



# Statens vegvesen

ITS TEKNOLOGI

VEGDIREKTØR → TRANSPORT OG SAMFUNN →  
TRANSPORTUTVIKLING

---

## Estimering av GDOP langs norske veier

---

Av :  
Sindre Havn, Johanne Karbø

Juli, 2024

---

# 1 Sammendrag

Vårt sammendrag er.....

## 2 Abstrakt

Norge ligger som kjent langt oppe på Globusen og GNSS satellittene går i baner som er langt mer sydlig. I praksis fører dette til at satellitter blir lavt over horisonten og kan da ofte bli skjult av fjell. Dersom man skal kjøre ned Romsdalen E136 så kjører man nede i en smal dal med vært høye fjell på hver side. Dersom vi i fremtiden skal bruke kjøretøy som krever GNSS for styring eller for maskinstyring så må vi ha en oversikt over hvor mange satellitter man kan se da dette vil påvirke hvor god posisjonsberegningene vil bli. Et mål på hvor god posisjonsbestemmelsen kan bli, kalles for GDOP (Geometric Dilution Of Position).

## 3 Innledning

Det skal utledes en metode for å estimere GDOP langs norske veier, basert på kartdata fra kartverket, og offentlig baneinformasjon fra satellitt systemene GPS, Galileo, og Glonass.

Problemstillingen er tredelt. Første del er å regne ut horisontkurven fra observasjonspunkt langs veien. Den andre delen er å finne ut hvilke satellitter som er synlige fra disse punktene. Tredje del er å regne ut GDOP basert på relativ posisjon til de synlige satellittene.

Dette dokumentet er ikke ment som en definitiv løsning, men heller for å gi innsikt i gjeldende problemstilling som strekker seg utenfor vegvesenets kompetanseområde. Det er derfor brukt noe enklere språk. Engelske fagbegrep er benyttet da det er tenkt å gjøre videre læring på internett mindre forvirrende.

## Table of Contents

<b>1</b>	<b>Sammendrag</b>	<b>i</b>
<b>2</b>	<b>Abstrakt</b>	<b>i</b>
<b>3</b>	<b>Innledning</b>	<b>i</b>
<b>4</b>	<b>GNSS og posisjonering</b>	<b>1</b>
4.1	Detaljert funksjonalitet . . . . .	2
<b>5</b>	<b>GDOP</b>	<b>8</b>
<b>6</b>	<b>Beregning av horisontkurve</b>	<b>9</b>
<b>7</b>	<b>GNSS og satellitter</b>	<b>9</b>
7.1	Baner til satellitter . . . . .	9
7.1.1	Keplerbane . . . . .	9
7.2	Satelitt data, RINEX . . . . .	11

---

7.3	Mask angle . . . . .	11
7.3.1	Mask angle for optimalisering . . . . .	13
7.4	Database for horisontvinkler og GNSS posisjon . . . . .	13
7.4.1	Presisjon . . . . .	13
<b>8</b>	<b>ArcGIS</b>	<b>15</b>
<b>9</b>	<b>Resultater</b>	<b>18</b>
<b>10</b>	<b>Diskusjon</b>	<b>18</b>
<b>11</b>	<b>Konklusjon</b>	<b>18</b>
	<b>Appendix</b>	<b>21</b>
A	Hello World Example . . . . .	21

---

## 4 GNSS og posisjonering

GNSS (Global Navigation Satellite system) er en samlebetegnelse for posisjoneringssystem som er åpen for allmenn bruk. Når man prater om satellitt-posisjonering i dagligtalen snakker man ofte om GPS, men det er kun USA's GNSS. Andre globalt dekkende GNSS'er er Galileo (EU), Glonass (Russland), BeiDou (Kina). QZSS (Japan) og IRNSS (India) er også GNSS'er, men disse er regionale satellittsystem som ikke gir dekning til Norge, og blir derfor ikke relevant for denne oppgaven. GNSS gjør at man kan beregne posisjonen til typisk  $\pm 5$  meter under åpen himmel [4], men snart ser vi på presisjon godt under meteren [16].

I sin enkleste form, kan man si at GNSS finner posisjon ved *trilaterering* (må ikke forveksles med triangulering) som vist i figur 1.

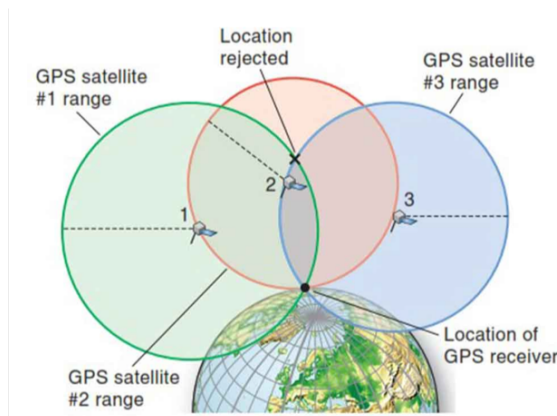


Figure 1: Illustrasjon av trilaterering [17]. Hver satellitt er tegnet med egen posisjon og et kuleskall med radius lik avstand til mottakeren. GNSS satellitter beveger seg 4 km per sekund, omtrent 20 000 km over jorden.

Kjenner man posisjon og avstand til én satellitt, vet man at posisjonen må være et sted på et tenk kuleskall rundt satellitten. Kjenner man posisjon og avstand til to, vet man at man befinner seg langs en sirkulær kurven der kuleskallene skjærer hverandre. Har man tre satellitter får man to skjæringspunkt, men fjerner det som er høyt over jordoverflaten. Skjæringspunktet du står igjen med er mottakerens posisjon. I praksis trenger man minst fire satellitter for å beregne posisjon, men mer om det i seksjon 4.1.

En GNSS mottaker trenger som sagt å kjenne både posisjonen, og avstanden til satellitten for å kunne estimere sin egen posisjon. Vi fokuserer på satellitters posisjon først.

GNSS satellitter sender ut detaljer om egen posisjon hele tiden. Problemet er at satellittene kommuniserer i overlappende frekvensbånd [13], man kan si de "høres like ut". Har du dekning til flere satellitter, og alle snakker på en gang, blir det som at de snakker i munnen på hverandre, og du har ikke sjans å vite hvem som sier hva. Derfor har hver GNSS satellitt sin egen "dialekt" som blir lagt på signalet. Dermed klarer du å skille mellom hvem som sier hva, og hvor hver enkel satellitt befinner seg over jorden. Alle GNSS mottakere har en innebygd oversikt over disse "dialektene".

Den neste problemstillingen er å måle avstand  $R$  til hver synlige satellitt. Det enkle svaret er at du måler tiden  $dt$  signalet bruker fra satellitten til mottakeren og multipliserer med lysfarten  $c$  (signal fra satellitter er bare lys, man ikke kan se med øyet).

$$R = c \times dt \tag{1}$$

## 4.1 Detaljert funksjonalitet

Nå skal vi bygge på seksjon 4, gi en grundigere forklaring av GNSS, og forklare hvordan ulike utfordringer løses. Dette er nødvendig for å bestemme presisjon og begrensninger til vegvesenet's tiltenkte system.

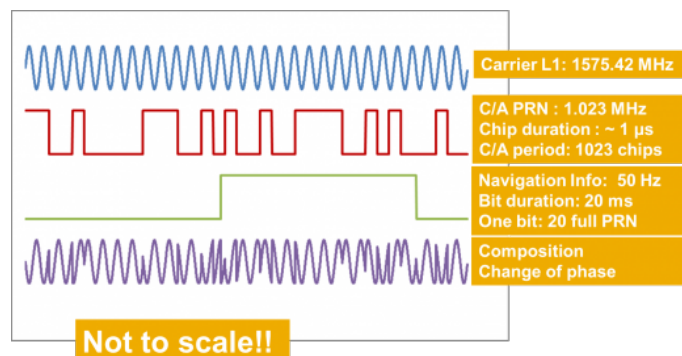


Figure 2: Signalstruktur til GNSS. Her er tallverdiene knyttet til L1 signal fra GPS satellitter. Signalene (lilla) er en sammensetning av en grunnfrekvens (blå), grov kurs kode C/A (rød) og navigasjons data (grønn) [13].

Figur 2 viser sammensetningen til et GNSS signal. Alle GNSS mottakere lytter i bestemte frekvensområder, sentrert rundt bæreølger. Gode mottakere tar imot signal på flere grunnfrekvenser, som gir enda mer presis og robust posisjonering. En av disse bæreølgerne er L1 som er ved 1575.42 MHz. Videre legges det på Coarse Acquisition (C/A) som er en pseudo-tilfeldeige støykode (PRNC). Dette er "dialekten" til hver enkelt satellitt, som gjør at man kan skille dem fra hverandre, og blir i tillegg brukt til å beregne avstand til satellitten. Til slutt har vi digital navigasjons data angir bandedate om seg selv og de andre satellittene. Det er verdt å legge merke til at data-raten på navigasjonsdataen er veldig lav, bare 50 bit per sekund (50 Hz). Det er nødvendig å sende den digitale dataen over lengre tid for å være sikker på at den blir riktig mottatt. Signalene fra GNSS satellittene er veldig svake når de treffer mottakere på jorda, og er utsatt for støy. Frekvensbåndene brukt av GNSS er derfor sårbare for annen aktivitet og tilegnet kun for GNSS kommunikasjon. Nasjonale GNSS satellitter har ofte ekstra egenskaper for nasjonens egen-interesse. Eksempelvis inneholder GPS signalet også en militær P-kode, men slike ikke-sivile funksjonaliteter i GNSS er ikke relevant for gjeldende problemstilling.

For å finne avstanden til satellittene, bruker man Coarse Acquisition (C/A) delen av signalet. Den er en pseudo-tilfeldig støykode (PRN kode) som er 1023 bit lang, og sendes på 1.023 MHz [14]. Mottakere har en kopi av PRN koden til hver satellitt. Siden mottakeren også kjenner til klokketiden til satellitten, som blir sendt gjennom navigasjonsdataen, kan mottakeren iterere gjennom PRN koden tidssynkronisert med satellitten. Sett utenfra vil vi ha to signal som går synkronisert, fra satellitten (som sender ut PRN kode i signalet) og mottaker (som iterer gjennom synkronisert kopi av PRN koden). Nedre halvdel av figur 3 viser signal fra satellitt (rød) og i mottaker (blå) som er i fase. Når mottakeren får lese PRN koden fra en satellitt, vil den se at PRN koden fra satellitten ligger noe etter sin egen kopi. Dette vises i øvre del av figur 3. Denne tidsforskyvningen mellom mottakeren sin kopi og koden fra satellitten er reisetiden til signalet som beveger seg med lyshastigheten.

GNSS mottakere har

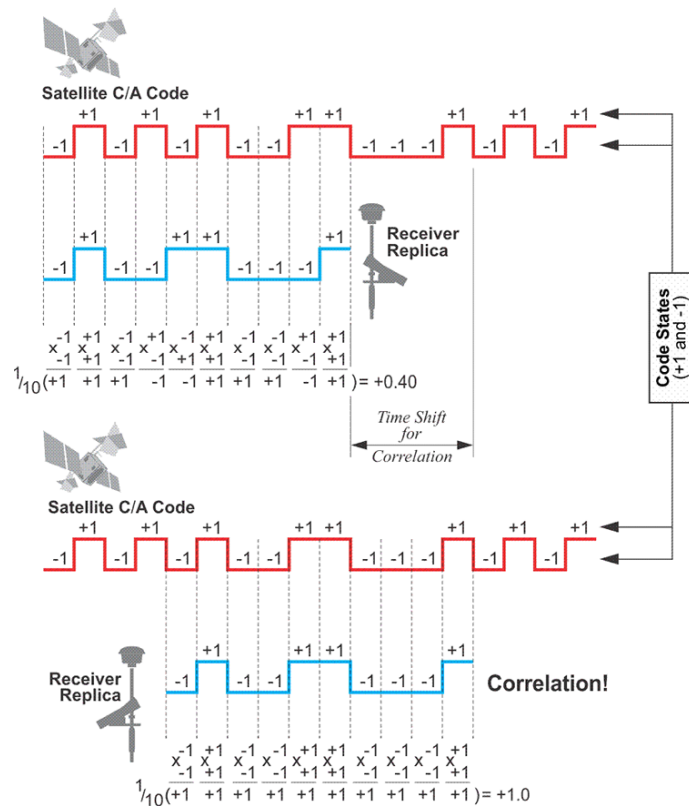


Figure 3: Illustrasjon av tidsforskyvning. Øvre halvdel viser C/A signal som ble sendt fra satellitten (rødt), og C/A signal som ble observert i mottakeren (blått). Korrelasjonen i figuren er normalisert mellom 0 og 1. Signalene er like, men gir lav korrelasjon pga tidsforskyvning. I nedre halvdel av figuren ligger signalene vedsiden hverandre. Da har vi maksimal relativ korrelasjon, og tidsforskyvningen er funnet. [2].

Formidlet litt mer matematisk brukes korrelasjon for å finne denne tidsforskyvningen mellom de to PRN kodene. Krysskorrelasjon er et mål på hvor like to signal er. Autokorrelasjon er når krysskorrelasjon brukes til å sammenligne et signal, med den tidsforskyvnet versjon av samme signal, for å finne tidsforskyvningen mellom dem. En viktig egenskap er at C/A har høy *autokorrelasjon* bare ved ett spesifikt punkt, som vist med rød kurve i figur 3. Mhp figur 3 bruker man maksimal autokorrelasjon til å finne tidsforskyvning til den periodiske PRN koden. Tiden en GNSS mottaker bruker for å finne tidsforskyvningen til de observerbare satellittene kaller vi ofte for "lock on tiden".

Et interessant poeng i figur 3 er at det binære signalet beskrives med -1 og 1, og ikke 0 og 1 som er vanlig. Det er hensiktsmessig da det gjør utregning av autokorrelasjon lettere.

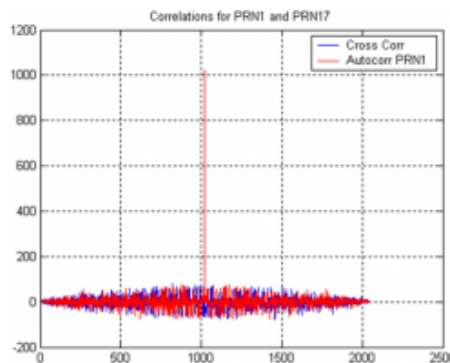


Figure 4: Eksempel på autokorrelasjon og kryss korrelasjon mellom GPS PRN1 og GPS PRN17, uten støy [6]. PRN = pseudo-random noise.

Figur 4 tar utgangspunktet i to GPS satellitter sine unike PRN koder, eller "dialekter". Den blå grafen viser krysskorrelasjonen er uniform-lav mellom de to PRN kodene, som betyr at en GNSS mottaker enkelt vil se forskjell på de. Lav krysskorrelasjon demonstrerer de robuste egenskapene til C/A signalet brukt i GNSS.

I tillegg til tidsforskyvning må mottakeren ta høyde for dopplerskift, som forvrenger signalet. Dopplerskiftet kommer av den relative farten mellom mottakeren og satellitten. GNSS satellittene beveger seg 4 km/s og har maksimal radiell fart  $v_{rad} = \pm 864 \text{ m/s}$  (bort fra jorden) [25]. Denne radielle farten vil føre til at signalet blir sammenpresset eller strukket ut avhengig om satellitten kommer nærmere, eller går fra deg. For å regne ut største mulige dopplerskift  $\Delta f$  tar vi utgangspunkt i  $f_{max} = 1575 \text{ MHz}$ , som er den høyeste grunnfrekvensen for GPS og Galileo:

$$\Delta f = \pm v_{rad} \frac{f_{max}}{c} = \pm 864 \cdot \frac{1575 \cdot 10^6 \text{ Hz}}{3 \cdot 10^8 \text{ m/s}} = \pm 4536 \text{ Hz} \quad (2)$$

For stasjonære mottakere er den største frekvensforskyvningen fra dopplerskift  $\pm 4497 \text{ Hz}$ . Klokkefeil bidrar til ytterligere 5 kHz forskyvning, som gjør at GNSS mottakere må utforske signal innenfor dopplerskift på  $\pm 10 \text{ kHz}$  [3].

Det holder altså ikke for en GNSS mottaker å finne tidsforskyvningen til PRN koden, den må også sjekke ulike frekvenser. Mhp figur 4, kan man tegne en ny korrelasjonsgraf i 3 dimensjoner som vist i figur 5, der vi har lagt inn dopplerskift som ny akse. Merk at sammenlignet med figur 4 ser vi bare på tidsforskyvninger opp til 1023, og ikke 2045 (omtrent dobbelt), dette er fordi autokorrelasjonene er symmetrisk om 0-skift, og vi antar at signalet fra satellitten alltid henger etter PRN koden til mottakeren, og aldri før. Merk at figuren ser bort fra klokkefeil, og antar dopplerskift  $\pm 5 \text{ kHz}$ .

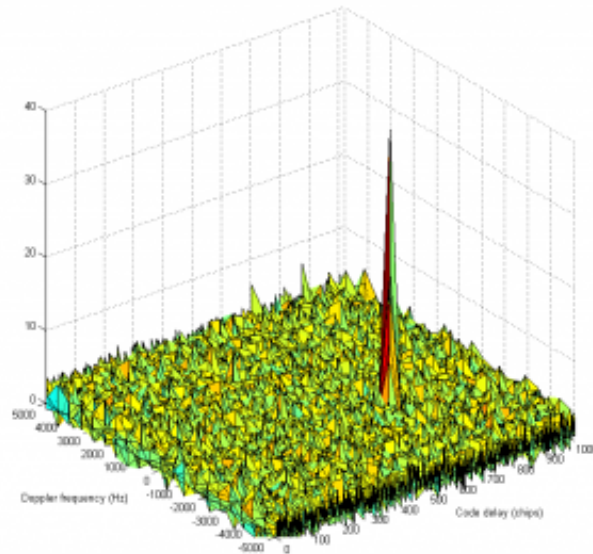


Figure 5: Korrelasjonsgraf for tidsforskyvning og dopplerskift [7].

Siden en GNSS mottaker beregner posisjon fra flere satellitter må den også finne toppunktet til autokorrelasjonen mhp tidsforskyvning, dopplerskift og ulike PRN koder fra ulike satellitter. Dette resulterer i enorme mengder beregninger som er energikrevende, som er grunnen til at batteritiden til pulsklokker kan reduseres fra 2 uker, til noen få timer.

optimaliseres med ca kjent posisjon og fart 7

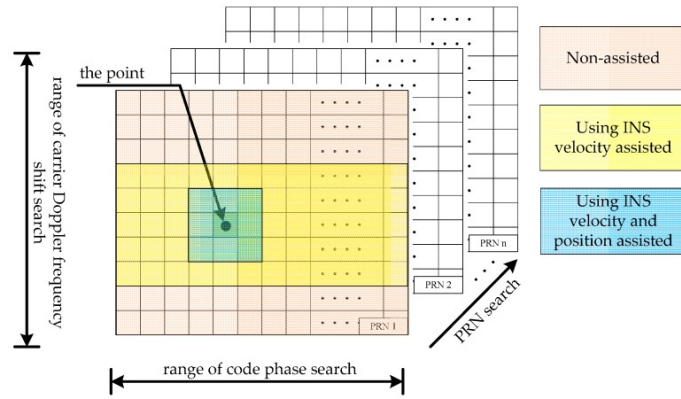


Figure 6: Korrelasjonsgraf for tidsforskyvning og dopplerskift [25].

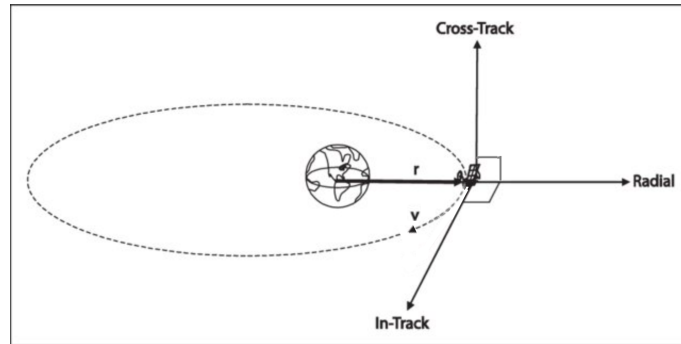


Figure 7: FIX

Så går vi videre til navigasjons-data-delen av GNSS signalet. Denne dataen kan inndeles i ephemeris, data om segsely; og almanac, data om andre satelitter.

Hver 30 sekund sender satelittenene ut data, kalt en "frame". Hver frame inneholder ephemeris data, samt korreksjons-opplysninger om blant annet ionosfæren og for klokke drift. Denne informasjonen blir brukt av mottakeren for å korrigere for noen av feilkildene vi har snakket om tidligere. Merk at klokkeeffekter fra relativitet er korrigert for i hardware [11].

Selvom mottakeren kun får oppdatert ephemeris hver 30 sekund, så klarer den å regne ut posisjon kontinuerlig. Dette er fordi mottakeren bruker tiden siden sist ephemeris data og tilhørende baneinformasjon fra satelitten, til å regne ut posisjonen til satelitten til enhver tid. Tiden siden sist mottatt ephemeris blir ofte kalt *time since reference epoch*. Når mottakeren kjenner posisjonen til satelitten er det bare å regne ut avstanden fra satelitten til mottakeren ved hjelp av autokorrelasjon som ble beskrevet tidligere i figur 3.

Hver frame vil også inneholde et grovt estimat (68% sannynlighet innenfor 1-2km avvik) på posisjonen til noen av satelittene i sitt eget satelittsystem [12]. Denne dataen om de andre satelittene kalles for "almanac" og har som formål å gi en GNSS mottaker en forventning om hvilke satelitter den kan se. Dette korter ned "lock on tiden" siden mottakeren ikke trenger å beregne krysskorrelasjon med PRN-koden til satelitter på motsatt side av kloden. Hver satelitt i en GNSS (feks galileo), har baneinformasjon på alle de andre satelittene tilhørende en sin egen GNSS (her ga-



lileo). Dataen om de andre satellittene fordeles ut over 25 slike ”frames” før de gjentas. Hver GNSS satellitt bruker altså 12.5 minutt ( $25 \times 30s$ ) på en meldingssyklus.

For å kunne bruke satellitter til å finne vår egen posisjon, er det viktig at vi kjenner satellittenes posisjon helt presis. Man skulle tro at GNSS satellittene kunne gå i forutsigbare baner, serlig siden de beveger seg gjennom vakuum. I virkeligheten er det en masse effekter som er med å påvirker banen, og hvordan den estimeres fra bakkestasjoner. Blant disse er gravitasjons perturbasjoner fra sola og alle planeter (+pluto), månen og tidevann; men også strålingstrykk, klokke drift og klokkeusikkerhet. De største støyfaktorene i posisjonering er fra dispersjon i ionosfæren, og refleksjonsstøy fra omgivelsene (multipath). Noen av effektene går det til en hvis grad ann å korrigere for, som J2 perturbasjon (pga jordas ellipsiodiske form) [9], og mesteparten av dispersjonen i ionosfæren. Mhp J2 perturbasjon blir ofte baner for GNSS satellitter oppgitt med en parameter *Rate of right ascension*  $\frac{d\Omega}{dt}$  som korrigerer for j2 perturbasjoner [10]. Dette er helt nødvendig ettersom J2 perturbasjonen endrer banen over en grad, hver dag ( $\frac{d\Omega}{dt} > 1^\circ$ ), mer om  $\frac{d\Omega}{dt}$  senere.

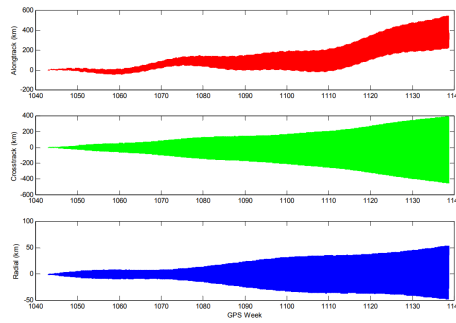


Figure 8: 3D Posisjonsfeil for satellitt (GPS med PRN 3) fra 1 Jan 2000 – 1 Nov 2001 [24].

Satelittene blir altså utsatt for kjente og tilfeldige krefter som kontinurelig endrer banen, som det er nytteløs å prøve å forutse. Figur 8 viser hvor mye en GNSS satellitt kan avvike fra start-banen sin, over ett år. For å løse problemet, er det plassert bakkestasjoner over hele kloden som har i oppgave å regne ut satellittenes bane, for så å sende satellittene oppdatert informasjon om sin egen bane. Bakkestasjonene sender også oppdatert klokkeinformasjon for å synkronisere satellittenes klokke med en masterklokke [8].

Ephemeris data oppdateres hver 2-3 timer og bør ikke brukes etter 4 timer da feilforplantningen øker eksponensielt [9]. Almanac data oppdateres vanligvis et par ganger i uken (ikke sjeldnere enn 6 dager) og har gyldighet på 90 dager. Det er gyldighetstiden til ephemeris data, som i største grad bestemmer gyldighetstiden til kartløsningen i seksjon 7.4.

Vurdering av ephemeris levetid er viktig for prosjektet. Feilforplantningen kan plottes over tid ved å måle avviket mellom en ”start-ephemerisen” og fremtidige ephemeris målinger. Det ble funnet to kilder som beskriver feil i satellittens bane over tid. De er fra 2012 [15] og 2021 [18], og synes i figur 9 og 10. Basert på disse kildene kan man forvente en 3D posisjon median-feil i størrelsesorden 20-30 meter etter 20 dager. Feilforplantning etter 20 dager blir en for trang tidshorisont for hva som er relevant for dette prosjektet.

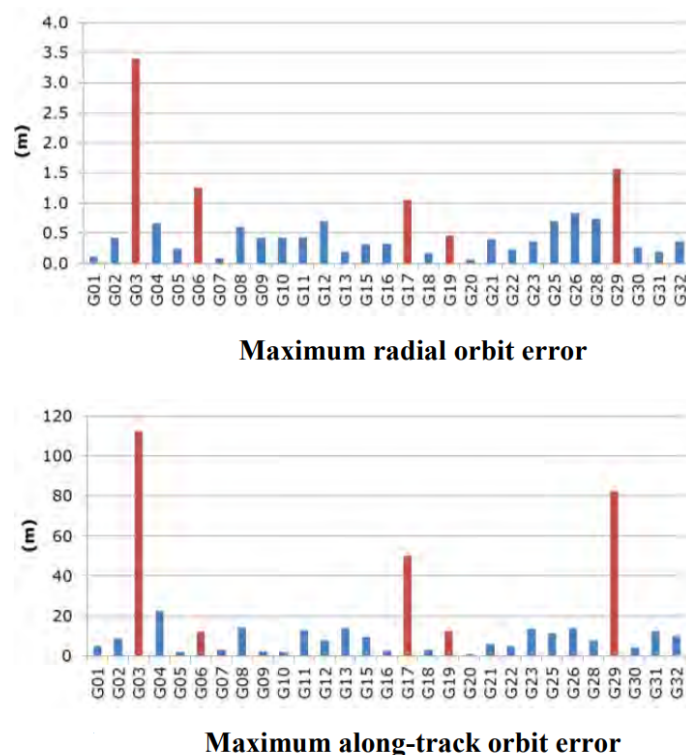


Figure 9: Maks feil radielt og "along track" etter 20 dager. Røde søyler merkerer satellitter som har baner som går "bak" jorden, og skjermes for sola. Disse banene har ekstra stor usikkerhet [15].

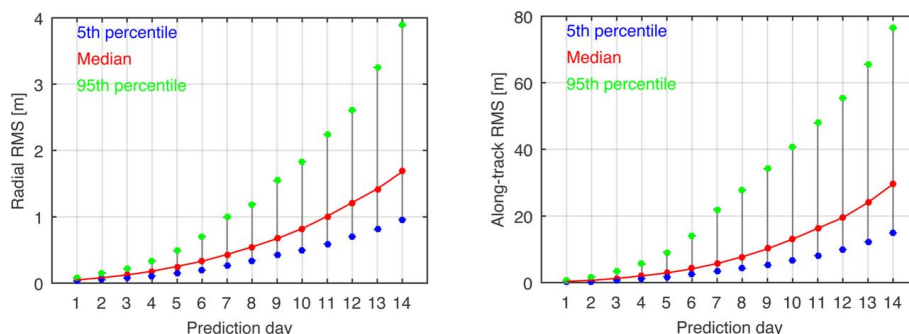


Figure 10: Radiell og along track feil etter [18].

Det ble derfor funnet to andre kilder med lengre beregningstid fra år 2001 [24] og 2004 [22], med figurer vist i 8 og 11. Studerer man figurene ser man at forventer 3D posisjon median-feil i størrelsesorden 7-20 kilometer etter 14 dager. På 14 dager er det altså snakk om kilometer med avvik, ikke meter som for 20 dager (fra 2012 og 2021 kildene). Uenigheten mellom de nyere og eldre kildene er av ukjent opphav. Gitt lengre utregningstid til de eldre kildene, blir data fra figur 12 brukt i videre vurdering.

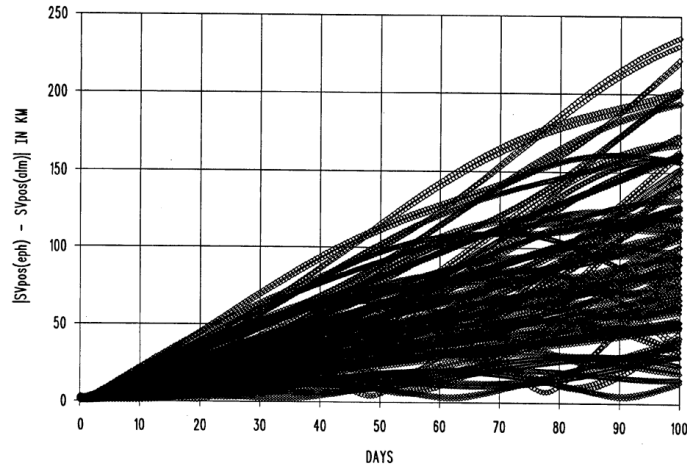


Figure 11: GPS satellitt drift over tid [22].

Age of Data	Minimum / Maximum (km)	Mean (km)	Median (km)	Standard Deviation (km)	90 <sup>th</sup> Percentile (km)	Maximum – Minimum (km)
<b>7 Days</b>	0.85 / 12.84	2.39	2.43	1.59	2.97	11.99
<b>14 Days</b>	0.85 / 203	7.20	7.35	3.52	8.46	202.21
<b>1 Month</b>	0.9 / 1056	28.1	28.3	12.1	33.0	1055.4
<b>3 Months</b>	1.0 / 26751	310	307	127	363	26750
<b>Half Year</b>	1.0 / 43839	1485	1481	544	1673	43839
<b>All 2000</b>	1.0 / 53147	4059	4051	1236	4391	53146
<b>2000 – 2001 (*1)</b>	1.0 / 56912	7461	7449	2315	8159	56911

\*1 – Until 1 Nov 2001

Figure 12: 3D Ephemeris feil over tid for GPS [24].

Weighted GDOP:

## 5 GDOP

GDOP (geometric dilution of precision) er et mål på usikkerhet på posisjon, som er beregnet med GNSS satellitter. GDOP er dimensjonsløst (bare ett tall) og man ønsker å ha så lav GDOP som mulig. GDOP blir en sampling av usikkerhet/DOP'er (dilution of precision) i horisontal og vertikal retning, samt tid. Derav HDOP, VDOP og TDOP. GDOP er bestemt som den euklidske avstanden mellom disse DOP-ene:

$$\text{GDOP} = \sqrt{\text{HDOP}^2 + \text{VDOP}^2 + \text{TDOP}^2} \quad (3)$$

For å tallfeste GDOP starter man med å regne ut avstands-vektoren fra observasjonspunktet  $x, y, z$  ut til den  $i$ -ende satellitten som befinner seg i punktet  $(x_i, y_i, z_i, \tau_i)$ , der  $\tau$  er tidpunktet.:

$$\vec{a} = (x_i - x, y_i - y, z_i - z) \quad (4)$$

Deretter finner vi den normaliserte avstanden  $\vec{d}$  (lengde = 1), ved å dividere med lengden:

$$\vec{d}_i = \left( \frac{x_i - x}{R_i}, \frac{y_i - y}{R_i}, \frac{z_i - z}{R_i} \right), \quad R_i = \text{abs}(\vec{a}) = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (5)$$

Vi definerer *Jacobi* matrisen  $J$ , der hver rad er gradienten/stigningstallet i retning x,y,z, og tid, til den  $i$ -ende satellitten. Det blir antatt  $n$ -antall synlige satellitter. Mhp formel 6 er det verdt å merke seg at gradienten i tid er satt lik 1 for hver satellitt.

$$J = \begin{bmatrix} d_1 & 1 \\ d_2 & 1 \\ \vdots & \vdots \\ d_n & 1 \end{bmatrix} = \begin{bmatrix} d_{x1} & d_{y1} & d_{z1} & 1 \\ d_{x2} & d_{y2} & d_{z2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ d_{xn} & d_{yn} & d_{zn} & 1 \end{bmatrix} = \begin{bmatrix} \frac{x_1-x}{R_1} & \frac{y_1-y}{R_1} & \frac{z_1-z}{R_1} & 1 \\ \frac{x_2-x}{R_2} & \frac{y_2-y}{R_2} & \frac{z_2-z}{R_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_n-x}{R_n} & \frac{y_n-y}{R_n} & \frac{z_n-z}{R_n} & 1 \end{bmatrix} \quad (6)$$

Den siste kolumnen med 1-ere i  $J$  er

Med  $J$ , regner vi ut kovarians-matrisen  $Q$ :

$$Q = (J^T J)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{tx} & \sigma_{ty} & \sigma_{tz} & \sigma_t^2 \end{bmatrix} \quad (7)$$

Det viser seg at

$Q$ , as the covariance matrix resulting from the least-squares normal matrix:

## 6 Beregning av horisontkurve

Det ble brukt....

## 7 GNSS og satellitter

### 7.1 Baner til satellitter

#### 7.1.1 Keplerbane

Objekter som går i lukket bane rundt et annet himmelegeme kan beskrives som en keplerbane. En "lukket bane" betyr at objektet går i som en sirkel eller ellipse og kommer tilbake til startpunktet sitt, etter en runde. GNSS satellitter går rundt jorda to ganger hvert stjernerdøgn (23 timer og 56 minutt).

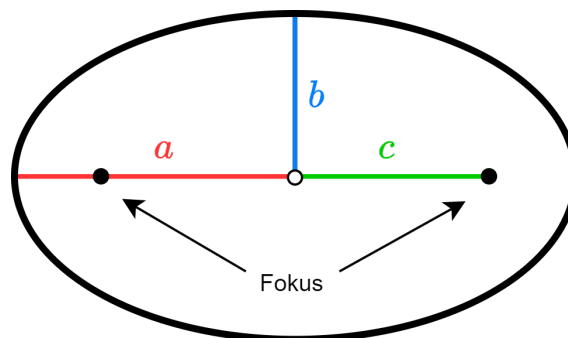


Figure 13: Ellipse.

En keplerbane beskrives av form og størrelse på en ellipse, og hvordan denne ellipsen er orientert rundt jorda, samt *True anomaly* som beskriver satellittens posisjonen gitt tidspunktet. En ellipse

som vist i figur 13 kan beskrives av  $a$  som er den største radiusen til ellipsen,  $b$  er minste radiusen til ellipsen, og  $c$  er avstanden fra sentrum ut til fokusene. I kontekst av satellitter som går i bane rundt jorda, vil jordas masse-senter befinner seg i et av fokusene. I GN beskrives med en semi-major axis ( $a$ ) som angir størrelsen, og eccentricity ( $e$ ) som beskriver hvor rund eller flat den er.

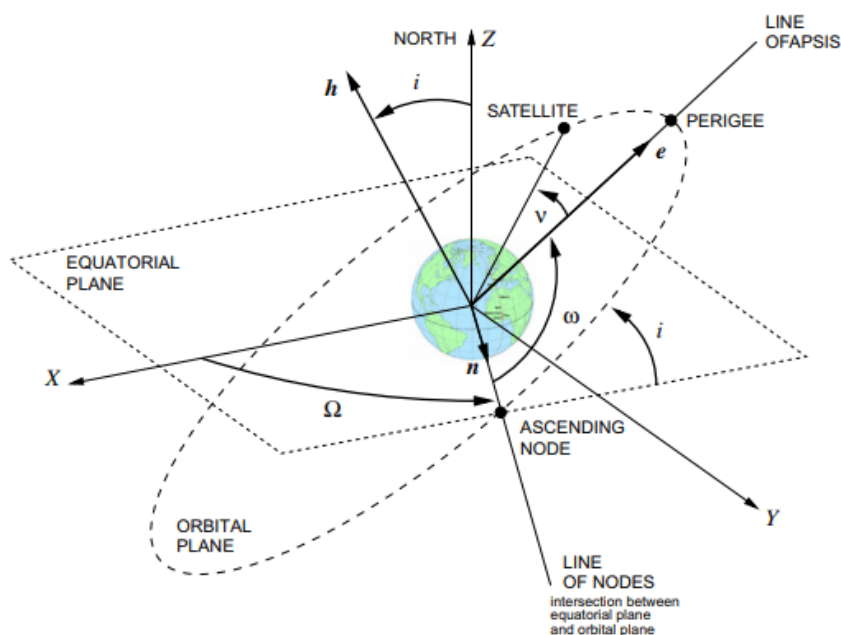


Figure 14: Illustrasjon av kepler bane [baneinfo].  $h$  er normalvektoren til den elliptiske banen.

En keplerbane som vist i figur 14 beskrives med seks parameter:

- Semi-major axis ( $a$ ):  
Lengste radiusen til en ellipse.
- Eccentricity ( $e$ ):  
Beskriver hvor rund eller flat ellipsen er.  $e = \frac{c}{a}$ , der  $c = \sqrt{a^2 - b^2}$ . Når  $e \rightarrow 0$  går kurven mot en ren sirkel, og når  $e \rightarrow 1$  går kurven mot en rett linje.  $e \in [0, 1]$ .
- Inclination ( $i$ ):  
Vinkelen mellom ekvatorplanet og *line of nodes* (fra figur 14).
- Right ascension of the ascending node (RAAN) ( $\Omega$ ):  
Vinkel mellom jorda's  $x$ -akse og *ascending node*. RAAN kan også tenkes på som den longitudinale retning til ellipsen. Ascending node er punktet der satellitten krysser ekvatorplanet fra sør til nord. For GNSS satellitter benyttes koordinatsystemet ECEF (Earth Centered, Earth Fixed). I ECEF blir  $x$ -aksen angitt ved nullmeridianen på ekvatorplanet,  $y$ -aksen er  $90^\circ$  øst fra  $x$ -aksen (på ekvatorplanet), og  $z$ -aksen ut fra den geografiske nordpolen.
- Argument of perigee ( $\omega$ ):  
Vinkel mellom *ascending node* og perigee. Argument og perigee kan også tenkes på som den latitudinale retningen til ellipsen. Perigee er punktet langs ellipsen som er nærmest jorda's masse-senter. Perigee er i praksis i den spisse enden av ellipsen nær fokuset der jorda befinner seg.
- True Anomaly ( $v$ ):  
Vinkel mellom perigee og satellitten. Forteller hvor på selve ellipsen satellitten befinner seg på et tidspunkt.

---

Det er verdt å merke seg at det brukes ulike standarder for å beskrive parameterene til en keplerbaner, men at de da fortsatt inneholder samme informasjon om banen, og at de enkelt kan gjøres om til de seks typene ovenfor. Eksempelvis kan du komme over  $\sqrt{a}$  istedenfor  $a$ , eller Mean anomaly istedenfor True Anomaly.

For å forstå keplerbaner anbefales det å prøve selv gjennom visualiseringsverktøy som [https://ssd.jpl.nasa.gov/tools/orbit\\_diagram.html](https://ssd.jpl.nasa.gov/tools/orbit_diagram.html) der man kan simulere banen til en satellitt rundt sola.

## 7.2 Satellitt data, RINEX

Receiver Independent Exchange format (RINEX) er et standardisert dataformat brukt i GNSS systemet [19]. I en verden der flere nasjoner etterhvert sendte opp sine egne navigasjonssatellitter ble det behov for et standardisert dataformat, slik at den moderne verden kunne bygges opp på et samlet og robust navigasjonsnettverk. RINEX er både det råe formatet som blir sendt fra satellittene, og formatet som blir distribuert og brukt her på bakken.

RINEX satellitt data er tilgjengelig i datasenteret GAGE (Geodetic Facility for the Advancement of Geoscience). GAGE drives av UNAVCO som er et globalt samarbeid av universitet [23].

\* GNSS ground stations kontrollerer og korrigerer baneinformasjonen distribuert av satellittene. Satellitt banene er ikke helt determinerbare pga faktorer som drift i klokke, og fra generell relativitet.

Det trengs tilsynelatende bare 3 satellitter for å estimere en posisjon ved bruk av GNSS, men dette gir stor TDOP. GNSS satellitter bruker atomklokker som gir neglisjerbar usikkerhet i tid, men det samme gjelder ikke for mottakeren (mobiltelefon, kjøretøy, fly osv...). Derfor får man en ukjent usikkerhet i tid, og trenger derfor en fjerde satellitt for å minke denne usikkerheten [21]. Litt som at man trenger 4 ligninger for å løse for 4 ukjente.

Sveinung Myhre - Leder av Cogito NTNU 2023/2024 - Fremtidens biler vil helt sikkert bruke GDOP heatmap i enten navigasjon eller posisjonering.

## 7.3 Mask angle

Signal fra GNSS satellitter som ligger lavt i horisonten blir ofte filtrert bort. Ved lav horisont-vinkel blir signalene forvridd gjennom prosessen *atmosfærisk refraksjon*. Signal fra GNSS satellittene er elektromagnetiske bølger, akkurat slik som synlig lys. Litt på samme måte som lyset bøyer seg når du ser gjennom et glass med vann, vil GNSS signalene bøye seg gjennom de ulike lagene i atmosfæren.

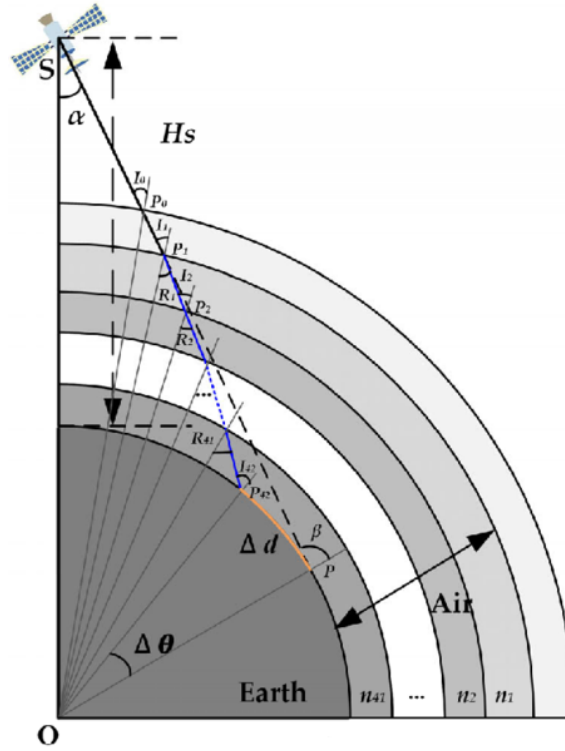


Figure 15: Illustrasjon hvordan atmosfærisk refraksjon forvrenger GNSS signal [1]. Relevante parametere er zenith angle  $\beta$  (vertikal vinkel) for en observerende i punkt  $P$ , og feil-posisjon fra atmosfærisk refraksjon  $\Delta d$ .

Figur 15 er hentet fra en artikkel om korrigering for atmosfærisk refraksjon [1]. I tabell 3 i artikkelen vises en sammenheng om at økende "zenith angle"  $\beta$  gir høyere "atmospheric refraction error"  $\Delta d$  (fra figur 15), som i praksis betyr stor VDOP. For en mottaker i et punkt  $P$  som observerer et signal fra bunnen av horisonten ( $\beta \rightarrow 90^\circ$ ) blir denne feilen veldig stor, og signalet bør forkastes. Det blir brukt korrigerings-teknikker der forvrengingen fra atmosfæren blir tatt høyde for, men for signal fra lave horisontvinkler blir forvrengingen for stor, og bør derfor forkastes.

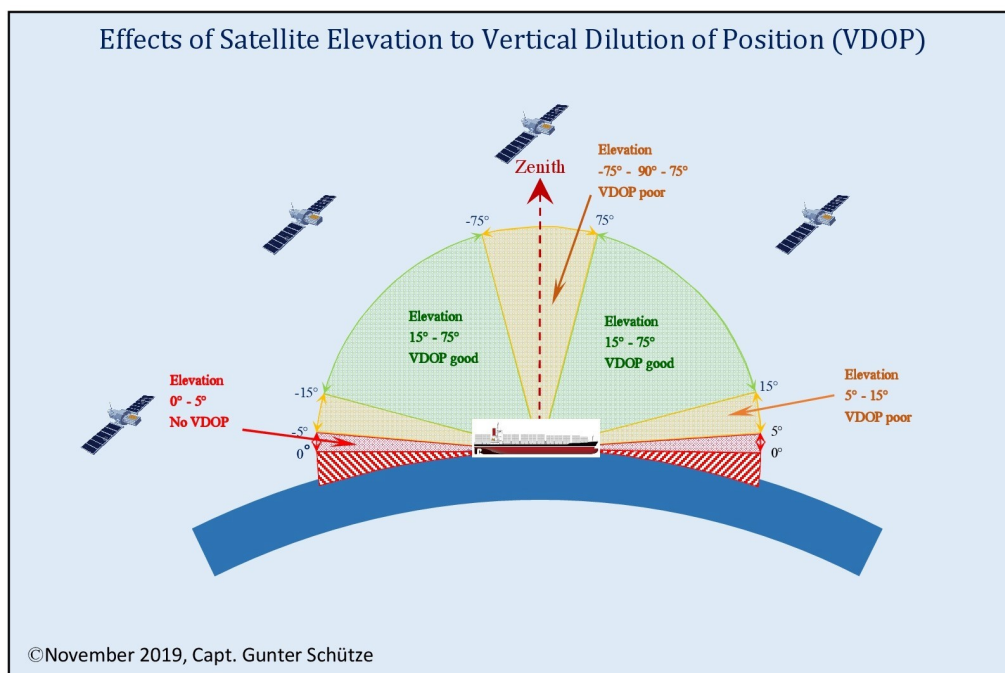


Figure 16: Illustrasjon som viser at GNSS signal fra lav horisontvinkel gir høy VDOP [20].

Derfor blir mask angle  $m_\alpha$  introdusert for å filtrere bort GNSS satellitter som ligger for lavt i horisonten.  $m_\alpha$  er vinkelen fra horisonten, i motsetning til  $\beta$  som er vinkelen fra jordas' normalvektor. GPS standarden [5] er  $m_\alpha = 5^\circ$ , men det blir brukt  $m_\alpha$  opp til  $20^\circ$ . En tilleggs-info her er at satellitter gruppert rett ovenfor mottakeren gir høy VDOP som resulterer i høy GDOP.

### 7.3.1 Mask angle for optimalisering

Det som er verdt å merke seg er at langs kysten eller fra et fjell kan vertikal vinkel  $\beta$  faktisk være større enn  $90^\circ$ . Skal man finne horisontlinjen når  $\beta > 90^\circ$ , feks ved bruk av "skyline"-verktøyet fra ArcGIS Pro, vil algoritmen bli ekstra treg. Dette skjer fordi man i høyden normalt har et større synsfelt, og algoritmen må derfor behandle mer kartdata. Ettersom man i praksis bør ignorere satellitter fra under en hvis vinkel, trenger man ikke å regne ut horisontlinjen under  $m_\alpha$ .

## 7.4 Database for horisontvinkler og GNSS posisjon

GDOP beregninger er ressurskrevende, og det er et alternativ å lage en enkel kartapplikasjon, der man kun gjør utregninger i punktet brukeren velger selv.

Dette vil ikke være tilstrekkelig ettersom beregning av GDOP i første omgang vil være et kartleggingsverktøy, og i fremtiden, kunne bli brukt av posisjonerings- og rutevalgslgoritmene i fremtidens biler. Det bør derfor opprettes en database med horisontvinkler, satellittposisjon, og GDOP langs norske veier.

### 7.4.1 Presisjon

Oppretting og drift av store databaser er dyrt. Heldigvis kan nevnt database generes på eksisterende data (høydekart, vegkart, satellitt-baner), og trenger derfor ikke kartlegging ute i felt. Unntaket er verifisering og testing.

Det bør foretas tester for å vurdere hvor stor presisjon som er hensiktsmessig. Hva som er



---

hensiktsmessig bestemmes av bruken; skal det kun være til kartlegging, eller til distribusjon i fremtidens vegtrafikk. De viktigste parameterene for presisjon er:

- Oppløsning på høydedata til å beregne horisontlinje (DEM50, DEM10, eller DEM1?)
- Oppløsning/steglengde langs veg-kurven.
- Oppløsning i tid (Satellittene beveger seg raskt over horisonten).
- Evt begrensing av mask angle  $m_\alpha$ .

Andre, mindre viktige parametere kan være bilens høyde, posisjon i veien (kjørefelt), algoritme som brukes for å beregne GDOP, og miljøfaktorer som temperatur og luftfuktighet

## Oppløsning av høydedata

Det er tenkelig at lav usikkerhet på horisontlinjen, feks 1%, er tilstrekkelig. Da trengs ikke DEM1 å brukes på store avstander da det har 2499 ganger flere datapunkter per rute, sammenlignet med DEM50. Det ville resultert i betydelig større beregninger og lagring. Det bør derfor lages en algoritme velger oppløsning på kartmodell, basert på avstand fra veien. Dette er spesielt viktig i bebygde strøk der GNSS signal kan bli skjermet av høye bygg kan gå usett inni en 50x50 meter rute (fra et DEM50 kart).

Det foreslås derfor å velge en maksimal usikkerhet for horisontlinje som er konstant, som brukes til å velge hvilken av de standardiserte kartmodellene som bør brukes, ved hvilken avstand.

FIX

Skriv formler for hvilken høydemodell-kartoppløsning som bør brukes, gitt avstand fra observatør og konstant max usikkerhet

FIX

Selvom DEM1, DEM10, og DEM50 er standarder, kan det også generes andre oppløsninger for gjennere "hopp i usikkerhet" mellom de ulike modellene, og gi mindre ressurskrevende utregning av horisontlinje. Fordelen med denne tilnærming virker ikke tilstrekkelig stor til å generere og bruke høydedata utenfor standardene.

## Oppløsning vegmodel

Dette er trolig parameteren som er mest fleksibel når det kommer til størrelsen på databasen. Det bør gjennom empirisk testing finnes ut hvor ofte horisontlinje med tilhørende GDOP bør regnes ut. Oppløsningen på posisjonsdata til norske veier er 1 meter (NVDB). Trolig trengs ikke beregning av horisontlinje og GDOP å gjøres oftere enn hver 50-500 meter, basert på hva behov testmålingene viser til. Det som er fint er at beregningen kan gjøres med en iterativ algoritme som eksmepelvis kan starte med å gjøre utregninger hver 200 meter. Dersom det sener blir funnet ut at høyere oppløsning er ønskelig, kan man forskyve algoritmen med 100 meter for så å nok en gang gjøre utregninger hver 200 meter. Man kan dermed oppnå dobbel oppløsningen, uten å gjenta tidligere utregninger.

Illustrasjon over hvordan oppløsning langs en rett vegkurve kan utvides gjennom flere iterasjoner:

Etter første utregningsrunde:

1 - - - - - 1 - - - - - 1 - - - - - 1 - - - - - 1

Etter andre utregningsrunde med dobbel oppløsning:

---

1 - - - - 2 - - - - 1 - - - - 2 - - - - 1 - - - - 2 - - - - 1 - - - - 2 - - - - 1

FIX

Skriv formler for hvilken høydemodell-kartoppløsning som bør brukes, gitt avstand fra observatør og konstant max usikkerhet

FIX

Oppløsning i tid

Begrensing av mask angle  $\alpha$

Tross mindre utregning ved bruk av mask angle som beskrevet i 7.3.1, bør man ikke begrense horisontlinje-søket til vinkel  $m_\alpha$ . Dette er fordi GNSS systemet er i konstant utvikling med fremtidige generasjoner GNSS satellitter og bedre signalbehandling, som betyr at man kan beregne posisjon fra satellitter som ligger enda lavere i horisonten. Mask angle  $m_\alpha$  bør heller være en dynamisk parameter som kan justeres i tråd med hva som gir lavest GDOP under test, til gitt tid. Horisontlinjen bør beregnes slik den faktisk er, så man ikke trenger å gjøre mer presis utregning i fremtiden.

Livstid for Database for horisontvinkler

Beregning av GDOP deles i to deler, GNSS satellitt-posisjoner, og beregning av horisontlinje på norske veier.

Satellittbilde vil endres over tid og vil derfor gjøre GDOP beregning utdatert.

Norske veier og terreng vil også endre seg, men i mye mindre grad.

## 8 ArcGIS

ArcGIS er et økosystem av kart løsninger og dataprogram som er aktivt brukt av vegvesenet. I dette prosjektet er ArcGIS Pro benyttet for "proof of concept"

For å beregne "zenith"-vinkel fra et gitt punkt til horisontlinjen er verktøyet "Skyline Graph" brukt. Man kan også visualisere 3D linjen til selve horisontlinjen med verktøyet "Skyline".

1. Fant DTM50 kart fra: <https://kartkatalog.geonorge.no/metadata/dtm-50/e25d0104-0858-4d06-bba8-d154514c11d2>

2. Trykket nedlastnings-ikonet opp til høyre og kom til: <https://kartkatalog.geonorge.no/nedlasting>

3. Under geografisk område, trykker "Velg fra kartblad", finner kartruten over Romsdalen som er 6901. Velger default Projeksjon (EUREF89 UTM sone 33, 2d) og Format DEM.

4. Når kartet er lastet ned, flytter den til "temp" mappen som er opprettet ytterst i C:-drive.

5. Åpne DEM filen i ArcGIS Pro test prosjektet.

6. Hent inn vegstrekning du ønsker å analysere fra [vegkart.no](http://vegkart.no). I vårt tilfelle henter vi E134 fra Dombås til Åndalsnes, som nedlastet SOSI fil. SOSI filen konverterer vi til shape-filer med <https://app.sosicon.espenandersen.no/>, ettersom standard ArcGIS ikke støtter .sos (SOSI) filer. Omgjøring gir ut fire filer .shp, .shx, .prj, og .dbf. Her er .shp hovedfilen som vi importerer ved å bruke verktøyet "Data From Path" som man finner i fanen Map->Layer->Data From Path. Filstien jeg måtte skrive inn her var "C:\36\_S817136\_S817.shp", *og kan være uelik, alt etter hvor du har lastet ned filene.*

---

7. Bruker verktøyet "Dissolve" med vegstrekningsfilen som input, uten å gjøre noe med de andre parameterene. Dette verktøyet slår sammen alle veiobjekt-strekningene fra .shp filen, til én sammenhengende strekning. Verktøyet generer ikke 3D punkter som evt ligger utenfor høydekartet (.dem), som for de 7 første punktene sett fra Dombås. Disse 7 punktene ligger utenfor kart 6901 som er det eneste høydekartet vi her benytter.

8. Bruk verktøy "Generate Points Along Lines" for med parameter "Point Placement"="By Distance". Vi setter "Distance" parameteren lik 1000 meter i dette tilfellet. Hadde vi ikke brukt "Dissolve" verktøyet før, ville alle veiobjekt-strekningene blitt behandlet hver for seg, og vi ville dermed ikke fått noen punkt på vegobjekt-strekninger under 1000 meter, og resultatet ville blitt feil.

9. Disse punktene vi genererte med gjevne mellomrom langs veien er 2D koordinater som ikke har noen høyde (Z value i ArcGIS). Derfor må vi kjøre disse punktene gjennom verktøyet "Interpolate Shape". Punktene langs linjene er "Input Features" og .dem høydefilen er "Input Surface".

10. De nye 3D punktene vi fikk kan vi putte inn i verktøyet "skyline". Dette generer horisontlinjen sett fra hver av 3D punktene. En liten detalje om skyline er at algoritmen går ovenfra og ned som forklart . Det betyr i .

11. Siste steg er å bruke verktøyet "Skyline Graph" for å vinne horisont vinkelen fra hvert punkt. Her er "Input Observer Point Features" de samme 3D punktene vi brukte i "Skyline" verktøyet. "Input Line Features" skal være horisontlinjene som ble output fra "Skyline" verktøyet.

- For å få XYZ koordinater inn i ønsket table der det bare finnes shape.\* fra før, bruk verktøyet "Add XY Coordinates". Da blir det lagt til X,Y,Z kolonner i datatabelen som har liste av point z. Nyttig for punktene med bestemt avstand langs vegkurven. Får ikke nedlastet punkt dataen bare gjennom shape-objekt-kolonnen.

For å finne horisontlin

Verktøyet "Skyline" i ArcGIS vil som utgangspunkt scanne hele høydekartet for potensielle fjell som kan begrense horisontlinjen. Det vil resultere i unødvendig store beregninger. For å optimalisere denne prosessen ønsker vi å estimere den lengste avstanden fra observasjonspunktet som "Skyline" trenger å lete før den sier seg fornøyd og stopper søket.

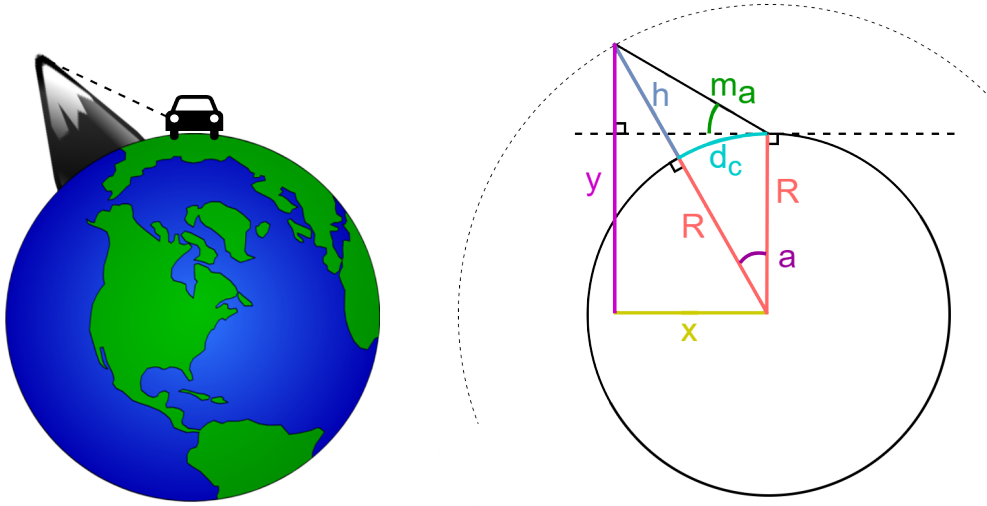


Figure 17: Illustrasjon.

Litt mer konkret ønsker vi å finne avstand  $d_c$  langs jordoverflaten til fjell med høyde  $h$ , jord-radius  $R$ ; gitt en mask angle  $m_a$ .  $d_c$  kan finnes med utgangspunkt i figur 17 der man begynner med å regne ut punktet  $(x,y)$  til fjelltoppen sett fra jordas sentrum:

$$y = R + \tan(m_a)x \quad (8)$$

$$x^2 + y^2 = (R + h)^2 \quad (9)$$

som kan løses for  $x$  ved innsetting av  $y$  og bruk av andregradsformelen slik at:

$$x(m_a) = \frac{-2R \tan(m_a) + \sqrt{(2R \tan(m_a))^2 - 4(1 + \tan(m_a)^2) \cdot (-2Rh - h^2)}}{2(1 + \tan(m_a)^2)} \quad (10)$$

Nå som vi kjenner punktet til fjelltoppen kan vi finne vinkel  $a$ :

$$a(m_a) = 90 - \arccos\left(\frac{x(m_a)}{R + h}\right) \quad (11)$$

Til bestemmer vi finner vi sirkelbuen  $d_c$  som funksjon av  $m_a$ :

$$d_c(m_a) = 2\pi R \cdot \frac{a(m_a)}{360} \quad (12)$$

Formelen for  $d_c$  er basert på noen antagesler og vi må bestemme noen verdier:

- Jorda tilnærmes en perfekt kule med radius  $R = 6371.0088$  km.
- Havnivået er ved overflaten på kula.
- Ingen norske veier er under havnivå.
- $h = 2.469$  km, som er bestemt av Norges høyeste fjell (Galdhøpiggen).
- Mask angle settes lik  $5^\circ$ .

Setter man disse verdiene inn i formel (12) får man  $d_c \approx 28$  km. Inni "skyline" verktøyet settes må parameterene `max_horizon_radius = 28000` og `use_curvature='CURVATURE'`.

---

## **9 Resultater**

Våre resultater var...

## **10 Diskusjon**

Vi diskuterer....

## **11 Konklusjon**

---

## References

- [1] Xiaohong Peng Wenwen Huang Xiaoyan Li Lin Yang Fansheng Chen. *MDPI, A Spatiotemporal Atmospheric Refraction Correction Method for Improving the Geolocation Accuracy of High-Resolution Remote Sensing Images*. URL: [https://www.researchgate.net/publication/364763913\\_A\\_Spatiotemporal\\_Atmospheric\\_Refraction\\_Correction\\_Method\\_for\\_Improving\\_the\\_Geolocation\\_Accuracy\\_of\\_High-Resolution\\_Remote\\_Sensing\\_Images](https://www.researchgate.net/publication/364763913_A_Spatiotemporal_Atmospheric_Refraction_Correction_Method_for_Improving_the_Geolocation_Accuracy_of_High-Resolution_Remote_Sensing_Images) (visited on 5th July 2024).
- [2] Penn State College. *The Delay Lock Loop*. URL: <https://www.e-education.psu.edu/geog862/node/1757> (visited on 10th July 2024).
- [3] The University of Tokyo Dinesh Manandhar CSIS. *Introduction to Global Navigation Satellite System (GNSS) Software GPS Receiver*. URL: <https://www.unoosa.org/documents/pdf/icg/2018/ait-gnss/13.SGR.pdf> (visited on 30th July 2024).
- [4] EUSPA Services. URL: <https://www.euspa.europa.eu/eu-space-programme/egnos/services> (visited on 8th July 2024).
- [5] Anthony Flores. *NAVSTAR GPS. Space Segment/Navigation User Segment Interfaces*. URL: <https://www.gps.gov/technical/icwg/IS-GPS-200L.pdf> (visited on 3rd July 2024).
- [6] GMV. *Correlation*. URL: <https://gssc.esa.int/navipedia/index.php/Correlators> (visited on 10th July 2024).
- [7] GMV. *Generic Receiver Description*. URL: [https://gssc.esa.int/navipedia/index.php/Generic\\_Receiver\\_Description](https://gssc.esa.int/navipedia/index.php/Generic_Receiver_Description) (visited on 30th July 2024).
- [8] GMV. *Precise Time Reference*. URL: [https://gssc.esa.int/navipedia/index.php/Precise\\_Time\\_Reference](https://gssc.esa.int/navipedia/index.php/Precise_Time_Reference) (visited on 18th July 2024).
- [9] J.M. Juan Zornoza J. Sanz Subirana and Spain. M. Hernández-Pajares Technical University of Catalonia. *GPS and Galileo Satellite Coordinates Computation*. URL: [https://gssc.esa.int/navipedia/index.php/GPS\\_and\\_Galileo\\_Satellite\\_Coordinates\\_Computation](https://gssc.esa.int/navipedia/index.php/GPS_and_Galileo_Satellite_Coordinates_Computation) (visited on 8th July 2024).
- [10] J.M. Juan Zornoza J. Sanz Subirana and Spain. M. Hernández-Pajares Technical University of Catalonia. *Perturbed Motion*. URL: [https://gssc.esa.int/navipedia/index.php?title=Perturbed\\_Motion](https://gssc.esa.int/navipedia/index.php?title=Perturbed_Motion) (visited on 8th July 2024).
- [11] J.M. Juan Zornoza J. Sanz Subirana and Spain. M. Hernández-Pajares Technical University of Catalonia. *Relativistic Clock Correction*. URL: [https://gssc.esa.int/navipedia/index.php/Relativistic\\_Clock\\_Correction](https://gssc.esa.int/navipedia/index.php/Relativistic_Clock_Correction) (visited on 9th July 2024).
- [12] JM. Juan Zornoza J. Sanz Subirana and Spain M. Hernandez-Pajares University of Catalonia. *GPS Navigation Message*. URL: [https://gssc.esa.int/navipedia/index.php/GPS\\_Navigation\\_Message](https://gssc.esa.int/navipedia/index.php/GPS_Navigation_Message) (visited on 8th July 2024).
- [13] JM. Juan Zornoza J. Sanz Subirana and Spain. M. Hernandez-Pajares University of Catalonia. *GNSS Signal*. URL: [https://gssc.esa.int/navipedia/index.php/GNSS\\_signal](https://gssc.esa.int/navipedia/index.php/GNSS_signal) (visited on 10th July 2024).
- [14] JM. Juan Zornoza J. Sanz Subirana and Spain. M. Hernandez-Pajares University of Catalonia. *GNSS Signal*. URL: <https://web.gps.caltech.edu/classes/ge111/Docs/GPSbasics.pdf> (visited on 10th July 2024).
- [15] Miguel M. Romay og María D. Laínez - GMV. *Generation of Precise Long-Term Orbit and Clock Prediction Products for A-GNSS*. URL: [https://magicgnss.gmv.com/publications/2012/ION\\_GNSS\\_2012\\_magicODTS\\_Long\\_Term\\_Predictions.pdf](https://magicgnss.gmv.com/publications/2012/ION_GNSS_2012_magicODTS_Long_Term_Predictions.pdf) (visited on 27th July 2024).
- [16] *New Galileo service set to deliver 20 cm accuracy*. URL: [https://www.esa.int/Applications/Satellite\\_navigation/New\\_Galileo\\_service\\_set\\_to\\_deliver\\_20\\_cm\\_accuracy](https://www.esa.int/Applications/Satellite_navigation/New_Galileo_service_set_to_deliver_20_cm_accuracy) (visited on 10th July 2024).
- [17] YuXiang Peng and Wayne A. Scales. *Ionospheric Remote Sensing with GNSS*. URL: <https://www.mdpi.com/2673-8392/1/4/94> (visited on 8th July 2024).
- [18] Oliver Montenbruck og Peter Steigenberger og Moritz Aicher. *A long-term broadcast ephemeris model for extended operation of GNSS satellites*. URL: <https://navi.ion.org/content/navi/68/1/199.full.pdf> (visited on 27th July 2024).

- 
- [19] *RINEX*. URL: <https://en.wikipedia.org/wiki/RINEX> (visited on 7th July 2024).
- [20] Gunter Schütze. *GPS (Part 2), physical and technical errors of GNSS - an error analysis*. URL: <https://www.linkedin.com/pulse/gps-part-2-physical-technical-errors-gnss-error-analysis-sch%C3%BCtze> (visited on 5th July 2024).
- [21] *The Global Positioning System*. URL: [https://oceanservice.noaa.gov/education/tutorial\\_geodesy/geo09\\_gps.html](https://oceanservice.noaa.gov/education/tutorial_geodesy/geo09_gps.html) (visited on 7th July 2024).
- [22] Robert Harbour Thomas M. King og George J. Geier og Mark Heng. *United States, Patent Application Publication, Pub. No.: US 2004/026338.6 A1*. URL: <https://patentimages.storage.googleapis.com/54/48/9a/df1523946094b1/US20040263386A1.pdf> (visited on 27th July 2024).
- [23] *UNAVCO*. URL: <https://en.wikipedia.org/wiki/UNAVCO> (visited on 7th July 2024).
- [24] David L.M. Warren. *BROADCAST VS PRECISE GPS EPHEMERIDES: A HISTORICAL PERSPECTIVE*. URL: <https://apps.dtic.mil/sti/tr/pdf/ADA401477.pdf> (visited on 27th July 2024).
- [25] Chunxi Zhang og Xianmu Li og Shuang Gao og Lu Wang. *Performance Analysis of Global Navigation Satellite System Signal Acquisition Aided by Different Grade Inertial Navigation System under Highly Dynamic Conditions*. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5467708/> (visited on 30th July 2024).

= Bibliography]

---

## Appendix

### A Hello World Example