v_\omega^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder. . . . .	35
5.3	Utvikling av farget støy ( $\mu^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder. . . . .	36
5.4	Vinkelhastighet ( $\tilde{\omega}^g$ ) for 3 individuelle gyroer som ligger i ro. Simulert i 10000 sekunder. . . . .	36

## Figurer

---

5.5	Allanvariansen for gyro 1. . . . .	37
5.6	Tilfelle 1: Estimert vinkelhastighet fra gyromatrise der målinger fra 3 gyroer er kombinert, men uten korrelasjon. Simulert i 10000 sekunder. . . . .	38
5.7	Tilfelle 1: Allanvariansen for gyromatrise der målinger fra 3 gyroer er kombinert, men uten korrelasjon. . . . .	38
5.8	Tilfelle 2: Estimert vinkelhastighet fra gyromatrise der målinger fra 3 gyroer er kombinert, med korrelasjon. Simulert i 10000 sekunder. . . . .	39
5.9	Tilfelle 1: Allanvariansen for gyromatrise der målinger fra 3 gyroer er kombinert, med korrelasjon. . . . .	40
5.10	Utvikling av RRW ( $\beta^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder. . . . .	40
5.11	Utvikling av ARW ( $v_\omega^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder. . . . .	41
5.12	Utvikling av farget støy ( $\mu^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder. . . . .	41
5.13	Vinkelhastighet ( $\tilde{\omega}^g$ ) for 6 individuelle gyroer som ligger i ro. Simulert i 10000 sekunder. . . . .	42
5.14	Tilfelle 3: Estimert vinkelhastighet fra gyromatrise der målinger fra 6 gyroer er kombinert, men uten korrelasjon. Simulert i 10000 sekunder. . . . .	43
5.15	Tilfelle 3: Allanvariansen for gyromatrise der målinger fra 6 gyroer er kombinert, men uten korrelasjon. . . . .	43
5.16	Tilfelle 4: Estimert vinkelhastighet fra gyromatrise der målinger fra 6 gyroer er kombinert, med korrelasjon. Simulert i 10000 sekunder. . . . .	44
5.17	Tilfelle 3: Allanvariansen for gyromatrise der målinger fra 6 gyroer er kombinert, med korrelasjon. . . . .	45

# Tabeller

3.1	Symboler . . . . .	18
5.1	Oppsummering av simuleringsresultater . . . . .	45



# Kapittel 1

## Innledning

### 1.1 Om oppgaven

Masteroppgaven er, som nevnt i forordet, gjennomført i samarbeid med UNIK og FFI. Oppgaveteksten lyder som følger:

*I denne oppgave skal en se på hvilke forbedringer en kan forvente seg dersom en benytter seg av MEMS-gyromatriser (flere MEMS-gyroer plassert på samme brikke). Det foreslås at kandidaten*

- *gjør et litteraturstudium for å finne ut hvilke metoder som er foreslått for å utnytte målingene fra MEMS-gyromatriser*
- *lager en matematisk modell av målingene fra en MEMS-gyromatrise hvor korrelasjonen mellom enkeltgyroene er tatt med*
- *genererer gyromålinger ved hjelp av simuleringer av denne modellen*
- *undersøker ulike estimatorer og finner ut hvilken nøyaktighet en kan forvente på vinkelhastighetsestimaten*
- *dersom tiden tillater det, vurderer ytelsen ved bruk av slike gyromatriser i et treghetsnavigasjonssystem*

## 1.2 Struktur

Denne rapporten er delt inn i 7 kapitler. Foruten innledningskapitlet, består disse av:

**Kapittel 2: MEMS** En introduksjon til MEMS-teknologien, og litt om tidligere forskning.

**Kapittel 3: Teoretisk grunnlag** Her beskrives det fundamentale teoretiske grunnlaget for undersøkelsene gjort i rapporten.

**Kapittel 4: Modeller** De matematiske modellene utledes. Både for enkeltgyro og gyromatrise.

**Kapittel 5: Resultater** Resultatene fra forsøk og simuleringer presenteres.

**Kapittel 6: Konklusjon** Diskusjon og konklusjon.

**Kapittel 7: Videre arbeid** Forslag til videre arbeid innenfor det samme temaet.

# Kapittel 2

## MEMS

### 2.1 MEMS-gyroer

Et stort antall ulike typer gyroskop er i dag kommersielt tilgjengelige. Nøyaktige gyroer er tradisjonelt relativt store og kostbare. En type rimelige, små gyroer er MEMS-gyroene (MEMS: Micro-Electro-Mechanical System). Spesielt i bil-, fly- og forsvarsindustrien, samt innenfor forbrukerelektronikk er disse mye brukt. For en del applikasjoner har disse hatt for dårlig nøyaktighet, og det er derfor av stor interesse å finne ut hvordan ytelsen av denne typen gyroer kan forbedres.

En måte å forbedre nøyaktigheten på kan være å kombinere flere MEMS-gyroer i såkalte gyromatriser. Her er tanken at man setter sammen flere gyroer av samme type på samme chip, og kombinerer målingene fra disse for å få en mer nøyaktig måling.

Selv om vi ikke skal dykke dypt ned i den generelle MEMS-teknologien kan det være lurt å oppsummere litt av bakgrunnen for at MEMS er en interessant teknologi å benytte seg av. Som tidligere nevnt er det mye brukt i blant annet bilindustrien. I eksempelvis teknologien bak ESP-systemer og airbagsensorer er akselerometre mye brukt, for å kunne måle akselerasjoner ved hurtige og uønskede endringer i bilens kjøremønster. Som navnet antyder, “Micro-Electro-Mechanical System”, består slike systemer av meget små komponenter (i størrelsesorden  $1 - 100 \mu m$ ) som har både elektriske og mekaniske funksjoner. Etterspørselen i bilindustrien gjorde etterhvert til at MEMS-sensorer ble ansett som en pålitelig og vel ansett teknologi, og masseproduksjonen har i løpet av de siste 30 årene gjort at prisene har sunket og ført til at disse komponentene har blitt kommersielt tilgjengelige (se [17], s. 191).

## 2.1. MEMS-gyroer

---

I motsetning til MEMS-akselerometre, tok det lengre tid før gyroskop basert på MEMS-teknologi ble kommersielt tilgjengelig. Stimulien fra industrien var ikke like sterk som for akselerometrene, men i de siste årene er det blitt investert og forsket adskillig mer på MEMS-gyroer. Gyroskop brukes til svært mye i dag, og et viktig område er nordsøking (gyrokompass). Et tradisjonelt magnetkompass fungerer bra i mange sammenhenger, men det er utsatt for forstyrrelser fra metaller i bakken, og i tillegg viser det magnetisk nordpol. Et gyrokompass vil, ved hjelp av jordas rotasjon, måle sann nord, og her slipper vi påvirkningen fra metaller i omgivelsene. Problemet er at dagens gyrokompass ofte er store, tunge og dyre. Med en nordsøker basert på MEMS-teknologi, kunne en tenke seg at man hadde et håndholdt kompass med høy grad av brukervennlighet og en tilfredsstillende nøyaktighet. Selv om MEMS-gyroene har dårligere nøyaktighet enn for eksempel fiberoptiske gyroer, er likevel fordelene med lav pris og liten størrelse såpass store at det er ønskelig at disse tar mer og mer over.

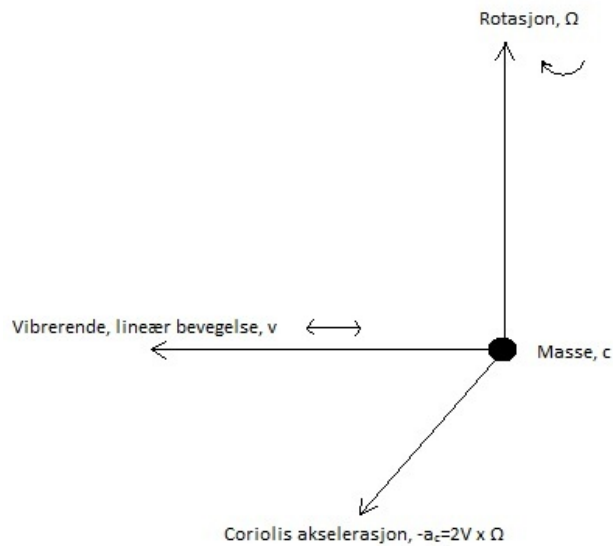
### 2.1.1 Virkemåte

En MEMS-gyro er, i likhet med et konvensjonelt gyroskop, en innretning (apparat) som brukes til å måle en rotasjonsbevegelse. Sammen med et eller flere akselerometre, som måler akselerasjon, kan et gyroskop være med på å måle bevegelsen til et objekt og bestemme hvor dette befinner seg til enhver tid (under ideelle forhold). Der det tradisjonelle gyroskopet med det spinnende hjulet baserer seg på loven om bevaring av bevegelsesmengde (Newtons 1. lov), hører MEMS-gyroskopet inn under gruppa av gyroskoper som utnytter *Corioliskraften*. Felles blant disse gyroene er at de er bygget opp uten roterende deler, men har i stedet vibrerende masser som detekterer rotasjonsbevegelser gjennom Corioliskraften.

Hvis vi tenker oss en masse  $c$ , (se figur 2.1) som blir utsatt for en lineær vibrerende bevegelse, der massen er plassert i en ramme som roterer rundt en akse som står vinkelrett på den lineære bevegelsen, vil et MEMS-gyroskop måle Corioliskraften som oppstår i dette systemet. Den vil da virke i en retning som står vinkelrett både på rotasjonsaksen  $\Omega$  og bevegelsesaksen  $v$ .

Dette er bare en kort beskrivelse av selve prinsippet, da MEMS-gyroene kan være bygget opp på forskjellige måter (simple oscillatorer, stemmegaffelgyroer og vinglass-resonatorer), men felles for disse er altså utnyttelse av Corioliskraften. Mer inngående beskrivelser er å finne i [17].





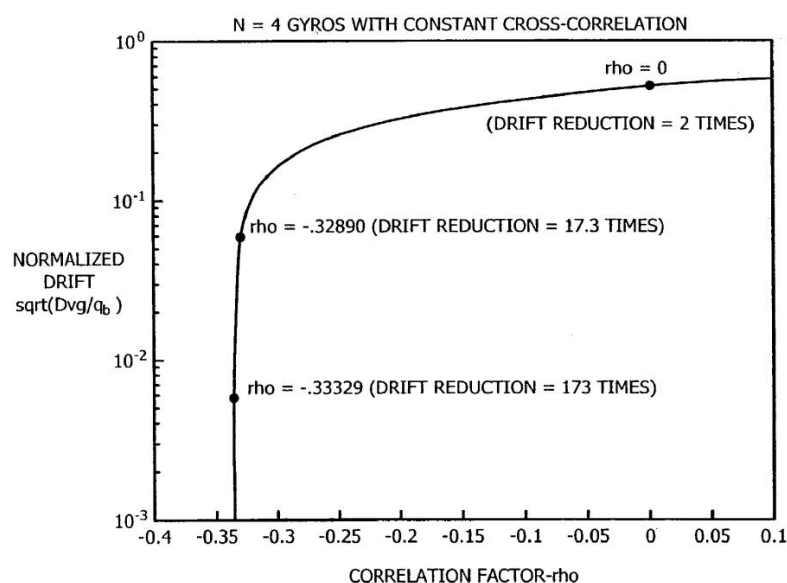
Figur 2.1: Hvordan corioliskraften oppstår i et MEMS-gyroskop [17]

## 2.2 Gyromatriser

Ideen med å kombinere sensorer med felles utgang har sitt utspring i flere fagmiljøer. Blant annet er det blitt laget medisinske beslutningssystemer og kombinerte navigasjonssystemer [12]. I 1998 klarte Allan (m.fl) å forbedre nøyaktigheten til tre armbåndsur (til \$6 per stk.) fra omtrent 30 sekunders feilvisning i løpet av en måned, ned til cirka ett sekunds feil per måned [1]. De plasserte de 3 urene tett på en plate i et rom, og fotograferte dem en gang om dagen til samme klokkeslett. Tiden klokken viste ble så sammenliknet via telefonlinjen mot et atom-ur hos NIST (National Institute of Standards and Technology). Siden telefonlinjas forsinkelse var i størrelsesorden 1 millisekund, antok de at hovedfeilkilden var temperaturens påvirkning på quartzkrystallene. Etersom tiden gikk observerte de tydelig korrelasjon mellom klokken (siden de ble utsatt for like temperaturvariasjoner), og de kunne trekke restfeilen til klokke 2 fra restfeilen til klokke 3 osv., etter at de var sammenliknet mot atom-uret. På denne måten fikk de fjernet den største feilkilden, og bare målestøy og ukorrelert, tilfeldig støy var igjen. Siden dette var så lite sammenliknet med resten, fikk de dermed økt prestasjonen betraktelig.

I tillegg er det i de siste årene blitt forsket på forbedring av MEMS-gyroskopenes

## 2.2. Gyromatriser



Figur 2.2: Drift av et virtuelt gyroskop (bestående av 4 gyroer) i forhold til korrelasjonsfaktor [5]

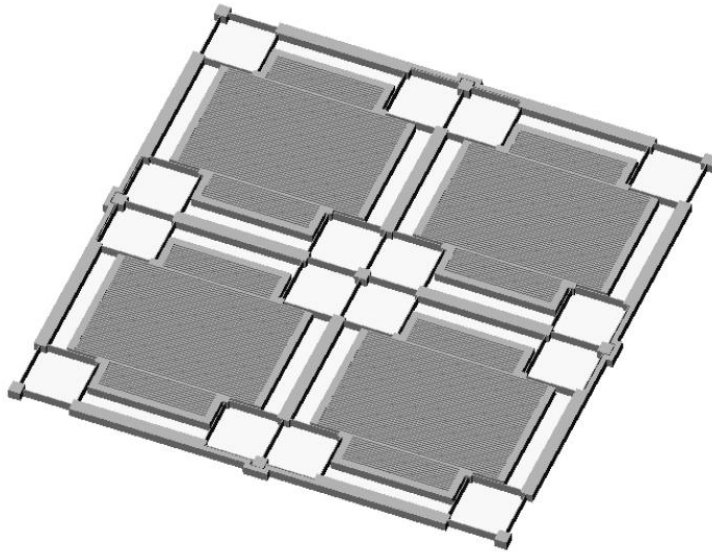
nøyaktighet, ved bruk av MEMS-gyromatriser. Bayard kombinerte i 2003 fire rimelige MEMS-gyroskop og designet en virtuell sensor med høyere grad av presisjon enn de opprinnelige gyroene. Han kalte denne teknologien for “virtual gyroscope” [5]. Her ble den tilfeldige støyen estimert ved hjelp av Kalmanfilter, slik at nøyaktigheten ble forbedret. Også Lam [11] og Chang [3] har gjort liknende forsøk, med gode resultater.

Det er slike gyromatriser, eller virtuelle gyroskop, vi skal se nærmere på i denne rapporten.

### 2.2.1 Korrelasjon mellom gyroene

Antakelser om korrelasjon mellom gyroene danner det teoretiske grunnlaget for de fleste som har forsket innenfor dette temaet. For å enklere estimere støyen som påvirker vinkelhastighetsmålingene fra gyroene, har for eksempel Bayard vist sammenhengen mellom korrelasjonsfaktor og forbedring, slik som vist på figur 2.2.

I utgangspunktet jobber jo hvert gyroskop uavhengig av andre og trolig er det ikke korrelasjon mellom dem, men om man kunne produsert disse samtidig med samme produksjonsutstyr, støpt de tett sammen på samme brikke



Figur 2.3: MEMS-gyroer plassert på en enkelt brikke [3]

innenfor et avgrenset område på bare noen få mikro-kvadratmeter, slik som vist på figur 2.3, kunne man antatt at det var korrelasjon tilstede.

Det er tydelig at korrelasjonen spiller en viktig rolle i forbedringene av nøyaktigheten, men Bayard kunne likevel i 2003 [5] vise at selv *uten* korrelasjon tilstede mellom gyroene ville det være mulig å oppnå forbedringer. Med  $N$  identiske og selvstendige gyroer viste han at det kunne være mulig å oppnå drift som var  $1/\sqrt{N}$  av driften til hver enkelt av  $N$  dem. Tilfellene *med* korrelasjon er likevel så mye bedre at dette er å foretrekke.



# Kapittel 3

## Teoretisk bakgrunn

### 3.1 Stokastiske variabler

Ordet stokastisk kommer fra gresk og betyr tilfeldig [6]. Navnet er beskrivende i seg selv, da en stokastisk variabel er umulig å forutsi nøyaktig. Dette er i motsetning til deterministisk, som betyr forutsigbar. Hvis vi har utfallet  $E_i$  i utfallsrommet  $U$ , så vil den stokastiske variabelen  $X$  ha en bestemt numerisk verdi  $x$  for alle  $i$ . Definisjon:

$$X : E \rightarrow x \in \mathbb{R} \quad (3.1)$$

der

$$P(E) = P(X = x) \in [0, 1] \quad (3.2)$$

Et typisk eksempel er et terningkast. Hvis vi har en tradisjonell terning med seks sider, aner vi ikke hvilken av sidene den lander på hvis vi foretar et kast. For terningen vil den stokastiske variabelen  $X$  betegne hvilke av utfallene  $\{1, 2, 3, 4, 5, 6\}$  vi risikerer å få. Vi kan få den samme verdien flere ganger, men aldri fler enn én verdi om gangen.

En stokastisk variabel  $X$  er gjerne beskrevet av en *sannsynlighetsfordeling*  $F(x)$ , som viser sannsynligheten  $P$  for hvert av utfallene. Definisjonen er [2]:

$$F(x) = P(X \leq x) \quad (3.3)$$

men den kan også beskrives av *sannsynlighetstetthetsfunksjonen*  $f(x)$ :

$$f(x) = \frac{dF(x)}{dx} \quad (3.4)$$

### 3.1. Stokastiske variabler

---

Forholdet mellom sannsynlighetsfordelingen og sannsynlighetstetthetsfunksjonen er:

$$F(x) = \int_{-\infty}^x f(x)dx \quad (3.5)$$

Det er verdt å merke seg at:

$$F(\infty) = \int_{-\infty}^{\infty} f(x)dx = 1 \quad (3.6)$$

I tillegg til disse verdiene, som gir et oversiktlig bilde av den stokastiske variabelen  $X$ , kan vi likevel hente ut mer informasjon gjennom *forventningsverdien*  $E\{X\}$ . Hvis vi hadde utført eksperimentet flere ganger, ville gjennomsnittet nærmet seg denne verdien. Definisjonen er:

$$\bar{x} = E\{X\} = \int_{-\infty}^{\infty} xf(x)dx \quad (3.7)$$

*Variansen* til en stokastisk variabel er et mål på spredningen. Det gjennomsnittlige, eller forventede avviket kaller vi standardavviket  $\sigma$ , som er kvadratroten av variansen:

$$Var(X) = \sigma^2 \quad (3.8)$$

, her er

$$\sigma^2 = \int_{-\infty}^{\infty} (x - E\{X\})^2 f(x)dx \quad (3.9)$$

$$= E\{(X - \bar{x})^2\} \quad (3.10)$$

Mens variansen er en egenskap som gjelder for én verdi, er *kovariansen*  $Cov(X, Y)$  en indikasjon på den statistiske korrelasjonen mellom de to stokastiske variablene  $X$  og  $Y$ . Definisjonen på denne sammenhengen er:

$$Cov(X, Y) = E\{(X - \bar{x})(Y - \bar{y})\} \quad (3.11)$$

Når vi under avsnitt 3.4 skal se på Kalmanfilteret, vil kovariansen dukke opp blant annet som  $P$ .  $P$  er kovariansmatrisa for tilstandsvektoren  $\bar{x}$ , der  $\bar{x}$  består av  $n$  elementer. På diagonalen i denne matrisa finner vi variansene til  $x_1, x_2, \dots, x_n$ , mens eventuelle ikke-diagonale elementer er krysskovariansen.

Korrelasjonskoeffisienten  $\rho$  kan vi finne ved å normalisere kovariansen med hensyn på standardavvikene til  $X$  og  $Y$  på følgende måte:

$$\rho = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (3.12)$$

Dette gir målet på den lineære avhengigheten mellom dem. Hvis  $\rho$  er 0 finnes ikke denne avhengigheten,  $X$  og  $Y$  er selvstendige. I det motsatte tilfellet, der  $\rho$  er  $\pm 1$ , er  $Y$  en lineær funksjon av  $X$ .

Den mest kjente statistiske fordelingen heter *normalfordelingen* (se figur 3.1), og vil bli brukt senere i oppgaven, blant annet for å beskrive hvit støy (beskrevet under avsnitt 4.1.1). Normalfordelingen kalles ofte Gausskurven eller Gaussisk fordeling, og sannsynlighetstetthetsfunksjonen er definert som:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad (3.13)$$

For vektoren  $\underline{x} = [x_1, x_2, \dots, x_n]^T$ , har vi den multivariate normalfordelingen

$$f_n(\underline{x}) = \frac{1}{(2\pi)^{n/2} \|P\|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\bar{x})^T P^{-1}(\underline{x}-\bar{x})} \quad (3.14)$$

hvor

$$\bar{x} = E\{\underline{x}\} \quad (3.15)$$

og

$$P = E\{(\underline{x} - \bar{x})(\underline{x} - \bar{x})^T\} \quad (3.16)$$

som angir henholdsvis middelveien og kovariansen til  $\underline{x}$ . For å angi at en stokastisk variabel er normalfordelt, skriver vi

$$\underline{X} \sim \mathcal{N}(\bar{x}, P) \quad (3.17)$$

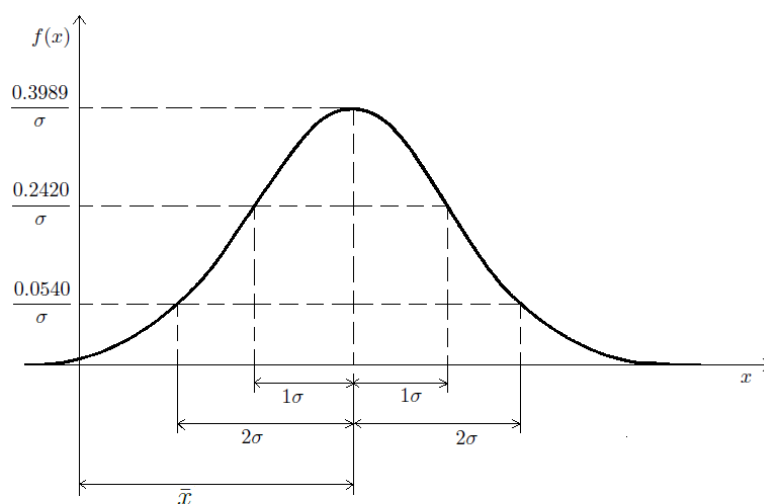
## 3.2 Stokastiske prosesser

En stokastisk prosess er en prosess som beskriver en samling (ensemble) av stokastiske variabler. En enkel forklaring er at den stokastiske prosessen  $X(t)$  er en stokastisk variabel  $X$  som endrer seg med tiden [4]. Grovt sett kan vi dele stokastiske prosesser inn i fire typer:

- Anta kontinuerlig tid  $t$  og at  $X$  er kontinuerlig for alle  $t$ . Vi har da en kontinuerlig stokastisk prosess. Et eksempel på dette er temperatur som funksjon av tid i løpet av en dag. Både tiden og temperaturen er kontinuerlige.

### 3.2. Stokastiske prosesser

---



Figur 3.1: Normalfordelingen [2]

- Hvis tiden er kontinuerlig og  $X$  diskret er  $X(t)$  en diskret stokastisk prosess. Eksempel: antall personer i en bygning til enhver tid. Antallet personer er diskret og tiden er kontinuerlig.
- Tilfellet der  $X$  er kontinuerlig og tiden er diskret, kaller vi en kontinuerlig tilfeldig (stokastisk) sekvens. Hvis vi for eksempel hadde målt temperaturer gjennom en uke og notert ned høyeste temperatur hver dag, ville vi hatt et slikt tilfelle. Temperatur ville vært gitt i kontinuerlig, mens tiden (dagene) hadde vært diskrete (mandag, tirsdag osv.)
- Den siste typen av stokastiske prosesser er når både tid og  $X$  er diskret. Dette kalles en diskret tilfeldig (stokastisk) sekvens. Hvis vi teller høyest antall personer i en bygning for hver dag i en uke, har vi et slikt tilfelle, der både antall personer og tid er diskrete.

Autokorrelasjonen forteller noe om den statistiske sammenhengen mellom samplene i en stokastisk prosess, og er definert ved [2]:

$$\phi_{xx} = E\{x(t_1)x(t_1 + \tau)\} \quad (3.18)$$

$$\phi_{xx}(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t_1)x(t_1 + \tau)f(x(t_1), x(t_2))dx(t_1)dx(t_2) \quad (3.19)$$



og krysskorrelasjonen, som forteller noe om sammenhengen mellom samplene i forskjellige stokastiske prosesser er:

$$\phi_{xy} = E\{x(t_1)y(t_2)\} \quad (3.20)$$

$$\phi_{xy}(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(t_1)y(t_1 + \tau)f(x(t_1), y(t_2))dx(t_1)dy(t_2) \quad (3.21)$$

Hvis  $E\{x(t_1)\}$ ,  $E\{x(t_2)\}$  og  $E\{y(t_2)\}$  er null, er disse korrelasjonsfunksjonene lik kovariansen til de respektive variablene. Hvis de blir normalisert med sine standardavvik har vi korrelasjonskoeffisientene (slik som vist i likning 3.12).

Vi sier at en stokastisk prosess er *stasjonær* hvis de statistiske egenskapene er konstante over tid. Dette vil si at sannsynlighetstetthetsfunksjonen for prosessen  $f(x_1, t_1)$  er uavhengig av måletidspunktet  $t_1$ . Da er også alle tilknyttede egenskaper som for eksempel forventningsverdien konstant. Hvis vi sier at  $t_2 = t_1 + \tau$  så er korrelasjonsfunksjonene bare avhengige av  $\tau$ :

$$\phi_{xx}(t_1, t_2)\phi_{xx}(\tau) = E\{x(t_1)x(t_1 + \tau)\} \quad (3.22)$$

$$\phi_{xy}(t_1, t_2)\phi_{xy}(\tau) = E\{x(t_1)y(t_1 + \tau)\} \quad (3.23)$$

For den stasjonære stokastiske prosessen  $x(t)$ , med autokorrelasjonen  $\phi_{xx}(\tau)$ , er effektspektraltettheten gitt ved:

$$\Phi_{xx}(\omega) = \int_{-\infty}^{\infty} \phi_{xx}(\tau)e^{-j\omega\tau} d\tau \quad (3.24)$$

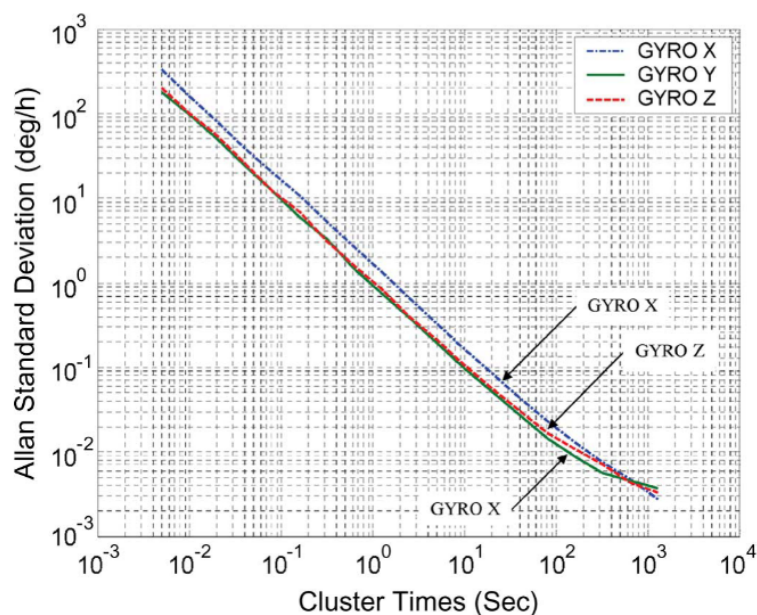
$$\phi_{xx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_{xx}(\omega)e^{j\omega\tau} d\omega \quad (3.25)$$

Dette viser hvordan styrken til signalet varierer i forhold til frekvens. Effektspektraltettheten forteller ved hvilke frekvenser variasjonene er store, og ved hvilke frekvenser variasjonene er små

### 3.3 Allanvarians

Allanvariansen, oppkalt etter David W. Allan, er et svært nyttig verktøy for å finne frekvensstabiliteter i blant annet oscillatorer, klokker og forsterkere. Når man estimerer en Allanvarians tas det hensyn til de underliggende støyprosessene, og metoden ser bort fra systematiske feilkilder som for eksempel frekvensdrift eller temperaturpåvirkninger. Dette gir et godt mål på stabiliteten til en oscillator, og man kan på bakgrunn av dette blant annet analysere hvor godt et gyroskop presterer.

### 3.3. Allanvarians



Figur 3.2: Eksempel på Allanvarians beregnet for 3 gyroer [9]

Fremgangsmåte: Et gyroskop måler en vinkelhastighet  $\omega(t)$  hvert  $T$  sekund, og vi ender opp med  $M$  målinger. Målingene betegnes som  $\omega_k = \omega(kT)$ . Datasettet ordnes så i grupper med lengder  $T, 2T, \dots, kT$  hvor  $k < M/2$  og vi beregner gjennomsnittsverdien til hver gruppe over lengden på gruppen.

Allanvariansen er definert som [16]

$$\sigma_{AVAR}^2(\tau) = \frac{1}{2} \langle (\omega_{k+m} - \omega_k)^2 \rangle = \frac{1}{2\tau^2} \langle (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \rangle \quad (3.26)$$

hvor  $\langle \rangle$  er ensemblegjennomsnittet,  $\tau = mT$  (lengden på gruppene).  $\theta_i$  er vinkelen beregnet fra følgende definisjon

$$\theta(t_i) = \int_{t_{i-1}}^{t_i} \omega(\bar{t}) d\bar{t} \quad (3.27)$$

der  $t_i = T_i$ . Formel for å beregne Allanvariansen:

$$\sigma_{AVAR}^2(\tau) = \frac{1}{2\tau^2(M-2m)} \sum_{k=1}^{M-2m} (\theta_{k+2m} - 2\theta_{k+m} + \theta_k)^2 \quad (3.28)$$

Sammenhengen mellom Allanvariansen og effektspektraltettheten er:

$$\sigma_{AVAR}^2(\tau) = 4 \int_0^\infty S_\omega(f) \frac{\sin^4(\pi f \tau)}{(\pi f \tau)^2} df \quad (3.29)$$

Når man har beregnet Allanvariansen og laget et log-log plot (som for eksempel i figur 3.2), kan man lese av hvor godt gyroskopet kan prestere. IEEE har i sin standard Std 952-1997 [16] beskrevet de forskjellige parametrene som beskriver støyen, og støyparametre som er spesielt interessante i forbindelse med Allanvariansen er bias instability og angular random walk. I kapittel 4 er den matematiske modellen for gyroskopet i denne rapporten beskrevet i detalj. MATLAB-koden som er i bruk for å beregne Allanvariansen i denne oppgaven ligger vedlagt i vedlegg B.7.

I følge [18] kan man ut ifra Allanvariansen finne bias instabilityen ved å lese av bunnpunktet i plottet. Bias instabilityen sier noe om hvor stor feilvisningen er over tid, eller hvor mye gyroskopet “vandrer”. Dette kommer blant annet av Rate Random Walk, som blir forklart nærmere i avsnitt 4.1.1. Man kan også lese av Angular Random Walk (avsnitt 4.1.1), den finner vi der samplingstiden er på 1s. Det vil bli analyser av Allanvariansen i kapittel 5 (resultater).

### 3.4 Kalmanfilter

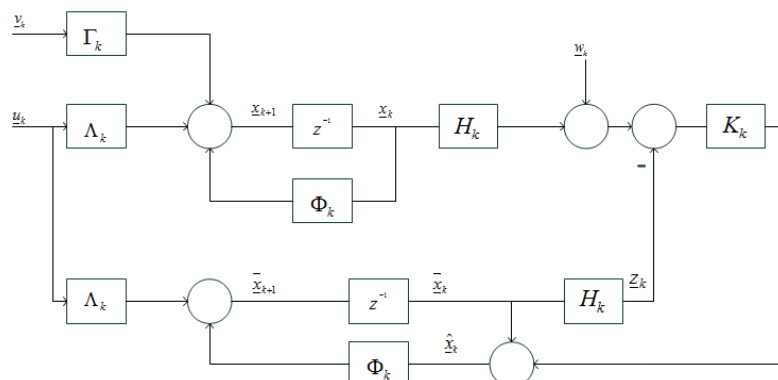
I denne oppgaven står Kalmanfilteret helt sentralt, som er en matematisk metode oppkalt etter Rudolf E. Kalman. Kalmanfilteret er et rekursivt filter som blir brukt for å estimere tilstandene i et lineært stokastisk dynamisk system gitt av tilstandsvektoren  $\underline{x}$  [2]. At et filter er rekursivt vil si at det ikke behøver å lagre tidligere målinger for å kunne beregne seg frem til et estimat. Kalmanfilteret har fordelen ved at det er en optimal estimator. En optimal estimator er en beregningsalgoritme som på grunnlag av kunnskap om prosess og sensorer, samt kunnskap om prosess- og målestøy kan beregne et minimum-varians estimat av tilstanden. Det vil si at vi ønsker å beskrive systemet gjennom en ordinær differensiallikning og en algebraisk likning:

$$\dot{\underline{x}} = F\underline{x} + G\underline{v} + L\underline{u} \quad (3.30)$$

$$\underline{z} = H\underline{x} + \underline{w} \quad (3.31)$$

og i tillegg trenger vi  $\underline{x}_0$ ,  $\underline{v}$  og  $\underline{w}$  beskrevet statistisk, samt at vi antar følgende betingelser:

### 3.4. Kalmanfilter



Figur 3.3: Blokkskjema av et diskret Kalmanfilter

$$\begin{aligned}
 E\{\underline{x}(t_0)\} &= \bar{\underline{x}}_0 & E\{(\underline{x}(t_0) - \bar{\underline{x}}_0)(\underline{x}(t_0) - \bar{\underline{x}}_0)^T\} &= \bar{P}_0 & E\{\underline{x}(t_0)\underline{v}^T(t)\} &= \underline{0} \\
 E\{\underline{v}(t)\} &= \underline{0} & E\{\underline{v}(t)\underline{v}^T(\tau)\} &= \delta(t - \tau)\tilde{Q}(t) & E\{\underline{x}(t_0)\underline{w}_k^T\} &= \underline{0} \\
 E\{\underline{w}_k\} &= \underline{0} & E\{\underline{w}_k\underline{w}_l^T\} &= \delta_{kl}R_k & E\{\underline{v}(t)\underline{w}_k^T\} &= \underline{0}
 \end{aligned}$$

#### 3.4.1 Kalmanfilterlikningene

Når man har laget en matematisk modell av det fysiske systemet man ønsker å analysere, kan man sette opp filtermodellen og implementere Kalmanfilteret på en datamaskin. I denne oppgaven er det brukt MATLAB for å benytte filteret på gyromålingene. Kalmanfilteret jobber på den måten at det beregner en Kalmanfilterforsterkning, og utvikler middelværdi og kovarians for hver tilstand over tid, slik at estimatet beregnes optimalt. Et eksempel på et diskret Kalmanfilter er vist på figur 3.3. Følgende likninger danner grunnlaget for Kalmanfilteret i denne rapporten [10] (TO=Tidsoppdatering, MO=Måleoppdatering):

#### Kontinuerlig Kalmanfilter

$$\begin{aligned}
 \dot{\hat{\underline{x}}}(t) &= F(t)\hat{\underline{x}}(t) + L(t)\underline{u}(t) + K(t)(z(t) - H(t)\hat{\underline{x}}(t)); \quad \hat{\underline{x}}(t_0) \text{ gitt} \\
 \dot{\hat{P}}(t) &= F(t)\hat{P}(t) + \hat{P}(t)F^T(t) + G(t)\tilde{Q}(t)G^T(t) - K(t)\tilde{R}(t)K^T(t); \quad \hat{P}(t_0) \text{ gitt}
 \end{aligned}$$

### Diskret Kalmanfilter

$$\left. \begin{aligned} \bar{\underline{x}}_{k+1} &= \Phi_k \hat{\underline{x}}_k + \Lambda_k \underline{u}_k; \quad \hat{\underline{x}}_0 \text{ gitt} \\ \bar{P}_{k+1} &= \Phi_k \hat{P}_k \Phi_k^T + \Gamma_k Q_k \Gamma_k^T; \quad \hat{P}_0 \text{ gitt} \end{aligned} \right\} \text{TO}$$

$$\left. \begin{aligned} \hat{\underline{x}}_k &= \bar{\underline{x}}_k + K_k (\underline{z}_k - H_k \bar{\underline{x}}_k) \\ K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} \\ \hat{P}_k &= (I - K_k H_k) \bar{P}_k \end{aligned} \right\} \text{MO}$$

### Kontinuerlig-diskret Kalmanfilter

$$\left. \begin{aligned} \dot{\hat{\underline{x}}}(t) &= F(t) \bar{\underline{x}}(t) + L(t) \underline{u}(t), \quad \bar{\underline{x}}(t_k^+) := \hat{\underline{x}}_k; \quad \hat{\underline{x}}_0 \text{ gitt} \\ \dot{\bar{P}}(t) &= F(t) \bar{P}(t) + \bar{P}(t) F^T(t) + G(t) \tilde{Q}(t) G^T(t), \quad \bar{P}(t_k^+) := \hat{P}_k; \quad \hat{P}_0 \text{ gitt} \end{aligned} \right\} \text{TO}$$

$$\left. \begin{aligned} \hat{\underline{x}}_k &= \bar{\underline{x}}_k + K_k (\underline{z}_k - H_k \bar{\underline{x}}_k), \quad \bar{\underline{x}}_k := \bar{\underline{x}}(t_k) \\ K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1}, \quad \bar{P}_k := \bar{P}(t_k) \\ \hat{P}_k &= (I - K_k H_k) \bar{P}_k \end{aligned} \right\} \text{MO}$$

### 3.4.2 Symbolforklaring

Tabell 3.1 på neste side viser et sammendrag av de symbolene og variablene som er brukt i forbindelse med Kalmanfilteret [10].

Kontinuerlig system		Diskret system	
<i>Symbol</i>	<i>Forklaring</i>	<i>Symbol</i>	<i>Forklaring</i>
$t$	tidspunkt $t$	$k$	tidspunkt $t_k$ (sample nr. $k$ )
$\underline{x}(t)$	tilstandsvektor (dim $\underline{x} = n_x$ )	$\underline{x}_k$	tilstandsvektor (dim $\underline{x} = n_x$ )
$\underline{v}(t)$	prosessstøy (dim $\underline{v} = n_v$ )	$\underline{v}_k$	prosessstøy (dim $\underline{v}_k = n_{v_k}$ )
$\underline{u}(t)$	pådrag (dim $\underline{u} = n_u$ )	$\underline{u}_k$	pådrag (dim $\underline{u}_k = n_{u_k}$ )
$\underline{z}(t)$	måling (dim $\underline{z} = n_z$ )	$\underline{z}_k$	måling (dim $\underline{z}_k = n_{n_z}$ )
$\underline{w}(t)$	målestøy (dim $\underline{w} = n_z$ )	$\underline{w}_k$	målestøy (dim $\underline{w}_k = n_{n_z}$ )
$F(t)$	systemmatrisa ( $n_x \times n_x$ )	$\Phi_k$	systemmatrisa ( $n_x \times n_x$ )
$G(t)$	prosessstøymatrisa ( $n_x \times n_v$ )	$\Gamma_k$	prosessstøymatrisa ( $n_x \times n_{v_k}$ )
$L(t)$	pådragsmatrisa ( $n_x \times n_u$ )	$\Lambda_k$	pådragsmatrisa ( $n_x \times n_u$ )
$H(t)$	målematrisa ( $n_z \times n_x$ )	$H_k$	målematrisa ( $n_z \times n_x$ )
$P_0$	initieell kovariansmatrise ( $n_x \times n_x$ )	$P_0$	initieell kovariansmatrise ( $n_x \times n_x$ )
$\tilde{Q}(t)$	prosessstøyens spektraltetthet ( $n_v \times n_v$ )	$Q_k$	prosessstøyens kovarians ( $n_{v_k} \times n_{v_k}$ )
$\tilde{R}(t)$	målestøyens spektraltetthet ( $n_z \times n_z$ )	$R_k$	målestøyens kovarians ( $n_z \times n_z$ )
$\delta(t - \tau)$	Diracs $\delta$ -funkt: $\int_{-\infty}^{\infty} f(\tau)\delta(t - \tau)d\tau = f(t)$	$\delta_{kl}$	Kronecker $\delta$ : $\begin{cases} \delta_{kl} = 1 & \text{for } k = l \\ \delta_{kl} = 0 & \text{for } k \neq l \end{cases}$

Tabell 3.1: Symboler

# Kapittel 4

## Modeller

### 4.1 Gyromodell

#### 4.1.1 Støykilder

En MEMS-gyro, i likhet med andre typer gyroskop, vil hele tiden bli utsatt for støy. Typisk for MEMS-gyroer er jo at de gjerne presterer dårligere enn tradisjonelle gyroer, derfor er det viktig å finne gode modeller for de forskjellige typene av feil. Det finnes en hel rekke støykilder, hvor noen er mer fremtredende enn andre. Siden vi i denne rapporten kun foretar målinger fra simulerte gyroer, er det av stor interesse å lage så korrekte matematiske modeller av støyen som mulig. Vi skal nå se på de støykildene som er tatt med i rapporten og i simuleringene.

#### ARW (Angular Random Walk)

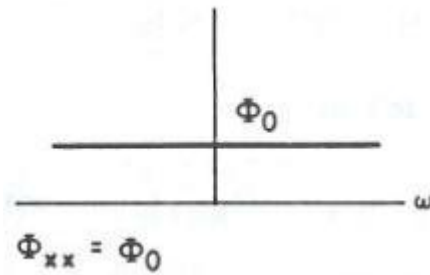
Angular Random Walk,  $v_{\omega}^g$ , er hvit støy som inntreffer på vinkelhastighetsnivå direkte under målingene. En hvit støyprosess er en stokastisk prosess med gjennomsnittsverdi 0 og standardavvik  $\sigma$ , slik det er vist på figur 4.2. Effektspektraltettheten til hvitstøyen vil være lik over alle  $t$ , det vil si at den er konstant (se figur 4.1). Den hvite støyen har fått sitt navn fordi den i likhet med hvitt lys inneholder alle frekvenser. Perfekt hvit støy har uendelig båndbredde, men i praksis er dette umulig. Hvit støy er derfor et idealisert begrep, men brukes i stort grad som en god tilnærming til naturlige støykilder.

De tilfeldige verdiene vi kan lese av fra en normalfordelt hvit støyprosess vil være ukorrelerte, og vi kan kalle denne støyen for Gaussisk hvit støy.

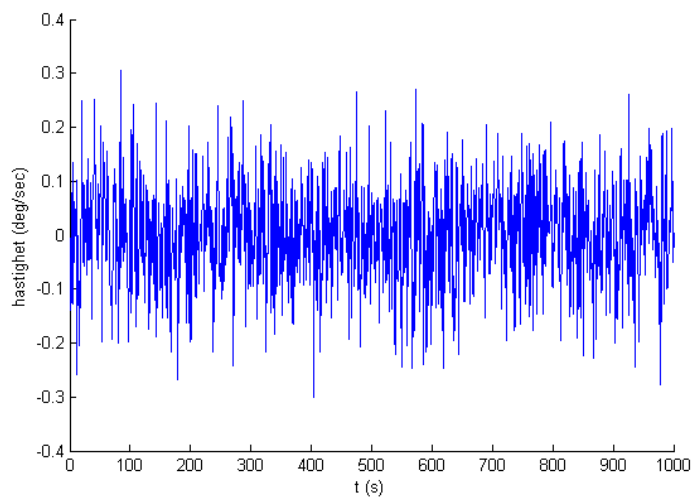
I tilfellet hvor vi måler vinkelhastigheter for gyroer kan navnet ARW i reali-

## 4.1. Gyromodell

---



Figur 4.1: Effektspektraltetthet for hvit støy [2]



Figur 4.2: Eksempel på hvit støy



teten være litt misvisende. Selv om navnet indikerer random walk, vil støyprosessen få en slik karakteristikk først når vi integrerer opp signalet for å lese av vinkelen. Da vil støyen bli forsterket og ligne det støybidraget som kommer fra RRW (Rate Random Walk) (se avsnitt 4.1.1).

### Farget støy

Målingene påvirkes også av farget støy. Det viser seg at å modellere den fargede støyen som en 1. ordens markovprosess gir simuleringresultater som likner svært mye på målinger fra en reell gyro.

Gauss-markovprosesser er en samlebetegnelse for spesielle random-prosesser som oppstår ved at hvit støy blir filtrert gjennom enkle filtre. Den kontinuerlige prosessen  $x(t)$  er en 1. ordens markovprosess hvis [2]

$$t_1 < t_2 < \dots < t_k$$

for alle  $k$ , og hvis

$$F[x(t_k) | x(t_{k-1}), \dots, x(t_1)] = F[x(t_k) | x(t_{k-1})] \quad (4.1)$$

Det vil si at sannsynlighetstetthetsfunksjonen for prosessen  $x(t_k)$  kun avhenger av verdien i forrige tidsskritt, altså  $x(t_{k-1})$ . Hvis den kontinuerlige prosessen  $x(t)$  er en 1 ordens markovprosess, kan den beskrives av differensiallikningen [2]

$$\frac{dx}{dt} + \frac{1}{T}x = v \quad (4.2)$$

der  $v$  er en hvit støy. Hvis vi har at sannsynlighetstetthetsfunksjonene til  $v$  og  $x$  er gaussiske, kalles dette en gauss-markov prosess. Den stasjonære gauss-markovprosessen beskrives av autokorrelasjonsfunksjonen [2]

$$\phi_{xx}(\tau) = \sigma^2 e^{-\frac{1}{T}|\tau|} + m^2 \quad (4.3)$$

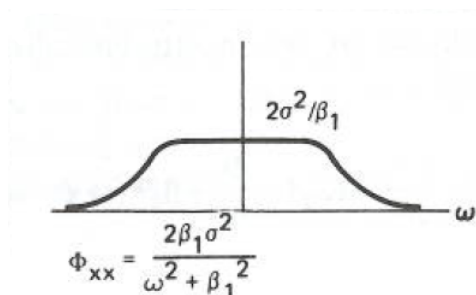
hvor  $T$  er korrelasjonstiden.

I vårt tilfelle kaller vi den fargede støyen for  $\mu^g$ , og vi kan nå sette opp differensiallikningen for den fargede støyen:

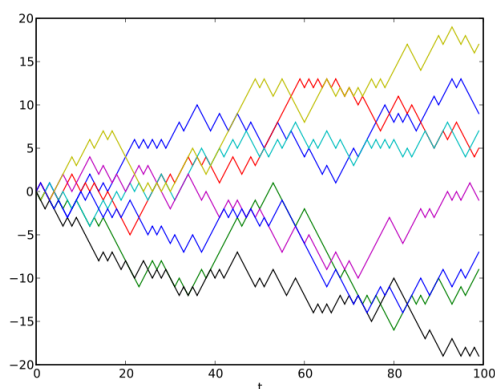
$$\dot{\mu}^g = -\frac{1}{T}\mu^g + v_\mu^g \quad (4.4)$$

## 4.1. Gyromodell

---



Figur 4.3: Effektspektraltetthet for markovprosess [2]



Figur 4.4: Eksempel på random walk

### RRW (Rate Random Walk)

RRW, eller Rate Random Walk, kaller vi i denne rapporten for  $\beta^g$ , der  $\dot{\beta}^g = v_\beta^g$ , og  $v_\beta^g$  er en hvit støy. Denne random-prosessen fungerer slik: vi ser for oss en person som står på en vei som går i nord/sør retning. Personen kaster kron ( $K$ ) og mynt ( $M$ ), der kron betyr et skritt nordover, mens mynt betyr et skritt sørover. Siden sannsynligheten for begge mulige utfall (kron og mynt) er like ( $P(K) = P(M) = \frac{1}{2}$ ), ville vi kanskje gjettet at personen etter mange forsøk ville havnet i utgangsposisjonen. Dette er ikke tilfelle. Siden personen hele tiden ville startet i et nytt, tilfeldig punkt, kunne han like gjerne beveget seg lenger og lenger unna startpunktet. Det vil derfor være helt tilfeldig hvor langt nord eller hvor langt sør personen ender opp etter en viss tid. Figur 4.4 viser et eksempel der man har kjørt 8 serier med random walk-prosesser.

### 4.1.2 Feilmodell

Vi ønsker å sette opp en matematisk modell av vinkelhastighetsmålingene fra et gyroskop som kan brukes til simuleringer. Ut fra de feilkildene vi har sett på, kan vi nå oppsummere de lineære differensiallikningene for farget støy og RRW:

$$\dot{\mu}^g = -\frac{1}{T}\mu^g + v_\mu^g \quad (4.5)$$

$$\dot{\beta}^g = v_\beta^g \quad (4.6)$$

Tilstandsvektoren for feilkildene:

$$\underline{x}' = \begin{bmatrix} \beta^g \\ \mu^g \end{bmatrix} \quad (4.7)$$

Tilstandsrommodellen blir:

$$\underbrace{\begin{bmatrix} \dot{\beta}^g \\ \dot{\mu}^g \end{bmatrix}}_{\underline{\dot{x}'}} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{T} \end{bmatrix}}_{F'} \underbrace{\begin{bmatrix} \beta^g \\ \mu^g \end{bmatrix}}_{\underline{x}'} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{G'} \underbrace{\begin{bmatrix} v_\beta^g \\ v_\mu^g \end{bmatrix}}_{\underline{v}'} \quad (4.8)$$

Vi kan nå sette opp den generelle målelikningen for en tre-akset gyro for målt vinkelhastighet ( $\tilde{\omega}^g$ ) basert på sann vinkelhastighet ( $\underline{\omega}^g$ ) drevet av feilkildene:

$$\tilde{\omega}^g = \underline{\omega}^g + \underline{\mu}^g + \underline{\beta}^g + \underline{v}_\omega^g \quad (4.9)$$

Komponentene er:

$$\underline{\omega}^g = \begin{bmatrix} \omega_x^g \\ \omega_y^g \\ \omega_z^g \end{bmatrix} \quad (4.10)$$

og tilsvarende for  $\underline{\mu}^g$ ,  $\underline{\beta}^g$  og  $\underline{v}_\omega^g$ . I denne oppgaven vil vi kun ta hensyn til én akse (f.eks. x-aksen), og variablene i likningen vil da bli:

$$\omega = \omega_x^g \quad (4.11)$$

$$\tilde{\omega} = \tilde{\omega}_x^g \quad (4.12)$$

$$\mu = \mu_x^g \quad (4.13)$$

## 4.1. Gyromodell

---

$$\beta = \beta_x^g \quad (4.14)$$

$$v = v_{\omega,x}^g \quad (4.15)$$

Vinkelhastighetsmålingene fra gyroen blir da:

$$\tilde{\omega}_k = \omega_k + \mu_k + \beta_k + v_\omega \quad (4.16)$$

### 4.1.3 Implementering av feilmodell i MATLAB

I MATLAB-simuleringene bruker vi matrisene fra likning 4.8, og i tillegg har vi

$$\hat{P}'_0 = \begin{bmatrix} \sigma_\beta^2 & 0 \\ 0 & \sigma_\mu^2 \end{bmatrix} \quad (4.17)$$

$$H' = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad (4.18)$$

$$T_s = 0.001 \quad (4.19)$$

, slik at vi kan simulere det kontinuerlig-diskrete systemet

$$\dot{\underline{x}}' = F' \underline{x}' + G' \underline{v}' \quad (4.20)$$

$$\underline{z}'_k = H' \underline{x}'_k + w'_k \quad (4.21)$$

Dette gir en måleserie fra étt enkelt gyroskop som står i ro med støykildene som eneste pådrag.

### Diskretisering

Som vi kan se av likningene 4.20 og 4.21, er systemet vi skal simulere på kontinuerlig-diskret form. Det vil si at vi har en kontinuerlig prosess med diskrete målinger, og det er derfor nødvendig å diskretisere systemet. Vi ønsker prosesslikningen på formen:

$$\underline{x}'_{k+1} = \Phi' \underline{x}'_k + \Gamma' \underline{v}' \quad (4.22)$$

Det er laget følgende MATLAB-funksjoner for å diskretisere:

**[La,Fi]=kp2dpLa(F,L,Ts):** I denne funksjonen finner vi  $\Lambda$  og  $\Phi$  på følgende måte:

$$\Lambda = \int_0^{\Delta t} e^{F \cdot \Delta t - \tau} \cdot L d\tau \quad (4.23)$$

og

$$\Phi = e^{F \cdot \Delta t} \quad (4.24)$$

**[S]=kp2dpS(F,G,Qb,Ts):** For å finne  $\Gamma$  finner vi først  $S = \Gamma Q \Gamma^T$ . Vi har at

$$S = \Gamma Q \Gamma^T = \int_0^{\delta t} \Phi G \tilde{Q} G^T \Phi^T d\tau \quad (4.25)$$

**[Ga]=kp2dpGa(F,G,Qb,Ts):** Vi spalter  $\Gamma$  og  $Q$  på følgende måte:  $S = UDU^T$  hvor  $U$  og  $D$  finnes ved hjelp av Cholesky-faktorisering. Siden  $S \geq 0$  (positiv semidefinit) vil  $D \geq 0$  og diagonal:

$$D^{\frac{1}{2}} = \begin{bmatrix} \sqrt{d_{11}} & 0 & \dots & 0 \\ 0 & \sqrt{d_{22}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sqrt{d_{nn}} \end{bmatrix} \quad (4.26)$$

$$\begin{aligned} \Rightarrow S &= UDU^T = UD^{\frac{1}{2}} \cdot I \cdot D^{\frac{1}{2}} \cdot U^T \\ \Rightarrow \Gamma &= UD^{\frac{1}{2}} \end{aligned}$$

### Pseudokode for enkeltgyro

% Diskretisering:

$$\Phi' = e^{F'T_s};$$

$$\Gamma' = kp2dpGa(F', G', \tilde{Q}, T_s);$$

% Initialiseringer:

$$\omega = \omega_0 = 0;$$

$$x'_0 \sim \mathcal{N}(0, \hat{P}'_0);$$

$$\mu_0 \sim \mathcal{N}(0, \sigma_\mu^2);$$

$$\beta_0 \sim \mathcal{N}(0, \sigma_\beta^2);$$

% Simulering av gyro:

for  $k = 0$  to  $N$

$$\underline{x}'_{k+1} = \Phi' \underline{x}'_k + \Gamma' * \underline{v}'_k;$$

$$\underline{z}'_k = H' \underline{x}'_k + \underline{w}'_k;$$

end



Figur 4.5: MEMS-gyroen STIM202 fra Sensoror

### 4.1.4 STIM202

STIM202 er en høytstående MEMS-gyromodul utviklet av Sensoror Technologies AS i Horten. Sensoror har spesialisert seg som produsent og utvikler av MEMS-sensorer, og miljøet rundt Sensoror har holdt på med forskning og utvikling av elektronikk siden 1960 [13]. Sensoror er kanskje mest kjent som en av de største aktørene på markedet innenfor TPMS<sup>1</sup>.

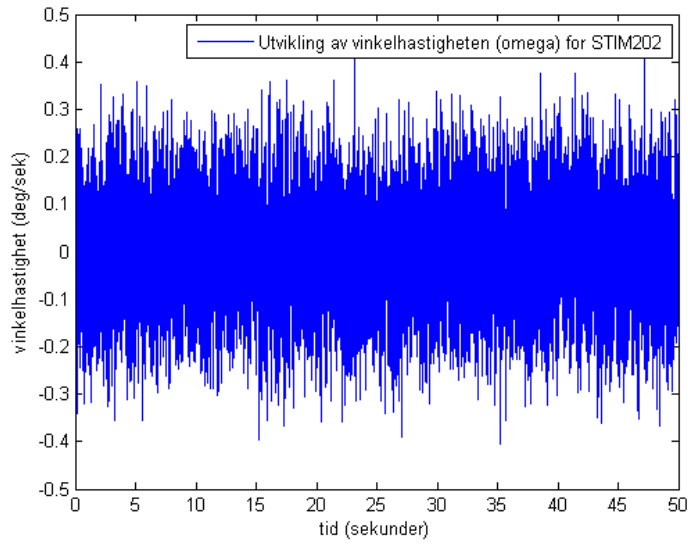
Det vært viktig både for motivasjon og relevans at simuleringene i denne oppgaven har vært så reelle som mulig. Det er derfor tatt utgangspunkt i noe av det beste som finnes på markedet i dag, og valget av gyro har i samarbeid med FFI falt på STIM202. STIM202 er meget konkurransedyktig; den har lite drift, høy samplingstid, lite støy og et forholdsvis enkelt brukergrensesnitt. Utdrag fra databladet ligger vedlagt i vedlegg A.

I 2010 ble Sensoror tildelt prisen “European High Precision MEMS Gyroscopes New Product Innovation Award” av Frost and Sullivan [14], og de fikk dermed anerkjennelse for sitt nye produkt. STIM202 leveres med gyroer med både 1, 2 og 3 akser, og er tilrettelagt for enkel overføring til PC med RS422.

I simuleringene av STIM202 er parametrene som er brukt hentet fra [8], der Grahn har estimert de parametrene som trengs ved hjelp av Maximum Likelihood-estimering. Det er som nevnt tidligere kun simulert med én akse. Figurene 4.6, 4.7 og 4.8 viser hvordan henholdsvis vinkelhastighet, farget støy

---

<sup>1</sup>Tire pressure monitoring system

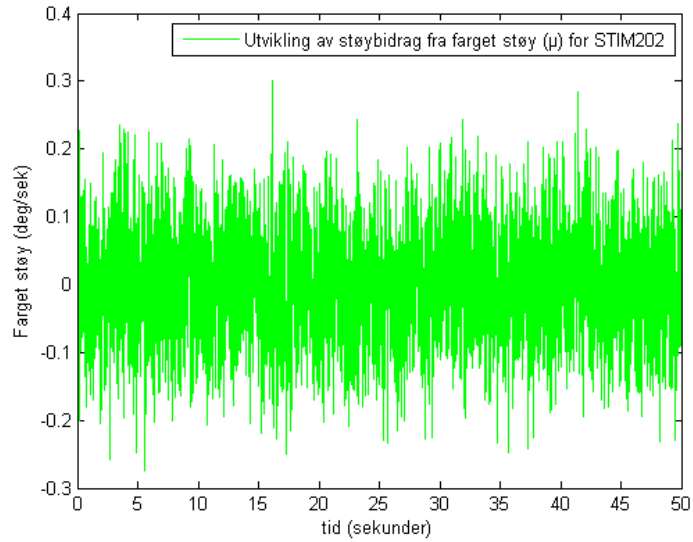


Figur 4.6: Simulert utvikling av vinkelhastighet,  $\omega^g$ , for STIM202 som ligger i ro (50 sekunder).

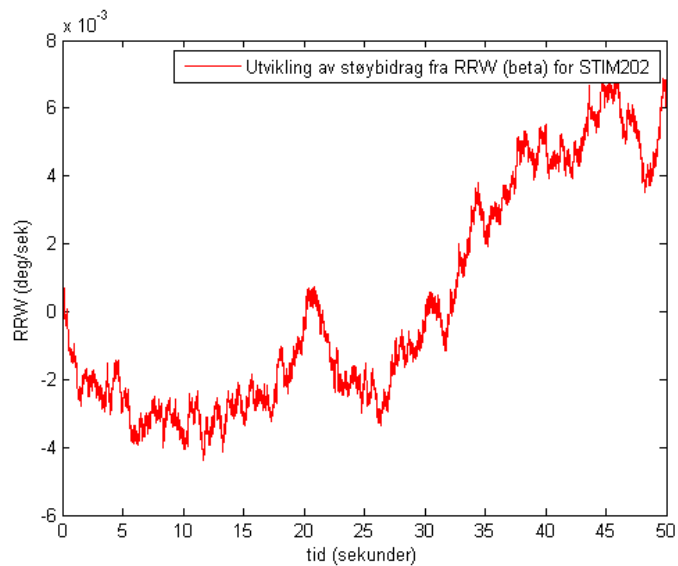
og RRW utvikler seg for en simulert en-akset gyro basert på disse parametrene. Figur 4.9 viser ARW (hvit støy i målingene).

## 4.1. Gyromodell

---

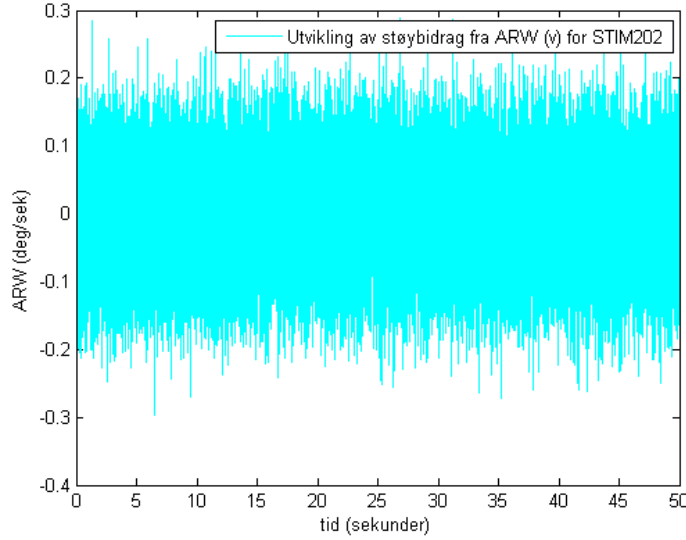


Figur 4.7: Simulert utvikling av støybidrag fra farget støy,  $\mu^g$ , for STIM202 som ligger i ro (50 sekunder).



Figur 4.8: Simulert utvikling av RRW,  $\beta^g$  fra farget støy for STIM202 som ligger i ro (50 sekunder).





Figur 4.9: Simulert utvikling av ARW,  $v_{\omega}^g$ , for STIM202 som ligger i ro (50 sekunder).

## 4.2 Gyromatrise

### 4.2.1 Modell av gyromatrise

En gyromatrise baserer seg på målinger fra flere gyroer som måler den samme vinkelhastigheten. Meningen er at gyromatrisa skal gi en mer nøyaktig måling enn det hver av gyroene klarer på egenhånd. I denne oppgaven er det tatt utgangspunkt i modellene fra [5] og [3]. Gyromålingene som disse gyromatrisene benytter seg av er derimot modellert litt annerledes, så det har vært nødvendig å tilpasse gyromatrisa i denne oppgaven slik at støy- og feilkilder stemmer overens med de modellene som ble presentert i 4.1. I tillegg til at gyroene skal være produsert samtidig, satt tett sammen på en brikke, og målingene fra disse kombineres, benytter man et Kalmanfilter som hjelper til med å estimere vinkelhastighetsmålingene og filtrere bort støy i sanntid.

Hvis vi husker målelikningen (likning 4.16), så gjaldt den for én gyro. Anta at vi har en gyromatrise med  $N$  enkeltgyroer der gyroene er identiske og av typen STIM202. Vi får da målelikningen for gyro nummer  $i$ :

$$\tilde{\omega}_{k,i} = \omega_{k,i} + \mu_{k,i} + \beta_{k,i} + v_{k,\omega_m} \quad (4.27)$$

hvor  $v_{\omega_m} \sim \mathcal{N}(0, Q_{\omega_m})$  som er gyromatrisas ARW. Det er gjennomgående i

## 4.2. Gyromatrise

---

denne rapporten at de tenkte simuleringsscenarioene tar for seg gyroer som ligger i ro, derfor er den sanne vinkelhastigheten  $\omega$  drevet av den hvite støyen på samme måte som for enkeltgyroene.

Det er ønskelig å sette opp systemet på formen  $\dot{\underline{x}} = F\underline{x} + G\underline{v}$ , og vi får at:

$$\underbrace{\begin{bmatrix} \dot{\beta}_1^g \\ \vdots \\ \dot{\beta}_i^g \\ \dot{\tilde{\omega}} \end{bmatrix}}_{\dot{\underline{x}}} = \underbrace{\begin{bmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{bmatrix}}_F \underbrace{\begin{bmatrix} \beta_1^g \\ \vdots \\ \beta_i^g \\ \tilde{\omega} \end{bmatrix}}_{\underline{x}} + \underbrace{\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}}_G \underbrace{\begin{bmatrix} v_{\beta_1}^g \\ \vdots \\ v_{\beta_i}^g \\ v_{\omega_m}^g \end{bmatrix}}_{\underline{v}} \quad (4.28)$$

Målingene er gitt ved:

$$\underline{z} = H\underline{x} + B\underline{w} \quad (4.29)$$

med matrisene

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & 1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix} \quad (4.30)$$

$$B = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \quad (4.31)$$

og målestøyvektoren

$$\underline{w} = \begin{bmatrix} v_{\mu_1}^g \\ \vdots \\ v_{\mu_i}^g \end{bmatrix} \quad (4.32)$$

Kovariansmatrisene for henholdsvis system- og målestøy er definert som

$$Q = \begin{bmatrix} Q_\beta & 0 \\ 0 & Q_{\omega_m} \end{bmatrix} \quad (4.33)$$

$$R = [Q_\mu] \quad (4.34)$$

$Q_\beta$  er kovariansmatrisa til RRW og  $Q_{\omega_m}$  er kovariansmatrisa til gyromatrisas ARW. I kovariansmatrisa  $Q$  kan man legge inn en eventuell korrelasjon mellom gyroene i gyromatrisa. Hvis matrisa  $Q$  inneholder ikke-diagonale elementer, er det tegn på en statistisk sammenheng mellom gyroene. Eksisterer det derimot ingen ikke-diagonale elementer, er det ingen korrelasjon mellom dem.

### 4.2.2 Implementering av gyromatrise i MATLAB

Framgangsmåten her gjelder for 3 gyroer, men MATLAB-koden for både 3 og 6 gyroer ligger vedlagt. Vi bruker matrisene fra avsnitt 4.2.1, i tillegg til:

$$P_0 = \begin{bmatrix} \sigma_{\beta_1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{\beta_2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\beta_3}^2 & 0 \\ 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix} \quad (4.35)$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (4.36)$$

Gyromatrisa i denne rapporten benytter seg av målinger som er lagret på forhånd, ved kjøring av flere gyroer samtidig. De lastes så inn i for-løkkka som vist i pseudokoden under.

#### Pseudokode

`% Diskretisering:`

`$\Phi = e^{FT_s};$`

`$\Gamma = kp2dpGa(F, G, Q, T_s);$`

`$x_0 \sim \mathcal{N}(0, \hat{P}_0);$`

`$v_{\omega_m,0} \sim \mathcal{N}(0, Q_0);$`

`% Simulering:`

`for  $k = 1$  to  $N$`

`$\underline{v}_k = [v_{\beta_1,k}^g; v_{\beta_2,k}^g; v_{\beta_3,k}^g; n_{\omega_k}];$`

`$\omega_k = \Phi \underline{x}_k + \Gamma \underline{v}_k;$`

`$\underline{z}_k = H \underline{x}_k + B \underline{\omega}_k;$`

`end`

## 4.2. Gyromatrise

---

```
% Kalmanfilter:

for k = 1 to N
   $\underline{w}_k = [v_{\mu_{1,k}}^g; v_{\mu_{2,k}}^g; v_{\mu_{3,k}}^g]$ 
  %Måleoppdatering:
   $K_k = \bar{P}_k H_k^T (H \bar{P}_k H_k^T + R_k)^{-1}$ ;
   $\hat{P}_k = (I - K_k H_k) \bar{P}_k$ ;
   $z_k = H x_k + B w_k$ ;
   $\hat{x}_k = \bar{x} + K_k (z_k - H \bar{x}_k)$ ;
  %Tidsoppdatering:
   $\bar{x}_{k+1} = \Phi \hat{x}_k + \Lambda u_k$ ;
   $\bar{P}_{k+1} = \Phi \hat{P}_k \Phi^T + \Gamma Q \Gamma^T$ 
end
```

# Kapittel 5

## Resultater

### 5.1 Generelt om simuleringene

I dette kapitlet vil simuleringene og resultatene med figurer bli presentert. Resultatene er delt opp i to hoveddeler, der gyromatriser med henholdsvis 3 og 6 gyroer er simulert. Til slutt i dette kapitlet oppsummeres resultatene. I alle simuleringene nedenfor er det MEMS-gyroskopet STIM202 (som ble beskrevet i avsnitt 4.1.4) som er simulert. Av praktiske årsaker (tidsbruk) er samplingsfrekvensen satt ned fra den reelle frekvensen på 1000Hz, til 50Hz (det vil si at samplingstiden  $T_s$  er endret fra 0.001 til 0.02). Dette fører til at gyroen i simuleringensmodellen presterer litt dårligere enn reelle målinger fra en STIM202. For å gjøre simuleringensresultatene så oversiktlige som mulige kan vi dele opp simuleringene i følgende fire deler:

- Tilfelle 1: Gyromatrise med kombinerte målinger fra 3 gyroer og kalmanfilter, uten korrelasjon
- Tilfelle 2: Gyromatrise med kombinerte målinger fra 3 gyroer og kalmanfilter, med korrelasjon
- Tilfelle 3: Gyromatrise med kombinerte målinger fra 6 gyroer og kalmanfilter, uten korrelasjon
- Tilfelle 4: Gyromatrise med kombinerte målinger fra 6 gyroer og kalmanfilter, med korrelasjon

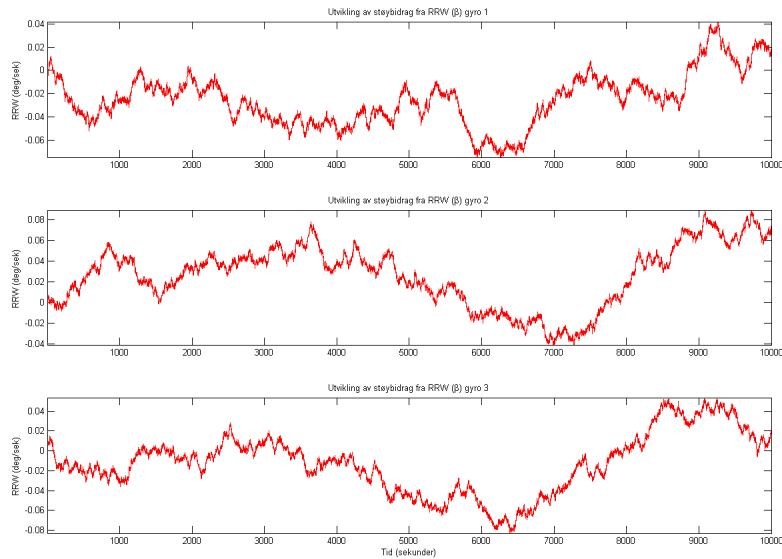
For hver gyromatrise som er simulert er gyromatrisas Allanvarians beregnet, slik at man kan sammenlikne mot allanvariansen til målingene fra en enkel gyro. På denne måten får man et inntrykk av hvor mye bedre gyromatrisa

presterer. Sammen med plot av vinkelhastighetsmålingene gir dette en god oversikt over forbedringene gyromatrisa gir. Felles for samtlige simuleringer er som sagt at gyroene ligger helt i ro, det vil si at sann vinkelhastighet  $\omega^g = 0 \text{ deg/s}$ , og det benyttes Kalmanfilter på vinkelhastighetsestimaterne fra gyromatrisene. Nærmere beskrivelse av innstillingene for hvert tilfelle er forklart under det enkelte avsnitt.

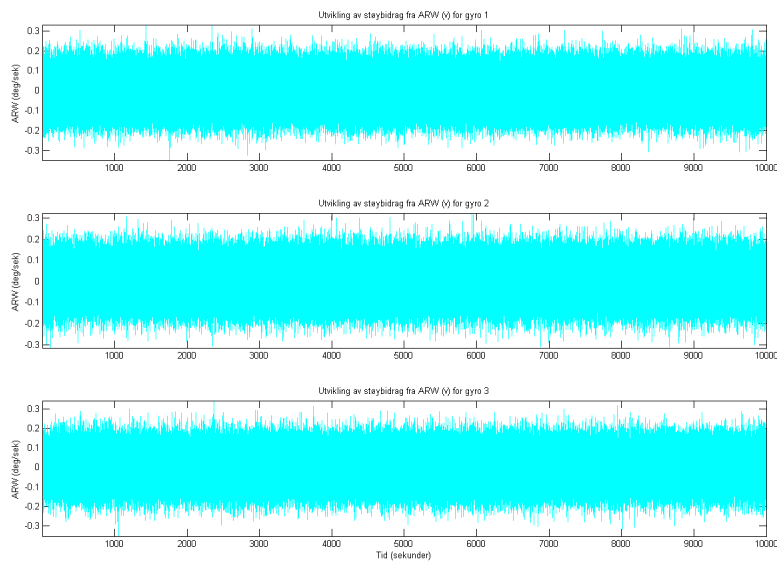
## 5.2 3 gyroer

Først simulerer vi 3 enkeltgyroer hver for seg i 10000 sekunder (cirka 2.75 timer). Hvis vi ser på figurene 5.1, 5.2 og 5.3 så viser disse hvordan støyen utvikler seg for hver av dem. Samlet er det disse støykildene som driver vinkelhastigheten  $\tilde{\omega}^g$ , og vi kan se vinkelhastighetsplottet for alle gyroene på figur 5.4. En kan her se at vinkelhastighetene til alle de tre gyroene gir et støyete signal omtrent rundt  $0 \text{ deg/s}$ , med et amplitude på  $\approx \pm 0,2$ . Som vi kan lese av databladet (som ligger vedlagt, se vedlegg A), er STIM202 oppgitt med en bias instability som gir drift på omtrent  $0,000139 \text{ deg/s}$ , samt en ARW på cirka  $0,0033 \text{ deg/s}$ . Gyroen vi tar utgangspunkt i presterer som nevnt litt dårligere, og plot av Allanvariansen kan vi se på figur 5.5. Vi leser av:  $\text{ARW} \approx 0,025 \text{ deg}/\sqrt{s}$  og  $\text{bias instability} \approx 0,008 \text{ deg/s}$ .

Det er målingene fra disse enkeltgyroene som blir brukt i de to neste avsnittene, der prestasjonsevnen til gyromatrisa testes.

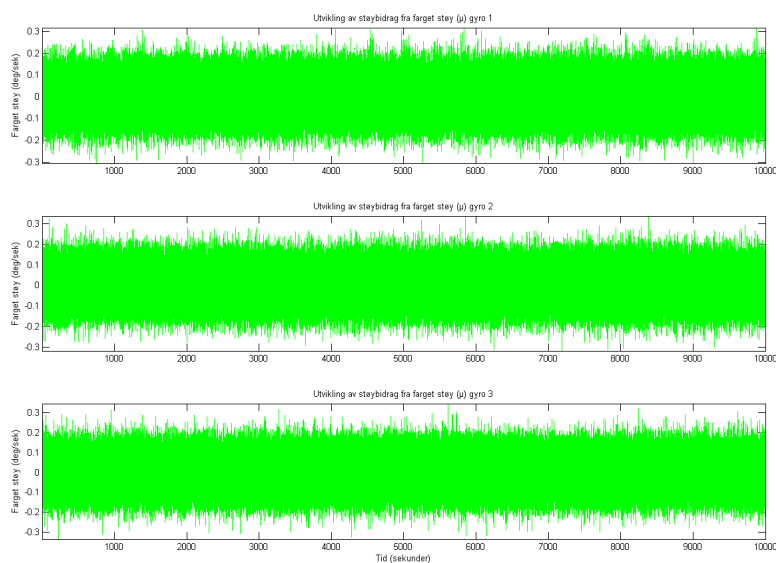


Figur 5.1: Utvikling av RRW ( $\beta^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder.

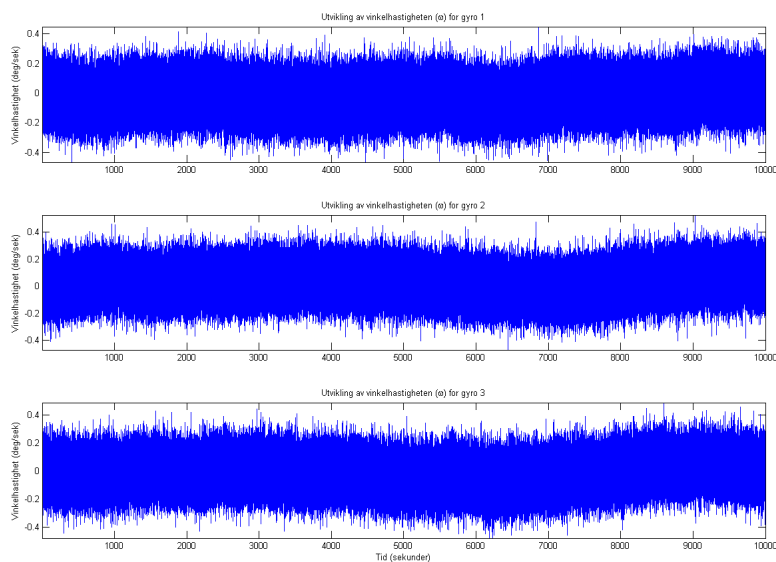


Figur 5.2: Utvikling av ARW ( $v_\omega^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder.

## 5.2. 3 gyroer

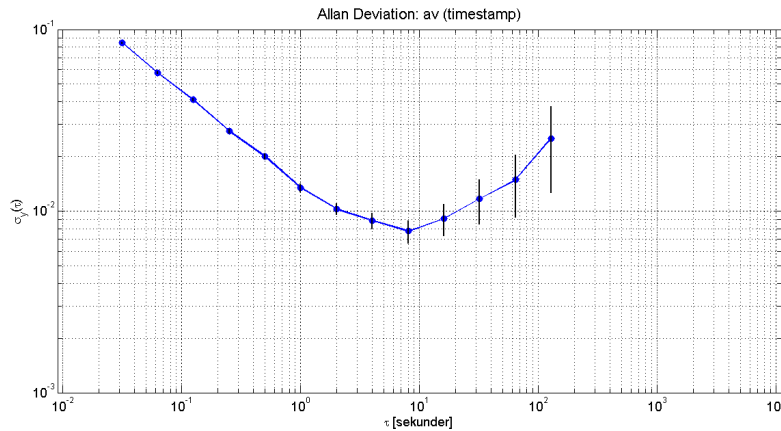


Figur 5.3: Utvikling av farget støy ( $\mu^g$ ) for gyro 1, 2 og 3. Simulert i 10000 sekunder.



Figur 5.4: Vinkelhastighet ( $\tilde{\omega}^g$ ) for 3 individuelle gyroer som ligger i ro. Simulert i 10000 sekunder.





Figur 5.5: Allanvariansen for gyro 1.

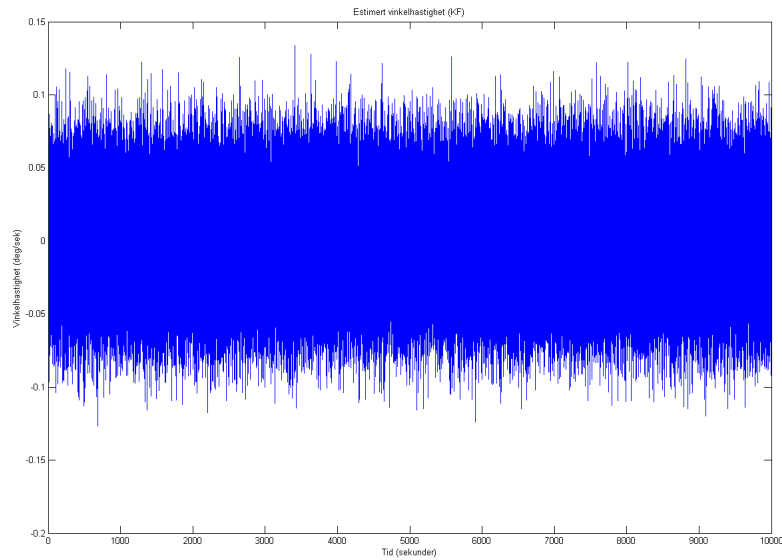
### 5.2.1 Tilfelle 1: Gyromatrise med 3 gyroer uten korrelasjon

Her ser vi på hvordan gyromatrisa presterer uten at det er korrelasjon mellom gyroene. Det vil si at kovariansmatrisa  $Q$  kun inneholder diagonale elementer, og vi modellerer den som identitetsmatrisa  $I$ :

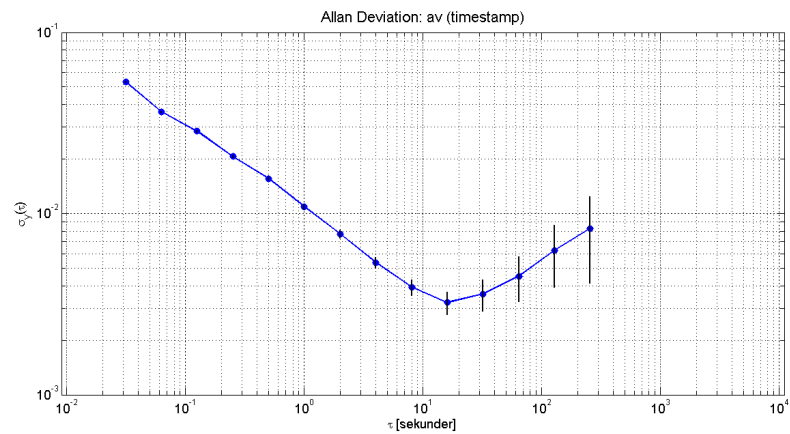
$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Resultatet av simuleringen ser vi på figur 5.6, der den estimerte vinkelhastigheten fra gyromatrisa er plottet. Allanvariansen er vist på figur 5.7, og vi leser av:  $ARW \approx 0,025 \text{ deg}/\sqrt{s}$ ,  $\text{bias instability} \approx 0,008 \text{ deg}/s$ . Målt vinkelhastighet  $\tilde{\omega}^g$  i det første tilfellet ligger rundt  $0 \text{ deg}/s$  med omtrent  $\pm 0,75 \text{ deg}/s$  som amplitude.

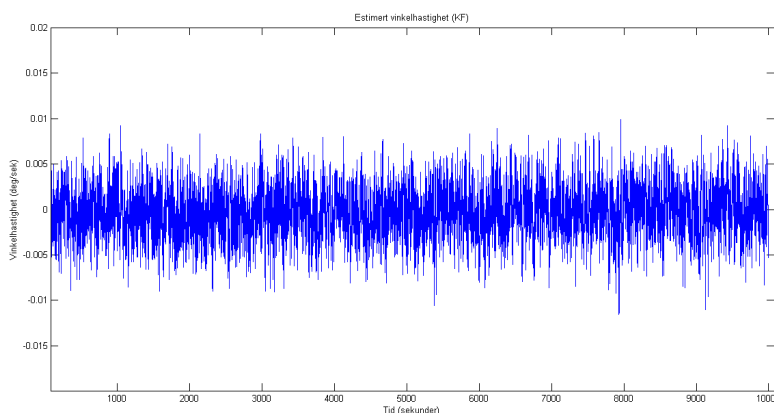
## 5.2. 3 gyroer



Figur 5.6: Tilfelle 1: Estimert vinkelhastighet fra gyromatrise der målinger fra 3 gyroer er kombinert, men uten korrelasjon. Simulert i 10000 sekunder.



Figur 5.7: Tilfelle 1: Allanvariansen for gyromatrise der målinger fra 3 gyroer er kombinert, men uten korrelasjon.



Figur 5.8: Tilfelle 2: Estimert vinkelhastighet fra gyromatrise der målinger fra 3 gyroer er kombinert, med korrelasjon. Simulert i 10000 sekunder.

### 5.2.2 Tilfelle 2: Gyromatrise med 3 gyroer med korrelasjon

I tilfelle 2 skal vi se hvordan gyromatrisa presterer **med** korrelasjon. Det viser seg at en korrelasjonskoeffisient på cirka  $-0,33$  er den mest hensiktsmessige, og vi setter opp koveriansmatrisa:

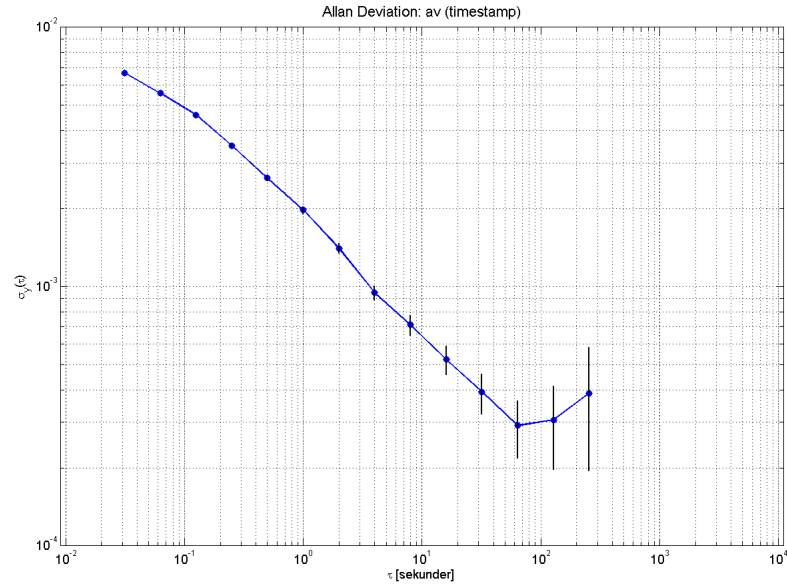
$$Q = \begin{bmatrix} 0.00189 & -0.0000163 & -0.0000163 & -0.0000163 \\ -0.0000163 & 0.00189 & -0.0000163 & -0.0000163 \\ -0.0000163 & -0.0000163 & 0.00189 & -0.0000163 \\ -0.0000163 & -0.0000163 & -0.0000163 & 0.0189 \end{bmatrix} \quad (5.2)$$

Vinkelhastighetsmålingene fra dette tilfellet kan vi se på figur 5.8, mens Allanvariansen er vist på figur 5.9. Den målte vinkelhastigheten  $\tilde{\omega}^g$  varierer med  $\pm 0,005$  deg/s og har middelerdi 0 deg/s. Hvis vi leser av på Allanvariansen ser vi at  $ARW \approx 0,002$  deg/ $\sqrt{s}$  og bias instability  $\approx 0,00028$  deg/s.

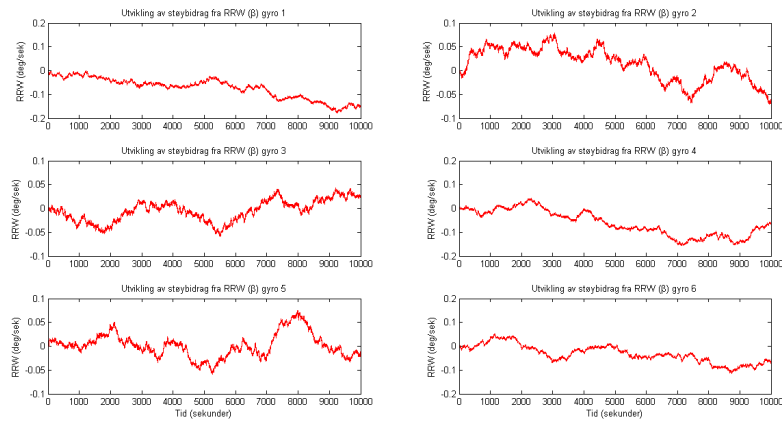
## 5.3 6 gyroer

Oppsettet her er helt likt som i simuleringene i avsnitt 5.2, forskjellen er at vi ønsker å se hva slags påvirkning det har på gyromatrisa hvis vi øker antall gyroer. Vi doubler derfor antallet til 6, og vi lager en måleserie fra disse 6 gyroene. Figurene 5.10, 5.11 og 5.12 viser støyene for hver av dem, mens figur 5.13 viser vinkelhastigheten  $\tilde{\omega}^g$  for alle 6. Allanvariansen for en enkelt av gyroene blir tilsvarende den som ble vist på figur 5.5.

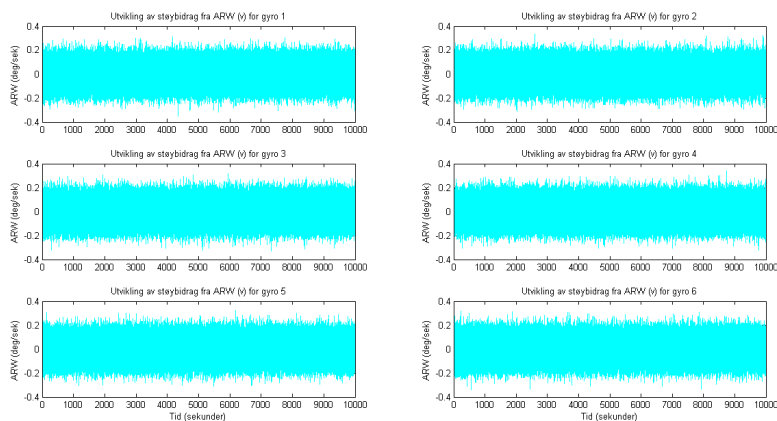
### 5.3. 6 gyroer



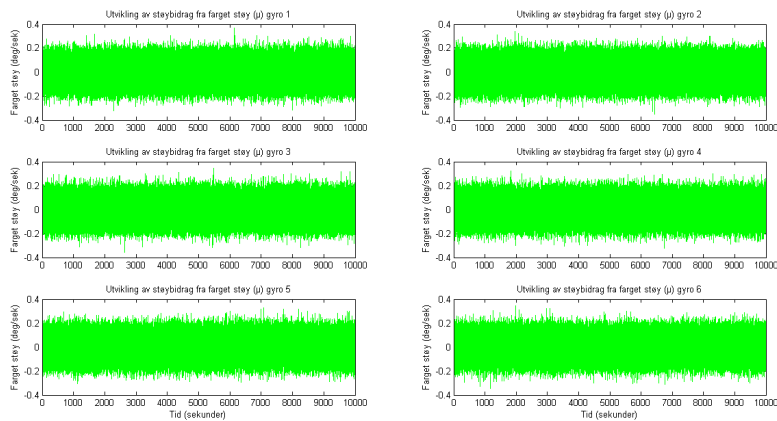
Figur 5.9: Tilfelle 1: Allanvariansen for gyromatrise der målinger fra 3 gyroer er kombinert, med korrelasjon.



Figur 5.10: Utvikling av RRW ( $\beta^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder.

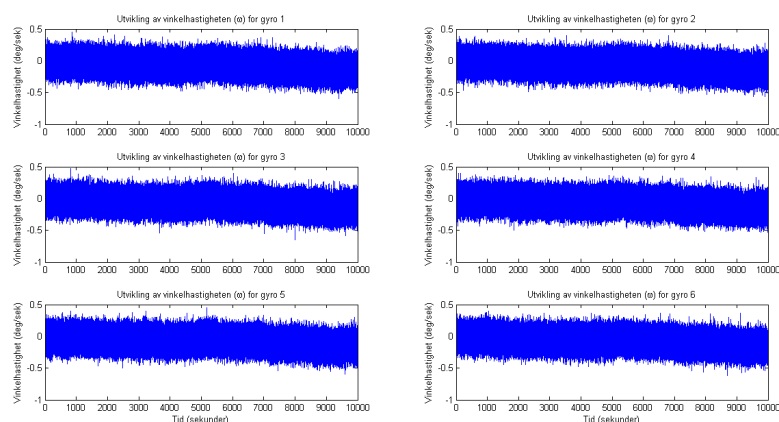


Figur 5.11: Utvikling av ARW ( $v_{\omega}^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder.



Figur 5.12: Utvikling av farget støy ( $\mu^g$ ) for gyro 1, 2, 3, 4, 5 og 6. Simulert i 10000 sekunder.

### 5.3. 6 gyroer



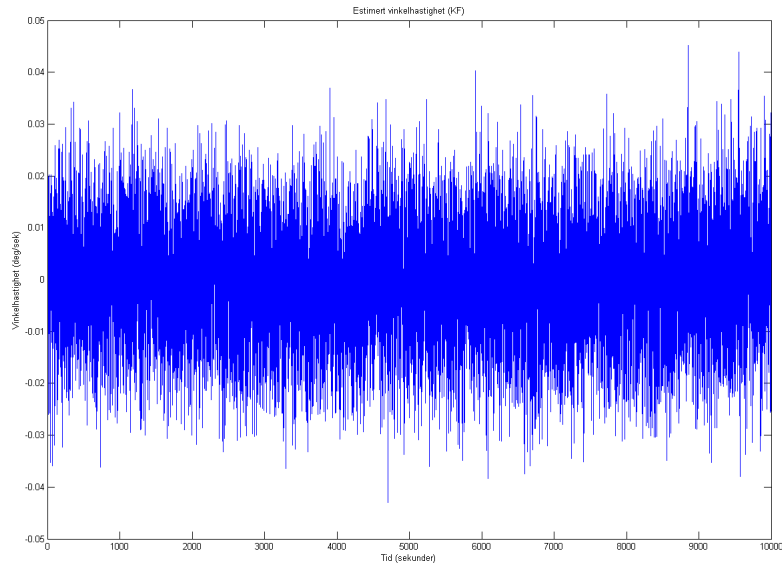
Figur 5.13: Vinkelhastighet ( $\tilde{\omega}^g$ ) for 6 individuelle gyroer som ligger i ro. Simulert i 10000 sekunder.

#### 5.3.1 Tilfelle 3: Gyromatrise med 6 gyroer uten korrelasjon

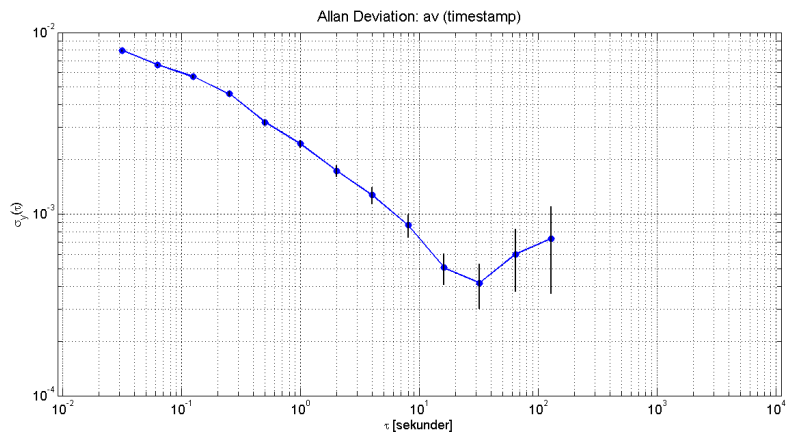
Ved å kombinere målingene fra avsnitt 5.3 i gyromatrisa, kan vi se hva slags forbedringer dette kan gi. Som i tilfellet med 3 gyroer ønsker vi først å se på tilfellet uten korrelasjon, og vi modellerer igjen kovariansmatrisa som identitetsmatrisa  $I$ :

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

Resultatet kan vi se i form av estimert vinkelhastighet  $\tilde{\omega}^g$  på figur 5.14 og Allanvariansen på figur 5.15. Plottet av Allanvariansen viser at vi i dette tilfellet får en ARW på  $\approx 0,0025 \text{ deg}/\sqrt{s}$  og en bias instability på  $\approx 0,00041 \text{ deg}/s$ . Målt vinkelhastighet  $\tilde{\omega}^g$  er et støyfylt signal som ligger rundt  $0 \text{ deg}/s$ , med en amplitude på  $\pm 0,02 \text{ deg}/s$ .

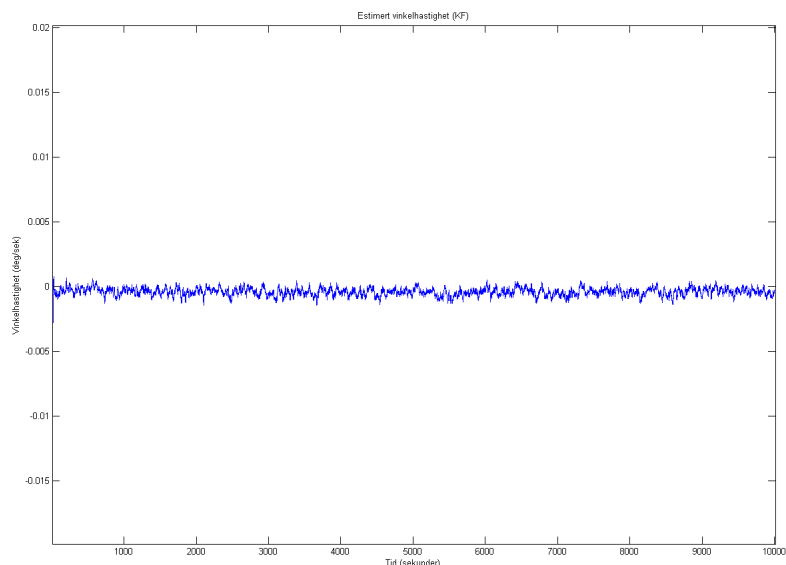


Figur 5.14: Tilfelle 3: Estimert vinkelhastighet fra gyromatrise der målinger fra 6 gyroer er kombinert, men uten korrelasjon. Simulert i 10000 sekunder.



Figur 5.15: Tilfelle 3: Allanvariansen for gyromatrise der målinger fra 6 gyroer er kombinert, men uten korrelasjon.

### 5.3. 6 gyroer



Figur 5.16: Tilfelle 4: Estimert vinkelhastighet fra gyromatrise der målinger fra 6 gyroer er kombinert, med korrelasjon. Simulert i 10000 sekunder.

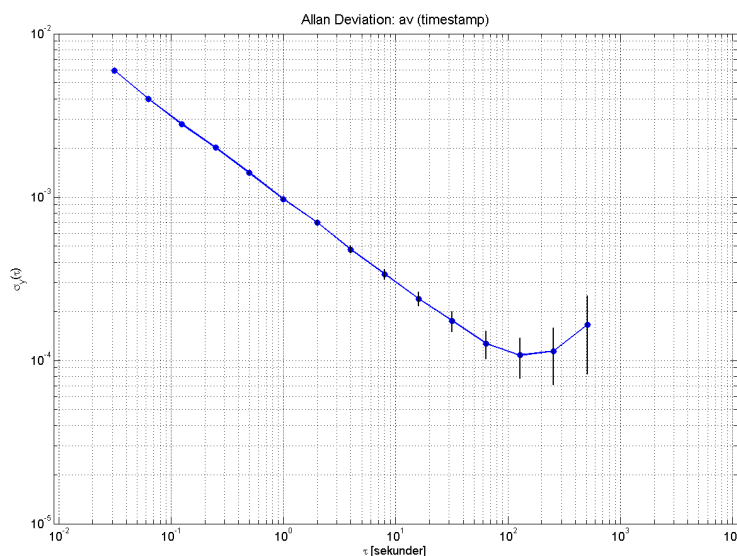
#### 5.3.2 Tilfelle 4: Gyromatrise med 6 gyroer med korrelasjon

Det siste tilfellet vi ser på er tilfelle 4: her kombinerer vi målinger fra 6 gyroer, og vi har korrelasjon. Her er det mest hensiktsmessig å ha en korrelasjonskoeffisient på omtrent  $-0,1667$ , og kovariansmatrisa  $Q$  blir her:

$$Q = \begin{bmatrix} 0.00189 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 \\ -0.0000163 & 0.00189 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 \\ -0.0000163 & -0.0000163 & 0.00189 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 \\ -0.0000163 & -0.0000163 & -0.0000163 & 0.00189 & -0.0000163 & -0.0000163 & -0.0000163 \\ -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & 0.00189 & -0.0000163 & -0.0000163 \\ -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & 0.00189 & -0.0000163 \\ -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & -0.0000163 & 0.0189 \end{bmatrix} \quad (5.4)$$

Hvis vi ser på figur 5.16 viser denne plot av estimert vinkelhastighet  $\tilde{\omega}^g$ , med middelerdi  $0 \text{ deg/s}$  og som varierer med  $\pm 0,0005 \text{ deg/s}$ . På figur 5.17 kan vi se Allanvariansen. Vi leser av og ser at  $ARW \approx 0,001 \text{ deg}/\sqrt{s}$  og bias instability  $\approx 0,00015 \text{ deg/s}$ .





Figur 5.17: Tilfelle 3: Allanvariansen for gyromatrise der målinger fra 6 gyroer er kombinert, med korrelasjon.

## 5.4 Oppsummering

Det er i dette kapittelet sett på 4 forskjellige tilfeller som har inneholdt forskjellig antall gyroer med og uten korrelasjon. Resultatene er presentert i figurer, og disse har blitt kommentert underveis. Det vil bli diskutert og konkludert i neste kapittel.

Tabell 5.1 er ment som en oppsummering, og viser parametrene fra de respektive simuleringene, slik at man enkelt kan sammenlikne ytelsen til hver av dem.

<i>Simulering</i>	<i>ARW</i>	<i>Bias instability</i>
Enkeltgyro	0,025 deg / $\sqrt{s}$	0,008 deg / s
Tilfelle 1	0,02 deg / $\sqrt{s}$	0,0033 deg / s
Tilfelle 2	0,002 deg / $\sqrt{s}$	0,00028 deg / s
Tilfelle 3	0,0025 deg / $\sqrt{s}$	0,00041 deg / s
Tilfelle 4	0,001 deg / $\sqrt{s}$	0,00015 deg / s

Tabell 5.1: Oppsummering av simuleringresultater

#### 5.4. Oppsummering

---

I tabellen er “Enkeltgyro” den gyroen vi modellerte med basis i STIM202. Det vil si en gyro med tilsvarende parametre som STIM202, bortsett fra nedsatt samplingsfrekvens (fra  $1000Hz$  til  $50Hz$ ). Tilfellene 1, 2, 3 og 4 er de gyromatrisene som ble simulert i avsnittene 5.2.1, 5.2.2, 5.3.1 og 5.3.2.

# Kapittel 6

## Konklusjon

Resultatene i denne rapporten er med på å støtte opp under forskningen til blant annet Chang [3] og Bayard [5], og viser at forbedringspotensialet til moderne MEMS-gyroer, ved hjelp av matematiske metoder og estimeringsalgoritmer med gyromatriser, helt klart er til stede.

Tabell 5.1 i avsnitt 5.4 viser sammendraget av de viktigste resultatene som ble gjort. Hvis vi ser på utgangspunktet, så hadde den opprinnelige gyromodellen en ARW på  $0,025 \text{ deg}/\sqrt{s}$  og en bias instability på  $0,008 \text{ deg}/s$ . Det betyr at den var relativt forstyrret av målestøy, og i tillegg “vandret” den omtrent  $28,8 \text{ deg}/s$ , slik at den med tiden hadde hatt en relativt stor feil.

Simuleringsforsøkene viste interessante resultater, der vi fikk se at det var forbedringspotensiale både med og uten korrelasjon. Uten korrelasjon var forbedringene relativt gode. Her var det tydelig at det hadde noe å si hvor mange gyroer vi hadde målinger fra, for i forsøket med 3 gyroer uten korrelasjon (tilfelle 1) ble driften redusert fra  $0,008 \text{ deg}/s$  til  $0,0033 \text{ deg}/s$  og ARW ble noe redusert fra  $0,025/\sqrt{s}$  til  $0,02/\sqrt{s}$ . Med 6 gyroer uten korrelasjon (tilfelle 3) kunne vi se reduksjon i drift fra  $0,008 \text{ deg}/s$  til  $0,00041 \text{ deg}/s$ , mens ARW ble hele 10 ganger bedre, nemlig fra  $0,025 \text{ deg}/\sqrt{s}$  til  $0,0025 \text{ deg}/\sqrt{s}$ .

Men ikke uventet var det tilfellene *med* korrelasjon som ga de beste resultatene: på tilfellet med 3 gyroer (tilfelle 2) ga gyromatrisa en reduksjon i driften på nesten  $\frac{1}{30}$  av den opprinnelige, mens det for tilfellet med 6 gyroer og korrelasjon (tilfelle 4), ble reduksjon på over  $\frac{1}{50}$ .

Dette viser at det sannsynligvis er Kalmanfilteret som er den viktigste faktoren i gyromatrisa. Kalmanfilteret klarer å filtrere bort deler av støyen som gyroene blir utsatt for, slik at vinkelhastighetsestimaterne blir mer nøyaktige

---

og de presterer bedre sammen. I tillegg til at Kalmanfiltreringen hjelper i seg selv, er det også tydelig at korrelasjon mellom gyroene (i prosesstøyens kovariansmatrise) bidrar til ytterligere forbedringer.

Utfordringen her er at for å få korrelasjon mellom gyroene må man få produsert gyroene på en slik måte at man oppnår dette. Som det ble nevnt i innledningen må MEMS-gyroene plasseres tett sammen på en liten brikke, og de må måle den samme aksene. STIM202, som i dag er den mest høyteknologiske MEMS-gyroen til Sensoror, leveres jo i praksis allerede med 3 gyroer, selv om de måler de 3 aksene X, Y og Z som står ortogonalt på hverandre. Det vil si at veien til gyromatriser i praksis kanskje ikke er så lang, hvis man greier å overbevise elektronikkindustrien om at det er verdt å satse på dette.

# Kapittel 7

## Videre arbeid

Det er flere ting som kan gjøres videre rundt arbeidet med gyromatriser, blant annet:

**Pådrag** I denne rapporten er det kun blitt sett på tilfeller der gyroene ligger i ro. Det kunne vært interessant og testet hvordan gyromatrisa presterte ved forskjellige pådrag, noe som hadde gitt mer reelle scenarioer i forhold til implementering i for eksempel et treghetsnavigasjonssystem.

**TNS** Det kunne blitt sett på et system der målingene fra gyromatrisa ble kombinert med målinger fra akselerometre, slik at en fikk et *treghetsnavigasjonssystem*. I eksempelvis moderne mobiltefonteknologi kunne dette vært ønskelig, da fler og fler av de nye modellene i dag blir levert med både gyroer og akselerometre. Det er svært sannsynlig at bransjen ville tatt godt imot forbedringsmuligheter for de integrerte MEMS-baserte treghetsnavigasjonssystemene som er på markedet i dag.

**Nordsøking** Dette ble også nevnt i innledningen. De tradisjonelle gyrokompassene som blir brukt i dag er gjerne for store og tunge til at de enkelt kan fraktes på lik linje med vanlige magnetkompass. Derfor kunne bruk av gyromatriser for nordsøking vært en interessant studie.



# Bibliografi

- [1] D.W. Allan et. al. Fine-Tuning Time In The Space Age. *IEEE Spectrum*, 1998.
- [2] Arthur Gelb (editor). *Applied Optimal Estimation*. The M.I.T. Press, 1974.
- [3] H. Chang, L. Xue, W.i Qin, G. Yuan, and W. Yuan. An Integrated MEMS Gyroscope Array with Higher Accuracy Output. *Sensors, MDPI*, 2008.
- [4] Dan Simon. *Optimal State Estimation*. Wiley, 2006.
- [5] David S. Bayard and Scott R. Ploen. High accuracy inertial sensors from inexpensive components. *US Patent. US20030187623A1*, 2003.
- [6] Gunnar G. Løvås. *Statistikk for universiteter og høyskoler, 2. utgave*. Universitetsforlaget, 2004.
- [7] M. A. Hopcroft. <http://www.mathworks.com/matlabcentral/fileexchange/13246-allan> (25.05.2011). Matlabprogram for beregning av Allanvarians.
- [8] Jørn Grahn. Estimering av MEMS-gyroparametre. Master's thesis, Fysisk institutt, Universitetet i Oslo, 2011.
- [9] Naser El Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and Modeling of Inertial Sensors Using Allan Variance. *IEEE Transactions on Instrumentation and Measurement, vol. 57*, 2008.
- [10] Oddvar Hallingstad. Standardmodeller og Kalmanfilterlikninger. *UNIK4500 Stokastiske Systemer*, 2005.
- [11] Quang M. Lam, T. Hunt, P. Sanneman, and S. Underwood. Analysis and Design of a Fifteen State Stellar Inertial Attitude Determination System. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003.

## Bibliografi

---

- [12] Quang M. Lam, Thomas Wilson Jr., Ronald Contillo, and Darrin Buck. Enhancing MEMS Sensors Accuracy Via Random Noise Characterization and Calibration. *Proceedings of SPIE*, 2004.
- [13] Sensoror. <http://www.sensoror.com/company/about-sensoror/timeline.aspx> (11.05.2011). About Sensoror - Timeline.
- [14] Sensoror. <http://www.sensoror.com/gyro-products/gyro-modules/ultra-high-performance/stim202.aspx> (11.05.2011). STIM202.
- [15] Sensoror. [http://www.sensoror.com/media/70534/datasheet\\_stim202\\_ts1439\\_r8.pdf](http://www.sensoror.com/media/70534/datasheet_stim202_ts1439_r8.pdf) (26.05.2011). Data Sheet STIM202 Multi-Axis Gyro Module.
- [16] The Institute of Electrical and Electronics Engineers. IEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros. 1998.
- [17] D.H. Titterton and J.L. Weston. *Strapdown Inertial Navigation Technology*. AIAA, 2nd edition, 2004.
- [18] VectorNav. [http://www.vectornav.com/index.php?option=com\\_content&view=article&id=26&Itemid=11](http://www.vectornav.com/index.php?option=com_content&view=article&id=26&Itemid=11) (25.05.2011). Gyro Sensor Model.



# Tillegg A

## Datablad: STIM202

På de to neste sidene ligger oppsummering av databladet til gyroen STIM202 vedlagt. For fullstendig dokumentasjon, se [15].

# ULTRA-HIGH PERFORMANCE GYRO MODULE



STIM202

PRODUCT BRIEF

## FEATURES

- $\pm 400^\circ/\text{s}$  input range
- 1, 2 or 3 axis capability
- Electronic axis alignment
- $0.5^\circ/\text{h}$  in-run bias stability
- $0.2^\circ/\sqrt{\text{h}}$  angular random walk
- $18^\circ/\text{h/g}$  linear acceleration effect
- 1500g shock capability
- 55g module weight



## DESCRIPTION

STIM is a cluster of 1, 2 or 3 high accurate MEMS-based gyros housed in a miniature package. Each axis is factory calibrated for bias and sensitivity, and compensated for temperature effects to provide high-accuracy measurements.

For many applications the excellent performance of STIM202 will replace FOG's and improve system solution with respect to robustness, reliability, size/weight, power and cost. This is accomplished by combining the well proven Sensoror ButterflyGyro™ with full digital operation.

### Input range and output formats

The STIM202 full-scale angular rate input range is  $400^\circ/\text{s}$ . Selectable output formats are angular rate or increment angle.

### 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.

### Interface

The unit is powered by a single +5V supply and communicates via a Plug & Play high-level RS422 interface.

### Flexibility and self diagnostics

The use of a 32-bit RISC ARM microcontroller provides flexibility in configuration, e.g. for choices of output unit, sampling frequency, LP filter cut-off frequency, RS422 bit rate and line termination ON/OFF. The gyro module provides a diagnostic function, that will flag any type of errors in the system. For more advanced users, the STIM202 may be set in Service Mode, where all configuration parameters may be intermediately or permanently changed by overwriting current settings in the flash memory.

Service Mode also provides ability to perform single measurements and to access more diagnostic information.

### Evaluation kits

Evaluation kits are available, supporting initial testing and device configuration. The starter kits are small, flexible and communicate via USB.

Additional PCI cards are offered for more extensive testing or characterization. The RS422 connection of the PCI card is directly compatible with the STIM202 communication cable included in the starter kit.

### Applications

The STIM202 is designed to replace FOGs in general, and enables new areas of use.

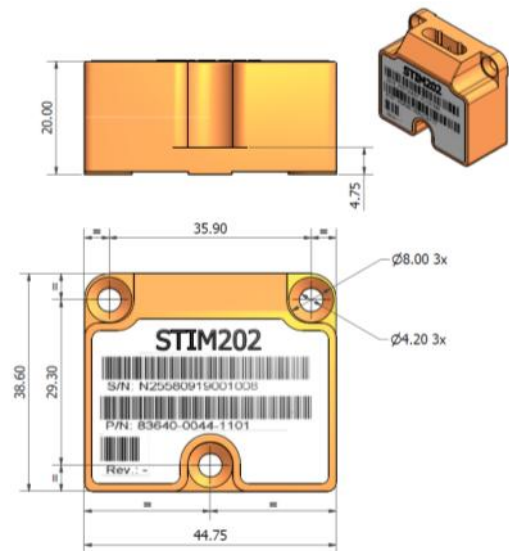
Typical applications for the product is attitude heading reference systems (AHRs), flight recorders, platform stabilization (e.g. antennas, cameras, gimbals, etc.), high performance industrial applications, unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs), space applications, etc.

SPECIFICATIONS

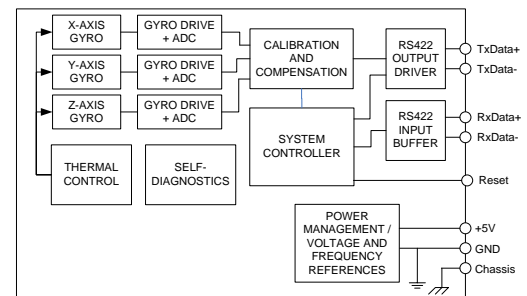
Parameter	Min	Nom	Max	Unit
Weight		55		g
Input range		±400		°/s
Resolution		24		bit
Operating temperature	-40		85	°C
Power supply	4.5	5.0	5.5	V
Supply current		200		mA
Start-up time			10	s
Sampling frequency			1000	SPS
Storage temperature	-50		90	°C
Dynamic overload			5000	°/s
Mechanical shock			1500	g
In-run bias stability		0.5		°/h
Angular random walk		0.2		°/√h
Bandwidth (-3dB)			262	Hz
Non-linearity (BSL over +/- 200 °/s)			200	ppm
Scale Factor accuracy		±0.2		%
Bias temperature accuracy (1σ)		±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)		125		Ω
<b>RESET PIN (NRST)</b>				
Logic levels	CMOS and TTL compatible			
Minimum hold time for reset		5		ms

MECHANICAL DIMENSIONS

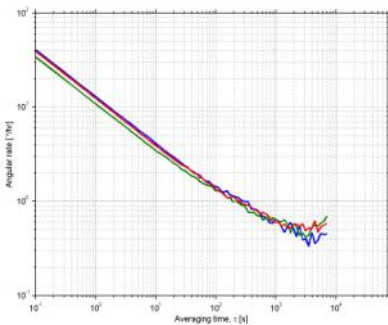
All dimensions in mm.



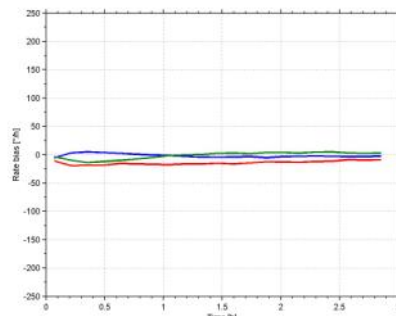
FUNCTIONAL BLOCK DIAGRAM



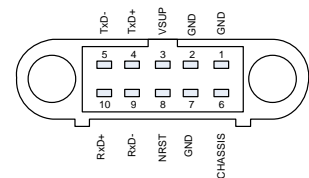
ALLAN VARIANCE



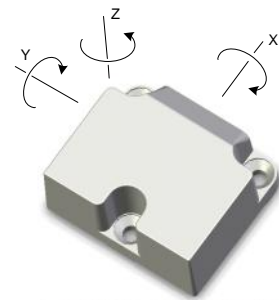
BIAS STABILITY



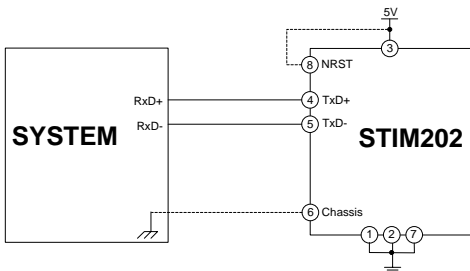
PIN OUT



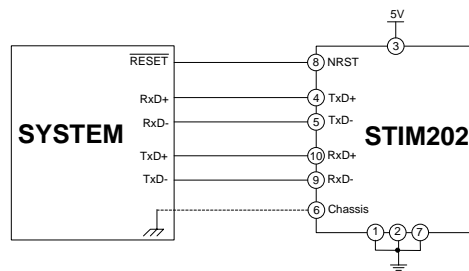
AXIS DEFINITIONS



ELECTRICAL CONNECTIONS



TRANSMIT ONLY



FULL FUNCTION



# Tillegg B

## Matlab

### B.1 kp2dpGa.m

---

```
function [Ga]=kp2dpGa(F,G,Qb,d)
```

```
% Finner verdien av Gamma
```

```
[S]=kp2dpS(F,G,Qb,d);
```

```
[L,D]=ldl(S);
```

```
Ga=L*D^(1/2);
```

---

kp2dpGa.m

### B.2 kp2dpS.m

---

```
function [S]=kp2dpS(F,G,Qb,d)
```

```
% Finner verdien av S (Gamma*Q*Gamma^T) for å kunne trekke ut Gamma
```

```
syms tau;
```

```
temp=int(expm(F*(d-tau))*G*Qb*G'*(expm(F*(d-tau))'),tau,0,d);
```

```
S=subs(temp);
```

---

kp2dpS.m

### B.3 Feilmodell3g.m

---

```
%% PROGRAM FOR Å SIMULERE FEILMODELLEN
```

```
% FOR 3 GYROER
```

### B.3. Feilmodell3g.m

---

```
%  $x' = F'x' + G'v'$ 
% der  $x'$  består av  $m_y$  (farget støy) og  $\beta$  (RRW)

Tb1=0.02; % Tidskonstant (korrelasjonstid)

Ts=0.02; % Samplingstid

N=500000; % Tidsskritt

% (alle  $m$ -ene i påfølgende variabler betegner "merket": ' ')
% Matriser for tilstandsrommodell

Fm=[0 0;
    0 -(1/Tb1)];
Gm=[1 0;
    0 1];
Lm=[0 0]';

H=[1 1];

P0est=[1e-6 0;
    0 0.0049];

Qbm=[1e-6 0;
    0 2*P0est(2,2)/Tb1]; % prosesstøyens spektraltetthet

Rd=0.0049;

% Diskretisering:

[Fim, Lam]=c2d(Fm, Lm, Ts);

[Gam]=kp2dpGa(Fm, Gm, Qbm, Ts);

%% - GYROSKOP 1
% Initialverider 1:
xm1=chol(P0est, 'lower')*randn(2,1);
um1=0;

% Simulering 1
for k=1:N

    sek(1,k)=Ts*k;

    arw1(:,k)=sqrt(Rd)*randn;
    z1(:,k)=H*xm1(:,k)+arw1(k);
```

---

```

    xm1(:,k+1)=Fim*xm1(:,k)+Lam*um1+(Gam*randn(2,1)); %feilkilder

end
sek1=sek;
sek1(1,N+1)=Ts*(k+1);
%%
%% - GYROSKOP 2
% Initialverider 2:
xm2=chol(P0est,'lower')*randn(2,1);
um2=0;

% Simulering 2
for k=1:N

    arw2(:,k)=sqrt(Rd)*randn;
    z2(:,k)=H*xm2(:,k)+arw2(k);

    xm2(:,k+1)=Fim*xm2(:,k)+Lam*um2+(Gam*randn(2,1));

end
%%
%% - GYROSKOP 3
% Initialverider 3:
xm3=chol(P0est,'lower')*randn(2,1);
um3=0;

% Simulering 3
for k=1:N

    arw3(:,k)=sqrt(Rd)*randn;
    z3(:,k)=H*xm3(:,k)+arw3(k);

    xm3(:,k+1)=Fim*xm3(:,k)+Lam*um3+(Gam*randn(2,1));

end
%%
save Feilmodell3.mat

%% Plotterutine for å sammenlikne 3 gyroer simulert

figure(1)
subplot(3,1,1)
plot(sek,z1,'b')
axis tight
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro 1');
subplot(3,1,2)
plot(sek,z2,'b')

```

### B.3. Feilmodell3g.m

---

```
axis tight
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro 2');
subplot(3,1,3)
plot(sek,z3,'b')
axis tight
ylabel('Vinkelhastighet (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro 3');

figure (2)
subplot(3,1,1)
plot(sek,xm1(1,(1:N)), 'r')
axis tight
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW (beta) gyro 1');
subplot(3,1,2)
plot(sek,xm2(1,(1:N)), 'r')
axis tight
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW (beta) gyro 2');
subplot(3,1,3)
plot(sek,xm3(1,(1:N)), 'r')
axis tight
ylabel('RRW (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra RRW (beta) gyro 3');

figure (3)
subplot(3,1,1)
plot(sek,xm1(2,(1:N)), 'g')
axis tight
ylabel('Farget støy (deg/sek)')
TITLE('Utvikling av støybidrag fra farget støy (my) gyro 1');
subplot(3,1,2)
plot(sek,xm2(2,(1:N)), 'g')
axis tight
ylabel('Farget støy (deg/sek)')
TITLE('Utvikling av støybidrag fra farget støy (my) gyro 2');
subplot(3,1,3)
plot(sek,xm3(2,(1:N)), 'g')
axis tight
ylabel('Farget støy (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra farget støy (my) gyro 3');

figure (4)
subplot(3,1,1)
```



```

plot(sek , arw1 , 'c')
axis tight
ylabel( 'ARW_(deg/sek)')
TITLE('Utvikling_av_støybidrag_fra_ARW(v)_for_gyro_1');
subplot(3,1,2)
plot(sek , arw2 , 'c')
axis tight
ylabel( 'ARW_(deg/sek)')
TITLE('Utvikling_av_støybidrag_fra_ARW(v)_for_gyro_2');
subplot(3,1,3)
plot(sek , arw3 , 'c')
axis tight
ylabel( 'ARW_(deg/sek)')
xlabel( 'Tid_(sekunder)')
TITLE('Utvikling_av_støybidrag_fra_ARW(v)_for_gyro_3');

figure(5)
hold on
plot(sek , z1 , 'r')
plot(sek , z2 , 'c')
plot(sek , z3 , 'b')
axis tight
ylabel( 'Vinkelhastighet_(deg/sek)')
xlabel( 'Tid_(sekunder)')
hold off

%% Beregning av Allanvarians
t=0:0.02:N/50;

DATA.freq=z1;
DATA.time=t;
tau=[];
name='av';
verbose=2;

[retval , s , errorb , tau] = allan(DATA,tau,name,verbose);

%% Plotterutine for å vise utvikling for én gyro
%{
figure(1)
plot(sek , z1 , 'b')
xlabel( 'tid_(sekunder)')
ylabel( 'vinkelhastighet_(deg/sek)')
LEGEND('Utvikling_av_vinkelhastigheten_(omega)_for_STIM202');

figure(2)
plot(sek , xm1(1,(1:N)) , 'r')
xlabel( 'tid_(sekunder)')

```

## B.4. Gyromatrise3.m

---

```
ylabel('RRW_(deg/sek)')
LEGEND('Utvikling_av_støybidrag_fra_RRW_(beta)_for_STIM202');

figure(3)
plot(sek,xm1(2,(1:N)),'g')
xlabel('tid_(sekunder)')
ylabel('Farget_støy_(deg/sek)')
LEGEND('Utvikling_av_støybidrag_fra_farget_støy_(my)_for_STIM202');

figure(4)
plot(sek,arw1,'c');
xlabel('tid_(sekunder)')
ylabel('ARW_(deg/sek)')
LEGEND('Utvikling_av_støybidrag_fra_ARW_(v)_for_STIM202');
```

---

Feilmodell3g.m

## B.4 Gyromatrise3.m

---

```
clear all;
```

```
%% PROGRAM FOR Å SIMULERE GYROMATRISSE MED 3 GYROER
```

```
load Feilmodell3.mat
```

```
% Systemmatriser:
```

```
F=[0 0 0 0; 0 0 0 0; 0 0 0 0; 0 0 0 0];
G=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
H=[1 0 0 1; 0 1 0 1; 0 0 1 1];
B=[1 0 0; 0 1 0; 0 0 1];
L=[0 0 0 0]';
```

```
%%Q=eye(4); % Kovariansmatrise uten korrelasjon
```

```
Q=[0.00189 -0.0000163 -0.0000163 -0.0000163; % Kovariansmatrise med korrelasjo
-0.0000163 0.00189 -0.0000163 -0.0000163;
-0.0000163 -0.0000163 0.00189 -0.0000163;
-0.0000163 -0.0000163 -0.0000163 0.0189];
```

```
% Diskretisering:
```

```
[Fi,La]=c2d(F,L,Ts);
[Ga]=kp2dpGa(F,G,Q,Ts);
```

```

u=0;
R=0.0049;

P=diag([0.1 0.1 0.1 0.1]);

X0=chol(P, 'lower')*randn(4,1);
X=X0;

% Simulering av gyromatrise med hvit støy som "sann omega"
for k=1:N
    W=[xm1(1,k) xm2(1,k) xm3(1,k) 0.1*randn]';
    XS(:,k+1)=Fi*X+Ga*W;
end

% setter startverdi for prediktert kovarians
P0est=P;
Pp=Fi*P0est*Fi'+Ga*Q*Ga';

% setter startverdi for prediktert tilstand
Xp(:,1)=Fi*X0+La*u;

for k = 1:N
    V=[xm1(2,k) xm2(2,k) xm3(2,k)]';

    %—————KF—————
    %

    % MO
    K=Pp*H'*inv(H*Pp*H'+R);
    Pe=(eye(4)-K*H)*Pp;
    z(:,k)=H*XS(:,k)+B*V;
    Xe(:,k)=Xp(:,k)+K*(z(k)-H*Xp(:,k));

    %IO
    Xp(:,k+1)=Fi*Xe(:,k)+La*u;
    Pp=Fi*Pe*Fi'+Ga*Q*Ga';

    Xe(:,k)=Xp(:,end);
    Pe=Pp;

end

```

## B.5. Feilmodell6g.m

---

```
save Array3.mat

figure(8)
plot(sek,Xe(4,:), 'b');
xlabel('Tid (sekunder)')
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Estimert vinkelhastighet (KF)')

sluttid=N/50;
t=0:0.02:sluttid;

DATA.freq=Xe(4,:);
DATA.time=t;
tau=[];
name='av';
verbose=2;

[retval, s, errorb, tau] = allan(DATA,tau,name,verbose);
```

---

Gyromatrise3.m

## B.5 Feilmodell6g.m

---

```
%% PROGRAM FOR Å SIMULERE FEILMODELLEN
% FOR 6 GYROER
%  $x' = F'x' + G'v'$ 
% der  $x'$  består av  $m_y$  (farget støy) og  $\beta$  (RRW)

Tb1=0.02; % Tidskonstant (korrelasjonstid)

Ts=0.02; % Samplingstid

N=500000; % Tidsskritt

% (alle m-ene i påfølgende variabler betegner "merket": ')
% Matriser for tilstandsrommodell
Fm=[0 0;
    0 -(1/Tb1)];
Gm=[1 0;
    0 1];
Lm=[0 0]';

H=[1 1];
```

---

```

P0est=[1e-6 0;
       0 0.0049];

Qbm=[1e-6 0;
     0 2*P0est(2,2)/Tb1];% prosesstøyens spektraltetthet

Rd=0.0049;

% Diskretisering:
[Fim,Lam]=c2d(Fm,Lm,Ts);
[Gam]=kp2dpGa(Fm,Gm,Qbm,Ts);
%-----

%% - GYROSKOP 1
% Initialverider 1:
xm1=chol(P0est,'lower')*randn(2,1);
um1=0;

% Simulering 1
for k=1:N

    sek(1,k)=Ts*k;

    arw1(:,k)=sqrt(Rd)*randn;
    z1(:,k)=H*xm1(:,k)+arw1(k);

    xm1(:,k+1)=Fim*xm1(:,k)+Lam*um1+(Gam*randn(2,1)); %feilkilder

end
%-----

sek1=sek;
sek1(1,N+1)=Ts*(k+1);

%% - GYROSKOP 2
% Initialverider 2:
xm2=chol(P0est,'lower')*randn(2,1);
um2=0;

% Simulering 2
for k=1:N

    arw2(:,k)=sqrt(Rd)*randn;
    z2(:,k)=H*xm1(:,k)+arw2(k);

    xm2(:,k+1)=Fim*xm2(:,k)+Lam*um2+(Gam*randn(2,1));

end

```

## B.5. Feilmodell6g.m

---

```
%%
```

```
%% - GYROSKOP 3
```

```
% Initialverider 3:
```

```
xm3=chol(P0est, 'lower')*randn(2,1);
```

```
um3=0;
```

```
% Simulering 3
```

```
for k=1:N
```

```
    arw3(:,k)=sqrt(Rd)*randn;
```

```
    z3(:,k)=H*xm1(:,k)+arw3(k);
```

```
    xm3(:,k+1)=Fim*xm3(:,k)+Lam*um3+(Gam*randn(2,1));
```

```
end
```

```
%%
```

```
%% - GYROSKOP 4
```

```
% Initialverider 4:
```

```
xm4=chol(P0est, 'lower')*randn(2,1);
```

```
um4=0;
```

```
% Simulering 4
```

```
for k=1:N
```

```
    arw4(:,k)=sqrt(Rd)*randn;
```

```
    z4(:,k)=H*xm1(:,k)+arw4(k);
```

```
    xm4(:,k+1)=Fim*xm4(:,k)+Lam*um4+(Gam*randn(2,1));
```

```
end
```

```
%%
```

```
%% - GYROSKOP 5 - %
```

```
% Initialverider 5:
```

```
xm5=chol(P0est, 'lower')*randn(2,1);
```

```
um5=0;
```

```
%xs5=zeros(1,N+1);
```

```
% Simulering 5
```

```
for k=1:N
```

```
    arw5(:,k)=sqrt(Rd)*randn;
```

```
    z5(:,k)=H*xm1(:,k)+arw5(k);
```

---

```

    xm5(:,k+1)=Fim*xm5(:,k)+Lam*um5+(Gam*randn(2,1));

end
%%

%% – GYROSKOP 6

xm6=chol(P0est,'lower')*randn(2,1);
um6=0;

% Simulering 6
for k=1:N

    arw6(:,k)=sqrt(Rd)*randn;
    z6(:,k)=H*xm1(:,k)+arw6(k);

    xm6(:,k+1)=Fim*xm6(:,k)+Lam*um6+(Gam*randn(2,1));
    %xut6(k)=xs6(k)+xm6(1,k)+xm6(2,k)+(0.06*randn);
end
%%

save Feilmodell6.mat

%% Plotterutine for å sammenlikne 6 gyroer simulert

figure(1)
subplot(3,2,1)
plot(sek,z1,'b')
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro1');
subplot(3,2,2)
plot(sek,z2,'b')
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro2');
subplot(3,2,3)
plot(sek,z3,'b')
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro3');
subplot(3,2,4)
plot(sek,z4,'b')
ylabel('Vinkelhastighet (deg/sek)')
TITLE('Utvikling av vinkelhastigheten (omega) for gyro4');
subplot(3,2,5)
plot(sek,z5,'b')
ylabel('Vinkelhastighet (deg/sek)')

```

## B.5. Feilmodell6g.m

---

```
xlabel('Tid (sekunder)')
TITLE('Utvikling av vinkelhastigheten ( $\omega$ ) for gyro 5');
subplot(3,2,6)
plot(sek, z6, 'b')
ylabel('Vinkelhastighet (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av vinkelhastigheten ( $\omega$ ) for gyro 6');

figure (2)
subplot(3,2,1)
plot(sek, xm1(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 1');
subplot(3,2,2)
plot(sek, xm2(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 2');
subplot(3,2,3)
plot(sek, xm3(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 3');
subplot(3,2,4)
plot(sek, xm4(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 4');
subplot(3,2,5)
plot(sek, xm5(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 5');
subplot(3,2,6)
plot(sek, xm6(1,(1:N)), 'r')
ylabel('RRW (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra RRW ( $\beta$ ) gyro 6');

figure (3)
subplot(3,2,1)
plot(sek, xm1(2,(1:N)), 'g')
ylabel('Farget støy (deg/sek)')
TITLE('Utvikling av støybidrag fra farget støy (my) gyro 1');
subplot(3,2,2)
plot(sek, xm2(2,(1:N)), 'g')
ylabel('Farget støy (deg/sek)')
TITLE('Utvikling av støybidrag fra farget støy (my) gyro 2');
subplot(3,2,3)
plot(sek, xm3(2,(1:N)), 'g')
```



```

ylabel('Fargetstøy (deg/sek)')
TITLE('Utvikling av støybidrag fra fargetstøy (my) gyro3');
subplot(3,2,4)
plot(sek, xm4(2, (1:N)), 'g')
ylabel('Fargetstøy (deg/sek)')
TITLE('Utvikling av støybidrag fra fargetstøy (my) gyro4');
subplot(3,2,5)
plot(sek, xm5(2, (1:N)), 'g')
ylabel('Fargetstøy (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra fargetstøy (my) gyro5');
subplot(3,2,6)
plot(sek, xm6(2, (1:N)), 'g')
ylabel('Fargetstøy (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra fargetstøy (my) gyro6');

figure(4)
subplot(3,2,1)
plot(sek, arw1, 'c')
ylabel('ARW (deg/sek)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro1');
subplot(3,2,2)
plot(sek, arw2, 'c')
ylabel('ARW (deg/sek)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro2');
subplot(3,2,3)
plot(sek, arw3, 'c')
ylabel('ARW (deg/sek)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro3');
subplot(3,2,4)
plot(sek, arw4, 'c')
ylabel('ARW (deg/sek)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro4');
subplot(3,2,5)
plot(sek, arw5, 'c')
ylabel('ARW (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro5');
subplot(3,2,6)
plot(sek, arw6, 'c')
ylabel('ARW (deg/sek)')
xlabel('Tid (sekunder)')
TITLE('Utvikling av støybidrag fra ARW(v) for gyro6');

figure(5)
hold on
plot(sek, z1, 'r')
plot(sek, z2, 'c')

```

## B.6. Gyromatrise6

---

```
plot(sek,z3,'b')
plot(sek,z4,'k')
plot(sek,z5,'g')
plot(sek,z6,'y')
ylabel('Vinkelhastighet (deg/sek)')
xlabel('Tid (sekunder)')
hold off
```

```
%% Beregning av Allanvarians
t=0:0.02:N/50;
```

```
DATA.freq=xut1;
DATA.time=t;
tau=[];
name='av';
verbose=2;
```

```
[retval, s, errorb, tau] = allan(DATA,tau,name,verbose);
```

---

Feilmodell6g.m

## B.6 Gyromatrise6

---

```
clear all;
```

```
%% PROGRAM FOR Å SIMULERE GYROMATRISER MED 6 GYROER
```

```
load Feilmodell6.mat
```

```
% Systemmatriser:
```

```
F=[0 0 0 0 0 0 0;
    0 0 0 0 0 0 0;
    0 0 0 0 0 0 0;
    0 0 0 0 0 0 0;
    0 0 0 0 0 0 0;
    0 0 0 0 0 0 0;
    0 0 0 0 0 0 0];
G=[1 0 0 0 0 0 0;
    0 1 0 0 0 0 0;
    0 0 1 0 0 0 0;
    0 0 0 1 0 0 0;
    0 0 0 0 1 0 0;
    0 0 0 0 0 1 0;
```

---

```

    0 0 0 0 0 0 1];
H=[1 0 0 0 0 0 1;
   0 1 0 0 0 0 1;
   0 0 1 0 0 0 1;
   0 0 0 1 0 0 1;
   0 0 0 0 1 0 1;
   0 0 0 0 0 1 1];
B=[1 0 0 0 0 0;
   0 1 0 0 0 0;
   0 0 1 0 0 0;
   0 0 0 1 0 0;
   0 0 0 0 1 0;
   0 0 0 0 0 1];
L=[0 0 0 0 0 0 0]';
% Diskretisering:

% Kovariansmatrise uten korrelasjon:
%Q=eye(7);

% Kovariansmatrise med korrelasjon:
Q=[0.00189 -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163;
   -0.0000163 0.00189 -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163;
   -0.0000163 -0.0000163 0.00189 -0.0000163 -0.0000163 -0.0000163 -0.0000163;
   -0.0000163 -0.0000163 -0.0000163 0.00189 -0.0000163 -0.0000163 -0.0000163;
   -0.0000163 -0.0000163 -0.0000163 -0.0000163 0.00189 -0.0000163 -0.0000163;
   -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163 0.00189 -0.0000163;
   -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163 -0.0000163 0.00189];

[Fi ,La]=c2d(F,L,Ts);
[Ga]=kp2dpGa(F,G,Q,Ts);

u=0;
R=0.0049;
P=diag([0.1 0.1 0.1 0.1 0.1 0.1 0.1]);

X0=chol(P, 'lower')*randn(7,1);

X=X0;

% Simulering av gyromatrise
for k=1:N
    W=[xm1(1,k) xm2(1,k) xm3(1,k) xm4(1,k) xm5(1,k) xm6(1,k) 0.1*randn]';
    XS(:,k+1)=Fi*X+Ga*W;
end

```

## B.6. Gyromatrise6

---

**end**

```
% setter startverdi for prediktert kovarians  
P0est=P;  
Pp=Fi*P0est*Fi'+Ga*Q*Ga';
```

```
% setter startverdi for prediktert tilstand  
Xp(:,1)=Fi*X0+La*u;
```

```
for k = 1:N  
V=[xm1(2,k) xm2(2,k) xm3(2,k) xm4(2,k) xm5(2,k) xm6(2,k)]';
```

```
%—————KF—————  
%
```

```
% MO  
K=Pp*H'*inv(H*Pp*H'+R);  
Pe=(eye(7)-K*H)*Pp;  
z(:,k)=H*XS(:,k)+B*V;  
Xe(:,k)=Xp(:,k)+K*(z(k)-H*Xp(:,k));
```

```
%TO  
Xp(:,k+1)=Fi*Xe(:,k)+La*u;  
Pp=Fi*Pe*Fi'+Ga*eye(7)*Ga';
```

```
Xe(:,k)=Xp(:,end);  
Pe=Pp;
```

**end**

```
save Array6.mat
```

```
figure(7)  
plot(sek,Xe(4,:), 'b');  
xlabel('Tid (sekunder)')  
ylabel('Vinkelhastighet (deg/sek)')  
TITLE('Estimert vinkelhastighet (KF)')  
hold off
```

```
sluttid=N/50;  
t=0:0.02:sluttid;
```

---

```

DATA.freq=Xe(4 ,:);
DATA.time=t;
tau=[];
name='av';
verbose=2;

[retval , s , errorb , tau] = allan(DATA,tau,name,verbose);

```

---

Gyromatrise6.m

## B.7 Allan.m

Følgende kode brukes til beregning av Allanvarians, og er skrevet av M.A. Hopcroft [7].

---

```

function [retval , s , errorb , tau] = allan(data,tau,name,verbose)
% ALLAN Compute the Allan deviation for a set of time-domain frequency data
% [RETVAl, S, ERRORB, TAU] = ALLAN(DATA,TAU,NAME,VERBOSE)
%
% Inputs:
% DATA should be a struct and have the following fields:
% DATA.freq or DATA.phase
%           A vector of fractional frequency measurements (df/f) in
%           DATA.freq *or* phase offset data (seconds) in DATA.phase .
%           If frequency data is not present, it will be generated by
%           differentiating the phase data.
%           If both fields are present, then DATA.freq will be used.
%
% DATA.rate or DATA.time
%           The sampling rate in Hertz (DATA.rate) or a vector of
%           timestamps for each measurement in seconds (DATA.time).
%           DATA.rate is used if both fields are present.
%           If DATA.rate == 0, then the timestamps are used.
%
% DATA.units (optional)
%           The units for the data. If present, the string DATA.units
%           is added to the plot y-axis label.
%
% TAU is an array of tau values for computing Allan deviation.
%           TAU values must be divisible by 1/DATA.rate (data points cannot be
%           grouped in fractional quantities!) and invalid values are ignored.
%           Leave empty to use default values.
% NAME is an optional label that is added to the plot titles.
% VERBOSE sets the level of status messages:
%           0 = silent & no data plots;
%           1 = status messages & minimum plots;

```

## B.7. Allan.m

---

```
%      2 = all messages and plots (default)
%
% Outputs:
% RETVAL is the array of Allan deviation values at each TAU.
% S is an optional output of other statistical measures of the data (mean,
% std, etc).
% ERRORB is an optional output containing the error estimates for a 1-sigma
% confidence interval. These values are shown on the figure for each point.
% TAU is an optional output containing the array of tau values used in the
% calculation (which may be a truncated subset of the input or default
% values).
%
% Example:
%
% To compute the Allan deviation for the data in the variable "lt":
% >> lt
% lt =
%      freq: [1x86400 double]
%      rate: 0.5
%
% Use:
%
% >> ad = allan(lt,[2 10 100],'lt data',1);
%
% The Allan deviation will be computed and plotted at
% tau = 2,10,100 seconds. 1-sigma confidence intervals will be indicated by
% vertical lines at each point. You can also use the default settings,
% which are usually a good starting point:
%
% >> ad = allan(lt);
%
% Notes:
% This function calculates the standard Allan deviation (ADEV), *not* the
% overlapping ADEV. Use "allan_overlap.m" for overlapping ADEV.
% The calculation is performed using fractional frequency data. If only
% phase data is provided, frequency data is generated by differentiating
% the phase data.
% No pre-processing of the data is performed, except to remove any
% initial offset (i.e., starting gap) in the time record.
% For rate-based data, ADEV is computed only for tau values greater than the
% minimum time between samples and less than the half the total time. For
% time-stamped data, only tau values greater than the maximum gap between
% samples and less than half the total time are used.
% The calculation for fixed sample rate data is *much* faster than for
% time-stamp data. You may wish to run the rate-based calculation first,
% then compare with time-stamp-based.
% Often the differences are insignificant. To show the "tau bins"
% (y_k samples) on the data plot, set the variable TAUBIN to 1 (search for
```

```
% "#TAUBIN"). You can choose between loglog and semilog plotting of
% results by commenting in/out the appropriate line. Search for "#PLOTLOG".
% I recommend installing "dsplot.m", which improves the performance of
% plotting large data sets. Download from File Exchange, File ID: #15850.
% allan.m will use dsplot.m if it is present on your MATLAB path.
% This function has been validated using the test data from NBS Monograph
% 140, the 1000-point test data set given by Riley [1], and the example data
% given in IEEE standard 1139-1999, Annex C.
% The author welcomes other validation results, see contact info below.
%
% For more information, see:
% [1] W. J. Riley, "The Calculation of Time Domain Frequency Stability,"
% Available on the web:
% http://www.ieee-uffc.org/frequency\_control/teaching.asp?name=paper1ht
%
% M.A. Hopcroft
%   mhopen@gmail.com
%
% I welcome your comments and feedback!
%
% MH Oct2010
% v2.22 tau truncation to integer groups; tau sort
%       plotting bugfix
% v2.20 synchronize updates across allan, allan_overlap, allan_modified
% v2.16 add TAU as output, fixed unusual error with dsplot v1.1
% v2.14 update plotting behaviour, default tau values
%
versionstr = 'allan_v2.22';

% MH Jun2010
% v2.12 bugfix for rate data row/col orientation
%       add DATA.units for plotting
%       use dsplot.m for plotting
%
% MH MAR2010
% v2.1  minor interface and bugfixes
%       update data consistency check
%
% MH FEB2010
% v2.0  Consistent code behaviour for all "allan_x.m" functions:
%       accept phase data
%       verbose levels
%
% MH JAN2010
% v1.84 code cleanup
% v1.82 typos in comments and code cleanup
```

## B.7. Allan.m

---

```
%      tau bin plotting changed for performance improvement
% v1.8  Performance improvements:
%      vectorize code for rate data
%      logical indexing for irregular rate data
% MH APR2008
% v1.62 loglog plot option
% v1.61 improve error handling, plotting
%      fix bug in regular data calc for high-rate data
%      fix bug in timestamp data calc for large starting gap
%      (thanks to C. B. Ruiz for identifying these bugs)
%      uses timestamps for DATA.rate=0
%      progress indicator for large timestamp data processing
% MH JUN2007
% v1.54 Improve data plotting and optional bin plotting
% MH FEB2007
% v1.5  use difference from median for plotting
%      added MAD calculation for outlier detection
% MH JAN2007
% v1.48 plotting typos fixes
% MH DEC2006
% v1.46 hack to plot error bars
% v1.44 further validation (Riley 1000-pt)
%      plot mean and std
% MH NOV2006
% v1.42 typo fix comments
% v1.4  fix irregular rate algorithm
%      irregular algorithm rejects tau less than max gap in time data
%      validate both algorithms using test data from NBS Monograph 140
% v1.3  fix time calc if data.time not present
%      add error bars (not possible due to bug in MATLAB R14SP3)
%      remove offset calculation
% v1.24 improve feedback
% MH SEP2006
% v1.22 updated comments
% v1.2  errors and warnings
% v1.1  handle irregular interval data

% defaults
if nargin < 4, verbose=2; end
if nargin < 3, name=''; end
if nargin < 2 || isempty(tau), tau=2.^(-10:10); end

% plot "tau bins"? #TAUBIN
TAUBIN=0;% set 0 or 1 %WARNING:this has a significant impact on performance

% Formatting for plots
FontName = 'Arial';
FontSize = 14;
plotlinewidth=2;
```



---

```

if verbose >= 1, fprintf(1, 'allan:_%s\n\n', versionstr); end

%% Data consistency checks
if ~(isfield(data, 'phase') || isfield(data, 'freq'))
error ...
('Either 'phase' or 'freq' must be present in DATA. help file for details. ');
end
if isfield(data, 'time')
    if isfield(data, 'phase') && (length(data.phase) ~= length(data.time))
        if isfield(data, 'freq') && (length(data.freq) ~= length(data.time))
error ...
('The time and freq vectors are not the same length. help for details. ');
        else
error('The time and phase vectors are not the same length. help for details. ');
        end
    end
    if isfield(data, 'phase') && (any(isnan(data.phase)) || any(isinf(data.phase)))
error('The phase vector contains invalid elements (NaN/Inf). [con3] ');
    end
    if isfield(data, 'freq') && (any(isnan(data.freq)) || any(isinf(data.freq)))
    error('The freq vector contains invalid elements (NaN/Inf). [con4] ');
    end
    if isfield(data, 'time') && (any(isnan(data.time)) || any(isinf(data.time)))
error('The time vector contains invalid elements (NaN/Inf). [con5] ');
    end
end

% sort tau vector
tau=sort(tau);

%% Basic statistical tests on the data set
if ~isfield(data, 'freq')
    if isfield(data, 'rate') && data.rate ~= 0
        data.freq=diff(data.phase).*data.rate;
    elseif isfield(data, 'time')
        data.freq=diff(data.phase).*diff(data.time);
    end
    if verbose >= 1, fprintf...
(1, 'allan: Fractional frequency data generated from phase data (M=%g).\n', ...
length(data.freq)); end
    data.time(1)=[]; % make time stamps correspond to freq data
end
if size(data.freq, 2) > size(data.freq, 1), data.freq=data.freq'; end % ensure columns
s.numpoints=length(data.freq);
s.max=max(data.freq);

```

## B.7. Allan.m

---

```
s.min=min(data.freq);
s.mean=mean(data.freq);
s.median=median(data.freq);
if isfield(data,'time')
    if size(data.time,2) >...
        size(data.time,1), data.time=data.time'; end % ensure columns
    s.linear=polyfit(data.time(1:length(data.freq)),data.freq,1);
elseif isfield(data,'rate') && data.rate ~= 0;
    s.linear=polyfit((1/data.rate:1/data.rate:...
        length(data.freq)/data.rate)',data.freq,1);
else
error('Either "time" or "rate" must be present in DATA. Type "help allan"');
end
s.std=std(data.freq);

if verbose >= 2
    fprintf(1,'allan: input data statistics:\n');
    disp(s);
end

% center at median for plotting
medianfreq=data.freq-s.median;
sm=[]; sme=[];

% Screen for outliers using 5x Median Absolute Deviation (MAD) criteria
s.MAD = median(abs(medianfreq)/0.6745);
if verbose >= 2
    fprintf(1,'allan: 5x MAD value for outlier detection: %g\n',5*s.MAD);
end
if verbose >= 1 && any(abs(medianfreq) > 5*s.MAD)
    fprintf...
(1,'allan: NOTE: There appear to be outliers in the freq data. See plot.\n');
end

%%%
% There are two cases, either using timestamps or fixed sample rate:

%% Fixed Sample Rate Data
% If there is a regular interval between measurements, calculation is much
% easier/faster
if isfield(data,'rate') && data.rate > 0 % if data rate was given
    if verbose >= 1, fprintf...
(1,'allan: regular data (%g data points @ %g Hz)\n',...
length(data.freq),data.rate); end

    % string for plot title
    name=[name '( ' num2str(data.rate) ' Hz) '];
```

---

```

% what is the time interval between data points?
tmstep = 1/data.rate;

% Is there time data? Just for curiosity/plotting, does not impact calculation
if isfield(data, 'time')
    % adjust time data to remove any starting gap; first time step
    % should not be zero for comparison with freq data
    dtime=data.time-data.time(1)+mean(diff(data.time));
    if verbose >= 2
        fprintf(1, 'allan: End of timestamp data: %g sec.\n', dtime(end));
        if (data.rate - 1/mean(diff(dtime))) > 1e-6
            fprintf...
(1, 'allan:NOTE: data.rate does not match average timestamped sample rate\n', ...
data.rate, 1/mean(diff(dtime)));
        end
    end
else
    % create time axis data using rate (for plotting only)
    dtime=(tmstep:tmstep:length(data.freq)*tmstep)'; % column oriented
end

% check the range of tau values and truncate if necessary
% find halfway point of time record
halftime = round(tmstep*length(data.freq)/2);
% truncate tau to appropriate values
tau = tau(tau >= tmstep & tau <= halftime);
if verbose >= 2, fprintf...
(1, 'allan: allowable tau range: %g to %g sec. (1/rate to total_time/2)\n', ...
tmstep, halftime); end

% save the freq data for the loop
dfreq=data.freq;
% find the number of data points in each tau group
m = data.rate.*tau;
% only integer values allowed (no fractional groups of points)
%tau = tau(m-round(m)<1e-8); % numerical precision issues (v2.1)
tau = tau(m=round(m)); % The round() test is only correct for values < 2^53
%n = m(m-round(m)<1e-8); % change to round(m) for integer test v2.22
m = m(m=round(m));
%n=round(m);

if verbose >= 1, fprintf(1, 'allan: calculating Allan deviation\n'); end

% calculate the Allan deviation for each value of tau
k=0; tic;
for i = tau
    if verbose >= 2, fprintf(1, '%g', i); end
    k=k+1;

```

## B.7. Allan.m

---

```

    % truncate frequency set to an even multiple of this tau value
    freq=dfreq(1:end-rem(length(dfreq),m(k)));
    % group the data into tau-length groups or bins
    f = reshape(freq,m(k),[]); % Vectorize!
    % find average in each "tau group", y_k (each column of f)
    fa=mean(f,1);
    % first finite difference
    fd=diff(fa);
    % calculate two-sample variance for this tau
    M=length(fa);
    sm(k)=sqrt(0.5/(M-1)*(sum(fd.^2)));

    % estimate error bars
    sme(k)=sm(k)/sqrt(M+1);

    if TAUBIN == 1
        % save the binning points for plotting
        fs(k,1:length(freq)/m(k):m(k):length(freq));...
        fval{k}=mean(f,1);
    end

end % repeat for each value of tau

if verbose >= 2, fprintf(1,'\n'); end
calctime=toc; if verbose >= 2, fprintf...
(1,'allan: Elapsed time for calculation: %e seconds\n',calctime); end

%% Irregular data (timestamp)
elseif isfield(data,'time')
    % the interval between measurements is irregular
    % so we must group the data by time
    if verbose >= 1, fprintf...
(1,'allan: irregular rate data (no fixed sample rate)\n'); end

    % string for plot title
    name=[name '(timestamp)'];

    % adjust time to remove any initial offset or zero
    dtime=data.time-data.time(1)+mean(diff(data.time));
    %dtime=data.time;
    % where is the maximum gap in time record?
    gap_pos=find(diff(dtime)==max(diff(dtime)));
    % what is average data spacing?
    avg_gap = mean(diff(dtime));

    if verbose >= 2

```

```

fprintf...
(1, 'allan: WARNING: irregular timestamp data (no fixed sample rate).\n');
fprintf...
(1, 'Calc time may be long and the results subject to interpretation.\n');
fprintf(1,...
'You are advised to est using an avg sample rate instead of timestamps.\n',...
1/avg_gap);
fprintf(1,...
'Continue at your own risk! (press any key to continue)\n');
    pause;
end

    if verbose >= 1
        fprintf(1, 'allan: End of timestamp data: %g sec\n', dtime(end));
        fprintf(1, 'Average rate: %g Hz (%g sec/measurement)\n', ...
            1/avg_gap, avg_gap);
        if max(diff(dtime)) ~= 1/mean(diff(dtime))
            fprintf(1, 'Max. gap: %g sec at position %d\n', ...
                max(diff(dtime)), gap_pos(1));
        end
        if max(diff(dtime)) > 5*avg_gap
            fprintf(1,...
'Max gap in time record is suspiciously large (>5x the average interval).\n');
        end
    end

    % find halfway point
    halftime = fix(dtime(end)/2);
    % truncate tau to appropriate values
    tau = tau(tau >= max(diff(dtime)) & tau <= halftime);
    if isempty(tau)
        error('allan: ERROR: no appropriate tau values (>%g s, <%g s)\n', ...
max(diff(dtime)), halftime);
    end

    % save the freq data for the loop
    dfreq=data.freq;
    dtime=dtime(1:length(dfreq));

    if verbose >= 1, fprintf(1,'allan: calculating Allan deviation...\n'); end

    k=0; tic;
    for i = tau
        if verbose >= 2, fprintf(1,'%d ',i); end

        k=k+1; fa = []; %f = [];
        km=0;

```

## B.7. Allan.m

---

```

% truncate data set to an even multiple of this tau value
freq=dfreq(dtime <= dtime(end)-rem(dtime(end),i));
time=dtime(dtime <= dtime(end)-rem(dtime(end),i));
%freq=dfreq;
%time=dtime;

% break up the data into groups of tau length in sec
while i*km < time(end)
    km=km+1;

    % progress bar
    if verbose >= 2
        if rem(km,100)==0, fprintf(1, '. '); end
        if rem(km,1000)==0, ...
            fprintf(1, '%g/%g\n', km, round(time(end)/i)); end
    end

    f = freq(i*(km-1) < time & time <= i*km);
    f = f(~isnan(f)); % make sure values are valid

    if ~isempty(f)
        fa(km)=mean(f);
    else
        fa(km)=0;
    end

if TAUBIN == 1 % WARNING: this has a significant impact on performance
    % save the binning points for plotting
    %if find(time <= i*km) > 0
        fs(k,km)=max(time(time <= i*km));
    %else
        if isempty(fs(k,km))
            fs(k,km)=0;
        end
        fval{k}=fa;
    end % save tau bin plot points

end

if verbose >= 2, fprintf(1, '\n'); end

% first finite difference of the averaged results
fd=diff(fa);
% calculate Allan deviation for this tau
M=length(fa);
sm(k)=sqrt(0.5/(M-1)*(sum(fd.^2)));

% estimate error bars
sme(k)=sm(k)/sqrt(M+1);

```

```

end

    if verbose == 2, fprintf(1, '\n'); end
    calctime=toc; if verbose >= 2, fprintf...
(1, 'allan: Elapsed time for calculation: %e seconds\n', calctime); end

else
    error('allan: WARNING: no DATA. rate or DATA. time! Type "help allan"');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Plotting

if verbose >= 2 % show all data

    % plot the frequency data, centered on median
    if size(dtime,2) > size(dtime,1), dtime=dtime';...
end % this should not be necessary, but dsplot 1.1 is a little bit brittle
    try
        % dsplot makes a new figure
        hd=dsplot(dtime, medianfreq);
    catch ME
        figure;
        hd=plot(dtime, medianfreq);
        if verbose >= 1, fprintf...
(1, ...
'Install dsplot.m for large data sets (FileExchange FileID: #15850)'; end
        if verbose >= 2, fprintf(1, '(Message: %s)\n', ME.message); end
    end
    set(hd, 'Marker', '.', 'LineStyle', 'none', 'Color', 'b'); % equivalent to '-.'
    hold on;

    % show center (0)
    plot(xlim, [0 0], ':k');
    % show 5x Median Absolute Deviation (MAD) values
    hm=plot(xlim, [5*s.MAD 5*s.MAD], '-r');
    plot(xlim, [-5*s.MAD -5*s.MAD], '-r');
    % show linear fit line
    hf=plot(xlim, polyval(s.linear, xlim)-s.median, '-g');
    title(['Data: ' name], 'FontSize', FontSize+2, 'FontName', FontName);
    %set(get(gca, 'Title'), 'Interpreter', 'none');
    xlabel('Time [sec]', 'FontSize', FontSize, 'FontName', FontName);
    if isfield(data, 'units')
ylabel(['data - median(data) [ ' data.units ' ]'], 'FontSize', ...
FontSize, 'FontName', FontName);

```

## B.7. Allan.m

---

```

else
ylabel('freq_□_median(freq)', 'FontSize',FontSize, 'FontName',FontName);
end
set(gca, 'FontSize',FontSize, 'FontName',FontName);
legend([hd hm hf],{ 'data_□(centered_□on_□median)',...
'5x_□MAD_□outliers', ['Linear_□Fit_□(' num2str...
(s.linear(1), '%g' ) ')]}, 'FontSize',max(10,FontSize-2));
% tighten up
xlim([ dtime(1) dtime(end)]);

% Optional tau bin (y_k samples) plot
if TAUBIN == 1
% plot the tau divisions on the data plot
rfs=size(fs,1);
colororder=get(gca, 'ColorOrder');
axis tight; kc=2;
%ap=axis;
for j=1:rfs
kc=kc+1; if rem(kc,length(colororder))==1, kc=2; end
%for b=1:max(find(fs(j,:))); % new form of "find" in r2009a
for b=1:find(fs(j,:), 1, 'last' );
% plot the tau division boundaries
%plot([fs(j,b) fs(j,b)],[ap(3)*1.1 ap(4)*1.1], '- ', 'Color', colororder(kc,:));
% plot tau group y values
if b == 1
plot([ dtime(1) fs(j,b)],[ fval{j}(b)-s.median fval{j}...
(b)-s.median], '- ', 'Color', colororder(kc,:), 'LineWidth',4);
else
plot([ fs(j,b-1) fs(j,b)],[ fval{j}(b)-s.median fval{j}...
(b)-s.median], '- ', 'Color', colororder(kc,:), 'LineWidth',4);
end
end
end
axis auto
end % End optional bin plot

end % end plot raw data

if verbose >= 1 % show ADEV results

% plot Allan deviation results
if ~isempty(sm)
figure

% Choose loglog or semilogx plot here #PLOTLOG
%semilogx(tau,sm, '-b', 'LineWidth', plotlinewidth, 'MarkerSize',24);
loglog(tau,sm, '-b', 'LineWidth', plotlinewidth, 'MarkerSize',24);

```



---

```

% in R14SP3, there is a bug that screws up the error bars on a semilog plot.
% When this is fixed in a future release,
% uncomment below to use normal errorbars
    %errorbar(tau,sm,sme,'.-b'); set(gca,'XScale','log');
    % this is a hack to approximate the error bars
hold on; plot([tau; tau],[sm+sme; sm-sme],...
    '-k','LineWidth',max(plotlinewidth-1,2));

    grid on;
    title(['Allan Deviation: ' name],...
        'FontSize',FontSize+2,'FontName',FontName);
    %set(get(gca,'Title'),'Interpreter','none');
    xlabel('tau [sekunder]','FontSize',FontSize,'FontName',FontName);
    if isfield(data,'units')
        ylabel(['sigma_y(tau) [ ' data.units ' ]'],...
            'FontSize',FontSize,'FontName',FontName);
    else
        ylabel('sigma_y(tau)','FontSize',FontSize,'FontName',FontName);
    end
    set(gca,'FontSize',FontSize,'FontName',FontName);
    % expand the x axis a little bit so that the errors bars look nice
    adax = axis;
    axis([adax(1)*0.9 adax(2)*1.1 adax(3) adax(4)]);

    % display the minimum value
    fprintf(1,'allan: Minimum ADEV value: %g at tau = %g seconds\n',...
        min(sm),tau(sm==min(sm)));

    elseif verbose >= 1
        fprintf(1,'allan: WARNING: no values calculated.\n');
        fprintf(1,...
        'Check that TAU > 1/DATA.rate and TAU values are divisible by 1/DATA.rate\n');
        fprintf(1,'Type "help allan" for more information.\n\n');
    end

end % end plot ADEV data

retval = sm;
errorb = sme;

return

```

---

allan.m