



Evaluering og forbedring av GPS posisjonsmåling for mobile enheter

av

**Hermod Finjord
Geir Øynes**

**Hovedoppgave til mastergraden i
Informasjons- og Kommunikasjonsteknologi**

**Høgskolen i Agder
Fakultet for teknologi
Grimstad, juni 2004**

Sammendrag

Det siste tiåret har det blitt drept rundt 100 000 sauer på beite hvert år og i 2003 ble erstatningen til bøndene på over 50 millioner kroner for tap av dyr. I de siste årene har det også vært flere alvorlige epidemier blant sauer i Europa. Med dette som utgangspunkt har det blitt lagt fram et forskriftsforslag som sier at eieren skal ha tilsyn til dyrene sine minst en gang i uka. Til dags dato har tilsyn av dyr på beite stort sett blitt gjort manuelt. I den senere tid har løsninger ved bruk av GPS posisjonsmåling blitt lansert. Mer avansert teknologi gjør at GPS mottakere stadig blir mindre og billigere, noe som fører til at GPS mottakere i økende grad implementeres sammen med annen type teknologi.

Kitron Development AS har i samarbeid med Grønvold Matheson technology utviklet en lokaliseringseenhet for sauer og storfe, basert på GPS posisjonsmåling og radiokommunikasjon. Radiobjelleprosjektet startet høsten 2000 og den første prototypen var ferdig høsten 2002. GPS enheten på sauene beregner sauens posisjon som sendes videre til en basestasjon ved hjelp av en VHF radiosender. Basestasjonen videresender alle posisjonsmålingene til en sentral server. Ved å logge seg på via internett kan bonden ha full oversikt over hvor dyrene hans til enhver tid befinner seg. Våren 2003 ble det laget 1000 enheter som den påfølgende sommeren ble utprøvd på dyr på beite. Dette utviklingsforsøket viste at det var tilsynelatende stor spredning i målenøyaktigheten. For slike systemer er det viktig å oppnå pålitelige og relativt nøyaktige posisjonsmålinger.

Vårt prosjekt viser at ved bruk av ”billige” GPS mottakere kan en under tilnærmet perfekte forhold oppnå en nøyaktighet på rundt 10 meter. Dette er tilfredsstillende i forhold til å lokalisere sau i naturen. Prosjektet vårt viser imidlertid at omgivelsene sauene lever under gjør at GPS mottakeren opplever dårlige forhold, noe som igjen kan føre til at nøyaktigheten i posisjonsmålingene blir dårligere. For at bonden skal kunne finne igjen dyrene sine er det viktig at posisjonen som oppgis blir så nøyaktig som mulig selv om forholdene er dårlige. En del av vår oppgave gikk derfor ut på å se om det i etterkant er mulig å forbedre nøyaktigheten i GPS data.

Vi har i denne oppgaven utviklet metoder/algoritmer, som kan brukes på GPS posisjonsmålinger i etterkant for å øke nøyaktigheten. Vi har også vist at tiden GPS mottakeren bruker på å starte opp er kritisk. Det er veldig viktig at mottakeren får nok tid til å beregne en nøyaktig posisjon. Får mottakeren for kort tid til å beregne en posisjon blir denne mest sannsynlig unøyaktig eller i verste fall ikke gyldig.

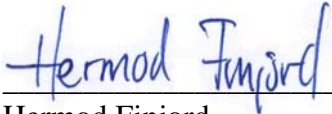
Ved å benytte oss av de oppstartstidene vi har kommet fram til og de metodene vi har utviklet for å øke nøyaktigheten, har vi vist at andelen posisjonsmålinger som har en nøyaktighet innenfor 10 meter øker fra ca 78 % til 93 %.

Forord

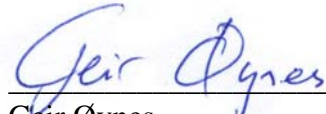
Denne oppgaven avslutter Masterutdanningen i Informasjon og Kommunikasjonsteknologi ved høgskolen i Agder. Arbeidet har pågått i vårsemesteret 2004 over en periode på 20 uker.

Vi vil rette en takk til vår veileder på HiA, førstelektor Rune Fensli, for verdifull veiledning og innspill under hele prosjektperioden. En stor takk går også til prosjektleder Erik Hardeng hos Kitron Developement for kyndig veiledning og for all informasjon i forhold til Radiobjelle prosjektet. Til slutt vil vi takke Aanon Grimnes, Tore Tveit og Tore Eikenes for all hjelp i forbindelse med feltforsøkene vi utførte.

Grimstad, 1. juni 2004



Hermod Finjord



Geir Øynes

Innholdsfortegnelse

1	INNLEDNING	1
1.1	Radiobjelle prosjektet.....	2
1.1.1	Dyreenheten	2
1.1.2	Basestasjonen.....	5
1.1.3	Sentral Server.....	5
1.2	Oppgavedefinisjon.....	6
1.3	Begrensninger.....	6
1.4	Mulige årsaker til problemet	6
2	INTRODUKSJON TIL KOORDINATER OG KOORDINATSYSTEM.....	9
2.1	GPS posisjonering	9
2.1.1	WGS-84	10
2.1.2	Konverteringseksempler	12
2.2	Koordinater og Kart.....	13
2.2.1	UTM.....	13
2.2.2	NGO.....	15
2.3	Usikkerhet ved koordinat transformasjon.....	16
3	INTRODUKSJON TIL GPS.....	17
3.1	Innledning.....	17
3.2	Satellitt segmentet	18
3.2.1	GPS signalet.....	19
3.2.2	Navigasjonsmelding.....	20
3.3	Kontroll segmentet	21
3.4	Bruker segmentet.....	22
3.4.1	Kald, varm og hotstart.....	22
3.4.2	NMEA-0183	23
3.5	Beregning av posisjon	24
3.6	Feilkilder	26
3.6.1	Feilkilder i satellitt	26
3.6.2	Feilkilder i mottaker.....	28
3.6.3	Feilkilder fra signalforplantning.....	28
3.6.4	Andre feilkilder	30
3.7	Differensiell GPS	32
4	METODE.....	34
4.1	Matlab testapplikasjon.....	34
4.2	Nøyaktighetsmålinger	35
4.3	Simuleringer	36
4.3.1	Midling.....	37
4.3.2	Sirkelmidling.....	37
4.3.3	HDOP midling	39
4.4	Oppstartsmålinger	39
5	RESULTAT	40
5.1	Nøyaktighetsmålinger	40
5.2	Simuleringer	41
5.3	Oppstartsmålinger	43
5.3.1	Oppstartstider.....	43
5.3.2	Nøyaktighet etter oppstart.....	47
5.4	Behandling av posisjonsdata	50
5.4.1	Filtrering	50
5.4.2	Midling.....	52
5.4.3	Sirkelmidling.....	54
5.4.4	Midling med HDOP	56

6	DRØFTING.....	60
6.1	Nøyaktighetsmålinger	60
6.2	Simuleringer	60
6.3	Oppstartsmålinger	61
6.3.1	Fri sikt målinger	61
6.3.2	Målinger med dårligere forhold	61
6.3.3	Målinger med bevegelse	62
6.3.4	Oppstartstider	62
6.4	Behandling av posisjonsdata	64
6.4.1	Filtrering	64
6.4.2	Midling	64
6.4.3	Sirkelmidling	65
6.4.4	Midling med HDOP	66
6.5	Videre arbeid	66
7	KONKLUSJON	67
	REFERANSER:.....	69
	FORKORTELSER.....	73
	VEDLEGG.....	74

Figurliste

Figur 1.1: Radiobjelle prosjektet	2
Figur 1.2: Dyreenheten [5]	3
Figur 1.3: Forenklet blokkskjema for dyreenhetene.....	4
Figur 1.4: Forenklet blokkskjema for basestasjon [6]	5
Figur 1.5: Kjøretur rundt i Grimstad	8
Figur 2-1: Forskjellen mellom geoide, ellipsoide og jordens topologi [9].....	9
Figur 2-2: WGS-84 koordinatsystem [10].....	11
Figur 2-3: Koordinater til en posisjon på jordens overflate [9].....	12
Figur 2-4: Sylinderen skjærer jordoverflaten og danner en sone [12].....	14
Figur 2-5: UTM sonebelter [13]	15
Figur 2-6: NGO1948 systemet [13].....	16
Figur 3-1: Satellitt segment, bruker segment og kontroll segment.....	18
Figur 3-2: Satellittene i bane rundt jorda [21]	18
Figur 3-3: CDMA	19
Figur 3-4: BPSK	20
Figur 3-5: GPS signalet [24].....	20
Figur 3-6: Navigasjonsmelding	21
Figur 3-7: Effekten av reflekterte signal.....	30
Figur 3-8: (a) God satellitt geometri, (b) Dårlig satellitt geometri.....	31
Figur 3-9: Differensiell GPS [41]	33
Figur 4-1: Brukergrensesnitt testapplikasjon.....	35
Figur 4-2: Plassering av stålrør over GPS mottaker	36
Figur 4-3: Sirkelmidling, R=Radius, X=antall repetisjoner	38
Figur 5-1: Spredningsdiagram fri sikt	40
Figur 5-2: Avstand fra fastpunkt	40
Figur 5-3: Antall satellitter mottakeren har kontakt med	41
Figur 5-4: Sammenlikning av våre og Kitrons data	42
Figur 5-5: Spredningsdiagram med rør over mottaker	42
Figur 5-6: Avstand fra fastpunkt med rør over mottaker.....	43
Figur 5-7: Varmstart i fri sikt	44
Figur 5-8: Varmstart i fri sikt med 2 km feil i posisjon sent til GPS mottaker	44
Figur 5-9: Kaldstart i fri sikt.....	45
Figur 5-10: Varmstart med dårlige forhold	45
Figur 5-11: Kaldstart med dårlige forhold.....	46
Figur 5-12: Varmstart mens GPS mottakeren er i bevegelse	46
Figur 5-13: Kaldstart mens GPS mottakeren er i bevegelse.....	47
Figur 5-14: Varmstart i fri sikt	48
Figur 5-15: Varmstart i fri sikt med 2 km feil i posisjon sent til GPS mottaker	48
Figur 5-16: Kaldstart i fri sikt.....	49
Figur 5-17: Varmstart med dårlige forhold	49
Figur 5-18: Kaldstart med dårlige forhold.....	50
Figur 5-19: Mer eller lik 5, 6, 7 og 8 satellitter	51
Figur 5-20: HDOP mindre eller lik 20, 10, 4, 2	52
Figur 5-21: Midling over 15 målinger hvor 10 må være gyldige.....	53
Figur 5-22: Midling over 60 målinger hvor 50 må være gyldige.....	54
Figur 5-23: Midling over 15 målinger hvor 10 må være gyldige.....	55

Figur 5-24: Midling over 60 målinger hvor 50 må være gyldige.....	56
Figur 5-25: Midles over 15, 10 målinger har HDOP lavere enn 2	57
Figur 5-26: Midles over 15, 10 målinger må ha HDOP lavere enn 4	58
Figur 5-27: Midling over 60 målinger, 50 må ha HDOP lavere enn 4.....	59

Tabelliste

Tabell 1.1: Antall sau på beite og tapstall [1].....	1
Tabell 1.2: Tidspunkt for når dyreenheten tar målinger.....	3
Tabell 1.3: Tilstand til sauen [5].....	4
Tabell 1.4: Kitrons målinger.....	7
Tabell 2.1: Forskjell på Bessel og modifisert Bessel ellipsoide [14]	15
Tabell 3.1: GPGGA meldingsformat [25]	23
Tabell 3.2: Navigation Initialization Data Format [25].....	24
Tabell 3.3: Feilkilder ved 1-sigma (68.3%) [10].....	26
Tabell 3.4: Typiske feilkilders innvirkning på totalfeilen ved bruk av GPS og DGPS [40]	32
Tabell 5-1: Antall gjenværende målinger etter filtrering.....	52
Tabell 6-1: Utregning av oppstartstid ved varmstart.....	63
Tabell 6-2: Utregning av oppstartstid ved kaldstart	63

1 Innledning

Kitron Development AS har i samarbeid med Grønvold Matheson technology utviklet en lokaliseringseenhet for sauer og storfe, basert på elektronisk posisjonering og radiokommunikasjon. Hovedformålet med Radiobjelle prosjektet er å oppnå bedre oppsyn av dyr på beite.

Det siste tiåret har det blitt drept rundt 100000 sauer på beite hvert år [1] og i 2003 ble erstatningen til bøndene på over 50 millioner kroner for tap av dyr [2]. Antall sauer på beite har vært ganske stabilt på rundt to millioner dyr de siste ti årene, men som tabell 1.1 viser har tapsprosenten økt fra 4,2 % i 1993 til 6,4 % i 2003.

Tabell 1.1: Antall sau på beite og tapstall [1]

Årstall	Antall dyr på beite [mill]	Tap [%]
1993	2,082	4,2
1994	2,182	4,6
1995	2,225	4,9
1996	2,235	5,9
1997	2,136	5,8
1998	2,095	6,0
1999	1,997	6,2
2000	2,010	5,8
2001	2,036	5,8
2002	2,063	6,0
2003	2,045	6,4

Med slike høye tapstall er det svært ønskelig å redusere tapene, både med hensyn på sauene levevilkår og de økonomiske tapene eieren blir påført.

I de siste årene har det vært flere alvorlige epidemier blant sauer i Europa. Dette har ført til et forslag til en ny forskrift som sier at eieren skal ha tilsyn til dyrene sine minst en gang i uka [3]. Dette vil være vanskelig å gjennomføre uten noen hjelpemidler for å finne eller overvåke dyrene. Til dags dato har tilsyn av dyr på beite stort sett blitt gjort manuelt. Slikt tilsyn er svært tidskrevende og krever store arbeidsressurser. Radiopeileutstyr er en metode som benyttes i dag men er en tidkrevende teknikk og krever ofte svært dyrt utstyr. Radiobjelleprosjektet vil gi eieren muligheten til å følge hvert enkelt dyr på en datamaskin hjemme i sin egen stue.

Når sauene skal leveres til slakteri etter beitesesongen kan lang innsankingstid medføre forsinkelser. Radiobjelleprosjektet kan hjelpe bøndene å gjøre innsankingen mer forutsigbar, noe som vil gjøre bøndene til bedre underleverandører for slakteriene. Det vil også være lettere for bøndene å dokumentere at dyrene har vært sykdomsfrie, ikke vært skadet og har hatt god tilgang på mat siden de kan følge hvert enkelt dyr [3].

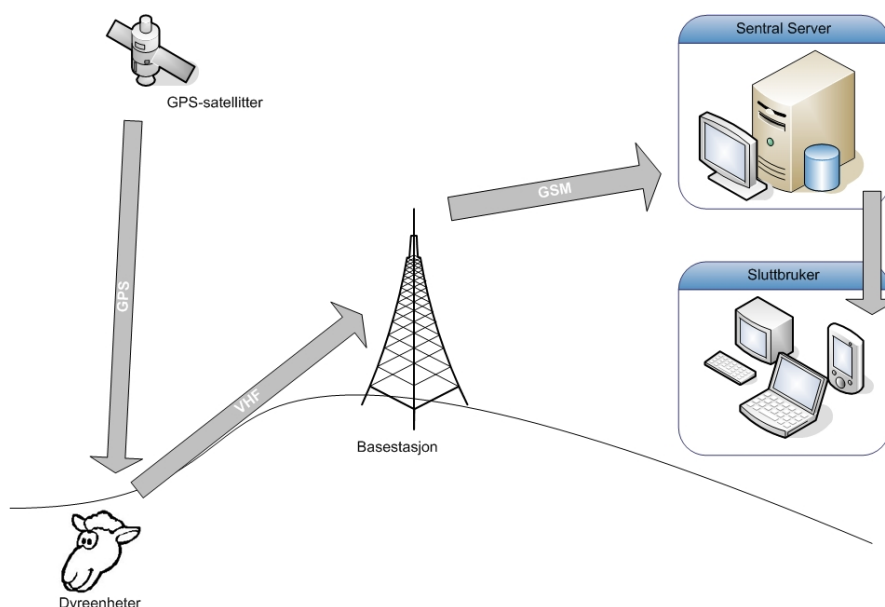
Det er også andre som har vist interesse for prosjektet. Norsk institutt for jord- og skogkartlegging (NIJOS) har som oppgave å undersøke informasjon om jord- skog-, utmarks- og landskapsressurser i Norge. Ved å ta i bruk loggene fra dyrene på beite kan NIJOS kartlegge hvor godt beiteområdene i Norge benyttes. Om bøndene tar i bruk den

informasjonen som NIJOS sitter inne med etter denne kartleggingen, vil de ha muligheten til å sette ut sauene i de beste beiteområdene. Dette er viktig for dyrenes kår da det i 2003 ble levert over 20.000 underernærte lam til slakt [4]. I tillegg vil bedre fôr gi bedre kjøtt som igjen kan bedre inntektene for eierne.

1.1 Radiobjelle prosjektet

Radiobjelleprosjektet ble startet høsten 2000 og er foreslått og patentert av Alexander Grønvold. Den første prototypeløsningen var ferdig høsten 2002. Våren 2003 ble det laget 1000 enheter som den påfølgende sommeren ble utprøvd på dyr på beite.

Radiobjelle prosjektet består av tre deler; dyreenheter, basestasjoner og en sentral server. Det er vanligvis et stort antall dyreenheter per basestasjon. Alle basestasjonene vil sende til en sentral server. Fra den sentrale serveren kan sluttbrukeren hente ut informasjon om sine dyr. Oppbygningen av Radiobjelle teknologien er illustrert i figur 1.1.



Figur 1.1: Radiobjelle prosjektet

1.1.1 Dyreenheten

Dyreenheten er den delen som henger på sauene. Dyreenheten inneholder fire hoveddeler; GPS-modul, VHF-radio, mikrokontroller og diverse sensorer. Bilde av dyreenheten som ble utprøvd sommeren 2003 er vist på figur 1.2.



Figur 1.2: Dyreenheten [5]

GPS modulen vil vekkes opp hver andre time på dagtid for å ta en posisjonsmåling. Hvis GPS modulen fikk en gyldig måling sist den var aktiv vil den benytte varmstart. Hvis ikke vil den benytte kaldstart. Klokkeslett for når GPS modulen blir vekket opp og maks antall sekunder den står på for å finne gyldig posisjonsmåling er vist i tabell 1.2.

Tabell 1.2: Tidspunkt for når dyreenheten tar målinger

Klokkeslett	Maks antall sekunder
4	250
6	50
8	50
10	50
12	50
14	180
16	50
18	50

Hvis GPS modulen finner gyldig måling før maks antall sekunder er gått, og den har flere enn seks satellitter tilgjengelig vil denne posisjonen benyttes og GPS modulen vil slås av. Posisjonen sendes til basestasjonen fire ganger i døgnet.

Mikrokontrolleren samler informasjon fra GPS modulen og de sensorene som er tilkoblet. Informasjonen som radiobjella sender til basestasjonen er følgende:

- Dyrets ID
- Tidspunkt målingen ble tatt
- GPS status (gyldig posisjonsmåling)
- Lengdegrad
- Breddegrad
- Antall satellitter GPS mottakeren har kontakt med
- Temperatur i omgivelsene
- Batterispenning

Hvert dyr får en unik ID. På denne måten kan man følge bevegelsene til hvert enkelt dyr.

GPS modulen gir informasjon om posisjon angitt i lengdegrad, breddegrad og høyde. I tillegg til dette gir den nøyaktig tidspunkt og dato. Denne tidsinformasjonen benyttes til å sende posisjonsmåling oftere i perioden hvor innsankingene skal finne sted. GPS modulen gir også ut informasjon om kvalitet på posisjonsmålingen samt diverse andre parametere. Kitron har kun valgt å sende GPS status og antall satellitter som GPS modulen har kontakt med til basestasjonen.

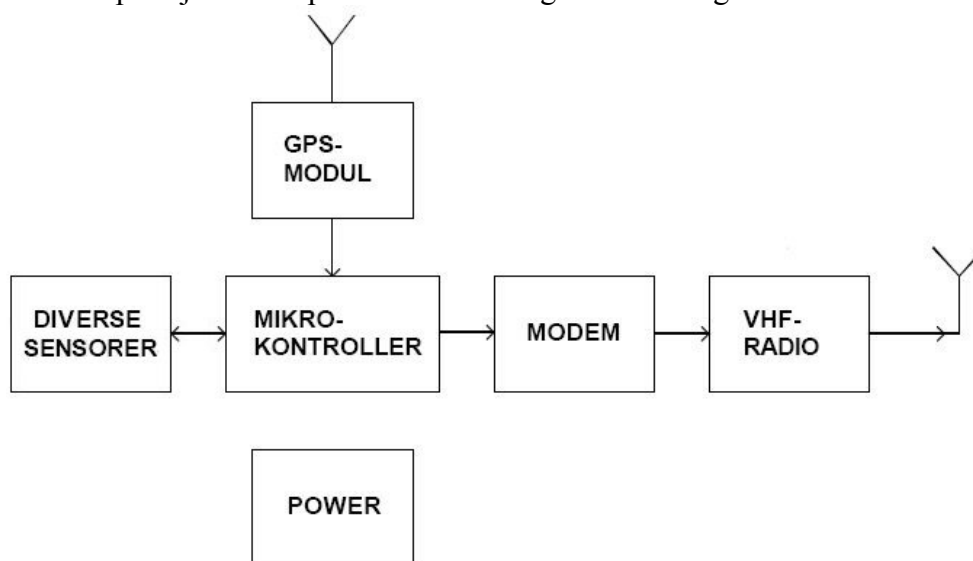
I prøveperioden ble tre ekstra sensor implementert. Den ene gav ut temperatur i omgivelsene. Den andre sensoren er et voltmeter som måler spenningen i batteriet og den tredje sensoren detekterer om sauene beveger seg. Bevegelsessensoren kan sammen med informasjonen fra GPS modulen gi god indikasjon på om sauene er død eller sitter fast. Hvordan dette fastslås er angitt i tabell 1.3.

Tabell 1.3: Tilstand til sauene [5]

Vibrasjon	GPS beveger seg	Status	Alarm
Ja	Ja	Dyret vandrer, OK	Nei
Ja	Nei	Dyret kan stå fast	Ja
Nei	Ja	Umulig, Tyvslakt?	Ja
Nei	Nei	Dyret er dødt	Ja

Kitron ser imidlertid for seg mulighetene til å implementere flere forskjellige sensorer. Dette kan være blant annet sensorer som måler puls og temperatur på sauene. Dette kan gi god varsling dersom det er sykdom i dyrebestanden.

For å sende dataene til basestasjonen benyttes en radio-modul som sender i VHF-båndet. Kitron valgte å søke om en egen frekvens til Radiobjelle prosjektet på grunn av stor trafikk i de ulisenserte frekvensbåndene, samt at de frie båndene ligger høyere i frekvens enn det Kitron ønsket med tanke på dekning og rekkevidde. Av Post- og Teletilsynet fikk Kitron tildelt en ledig frekvens til prosjektet på 83 MHz. Kitron har imidlertid valgt at det kun sendes fra dyreenheten til basestasjonen. Dette betyr at det ikke er mulighet for basestasjonen å be om retransmisjon om det skulle bli feil i overføringen. Dette har Kitron løst ved at hver posisjon blir repetert i tre etterfølgende meldinger.



Figur 1.3: Forenklet blokkdiagram for dyreenhetene

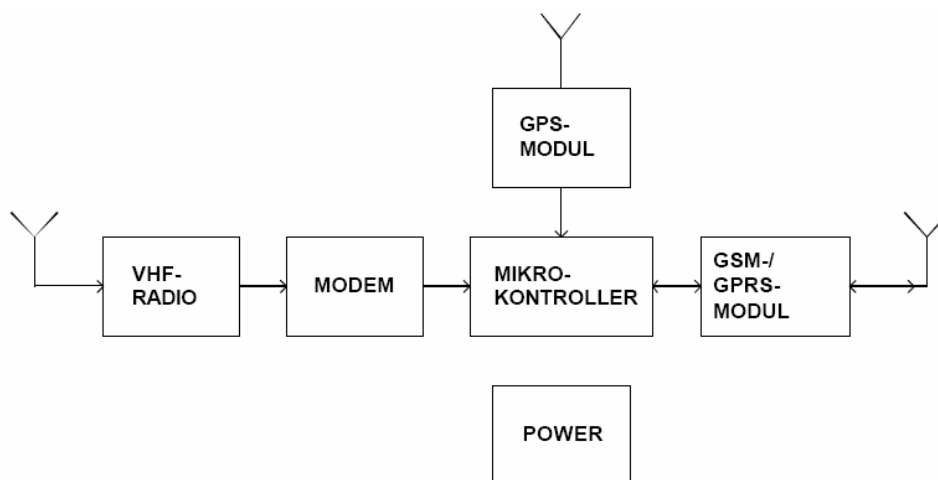
1.1.2 Basestasjonen

Basestasjonen mottar dataene fra alle dyreenhetene. Denne blir strategisk plassert i terrenget, gjerne på en høyde for best å kunne motta meldingene fra alle dyreenhetene. Den er selvforsynt med energi ved hjelp av et solcellepanel.

Basestasjonen inneholder tre antenner. En VHF antenne for kommunikasjon med dyreenheten og en GSM antenne for å sende meldingene fra basestasjonene til den sentrale serveren. En GPS antenne er tatt med for å kunne implementere DGPS i fremtiden. Basestasjonen kan også fjernstyres 100 % via GSM-nettet. Dette er svært nyttig da basestasjonen ofte ikke er lett tilgjengelig.

I tillegg til informasjonen som kommer fra dyreenheten blir følgende data lagt til meldingen før den sendes videre til den sentrale serveren.

- Tidspunkt for når meldingen blir lagret
- Tidspunktet for basestasjonen
- Dato for basestasjonen
- Lengdegrad til basestasjonen
- Breddegrad til basestasjonen
- Antall satellitter basestasjonen har kontakt med
- Høyde



Figur 1.4: Forenklet blokkskjema for basestasjon [6]

1.1.3 Sentral Server

Den sentrale serveren mottar meldingene fra alle basestasjonene. Her blir alle dataene lagret i en database.

Sluttbrukeren vil få tilgang til disse dataene via internett, hvor man har mange forskjellige muligheter til å vise dataene. Man kan velge å få ut dataene i lengdegrad og breddegrad eller grafisk på et kart. På kartet kan man velge om man vil se hele "flokk", eller om en bare ønsker å se bevegelsene til en enkelt sau i et angitt tidsrom.

Sluttbruker vil også ha mulighet til å motta alarmer på SMS. Dette er implementert for å raskere nå eier om noe skulle skje med dyrene. Det kan også bli aktuelt å kunne motta

posisjon på SMS. På denne måten kan man motta siste oppdaterte informasjon om sauens posisjon når man driver innsanking og man gjerne ikke har tilgang til internett.

1.2 Oppgavedefinisjon

"I tilknytning til et pågående utviklingsprosjekt, der det benyttes GPS posisjonsmåling for trådløse enheter, er det gjennom sommeren 2003 registrert ca 20 000 posisjoner fra objekter. Det er tilsynelatende stor spredning i målenøyaktighet, og for mange anvendelsesformål kan det være behov for å oppnå pålitelige og relativt nøyaktige posisjonsmålinger.

Gjennom oppgaven er det ønskelig å vurdere hvilke faktorer som påvirker nøyaktighet og ytelse til en mobil GPS-mottaker som brukes under varierende betingelser. Det er ønskelig å teste målenøyaktighet gjennom feltforsøk som grunnlag for å foreslå mulige forbedringer som kan implementeres i mobile GPS enheter."

I begrepet relativt nøyaktige målinger sier Kitron i sin endelige spesifisering for Radiobjelle prosjektet følgende: *"Posisjonsregistrering skal være nøyaktig nok til at dyret kan identifiseres fra registrert posisjon visuelt eller basert fra lyd på tradisjonell bjelle"*. Etter nærmere samtaler med Kitron kom det frem at posisjonsmålingene bør være innenfor 10 meter. Den viktigste årsaken til at man trenger så nøyaktige målinger er å gjøre jobben enklere når en skal drive innsanking, finne døde eller syke dyr og dyr som sitter fast.

Videre i spesifikasjonen sier Kitron: *"Systemet skal være fullt operativt i minst 90 % av tilgjengelig periode"*. Vi vil i henhold til dette utsagnet forsøke å oppnå posisjonsmålinger innenfor 10 meter i 90 % av tilfellene.

I tillegg har Kitron ytret ønske om at rapporten skal inneholde en fyldig teoridel.

1.3 Begrensninger

Det finnes en del begrensninger i vår oppgave i forhold til hvordan utstyret vil bli benyttet i virkeligheten. Vi har ikke hatt anledning til å benytte nøyaktig det samme utstyret som Kitron benytter. Utstyret vi har benyttet er en GPS mottakeren som kan kobles direkte til datamaskinen via et USB grensesnitt. GPS mottakeren vi benyttet er at type Sapphire GPS Mouse, produsert av Royaltek.

Vi har også utført de fleste målingene våre i vinterhalvåret, mens sauene har sin beitesesong på sommerhalvåret.

Prosjektet har en varighet på 20 uker.

1.4 Mulige årsaker til problemet

I starten av prosjektet satte vi opp noen hypoteser på hva årsaken til problemene Kitron hadde under feltperioden kunne være. Disse ble utformet på følgende måte:

- Kitron gir ikke GPS modulen tilstrekkelig tid slik at den skal være i stand til å beregne en gyldig posisjon.

- Forholdene sauen lever under fører til at GPS mottakeren opplever forholdene som dårlige og dermed er den ikke i stand til å beregne en nøyaktig posisjon.
- Det må være mulig å finne en algoritme som kan brukes på dataene i etterkant slik at nøyaktigheten i målingene kommer innenfor de kravene Kitron har satt.

For å spare batteriet skrur Kitron av GPS mottakeren mellom hver gang en posisjon beregnes. Tiden GPS mottakeren står på er forhåndsprogrammert i dyreenheten og om gyldig posisjon ikke er funnet i løpet av denne tiden vil likevel GPS mottakere slås av. En årsak til at det blir så få gyldige posisjonsmålinger kan være at GPS mottakere er på i for korte perioder av gangen.

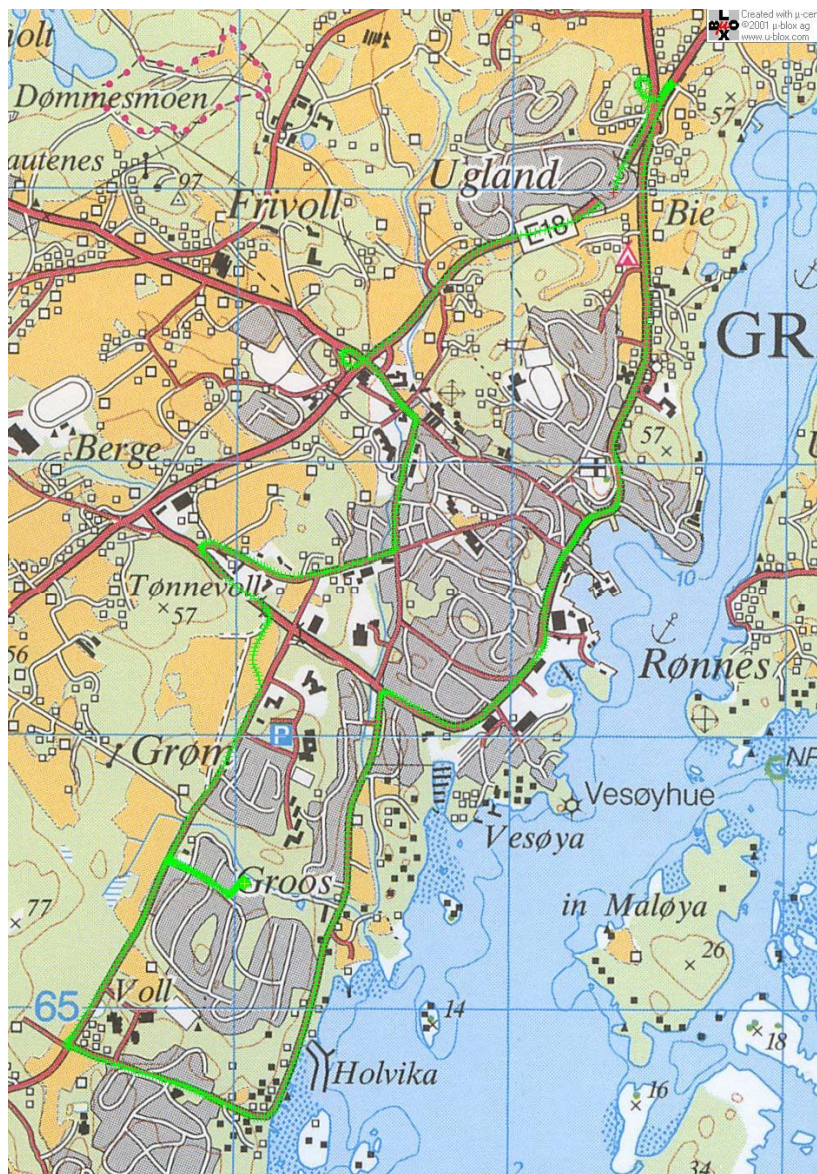
Tabell 1.4 viser hvor mange prosent av målingene fra de forskjellige testområdene som er gyldige. Gammeldalen øst for Tynset er et kupert område med mye skog mens Holtålen nord for Røros er åpent fjellområde. At GPS mottakeren ikke oppnår bedre resultat i Holtålen enn i Gammeldalen kan tyde på at terrenget spiller liten rolle i hvordan resultatene blir. Det er derfor nærliggende å tro at det er andre årsaker til problemene enn terrenget. Vi kan derfor tenke oss at det er andre ytre påvirkninger som spiller en rolle for hvor gode målingene blir. Dette kan være årsaker som Kitron vanskelig kan gjøre noe med, for eksempel vegetasjon, sauens ull og GPS antennens posisjon når sauen sover eller hviler. Om dette er tilfelle kan løsningen være å lage en algoritme for å forbedre resultatene.

Tabell 1.4: Kitrons målinger

Lokasjon	Antall ugyldige målinger [%]
Bygland	69 %
Gammeldalen	75 %
Holtålen Killindalskleppen	80 %
Holtålen Litjscarven	69 %
Vestmarka	96 %

Kitron har imidlertid i ettertid i et møte med GPS leverandøren Trimble funnet ut at de har en designmessig feil i sin konstruksjon av GPS antennen. Det skal være at jordplan under GPS antennen for at denne skal fungere optimalt. Trimble sin representant Tomas Rehn mente at dette kan være en betydelig årsak til at GPS mottakerne ikke finner gyldig posisjonsmåling i så stor grad som de skal. Han viste i denne sammenhengen til en intern undersøkelse som var utført hos Trimble.

For å undersøke om standard GPS utstyr er i stand til å levere den nøyaktigheten Kitron ønsker, benyttet vi mottakeren vår på en kjøretur rundt i Grimstad området. Koordinatene GPS mottakeren beregnet ble plottet inn på et standard Norgeskart med målestokk 1:50000. Som vi ser av figur 1.5 gir GPS mottakeren god nøyaktighet i forhold til oppløsningen på kartet. Dette er selv om GPS mottakeren beveger seg i relativt høy fart og det tidvis er mange objekter som tilsynelatende skulle sperre signalene fra flere satellitter. Hvis en går ut fra at et standard Norgeskart er nøyaktig kan vi konkludere med at nøyaktigheten til vår mottaker tilfredsstillir Kirtons krav. Dette viser at systemet Kitron har utviklet skal være i stand til å beregne sauens posisjon med en nøyaktighet som tilfredsstillir kravene.



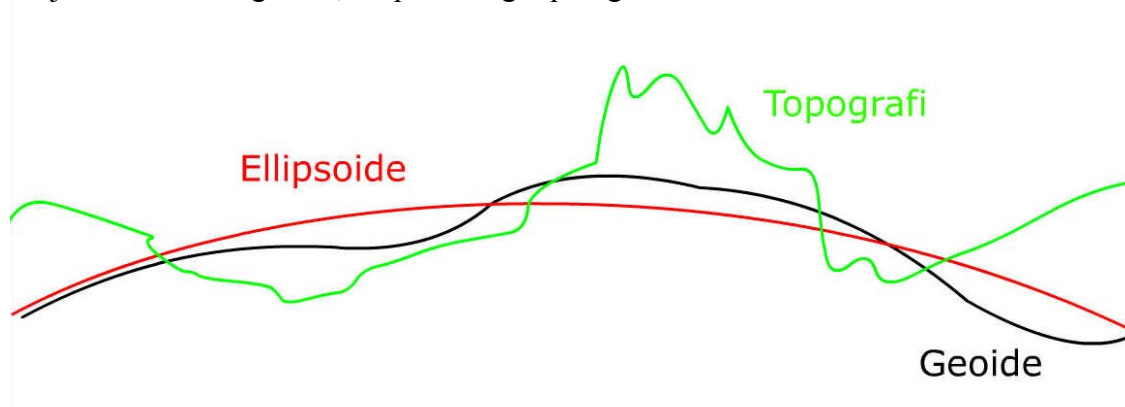
Figur 1.5: Kjøretur rundt i Grimstad

2 Introduksjon til koordinater og koordinatsystem

Meningen med dette kapitlet er å gi en innføring i de forskjellige koordinattypene vi har vært innom i løpet av prosjektet. Koordinatene fra GPS mottakeren oppgis i WGS84 koordinater og koordinatene til referansepunktet vårt ble gitt i NGO formatet. Disse måtte konverteres over til UTM slik at vi kunne behandle dataene i Matlab.

For at GPS systemet skal kunne beregne en nøyaktig posisjon på jorda må det ligge et rammeverk i bunnen som beskriver koordinater og referanser. Geodesi er vitenskapen som omhandler jordas form, gravitasjon og endring. Geodesi kan deles inn i tre hovedfelter: geodetisk posisjonering (landmåling), gravitasjonsfelter og geodynamikk [7].

Et geodetisk system er et rammeverk for å bestemme koordinater på jordas overflate i forhold til en referanse ellipsoide og geoiden [8]. Geoiden er en tenkt flate med lik tyngdekraft overalt, en såkalt ekvigravitasjonsflate. Flaten dannes av verdenshavenes overflate og dens tenkte fortsettelse under landmassene. Dersom hele jordoverflaten var dekket med hav og det ikke var bevegelse i havet eller atmosfæren, ville havoverflaten vært en ekvigravitasjonsflate [7]. En ellipsoide defineres som en lukket, krum flate som av ethvert plan blir skåret i en ellipse [9]. Denne ellipsoiden brukes som referanse for posisjonsberegninger. Ellipsoiden brukes fordi det er en matematisk figur med jevn overflate som er lett å håndtere, i motsetning til geoiden som blir påvirket av ulike tettheter i jordens indre. Ellipsoiden avviker ofte noe fra geoiden. Figur 2-1 viser hvordan forskjellene mellom geoid, ellipsoide og topografi kan være.



Figur 2-1: Forskjellen mellom geoid, ellipsoide og jordens topologi [9]

2.1 GPS posisjonering

Jordens topologi har en veldig uregelmessig karakter. Det er derfor vanskelig å bruke jordoverflaten som referanse for posisjonsberegninger. Det brukes derfor en glatt matematisk modell for å tilnærme jordens uregelmessigheter. Mer presist kan en si at den matematiske modellen prøver å tilnærme jordens gjennomsnittlige havnivå, geoiden.

Et geodetisk datum omfatter størrelse, form og referansepunkt til denne matematiske modellen. Hensikten med et datum er å kunne gi en entydig posisjonsbeskrivelse av ethvert punkt på jorda. Datumet angir også referansepunktet og orienteringen på

koordinatsystemet som benyttes for å angi en posisjon. De fleste datum baserer seg på en ellipsoide som en tilnærming av jordoverflaten. Det finnes mange forskjellige ellipsoider som representerer jordas form, og enda flere datum basert på disse ellipsoidene. Et datum kan for eksempel ta hensyn til en lokal form på jorda i et aktuelt område eller dekke hele jordoverflaten. Det finnes både vertikale og horisontale datum, men når ordet datum blir brukt for seg selv er det som oftest snakk om et horisontalt datum [8].

Et vertikalt datum er en referanse som høyden blir beregnet ut fra. Geoiden blir i de fleste tilfeller brukt som vertikalt datum. Horisontale datum er et sett av parametere som fastsetter en ellipsoide til en spesifikk posisjon på jorda. Punktet hvor ellipsoiden er tilordnet kalles datumpunktet. Det finnes to forskjellige typer horisontale datum; lokale og geosentriske. Et lokalt datum er et sett av parametere som i forhold til et gitt punkt på jorda definerer en ellipsoide som på best mulig måte tilnærmer seg geoiden innenfor et spesielt område.

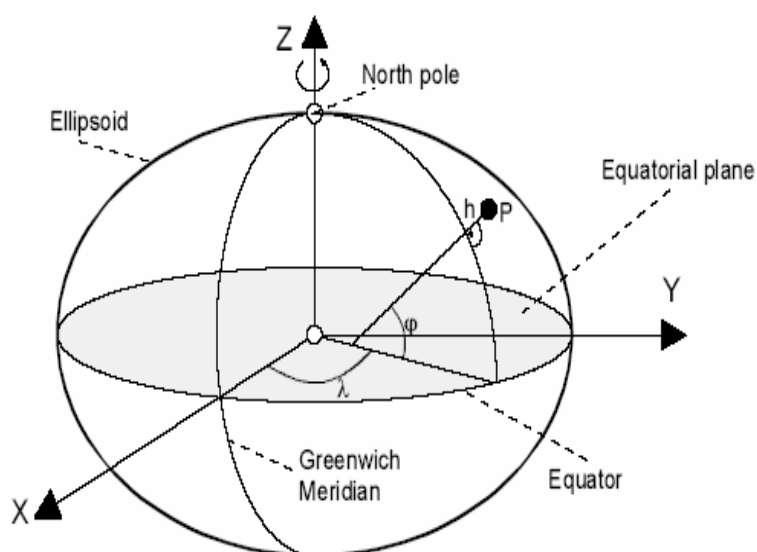
Som oftest brukes fem parametere for å definere et lokalt datum; lengdegrad, breddegrad og geoidhøyden til datumpunktet samt referanse ellipsoidens store og lille halvakse. Et tall for ellipsoiden flattrykning kan også inkluderes. Lokale datum kan være definert for alt fra ørsmå øyer i stillehavet til store regioner av jordkloden. Ved datumpunktet vil avstanden mellom geoiden og ellipsoiden, geoideseparasjonen, være null. Hvis en beveger seg fra dette punktet vil geoideseparasjonen øke.

Et geosentrisk horisontalt datum spesifiserer at referanseellipsoidens sentrum samsvarer med jordens massesentrum. Det trengs minimum åtte parametere for å definere et geosentrisk datum. Tre spesifiserer posisjonen til koordinatsystemets origo, tre spesifiserer koordinatsystemets orientering og to spesifiserer referanse ellipsoidens dimensjon. Geosentriske datum dekker store deler av jordkloden og i enkelte tilfeller hele jorda. Geoideseparasjonen er relativt lav innenfor det området datumet dekker.

Hundrevis av forskjellige datum er blitt brukt for å beregne posisjoner på jorden siden de første estimatene av jordas størrelse ble beregnet av Aristoteles ca. år 300 før Kristus. De første datum som kom ble utviklet for enkelte land og regioner. I Norge er datumene NGO1948 (Norges Geografiske Oppmåling 1948) og ED1950 (Europeisk Datum 1950) lenge vært brukt. Disse er nøyaktige innenfor Norge og Europa og er derfor ikke brukbare for andre deler av jorda. Etter hvert som satellitt teknologien ble tatt i bruk ble det behov for et datum som kunne brukes for hele jordkloden.

2.1.1 WGS-84

Datumet som brukes for GPS posisjonering kalles WGS-84 (World Geodetic System 1984). Det består av et tredimensjonalt kartesisk koordinatsystem og en ellipsoide. Dette er et geosentrisk datum som enten kan angi posisjoner over hele jordkloden med kartesiske koordinater eller med lengdegrad, breddegrad og høyde over ellipsoiden.

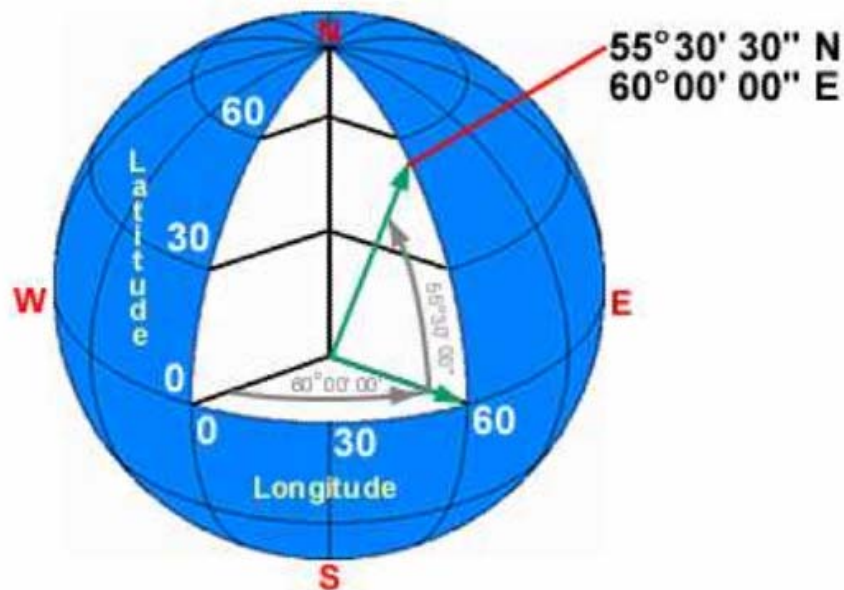


Figur 2-2: WGS-84 koordinatsystem [10]

Ellipsoiden er designet for å tilnærme seg geoiden til jorda på best mulig måte. Dette betyr naturlig nok at den ikke vil være like godt tilnærmet lokalt, slik som et lokalt datum vil være. WGS-84 gir for eksempel et havnivå i Oslofjorden på rundt 40m [9]. Geoideseparasjonen er generelt aldri mer en ca 100 m [8].

Forskjellige koordinatsystem for å spesifisere en posisjon på jordens overflater er blitt brukt i lange tider. Den vanligste måten å angi en posisjon på jorda er å benytte vinkler som måles ut fra jordas sentrum. Det geografiske koordinatsystemet som brukes for å bestemme en posisjon tar utgangspunkt i jordas rotasjonsakse. Ekvatorplanet og det vertikale planet gjennom Greenwich i England, nullmeridianen, er definert som referanser for disse vinklene. Vinkelen mellom et punkt på jorden og ekvatorplanet kalles *breddegrad* (Latitude). En posisjon på ekvator har en breddegrad på 0° , posisjoner nord og sørover angis med en vinkel på mellom 0° og 90° . Vinkelen langs ekvator mellom et punkt på jorda og nullmeridianen kalles *lengdegrad* (Longitude) og angis med en vinkel mellom 0° og 180° . Sammen med vinklene må det angis på hvilken del av jorden en befinner seg. Dette angis med N(nord), S(sør), Ø(øst) og V(vest).

Alle steder på Jorden kan angis entydig ved hjelp av lengde og breddegrader. Grimstad sentrum har f.eks. lengdegrad 8.5931° øst og breddegrad 58.3406° nord. En vektor tegnet fra jordens sentrum til et punkt 58.3° over ekvator og 8.59° øst for Greenwich vil gå gjennom Grimstad. En vanlig måte å angi en posisjon på er med Grader/Minutt/Sekund koordinater. En grad blir delt inn i $60'$ (minutter), og et minutt i $60''$ (sekunder). Mange typer GPS mottakere angir posisjonen med denne type koordinater. I dette formatet vil koordinatene til Grimstad bli: $8^\circ 035'35''$, $58^\circ N20'26''$ (se konverteringseksemplene nedenfor).



Figur 2-3: Koordinater til en posisjon på jordens overflate [9]

2.1.2 Konverteringseksempler

Koordinatene fra vår GPS mottaker inneholder bare grader og minutter, de må derfor transformeres for å få både minutter og sekunder.

Grader, Minutt → Grader, Minutt, Sekund

Koordinatene til HiA Grimstad gitt i Grader/Minutt:

Breddegrad: $58^{\circ}20.2548'$

Lengdegrad: $8^{\circ}34.8597'$

Koordinatene til HiA Grimstad gitt i Grader/Minutt/Sekund:

Breddegrad: $58^{\circ}20.2548' \rightarrow 58^{\circ}20' + 0.2548' \cdot \frac{60''}{1'} = \underline{\underline{58^{\circ}20'15''}}$

Lengdegrad: $8^{\circ}34.8597' \rightarrow 8^{\circ}34' + 0.8597' \cdot \frac{60''}{1'} = \underline{\underline{8^{\circ}34'52''}}$

Grader, Minutt, Sekund → Desimal

Koordinatene til HiA Grimstad gitt som desimalverdi:

Breddegrad: $58^{\circ} + 20' \cdot \frac{1^{\circ}}{60'} + 15'' \cdot \frac{1^{\circ}}{3600''} =$
 $58^{\circ} + 0.3333^{\circ} + 0.004167^{\circ} = \underline{\underline{58.3375^{\circ}}}$

Lengdegrad: $8^{\circ} + 34' \cdot \frac{1^{\circ}}{60'} + 52'' \cdot \frac{1^{\circ}}{3600''} =$
 $8^{\circ} + 0.5666^{\circ} + 0.01444^{\circ} = \underline{\underline{8.581^{\circ}}}$

Desimal → Grader, Minutt, Sekund

Koordinatene til HiA Grimstad gitt i Grader, Minutt, Sekund:

Breddegrad:

$$58^\circ + 0.3375^\circ \rightarrow 58^\circ + 0.3375^\circ \cdot \frac{60'}{1^\circ} =$$

$$58^\circ + 20.25' \rightarrow 58^\circ 20' + 0.25' \cdot \frac{60''}{1'} = \underline{\underline{58^\circ 20' 25''}}$$

Lengdegrad:

$$8^\circ + 0.5810^\circ \rightarrow 8^\circ + 0.5810^\circ \cdot \frac{60'}{1^\circ} =$$

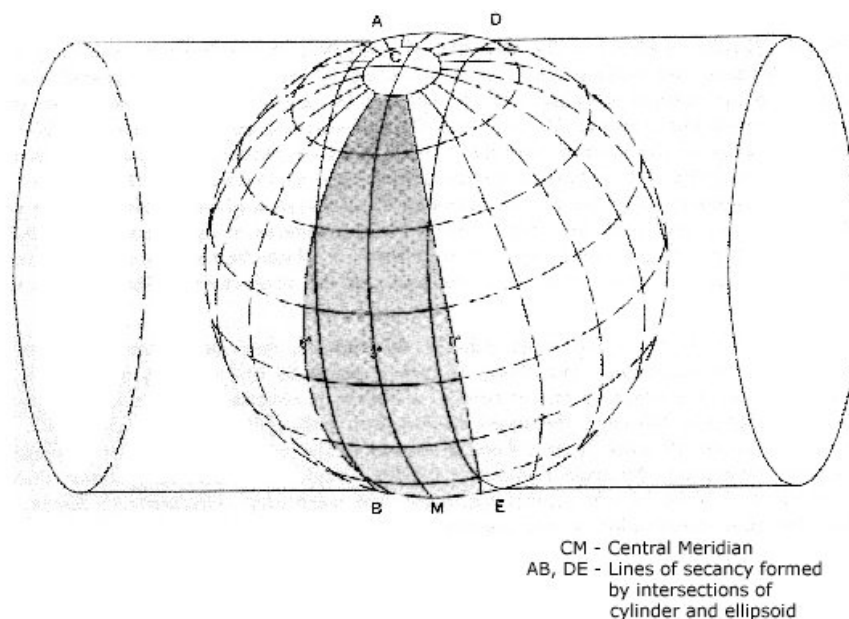
$$8^\circ + 34.86' \rightarrow 8^\circ 34' + 0.86' \cdot \frac{60''}{1'} = \underline{\underline{8^\circ 34' 52''}}$$

2.2 Koordinater og Kart

En utfordring består i å avbilde deler av jordens krumme overflate så eksakt som mulig på et todimensjonalt kart. Teknikken kalles for projeksjon, og beskriver overgangen fra terrenget (3D) til papiret (2D). Et kart er en plan forminsket grafisk avbildning av jordoverflata i en bestemt målestokk. Det gir ikke et korrekt bilde av jorden, men mulighet til å forestille seg hvordan jorden i virkeligheten ser ut. Det finnes mange projeksjonsmetoder, alt etter hvilke egenskaper en ønsker at kartbildet skal få. Hovedmetoder er sylindereprojeksjoner, flateprojeksjoner og koniske projeksjoner. Alle projeksjoner medfører feil av et eller annet slag og feilen slår ut på areal, form, avstand eller retning [11].

2.2.1 UTM

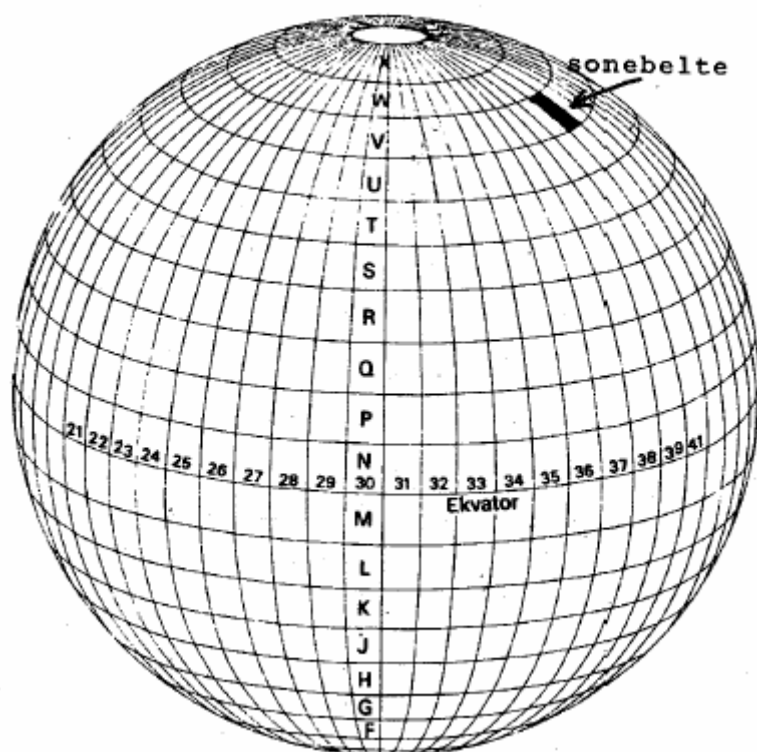
UTM (Universal Transverse Mercator) er en projeksjon som benyttes for å projisere jordoverflaten over til et 2-dimensjonelt kart. Dette er en spesiell sylindereprojeksjon hvor sylindere er i "transversal" stilling, dvs. en liggende sylindere. Denne sylindere kan enten være slik at den tangerer jordoverflaten langs en meridian eller den kan skjære jordoverflaten i en viss avstand fra tangeringsmeridianen. Tangeringsmeridianen kalles da ofte sentralmeridian. Sentralmeridianen og ekvator avbildes som rette linjer. Sylindere som benyttes i UTM er litt mindre enn jord ellipsoiden. Dette fører til at sylindere skjærer jordoverflaten en viss avstand fra sentralmeridianen. Rutenettet er rektangulært, og meridianene har konstant avstand, mens avstanden mellom breddegradssirkulene øker mot polene. Flate og avstandsfeil blir derfor større på høye breddegrader. Området som sylindere "skjærer" ut kalles en sone.



Figur 2-4: Sylindren skjærer jordoverflaten og danner en sone [12]

Problemet med å projisere jordoverflaten over i et plan er at areal, vinkler og avstander ikke stemmer overens med tilsvarende størrelser målt på jordoverflaten. For å få dette til projiseres en liten del av jorda av gangen slik at fortegningen ikke blir så markant. For å få projisert hele jorda anvendes flere sylindere langs forskjellige sentralmeridianer. Hver nye sylinder og sentralmeridian dekker en ny UTM sone. UTM er en projeksjon hvor jorda er delt i 60 soner, hvor hver sone er 6° brede.

Hver sone har sitt eget koordinatsystem med origo på ekvator. 183° øst i forhold til Greenwich meridianen regnes som referanse i UTM systemet. Dette er UTM sone 1. Norge dekkes av sonene 32-36. Sone 32 er utvidet til å ta med hele Sør-Norge fra Trondheim og sørover. Koordinatsystemet i UTM sonene har positiv X-akse nordover langs sentralmeridianen med verdi null for ekvator og enheten er meter [m]. I stedet for X er det vanlig å kalle denne verdien for "Northing". På den nordlige halvkule er X verdiene null ved ekvator og øker nordover. På sørlige halvkule starter X verdien på 10 000 000 ved ekvator og synker mot sør. Y-aksen ligger langs ekvator, og verdien starter på 500 000 ved sentralmeridianen i hver sone. Dette er blitt gjort for å unngå negative koordinatverdier. I stedet for Y- kalles koordinatverdiene ofte for "Easting". For å finne sin posisjon på et verdenskart må en vite om en befinner seg på sørlige eller nordlige halvkule, samt hvilken sone en befinner seg i. UTM systemet omhandler også begrepet belte. Beltene som blir avgrenset av parallell sirkler er på 8° nord/sør, og er blitt gitt bokstaver fra C til X fra sør mot nord. Sone 32 og belte V blir kalt sonebelte 32V.



Figur 2-5: UTM sonebelter [13]

Transversal sylinderprojeksjon fungerer best å bruke når en skal projisere landområder som strekker seg i øst-vest retning langs ekvator.

2.2.2 NGO

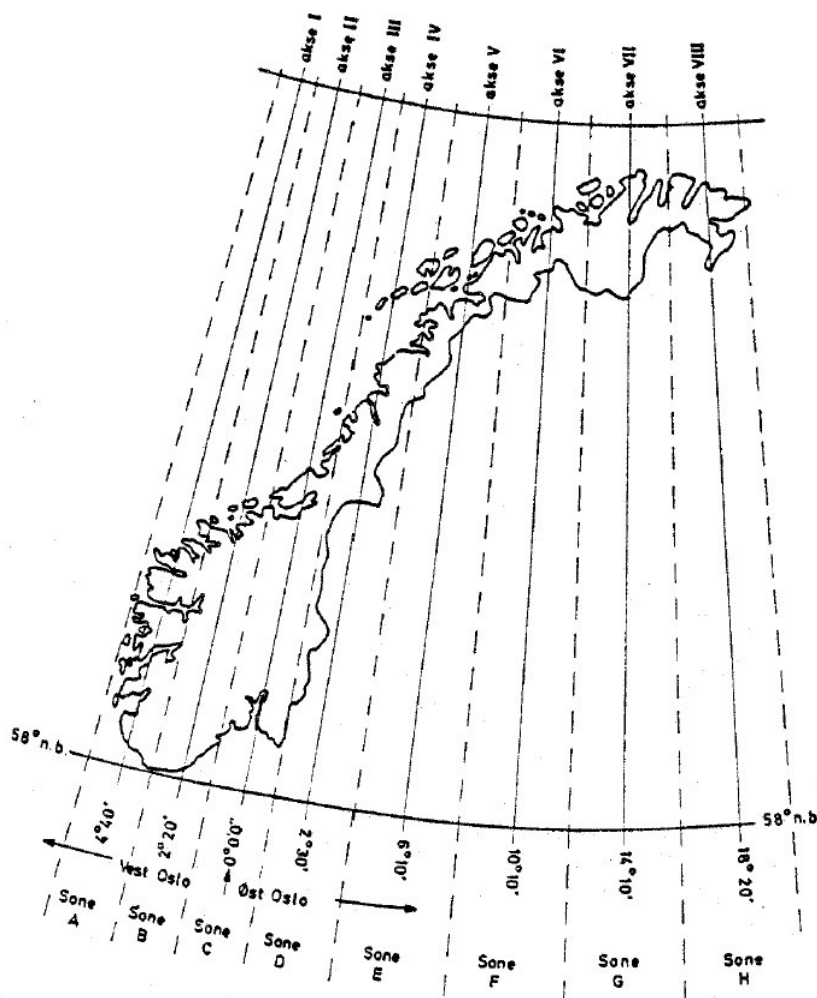
I Norge har det i tiden etter 2.verdenskrig vært benyttet to geodetiske datum, NGO1948 og ED1950. Disse er baserte på liggende sylinderprojeksjoner. Projeksjonene har ulike størrelse og form på jordkloden og projeksjonssylindrene er plassert forskjellig.

NGO1948 er et horisontalt datum som kun benyttes i Norge. Datumet ble beregnet av Norges Geografiske Oppmåling rundt 1948, derav navnet NGO1948. Fram til 1992 var dette det offisielle datumet i Norge [11]. NGO1984 er basert på Bessels ellipsoide fra 1848. Akselengdene i den norske Bessel ellipsoiden er ikke identisk med originalen. Den har derfor fått navnet ”norske Bessel Ellipsoide” eller ”Modifisert Bessel ellipsoide” [13].

Tabell 2.1: Forskjell på Bessel og modifisert Bessel ellipsoide [14]

Halvakse	Bessel 1841	Modifisert Bessel	Differanse
A	6377397,15500 m	6377492,0176 m	94,86260 m
B	6356078,96325 m	6356173,5083 m	94,54505 m

Projeksjonstypen som benyttes kalles *transversal Mercator-projeksjon* eller *Gauss-Krüger-projeksjon*. I NGO1948 benyttes en sylinder som akkurat tangerer jordoverflaten. Norge deles inn i 8 soner ved hjelp av 8 forskjellige projeksjonssylindere. Akse III er lagt gjennom Oslo Observatorium på 10°43'22.5" Øst [9], og danner nullmeridianen for systemet. Figur 2-6 viser hvor de forskjellige sonene ligger. Ut fra figuren ser vi at Grimstad ligger innenfor sone 2.



Figur 2-6: NGO1948 systemet [13]

Koordinatene uttrykkes som kartplan koordinater N(nord) og Ø(øst), og angis i meter fra NGO sonens origo. Soneaksenes origo er på 58° nordlig bredde for alle sonene. X-aksen er positiv nordover langs tangeringsmeridianen, mens Y-aksen er vinkelrett på denne med positiv retning østover. For å gi en entydig posisjon må man i tillegg til koordinatene N og Ø angi hvilken sone koordinatene gjelder for.

2.3 Usikkerhet ved koordinat transformasjon

Å transformere koordinater mellom system som tar utgangspunkt i samme datum er matematisk eksakt og kan gjøres uten tap av nøyaktighet [15]. Å transformere mellom system som benytter forskjellige datum er ikke eksakt og ikke alltid trivielt. Transformasjonen er basert på erfaringsformler gjeldende for et avgrenset område. Datumtransformasjon kan ikke gjøres uten tap av nøyaktighet [16].

Siden koordinattransformasjon ofte er en stor prosess valgte vi å benytte oss av Statens Kartverks konverteringskalkulator, Norgesglasset [17], til å transformere koordinater fra NGO til UTM. Statens Kartverk oppgir en unøyaktighet på 0,5 meter når det transformeres fra et datum til et annet. Matlab har innebygde funksjoner for å transformere fra WGS84 til UTM. Nøyaktigheten i disse transformasjonene er ikke oppgitt.

3 Introduksjon til GPS

3.1 Innledning

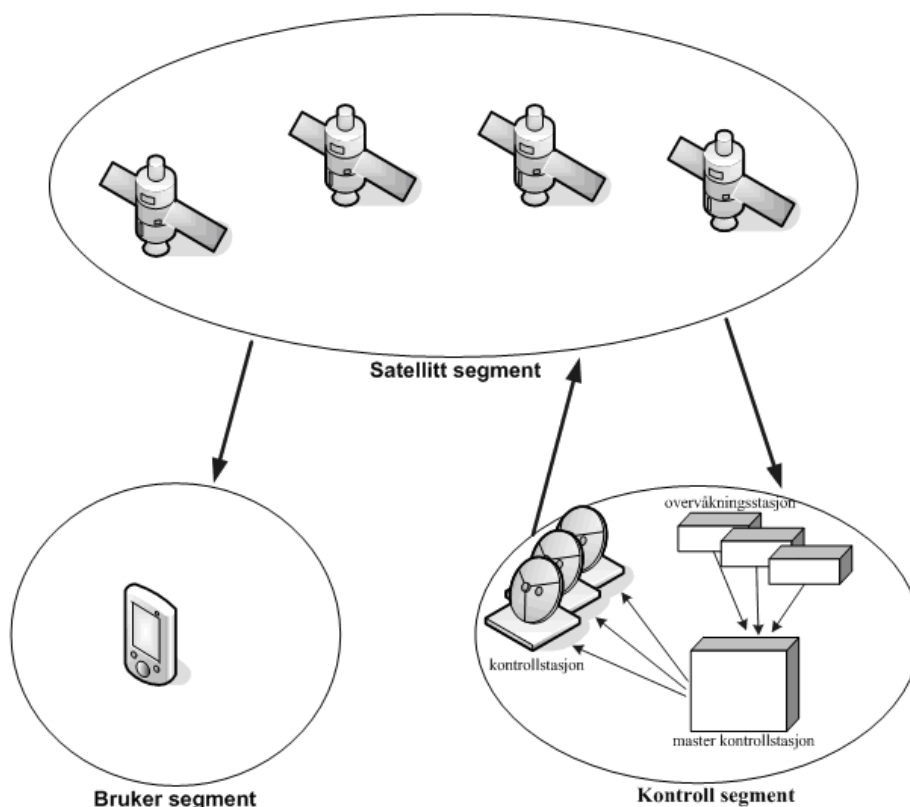
Den egentlige betegnelsen på GPS er NAVSTAR-GPS. Dette står for Navigation System with Timing And Ranging Global Positioning System. Siden det vanligvis i faglitteratur bare benyttes betegnelsen GPS, kommer også vi til å benytte denne betegnelsen. Dette satellittbaserte navigasjonssystemet er utviklet av det amerikanske forsvarsdepartementet (US Department of Defense, DoD). Den første satellitten ble skutt opp i 1978 og i 1985 var det komplett med de 24 satellittene som trengs for å dekke hele jordoverflaten. Det er i dag 29 satellitter [18] som går i bane rundt jorden. Det betyr at det er 5 satellitter flere enn det som strengt tatt er nødvendig.

Systemet er delt mellom det sivile og det militære. Det sivile signalet kalles Standard Positioning Service (SPS) og kan benyttes gratis av alle. Det militære signalet kalles Precision Positioning Service (PPS) og kan kun benyttes av DoD eller personer godkjent av DoD.

Det er flere viktige grunner til at GPS er blitt mer populært med tiden. Tidligere var det kun det militære som fikk dataene som ga best posisjonsbestemmelse. For det sivile markedet benyttet man en teknikk man kaller Selective Availability (SA) som gir dårligere posisjonsbestemmelse. Dette fikk man til ved å legge inn feil i klokke-dataene som ble sendt ut til GPS mottakerne. Men i mai 2000 ble det bestemt fra det hvite hus og president Bill Clinton at SA skulle slås av. Dette medførte at nøyaktigheten ble mer enn ti ganger bedre enn det var med SA. USA har imidlertid beholdt seg retten til å skru på SA igjen om de føler at deres nasjonale sikkerhet er truet [19].

En annen viktig grunn er E911 direktivet som er bestemt i USA. Hver tredje nødsamtale kommer i dag fra en mobiltelefon og man tror dette bare vil øke [20]. Dette er grunnlaget for E911 direktivet som skal gjennomføres i to faser. Fase I skal gi mottakeren av nødsamtalene telefonnummer til mobiltelefonen og lokasjonen til basestasjonen mobiltelefonen benytter. Fase II skal gi en mer presis lokasjon på mobiltelefonen med en nøyaktighet på 50-100 meter. Det er her sett for seg at mange leverandører vil benytte seg av GPS mottaker i mobiltelefonene for å kunne gi en nøyaktig posisjon. Dette vil føre til en økt masseproduksjon av GPS mottakere som igjen vil føre til billigere GPS mottakere. Fase II ble innført i 2001 og skal være gjennomført i løpet av 2005.

GPS systemet består av tre segmenter. Disse er satellitt segment, bruker segment og kontroll segment. Disse er illustrert i figur 3-1.

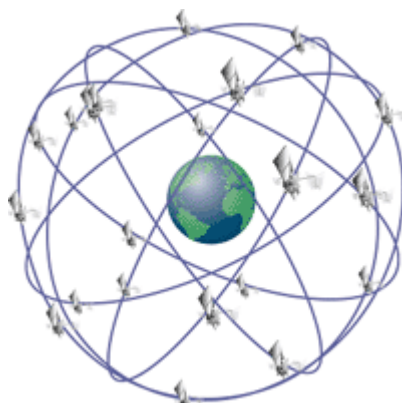


Figur 3-1: Satellitt segment, bruker segment og kontroll segment

3.2 Satellitt segmentet

GPS systemet består av minst 24 satellitter. Satellittene er bygd for å ha en levetid på 10 år og man har derfor skutt opp noen flere satellitter slik at det ikke skaper noen problemer når en satellitt skal skiftes ut. Det er nå 29 Block II/IIA/IIR satellitter som går i bane rundt jorden [18]. Banen satellitten går i har en høyde på 20180 km og en vinkel på 55° i forhold til ekvator. Omløpstiden er i underkant av 12 timer, men tatt i betraktning at jorden også snurrer rundt sin egen akse vil en runde rundt jorden ta rett i underkant av 24 timer.

Figur 3-2 viser hvordan satellittene går i bane rundt jorden. De er blitt plassert i bane på en slik måte at man til en hver tid skal kunne få kontakt med minst fire satellitter uansett hvor man oppholder seg på jordkloden [10].



Figur 3-2: Satellittene i bane rundt jorda [21]

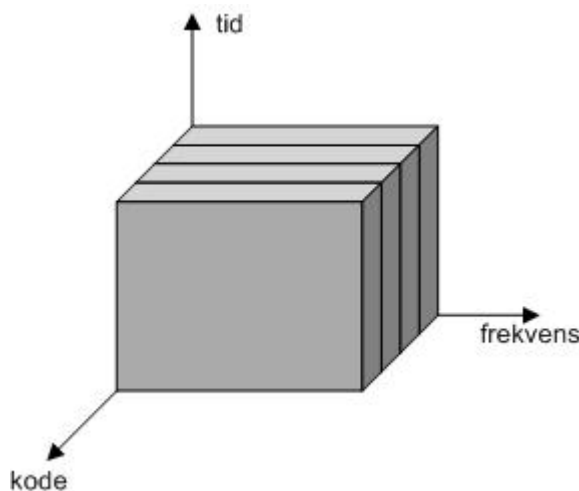
Alle satellittene inneholder inntil 4 atom klokker. Disse klokkene blir korrigert av kontroll segmentene på jorden. Det er svært viktig at klokkene i satellittene til en hver til er korrekt da en feil på 1µs kan gi en feil i posisjonsmålingen på 300m.

3.2.1 GPS signalet

GPS satellittene sender på to bærefrekvenser. Disse blir generert av klokkefrekvensen til atomklokken med frekvens på $f_0 = 10.23$ MHz. Bærefrekvensene er $L1 = 1575.42$ MHz og $L2 = 1227.60$ MHz. Disse signalene blir sendt med en veldig lav effekt på 20-50 W [22].

L1 signalet blir modulert av to koder. Disse kalles C/A-kode (Coarse/Aquisition) og P-kode (Precise). For å hindre at disse signalene interfererer med hverandre er den ene faseforskjøvet 90° . L2 blir bare modulert av P-koden.

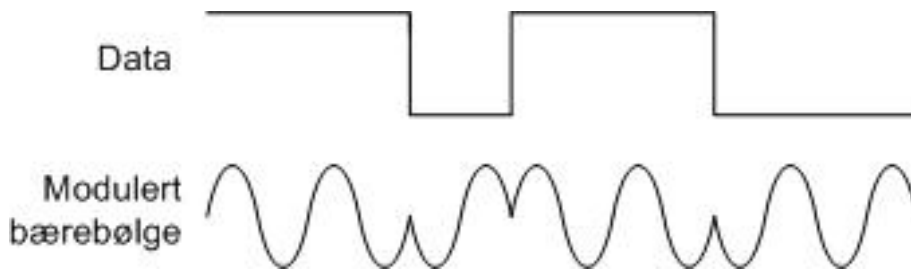
C/A-koden er 1023 bit lang, noe som tilsvarer at koden blir repetert hvert millisekund. C/A koden blir addert sammen med dataene. Resultatet modulerer fasen til bæresignalet. Siden alle satellittene har forskjellig C/A-kode og disse kodene har gode korrelasjonsegenskaper vil man kunne skille signalene fra de forskjellige satellittene fra hverandre. Denne teknikken kalles CDMA (Code Division Multiple Access) og er illustrert i figur 3-3. Disse kodene er lagt inn i vanlige sivile GPS mottakere og gir tilgang til SPS



Figur 3-3: CDMA

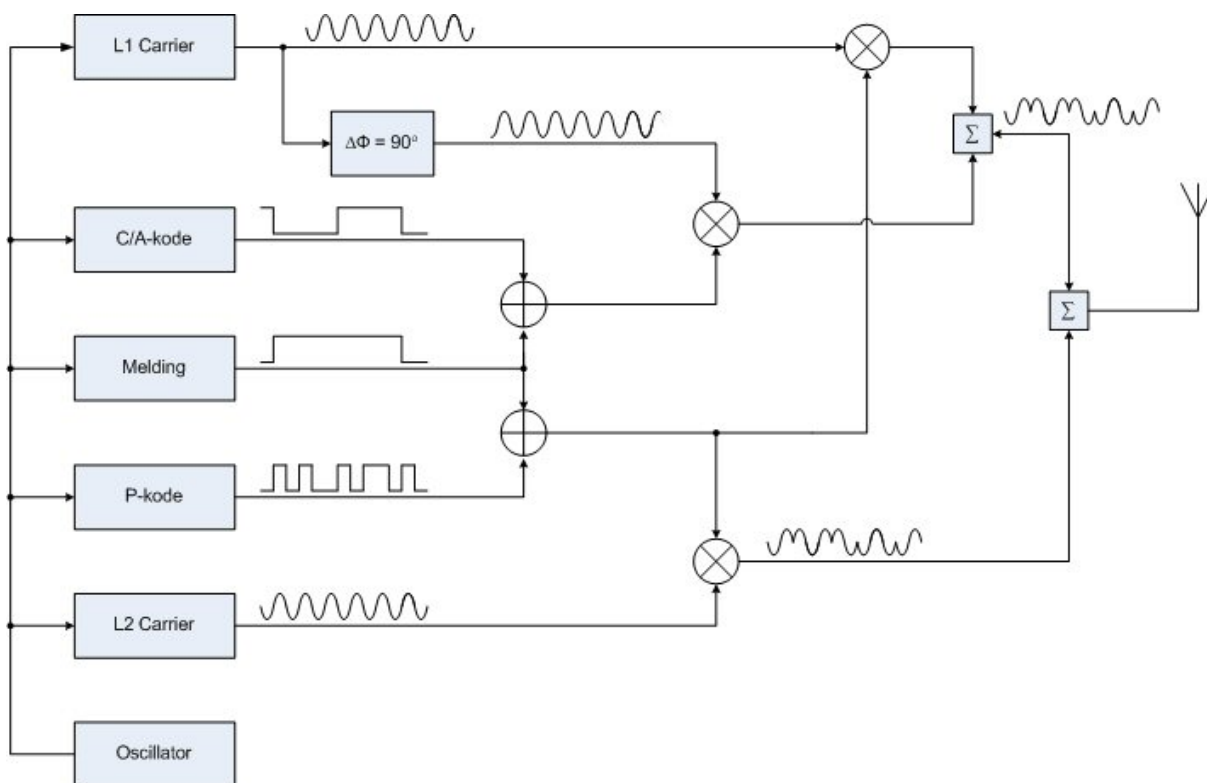
P-koden er en lengre kode enn C/A-koden og blir repetert hver uke. Denne blir modulert på samme måte som C/A-koden, altså ved hjelp av CDMA. Når anti-spoofing (A/S) er aktivert vil P-koden bli kryptert og resultatet kalles Y-kode. A/S ble implementert på permanent basis i 1994 [23]. P-koden (eller Y-koden) gir tilgang til PPS og er lagt inn i militære GPS mottakere.

For å modulere kodene på bærebølgen benyttes modulasjonsteknikken binary phase shift keying (BPSK). Hver gang dataene skifter tilstand får bærebølgen et faseskift på 180° . BPSK er illustrert i figur 3-4.



Figur 3-4: BPSK

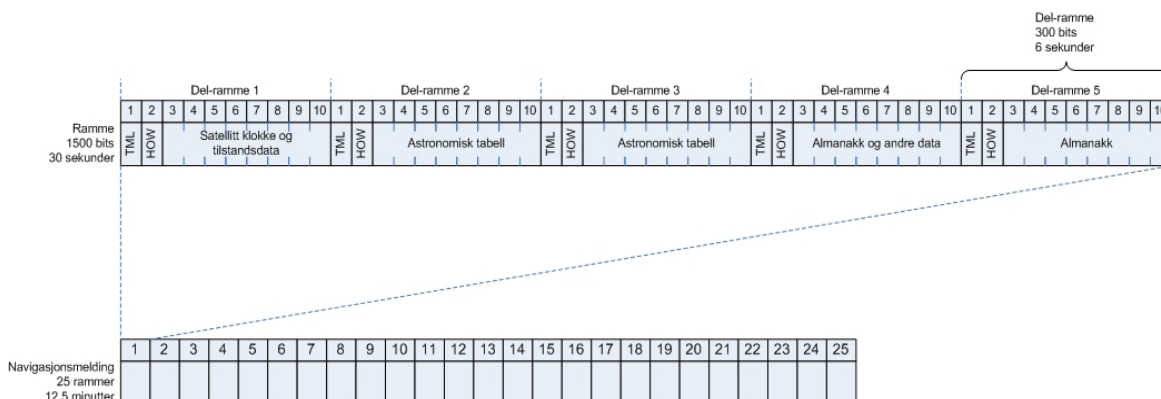
Det endelige blokkdiagrammet for signalet som blir sendt fra satellitten er gitt i sin helhet i figur 3-5.



Figur 3-5: GPS signalet [24]

3.2.2 Navigasjonsmelding

Navigasjonsmeldingen er den meldingen som blir sendt fra satellittene til GPS mottakerne. Overføring av en navigasjonsmelding fra satellitten sendes på frekvens L1 og L2 på 50 bps og overføringen tar 12.5 minutt. Navigasjonsmeldingen er illustrert i figur 3-6.



Figur 3-6: Navigasjonsmelding

En ramme inneholder fem delrammer. De tre første delrammene er for den spesifikke satellitten, mens de to neste inneholder informasjon om de resterende satellittene.

Alle delrammene starter med TLM (Telemetry word) og HOW (Handover Word). TLM meldingen inneholder en 8 bit sekvens for synkronisering, 16 reserverte bit og 6 paritetsbit. Det neste ordet i en delramme er HOW. HOW inneholder 17 bit TOW (Time of Week), 7 bit ID for delrammen og 6 paritetsbit. TOW er en teller som øker med en hvert sjette sekund og blir nullstilt 00:00:00 hver søndag.

Delramme 1 inneholder nøyaktig tidsinformasjon om når signalet ble sendt og parametere for å korrigere tidsforsinkelsen og klokken i GPS mottakeren. Siden denne meldingen blir repetert hvert 30 sekund er dette hyppigheten man mottar nøyaktig tidsinformasjon. For tidsinformasjon mellom dette ser man på hvor mye signalet er forskjøvet i forhold til når man forventet å motta signalet. I tillegg inneholder denne delrammen informasjon om satellittens tilstandsdata samt et ukenummer.

Delramme 2 og 3 inneholder astronomisk tabell for spesifikke satellitten. Dette er satellittens meget nøyaktige banedata.

Mens delramme 1-3 blir repetert hvert 30 sekund, blir delramme 4 og 5 repetert hvert 12,5 minutt. Grunnen til dette er at det er 25 forskjellige versjoner av delramme 4 og 5. Delramme 5 inneholder almanakk og tilstandsdata til satellitt 1 til 24 samt almanakk referansetid og ukenummer. Delramme 4 inneholder almanakken til satellittene 25 til 32, data for ionosfæren og UTC (Coordinated Universal Time) data. Almanakken er mindre detaljerte data om klokke og astronomisk tabell for alle satellittene. Dataene for ionosfæren benyttes til å regne ut tidsforsinkelsen som ionosfæren forårsaker [kapittel 3.6.3]. UTC data benyttes for å kunne relatere GPS tid til UTC tid, samt informasjon om fremtidig planlagte og nylig skjedd endringer i UTC tid.

3.3 Kontroll segmentet

Kontroll segmentet består av tre forskjellige elementer. Dette er master kontrollstasjon, overvåkningsstasjon og kontrollstasjon.

Det finnes fem overvåkningsstasjoner. Disse er plassert rundt hele jorden i nærhet av ekvator. Med denne plasseringen vil alle satellittene være overvåket 92 % av tiden [26].

Disse enhetene overvåker alle satellittene de får kontakt med og beregner hvert 1,5 sekund avstand til satellittene. Disse dataene blir samlet og sendt hvert 15 minutt til master kontrollstasjonen.

Master kontrollstasjonen er blitt plassert i Colorado Springs. Denne har som oppgave å motta data fra overvåkningsstasjonen og beregner satellittenes bane og klokke parametere. Ut fra dette avgjør kontrollstasjonen om det er behov å ta aksjon for å få satellitten inn i den opprinnelige banen eller å korrigere klokken. Denne informasjonen blir så sendt til en av de tre kontrollstasjonene for å videresendes til satellittene.

Kontrollstasjonene finnes det tre av og deres oppgave er å ha kommunikasjon med satellittene. Disse videresender informasjonen de har mottatt fra master kontrollstasjonen. Dette skjer en til to ganger i døgnet per satellitt. Hvis satellittene ikke får kontakt med master kontrollstasjonene har de imidlertid mulighet til å beregne disse dataene selv ved hjelp lagrede data. Dette kan være fra 14 dager for den eldste typen satellitter (Block II) til over 180 dager for de nye typene satellittene (Block IIR) [23]. Nøyaktigheten vil imidlertid bli dårligere etter hvert som tiden går uten kontakt med en kontrollstasjon.

3.4 Bruker segmentet

Bruker segmentet inkluderer militært og sivilt utstyr som mottar og kan nyttegjøre seg av GPS signaler. GPS mottakere er installert i de fleste typer militært utstyr, som f.eks jagere, stridsvogner og personlig utrustning. Den største delen av brukersegmentet er den sivile delen. Det brukes GPS mottakere både i skip, fly og biler for navigering. Etter hvert blir GPS mottakerne bedre, billigere og mindre og kan implementeres i all slags elektronisk utstyr.

3.4.1 Kald, varm og hotstart

Tiden GPS mottakeren bruker for å gi første posisjon varierer avhengig av hvilken informasjon som er blitt lagret siden forrige gang GPS mottakeren var på og hvor lenge det er siden sist den var på. Det er tre forskjellige måter å starte GPS mottakeren på:

Ved kaldstart har GPS mottakeren ingen kjennskap til siste posisjon, tidspunkt eller astronomiske tabell for satellittene. Det eneste den har kjennskap til er almanakken. Dette er den oppstarten som tar lengste tid.

Ved varmstart har GPS mottakeren kjennskap til den siste posisjonen, omtrentlig tid og almanakken. Dette hjelper GPS mottakeren til å finne satellittene raskere og kan dermed også gi en posisjon raskere enn ved kaldstart.

Ved hotstart har GPS mottakeren ikke vært av i mer enn to timer og kjenner den siste posisjonen, omtrentlig tid og almanakken. Og siden GPS mottakeren ikke har vært avslått i mer enn to timer vil man regne den astronomiske tabellen til satellittene som gyldig. Dette gir den raskeste oppstarten.

3.4.2 NMEA-0183

NMEA-0183 er en standard laget av National Marine Electronics Association. Denne standarden kom i 1983 og er laget for seriell bus med en hastighet på 4800 baud. Denne protokollen benyttes til å sende data til og fra GPS mottakeren.

Alle meldinger starter med et \$-tegn. De to neste tegnene viser til hva slags utstyr vi har med å gjøre. For eksempel er GP en GPS mottaker og WI er et vær instrument. De tre neste tegnene angir meldingstype. Deretter kommer alle dataene, som er avskilt med komma. Når det er slutt på dataene kommer det en * etterfulgt av en sjekksum for hele meldingen. Meldingen blir deretter avsluttet med <CR><LF>.

Under er et eksempel på en GPGGA melding med tilhørende forklaring i tabell 3.1.

```
$GPGGA,151738.264,5820.2542,N,00834.8598,E,1,07,1.4,62.3,M,,,,0000*3E<CR><LF>
```

Tabell 3.1: GPGGA meldingsformat [25]

Felt	Beskrivelse
\$	Start på melding
GP	Viser at informasjonen kommer fra en GPS mottaker
GGA	Angir meldingstype
151738.264	UTC tid: 15:17:38,264
5820.2542	Breddegrad: 58° 20,2542 minutter
N	Nordlige breddegrader (N = nord, S = sør)
00834.8598	Lengdegrad: 8° 34,8598 minutter
E	Østlige lengdegrad (E = øst, W = vest)
1	GPS modus, 0 = ingen GPS, 1 = SPS, 2 = DGPS, 3 = PPS
07	Antall satellitter GPS mottakeren har kontakt med
1.4	HDOP
62.3	Høyde over ellipsoide
M	Måleenhet for høyden
	Høydeforskjell mellom ellipsoide og geoide
	Måleenhet for høydeforskjellen
	Alder for DGPS data
0000	Identifikasjon til DGPS basestasjonen
*	Separator for sjekksum
3E	Sjekksum for hele meldingen
<CR><LF>	Slutt på meldingen

For å starte GPS mottakeren med varm eller kaldstart benyttes meldingen Navigation Initialization Data Format (PSRF101). Ved varmstart trenger GPS mottakeren posisjon, TOW og uke nummer. Under er et eksempel på en PSRF101 melding hvor vi har benyttet varmstart med tilhørende forklaring i tabell 3.2.

```
$PSRF101,3318591,500767,5405543,0,306500,243,12,3*13<CR><LF>
```

Tabell 3.2: Navigation Initization Data Format [25]

Felt	Beskrivelse
\$	Start på melding
PSRF101	Meldingstype
3318591	X koordinat
500767	Y koordinat
5405543	Z koordinat
0	Klokke offset
306500	TOW (Time Of Week)
243	GPS uke nummer
12	Antall kanaler
3	3 ved varmstart og 4 ved kaldstart
*	Separator for sjekksum
13	Sjekksum for hele meldingen
<CR><LF>	Slutt på meldingen

3.5 Beregning av posisjon

Dette delkapitelet er skrevet med utgangspunkt i [10].

Når GPS mottakeren skal beregne sin posisjon vil det være fire ukjente. Disse er lengdegrad, breddegrad, høyde og tidsfeil. Tidsfeilen inngår siden klokken i GPS mottakeren ikke er synkronisert med klokken i satellitten. Man må derfor også beregne tidsfeilen for å finne en nøyaktig måling. Grunnen til at man har valgt å gjøre det på denne måten er at det trengs mindre nøyaktig klokke i GPS mottakeren. Dette vil gjøre den både billigere og mindre av størrelse. Med fire ukjente trenger man kontakt med fire satellitter for å få en posisjonsmåling. I utregningene vil GPS mottakeren benytte de fire satellittene som er plassert lengst fra hverandre på himmelen, noe som vil gi best posisjonsbestemmelse (se kapittel 3.6.4).

Prinsippet for å regne ut posisjonen er at man regner ut distansen til satellittene. Denne distansen regnes ut ved hjelp av transittid og lyshastighet. Distansen vil være radius i en kule rundt satellitten. GPS mottakeren vil da være et sted på denne kulen. Om man lager slike kuler rundt flere satellitter vil GPS mottakeren være i punktet hvor disse kulene krysser hverandre.

Når man beregner posisjonen vil man først finne hvor lang tid signalet har brukt med hensyn på tidspunktet som står i meldingen fra satellitten og tidspunktet denne meldingen ble mottatt av GPS mottakeren. Vi kaller denne $\Delta t_{\text{målt}}$. Dette tallet vil være signalets overføringstid (Δt) samt tidsfeilen mellom satellittens klokke og GPS mottakerens klokke (Δt_0). Δt er forskjellig for alle satellittene siden denne avhenger av avstanden mellom satellitt og bruker. Δt_0 er imidlertid konstant siden alle satellittene har synkroniserte klokker. Ved hjelp av den målte tidsforskjellen kan man regne ut pseudo-avstand (PSR). PSR inneholder den faktiske avstanden (R) samt en feil forårsaket av Δt_0 . Vi kan sette opp følgende likninger.

$$\Delta t_{\text{målt}} = \Delta t + \Delta t_0 \quad (1.1)$$

$$PSR = \Delta t_{m\ddot{a}lt} * c = (\Delta t + \Delta t_0) * c = R + \Delta t_0 * c \quad (1.2)$$

Vi kan videre sette opp likningen for PSR ved hjelp av kartesiske koordinater. Siden vi trenger målinger fra fire satellitter kan vi sette opp fire likninger.

$$PSR_i = \sqrt{(X_{sat_i} - X_{user})^2 + (Y_{sat_i} - Y_{user})^2 + (Z_{sat_i} - Z_{user})^2} + \Delta t_0 * c, \text{ for } i = 1,2,3,4 \quad (1.3)$$

Siden denne likningen ikke er lineær, benyttes Taylor-rekker for å løse likningssettet. Vi benytter her en forenkling av Taylor-rekken med bare første ledd som vist i likning (1.4).

$$f(x) = f(x_0) + \frac{df(x_0)}{dx_0} * \Delta x \quad (1.4)$$

For å benytte denne metoden må vi ha en estimert posisjon som er i nærheten av den virkelige verdien. Vi kaller disse verdiene $X_{estimert}$, $Y_{estimert}$ og $Z_{estimert}$ og man finner disse ved hjelp av likning (1.5). Dette er utregnede koordinater for GPS mottakeren uten å ta hensyn til tidsfeilen.

$$R_{estimert_i} = \sqrt{(X_{sat_i} - X_{estimert})^2 + (Y_{sat_i} - Y_{estimert})^2 + (Z_{sat_i} - Z_{estimert})^2} \text{ for } i = 1,2,3,4 \quad (1.5)$$

Vi benytter så Taylor-rekken som vist i likning (1.4) på likning (1.3). Siden vi har en likning med flere ukjente må vi benytte partiell derivering.

$$\begin{aligned} PSR_i &= R_{estimert_i} + \frac{\partial(R_{estimert_i})}{\partial x} * \Delta x + \frac{\partial(R_{estimert_i})}{\partial y} * \Delta y + \frac{\partial(R_{estimert_i})}{\partial z} * \Delta z + \Delta t_0 * c \\ &= R_{estimert_i} + \frac{X_{estimert} - X_{sat_i}}{R_{estimert_i}} * \Delta x + \frac{Y_{estimert} - Y_{sat_i}}{R_{estimert_i}} * \Delta y + \frac{Z_{estimert} - Z_{sat_i}}{R_{estimert_i}} * \Delta z + \Delta t_0 * c \end{aligned} \quad (1.6)$$

Likningene vi har i (1.6) kan vi nå løse ved hjelp av lineær algebra med hensyn på Δx , Δy , Δz og Δt_0 . Vi får dermed resultatet som vist i likning (1.7)

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta t_0 \end{bmatrix} = \begin{bmatrix} \frac{X_{estimert} - X_{sat_1}}{R_{estimert_1}} & \frac{Y_{estimert} - Y_{sat_1}}{R_{estimert_1}} & \frac{Z_{estimert} - Z_{sat_1}}{R_{estimert_1}} & c \\ \frac{X_{estimert} - X_{sat_2}}{R_{estimert_2}} & \frac{Y_{estimert} - Y_{sat_2}}{R_{estimert_2}} & \frac{Z_{estimert} - Z_{sat_2}}{R_{estimert_2}} & c \\ \frac{X_{estimert} - X_{sat_3}}{R_{estimert_3}} & \frac{Y_{estimert} - Y_{sat_3}}{R_{estimert_3}} & \frac{Z_{estimert} - Z_{sat_3}}{R_{estimert_3}} & c \\ \frac{X_{estimert} - X_{sat_4}}{R_{estimert_4}} & \frac{Y_{estimert} - Y_{sat_4}}{R_{estimert_4}} & \frac{Z_{estimert} - Z_{sat_4}}{R_{estimert_4}} & c \end{bmatrix}^{-1} * \begin{bmatrix} PSR_1 - R_{estimert_1} \\ PSR_2 - R_{estimert_2} \\ PSR_3 - R_{estimert_3} \\ PSR_4 - R_{estimert_4} \end{bmatrix} \quad (1.7)$$

Vi har nå funnet tall for å korrigere den posisjonen vi gjettet på i likning (1.5). Dette gjør vi som vist i likning (1.8).

$$\begin{aligned} X_{estimert_ny} &= X_{estimert_gammel} + \Delta x \\ Y_{estimert_ny} &= Y_{estimert_gammel} + \Delta y \\ Z_{estimert_ny} &= Z_{estimert_gammel} + \Delta z \end{aligned} \tag{1.8}$$

Det vi oppnår er at vi har fått et bedre estimat av posisjonen enn den vi fant fra likning (1.5). Likning (1.7) og (1.8) kan utføres flere ganger til man oppnår at posisjonen er innenfor en ønsket feilmargin, altså at Δx , Δy , Δz og Δt_0 er mindre enn en forhåndsbestemt verdi.

Dette er en forenklet versjon av hvordan utregningene blir utført. GPS mottakeren vil i tillegg ta hensyn til feil forårsaket av f.eks. ionosfæren og troposfæren. Det finnes matematiske modeller som GPS mottakeren benytter for å minimere disse feilene

3.6 Feilkilder

Det er mange faktorer som virker inn på nøyaktigheten til GPS systemet. Posisjonsberegningene blir påvirket av flere feilkilder som i hovedsak kan klassifiseres innenfor tre klasser; de som stammer fra satellitten, de som stammer fra mottakeren og feil som kommer fra signalforplantningen mellom satellitt og mottaker. I de fleste tilfeller vil mottakeren ha implementert metoder for å kompensere best mulig for disse feilene.

Tabell 3.3: Feilkilder ved 1-sigma (68.3%) [10]

Feilkilde	Horisontal feil [m]
Fra satellitt:	
Satellittbane data	2.1
Satellitt klokke	2.1
Fra mottaker:	
Refleksjoner av signalet (Multipath)	1.4
Mottaker beregninger	0.5
Signal forplantning:	
Effekter fra ionosfæren	4.0
Effekter fra troposfæren	0.7
Totalfeil	
Vertikal feil (1 sigma (68.3 %) VDOP=2.5)	12.8
Horisontal feil (1 sigma (68.3 %) HDOP=2.0)	10.2

Det er mange måter å representere nøyaktigheten i GPS systemet. Ofte blir den horisontale feilen uttrykket med standardavvik, såkalte sigma verdier. Betegnelsen sigma er ekvivalent med standardavviket til en variabel. 1 sigma og 2 sigma er et estimat på sannsynligheten for at en hendelse skal oppstå. 1 sigma tilsvarer en sannsynlighet på 68.3 % og 2 sigma tilsvarer en sannsynlighet på 95.4 % [27]. Tallene i tabell 3.3 er 1 sigma verdier og vil inntreffe i 68.3 % av tilfellene.

3.6.1 Feilkilder i satellitt

Overvåkningsstasjonene følger hver GPS satellitt 92 % av tiden [26]. Data fra hver satellitt lagres hos overvåkningsstasjonene og blir videresendt til Master kontrollstasjonen i Colorado Springs, USA. Her blir parametere som beskriver satellittenes bane og

klokkeoffset estimert. Disse parameterne sendes ut til en av de tre kontrollstasjonene som igjen sender dataene opp til satellittene.

Feil i banedata

For at en mottaker skal kunne beregne en posisjon på jorden er den avhengig av å vite nøyaktig posisjon til satellittene den har kontakt med. Satellittene kan ikke beregne sin egen posisjon, derfor blir posisjonen gitt til satellittene ved hjelp av GPS systemets kontrollstasjoner. Hver satellitt har en almanakk som inneholder informasjon om banene til alle 29 satellittene. Disse posisjonene er ikke veldig nøyaktige og kan være gyldige i flere måneder. Almanakken sendes til GPS mottakerne på jorda slik at de til en hver tid vet hvor hver satellitt befinner seg. Formålet med almanakken er å gi mottakerne et estimat på hvor hver satellitt til en hver tid befinner seg. Dette gjør at mottakerne lettere og raskere blir i stand til å beregne en nøyaktig posisjon.

En GPS mottaker bruker ikke dataene i almanakken til posisjonsberegning, til dette trenger mottakeren satellittenes nøyaktig posisjon. Det amerikanske forsvarsdepartementet kan forutsi satellittenes posisjon med høy nøyaktighet. Presise banedata kalles Efemeride. Efemeriden er en astronomisk dagbok som inneholder himmellegemers posisjon i forhold til jorden. Disse dataene er veldig nøyaktige og er kun gyldige i 2 timer [28]. En satellitts efemeride blir beregnet ut fra observasjonene til overvåkningsstasjonene i GPS systemet. Overlappende GPS data fra 4-timers perioder brukes for å estimere satellittenes banedata en time fram i tid. Dette er en operasjon som er vanskelig å få helt nøyaktig. Å modellere alle kreftene som virker inn på en satellitt er vanskelig og dette fører til små feil i den estimerte satellittposisjonen. En mottaker får oppdatere efemeridedata fra en satellitt hvert 30. sekund.

Efemeride feil fra en spesifikk satellitt er lik for alle GPS mottakere på jorden. Siden de forskjellige mottakerne mottar signalet fra satellitten med forskjellig vinkel vil feilen ha forskjellig innvirkning for deres posisjonsberegninger. Banefeil vil typisk påvirke en posisjon med en feil på ca 2 meter [29].

Noen applikasjoner trenger mer nøyaktige banedata enn andre. For å støtte disse applikasjonene gir IGS (International GPS Service for Geodynamics), NGS (U.S. National Geodetic Survey) og Geomatics Canada, ut helt nøyaktige satellitt banedata. Dataene baserer seg på GPS observasjoner fra et globalt GPS nettverk som styres av IGS. De minst presise dataene er tilgjengelig 12 timer i etterkant, mens de mest presise dataene ikke er tilgjengelige før etter 12 dager. Brukerne kan laste ned dataene gratis fra hjemmesiden til IGS

Klokkefeil i satellitt

Det fundamentale med GPS er å beregne avstanden mellom satellitt og mottaker. Dette blir gjort med utgangspunkt i satellittenes klokke. Hver satellitt inneholder fire atomur. Disse klokkene er meget nøyaktige men ikke helt perfekte. Klokkene drifter mellom 1 og 2 sekunder pr 10^{13} sekund, noe som utgjør mellom 8.64 og 17.64 ns pr dag. Denne klokkefeilen kan føre til en posisjonsfeil på mellom 2.59 og 5.18 meter [29].

GPS kontrollsegmentet kontrollerer klokkene i hver satellitt. Hvor mye klokkene drifter blir beregnet og sendt ut sammen med efemeride informasjonen til mottakerne.

Alle brukerne som har kontakt med en og samme satellitt opplever den samme feilen. Ved å benytte seg av korrigeringsinformasjonen som ligger i meldingen fra satellitten kan posisjonsfeilen nesten elimineres. Metoden reduserer imidlertid ikke feilen helt. Etter at metoden er benyttet kan feilen være på noen nanosekunder. En tommelfingerregel sier at 1ns feil tilsvarer en posisjonsfeil på 30cm [29].

3.6.2 Feilkilder i mottaker

Klokkefeil i mottaker

GPS mottakere bruker billige klokker som ofte styres av en quartz krystall. Sammenlignet med klokkene i satellittene blir derfor klokkefeilen fra mottakerklokka veldig stor [30], [31]. Denne feilen kan imidlertid reduseres kraftig ved at den blir behandlet som en egen ukjent parameter i posisjonsberegningprosessen. Noen mottakere bruker eksterne ur for å få posisjonen så nøyaktig som mulig. Å benytte atomur er imidlertid ikke en praktisk løsning siden et atomur veier rundt 20kg [32] og koster mellom 25 000 og 50 000 dollar [33].

Elektronisk referansepunkt i antenne

En GPS antenne omformer energien i signalet fra satellitten til elektriske signaler som mottakeren håndterer. Punktet hvor signalene blir tatt imot i antennen, kalles antennes Fase Senter (Antenna Phase Center). Dette punktet samsvarer som oftest ikke med antennens fysiske midtpunkt, men er avhengig av satellittens posisjon i forhold til mottaker og effekten i det mottatte signalet. Fase senteret kan derfor være forskjellig for de forskjellige signalene som mottakeren tar imot.

Variasjonene i fase senteret er forskjellige fra antenne til antenne, det er derfor vanskelig å lage matematiske modeller for å beregne posisjonsfeilen som genereres. Posisjonsfeilen på grunn av antenne er typisk på 1-2 centimeter [34]. På grunn av denne forholdsvis lille feilen kan dette bidraget sees bort fra i de vanligste GPS applikasjonene.

3.6.3 Feilkilder fra signalforplantning

En GPS posisjon beregnes ut fra avstanden mellom mottaker og satellittene den har kontakt med. Mottakeren beregner avstanden ved å multiplisere tiden signalet har brukt fra satellitten med lysets hastighet. Problemet er at lysets hastighet varierer med forholdene i atmosfæren.

Forsinkelse i Ionosfæren

Tidsforsinkelsen gjennom ionosfæren blir regnet som det største bidraget til den totale posisjonsfeilen. I spesielle perioder med maksimum ionosfærisk aktivitet kan forsinkelsen komme opp i 300ns, noe som kan gi ca 100 meter feil i posisjonen [35]. På grunn av energien i strålingen fra sola blir gassatomene i den øverste delen av atmosfæren ionisert. Ved ionisering løsrives et eller flere elektroner fra et molekyl [36]. Dette resulterer i et stort antall frie negativt ladede elektroner samt et stort antall positivt ladede atomer og molekyler som kalles ioner. Området av atmosfæren hvor gass ionisering skjer kalles ionosfæren, 50 – 1000 kilometer fra jordas overflate.

De frie elektronene innvirker på GPS signalets fart, retning og polarisasjon. Forsinkelsen i ionosfæren er avhengig av antallet frie elektroner signalet treffer, samt at forsinkelsen er

omvendt proporsjonalt med frekvensen opphøyd i andre [37]. Dette betyr at lave frekvenser vil få større tidsforsinkelse enn høye frekvenser, noe som igjen betyr at L2 har større tidsforsinkelse enn L1. Signalet kan også bøyes av slik at retningen endres når det treffer et fritt elektron.

Antall frie elektroner blir angitt av Total Electron Count (TEC), som er et tall på elektron tettheten langs signalets bane. TEC avhenger av forskjellige faktorer [29]:

- **Tid på dagen**, Elektrontettheten er høyere om dagen enn om natten. Høyeste verdi er oftest om ettermiddagen og minste verdi er ofte rundt midnatt. Om natten er ionosfæreforsinkelsen ca fem til ti ganger mindre enn om dagen [38].
- **Tid på året**, Elektrontettheten er høyere om vinteren enn om sommeren.
- **Sol syklus**, Elektrontettheten når sin maksimale verdi ca hvert 11. år. Dette kommer av at aktiviteten av gasssekspløsjoner på sola har en syklus på ca 11 år.
- **Geografisk lokasjon**, Elektrontettheten er relativt ustabil rundt ekvator og ved jordas magnetiske poler [38].

De fleste mottakerne har implementert matematiske modeller som kan redusere effekten fra ionosfæren med omkring 50 % [32]. Feilen som gjenstår kan fremdeles være betydelig. Ved å benytte både L1 og L2 kan en mottaker beregne tidsforsinkelsen for deretter å justere posisjonsberegningen ut fra dette. Differensiering mellom L1 og L2 betyr å subtrahere det ene signalet fra det andre, feilen som er felles for de to signalene blir dermed eliminert. Mottakere som bare benytter en frekvens får derimot problemer med å beregne korrekt posisjon på grunn av tidsforsinkelsen ionosfæren har laget. For GPS mottakeren vil det se ut som signalet har brukt lengre tid enn det egentlig har. Dette oppfattes som om avstanden til satellitten er større enn den egentlig er og fører til at mottakerens posisjon ikke blir helt nøyaktig.

Effekten fra signalavbøyning kan minimaliseres ved å benytte en *cut-off* høydevinkel på 10°-15°. Dette betyr at signal som har mindre vinkel enn 10°-15° i forhold til horisonten, ikke vil bli oppfattet av mottakeren.

Mange modeller for å beregne tidsforsinkelsen i ionosfæren er utviklet for brukere som benytter en enkelt frekvens. Disse kan deles inn i to forskjellige kategorier:

- **“State of the art” modeller**, Disse modellene trenger hundrevis av variabler og koeffisienter, noe som gjør utregning og beregning veldig tungt og gjør dermed modellene uegnet for simulering. Modellene er derimot veldig nøyaktige. Typiske modeller er International Reference Ionosphere (IRI) og Bent modellen.
- **Enkle modeller**, Dette er modeller som bruker ideelle beskrivelser av ionosfærens oppførsel. Disse modellene egner seg bedre til simulering og beregning siden de ikke er så omfattende. Typiske modeller er Klobuchars modell, som benytter koeffisienter som blir sendt i GPS navigasjon meldingen. GPS mottakere som kun opererer med en frekvens kan benytte denne modellen for å beregne effekten ionosfæren har på signalet.

Forsinkelse i Troposfæren

Troposfæren er den delen av atmosfæren som er mellom 0 og 50 kilometer over jordoverflata. I denne delen forsinkes alle signaler under 15 GHz like mye. Dette betyr at en ikke kan kombinere L1 og L2 for å beregne tidsforsinkelsen siden forsinkelsen er like stor. Forsinkelsen i troposfæren er avhengig av temperatur, trykk og fuktighet rundt GPS signalets bane. Signal fra satellitter med forskjellige høydevinkel har forskjellig bane

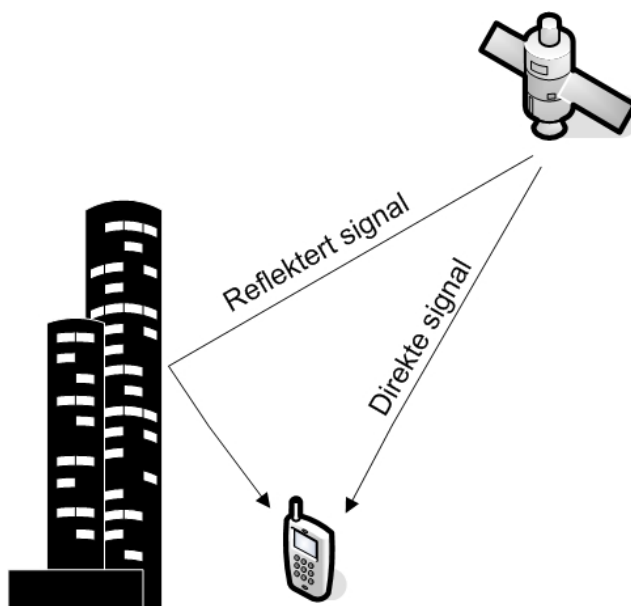
gjennom troposfæren. Den minste tidsforsinkelsen opplever GPS mottakeren fra en satellitt loddrett over seg (senit) og den største forsinkelsen fra satellitter nær horisonten. For satellitter i senit vil posisjonsfeilen være ca 2 meter, mens feilen fra en satellitt med høydevinkel på 15° vil typisk være rundt 9 meter [29].

Forsinkelser fra troposfæren kan deles i to komponenter, tørre og våte. De tørre komponentene representerer 90 % av totalforsinkelsen og kan forutsis med stor nøyaktighet ved hjelp av forskjellige matematiske modeller, samt temperatur og trykk på jordoverflaten. De våte komponentene avhenger av mengden vanddamp signalet opplever gjennom troposfæren. Disse komponentene er i motsetning til de tørre vanskelig å forutsi. Flere matematiske modeller som benytter metrologiske data er laget, men det viser seg at det er lite korrelasjon mellom de våte komponentene og metrologiske data målt fra jordas overflate.

3.6.4 Andre feilkilder

Feil på grunn av refleksjoner

En GPS mottaker har en rundstrålende antenne som tar inn signaler fra alle kanter. Et radiosignal reflekteres fra alle objekter det treffer. Dette fører til at GPS mottakeren kan ta inn ett og samme signal fra en satellitt på forskjellige tidspunkt. Først det som kommer direkte fra satellitten, så alle de reflekterte. Dette kan medføre at GPS mottakeren får problemer med å finne nøyaktig tidspunkt for når meldingen først blir mottatt.



Figur 3-7: Effekten av reflekterte signal

Feil av denne typen påvirker beregningen av avstanden til satellitten, men med nyere og bedre teknologi i mottakerne blir feilene redusert dramatisk. En måte å redusere virkningen av de reflekterte signalene er å benytte en spesiell type antenne som demper disse.

Satellitt geometri

Nøyaktigheten til GPS systemet avhenger også av den geometriske posisjonen til satellittene mottakeren har kontakt med. Hvis en GPS mottaker har kontakt med fire satellitter som alle er plassert nordvest for mottakeren, oppleves satellittgeometrien som

dårlig. Dette ansees som dårlig fordi alle målingene blir gjort fra samme retning. Dette betyr at trianguleringen blir vanskelig og at området satellitten kan befinne seg innenfor blir stort (se figur 3-8).

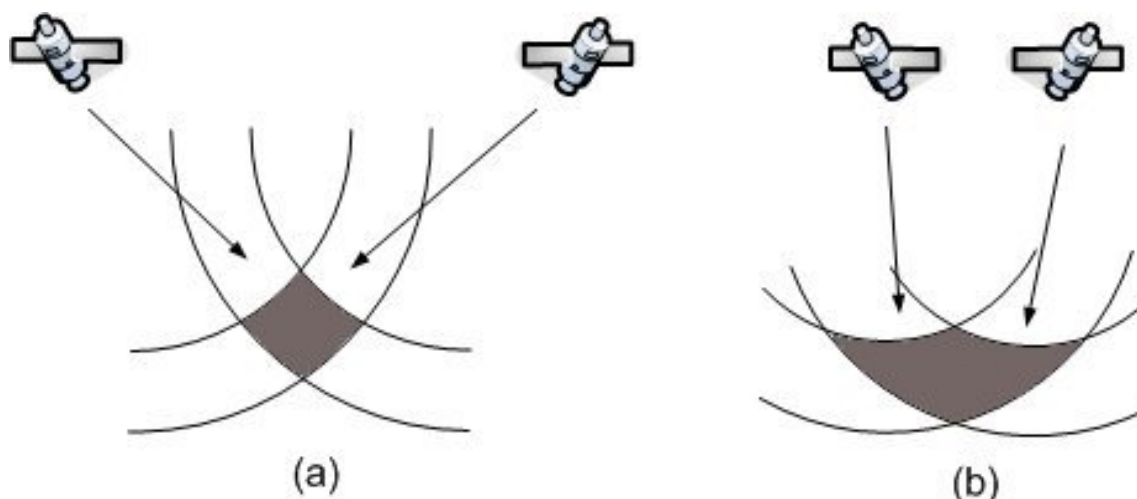
Satellittenes geometriske konfigurasjon angis av DOP (Dilution of Precision). DOP kan defineres som en skaleringsfaktor for GPS posisjonens nøyaktighet basert på mottakerens posisjon samt geometrien til satellittene. Det finnes forskjellige typer DOP:

- PDOP: Positional DOP, verdi som angir nøyaktigheten i GPS mottakerens posisjonsberegning i det tre dimensjonale rom. (Deles i HDOP og VDOP)
- HDOP: Horizontal DOP, verdi som angir nøyaktigheten i GPS mottakerens posisjonsberegning i XY planet, lengde/breddegrad.
- VDOP: Vertical DOP, verdi som angir nøyaktigheten i GPS mottakerens posisjonsberegning i Z planet, høyde.
- TDOP: Time DOP
- GDOP: Geometrical DOP, kombinasjon av PDOP og TDOP.

DOP verdier kan beregnes ved å benytte tilnærmede koordinatverdier, noe som betyr at DOP verdiene kan beregnes uten å gjøre noen målinger.

Nøyaktigheten i posisjonsmålingene er proporsjonale med DOP verdien, dette betyr at feilen i posisjonsberegningen øker med en faktor på to hvis DOP verdien doubles [10]. DOP verdien kan i enkelte sammenhenger være misvisende. Det er derfor viktig at DOP verdier kun blir brukt til å indikere at GPS systemet kanskje ikke er i stand til å produsere nøyaktige nok målinger. En høy DOP verdi vil ikke automatisk si at posisjonen som oppgis av systemet er unøyaktig [39].

God satellitt geometri oppnås når satellittene er spredt over himmelen. figur 3-8 viser et eksempel på satellitt geometri ved bruk av to satellitter. Mottakeren vil være plassert innenfor de grå feltene. På grunn av målefeilene som forekommer vil den beregnede avstanden fra mottaker til satellitt ikke være eksakt. Sirkelbuene på figuren representerer lengste og korteste avstand mellom mottaker og satellitt som GPS mottakeren beregner. Ved å kombinere beregninger fra to satellitter vet en at mottakeren vil befinne seg innenfor de grå områdene. Hvis størrelsen på det grå området er lite vil den beregnede mottakerposisjonen være temmelig presis [29], og DOP verdien vil være lav.



Figur 3-8: (a) God satellitt geometri, (b) Dårlig satellitt geometri

PDOP har størst effekt på posisjonsberegningene, jo høyere verdi desto større feil i den beregnede posisjonen. Verdier på rundt 4 og lavere betyr at posisjonen er relativt nøyaktig. I de fleste tilfeller vil PDOP gjennomsnittlig ligge ned mot to.

Basert på informasjonen i almanakken analyserer GPS mottakeren posisjonen til satellittene som er tilgjengelige. Ut fra denne analysen velger mottakeren de fire satellittene som til sammen gir lavest DOP for å benytte til posisjonsberegninger. En GPS mottaker beregner alltid posisjoner ut fra de fire satellittene som gir best resultat.

3.7 Differensiell GPS

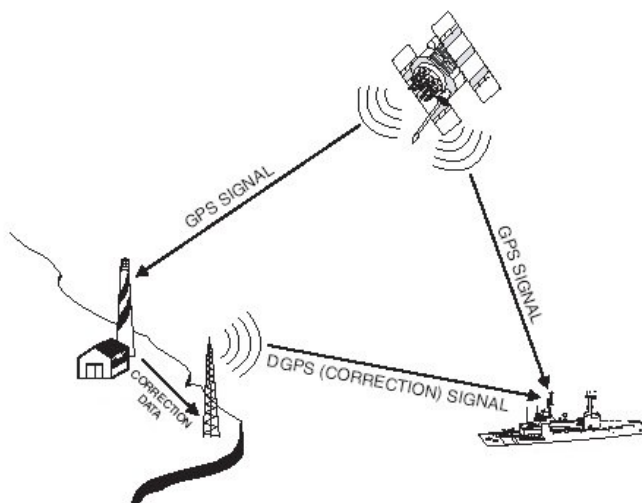
Differensiell GPS (DGPS) er en metode for å øke nøyaktigheten i posisjonene som GPS systemet beregner. DGPS kan gi posisjoner med en nøyaktighet på noen få meter for enheter i bevegelse og enda bedre nøyaktighet for stasjonære enheter [40].

Tabell 3.4: Typiske feilkilders innvirkning på totalfeilen ved bruk av GPS og DGPS [40]

Typiske feil i meter (per satellitt)	Standard GPS [m]	Differensiell GPS [m]
Satellitt klokke	1.5	0
Banefeil	2.5	0
Ionosfære	5.0	0.4
Troposfære	0.5	0.2
Refleksjoner av signalet	0.3	0.3
Støy hos mottaker	0.6	0.6

DGPS benytter to mottakere som samarbeider med hverandre, en stasjonær og en mobil. Den stasjonære mottakeren er nøkkelen i systemet, den knytter alle målingene til et nøyaktig oppmålt fast punkt.

Hver GPS mottaker bruker signaler fra fire satellitter for å beregne en posisjon. Hvert av disse signalene fører til feil i posisjonsberegningene pga. årsakene nevnt i kapittel 3.6. Siden hvert signal som inngår i posisjonsberegningen inneholder feilkilder vil den beregnede posisjonen få en feil som er en sammensetning av feilene fra hvert signal. DGPS systemet går ut på å bruke to mottakere med en viss avstand mellom, f.eks. noen titalls kilometer. Siden satellittene er så langt unna jordkloden og avstanden mellom mottakerne er relativt liten i forhold, kan en si at signalene fra satellittene har gått gjennom samme del av atmosfæren, og derfor blitt utsatt for samme type påvirkning. Feilene i hvert signal vil derfor være tilnærmet lik for de to GPS mottakerne. Den stasjonære mottakeren som i utgangspunktet vet sin posisjon angriper likningene for posisjonsberegning baklengs. I stedet for å beregne en posisjon ut fra signalet fra satellitten, benytter mottakeren sin posisjon til f.eks. å beregne tiden det vil ta å sende en melding fra satellitt til mottaker. Ved å sammenlikne dette resultatet med tiden det tok å sende signalet kommer mottakeren fram til en "feilkorreksjonsfaktor". Denne faktoren sendes til den mobile GPS mottakeren som bruker den til å korrigere sine posisjonsberegninger. Siden referanse mottakeren ikke kan vite hvilke satellitter den mobile mottakeren kommuniserer med, beregner den feilen for alle de satellittene som kan være aktuelle og legger denne informasjonen inn i et standard format og sender det til den mobile mottakeren. Korreksjonsdata kan f.eks. sendes via RDS-systemet (Radio Data System) for FM-signal, eller via andre radiosystem [9].



Figur 3-9: Differensiell GPS [41]

4 Metode

I dette kapittelet vil vi beskrive hvilke metoder vi har brukt for å komme fram til de resultatene som ligger til grunn for drøftingene senere i rapporten.

Dataene Kitron samlet inn i løpet av feltperioden sommeren 2003 viste at ca 70 % av alle målingene de hadde mottatt var ugyldige. Ut fra denne informasjonen samt teori og innspill fra veileder og ansatte hos Kitron satte vi i starten opp noen hypoteser på hva vi trodde problemet kunne være. Det var mest nærliggende for oss å velge en eksperimentell design på oppgaven for å prøve å verifisere disse antagelsene og hypotesene.

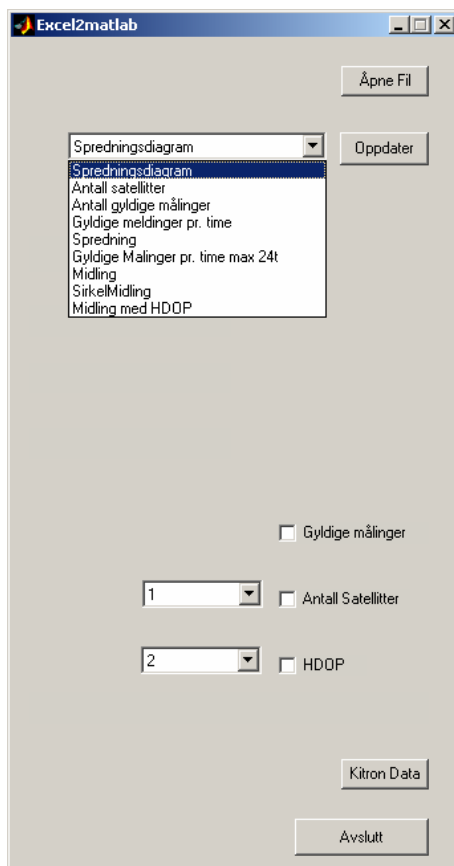
Vi valgte å tilnærme oss problemstillingene på to måter. For å undersøke om det er mulig å behandle ”dårlige” GPS data i etterkant for å få bedre nøyaktighet, satte vi opp en tildekket mottaker. Sauer lever ofte i skog og kratt, dette er omgivelser som kan påvirke posisjonsberegningene i GPS mottakeren. På grunn av at det var vanskelig for oss å gjenskape forsøkene Kitron gjorde under feltperioden, ønsket vi å simulere de forholdene vi tror sauer lever under på best mulig måte. Dette ble gjort ved at vi satte forskjellige stålrør med forskjellig utforming over mottakeren. De forskjellige rørene ga veldig forskjellige utfall med hensyn til hvor mange ugyldige målinger som oppsto. Etter mange forsøk fikk vi til slutt bra likhet med de resultatene som Kitron oppnådde under feltperioden. Ved hjelp av forskjellige metoder prøvde vi å øke nøyaktigheten i disse dataene. Disse metodene er beskrevet i kapittel 4.3.1 til 4.3.3.

Vi lurte også på om de dårlige resultatene Kitron oppnådde kunne ha noe med at GPS mottakeren i dyreenheten fikk for kort tidsom til å beregne en posisjon. For å sjekke hvor lang tid en GPS mottaker trenger for å levere en gyldig posisjon og om denne posisjonen blir mer nøyaktig etter som tiden går, gjorde vi en rekke oppstartsmålinger.

4.1 Matlab testapplikasjon

For å analysere posisjonsmålingen vi gjorde laget vi en testapplikasjon i Matlab. Figur 4-1 viser brukergrensesnittet til testapplikasjonen. Vi benyttet dette programmet til å analysere dataene samt å eksperimentere med de forskjellige algoritmene vi utviklet. I programmet kan brukeren blant annet sette de kriteriene han vil for hvordan en gyldig måling skal være, eller angi radius og antall repetisjoner benyttet i sirkelmidlingsalgoritmen. Applikasjonen viser resultatene i form av spredningsdiagram og histogram.

Appendiks D gir en fyldigere beskrivelse av applikasjonen, samt kildekode.



Figur 4-1: Brukergrensesnitt testapplikasjon

4.2 Nøyaktighetsmålinger

GPS teorien sier at nøyaktigheten til systemet skal være rundt 5 meter. For å få verifisert dette gjorde vi to forskjellige nøyaktighetsmålinger.

Det første vi gjorde var å logge GPS data mens vi kjørte rundt i Grimstad sentrum. Deretter plottet vi disse målingene med et standard Norgeskart med målestokk 1:50000. For å logge dataene bruket vi evalueringstøytøyet μ -Center fra GPS produsenten μ -Blox. Dette programmet plottet de beregnede posisjonene inn på et kart.

Den beste måten å sjekke nøyaktigheten i GPS systemet er å sjekke hver posisjonsmåling opp mot et fast punkt på jorden. Vi fikk Aanon Grimnes ved byggavdelingen på HiA til å måle opp de nøyaktige koordinatene til det punktet vi ønsket å måle nøyaktigheten ut fra. Ved å plote avstanden mellom posisjonen GPS mottakeren beregner og det oppmålte punktet får vi et godt bilde over nøyaktigheten til systemet. Vi valgte å gjøre disse målingene over en periode på ca. 24 timer for da å se om det er store forskjeller i nøyaktigheten i løpet av et døgn. En annen grunn er at satellittene bruker 24 timer på å komme tilbake til samme posisjon sett fra jorda. Dette fører til at satellitt konstellasjonen ”starter på nytt” etter ett døgn. Ved å måle over 24 timer får vi med alle de forskjellige satellitt konstellasjonene.

For å beregne avstander og plote disse brukte vi programmet Matlab 6.5 fra Mathworks. Koordinatene til det faste punktet ble gitt i NGO formatet. Siden Matlab har innebygde funksjoner som benytter UTM formatet, valgte vi å konvertere NGO koordinatene ved

hjelp av Statens Kartverks online konverteringskalkulator, Norgesglasset [17]. En annen grunn til å velge UTM er at denne standarden benytter meter [m]. Det er visuelt enklere å forstå en graf hvor avstanden er gitt i meter, i forhold til at de hadde vært gitt i grader. Koordinatene fra GPS mottakeren er gitt i WGS84 standarden. For å konvertere disse til UTM benyttet vi innebygde funksjoner i Matlab.

4.3 Simuleringer

Vi ønsket å simulere sauens omgivelser så virkelighetslikt som mulig for å sjekke om nøyaktigheten blir dårligere om vi innfører dårligere forhold for GPS mottakeren. Vi vurderte først å foreta målinger i de omgivelsene sauer normalt befinner seg, men gikk bort fra dette av to grunner. For det første ville det vært vanskelig å foreta mange målinger ute i det fri med tanke på batterikapasiteten på våre bærbare maskiner. En annen grunn er at vi foretok våre målinger i perioden februar-april. Skogen har ikke blad på denne tiden av året og vil ikke ”forverre” forholdene i like stor grad som om sommeren/høsten når sauene er på beite.

Vi ønsket å få tilsvarende målinger som Kitron fikk under feltperioden sommeren 2003. For å verifisere at vi hadde fått samme resultat som Kitron sammenlignet vi våre data med dataene fra feltperioden. De eneste parametrene som var sammenlignbare var antall gyldige målinger og fordelingen over hvor mange satellitter mottakeren hadde hatt kontakt med. Ut fra dataene vi samlet inn prøvde vi å komme fram til metoder/algoritmer for å øke nøyaktigheten.

For å drøfte nøyaktigheten var det viktig med et kontrollerbart testoppsett. Ved å sette et stålrør over mottakeren fikk vi målinger med en feilprosent på rundt 60-80% over et døgn. Stålrøret var 3mm tykt, 10,5 cm høyt og 7,5 cm i diameter. For å få forskjellige grader av feilprosent bygget vi opp GPS mottakeren inne i røret med papplater på ca 2 mm. Disse målingene ble tatt på taket av gammelbygget på HiA i Grimstad. Som referanse brukte vi det nøyaktig oppmålte punktet.



Figur 4-2: Plassering av stålrør over GPS mottaker

Vi brukte GPS programmet Visual GPS for å logge målingene. Programmet lagrer posisjonene i en tekstfil. For å analysere dataene ønsket vi å benytte Matlab, dette

verktøyet brukte utrolig lang tid på å importere disse tekstfilene. Vi laget derfor et program i Visual Basic som overførte dataene fra tekstformatet til et Excel regneark som lettere lot seg importere til Matlab. Dette programmet er vist i vedlegg B.

For å øke nøyaktigheten prøvde vi først å sette våre egne krav til hva som skulle være en gyldig posisjon. Et alternativ var kun å godta målinger med lav HDOP. Vi valgte HDOP fordi denne verdien sier noe om den horisontale nøyaktigheten i den beregnede posisjonen. Alternativet var å se på PDOP eller VDOP, men disse verdiene sier også noe om den vertikale nøyaktigheten i posisjonen. For sauens posisjon er ikke dette så interessant. Et annet alternativ var å godta målinger etter hvor mange satellitter GPS mottakeren hadde kontakt med. Vi ønsket å finne ut om nøyaktigheten ble bedre ved å filtrere bort de ”dårligste” målingene.

Vi laget også tre forskjellige algoritmer som vi brukte på posisjonsmålingene. Alle disse algoritmene baserer seg på å ta middelveien av et antall målinger etter forskjellige kriterier. Ved å ta middelveien av et gitt antall målinger oppnår man at en eventuell tilfeldig feil jevnes ut.

Vi valgte å ta middelveien over 15 målinger, noe som tilsvarer en tidsperiode på 15 sekunder. Vi vurderte også å midle over en lengre periode, men med tanke på sauens bevegelser kan en lengre periode gi en veldig feil posisjon. En sau som løper i en bestemt retning med en fart på 25 km/t tilbakelegger en strekning på rundt 400 meter i løpet av 60 sekunder. Å midle alle posisjonene som beregnes i løpet av disse 60 sekundene vil gi et veldig unøyaktig resultat. Ved å velge 15 sekunder vil fortsatt unøyaktigheten bli betydelig, men vi antar at det ikke er så ofte en sau beveger seg så fort i løpet av 15 sekunder. 25 km/t i 15 sekunder kan betegnes som et ”worst case” tilfelle. Lengre midlingsintervaller kan benyttes under spesielle omstendigheter f.eks. når det blir gitt alarm. En alarm kan f.eks. bety at sauen står fast eller er skadet. I disse tilfellene er det veldig viktig å få en nøyaktig posisjon siden dyret må finnes så raskt som mulig. Siden en alarm antyder at dyret er i ro, kan det trykt midles over en lengre periode, f.eks. 60 sekunder.

4.3.1 Midling

En mulig løsning for å oppnå mer nøyaktige posisjonsmålinger er å ta gjennomsnitt av flere målinger. Disse posisjonsmålingene må tas over en kort periode for å være helt sikker på at sauen ikke har beveget seg over større distanser. Vi setter også som betingelse at de målingene vi benytter for å ta gjennomsnitt skal være gyldige posisjonsmålinger. Posisjonen til en ugyldig måling er en repetisjon på forrige gyldige posisjonsmåling og er derfor ikke ønskelig å benytte i gjennomsnittet.

I testapplikasjonen vi laget i Matlab er det muligheter til å velge antall målinger som skal tas og antall målinger av disse som må være gyldige for at gjennomsnittet skal beregnes.

4.3.2 Sirkelmidling

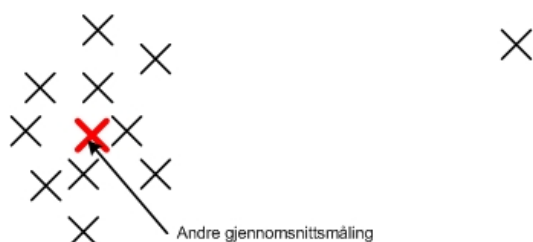
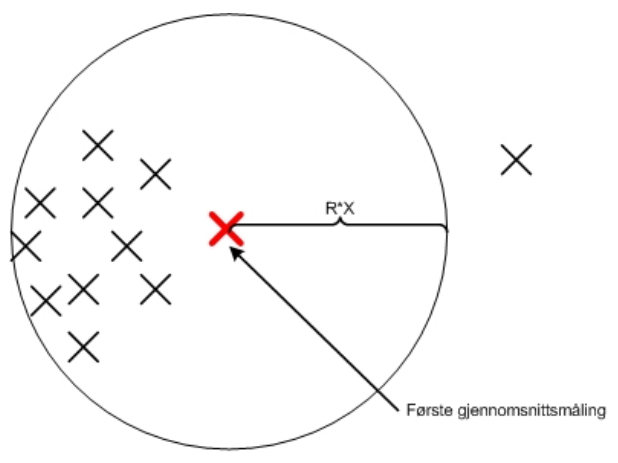
Denne metoden er mer kompleks og krever mer prosessering. Målet er kun å ta gjennomsnittsverdien av målinger som ligger samlet innenfor en gitt distanse og dermed utelukke enkeltmålinger som er havnet et langt stykke unna. For å finne området hvor det

er samlet flest posisjonsmålinger kan vi ikke ta utgangspunkt i en tilfeldig valgt måling. Denne tilfeldige målingen kan ligge langt unna det området hvor hovedtyngden av målingene ligger. Vi mister da muligheten til å regne ut en gjennomsnittsverdi av de målingene som ligger samlet.

Vi prøvde derfor å "sirkle" inn de målingene som er samlet innenfor en på forhånd valgt radius R . Denne operasjonen skal utføres X antall ganger, hvor X må velges av brukeren. Det første som gjøres er at gjennomsnittet av alle de gyldige posisjonsmålingene i måleserien beregnes. Det er svært sannsynlig at denne middelveiden vil ligge nært opptil det området hvor hovedtyngden av målingene befinner seg om GPS mottakeren bare i enkelte tilfeller beregner posisjoner som er langt unna. Dette gjennomsnittspunktet danner så sentrum for en ny midlingsoperasjon. Det midles deretter over de posisjonsmålingene som ligger innefor en radius på $R \cdot X$ fra det første gjennomsnittspunktet. Parameteren X angir hvor mange ganger vi ønsker å gjøre midlingsoperasjonen. Den nye middelveiden brukes som sentrum i en ny midlingsoperasjon, hvor middelveiden av alle målingene som ligger innefor en avstand på $R \cdot (X-1)$ beregnes. Slik fortsetter algoritmen til vi står igjen med målingene som er innefor en distanse på $R \cdot 1$.

På denne måten vil posisjonsmålinger som er langt unna de resterende målingene ikke bli med i beregningen av gjennomsnittsmålingen. Prinsippet på sirkelmidling er vist i figur 4-3.

I applikasjonen er det mulig å angi antall målinger som det skal midles over, antall målinger som må være igjen for at endelig gjennomsnittsmåling skal bli tatt, radius R og antall repetisjoner X .



Figur 4-3: Sirkelmidling, R =Radius, X =antall repetisjoner

4.3.3 HDOP midling

Denne metoden er ganske lik midlingsalgoritmen i kapittel 4.3.1. Men ved denne metoden setter vi ikke bare som betingelse at posisjonsmålingene må være gyldige men de må også ha en HDOP bedre enn en valgt verdi. Kun de posisjonsmålingene som tilfredsstill disse kravene vil bli benyttet til å regne gjennomsnittsverdien.

I applikasjonen vil man ved HDOP midling ha mulighet til å angi antall målinger som skal tas, antall målinger som må tilfredsstill kravene og kravet til HDOP som målingene må tilfredsstill.

4.4 Oppstartsmålinger

Ved gjennomgang av GPS teorien og Radiobjellas spesifisering fikk vi mistanke om at GPS mottakeren har for liten tid til å beregne en posisjon. For å sjekke dette gjorde vi noen oppstartsmålinger under forskjellige betingelser. Det vi ønsket å få svar på var hvor lang tid det tar fra vi slår på GPS mottakeren til den kommer med første gyldige måling. Denne tidsperioden kalles i teorien for TTFF (Time To First Fix). Vi ønsket også å finne ut om den første posisjon som beregnes er nøyaktig nok eller om posisjonen blir mer nøyaktig etter hvert som tiden går.

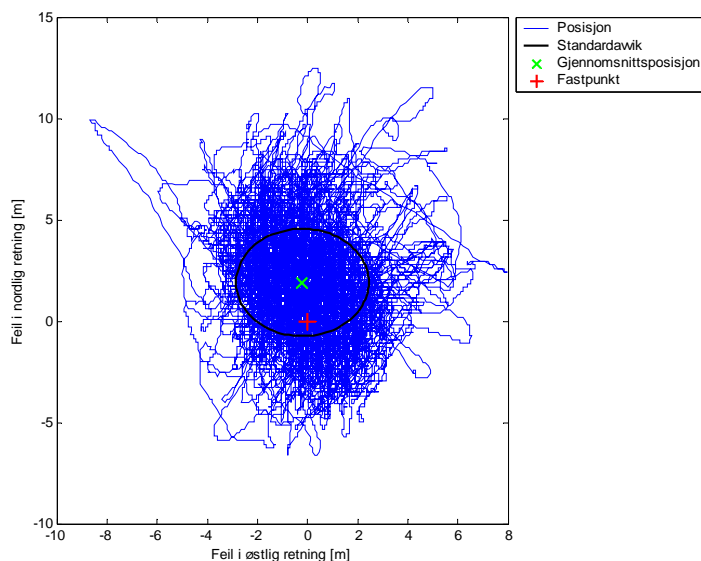
Det første vi gjorde var å ta kald- og varmstartsmålinger under perfekte omgivelser, dvs. på taket av HiA i fri sikt. Dette var for å finne ut hvor bra oppstartstiden til GPS mottakeren kan være. μ -Center har den funksjonen at brukeren kan velge å restarte mottakeren både med kald og varmstart. Vi restartet GPS mottakeren hvert tredje minutt. Dataene ble logget vha. μ -Center og i etterkant analysert i Matlab 6.5. Grunnen til at vi valgte å måle i treminutters intervaller var å finne ut om nøyaktigheten ble bedre jo lengre tid etter TTFF. Som referanse brukte vi det fast oppmålt punktet på taket av HiA.

Siden GPS mottakeren skal sitte på en sau ønsket vi også å finne ut om bevegelse hadde noe å si på TTFF. Vi restartet mottakeren både med kald og varmstart mens vi beveget oss rundt med mottakeren. Siden vi gikk rundt i Grimstad mens vi foretok målingene hadde vi ikke noe fast punkt å referere målingene til. Å finne ut om nøyaktigheten til den beregnede posisjonen ble bedre og bedre etter TTFF var derfor vanskelig. Vi valgte å restarte GPS mottakeren noen få sekunder etter at den hadde fått første gyldige måling. Ved varmstart sendte vi en tilnærmet posisjon, et tilnærmet tidspunkt og GPS Time Of Week til mottakeren. Ved hjelp av disse parameterne vil mottakeren i teorien ha større forutsetninger for raskere å beregne en nøyaktig posisjon enn ved kaldstart. Posisjonene ble analysert i Matlab 6.5 og sammenliknet med oppstartsmålingene fra taket. Programkoden for analyse av oppstartsmålingene er gitt i vedlegg C.

5 Resultat

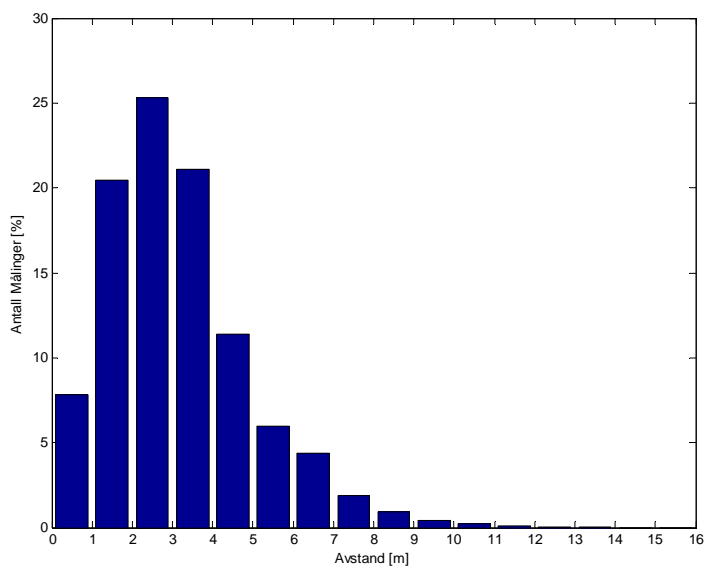
5.1 Nøyaktighetsmålinger

For å sjekke nøyaktigheten til GPS-systemet målte vi posisjonen over et døgn. Spredningsdiagrammet i figur 5-1 viser hvordan den beregnede posisjonen til et fast punkt ”drifter” i løpet av et døgn.



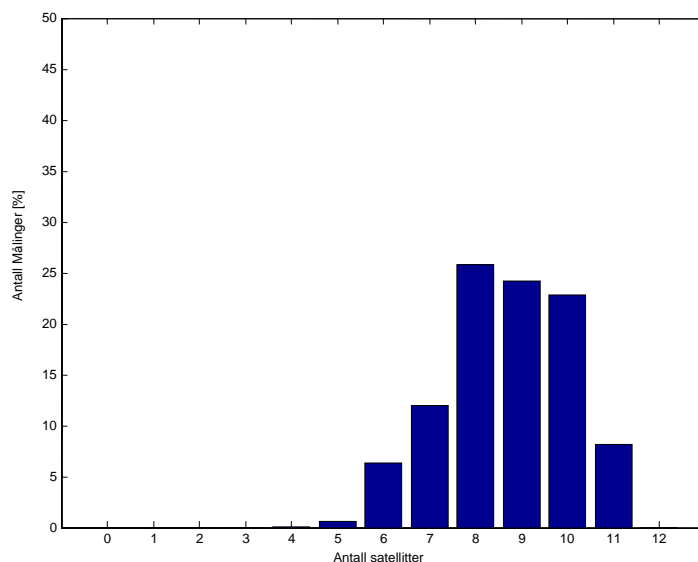
Figur 5-1: Spredningsdiagram fri sikt

For å gi en bedre visualisering over avviket mellom den beregnede posisjonen og det faste punktet, sorterte vi avstandene til de beregnede posisjonene etter størrelse. Figur 5-2 viser hvordan avstanden mellom posisjonene og fastpunktet fordelte seg over et døgn.



Figur 5-2: Avstand fra fastpunkt

Ut fra hvordan satellittkonstellasjonen ser ut velger GPS mottakeren fire satellitter den skal benytte for å beregne en posisjon. Når GPS mottakeren har kontakt med flere satellitter har den flere kombinasjoner av 4 satellitter å velge mellom. Mottakeren velger da den konstellasjonen som gir lavest HDOP. Figur 5-3 viser hvor mange satellitter vår mottaker hadde kontakt med i løpet av et døgn.

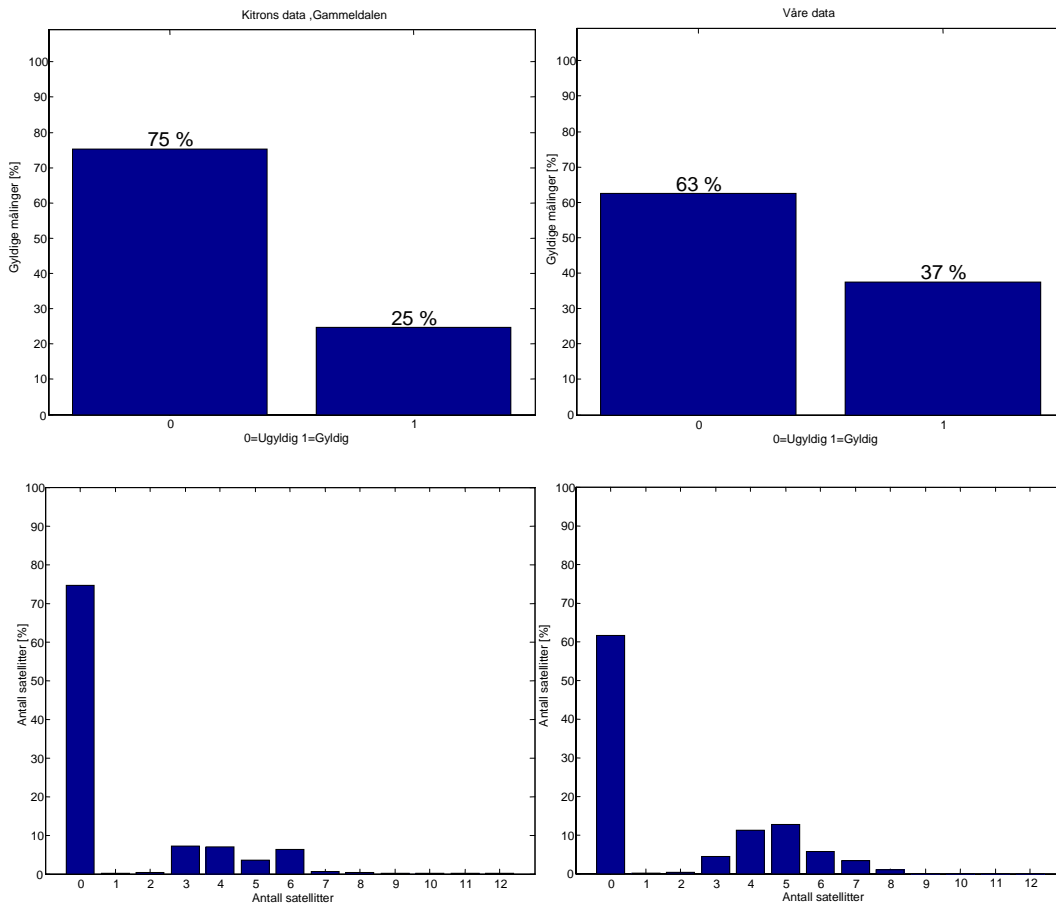


Figur 5-3: Antall satellitter mottakeren har kontakt med

5.2 Simuleringer

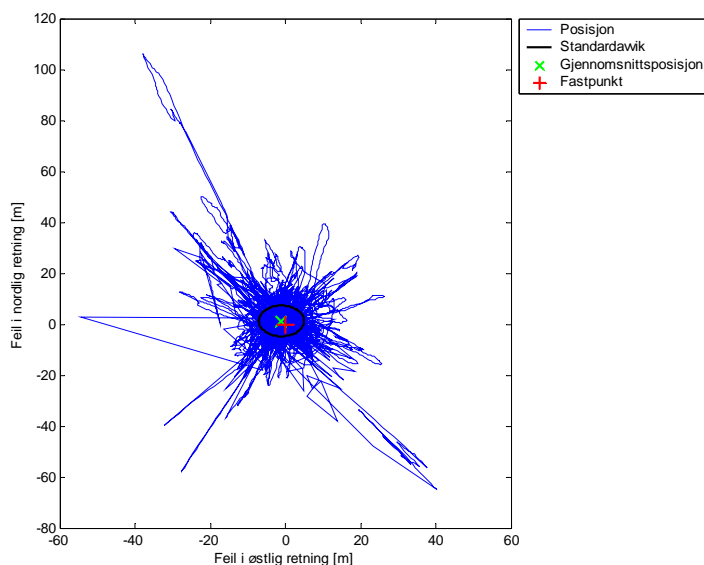
Dataene Kitron samlet inn sommeren 2003 hadde ca. 70 % ugyldige målinger. Siden vi ikke hadde mulighet til å gjenskape forsøkene Kitron gjorde, måtte vi prøve på annen måte å få tilnærmet like resultater. Vi prøvde å legge forskjellige stålrør over mottakeren for å forverre forholdene GPS mottakeren opplever. Etter mange forsøk fikk vi til slutt en måling som var sammenlignbar med Kitrons målinger.

Figur 5-4 viser Kitrons data fra Gammeldalen sammenlignet med våre data. Figuren viser at vi fikk ca. samme prosentvis ugyldighet som Kitron, samtidig som fordelingen over antall satellitter mottakeren hadde kontakt med var tilnærmet lik.



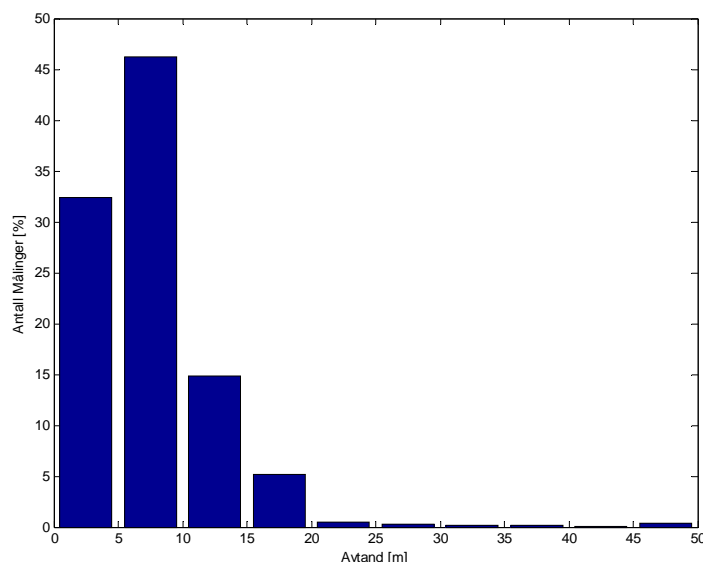
Figur 5-4: Sammenlikning av våre og Kitrons data

Figur 5-5 viser hvor mye den beregnede posisjonen drifter i forhold til det faste punktet vi refererte målingene til. Figuren viser kun de gyldige målingene.



Figur 5-5: Spredningsdiagram med rør over mottaker

For lettere å kunne sammenlikne måleresultatene med frisikt målingene, sorterte vi målingene etter avstanden til fastpunktet. Figur 5-6 viser avviket i avstanden mellom den beregnede posisjonen og fastpunktet når vi plasserte et rør over mottakeren. I søylen helt til høyre er alle målinger over 50 meter unna fastpunktet inkludert.



Figur 5-6: Avstand fra fastpunkt med rør over mottaker

5.3 Oppstartsmålinger

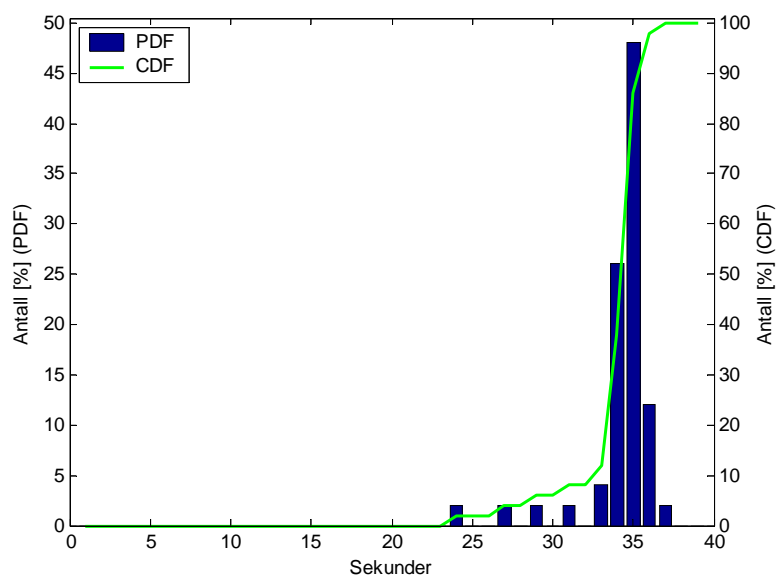
5.3.1 Oppstartstider

Hver figur vi presenterer her inneholder to grafer. Grafene vi presenterer er et resultat av måleserier på femti målinger.

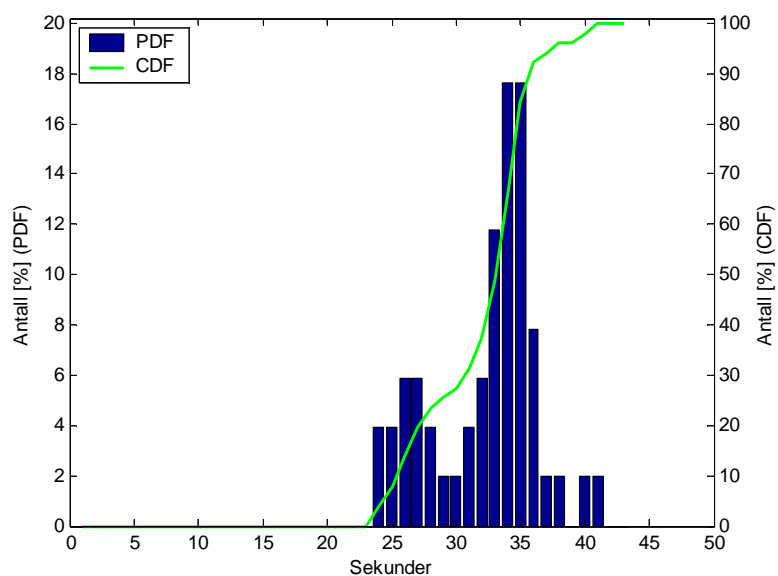
PDF-en (Probability Density Function) viser et histogram over hvor raskt GPS mottakeren bruker på å gi første posisjonsmåling. Når man ser på PDF-en må man se på x-aksen og venstre y-akse. X-aksen viser hvor mange sekunder GPS mottakeren har brukt på å gi første gyldige posisjonsmåling og venstre y-aksen angir hvor mange målinger i % som har benyttet dette antall sekunder.

Når man ser på CDF-en (Cumulative Density Function) må man se på x-aksen og høyre y-akse. Høyre y-aksen viser hvor mange målinger i % som har benyttet antall sekunder angitt på x-aksen eller mindre på å finne første gyldige posisjonsmåling.

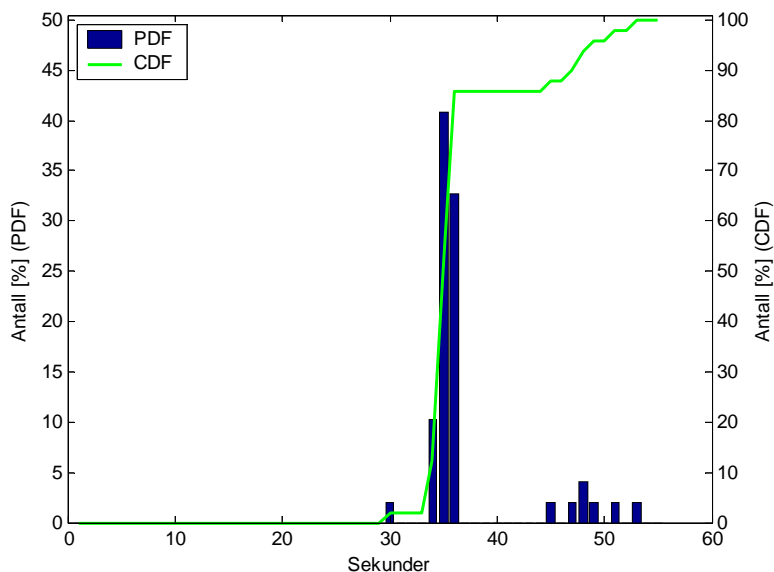
Figur 5-7 og figur 5-8 viser målinger gjort med GPS mottakeren i fri sikt hvor vi benytter varmstart. I figur 5-8 har vi gitt GPS mottakeres en posisjon som er 2 km feil. Figur 5-9 viser målinger gjort i fri sikt og ved bruk av kaldstart.



Figur 5-7: Varmstart i fri sikt

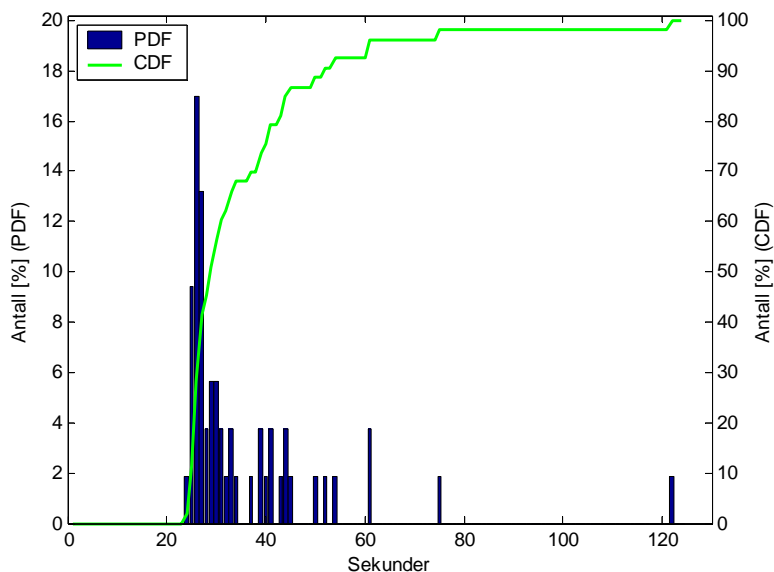


Figur 5-8: Varmstart i fri sikt med 2 km feil i posisjon sent til GPS mottaker

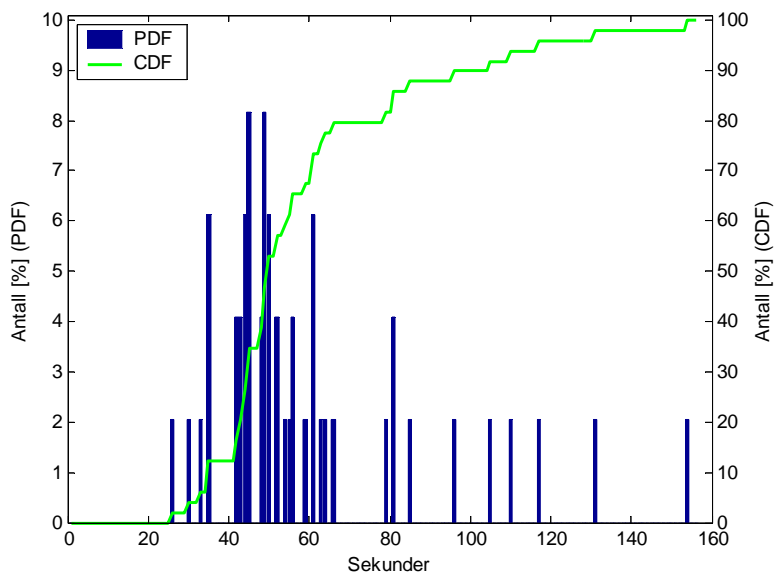


Figur 5-9: Kaldstart i fri sikt

Figur 5-10 og figur 5-11 viser målinger gjort med noe dårligere forhold enn i figur 5-7 og figur 5-9. Vi har her blokkert signalene fra noen satellitter ved å sette et rør over GPS mottakeren slik at den har mellom fire og sju satellitter tilgjengelig.

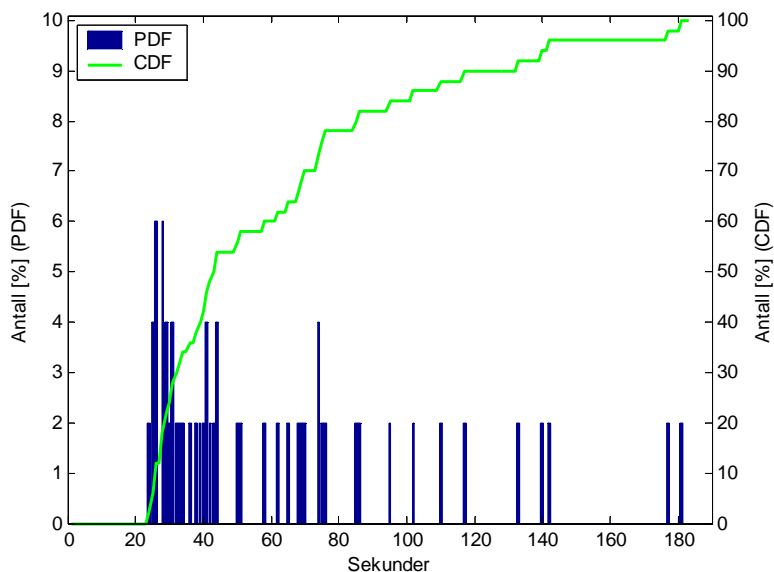


Figur 5-10: Varmstart med dårlige forhold

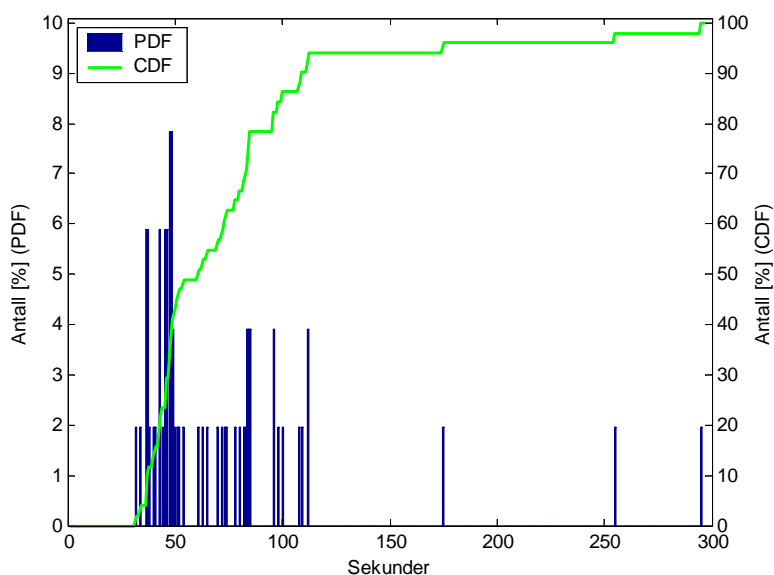


Figur 5-11: Kaldstart med dårlige forhold

På figur 5-12 og figur 5-13 har vi hatt GPS mottakeren i bevegelse mens vi utfører kald og varmstart. GPS mottakeren hadde fri sikt og bevogde seg i rolig gangfart. Omgivelsene var relativt åpne, men med enkelte hus, trær og små hauger innimellom.



Figur 5-12: Varmstart mens GPS mottakeren er i bevegelse



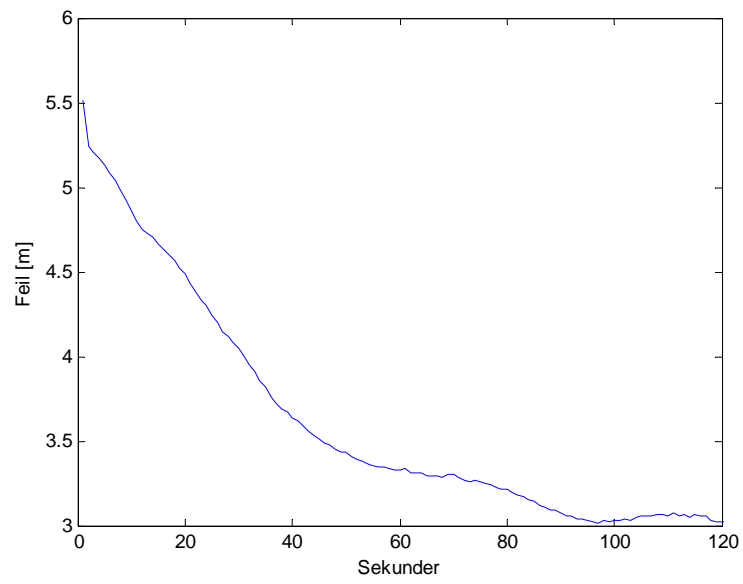
Figur 5-13: Kaldstart mens GPS mottakeren er i bevegelse

5.3.2 Nøyaktighet etter oppstart

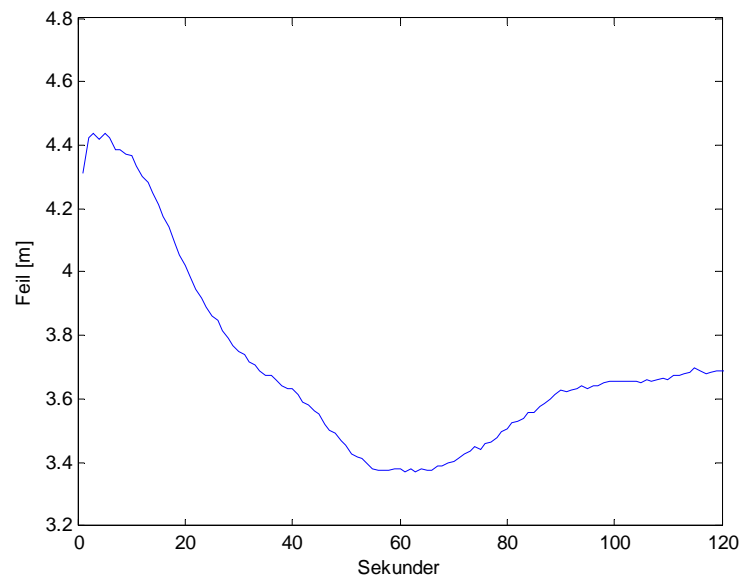
Etter at vi fikk første posisjonsmåling fortsatte loggingen noen minutter slik at vi kunne se hvordan nøyaktigheten var etter at vi fant første gyldige måling. Feilen er beregnet ut fra avstanden til det nøyaktig oppmålte punktet vi hadde.

Grafene er et resultat av femti forsøk og representerer et gjennomsnitt av alle disse målingene. X-aksen viser hvor mange sekund som er gått siden første gyldige posisjonsmåling, og y-aksen viser hvor stor mange meter punktet vi får fra GPS mottakeren er fra det faste punktet.

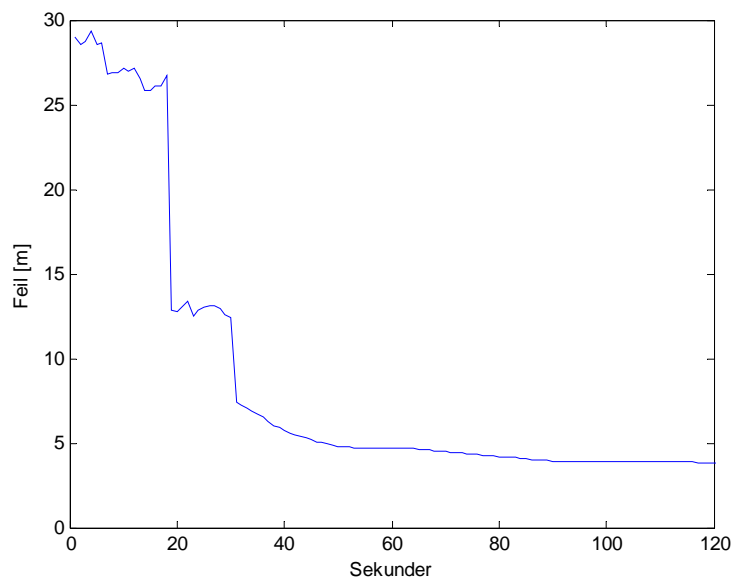
Figur 5-14 og figur 5-15 viser målinger gjort med GPS mottakeren i fri sikt hvor vi benytter varmstart. I figur 5-15 har vi gitt GPS mottakeres en posisjon som er 2 km feil. Figur 5-16 viser målinger gjort i fri sikt og ved bruk av kaldstart.



Figur 5-14: Varmstart i fri sikt

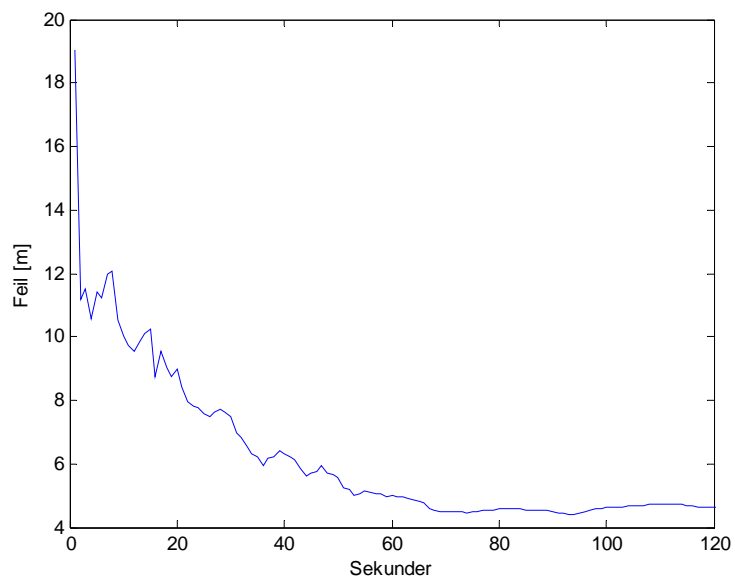


Figur 5-15: Varmstart i fri sikt med 2 km feil i posisjon sent til GPS mottaker

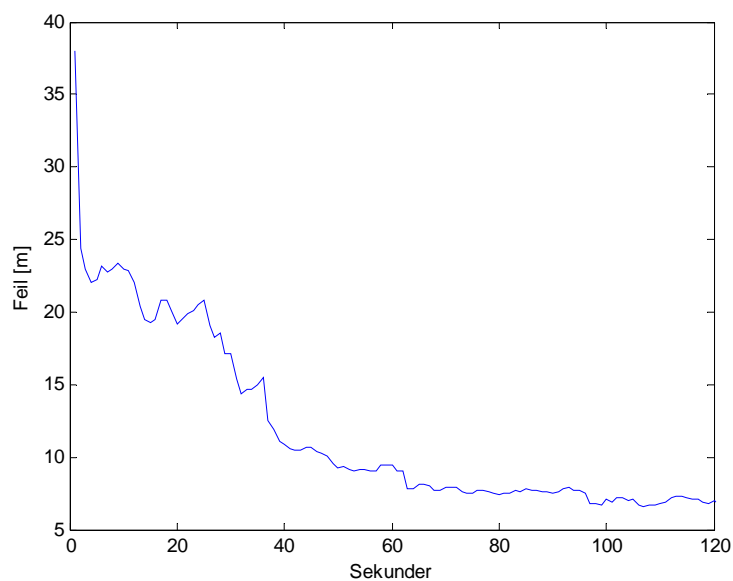


Figur 5-16: Kaldstart i fri sikt

Figur 5-17 og figur 5-18 viser målinger gjort med noe dårligere forhold enn i figur 5-14 og figur 5-16. Vi har her blokkert signalene fra noen satellitter ved å sette et rør over GPS mottakeren slik at den har mellom fire og sju satellitter tilgjengelig.



Figur 5-17: Varmstart med dårlige forhold



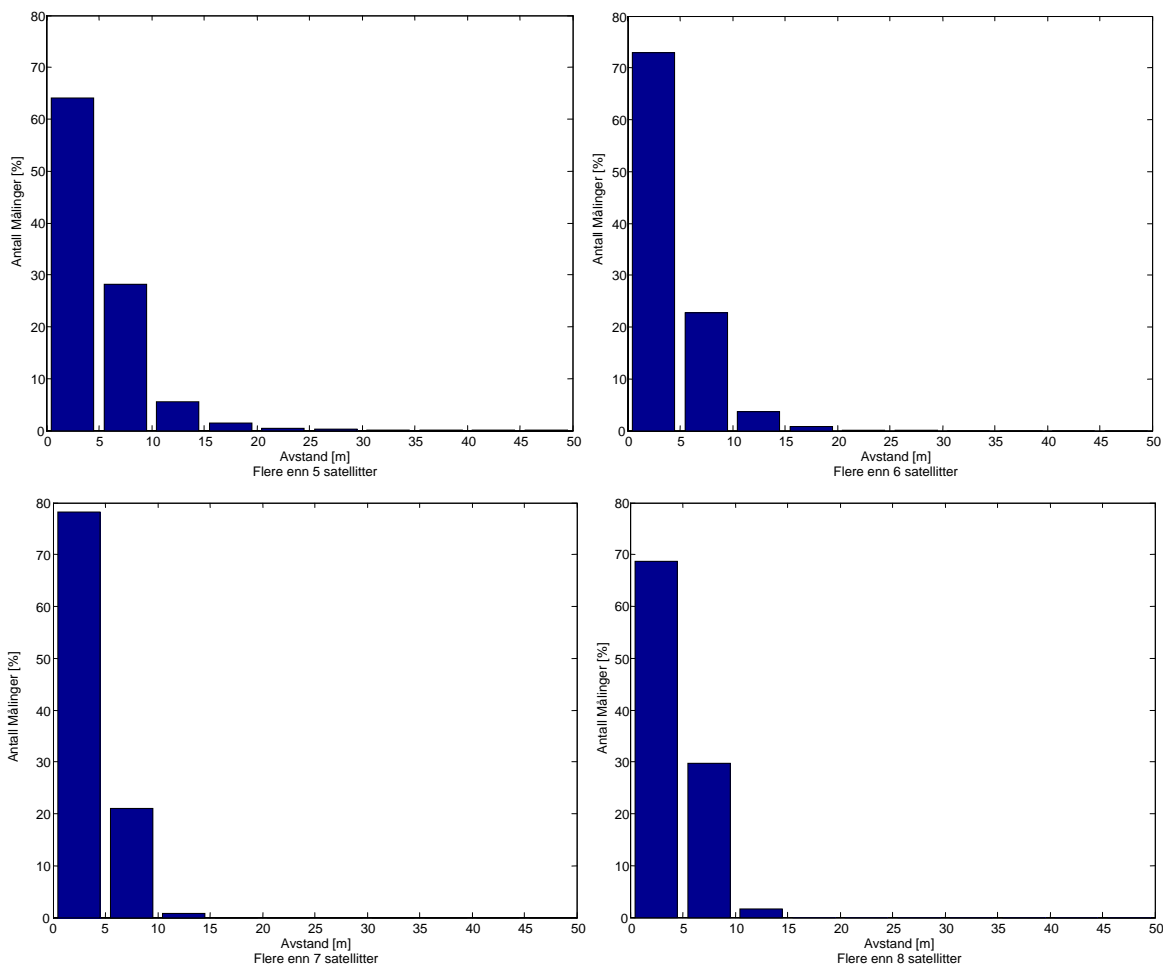
Figur 5-18: Kaldstart med dårlige forhold

5.4 Behandling av posisjonsdata

5.4.1 Filtrering

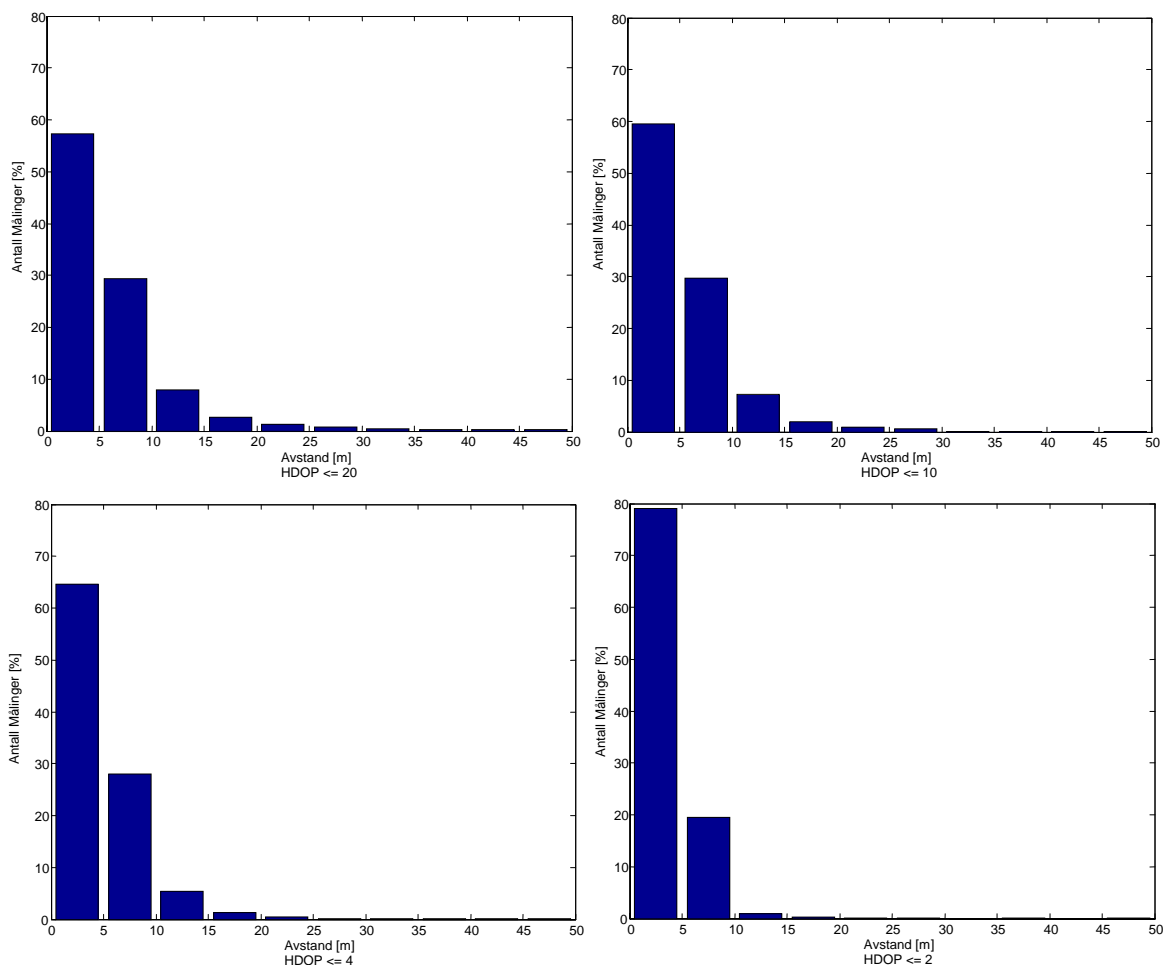
For å øke nøyaktigheten i posisjonsmålingen ønsket vi å sette diverse kriterier for hva som defineres som en gyldig måling. Dette kapitlet viser resultatene etter at vi har satt disse kriteriene.

Figur 5-19 viser avviket mellom den beregnede posisjonen og posisjonen til fastpunktet. Vi har her valgt ut målinger etter hvor mange satellitter mottakeren hadde kontakt med i tidspunktet når beregningene blir tatt. For tilhørende spredningsdiagram, se vedlegg A.



Figur 5-19: Mer eller lik 5, 6, 7 og 8 satellitter

En annen måte å velge ut posisjonsdata er å se på spredningsfaktoren, HDOP, for satellittene. Figur 5-20 viser avviket mellom de beregnede posisjonene og fastpunktet. Figuren viser avstandene når HDOP er lavere enn 20, 10, 4 og 2. For tilhørende spredningsdiagram, se vedlegg A.



Figur 5-20: HDOP mindre eller lik 20, 10, 4, 2

Tabellen under viser hvor mange målinger som gjenstår når vi setter de forskjellige kriterier på hva som skal defineres som en gyldig måling.

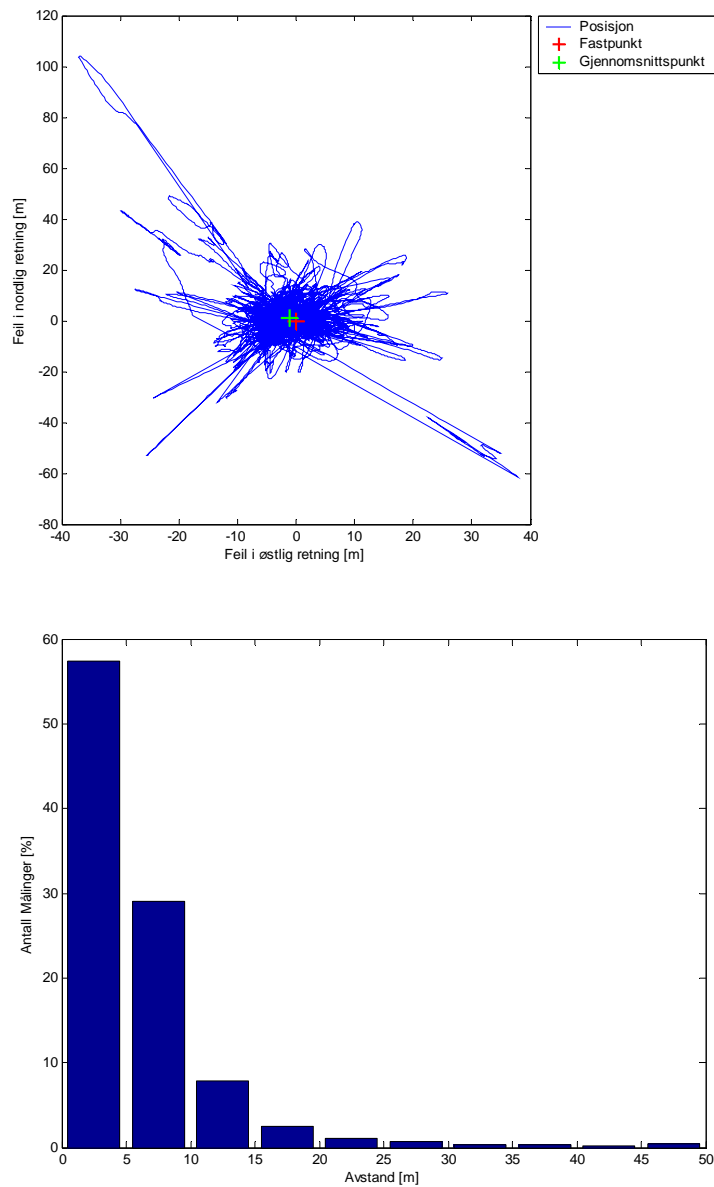
Tabell 5-1: Antall gjenværende målinger etter filtrering

Kriterium	Antall gjenværende målinger
Bare gyldige målinger	37 %
Mer eller lik 5 satellitter	23 %
Mer eller lik 6 satellitter	10 %
Mer eller lik 7 satellitter	4 %
Mer eller lik 8 satellitter	1 %
HDOP mindre eller lik 20	37 %
HDOP mindre eller lik 10	35 %
HDOP mindre eller lik 4	28 %
HDOP mindre eller lik 2	13 %

5.4.2 Midling

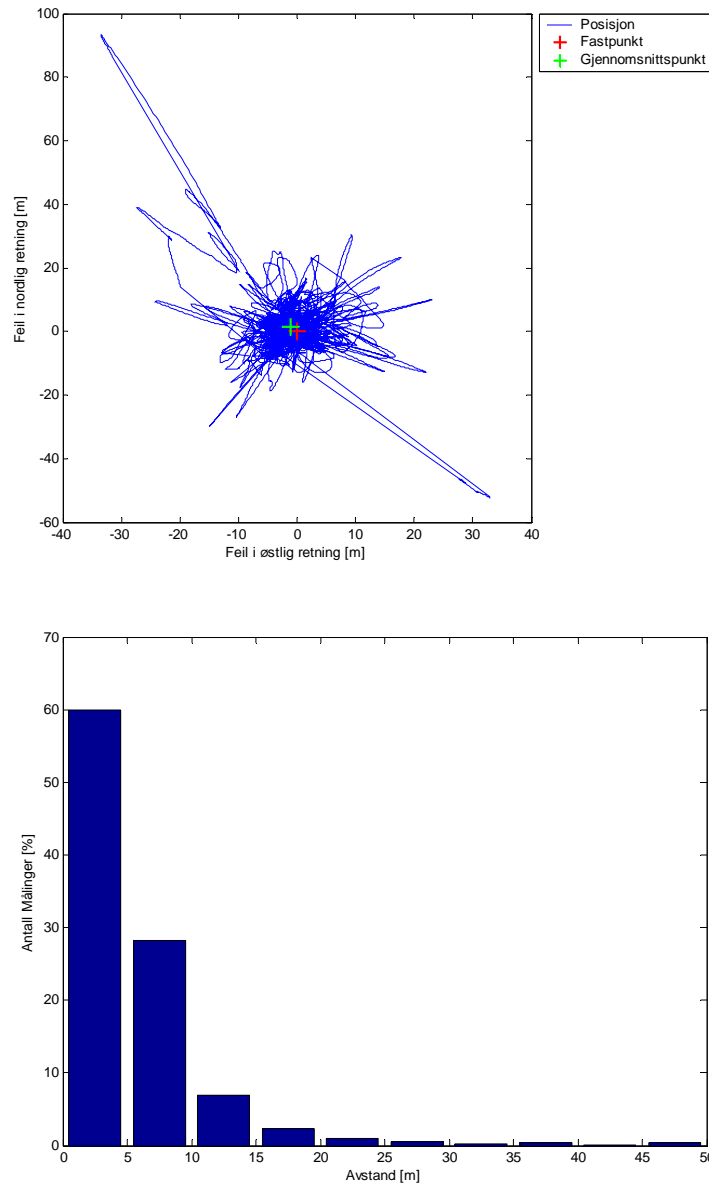
Vi prøvde også å komme fram til noen algoritmer som kan brukes på dataene for å gjøre nøyaktigheten bedre.

Figur 5-21 viser resultatene etter at vi har benyttet midlingsalgoritmen hvor kriteriet er at minimum 10 av 15 målinger er gyldige. Figuren viser både spredningsdiagram og avstandsavviket mellom posisjonene og fastpunktet. Etter midling gjenstår 37 % av alle målingene i måleserien.



Figur 5-21: Midling over 15 målinger hvor 10 må være gyldige

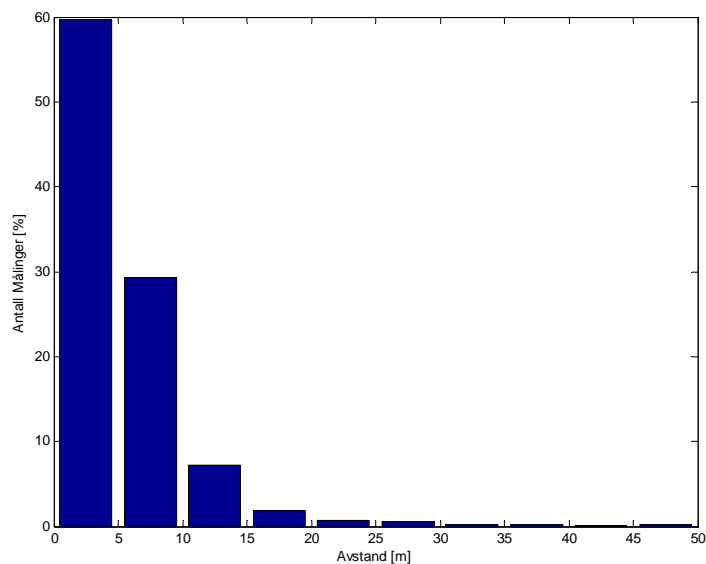
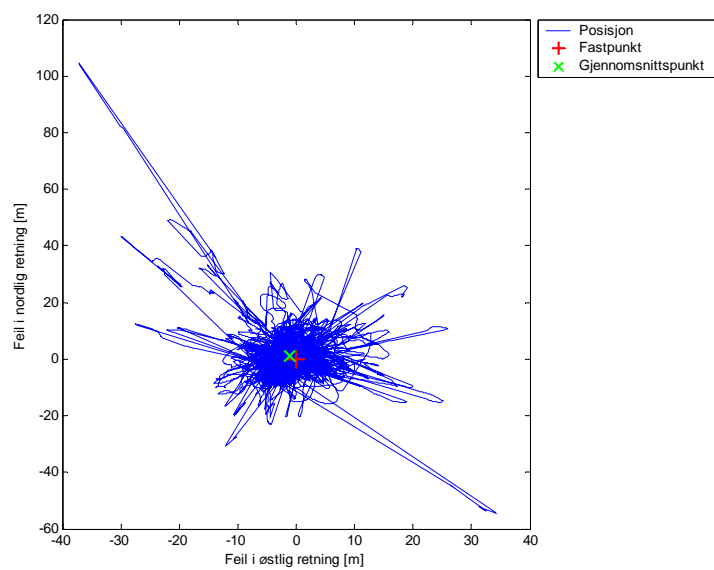
Figur 5-22 viser spredningsdiagrammet og avviket i avstanden mellom posisjonene og fastpunktet ved bruk av midlingsalgoritmen. Her er kriteriet for å midle er at minimum 50 av 60 målinger er gyldige. I figuren gjenstår 36 % av alle målingene i måleserien.



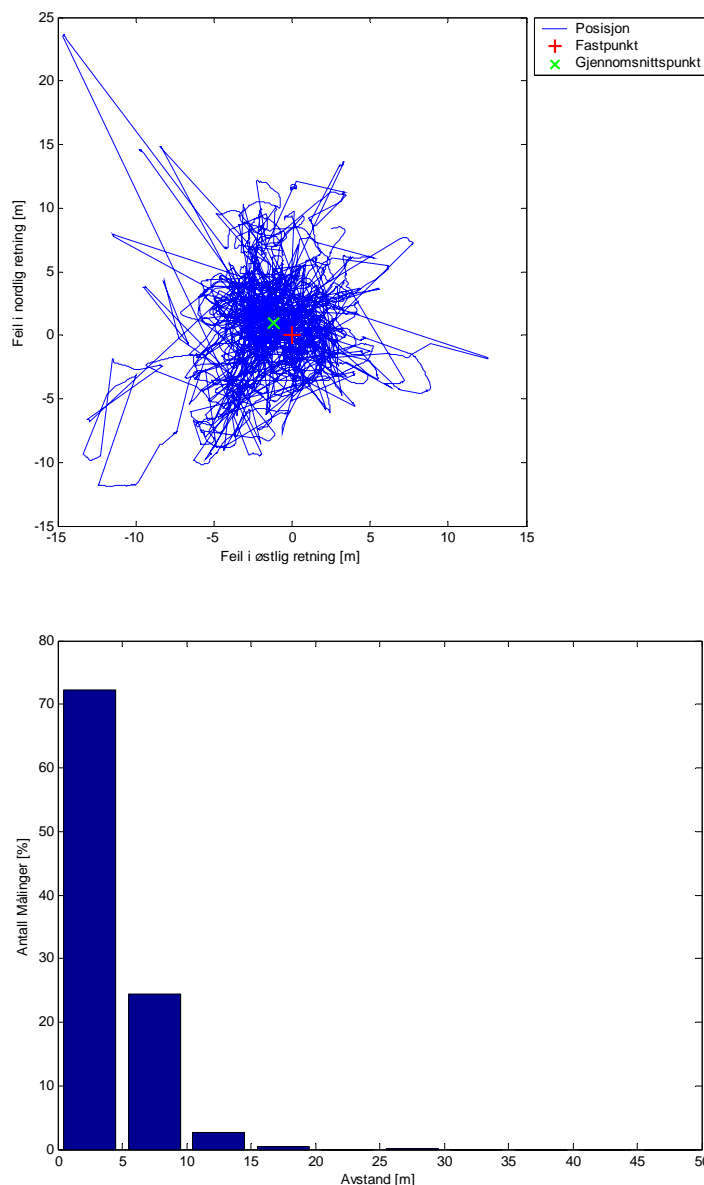
Figur 5-22: Midling over 60 målinger hvor 50 må være gyldige

5.4.3 Sirkelmidling

Vi prøvde også å tilnærme oss mer nøyaktig posisjon vha. sirkelmidlings algoritmen. Figur 5-23 og figur 5-24 viser spredningsdiagrammet og avstandene mellom posisjonene og fastpunktet når det henholdsvis midles over 15 og 60 målinger hvor 10 og 50 målinger må være gyldige. I figur 5-23 gjenstår 35 % av målingen og i figur 5-24 gjenstår 27 % av målingene.



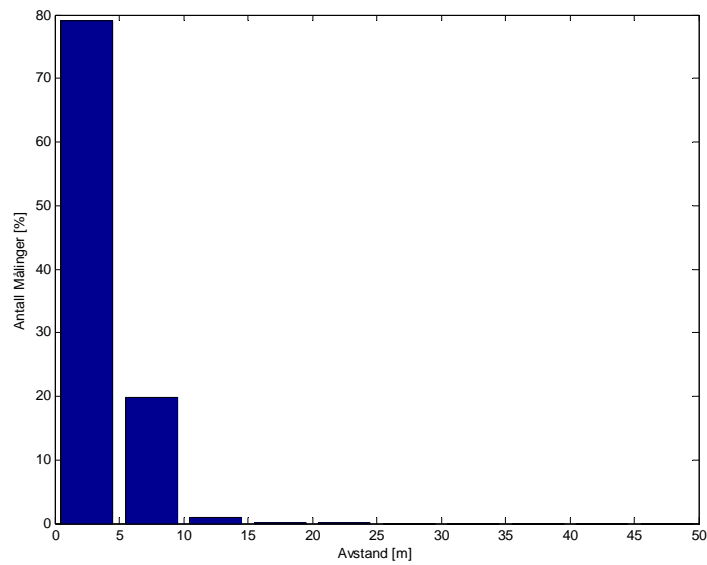
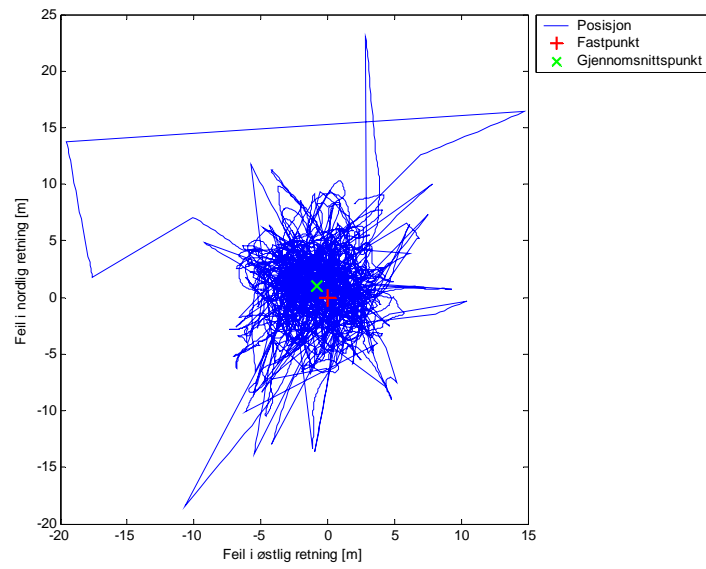
Figur 5-23: Midling over 15 målinger hvor 10 må være gyldige
 $R=2$, $X=4$



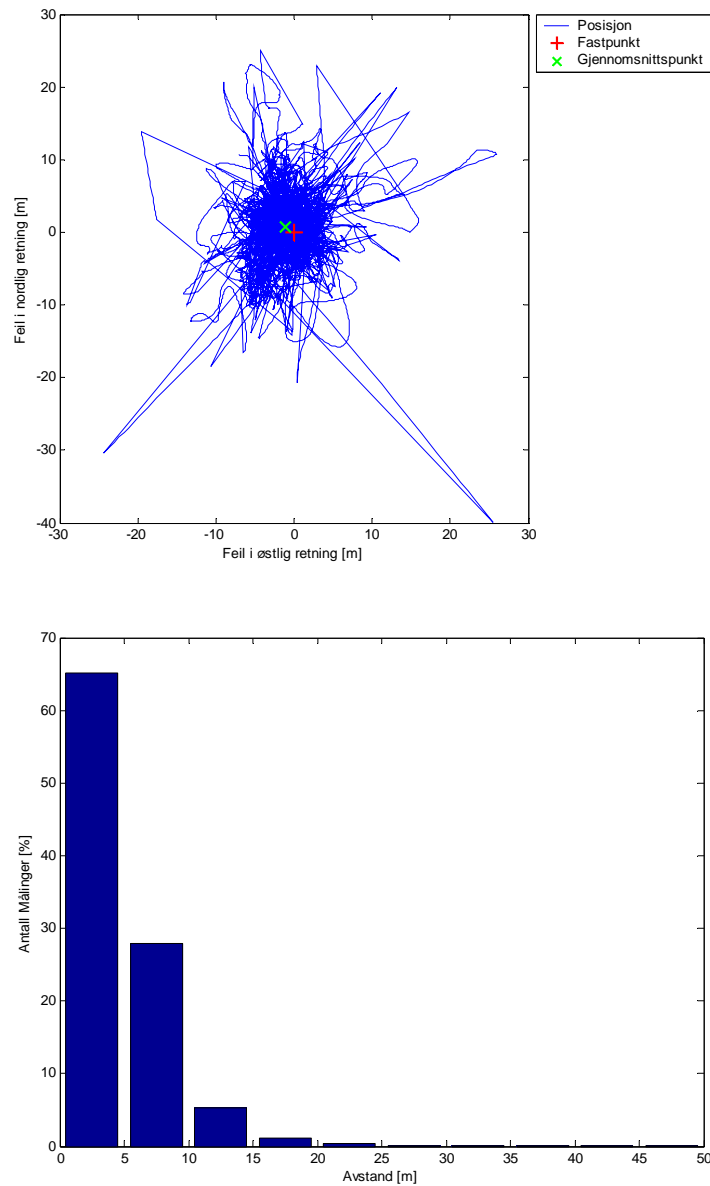
**Figur 5-24: Midling over 60 målinger hvor 50 må være gyldige
R=2, X=4**

5.4.4 Midling med HDOP

Disse resultatene viser nøyaktigheten etter at vi har midlet posisjoner med et visst kriterium til størrelsen på HDOP. Figur 5-25 og figur 5-26 viser spredningsdiagrammet samt avviket mellom posisjonene og fastpunktet. Det midles over målinger hvor minimum 10 av 15 målinger må ha lavere HDOP enn henholdsvis 2 og 4. Det midles bare over målingene som tilfredsstillere kravene. I figur 5-25 gjenstår 15 % og i figur 5-26 gjenstår 28 % av målingene i måleserien.

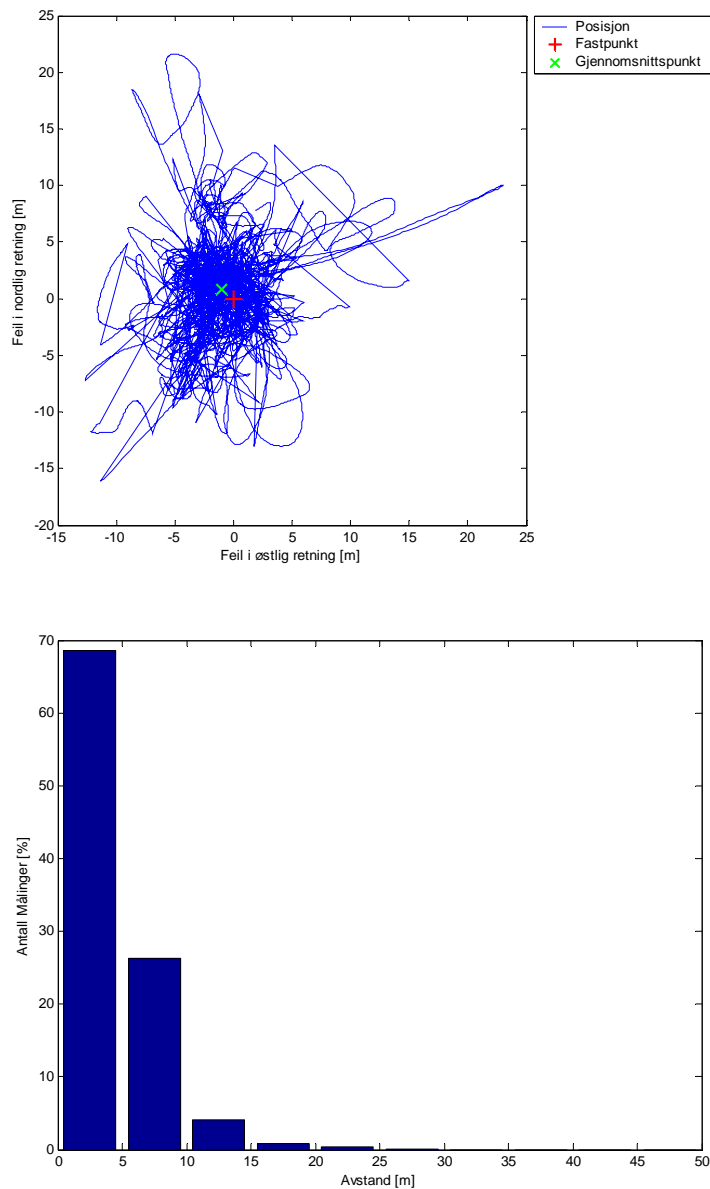


Figur 5-25: Midles over 15, 10 målinger har HDOP lavere enn 2



Figur 5-26: Midles over 15, 10 målinger må ha HDOP lavere enn 4

Figur 5-27 viser spredningsdiagrammet og avviket mellom posisjon og fastpunkt når det midles over målinger hvor 50 av 60 målinger må ha lavere HDOP enn 4. I figuren gjenstår 26 % av målingene fra måleserien.



Figur 5-27: Midling over 60 målinger, 50 må ha HDOP lavere enn 4

6 Drøfting

6.1 Nøyaktighetsmålinger

Kitron ønsket at nøyaktigheten i dyreenheten skulle være rundt 10 meter. Forsøkene vi har gjort viser at nøyaktigheten til GPS mottakeren ligger rundt 10 meter. Figur 5-2 viser at ca 98 % av alle posisjonsmålingene under tilnærmet perfekte forhold ligger innenfor 10 meter. Ut fra spredningsdiagrammet i figur 5-1 ser vi at flesteparten av målingene ligger samlet mens noen målinger ligger forholdsvis langt unna fastpunktet

Figur 5-1 viser også gjennomsnittsposisjon av alle målingene samt det faste punktet vi refererte målingene til. Disse skulle ideelt sett vært i samme punktet, noe figuren viser at de ikke er. Dette skyldes sannsynligvis unøyaktigheter som oppstår når det konverteres fra NGO og WGS84 til UTM koordinater, samt unøyaktigheter i GPS mottakeren.

Mottakeren vi brukte under prosjektet har 12 kanaler og kan derfor ikke ha kontakt med flere en 12 satellitter på en gang. Når GPS mottakeren har kontakt med flere satellitter vil det være enklere å finne en kombinasjon på 4 satellitter som gir en lav HDOP. Lav HDOP er en bra antydning på at posisjonen som beregnes er nøyaktig. Figur 5-3 viser at mottakeren har hatt kontakt med mellom 6 og 11 satellitter gjennom døgnet. Slike målinger vil gi en lav HDOP noe som gir en god indikasjon på at posisjonen er nøyaktig. Ut fra dette kan vi konkludere med at målingene vi har tatt er bra.

6.2 Simuleringer

Kitrons data fra feltperioden sommeren 2003 hadde ca 70 % ugyldige målinger. Figur 5-4 viser at 63 % av målingene våre ble ugyldige da vi satte et stålrør over mottakeren. Fordelingen over hvor mange satellitter mottakeren har hatt kontakt med er også tilnærmet lik. Vi kan derfor konkludere med at vi har klart å simulere Kitrons data på en tilfredsstillende måte. Vi valgte denne måleserien på 63 % fordi feilmålingene spredte seg over hele måleperioden. Andre målinger gav oss større prosentvis ugyldighet, men kunne f.eks. ha flere timer etter hverandre med bare ugyldige målinger. Vi anså det som mer realistisk at feilmålingen spredte seg utover i motsetning til flere timer etter hverandre med ugyldige målinger.

Sammenligner vi figur 5-3 med våre data i figur 5-4 viser det tydelig at forholdene innvirker på antall satellitter mottakeren har kontakt med. Gjennomsnittet ligger på 5 satellitter når vi forverrer forholdene, mot 9 under tilnærmet perfekte forhold. Jo færre satellitter en mottaker har kontakt med jo vanskeligere er det å finne en kombinasjon på 4 satellitter som gir en lav HDOP. Lav HDOP gir en god antydning på at posisjonen som er beregnet er nøyaktig. Dette er igjen en indikasjon på at forholdene en sau lever under kan føre til at nøyaktigheten til systemet blir dårligere.

Figur 5-5 og figur 5-6 viser hvordan den beregnede posisjonen drifter gjennom måleperioden. Ut fra figurene ser vi at unøyaktigheten i den beregnede posisjonen øker når noen av signalene blir blokkert av stålrøret. Hvis vi sammenlikner med figur 5-1 og figur 5-2 ser vi tydelig at nøyaktigheten til GPS systemet blir dårligere når forholdene forverres.

Figur 5-6 viser at ca 78 % av de gyldige målingene ligger innenfor en sirkel med radius på 10 meter.

Sauer lever ofte i skog og ulendt terreng som kan innvirke på forholdene GPS mottakeren opplever. Med tanke på at forholdene reduserer nøyaktigheten kan vi ut fra figurene i kapittel 5.1 og 5.2 si at det kanskje vil bli vanskelig å oppnå en nøyaktighet på 10 meter for dyreenheten når nøyaktigheten i tilnærmet perfekte forhold ligger rundt 10 meter. I beste fall vil sauer som lever i omgivelser som ikke forstyrrer GPS mottakeren oppleve en nøyaktighet på rundt 10 meter.

6.3 Oppstartsmålinger

6.3.1 Fri sikt målinger

I seks av de åtte gangene GPS mottakeren i dyreenheten startes i løpet av et døgn har den bare 50 sekunder til å finne en gyldig posisjon. Figur 5-7 og figur 5-9 viser at dette stort sett er nok både ved kald- og varmstart i fri sikt. Ved varmstart er alle målingene godt innefor 50 sekunder og som vi ser av figur 5-14 er målingene også godt innefor en nøyaktighet på 10 meter allerede fra første sekund. Dette viser at 50 sekunder er nok når varmstart benyttes i fri sikt.

Når det gjelder kaldstart er det i 4 % av forsøkene ikke nok med 50 sekunder for å finne gyldig posisjonsmåling. Figur 5-16 viser at GPS mottakeren har en gjennomsnittlig unøyaktighet på omtrent 30 meter ved første gyldige posisjonsmåling. Etter dette trenger den gjennomsnittlig 30 sekunder før nøyaktigheten er bedre enn ti meter.

Forskjellene på hvor lang tid GPS mottakeren trenger for å finne gyldig posisjonsmåling er relativt små mellom varm- og kaldstart i fri sikt. Vi ser at hovedtyngden av målinger ligger rundt 35 sekunder. Forskjellen ligger i at oppstarten ved kaldstart har en del målinger som benytter mellom ti og tjue sekunder mer. Likevel er den største forskjellen mellom varm- og kaldstart i fri sikt hvor lang tid GPS mottakeren benytter på å komme innenfor en nøyaktighet på ti meter etter første gyldige posisjonsmåling.

I figur 5-8 og figur 5-15 har vi startet GPS mottakeren med varmstart med en posisjon som er omtrent 2 km feil i forhold til der GPS mottakeren virkelig er. Som vi ser av figurene er ikke resultatene blitt noe dårligere enn når vi gav en relativt nøyaktig posisjon, men faktisk blitt marginalt bedre. Grunnen til dette tror vi er tilfeldigheter, og vi ser at hovedtyngden av målinger ligger på samme sted. Dette tyder på at det ikke har noen betydning om posisjonen vi gir GPS mottakeren er noen km feil.

6.3.2 Målinger med dårligere forhold

Figur 5-10 viser oppstart med varmstart i dårligere forhold. Grafen viser at 88 % av forsøkene gir gyldig posisjonsmåling i løpet av 50 sekunder og som vi ser av figur 5-17 bruker GPS mottakeren i gjennomsnitt 15 sekunder på å finne en måling med nøyaktighet bedre enn ti meter.

Ved kaldstart i dårlige forhold som vist i figur 5-11 har 54 % av forsøkene gitt gyldig posisjonsmåling innen 50 sekunder. For å få en nøyaktighet bedre enn 10 meter kan vi se

av figur 5-18 at det går i gjennomsnitt nesten 50 sekunder før GPS mottakeren oppnår dette.

Vi ser at forskjellene mellom varm- og kaldstart er blitt større når GPS mottakeren har dårligere forhold. Ved varmstart har 90 % av forsøkene gitt gyldig posisjonsmåling etter ca 50 sekunder, mens kaldstart trenger over 100 sekunder for å oppnå 90 % gyldige posisjonsmålinger. I tillegg er det en betydelig forskjell på hvor lang tid GPS mottakeren trenger på å finne en måling innenfor 10 meters nøyaktighet.

6.3.3 Målinger med bevegelse

Figur 5-12 viser at GPS mottakeren i bevegelse og ved bruk av varmstart kun gir ut gyldig posisjonsmåling i 56 % av forsøkene i løpet av 50 sekunder. Det er her skjedd en stor endring fra varmstart forsøkene i fri sikt. Forsøkene viser også at GPS mottakeren kan få problemer med å finne gyldig posisjonsmåling når den har 180 sekunder for å finne posisjon.

Ved kaldstart og GPS mottakeren i bevegelse som vist i figur 5-13 er det 44 % av forsøkene som gir gyldig posisjonsmåling i løpet av 50 sekunder. Her viser forsøkene at også 250 sekunder ikke alltid er nok til å finne en gyldig posisjon.

Forsøkene viser at forskjellene mellom varm- og kaldstart ikke er veldig stor når GPS mottakeren er i bevegelse. CDF-en viser at 80 % av forsøkene har funnet gyldig posisjonsmåling etter 100 sekunder både ved varm- og kaldstart. Den store forskjellen mellom disse grafene er at den høyeste verdien for kaldstart er betydelig høyere enn den høyeste verdien for varmstart.

6.3.4 Oppstartstider

Forsøkene viser at Kitron ikke har gitt GPS mottakeren nok tid til å finne en gyldig posisjon. Vi har ikke hatt noen mulighet til å undersøke varm- og kaldstart i bevegelse i kombinasjon med dårlige forhold da dette var vanskelige betingelser å kontrollere. Likevel vil det være nærliggende å tro at disse målingene ville gitt enda dårligere resultater enn målingene med bare bevegelse. Hvor lang tid det går før GPS mottakeren finner en gyldig posisjonsmåling innefor ti meter vil trolig også øke.

For å anslå oppstartstider GPS mottakeren trenger for å finne en gyldig posisjonsmåling innfor 10 meter tar vi utgangspunkt i resultatene hvor GPS mottakeren er i bevegelse. Vi vil deretter legge til et anslag for hvor mange sekunder dårlige forhold påvirker resultatene og et tillegg for hvor lenge det går før målingene er innefor 10 meter.

Kitron ønsket at systemet skulle være operativt 90 % av gjeldende periode. Vi valgte derfor bare å ta med 90 % av målingene i måleserien. Noen av våre forsøk har gitt veldig høye ”topp” målinger. Ved å ikke ta med alle målingene unngår vi urimelig høye oppstartsmålinger.

Av figur 5-12 ser vi at 90 % av målingene ligger innenfor en oppstartstid på rundt 120 sekunder. Sammenlikner vi figur 5-7 og figur 5-10 ser vi at mottakeren bruker ca. 20 sekunder mer ved oppstart i dårlige forhold enn i friskt. Vi antar at denne økningen vil

være den samme hvis mottakeren har dårlige forhold samtidig som den er i bevegelse. Ved varmstart under dårlige forhold bruker mottakeren ca 15 sekunder etter første gyldige måling på å komme innenfor 10 meter. Grafen i figur 5-17 representerer bare et gjennomsnitt, noe som betyr at flere målinger vil trenge lengre tid enn figuren tilsier på å komme innenfor 10 meter. Om GPS mottakeren i tillegg var i bevegelse ville denne tiden øke. Vi gjør et anslag på at den trenger dobbelt så lang tid ved dårlige forhold kombinert med bevegelse. Dette tilsvarer 30 sekunder og totalt oppstartstid ved varmstart blir da:

Tabell 6-1: Utregning av oppstartstid ved varmstart

TTFE ved bevegelse (90 %)	120 sek
Tidstillegg for dårlige forhold	20 sek
Tidstillegg for å komme innenfor 10 meter	30 sek
Total oppstartstid ved varmstart	170 sek

Om vi benytter tilsvarende fremgangsmåte ved kaldstart vil den totale oppstartstiden ved kaldstart bli:

Tabell 6-2: Utregning av oppstartstid ved kaldstart

TTFE ved bevegelse (90 %)	120 sek
Tidstillegg for dårlige forhold	70 sek
Tidstillegg for å komme innenfor 10 meter	100 sek
Total oppstartstid ved kaldstart	290 sek

Slik dyreenheten er i dag gir Kitron GPS mottakeren like lang tid til å finne gyldig posisjonsmåling både ved varm- og kaldstart. For å hindre at GPS mottakeren står på lengre enn nødvendig er en mulighet å gi GPS mottakeren lengre tid ved kaldstart enn ved varmstart. Dette kan være hensiktsmessig siden forsøkene viser at det er en merkbar forskjell mellom varm- og kaldstart og siden batterikapasitet er et kritisk punkt for Kitron.

Om GPS mottakeren finner første posisjonsmåling før 170 sekunder ved varmstart eller 290 sekunder ved kaldstart, kan Kitron likevel velge å skru av GPS mottakeren før tiden er gått ut. Det viktige er at tiden det tar å finne en posisjon innenfor 10 meter blir opprettholdt. Dette betyr at etter TTFE bør dyreenheten vente henholdsvis 30 og 100 sekunder ved varm- og kaldstart før målingen blir tatt og GPS mottakeren blir skrudd av.

En mulig løsning for å unngå de lange oppstartstidene ved kaldstart er å bare starte GPS mottakeren med varmstart. Informasjonen den trenger for å ta varmstart er posisjon, TOW og uke nummer. TOW og uke nummer kan enkelt regnes ut om GPS mottakeren har mottatt denne informasjonen en gang tidligere. Hvis GPS mottakeren i forrige oppstart ikke fikk gyldig posisjonsmåling finnes det to måter å velge ut en posisjon på. Den ene løsningen er å sende med en forhåndsbestemt posisjon. Dette kan være posisjonen til basestasjonen eller posisjonen til et sted man på forhånd regner med at sauene vil oppholde seg i nærheten av. Det som vil være problematisk med denne løsningen er at posisjonen må kodes inn i hver dyreenhet og at denne kun vil være brukelig i det området den er programmert for. Den andre løsningen er å benytte den siste gyldige posisjonsmålingen uansett hvor lenge det er siden denne ble tatt. Denne vil sannsynligvis ikke være alt for langt unna der hvor sauen virkelig er, siden de vanligvis oppholder seg i et lite geografisk område. Siden forsøkene våre har vist at GPS mottakeren ikke benytter lengre tid til å finne

gyldig posisjon innefor 10 meter selv om den mottar en unøyaktig posisjon, er dette en god løsning.

6.4 Behandling av posisjonsdata

6.4.1 Filtrering

GPS mottakeren kan i enkelte tilfeller gi en gyldig posisjon selv om den har kontakt med 3 eller færre satellitter, men dette innebærer imidlertid ikke at posisjonen som er beregnet er nøyaktig. For å gi en posisjon i tre dimensjoner med lengdegrad, breddegrad og høyde må mottakeren ha kontakt med 4 satellitter. Posisjoner som er beregnet ut fra 3 eller færre satellitter kan bli veldig unøyaktige og bør ikke benyttes. Figur 5-19 viser hvordan nøyaktigheten vil bli dersom vi setter våre egne krav til hva som er en gyldig måling. I disse forsøkene har vi satt at en måling er gyldig hvis den er beregnet når mottakeren har kontakt med flere enn 5, 6, 7 og 8 satellitter. Vi ser at prosentandelen av målingene som ligger innenfor 10 meter øker fra ca 92 % med flere enn 5 satellitter til ca 99 % med flere enn 8 satellitter. Ulempen med å sette disse kravene er at mesteparten av målingene blir filtreres bort. I utgangspunktet var 37 % av målingene gyldige, dvs. at mottakeren hadde kontakt med en eller flere satellitter i 37 % av beregningene i måleperioden. Andelen målinger som gjenstår etter de forskjellige forsøkene er presentert i tabell 5-1. Som vi ser av tabellen gjenstår bare 10 % av målingene hvis vi godtar en nøyaktighet på 94 % innenfor 10 meter. Dette er veldig få målinger når vi ser av figur 5-4 at antall målinger som gir en fullstendig posisjonsbeskrivelse i tre dimensjoner (4 satellitter eller flere) er ca. 35 %. Det er ingen automatikk i at målinger hvor GPS mottakeren kun har hatt kontakt med 4 satellitter gir en unøyaktig posisjon. Mange nøyaktige målinger hvor GPS mottakeren har hatt kontakt med færre enn 6 satellitter vil kanskje filtreres bort hvis kriteriet for en gyldig måling er satt til kontakt med 6 satellitter eller flere. Figur 5-4 viser at flesteparten av de gyldige målingen i våre data ligger rundt 5 satellitter.

En annen måte å filtrere målingene på er å se på HDOP. For hver posisjonsberegning GPS mottakeren gjør beregner den HDOP verdien for de fire satellittene den benytter. Ved lav HDOP er det mer sannsynlig at posisjonen er nøyaktig. Vi satte i dette forsøket at en gyldig måling skulle ha en HDOP verdi lavere enn 20, 10, 4 og 2. Resultatene vi fikk vises i figur 5-20. Her ser vi at antall målinger som ligger innenfor 10 meter er fra 87 % for HDOP mindre enn 20, til 98 % for HDOP mindre enn 2. Også her filtreres vi bort mye av målingene, men ikke i like stor grad som ved filtrering etter hvor mange satellitter mottakeren hadde kontakt med. Ved bare å godta målinger med lavere HDOP enn 4 ser vi av tabell 5-1 at vi står igjen med 28 % av målingene. 92 % av disse målingene ligger innenfor 10 meters nøyaktighet. Dette resultatet er ganske likt resultatet vi fikk ved bare å se på målinger med flere enn 5 satellitter. Vi anser denne metoden som bedre fordi det kan hende at HDOP ligger innenfor kriteriene selv om mottakeren har kontakt med færre enn 5 satellitter. Ved bare å se på målinger med HDOP lavere enn 2 ligger 98 % av målingene innenfor 10 meter. Dette er et bra resultat, men her gjenstår bare 13 % av alle målingene, noe som er litt i minste laget.

6.4.2 Midling

Etter at vi midlet over 15 målinger fikk vi resultatene som vist i figur 5-21. Ved å benytte denne metoden ser vi at ca 86 % av alle målingene ligger innenfor 10 meter. I forhold til

Kitrons ønske om en nøyaktighet på 10 meter er dette resultatet litt for dårlig. Dette vises spesielt i spredningsdiagrammet, hvor vi ser at enkelte målinger ligger over 100 meter unna fastpunktet. Vi kan godta at noen få prosent av målingene ligger utenfor 10 meter, men når disse målingene ligger over 100 meter unna, er resultatet for dårlig. Sammenlikner vi spredningsdiagrammet i figur 5-21 med spredningsdiagrammet i figur 5-3 ser vi at disse er temmelig like. Noen av de målingene som ligger ca 30-40 meter unna fastpunktet har vi fått bort, men målingene som ligger lengst unna finnes fremdeles. For at resultatene skal bli bra må disse forsvinne. Vi hadde håpet at mottakeren bare sporadisk ville beregne posisjoner som befant seg langt unna fastpunktet. Dette viser seg ikke å være tilfellet. Hvis det kun hadde vært en måling over 100 meter unna, ville denne målingen mest sannsynlig forsvinne ved hjelp av å midle. Når vi fortsatt har målinger som ligger over 100 meter unna fastpunktet etter å ha midlet over 15 målinger kan det tyde på at mottakeren har beregnet flere posisjoner etter hverandre som ligger langt unna. Hvis dette er tilfelle vil ikke disse forsvinne ved å midle over en så kort periode.

Figur 5-22 viser resultatet etter at vi har midlet over 60 målinger. Vi har her klart å redusere avstanden med noen få meter til de målingene som ligger lengst unna. Dette er på langt nær nok i forhold til de kriteriene Kitron satte opp. Figuren viser at ca 89 % av målingene ligger innenfor 10 meter. Dette er ikke særlig mye bedre enn når vi midlet over 15 målinger. Det er heller ikke ønskelig å midle over en periode på 60 sekunder, siden dette kan gjøre den endelige posisjonen veldig tilfeldig.

Begge disse midlingseksemplene førte til en økning av målinger innenfor 10 meter fra ca 78 % til rundt 87 %. Dette er en liten økning som ikke øker nøyaktigheten noe vesentlig. Denne metoden krever mye regnekapasitet og med den beskjedne gevinsten man oppnår er ikke metoden noe særlig å satse på.

6.4.3 Sirkelmidling

Vi prøvde også å forbedre måleresultatene ved hjelp av sirkelmidlingsalgoritmen (se kapittel 4.3.2). Figur 5-23 viser resultatene etter at vi har midlet over 15 målinger hvor minimum 10 målinger skulle være innenfor en radius på 2 meter. Resultatene fra dette forsøket ser vi er temmelig likt resultatene i figur 5-21. 89 % av målingene ligger innefor 10 meter, noe som er ørlite bedre enn tidligere resultater. Som vi ser av figuren har vi ikke klart å fjerne målingene som ligger lengst unna fastpunktet. Dette fører til at metoden ikke egner seg særlig godt når det bare midles over 15 målinger.

Ved å midle over en lengre periode blir resultatet bedre. Figur 5-24 viser at 95 % av alle målingene etter at vi har midlet over 60 målinger ligger innenfor 10 meter. Betingelsen her er at 50 målinger skal ligge innenfor en radius på 2 meter. Sammenlikner vi spredningsdiagrammet i figur 5-24 med resultatet i figur 5-5 ser vi tydelig at nøyaktigheten er blitt bedre. Det er så å si ingen målinger som ligger lengre enn 30 meter unna fastpunktet. Dette er et tilfredsstillende resultat, men å midle over en periode på 60 sekunder er for lenge hvis sauen er i bevegelse. Metoden egner seg best når en med sikkerhet vet at sauen er i ro. Når en alarm utløses kan man med stor sannsynlighet si at sauen ikke beveger seg, å midle over 60 sekunder egner seg derfor best til bruk når en alarm utløses.

6.4.4 Midling med HDOP

Figur 5-20 viser at nøyaktigheten i målingene øker med minkende HDOP. Vi prøvde derfor å midle målinger etter hvilken HDOP verdi de hadde. Figur 5-25 viser resultatet etter at vi har midlet over 15 målinger som har lavere HDOP enn 2. Som spredningsdiagrammet figuren viser blir nøyaktigheten betraktelig forbedret i forhold til utgangspunktet i figur 5-5. 99 % av alle målingene ligger innenfor en nøyaktighet på 10 meter. Minuset med å sette strenge krav til HDOP er at mange av målingene blir filtrert bort. På figur 5-25 gjenstår bare 15 % av alle målingene. Dette viser at å oppnå en HDOP på 2 og lavere er vanskelig. Sauen befinner seg i omgivelser som kanskje gjør forholdene for GPS mottakeren dårligere. Strenge krav til HDOP kan føre til at flestparten av målingene blir filtrert bort. Vi satte derfor litt snillere krav til HDOP verdien til målingene. Figur 5-26 viser resultatene når vi midler over de målingene som har lavere HDOP en 4. Vi ser her at nøyaktigheten er blitt dårligere enn ved midling av målinger med HDOP lavere enn 2, men er fortsatt mye bedre enn utgangspunktet. Figuren viser at ca 93 % av målingene ligger innenfor 10 meter. Det gjenstår 28 % av målingene fra måleserien. Når 37 % av målingene ble klassifisert som gyldige fra GPS mottakeren, må dette sies å være et tilfredsstillende resultat.

Ved å øke midlingsintervallet til 60 målinger øker nøyaktigheten til 95 % av målingene innenfor 10 meter. Figur 5-27 viser resultatene når vi midler over 60 målinger med lavere HDOP enn 4. Selv om resultatet blir bra er det ikke gunstig å midle over en så lang periode hvis sauen er i bevegelse. Å midle over perioder på 60 sekunder bør kun benyttes ved alarm, når vi vet med sikkerhet at sauen er i ro.

6.5 Videre arbeid

I dette prosjektet har vi simulert sauens omgivelser på best mulig måte. Det som gjenstår er å teste på sau i ”ekte” omgivelser for å se om konklusjonene vi tar er riktige.

Vi har benyttet en annen type GPS mottaker enn mottakeren som er i dyreenheten. For å være helt sikre på at oppstartstidene vi foreslår stemmer, bør Kitron sjekke mottakeren som skal være i den endelige versjonen av dyreenheten. De bør finne ut hvor lang tid mottakeren bruker på å finne første gyldige måling og hvor lang den bruker for å komme innenfor 10 meter etter første gyldige måling. Dette bør gjøres både under gode og dårlige forhold, helst i de omgivelsene sauene skal oppholde seg.

7 Konklusjon

Kitron ønsket at nøyaktigheten til Radiobjellesystemet skulle ligge rundt 10 meter. Vi har i denne oppgaven vist at en kan oppnå denne nøyaktigheten under tilnærmet perfekte forhold. Vi har også sett at forholdene GPS mottakeren opplever kan føre til at nøyaktigheten blir dårligere.

Når forholdene blir dårligere øker unøyaktigheten. Disse unøyaktighetene kan imidlertid minimeres i etterkant vha. enkle algoritmer. Som resultatene i kapittel 5.4 viser er midling med HDOP den algoritmen som fungerer best. Simuleringene våre viser at nøyaktigheten øker fra rundt 78 % av målingene innenfor 10 meter til rundt 93 %, ved bruk av denne algoritmen.

Oppstartsmålingene viser at Kitron ikke gir GPS mottakeren nok tid til å beregne en nøyaktig posisjon. Forsøkene våre viser at det er stor forskjell på hvor lang tid GPS mottakeren trenger ved varm- og kaldstart. De viser også at det er minimale forskjeller mellom varmstart med en nøyaktig forrige posisjon, og varmstart med en unøyaktig forrige posisjon. For å unngå de lange oppstartstidene foreslår vi at Kitron kun benytter varmstart. GPS mottakeren trenger TOW, ukenummer og forrige posisjon for å utføre varmstart. Vi foreslår å benytte siste gyldige posisjonsverdi uansett hvor lenge siden denne ble beregnet som forrige posisjon. Denne posisjonen vil alltid være nøyaktig nok, slik at mottakeren raskt skal bli i stand til å beregne en posisjon innenfor 10 meter. Oppstartstiden bør settes til rundt 170 sekunder slik at sannsynligheten for at posisjonen som beregnes er innenfor 10 meter blir så stor som mulig. Den foreslåtte tiden tar også hensyn til at sauen kan være i bevegelse og at forholdene GPS mottakeren opplever kan være dårlige under måleperioden.

I dag godtar Kitron en posisjonsmåling om GPS mottakeren har kontakt med 6 satellitter eller flere. Selv om denne målingen er gyldig vil det ikke si at målingen ligger innenfor en nøyaktighet på 10 meter. Forsøkene våre viser at GPS mottakeren trenger tid for å komme innenfor 10 meter. Ved varmstart foreslår vi derfor å vente rundt 30 sekunder etter første gyldige måling før posisjonsmålingen blir godtatt.

En er ikke garantert at målingen som beregnes etter 170 sekunder er innenfor 10 meter. Det er heller ikke garantert at GPS mottakeren har kontakt med nok satellitter til å beregne en gyldig posisjon i det hele tatt. Hvis sauen er i et område der den umulig kan ha kontakt med 4 satellitter eller flere, er det ikke gunstig å vente i det uendelige for å prøve å beregne en posisjon. Siden dyreenheten er drevet av batteri bør tiden hvor det kun måles ugyldige målinger begrenses til et minimum. Hvis GPS mottakeren ikke er i stand til å beregne en posisjon etter 170 sekunder foreslår vi at det ventes en gitt periode før den gjør et nytt beregningsforsøk. Målet er at hvis GPS mottakeren ikke klarer å beregne en gyldig posisjon etter 170 sekunder skal den med stor sannsynlighet være det etter å ha ventet en gitt tidsperiode.

Ved vanlig drift bør midlingsalgoritmene vi har utviklet ikke brukes over perioder på mer en 15 sekunder. Dette skyldes at sauen kan være i bevegelse mens målingene pågår. Å benytte algoritmene har imidlertid ikke gitt vesentlig bedre resultat enn om målingene bare filtreres etter HDOP verdien. Algoritmene er også vesentlig mer ressurskrevende å

gjennomføre. Under vanlig drift vil vi derfor foreslå å angi at kun målinger med lavere HDOP enn 4 er gyldige målinger. Dette må imidlertid gjøres etter perioden på 30 sekunder som mottakeren har for å komme innenfor 10 meters nøyaktighet. De første målingene etter TTFF trenger ikke ligge innenfor 10 meter selv om HDOP verdien er innenfor de gitte kravene.

Ved alarm er sannsynligheten for at sauen står stille relativt stor og midlingsalgoritmene vi har utviklet kan derfor benyttes over en lengre periode. Det er her spesielt viktig å ha korrekt posisjonsmåling siden sauen her kan sitte fast eller være skadet. Forsøkene våre viser at midling med HDOP over 60 sekunder og HDOP lavere enn 4 gir svært gode resultater. Algoritmen krever en del prosessering men det er ikke noe i veien for at f.eks. basestasjonen kan utføre disse beregningene.

Resultatene vi har oppnådd i denne oppgaven verifiserer de hypotesene vi satte oss i starten av prosjektet (se kapittel 1.4). Ved å benytte metodene vi har kommet fram til er det er fullt mulig å oppnå en nøyaktighet på 10 meter ved bruk av relativt billig GPS utstyr både under gode og dårlige forhold. Dette fører til at bønder lettere kan ha oversikt over buskaper sin noe som igjen kan føre til lavere utgifter både for bonden selv og den norske stat. Når vi kan oppnå nøyaktige resultater for bærbare enheter under de fleste forhold er det bare fantasien som begrenser bruksområdene for denne type teknologi.

Referanser:

- [1] Landbruksdepartementet, (26. april 2004)
<http://odin.dep.no/archive/ldvedlegg/01/35/Utvik073.pdf>
- [2] Direktoratet for naturforvaltning, (26. april 2004)
<http://www.naturforvaltning.no/wbch3.exe?p=1427>
- [3] Prosjektrapport Kitron: Radiobjelle, sluttrapport 2002, Revisjon 1, ”Utvikling og utprøving av en ”radiobjelle” for sauer på beite”, Av: Erik Hardeng
- [4] Dyrevernalliansen, ”Dyrevern fakta: Sau”, (26. april 2004)
http://www.dyrevernalliansen.org/fakta/f_15.php
- [5] Prosjektrapport Kitron: Radiobjelle, industrialisering 2003, Revisjon 2, ”Industrialisering av en ”radiobjelle” for sauer, storfe og reinsdyr på utmarksbeite”, Av: Erik Hardeng
- [6] Prosjektrapport Kitron: SW Design Description for basestasjon, Revisjon PA5, ”SW Design Description for basestasjon”
- [7] Caplex nettleksikon, (13. februar 2004)
<http://www.caplex.no>
- [8] US Marine Corps, ”Marine Artillery Survey Operations”, (13. februar 2004),
http://www.tpub.com/content/USMC/mcwp3167/css/mcwp3167_45.htm
- [9] Nasjonalt senter for romrelatert opplæring, ”Romteknologi og satellitteknikk”, Av: Gunnar Stette
<http://romteknologi.no/books/36/7.html> (14. februar 2004)
<http://www.romteknologi.no/books/33/13.html> (27. april 2004)
- [10] µ-blox AG, ”GPS Basic, Introduction to the system, Application overview”, Av: Jean-Marie_Zogg, (7. februar 2004)
<http://www.u-blox.com/customersupport/docs/GPS-X-02007.pdf>
- [11] NovaPOSTEN, ”Geoide, projeksjon, koordinatsystem og DAK?”, Av: Hans-Jørg Schneidt, (20 april 04)
http://www.novapoint.com/NovaPosten/news/n_0403/09.asp
- [12] Virtual Terrain Project, ”Coordinate Systems and Map Projections”, (20 april 04)
<http://www.vterrain.org/Projections/8358013.png>
- [13] Universitetet i Oslo, Landmåling og fotogrammetri, ”Kompendium i Landmåling”, Av: Trond Eiken
<http://www.geografi.uio.no/undervisning/kurs/gg219/kap-1.pdf>

- [14] Norges Landbrukshøgskole, Institutt for kartfag, landmåling ”Hvorfor datum, referanseflater, punktgrunnlag etc.?” (20 april 04)
<http://www.nlh.no/institutt/ikf/landm/courses/lan102/datum.html>
- [15] Statens Kartverk, ”Transformasjoner”, (20 april 04)
<http://www.statkart.no/IPS/?module=Articles;action=ArticleFolder.publicOpenFolder;ID=725>
- [16] Informasjon fra Regjering og Departement, Miljøverndepartementet, ”Digitale planer etter plan- og bygningsloven - T-1412”, (20 april 04)
odin.dep.no/filarkiv/158085/veil_dig_plan_020228_KO_6_kap3.pdf
- [17] Statens Kartverk, Norgesglasset, (6. mai 2004)
<http://ngis2.statkart.no/norgesglasset/default.html>
- [18] U.S. Naval Observatory, “Current GPS constellation”, (29. januar 2004)
<http://tycho.usno.navy.mil/gpscurr.html>
- [19] The White House, “Statement by the president regarding the united states’ decision to stop degrading global positioning system accuracy”, (16. februar 2004)
http://clinton4.nara.gov/WH/EOP/OSTP/html/0053_2.html
- [20] GPS world, ”E911 Caller Location - Only the Beginning”, Av: Scott D. Pomerantz (16. februar 2004)
<http://www.gpsworld.com/gpsworld/article/articleDetail.jsp?id=26655>
- [21] Federal Aviation Administration, (11. februar 2004)
<http://gps.faa.gov/GPSbasics/index.htm>
- [22] Garmin, ”GPS Guide for beginners”, (29. januar 2004)
http://www.garmin.com/manuals/GPSGuideforBeginners_Manual.pdf
- [23] B. Hofmann-Wellenfor, H. Lichtenegger and J. Collins, “GPS – Theory and Practice, Theory and Practice, Fifth, revised edition”, Springer-Verlag Wien New York, ISBN: 3-211-83534-2, 2001
- [24] Berkeley Seismological Laboratory, “GPS data processing with GAMIT/GLOBK”, Av: Maurizio Battaglia, (16. februar 2004)
<http://www.seismo.berkeley.edu/~battag/GAMITwrkshp/lecturenotes/unit1/unit1.html>
- [25] Royaltek Company LTD, ”Software Interface(SIRF and NMEA)”, (20 april 04)
[http://www.royaltek.com/Manuals/software%20interface%201.1\(SIRF\).pdf](http://www.royaltek.com/Manuals/software%20interface%201.1(SIRF).pdf)
- [26] Mercator, Inc., “Introduction”, (29. januar 2004)
<http://www.mercat.com/QUEST/Intro.htm>
- [27] Earth Observation Magazine - The Magazine for Geographic, Earth, Mapping Information. (23. februar 2004)

- <http://www.eomonline.com/Common/Archives/June96/gps.htm>
- [28] Wilderness Navigation, "GPS Basic", (5. februar 2004)
http://members.aol.com/annapress/GPS1_1.html
- [29] Ahmed El-Rabbany, "Introduction To GPS, The Global Positioning System", Artech House Inc., ISBN: 1-58053-183-0, 2002, Kapittel 3
- [30] Alfred Kleusberg & Richard B. Langley, "The Limitations of GPS", *GPS World*, Mai/April 1990
- [31] TrueTime, Inc., "Stochastic Model Estimation of Network Time Variance", Av: Douglas J. Arnold, (5. februar 2004)
<http://www.truetime.net/pdf/Stochastic-Model-Estimation-of-Network-Time-Variance.pdf>
- [32] TopCon, "Basics of High-Precision Global Positioning Systems", Av: Javad and Nedda Ashjaee, (5. februar 2004)
<http://www.topconps.com/gpstutorial/Chapter3.html>
- [33] Government Computer News, "The clock ticks toward accurate network timing", Av: William Jackson, (30. januar 2004)
http://www.gcn.com/21_9/tech-report/18478-1.html
- [34] B. Hofmann-Wellenfor, H. Lichtenegger and J. Collins, "GPS – Theory and Practice, Theory and Practice, Fifth, revised edition", Springer-Verlag Wien New York, ISBN: 3-211-83534-2, 2001, side 124-125
- [35] University of Nottingham UK, "The Ionospheric Delay Effort For GPS Single - Frequency Users - Analysis Study For Simulation Purposes." Av: Ashraf Farah, (3. februar 2004)
www.nottingham.ac.uk/iessg/papers/isgpap0212.pdf
- [36] Syk av huset?, "Radioaktiv stråling og fare for kreft", Av: Morten Lande, (4. februar 2004)
<http://www.sykavhuset.no/radioaktivitet.htm>
- [37] Educational Observatory Institute, "GPS Errors & Estimating Your Receiver's Accuracy", Av: Sam Wormley, (29. januar 2004)
http://www.edu-observatory.org/gps/gps_accuracy.html
- [38] Satellite Navigation And Positioning Group, "Principles and Practice of GPS Surveying", Av: Chris Rizos, (3. februar 2004)
http://www.gmat.unsw.edu.au/snap/gps/gps_survey/chap6/627.htm
- [39] Mercator, Inc., "Accuracy Measures", (23. februar 2004)
<http://www.mercat.com/QUEST/Accuracy.htm>
- [40] Trimble Navigation Limited , About GPS Technology

<http://www.trimble.com/gps/whydgps.html> (1. mars 2004)

<http://www.trimble.com/gps/errorsources.html> (2. mars 2004)

- [41] Garmin, ”GBR 23 - Beakon Receiver - Owner manual and reference guide”, (2. mars 2004)

http://www.garmin.com/manuals/GBR23_OwnersManual.PDF

Forkortelser

A/S	Anti / Spoofing
BPS	Bits Per Second
BPSK	Binary Phase Shift Keying
C/A	Coarse / Acquisition
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
DGPS	Differential Global Positioning System
DoD	U.S. Department of Defense
DOP	Dilution Of Precision
ED1950	European Datum 1950
FM	Frequency Modulation
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HDOP	Horizontal Dilution of Precision
HOW	Handover Word
IGS	International GPS Service for Geodynamics
NAVSTAR	NAVigation Satellite Timing And Ranging
NGO1948	Norsk Geografiske Oppmåling 1948
NGS	National Geodetic Survey
NIJOS	Norsk Institutt for Jord- Og Skogkartlegging
NMEA	National Marine Electronics Association
PDF	Probability Density Function
PPS	Precision Positioning Service
RDS	Radio Data System
SA	Service Availability
SPS	Standard Positioning Service
TEC	Total Electron Count
TLM	Telemetry Word
TOW	Time Of Week
TTF	Time To First Fix
UTC	Coordinated Universal Time (Likt som Greenwich Mean Time, GMT)
UTM	Universal Transverse Mercator
VDOP	Vertical Dilution Of Position
VHF	Very High Frequency (30-300 MHz)
WGS84	World Geodetic System 1984

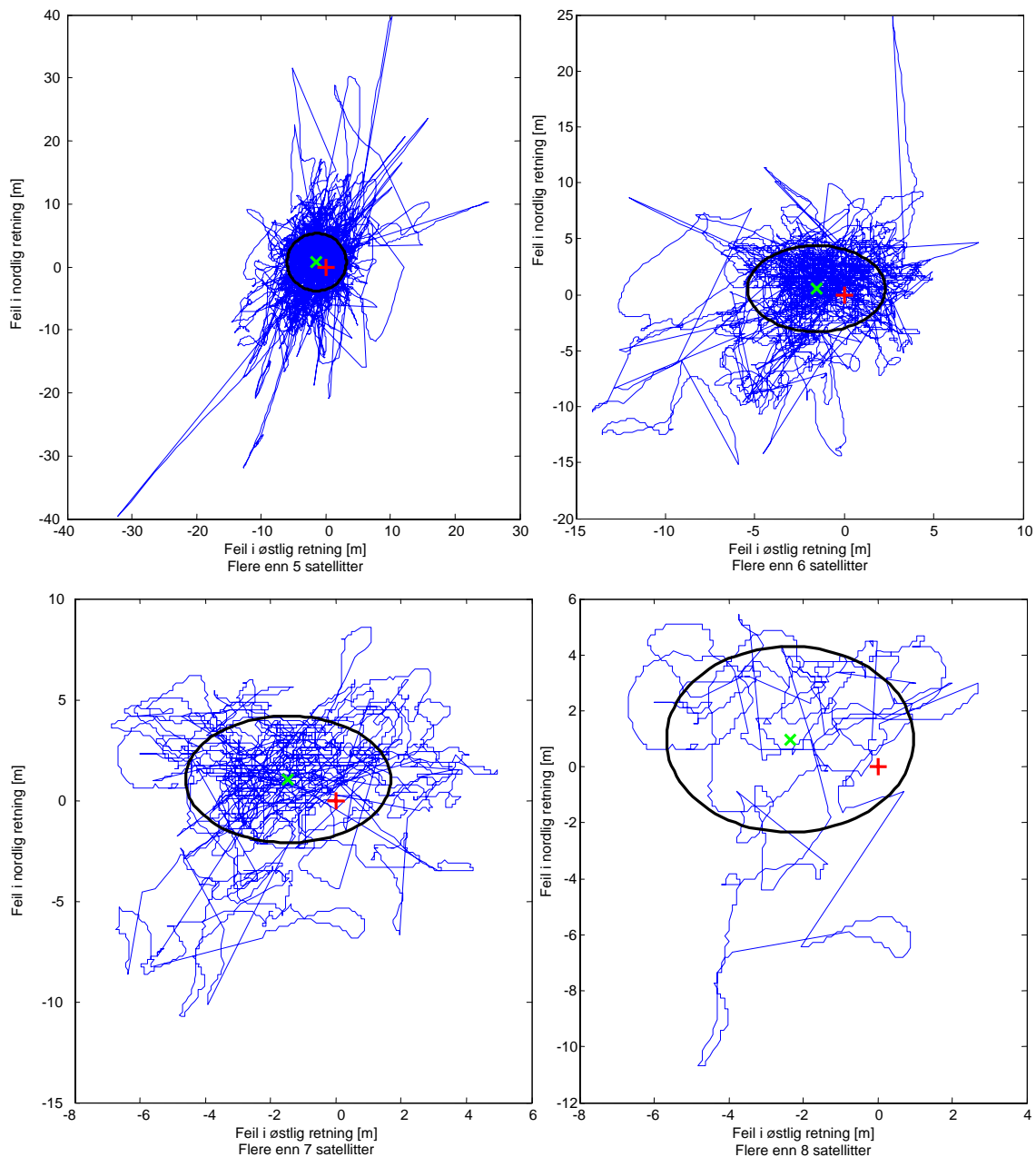
Vedlegg

A	SPREDNINGSDIAGRAM VED FILTRERING.....	A.1
A.1	Spreningsdiagram med satellitt filtrering	A.1
A.2	Spreningsdiagram med HDOP filtrering	A.2
B	KONVERTERINGSPROGRAMMET TXTTOXLS.....	B.1
B.1	Brukergrensesnitt.....	B.1
B.2	Programkode.....	B.2
	B.2.1 TXTtoXLS vindu	B.2
	B.2.2 Hjelp vindu.....	B.4
C	APPLIKASJON FOR ANALYSE AV OPPSTARTSMÅLINGER.....	C.1
C.1	Programkode.....	C.1
D	TESTPROGRAM FOR BEHANDLING AV POSISJONSDATA	D.1
D.1	Brukergrensesnitt.....	D.1
D.2	Kildekode	D.2

A Spredningsdiagram ved filtrering

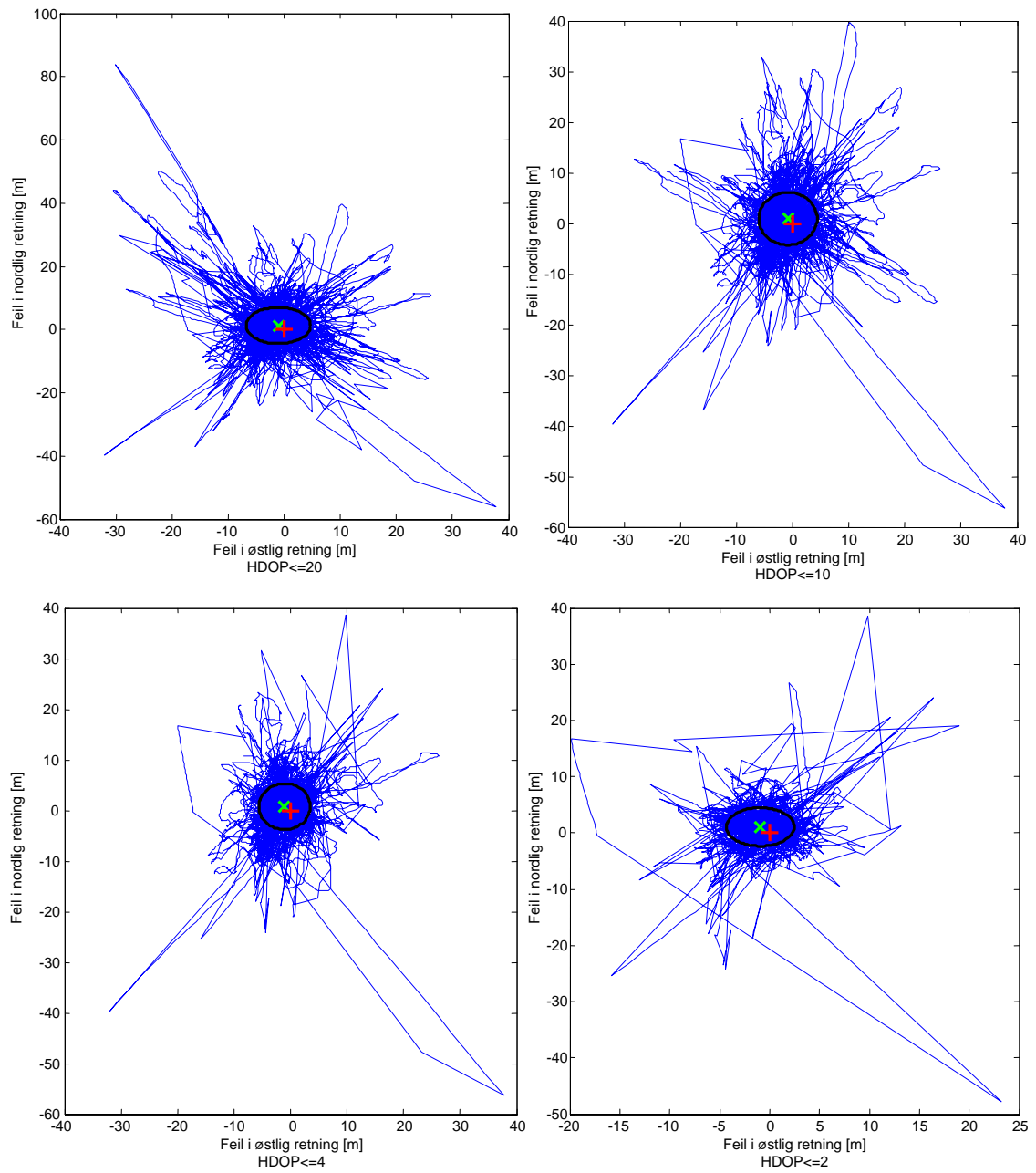
Her viser vi spredningsdiagrammet når vi filtrerer målingene etter antall satellitter GPS mottakeren har kontakt med og etter HDOP.

A.1 Spreningsdiagram med satellitt filtrering



Figur A-1: Mer eller lik 5, 6, 7 og 8 satellitter

A.2 Sprenningsdiagram med HDOP filtrering



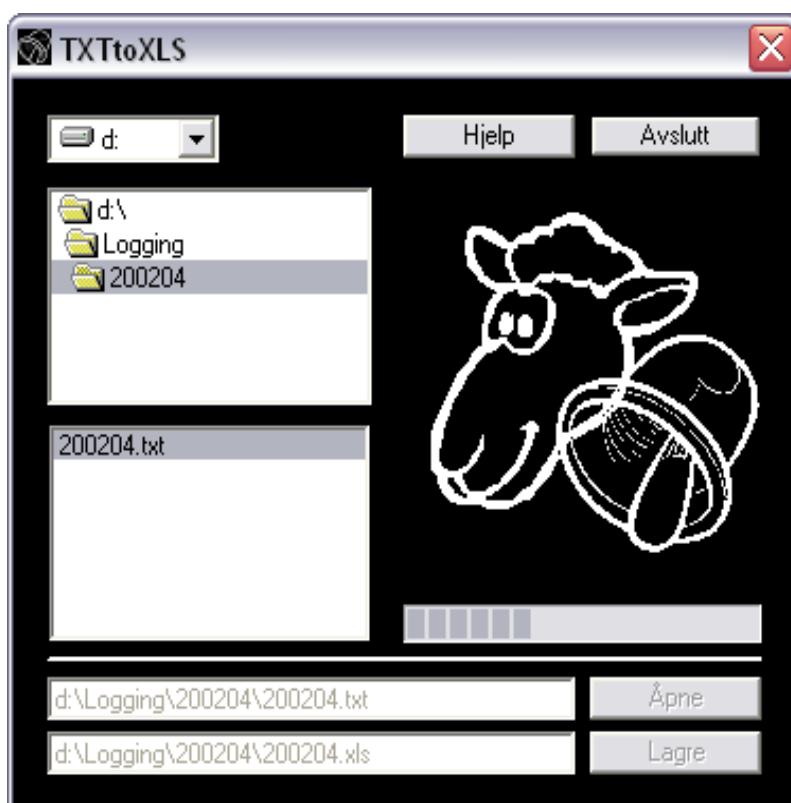
Figur A-2: HDOP mindre eller lik 20, 10, 4 og 2

B Konverteringsprogrammet TXTtoXLS

Vi vil her presentere brukergrensesnitt til TXTtoXLS programmet vårt. Dette programmet er laget for å konvertere loggfilene våre som er i tekst format over i excel format. Programmet henter bare ut den informasjonen som vi mener er relevant for oss. Grunnen til at vi ønsker å ha logg filene i excel format er at de er betydelig raskere å importere i matlab.

B.1 Brukergrensesnitt

Figur B-1 og Figur B-2 viser bruker grensesnittet til programmet. Programmet er enkelt i bruk og hjelpvinduet forklarer hvordan programmet skal brukes.



Figur B-1: Brukergrensesnitt til TXTtoXLS



Figur B-2: Brukergrensesnitt til Hjelp-vindu

B.2 Programkode

B.2.1 TXTtoXLS vindu

```
Private Sub Avslutt_Click()  
    End  
End Sub  
  
Private Sub Dir1_Change()  
    File1.Path = Dir1.Path  
    Load.Enabled = False  
    Save.Enabled = False  
End Sub  
  
Private Sub Dir1_Click()  
    Load.Enabled = False  
    Save.Enabled = False  
    Load_Text.Text = ""  
    Save_Text.Text = ""  
    ProgressBar1.Value = 0.0001  
End Sub  
  
Private Sub Drive1_Change()  
    Dir1.Path = Drive1.Drive + "\"  
    Load_Text.Text = ""  
    Save_Text.Text = ""  
    ProgressBar1.Value = 0.0001  
End Sub  
  
Private Sub File1_Click()  
    Load_Text.Text = ""  
    Save_Text.Text = ""  
    ProgressBar1.Value = 0.0001  
    Save.Enabled = False  
  
    File1.Path = Drive1.Drive  
    File1.FileName = Drive1.Drive  
    File1.Pattern = "*.txt"  
  
    fil = Split(File1.FileName, ".")  
  
    If fil(1) = "txt" Then Load.Enabled = True  
End Sub  
  
Private Sub Form_Load()  
    Dir1.Path = "c:\"  
    Dir1.Path = Drive1.Drive  
    File1.Pattern = "*.txt"  
End Sub  
  
Private Sub Hjelp_Click()  
    Hjelp_Vindu.Visible = True  
End Sub  
  
Private Sub Load_Click()  
    ProgressBar1.Value = 0.0001  
    fil = Split(File1.FileName, ".")  
    If Right(Dir1.Path, 1) = "\" Then  
        Load_Text.Text = Dir1.Path + File1.FileName  
        Save_Text.Text = Dir1.Path + fil(0) + ".xls"  
    Else
```

```

        Load_Text.Text = Dir1.Path + "\" + File1.FileName
        Save_Text.Text = Dir1.Path + "\" + fil(0) + ".xls"
    End If
    Save.Enabled = True
End Sub

Private Sub Save_Click()
    Screen.MousePointer = vbHourglass
    Load.Enabled = False
    Save.Enabled = False
    ProgressBar1.Value = 0.0001

    Open Load_Text.Text For Input As #1

    Dim xlapp As Excel.Application
    Dim xlbook As Excel.Workbook
    Dim xlsheet As Excel.Worksheet

    Set xlapp = CreateObject("excel.application")
    Set xlbook = xlapp.Workbooks.Add
    Dim ark As Integer
    ark = 1
    Set xlsheet = xlbook.Worksheets(ark)

    xlsheet.Activate
    xlsheet.StandardWidth = 20

    Length = LOF(1)
    Length = (-Sgn(Length / 128) * Int(-Abs(Length / 128)))

    Dim location As Long
    Dim i As Long
    i = 1

    Do Until EOF(1)
        Line Input #1, temp
        'temp = Mid(temp, 10)          'Benyttet til loggfiler fra u-center
        myarray = Split(temp, ",")

        If myarray(0) = "$GPGGA" Then
            xlsheet.Cells(i, 1) = myarray(1)
            xlsheet.Cells(i, 2) = myarray(2)
            xlsheet.Cells(i, 3) = myarray(4)
            xlsheet.Cells(i, 4) = myarray(6)
            xlsheet.Cells(i, 5) = myarray(7)
            xlsheet.Cells(i, 6) = myarray(8)
            xlsheet.Cells(i, 7) = myarray(9)

            If i = 65536 Then
                ark = ark + 1
                Set xlsheet = xlbook.Worksheets(ark)
                i = 0
            End If
            i = i + 1

            location = Loc(1)
            ProgressBar1.Value = (location) / Length * 100
        End If
    Loop

```

```
Loop

Close 1

Call xlbook.Close(SaveChanges:=True, FileName:=Save_Text.Text)

ProgressBar1.Value = 100
Screen.MousePointer = vbArrow
Load.Enabled = True
Save.Enabled = True
End Sub
```

B.2.2 Hjelp vindu

```
Private Sub Form_Load()
    Labell = "Velg .txt til som skal overføres til .xls. " & _
    "Trykk på Åpne-knappen for å åpne filen. Du ser da filen " & _
    "som er åpnet og banen til filen som blir lagret. Trykk " & _
    "på lagre for å lagre filen. Finnes det flere enn 65536 " & _
    "GPGGA meldinger, vil dette vli lagret i neste ark da excel " & _
    "ikke har støtte for større ark. Ikke åpne nytt Excel dokument " & _
    "når konvertering pågår"
End Sub

Private Sub OK_Click()
    Hjelp_Vindu.Visible = False
End Sub
```

C Applikasjon for analyse av oppstartsmålinger

Her presenterer vi Matlab koden som blir benyttet til å analysere oppstartsmålingene vi har gjort. Programmet gir en figur som viser hvor lang tid GPS mottakeren bruker på å gi ut første gyldige posisjonsmåling og en figur som viser nøyaktigheten etter dette.

C.1 Programkode

```
clc
clear

mstruct = defaultm('utm');
mstruct.zone = '32V';
mstruct.geoid = almanac('earth','geoid','m','grs80');
mstruct = defaultm(utm(mstruct));

filnavn='d:\logging\kaldstart\kaldstart.xls'; %Bane til fil som skal
analyseres

PosisjonsData=xlsread(filnavn,'Ark1');
%PosisjonsData=cat(1,PosisjonsData,xlsread(filnavn,'Ark2'));
%PosisjonsData=cat(1,PosisjonsData,xlsread(filnavn,'Ark3'));

[Rader,Kolonner] = size(PosisjonsData);

y = 6466369.814;
x = 475467.100;

%Gjør om lengdegrad til desimalform
Breddegrad_desimal=floor(PosisjonsData(:,2)./100)+(PosisjonsData(:,2)./100-
floor(PosisjonsData(:,2)./100)).*100/60;
Lengdegrad_desimal=floor(PosisjonsData(:,3)./100)+(PosisjonsData(:,3)./100-
floor(PosisjonsData(:,3)./100)).*100/60;

%Konverterer posisjon fra WGS84 til UTM sone32
[Lengdegrad_UTM,Breddegrad_UTM] =
mfwdtran(mstruct,Breddegrad_desimal,Lengdegrad_desimal);

%%%%%%%%%%%%%%Histogram tid
antall=length(PosisjonsData);
j=1;
ny_en=1;

for i=1:antall
    if PosisjonsData(i,4)==0 && ny_en==1
        tid=PosisjonsData(i-1,1);
        ny_en=0;
    elseif PosisjonsData(i,4)==1 && ny_en==0
        histogramData(j)=hms2sec(PosisjonsData(i,1)/100)-
hms2sec(tid/100);
        j=j+1;
        ny_en=1;
    end
end

antall_Hist=length(histogramData);
x_akse=1:max(histogramData)+2;
```



```

Hist=hist(histogramData,x_akse)./antall_Hist*100;
antall_Hist=length(Hist);
for i=1:antall_Hist
    cdf(i)=sum(Hist(1:i));
end

bar_handles = bar(x_akse,Hist);
ylim([0 ceil(max(Hist)/10)*10+ceil(max(Hist)/10)*10/100]); %For å
skalere y-aksene likt
xlim([0 ceil(antall_Hist/10)*10])
ax1 = gca;

xlabel('Sekunder')
ylabel('Antall [%] (PDF)');

ax2 = axes('Position',get(ax1,'Position'),...
    'XAxisLocation','bottom',...
    'YAxisLocation','right',...
    'Color','none',...
    'XColor','r','YColor','r');
plot_handles = plot(x_akse,cdf,'LineWidth',1.5,'color','g');
ylim([0 101])

legend_handles = [bar_handles(1);plot_handles];
legend(legend_handles,'PDF','CDF',2);

set(ax2,'YAxisLocation','right','Color','none','XTickLabel',[])
set(ax2,'XLim',get(ax1,'XLim'),'Layer','top')

ylabel('Antall [%] (CDF)')

ylimits1 = get(ax1,'YLim'); %"lik" y-akse
ylimits1(2)=floor(ylimits1(2));
yinc1 = (ylimits1(2)-ylimits1(1))/10;
aal=[ylimits1(1):yinc1:ylimits1(2)];
set(ax1,'YTick',[ylimits1(1):yinc1:ylimits1(2)])

pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Nøyaktighet etter fix
feil=sqrt((Lengdegrad_UTM-x).^2+(Breddegrad_UTM-y).^2);

antall=length(PosisjonsData);
k=1;
noyaktighet(k)=0;
antall_noyaktighet(k)=0;
antall_verdier=1;
ny_en=2;
etter_forste=0;

for i=1:antall
    if ((PosisjonsData(i,4)==0 && ny_en==0) || (PosisjonsData(i,4)==0 &&
ny_en==2))
        k=1;
        ny_en=1;
        etter_forste=1;
    elseif PosisjonsData(i,4)==1 && k>=antall_verdier && ny_en~=2
        noyaktighet(k)=feil(i);
        antall_noyaktighet(k)=1;

        k=k+1;
        antall_verdier=max(k,antall_verdier);
    end
end

```

```
        ny_en=0;
    elseif PosisjonsData(i,4)==1 && ny_en~=2
        noyaktighet(k)=noyaktighet(k)+feil(i);
        antall_noyaktighet(k)=antall_noyaktighet(k)+1;

        k=k+1;
        ny_en=0;
    end
end

noyaktighet=noyaktighet./antall_noyaktighet;

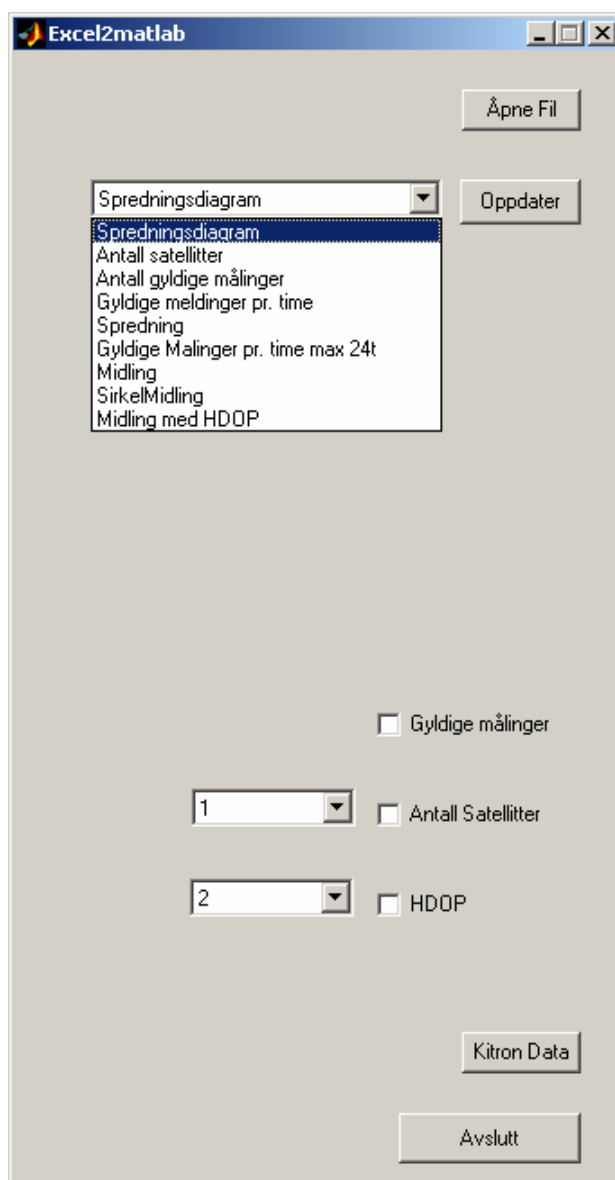
lengde=length(noyaktighet);
x_akse=1:lengde;
delete(gcf)
plot(x_akse,noyaktighet)
xlim([0 120]);
xlabel('Sekunder')
ylabel('Feil [m]')
```

D Testprogram for behandling av posisjonsdata

For å analysere posisjonsmålingen vi gjorde laget vi et testprogram i Matlab. Vi benyttet dette programmet til å analysere dataene samt å eksperimentere med de forskjellige algoritmene vi utviklet. Applikasjonen leser Excel filer med følgende format i kolonnene fra A og utover: Klokkeslett, Breddegrad, Lengdegrad, Gyldighet, Antall satellitter, HDOP, Høyde. Applikasjonen tar utgangspunkt i fastpunktet vi refererte alle målingene våre til. Det er derfor ikke mulig å benytte dette programmet med data fra andre målepunkt uten å modifisere koden.

I programmet kan brukeren blant annet sette de kriteriene han vil for hvordan en gyldig måling skal være. Hvilke valg som kan gjøres og kriterier som kan settes er avhengig av hvilken "funksjon" brukeren ønsker å benytte. Brukeren får presentert resultatene i form av et spredningsdiagram og et histogram som viser avstandene fra fastpunktet.

D.1 Brukergrensesnitt



Figur D-1: Brukergrensesnitt Testprogram

Figur D-1 viser de forskjellige operasjonene brukeren kan utføre, samt noen av kriteriene brukeren kan sette til en gyldig måling. Figuren inneholder også fire andre felt hvor brukeren kan taste inn forskjellige kriterier etter hvilken operasjon han velger. Disse er ikke vist i figuren.

Funksjonene:

- **Spredningsdiagram**
Programmet tegner spredningsdiagrammet for den valgte måleperioden. Det kan settes forskjellige kriterier til en godkjent måling.
- **Antall Satellitter**
Programmet viser et histogram over hvor mange satellitter mottakeren har hatt kontakt med i løpet av måleperioden.
- **Antall Gyldige Målinger**
Programmet viser et histogram over hvor mange målinger som er gyldige/ugyldige i måleperioden.
- **Gyldige Målinger Pr. Time**
Programmet viser et histogram over hvor mange gyldige/ugyldige målinger det er pr. time i måleperioden.
- **Spredning**
Programmet deler inn avstandene mellom beregnet posisjon og fastpunktet i ”båser”. Brukeren velger størrelsen på hver ”bås”. Det kan også settes forskjellige kriterier til en godkjent måling. De gjenværende målingene vises i et histogram.
- **Gyldige Målinger Pr Time Max 24t**
Programmet viser et histogram over antall gyldige/ugyldige målinger pr. time i måleperioden. Histogrammet viser kun 24 timer, noe som betyr at 1. og 24. time legges sammen.
- **Midling**
Programmet kjører Midlings-algoritmen. Brukeren kan velge hvor mange målinger som skal undersøkes samtidig, og hvor mange av disse som må være gyldige for at det skal midles.
- **Sirkelmidling**
Programmet kjører Sirkelmidlings-algoritmen. Brukeren velger hvor mange målinger som skal undersøkes, og hvor mange av disse som må være gyldige. Det kan også settes hvor stor radius sirklene skal ha og hvor mange som skal brukes.
- **Midling med HDOP**
Programmet kjører MidlingMedHDOP-algoritmen. Brukeren velger hvor mange målinger som skal undersøkes samtidig, og hvor mange av disse som må ha HDOP innefor det valgte HDOP kriteriet.

D.2 Kildekode

```
function varargout = Excel2matlab(varargin)
% EXCEL2MATLAB M-file for Excel2matlab.fig

% Dette er et testprogram som er utviklet for å analysere GPS data.
% Formatet må være i excel format. Kolonnene i regnearket må være
som
% følter: TID, Breddegrad(desimalverdi), Lengdegrad(desimalverdi),
% Gyldighet, Antall satellitter, HDOP, Høyde
%
```

```

%      Programmet er ikke optimalisert

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @Excel2matlab_OpeningFcn, ...
                  'gui_OutputFcn',  @Excel2matlab_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Excel2matlab is made visible.
function Excel2matlab_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Excel2matlab (see VARARGIN)

% Choose default command line output for Excel2matlab
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Excel2matlab wait for user response (see UIRESUME)
% uiwait(handles.Excel2matlab);

% --- Outputs from this function are returned to the command line.
function varargout = Excel2matlab_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Kjøres når åpneknappen trykkes.
function apneknapp_Callback(hObject, eventdata, handles)
% hObject    handle to apneknapp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    set(handles.Gylidge_Malinger_Chk, 'Value', 0);
    set(handles.Antall_Satellitter_Chk, 'Value', 0);
    set(handles.HDOP_Chk, 'Value', 0);

    data = defaultm('utm');
    data.zone = '32V';
    data.geoid = almanac('earth', 'geoid', 'm', 'grs80');

```

```

data = defaultm(utm(data));

data.Breddegrad_fast=6466369.814;
data.Lengdegrad_fast=475467.100;

[FileName,PathName] = uigetfile('*.xls','Velg XLS-file');
filnavn=cat(2,PathName,FileName);
[Type, Beskrivelse] = xlsinfo(filnavn); %Finner antall ark xls fila
antallark=length(Beskrivelse);

for i=1:antallark %Sløyfa leser alle ark i excelfila
    if i==1
        PosisjonsData=xlsread(filnavn,char(Beskrivelse(i)));
    else

PosisjonsData=cat(1,PosisjonsData,xlsread(filnavn,char(Beskrivelse(i))));
    end
end

[rows,columns] = size(PosisjonsData); %Finner antall koller og
rader

Breddegrad_desimal=floor(PosisjonsData(:,2)./100)+(PosisjonsData(:,2)./10
0-floor(PosisjonsData(:,2)./100)).*100/60; %Gjør om breddegrad til
desimalform

Lengdegrad_desimal=floor(PosisjonsData(:,3)./100)+(PosisjonsData(:,3)./10
0-floor(PosisjonsData(:,3)./100)).*100/60; %Gjør om lengdegrad til
desimalform

data.Breddegrad_WGS84=Breddegrad_desimal;
data.Lengdegrad_WGS84=Lengdegrad_desimal;
set(handles.Antall_Satellitter_Chk,'Visible','On');
set(handles.Gylidge_Malinger_Chk,'Visible','On');
set(handles.HDOP_tekst,'Visible','On');
set(handles.HDOP_Chk,'Visible','On');

%HUSK AT KOORDINATSYSTEMET FOR KART ER OMVENDT FRA VANLIGE
KOORDINATSYSTEM!!!!
[Lengdegrad_UTM,Breddegrad_UTM] =
mfwdtran(data,Breddegrad_desimal,Lengdegrad_desimal); %konverterer fra
WGS84 til UTM sone32

data.Tid=PosisjonsData(:,1);
data.Breddegrad=Breddegrad_UTM;
data.Lengdegrad=Lengdegrad_UTM;
data.Gyldig=PosisjonsData(:,4);
data.AntallSatellitter=PosisjonsData(:,5);
data.HDOP=PosisjonsData(:,6);
data.Hoyde=PosisjonsData(:,7);
data.Rader=rows;
data.Kolonner=columns;
data.Lengste=50;
data.Storrelse=5;
data.GyldigeMalinger_CHK_Status=0;
data.Antall_Satellitter_CHK_Status=0;
data.Antall_Satellitter_Tekst=1;
data.HDOP_CHK_Status=0;
data.HDOP_tekst=100;
data.tekstFelt3=5;
data.tekstFelt4=3;

```

```

setappdata(gcf, 'metricdata', data);

set(handles.lengsteAvstand, 'String', data.Lengste);
set(handles.storrelse, 'String', data.Storrelse);

set(handles.oppdateringsknapp, 'Enable', 'On');
set(handles.meny, 'Enable', 'On');
set(handles.Antall_Satellitter_Tekst, 'Visible', 'On');

hoyeste_Antall_Satellitter=max(data.AntallSatellitter); %Setter
menyverdiene i Antall Satellitt popupmeny
for i=1:hoyeste_Antall_Satellitter
s(i)=i;
menyString(i)=cellstr(int2str(s(i)));
end

set(handles.Antall_Satellitter_Tekst, 'String', menyString);

% --- Executes during object creation, after setting all properties.
function meny_CreateFcn(hObject, eventdata, handles)
% hObject    handle to meny (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject, 'BackgroundColor', 'white');
else

set(hObject, 'BackgroundColor', get(0, 'defaultUicontrolBackgroundColor'));
end

set(hObject, 'String', {'Spredningsdiagram', 'Antall satellitter',
'Antall gyldige målinger', 'Gyldige meldinger pr. time',
'Spredning', 'Gyldige Malinger pr. time max
24t', 'Midling', 'SirkelMidling', 'Midling med HDOP'});

% --- Executes on selection change in meny.
function meny_Callback(hObject, eventdata, handles)
% hObject    handle to meny (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
popup_sel_index = get(handles.meny, 'Value');
switch popup_sel_index
case 1
    set(handles.lengsteAvstand, 'Visible', 'Off');
    set(handles.storrelse, 'Visible', 'Off');
    set(handles.storrelseTekst, 'Visible', 'Off');
    set(handles.lengsteAvstandTekst, 'Visible', 'Off');
    set(handles.Antall_Satellitter_Tekst, 'Visible', 'On');
    set(handles.Antall_Satellitter_Chk, 'Visible', 'On');
    set(handles.Gyldige_Malinger_Chk, 'Visible', 'On');
    set(handles.HDOP_Chk, 'Visible', 'On');
    set(handles.HDOP_tekst, 'Visible', 'On');
    set(handles.Antall_Gyldige_Malinger, 'Visible', 'On');
    set(handles.tekstFelt3_tekst, 'Visible', 'Off');
    set(handles.tekstFelt4_tekst, 'Visible', 'Off');
    set(handles.tekstFelt3, 'Visible', 'Off');

```

```

        set(handles.tekstFelt4, 'Visible', 'Off');

    case 2
        set(handles.lengsteAvstand, 'Visible', 'Off');
        set(handles.storrelse, 'Visible', 'Off');
        set(handles.storrelseTekst, 'Visible', 'Off');
        set(handles.lengsteAvstandTekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
        set(handles.Gylidge_Malinger_Chk, 'Visible', 'Off');
        set(handles.HDOP_Chk, 'Visible', 'Off');
        set(handles.HDOP_tekst, 'Visible', 'Off');
        set(handles.Antall_Gyldige_Malinger, 'Visible', 'Off');
        set(handles.tekstFelt3, 'Visible', 'Off');
        set(handles.tekstFelt4, 'Visible', 'Off');
        set(handles.tekstFelt3_tekst, 'Visible', 'Off');
        set(handles.tekstFelt4_tekst, 'Visible', 'Off');

    case 3
        set(handles.lengsteAvstand, 'Visible', 'Off');
        set(handles.storrelse, 'Visible', 'Off');
        set(handles.storrelseTekst, 'Visible', 'Off');
        set(handles.lengsteAvstandTekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
        set(handles.Gylidge_Malinger_Chk, 'Visible', 'Off');
        set(handles.HDOP_Chk, 'Visible', 'Off');
        set(handles.HDOP_tekst, 'Visible', 'Off');
        set(handles.Antall_Gyldige_Malinger, 'Visible', 'Off');
        set(handles.tekstFelt3, 'Visible', 'Off');
        set(handles.tekstFelt4, 'Visible', 'Off');
        set(handles.tekstFelt3_tekst, 'Visible', 'Off');
        set(handles.tekstFelt4_tekst, 'Visible', 'Off');

    case 4
        set(handles.lengsteAvstand, 'Visible', 'Off');
        set(handles.storrelse, 'Visible', 'Off');
        set(handles.storrelseTekst, 'Visible', 'Off');
        set(handles.lengsteAvstandTekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
        set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
        set(handles.Gylidge_Malinger_Chk, 'Visible', 'Off');
        set(handles.HDOP_Chk, 'Visible', 'Off');
        set(handles.HDOP_tekst, 'Visible', 'Off');
        set(handles.Antall_Gyldige_Malinger, 'Visible', 'Off');
        set(handles.tekstFelt3, 'Visible', 'Off');
        set(handles.tekstFelt4, 'Visible', 'Off');
        set(handles.tekstFelt3_tekst, 'Visible', 'Off');
        set(handles.tekstFelt4_tekst, 'Visible', 'Off');

    case 5
        set(handles.lengsteAvstand, 'Visible', 'On');
        set(handles.storrelse, 'Visible', 'On');
        set(handles.storrelseTekst, 'Visible', 'On');
        set(handles.lengsteAvstandTekst, 'Visible', 'On');
        set(handles.lengsteAvstandTekst, 'String', 'Angi lengste
avstand');
        set(handles.storrelseTekst, 'String', 'Angi størrelsen på hver
søyle');
        set(handles.Antall_Satellitter_Tekst, 'Visible', 'On');
        set(handles.Antall_Satellitter_Chk, 'Visible', 'On');
        set(handles.Gylidge_Malinger_Chk, 'Visible', 'On');
        set(handles.HDOP_Chk, 'Visible', 'On');
        set(handles.HDOP_tekst, 'Visible', 'On');

```



```

set(handles.Antall_Gyldige_Malinger, 'Visible', 'On');
set(handles.tekstFelt3, 'Visible', 'Off');
set(handles.tekstFelt4, 'Visible', 'Off');
set(handles.tekstFelt3_tekst, 'Visible', 'Off');
set(handles.tekstFelt4_tekst, 'Visible', 'Off');

case 6
set(handles.lengsteAvstand, 'Visible', 'Off');
set(handles.storrelse, 'Visible', 'Off');
set(handles.storrelseTekst, 'Visible', 'Off');
set(handles.lengsteAvstandTekst, 'Visible', 'Off');
set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
set(handles.Gyldige_Malinger_Chk, 'Visible', 'Off');
set(handles.HDOP_Chk, 'Visible', 'Off');
set(handles.HDOP_tekst, 'Visible', 'Off');
set(handles.Antall_Gyldige_Malinger, 'Visible', 'Off');
set(handles.tekstFelt3, 'Visible', 'Off');
set(handles.tekstFelt4, 'Visible', 'Off');
set(handles.tekstFelt3_tekst, 'Visible', 'Off');
set(handles.tekstFelt4_tekst, 'Visible', 'Off');

case 7
set(handles.lengsteAvstand, 'Visible', 'On');
set(handles.storrelse, 'Visible', 'On');
set(handles.storrelseTekst, 'Visible', 'On');
set(handles.lengsteAvstandTekst, 'Visible', 'On');
set(handles.lengsteAvstandTekst, 'String', 'Antall måliger det
skal middles over');
set(handles.storrelseTekst, 'String', 'Minimum antall gyldige
måliger')
set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
set(handles.Gyldige_Malinger_Chk, 'Visible', 'Off')
set(handles.HDOP_Chk, 'Visible', 'Off');
set(handles.HDOP_tekst, 'Visible', 'Off');
set(handles.Antall_Gyldige_Malinger, 'Visible', 'On');
set(handles.tekstFelt3, 'Visible', 'Off');
set(handles.tekstFelt4, 'Visible', 'Off');
set(handles.tekstFelt3_tekst, 'Visible', 'Off');
set(handles.tekstFelt4_tekst, 'Visible', 'Off');

case 8
set(handles.lengsteAvstand, 'Visible', 'On');
set(handles.storrelse, 'Visible', 'On');
set(handles.storrelseTekst, 'Visible', 'On');
set(handles.lengsteAvstandTekst, 'Visible', 'On');
set(handles.lengsteAvstandTekst, 'String', 'Antall måliger det
skal middles over');
set(handles.storrelseTekst, 'String', 'Minimum antall gyldige
måliger')
set(handles.Antall_Satellitter_Tekst, 'Visible', 'Off');
set(handles.Antall_Satellitter_Chk, 'Visible', 'Off');
set(handles.Gyldige_Malinger_Chk, 'Visible', 'Off')
set(handles.HDOP_Chk, 'Visible', 'Off');
set(handles.HDOP_tekst, 'Visible', 'Off');
set(handles.Antall_Gyldige_Malinger, 'Visible', 'On');
set(handles.tekstFelt3, 'Visible', 'On');
set(handles.tekstFelt4, 'Visible', 'On');
set(handles.tekstFelt3_tekst, 'Visible', 'On');
set(handles.tekstFelt4_tekst, 'Visible', 'On');
set(handles.tekstFelt3_tekst, 'String', 'Radius størrelse [m]');
set(handles.tekstFelt4_tekst, 'String', 'Antall sirkler')

```

```

case 9
    set(handles.lengsteAvstand,'Visible','On');
    set(handles.storrelse,'Visible','On');
    set(handles.storrelseTekst, 'Visible', 'On');
    set(handles.lengsteAvstandTekst, 'Visible','On')
    set(handles.lengsteAvstandTekst, 'String', 'Antall målinger det
skal middles over');
    set(handles.storrelseTekst, 'String', 'Antall målinger med gyldig
HDOP')
    set(handles.Antall_Satellitter_Tekst,'Visible','Off');
    set(handles.Antall_Satellitter_Chk,'Visible','Off');
    set(handles.Gyldige_Malinger_Chk,'Visible','Off')
    set(handles.HDOP_Chk,'Visible','Off');
    set(handles.HDOP_tekst,'Visible','Off');
    set(handles.Antall_Gyldige_Malinger,'Visible','On');
    set(handles.tekstFelt3,'Visible','On');
    set(handles.tekstFelt4,'Visible','Off');
    set(handles.tekstFelt3_tekst,'Visible','On');
    set(handles.tekstFelt4_tekst,'Visible','Off');
    set(handles.tekstFelt3_tekst, 'String', 'HDOP lavere enn:');
end

```

```

% --- Kjøres når oppdateringsknappen trykkes.
function oppdateringsknapp_Callback(hObject, eventdata, handles)
% hObject      handle to oppdateringsknapp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

popup_sel_index = get(handles.meny, 'Value');
switch popup_sel_index
    case 1
        spredningsDiagram(handles);
    case 2
        antallSatellitt;
    case 3
        antallGyldigeMalinger;
    case 4
        PrTime;
    case 5
        spredning(handles);
    case 6
        PrTime24;
    case 7
        Middling(handles);
    case 8
        SirkelMiddling(handles);
    case 9
        SnittMedHDOP(handles);
end

```

```

% --- Kjøres når avsluttknappen trykkes.
function avsluttknapp_Callback(hObject, eventdata, handles)
% hObject      handle to avsluttknapp (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    delete(handles.Excel2matlab)

```

```

% Funksjonen plotter alle posisjonene i måleserien som er åpnet. Det kan
velges hvilke kriterier som skal settes
function spredningsDiagram(handles)
    data=getappdata(gcbf,'metricdata');
    utm_x_fast=data.Breddegrad_fast; %eksaket punkt på tak HIA nordlig
    rettning
    utm_y_fast=data.Lengdegrad_fast; %eksaket punkt på tak HIA østlig
    rettning

    if data.GyldigeMalinger_CHK_Status==0 &&
    data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==0
        Gyldige=': ';
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
        Breddegrad_WGS84=data.Breddegrad_WGS84(Gyldige);
        Lengdegrad_WGS84=data.Lengdegrad_WGS84(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
    data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==1
        Gyldige=find(data.HDOP<data.HDOP_tekst);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
        Breddegrad_WGS84=data.Breddegrad_WGS84(Gyldige);
        Lengdegrad_WGS84=data.Lengdegrad_WGS84(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==1 &&
    data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==0
        Gyldige=find(data.Gyldig);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
        Breddegrad_WGS84=data.Breddegrad_WGS84(Gyldige);
        Lengdegrad_WGS84=data.Lengdegrad_WGS84(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==1 &&
    data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==1
        Gyldige=find(data.Gyldig);
        Breddegrad_gyldige=data.Breddegrad(Gyldige);
        Lengdegrad_gyldige=data.Lengdegrad(Gyldige);

        Breddegrad_WGS84_gyldige=data.Breddegrad_WGS84(Gyldige);
        Lengdegrad_WGS84_gyldige=data.Lengdegrad_WGS84(Gyldige);
        HDOP_gyldige=data.HDOP(Gyldige);
        HDOP_gyldig_under_antall=find(HDOP_gyldige<data.HDOP_tekst);

        Breddegrad_UTM=Breddegrad_gyldige(HDOP_gyldig_under_antall);
        Lengdegrad_UTM=Lengdegrad_gyldige(HDOP_gyldig_under_antall);

        Breddegrad_WGS84=Breddegrad_WGS84_gyldige(HDOP_gyldig_under_antall);
        Lengdegrad_WGS84=Lengdegrad_WGS84_gyldige(HDOP_gyldig_under_antall);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
    data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==0

        Gyldige=find(data.AntallSatellitter>=data.Antall_Satellitter_Tekst);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
        Breddegrad_WGS84=data.Breddegrad_WGS84(Gyldige);
        Lengdegrad_WGS84=data.Lengdegrad_WGS84(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
    data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==1

        Gyldige=find(data.AntallSatellitter>=data.Antall_Satellitter_Tekst);

```

```

Breddegrad_gyldige=data.Breddegrad(Gyldige);
Lengdegrad_gyldige=data.Lengdegrad(Gyldige);
Breddegrad_WGS84_gyldige=data.Breddegrad_WGS84(Gyldige);
Lengdegrad_WGS84_gyldige=data.Lengdegrad_WGS84(Gyldige);
HDOP_gyldige=data.HDOP(Gyldige);

HDOP_gyldige_Antall=find(HDOP_gyldige<data.HDOP_tekst);

Breddegrad_UTM=Breddegrad_gyldige(HDOP_gyldige_Antall);
Lengdegrad_UTM=Lengdegrad_gyldige(HDOP_gyldige_Antall);
Breddegrad_WGS84=Breddegrad_WGS84_gyldige(HDOP_gyldige_Antall);
Lengdegrad_WGS84=Lengdegrad_WGS84_gyldige(HDOP_gyldige_Antall);
elseif data.GyldigeMalinge_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==0
Gyldige=find(data.Gyldig);

Breddegrad_gyldige_UTM=data.Breddegrad(Gyldige);
Lengdegrad_gyldige_UTM=data.Lengdegrad(Gyldige);
Breddegrad_gyldige_WGS84=data.Breddegrad_WGS84(Gyldige);
Lengdegrad_gyldige_WGS84=data.Lengdegrad_WGS84(Gyldige);
AntallSatellitter_gyldige=data.AntallSatellitter(Gyldige);

Gyldige_RiktigAntall_Sat=find(AntallSatellitter_gyldige>=data.Antall_Sate
llitter_Tekst);

Breddegrad_UTM=Breddegrad_gyldige_UTM(Gyldige_RiktigAntall_Sat);
Lengdegrad_UTM=Lengdegrad_gyldige_UTM(Gyldige_RiktigAntall_Sat);

Breddegrad_WGS84=Breddegrad_gyldige_WGS84(Gyldige_RiktigAntall_Sat);

Lengdegrad_WGS84=Lengdegrad_gyldige_WGS84(Gyldige_RiktigAntall_Sat);
elseif data.GyldigeMalinge_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==1
Gyldige=find(data.Gyldig);

Breddegrad_gyldige_UTM=data.Breddegrad(Gyldige);
Lengdegrad_gyldige_UTM=data.Lengdegrad(Gyldige);
Breddegrad_gyldige_WGS84=data.Breddegrad_WGS84(Gyldige);
Lengdegrad_gyldige_WGS84=data.Lengdegrad_WGS84(Gyldige);
AntallSatellitter_gyldige=data.AntallSatellitter(Gyldige);
HDOP_gyldige=data.HDOP(Gyldige);

Gyldige_RiktigAntall_Sat=find(AntallSatellitter_gyldige>=data.Antall_Sate
llitter_Tekst);

Breddegrad_RiktigAntall_Sat=Breddegrad_gyldige_UTM(Gyldige_RiktigAntall_S
at);

Lengdegrad_RiktigAntall_Sat=Lengdegrad_gyldige_UTM(Gyldige_RiktigAntall_S
at);

Breddegrad_RiktigAntall_Sat_WGS84=Breddegrad_gyldige_WGS84(Gyldige_Riktig
Antall_Sat);

Lengdegrad_RiktigAntall_Sat_WGS84=Lengdegrad_gyldige_WGS84(Gyldige_Riktig
Antall_Sat);
HDOP_RiktigAntall_Sat=HDOP_gyldige(Gyldige_RiktigAntall_Sat);

```

```

HDOP_RiktigAntall=find(HDOP_RiktigAntall_Sat<data.HDOP_tekst);

Breddegrad_UTM=Breddegrad_RiktigAntall_Sat(HDOP_RiktigAntall);
Lengdegrad_UTM=Lengdegrad_RiktigAntall_Sat(HDOP_RiktigAntall);

Breddegrad_WGS84=Breddegrad_RiktigAntall_Sat_WGS84(HDOP_RiktigAntall);
Lengdegrad_WGS84=Lengdegrad_RiktigAntall_Sat_WGS84(HDOP_RiktigAntall);
end

AntallGyldigeMalinger=floor(length(Breddegrad_UTM)/data.Rader*100);
set(handles.Antall_Gyldige_Malinger, 'String', strcat('Antall
malinger som gjenstår:',int2str(AntallGyldigeMalinger),' %'));

[Breddegrad_mean,Lengdegrad_mean] =
meanm(Breddegrad_WGS84,Lengdegrad_WGS84,'degrees');
[Lengdegrad_mean,Breddegrad_mean] =
mfwdtran(data,Breddegrad_mean,Lengdegrad_mean);

%%std
dist = stdist(Breddegrad_WGS84,Lengdegrad_WGS84,'degrees'); %beregner
standardavvik
dist_m = deg2km(dist)*1000;

[latc,longc] = scirclel(Lengdegrad_mean-utm_y_fast,Breddegrad_mean-
utm_x_fast,dist_m);
[latc_2std,longc_2std] = scirclel(Lengdegrad_mean-
utm_y_fast,Breddegrad_mean-utm_x_fast,2*dist_m);
[latc_3std,longc_3std] = scirclel(Lengdegrad_mean-
utm_y_fast,Breddegrad_mean-utm_x_fast,3*dist_m);

figure
Posisjon_handles=plot(Lengdegrad_UTM-utm_y_fast,Breddegrad_UTM-
utm_x_fast,'color','b'); %plotter posisjonene
hold on
Standardavvik_handles=plot(latc,longc,'k','LineWidth',1.5);
Gjennomsnitt_handles=plot(Lengdegrad_mean-utm_y_fast,Breddegrad_mean-
utm_x_fast,'gx','MarkerSize',12,'LineWidth',1.5); %Skriver ut
gjennomsnittsposisjon

Fastpunkt_handles=plot(0,0,'r+','MarkerSize',12,'LineWidth',1.5);
%Skriver ut midtpunkt

legend([Posisjon_handles;Standardavvik_handles;Gjennomsnitt_handles;Fastp
unkt_handles],'Posisjon','Standardavvik','Gjennomsnittsposisjon','Fastpun
kt',-1);
ylabel('Feil i nordlig retning [m]')
xlabel('Feil i østlig retning [m]')

% Funksjonen lager og skriver ut et histogram over hvor mange satellitter
mottakeren har hatt kontakt med i løpet av måleperioden
function antallSatellitt
data=getappdata(gcf,'metricdata');
satellitt=0:1:12; %Alternativ 1 til 12 satellitter
antallSatellittHistogram=histc(data.AntallSatellitter,satellitt);
%Beregner histogram over antall satellitter
antallSatellittHistogram=antallSatellittHistogram/data.Rader*100;
%Resultatet i prosent

figure

```

```

    antallsat_handles=bar(satellitt,antallSatellittHistogram); %Skriver
ut histogram
    axis([-1 13 0 100])
    ylabel('Antall satellitter [%]')
    xlabel('Antall satellitter')

% Funksjonen lager og viser et histogram over antallet ugyldige målinger
i måleserien
function antallGyldigeMalinger
    data=getappdata(gcbf,'metricdata');
    gyldighet=0:1:1; %Gyldig=1, Ugyldig=0
    antallGyldigeMalingerHistogram=histc(data.Gyldig,gyldighet);

antallGyldigeMalingerHistogram=antallGyldigeMalingerHistogram/data.Rader*
100; %Resultat i prosent

    figure
    bar(gyldighet,antallGyldigeMalingerHistogram)
    axis([-0.5 1.5 0 109])

text(0,antallGyldigeMalingerHistogram(1,1)+3,strcat(num2str(round(antallG
yldigeMalingerHistogram(1,1))),',
%'),'FontSize',18,'Color','black','HorizontalAlignment','center')

text(1,antallGyldigeMalingerHistogram(2,1)+3,strcat(num2str(round(antallG
yldigeMalingerHistogram(2,1))),',
%'),'FontSize',18,'Color','black','HorizontalAlignment','center')
    ylabel('Gyldige målinger [%]')
    xlabel('0=Ugyldig 1=Gyldig')

function PrTime
    data=getappdata(gcbf,'metricdata');

    AvrundetTid(:,1)=floor(data.Tid/10000); %Posisjon(:,1) er tidspunkt.
Regner om til hele time
    time=AvrundetTid(1,1); %leser første tidspunkt
(time)
    gyldig=0:1:1; %0=ugyldig 1=Gyldig
    forstePosTime=1;
    timeNr=1; %timeNr

    for teller=1:1:data.Rader
        if time~=AvrundetTid(teller,1)
%En time lest
            histData(1:2,timeNr)=histc(data.Gyldig(forstePosTime:teller-
1),gyldig); %lager hist-data for denne timen
            timeNr=timeNr+1;
            forstePosTime=teller;
%første posisjon i ny time
            time=AvrundetTid(teller,1);
            elseif teller==data.Rader %siste melding er lest

histData(1:2,timeNr)=histc(data.Gyldig(forstePosTime:teller),gyldig);
%lager hist-data for siste timen
            end
            end

            histData=histData'; %invers
            gyldigePerTime=sum((histData),2); %Antall gyldige
målinger per time

```

```

histData(:,1)=histData(:,1)./gyldigePerTime;      %Histdata i %
histData(:,2)=histData(:,2)./gyldigePerTime;      %Histdata i %
histData(:,1)=histData(:,1).*100;                 %Resultat i prosent
histData(:,2)=histData(:,2).*100;                 %Resultat i prosent

figure
bar(histData,'stack'), colormap(winter);           %plotter
ylim([0,109]);
ylabel('Antall målinger')
xlabel('Antall timer etter loggstart')

% --- Executes during object creation, after setting all properties.
function lengsteAvstand_CreateFcn(hObject, eventdata, handles)
% hObject    handle to lengsteAvstand (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function lengsteAvstand_Callback(hObject, eventdata, handles)
% hObject    handle to lengsteAvstand (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');
    lengste = str2double(get(hObject, 'String'));
    data = getappdata(gcf, 'metricdata');
    data.Lengste = lengste;
    setappdata(gcf, 'metricdata', data);
% Hints: get(hObject,'String') returns contents of lengsteAvstand as text
%         str2double(get(hObject,'String')) returns contents of
lengsteAvstand as a double

% --- Executes during object creation, after setting all properties.
function storrelse_CreateFcn(hObject, eventdata, handles)
% hObject    handle to storrelse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function storrelse_Callback(hObject, eventdata, handles)
% hObject    handle to storrelse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

    %data = getappdata(gcbf, 'metricdata');
    storrelse = str2double(get(hObject, 'String'));
    data = getappdata(gcbf, 'metricdata');
    data.Storrelse = storrelse;
    setappdata(gcbf, 'metricdata', data);
% Hints: get(hObject, 'String') returns contents of storrelse as text
%       str2double(get(hObject, 'String')) returns contents of storrelse
as a double

% Funksjonen lager og viser et histogram over avstandene mellom de
% beregnede posisjonene og fastpunktet
function spredning(handles)
    data = getappdata(gcbf, 'metricdata');

    utm_x=data.Breddegrad_fast;    %eksaket punkt på tak HIA nordlig
    rettning
    utm_y=data.Lengdegrad_fast;    %eksaket punkt på tak HIA østlig
    rettning

    if data.GyldigeMalinger_CHK_Status==0 &&
data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==0
        Gyldige=': ';
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==1
        Gyldige=find(data.HDOP<data.HDOP_tekst);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==0
        Gyldige=find(data.Gyldig);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==0 && data.HDOP_CHK_Status==1
        Gyldige=find(data.Gyldig);

        Breddegrad_gyldige=data.Breddegrad(Gyldige);
        Lengdegrad_gyldige=data.Lengdegrad(Gyldige);
        HDOP_gyldige=data.HDOP(Gyldige);

        HDOP_gyldig_under_antall=find(HDOP_gyldige<data.HDOP_tekst);

        Breddegrad_UTM=Breddegrad_gyldige(HDOP_gyldig_under_antall);
        Lengdegrad_UTM=Lengdegrad_gyldige(HDOP_gyldig_under_antall);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==0

Gyldige=find(data.AntallSatellitter>=data.Antall_Satellitter_Tekst);
        Breddegrad_UTM=data.Breddegrad(Gyldige);
        Lengdegrad_UTM=data.Lengdegrad(Gyldige);
    elseif data.GyldigeMalinger_CHK_Status==0 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==1

Gyldige=find(data.AntallSatellitter>=data.Antall_Satellitter_Tekst);

        Breddegrad_gyldige=data.Breddegrad(Gyldige);
        Lengdegrad_gyldige=data.Lengdegrad(Gyldige);
        HDOP_gyldige=data.HDOP(Gyldige);

```



```

HDOP_gyldige_Antall=find(HDOP_gyldige<data.HDOP_tekst);

Breddegrad_UTM=Breddegrad_gyldige(HDOP_gyldige_Antall);
Lengdegrad_UTM=Lengdegrad_gyldige(HDOP_gyldige_Antall);
elseif data.GyldigeMalinge_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==0
Gyldige=find(data.Gyldig);

Breddegrad_gyldige_UTM=data.Breddegrad(Gyldige);
Lengdegrad_gyldige_UTM=data.Lengdegrad(Gyldige);
AntallSatellitter_gyldige=data.AntallSatellitter(Gyldige);

Gyldige_RiktigAntall_Sat=find(AntallSatellitter_gyldige>=data.Antall_Sate
llitter_Tekst);

Breddegrad_UTM=Breddegrad_gyldige_UTM(Gyldige_RiktigAntall_Sat);
Lengdegrad_UTM=Lengdegrad_gyldige_UTM(Gyldige_RiktigAntall_Sat);
elseif data.GyldigeMalinge_CHK_Status==1 &&
data.Antall_Satellitter_CHK_Status==1 && data.HDOP_CHK_Status==1
Gyldige=find(data.Gyldig);

Breddegrad_gyldige_UTM=data.Breddegrad(Gyldige);
Lengdegrad_gyldige_UTM=data.Lengdegrad(Gyldige);
AntallSatellitter_gyldige=data.AntallSatellitter(Gyldige);
HDOP_gyldige=data.HDOP(Gyldige);

Gyldige_RiktigAntall_Sat=find(AntallSatellitter_gyldige>=data.Antall_Sate
llitter_Tekst);

Breddegrad_RiktigAntall_Sat=Breddegrad_gyldige_UTM(Gyldige_RiktigAntall_S
at);

Lengdegrad_RiktigAntall_Sat=Lengdegrad_gyldige_UTM(Gyldige_RiktigAntall_S
at);

HDOP_RiktigAntall_Sat=HDOP_gyldige(Gyldige_RiktigAntall_Sat);

HDOP_RiktigAntall=find(HDOP_RiktigAntall_Sat<data.HDOP_tekst);

Breddegrad_UTM=Breddegrad_RiktigAntall_Sat(HDOP_RiktigAntall);
Lengdegrad_UTM=Lengdegrad_RiktigAntall_Sat(HDOP_RiktigAntall);
end

AntallGyldigeMalinge=length(Breddegrad_UTM)/data.Rader*100;
set(handles.Antall_Gyldige_Malinge, 'String', strcat('Antall
malinge som gjenstår:',int2str(AntallGyldigeMalinge),' %'));

feil_bredde=Breddegrad_UTM-utm_x;
feil_lengde=Lengdegrad_UTM-utm_y;

feil=sqrt(feil_lengde.^2+feil_bredde.^2); %avstand fra nøyaktig punkt
vektorFeil=data.Storrelse/2:data.Storrelse:data.Lengste; %vektor til
x-aksen i histogram

histFeil=hist(feil,vektorFeil)/length(feil); %lage histogram

figure
bar(vektorFeil,histFeil*100) %tegne hisogram

```

```

ylabel('Antall Målinger [%]')
xlabel('Avstand [m]')

% Funksjonen lager og viser et histogram over antall målinger pr time i
løpet av måleperioden
function PrTime24%(Tid,GyldigMaling,antallMaling)
    data=getappdata(gcf,'metricdata');
    Tid=floor(data.Tid/10000); %Posisjon(:,1) er tidspunkt. Regner om
til hele time

    gyldig=0:1; %0=ugyldig 1=Gyldig
    forstePosTime=1; %første posisjon i ny time
    histData(1,:)=zeros(1,24); %initialiseringsverdi for histData
    histData(2,:)=zeros(1,24); %initialiseringsverdi for histData

    if Tid(1,1)~=0 %leser første tidspunkt (time)
        klokke=Tid(1,1); %klokke
    else %time 0 = 24
        klokke=24;
    end

    for teller=1:1:data.Rader
        if Tid(forstePosTime,1)~=Tid(teller,1) %En time lest

histData(1:2,klokke)=histData(1:2,klokke)+histc(data.Gyldig(forstePosTime
:teller-1),gyldig);%lager hist-data for denne timen
            forstePosTime=teller; %første posisjon i ny time

            if Tid(teller,1)~=0 %leser første tidspunkt (time)
                klokke=Tid(teller,1);
            else %time 0 = 24
                klokke=24;
            end

            elseif teller==data.Rader %siste melding er lest

histData(1:2,klokke)=histData(1:2,klokke)+histc(data.Gyldig(forstePosTime
:teller),gyldig);%lager hist-data for siste timen
        end
    end

    histData=histData'; %invers
    gyldigePerTime=sum((histData),2); %Antall gyldige målinger per time

    histData(:,1)=histData(:,1)./gyldigePerTime; %Histdata i %
    histData(:,2)=histData(:,2)./gyldigePerTime; %Histdata i %

    figure
    bar(histData,'stack'), colormap(winter) %plotter
    ylabel('Antall målinger')
    xlabel('Time i døgnet')

% --- Kjøres når Kitons Data knappen trykkes.
function kitron_data_knapp_Callback(hObject, eventdata, handles)
% hObject handle to kitron_data_knapp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
    openfig('D:\HOVEDPROSJEKT\Matlab\Lese
Excelfil\Kitons_data.fig','new')

```

```

% --- Executes on button press in Gylidge_Malinger_Chk.
function Gylidge_Malinger_Chk_Callback(hObject, eventdata, handles)
% hObject      handle to Gylidge_Malinger_Chk (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');
    data.GyldigeMalinger_CHK_Status=get(hObject,'Value');
    setappdata(gcf, 'metricdata', data);
% Hint: get(hObject,'Value') returns toggle state of Gylidge_Malinger_Chk

% --- Executes on button press in Antall_Satellitter_Chk.
function Antall_Satellitter_Chk_Callback(hObject, eventdata, handles)
% hObject      handle to Antall_Satellitter_Chk (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');
    data.Antall_Satellitter_CHK_Status=get(hObject,'Value');
    setappdata(gcf, 'metricdata', data);
% Hint: get(hObject,'Value') returns toggle state of
Antall_Satellitter_Chk

% --- Executes during object creation, after setting all properties.
function Antall_Satellitter_Tekst_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Antall_Satellitter_Tekst (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
%set(hObject, 'String', menyString);

function Antall_Satellitter_Tekst_Callback(hObject, eventdata, handles)
% hObject      handle to Antall_Satellitter_Tekst (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');

    popup_sel_index = get(handles.Antall_Satellitter_Tekst, 'Value');
    switch popup_sel_index
    case 1
        data.Antall_Satellitter_Tekst=1;
    case 2
        data.Antall_Satellitter_Tekst=2;
    case 3
        data.Antall_Satellitter_Tekst=3;
    case 4
        data.Antall_Satellitter_Tekst=4;
    case 5
        data.Antall_Satellitter_Tekst=5;
    case 6
        data.Antall_Satellitter_Tekst=6;
    case 7
        data.Antall_Satellitter_Tekst=7;
    case 8

```

```

        data.Antall_Satellitter_Tekst=8;
    case 9
        data.Antall_Satellitter_Tekst=9;
    case 10
        data.Antall_Satellitter_Tekst=10;
    case 11
        data.Antall_Satellitter_Tekst=11;
    case 12
        data.Antall_Satellitter_Tekst=12;
    end
    setappdata(gcf, 'metricdata', data);
% Hints: get(hObject,'String') returns contents of
Antall_Satellitter_Tekst as text
%       str2double(get(hObject,'String')) returns contents of
Antall_Satellitter_Tekst as a double

% --- Executes during object creation, after setting all properties.
function HDOP_tekst_CreateFcn(hObject, eventdata, handles)
% hObject    handle to HDOP_tekst (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
set(hObject, 'String', {'2','4','6','8','10','15','20'});
% Hint: popuymenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in HDOP_tekst.
function HDOP_tekst_Callback(hObject, eventdata, handles)
% hObject    handle to HDOP_tekst (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');

    popup_sel_index = get(handles.HDOP_tekst, 'Value');
    switch popup_sel_index
    case 1
        data.HDOP_tekst=2;
    case 2
        data.HDOP_tekst=4;
    case 3
        data.HDOP_tekst=6;
    case 4
        data.HDOP_tekst=8;
    case 5
        data.HDOP_tekst=10;
    case 6
        data.HDOP_tekst=15;
    case 7
        data.HDOP_tekst=20;
    end
    setappdata(gcf, 'metricdata', data);
% Hints: contents = get(hObject,'String') returns HDOP_tekst contents as
cell array
%       contents{get(hObject,'Value')} returns selected item from
HDOP_tekst

```

```

% --- Executes on button press in HDOP_Chk.
function HDOP_Chk_Callback(hObject, eventdata, handles)
% hObject      handle to HDOP_Chk (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
    data = getappdata(gcf, 'metricdata');
    data.HDOP_CHK_Status=get(hObject, 'Value');
    setappdata(gcf, 'metricdata', data);
% Hint: get(hObject, 'Value') returns toggle state of HDOP_Chk

function Middling(handles);
    data = getappdata(gcf, 'metricdata');

    utm_x=data.Breddegrad_fast;           %eksaket punkt på tak
HIA nordlig rettning
    utm_y=data.Lengdegrad_fast;         %eksaket punkt på tak
HIA østlig rettning

    minimum_antall_gyldig=data.Storrelse;
    antall_per_middel=data.Lengste-1;
    j=1;

    for i=1:data.Rader-antall_per_middel
        antall_gyldig=sum(data.Gyldig(i:i+antall_per_middel));

        if(antall_gyldig>=minimum_antall_gyldig)
            gyldig=find(data.Gyldig(i:i+antall_per_middel))+i-1;

gyldigLengdegrad_UTM(j,1)=sum(data.Lengdegrad(gyldig))/antall_gyldig;
gyldigBreddegrad_UTM(j,1)=sum(data.Breddegrad(gyldig))/antall_gyldig;
            j=j+1;
        end

    end

    x=-10:0.1:10;

    Breddegrad_mean=mean(gyldigBreddegrad_UTM);
    Lengdegrad_mean=mean(gyldigLengdegrad_UTM);

    figure
    Posisjon_handles=plot(gyldigLengdegrad_UTM-
data.Lengdegrad_fast,gyldigBreddegrad_UTM-
data.Breddegrad_fast,'color','b');
    hold on
    Fastpunkt_handles=plot(0,0,'r+', 'MarkerSize',12, 'LineWidth',2);
%Skriver ut midtpunkt
    Gjennomsnittspunkt_handles=plot(Lengdegrad_mean-
data.Lengdegrad_fast,Breddegrad_mean-
data.Breddegrad_fast,'g+', 'MarkerSize',12, 'LineWidth',2);

legend([Posisjon_handles;Fastpunkt_handles;Gjennomsnittspunkt_handles], 'P
osisjon', 'Fastpunkt', 'Gjennomsnittspunkt', -1);
    ylabel('Feil i nordlig retning [m]')
    xlabel('Feil i østlig retning [m]')

    feil_bredde=gyldigBreddegrad_UTM-utm_x;
    feil_lengde=gyldigLengdegrad_UTM-utm_y;

    feil=sqrt(feil_lengde.^2+feil_bredde.^2); %avstand fra nøyaktig punkt

```

```

    set(handles.Antall_Gyldige_Malinger, 'String', strcat('Antall
målinger som
gjensstår:',int2str(length(gyldigBreddegrad_UTM)/data.Rader*100),'%'));

vektorFeil=5/2:5:50;      %vektor til x-aksen i histogram

histFeil=hist(feil,vektorFeil)/length(feil); %lage histogram

figure
bar(vektorFeil,histFeil*100)      %tegne hisogram
ylabel('Antall Målinger [%]')
xlabel('Avstand [m]')

% --- Executes during object creation, after setting all properties.
function Antall_Gyldige_Malinger_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Antall_Gyldige_Malinger (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
%if ispc
%   set(hObject,'BackgroundColor','white');
%else
%
set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
%end

function Antall_Gyldige_Malinger_Callback(hObject, eventdata, handles)
% hObject    handle to Antall_Gyldige_Malinger (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
Antall_Gyldige_Malinger as text
%       str2double(get(hObject,'String')) returns contents of
Antall_Gyldige_Malinger as a double

% Funksjonen viser de posisjonen etter at sirkelmiddlingsalgoritmen er
brukt
function SirkelMiddling(handles)
    data = getappdata(gcf, 'metricdata');

    utm_x=data.Breddegrad_fast; %eksaket punkt på tak HIA nordlig
rettning
    utm_y=data.Lengdegrad_fast; %eksaket punkt på tak HIA østlig
rettning

    minimum_antall_gyldig=data.Storrelse;
    antall_per_middel=data.Lengste-1;
    j=1;
    radius=data.tekstFelt3;
    antall_sirkler=data.tekstFelt4;

    for i=1:data.Rader-antall_per_middel-1
        gyldig=find(data.Gyldig(i:i+antall_per_middel))+i-1;
        antall_gyldig=length(gyldig);

```

```

if(antall_gyldig>=minimum_antall_gyldig)
    snitt_lengde=sum(data.Lengdegrad(gyldig))/antall_gyldig;
    snitt_bredde=sum(data.Breddegrad(gyldig))/antall_gyldig;

    for antall_radius=antall_sirkler:-1:1
        R=radius*antall_radius;

        avstand=sqrt((snitt_lengde-
data.Lengdegrad(gyldig)).^2+(snitt_bredde-data.Breddegrad(gyldig)).^2);
        Gyldig=gyldig(find(avstand<=R));

        antall_gyldig=length(Gyldig);

        if antall_radius==1 &&
antall_gyldig>=minimum_antall_gyldig

gyldigLengdegrad_UTM(j,1)=sum(data.Lengdegrad(Gyldig))/antall_gyldig;

gyldigBreddegrad_UTM(j,1)=sum(data.Breddegrad(Gyldig))/antall_gyldig;
        j=j+1;
        end
    end
end
end

Breddegrad_mean=mean(gyldigBreddegrad_UTM);
Lengdegrad_mean=mean(gyldigLengdegrad_UTM);

figure
Posisjon_handles=plot(gyldigLengdegrad_UTM-
data.Lengdegrad_fast,gyldigBreddegrad_UTM-data.Breddegrad_fast);
hold on
Fastpunkt_handles=plot(0,0,'r+','MarkerSize',12,'LineWidth',2);
%Skriver ut midtpunkt
Gjennomsnittspunkt_handles=plot(Lengdegrad_mean-
data.Lengdegrad_fast,Breddegrad_mean-
data.Breddegrad_fast,'gx','MarkerSize',12,'LineWidth',2);

legend([Posisjon_handles;Fastpunkt_handles;Gjennomsnittspunkt_handles],'P
osisjon','Fastpunkt','Gjennomsnittspunkt',-1);

ylabel('Feil i nordlig retning [m]')
xlabel('Feil i østlig retning [m]')
[Rader_radius,columns] = size(gyldigLengdegrad_UTM);
AntallGyldigeMalinger=Rader_radius/data.Rader*100;
set(handles.Antall_Gyldige_Malinger,'String',strcat('Antall
målinger som gjenstår:',int2str(AntallGyldigeMalinger),'%'));

feil_bredde=gyldigBreddegrad_UTM-utm_x;
feil_lengde=gyldigLengdegrad_UTM-utm_y;

feil=sqrt(feil_lengde.^2+feil_bredde.^2); %avstand fra nøyaktig punkt
vektorFeil=5/2:5:50; %vektor til x-aksen i histogram

histFeil=hist(feil,vektorFeil)/length(feil); %lage histogram

figure
bar(vektorFeil,histFeil*100) %tegne hisogram
ylabel('Antall Målinger [%]')
xlabel('Avstand [m]')

```

```

% --- Executes during object creation, after setting all properties.
function tekstFelt3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tekstFelt3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function tekstFelt3_Callback(hObject, eventdata, handles)
% hObject    handle to tekstFelt3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of tekstFelt3 as text
% str2double(get(hObject,'String')) returns contents of tekstFelt3 as a
double
    tekstFelt3 = str2double(get(hObject, 'String'));
    data = getappdata(gcf, 'metricdata');
    data.tekstFelt3 = tekstFelt3;
    setappdata(gcf, 'metricdata', data);

% --- Executes during object creation, after setting all properties.
function tekstFelt4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tekstFelt4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function tekstFelt4_Callback(hObject, eventdata, handles)
% hObject    handle to tekstFelt4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    tekstFelt4 = str2double(get(hObject, 'String'));
    data = getappdata(gcf, 'metricdata');
    data.tekstFelt4 = tekstFelt4;
    setappdata(gcf, 'metricdata', data);
% Hints: get(hObject,'String') returns contents of tekstFelt4 as text
%       str2double(get(hObject,'String')) returns contents of tekstFelt4
as a double

%Funksjonen viser posisjonsmålinger etter at SnittMedHDOP algoritmen er
%brukt
function SnittMedHDOP(handles)

```



```

data = getappdata(gcbf, 'metricdata');
utm_x=data.Breddegrad_fast; %eksaket punkt på tak HIA nordlig
rettning
utm_y=data.Lengdegrad_fast; %eksaket punkt på tak HIA østlig rettning

minimum_antall_gyldig_HDOP=data.Storrelse;
antall_per_middel=data.Lengste-1;
j=1;
HDOP_bedre_enn=data.tekstFelt3;

for i=1:data.Rader-antall_per_middel
    gyldig=find(data.HDOP(i:i+antall_per_middel)<=HDOP_bedre_enn)+i-
1;
    antall_gyldig_HDOP = length(gyldig);

    if antall_gyldig_HDOP>=minimum_antall_gyldig_HDOP

gyldigLengdegrad_UTM(j,1)=sum(data.Lengdegrad(gyldig))/antall_gyldig_HDOP
;

gyldigBreddegrad_UTM(j,1)=sum(data.Breddegrad(gyldig))/antall_gyldig_HDOP
;
        j=j+1;
    end
end

Breddegrad_mean=mean(gyldigBreddegrad_UTM);
Lengdegrad_mean=mean(gyldigLengdegrad_UTM);

figure
Posisjon_h=plot(gyldigLengdegrad_UTM-
data.Lengdegrad_fast,gyldigBreddegrad_UTM-data.Breddegrad_fast);
hold on
Fastpunkt_h=plot(0,0,'r+', 'MarkerSize',12, 'LineWidth',2);
%Skriver ut midtpunkt
Gjennomsnittspunkt_h=plot(Lengdegrad_mean-
data.Lengdegrad_fast,Breddegrad_mean-
data.Breddegrad_fast, 'gx', 'MarkerSize',12, 'LineWidth',2);

legend([Posisjon_h;Fastpunkt_h;Gjennomsnittspunkt_h], 'Posisjon', 'Fastpunk
t', 'Gjennomsnittspunkt',-1);

xlabel('Feil i østlig retning [m]');
ylabel('Feil i nordlig retning [m]');
[Rader_HDOP,columns] = size(gyldigLengdegrad_UTM);
set(handles.Antall_Gyldige_Malinger, 'String', strcat('Antall
målinger som gjenstår:',int2str(Rader_HDOP/data.Rader*100),'%'));

feil_bredde=gyldigBreddegrad_UTM-utm_x;
feil_lengde=gyldigLengdegrad_UTM-utm_y;

feil=sqrt(feil_lengde.^2+feil_bredde.^2); %avstand fra nøyaktig punkt
vektorFeil=5/2:5:50; %vektor til x-aksen i histogram

histFeil=hist(feil,vektorFeil)/length(feil); %lage histogram

figure
bar(vektorFeil,histFeil*100) %tegne hisogram
ylabel('Antall Målinger [%]')
xlabel('Avstand [m]')

```