



Norges miljø- og
biovitenskapelige
universitet

Masteroppgave 2016 30 stp
Institutt for matematiske realfag og teknologi

GPS: Forsinkelse gjennom troposfæren, posisjon og meteorologisk data-assimilering

GPS: Tropospheric Delay, Position and
Meteorological Data Assimilation

Hanne Disch Kihle
Miljøfysikk og fornybar energi

Forord

Etter 5 fantastiske år på Ås leveres nå masteroppgaven som en avslutning på min mastergrad i Miljøfysikk og fornybar energi ved NMBU. Oppgavens omfang er 30 studiepoeng og ble skrevet våren 2016.

Mange arbeidstimer har gått med på denne oppgaven, og likevel føles det som jeg så vidt har rukket å starte før den nå skal leveres. Et samarbeid mellom Kartverket og Meteorologisk Institutt er veldig spennende og her er det enda mye som kan undersøkes.

Først og fremst vil jeg takke min hovedveileder Oddgeir Kristiansen for god veiledning og ikke minst for at jeg fikk muligheten til å sitte på kartverkskontoret i Oslo. Her har jeg fått mulighet til å bli kjent med mange flotte mennesker, som har hjulpet meg mye. En spesiell takk til Geir Arne Hjelle og Michael Dähnn for hjelp med Latex, Python og GIPSY. Jeg har også fått god hjelp fra Roger Randriamampianina, Mareile Astrid Wolff, Snorre Stavik Rønningen og Christoffer Elo fra Meteorologisk Institutt. En stor takk går også til min NMBU-veileder Arne Auen Grimenes.

Deretter vil jeg takke Ingrid, Birgit, Torhild, Sigrun, Tone og Maren Anna for fantastiske år på NMBU. Takket være dere har min studietid på Ås vært kjempebra, og savnet vil bli stort når jeg nå flytter til hovedstaden.

Tilslutt vil jeg takke kjæresten min, Pelle, samt foreldre og søsken for gode råd og støtte.

Ås, 13.mai 2015

Hanne Disch Kihle

Sammendrag

GPS-signalets forsinkelse gjennom troposfæren påvirker beregningen av posisjonen i geodesi. Ulik a priori informasjon om trykk, temperatur og vanndamptrykk fra forskjellige kilder blir i denne oppgaven testet for å se hvilke av metodene som gir best resultat i posisjonberegning. Det ble i tillegg testet hvilke av de ulike metodene som egner seg best til estimering av total forsinkelse, som kan brukes i meteorologisk data-assimilering. Metodene som testes er Vienna Mapping Function (VMF1) basert på en numerisk værmodell, Global Temperature and Pressure Model (GPT2), og en lokal numerisk værmodell fra meteorologisk institutt (MET). På grunn av en høydedifferansen mellom den lokale værmodellen og faktisk høyde for stasjoner ble det i tillegg undersøkt flere metoder for beregning av a priori informasjon, der en høydekorrigering ble inkludert.

Det var forskjeller i total forsinkelse for de ulike metodene. Det kom tydelig frem her at det var en feil-modellering av den lokale værmodellen og en høydekorreksjon var derfor nødvendig. Det var en breddegradsavhengighet mellom gjennomsnittlig total forsinkelse for VMF1 og GPT2, med en større forskjell i nord iforhold til sør. Etter høydekorrigeringen ble det liten forskjell mellom VMF1 og MET. Ved å se nærmere på stasjonene kunne det se ut som MET hadde en finere tidsoppløsning enn VMF1, når en så på a priori informasjon.

Av resultatene var det VMF1 og MET-metodene som ga best resultater for RMS-verdien til vertikal posisjon langs kysten i sør, mens det på innlandet var mer tilfeldig hvilke som ga best resultat. På noen innlandsstasjoner ble GPT2 den beste metoden. VMF1 med ultra rapid baneparametere ga høyest RMS-resultater. Når atmosfærisk loading ble inkludert fikk VMF1 og MET en relativt stor reduksjon i RMS på innlandet og mot nord. VMF1 ble den beste metoden etter korreksjonen og GPT2 den dårligste.

I data-assimileringen på MET kunne vi se at lignende funn ble gjort: GPT2 ga for noen stasjoner best resultater på innlandet, mens motsatt var tilfelle langs kysten. Generelt ble resultatene langs kysten dårligere enn på innlandet og nord, sannsynligvis fordi det er større endringer fra dag til dag i trykk, temperatur og vanndamptrykk ved kysten.

Det konkluderes med at atmosfærisk loading bør inkluderes når en skal se på forsinkelsen gjennom troposfæren, og at det bør skje en høydekorrigering for MET-dataene.

Abstract

Delay in the GPS-signal through the troposphere has a great impact on position in geodesy. In this thesis different a priori information about pressure, temperature and water vapor from different sources are tested to see which one gives the best result when looking at position. In addition this delay can also be used for meteorological purposes. The total delays in zenith are investigated to see which method gives the best result in the meteorological data assimilation. The different models for a priori information used here is Vienna Mapping Function (VMF1), based on a numerical weather model, a Global Temperature and Pressure model (GPT2) and a local numeric weather model directly from MET (MET). Because the local model uses a different height corresponding to the same longitude and latitude as the site, it was also investigated different methods correcting for this height difference.

The results showed that there is a difference in the total delay between the methods. There was a great impact of the mismodelling from the local MET-model when looking at this delay, making it clear that the height correction was necessary. The difference between the mean of the total delay for VMF1 and GPT2 was latitude dependent, with higher total delay in the north when using GPT2 compared to the south. The difference between the mean total delay for VMF1 and the MET-model when corrected for height was very small, but when looking closely at some stations it seemed like the local models had a higher time resolution than VMF1, when looking at a priori information.

VMF1 and the MET corrected for height had low RMS-values in the southern coast, and higher in the inland and north. GPT2 showed for some inland stations better results than the other methods. VMF1 with ultra-rapid orbits gave the worst results when looking at the RMS-values. When corrected for atmospheric loading VMF1 become the best method whereas GPT2 the worst.

The data-assimilation from MET showed similar results, and GPT2 gave, for some stations, better results in the inland and VMF1 better near the coast. In general the result was better in the inland and the north, and worse near the southern coast. This can be due to the rapid changes in temperature, pressure and water vapor here.

The conclusion is that for further studies height correction and atmospheric loading should be included.

Innhold

1. Innledning	1
2. Teori	3
2.1. Observasjonsligning	3
2.2. Troposfæreforsinkelse	4
2.2.1. Integrated water vapor	8
2.2.2. Vienna Mapping function (VMF1)	9
2.2.3. Global Pressure and Temperature model (GPT2)	10
2.2.4. MET-data: AROME-MetCoOp	11
2.2.5. Atmosfærisk loading	12
2.3. Presis enkeltpunkts stedfesting (PPP)	12
2.3.1. Kalmanfilter	14
3. Metode	19
3.1. Troposfæremodellering med GIPSY PPP	19
3.1.1. MET-data: A priori informasjon om ZHD og ZWD	20
3.2. Data-assimilering MET	22
4. Resultater GIPSY PPP	27
4.1. Hydrostatisk forsinkelse i senit	29
4.1.1. Met-direkte	29
4.1.2. MET-modellene	33
4.2. Estimert våt forsinkelse i senit	33
4.3. Total forsinkelse i senit	37
4.3.1. MET-metodene	37
4.4. Tidsserieanalyse	41
4.4.1. MET-metodene	45
4.4.2. Tidsserieanalyse	46
4.4.3. Korreksjon for atmosfærisk loading	53
5. Resultater data-assimilering	61
5.1. Brukbare data	61
5.2. Tidsserieanalyse MET	65
6. Diskusjon	73
7. Konklusjon	77

A. Høydeforskjell	89
B. Ionosfærefri lineærkombinasjon	91
C. Fra geopotensial til geomterisk høyde	93
D. Nedbør 2015	95
D.0.1. November	95
D.0.2. Desember	96
E. Kildekode	97
E.1. MET2TDP	97
E.2. COST-format	100

Forkortelser

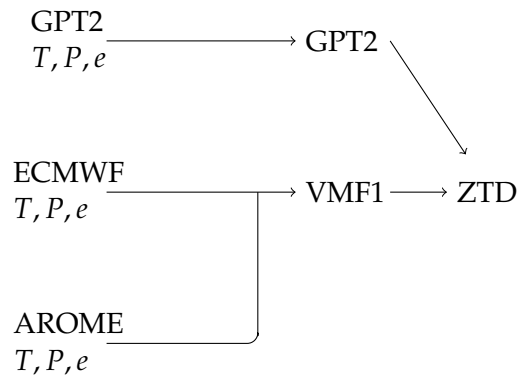
Forkortelse	Forklaring
GPS	Global Positioning System
ECMWF	European Centre for Medium-Range Weather Forecast
AROME- MetCoOp	Applications and Research to Operations at Mesoscale Meteorological Co-operation on Operational
NWP	Numerical weather Prediction
NMF	Niell Mapping Function
VMF	Vienna Mapping Function
VMF1	Vienna Mapping Function oppdatert
GMF	Global Mapping Function
GPT	Global Pressure and Temperature modell
GPT2	Global Pressure and Temperature modell oppdatert
MET	Meteorologisk institutt
P-kode	Precise / Protected kode
C/A-kode	Clear / Acquisition-kode
ZTD	Zenith Total Delay, total forsinkelse i senit
ZHD	Zenith Hydrostatic Delay, hydrostatisk forsinkelse i senit
ZWD	Zenith Wet Delay, våt forinkelse i senit
IWV	Integrated Water Vapor
PPP	Precise Point Positioning
JPL	Jet Propulsion Laboratory
GIPSY-OASIS	GPS-inferred Positioning SYstem and Orbit Analysis Simulation Software
IERS	International Earth Rotation Service
IGS	International GPS Service
calteck	California Institute of Technology
ALADIN	Aire Limitée Adaptation Dynamique Développement International
HARMONIE	HIRLAM ALADIN Regional Meso-scale Operational NWP In Europe
HIRLAM	High Resolution Limited Area Model

1. Innledning

Global positioning system (GPS) er i dag en nøyaktig metode for å beregne posisjon. Metoden baserer seg på tiden et signal bruker fra en satellitt til en mottaker. For å beregne en posisjon er det ulike forsinkelser man må ta hensyn til på veien fra satellitten til en GPS-stasjonen. En av forsinkelsene som oppstår er den gjennom troposfæren. Denne forsinkelsen kan igjen deles opp i to deler; en hydrostatisk og en våt del. For beregning av disse forsinkelsene trengs det så korrekt a priori informasjon om trykket, vanndamptrykket og temperaturen ved stasjonen som mulig. I tillegg har man ulike avbildningsfunksjoner som avbilder signalet fra senit til en gitt elevasjonsvinkel. A priori informasjonen kan hentes fra ulike kilder, som igjen kan påvirke resultatene. Man kan ha a priori informasjon fra globale modeller eller numeriske værmodeller slik som ECMWF (European Centre for Medium-Range Weather Forecast) og AROME-MetCoOp (Applications and Research to Operations at Mesoscale - Meteorological Co-operation on Operational). Det er svært viktig med nøyaktig informasjon om denne forsinkelsen for å estimere posisjon og våt forsinkelse. I tillegg til beregning av posisjon kan denne forsinkelsen utnyttes til meteorologiske data-assimileringer da denne forsinkelsen, spesielt den våte delen, kan være nyttig for værprognoser.

Det er gjort mange studier for å finne optimal beregning av forsinkelsene: Niell [1] foreslo i 1996 en modell kalt Niell mapping function. I 2003 kom Boehm og Schuh [2] med en forbedret avbildningsfunksjon basert på den numeriske værmodellen ECMWF, Vienna Mapping Function (VMF). I 2006 kom en oppdatert versjon av denne, VMF1, som igjen viste forbedringer. I dag er det VMF1 som er anbefalt å bruke dersom man har tilgang til numeriske værmodeller. Global Mapping Function (GMF) og Global Pressure and Temperature modell (GPT) kom som et alternativ til VMF1 også i 2006, og ble regnet som en god backupløsning dersom numeriske værmodeller ikke skulle være tilgjengelige [3]. Lager et. al. [4] beskrev i 2013 en forbedret versjon av GMF/GPT, GPT2, som forbedret årlig og semiårlig høydeamplitude for posisjon med opp til 40%. Mest mulig korrekt informasjon om trykk, temperatur og vanndamptrykk kan en også finne fra andre værmodeller. For eksempel bruker Meteorologisk Institutt (MET) den numeriske værmodellen modellen AROME-MetCoOp, som har et tettere grid enn ECMWF og kan gi et mer nøyaktig resultat.

I denne oppgaven sees det på hvordan de ulike metodene VMF1, GPT2 og MET-data (AROME), som alle har ulik a priori informasjon om trykk, temperatur og vanndamptrykk påvirker beregning av den totale forsinkelsen gjennom troposfæren. VMF1 og GPT2 har i tillegg delvis forskjellig informasjon knyttet til avbildningsfunksjonene. I figur 1.1 er det illustrert at a priori informasjon fra ulike kilder, samt ulike avbildningsfunksjon for GPT2 og VMF1, brukes til å beregne total forsinkelse i senit. Det er un-



Figur 1.1.: Figuren viser en skjematisk oversikt over oppgaven. Her hentes a priori informasjon om trykk, temperatur og vanndamptrykk fra ulike kilder; GPT2, ECMWF og data direkte fra MET (AROME-MetCoOp). Deretter estimeres total forsinkelse i senit (ZTD).

dersøkt hvordan dette påvirker vertikal posisjon, samt hvilken metode som egner seg for bruk i data-assmilering hos MET. For å se om MET også kan bruke total forsinkelse fra GPS til værprognoser ble metoden VMF1-ultra testet. VMF1-ultra skiller seg fra VMF1 ved at den har såkalte ultra-rapid bane-og klokkeprodukter, som oppdateres hver 2.time, slik at analysen kan skje i nær-sanntid. Det ble oppdaget underveis i analysen at MET-dataene var påvirket av en høydeforskjell mellom AROME-modellen og korrekt stasjonshøyde, slik at det i tillegg ble kjørt 2 andre analyser med ulike måter å korrigere høyden på; MET-standard og MET-mellom. Begge disse metodene baserer seg på en standard atmosfæremodell. Atmosfærisk loading ble også inkludert som en tilleggsoppgave for undersøkelse av vertikal posisjon for å se hvordan dette påvirket resultatene. Analysen er kjørt for 217 stasjoner i Norge for perioden november-desember 2015. Grunnen til valget av denne perioden er at det i disse månedene generelt var mye nedbør og det var blant annet det en storm på sørvestlandet rundt 4.-7.desember, som gjør det interessant å se i hvilken grad været påvirker resultatene. Stasjonene Stavanger, Trysil, Tromsø, Valhall og Lofthus er valgt som representative stasjoner som er sett nærmere på.

Grunnet begrenset tid er hovedfokuset lagt på den geodetiske teorien og resultatene.

2. Teori

2.1. Observasjonsligning

Satellittnavigasjon er basert på elektromagnetiske bølger og i GPS sendes det ut signaler på to frekvenser: $L_1 = 1575,42$ MHz og $L_2 = 1227,6$ MHz [5]. Begge frekvensene kan ha ulike modulasjoner, altså ulike signaler i samme frekvens. I dag har frekvens L_1 to modulasjoner; P-kode og C/A-koden. P-koden er den mest presise, men er styrt av militære og kan derfor ikke brukes av uautoriserte personer, mens den mindre presise C/A-koden er åpen for sivil bruk. L_2 båndet inneholder kun én modulasjon, P-koden.

For å få nøyaktig beregning av posisjon må både kode- og fasemåling beregnes, der kodemålinga gir en nøyaktighet på m-nivå, mens fasemålingen måler avstanden ved å se på fasen mellom signalet i satellitten og i mottakeren og gjør at man får en nøyaktighet ned mot cm [6, side 252–258].

Avstanden mellom satellitt og mottaker for kodemålingen kalles pseudorange og er gitt ved forskjellen mellom klokkene til mottaker og satellitt multiplisert med lysfarten i vakuum.

$$P = c(t_r - t^s) + c \cdot dt_r - c \cdot dt^s \quad (2.1)$$

der c er lysets hastighet, t_r er tid avlest i mottakerklokke når signalet mottas og t^s er tid ved satellitten når signalet sendes ut. dt_r og dt^s er klokkefeilene i henholdsvis mottaker og satellitt [6]. Relativistiske effekter er ikke tatt hensyn til i (2.1).

Fordi de elektromagnetiske bølgene som sendes brytes gjennom atmosfæren og skaper en forsinkelse må dette inkluderes i ligningen. Ligningen blir da

$$P = R + c \cdot dt_r + A + c \cdot dt^s + \epsilon_R, \quad (2.2)$$

$$R = \sqrt{(x^s(t^s) - x_r(t_r))^2 + (y^s(t^s) - y_r(t_r))^2 + (z^s(t^s) - z_r(t_r))^2} = c(t_r - t^s) \quad (2.3)$$

der R er avstanden mellom satellitt og mottaker dt_r er klokkefeilen til mottaker, A er atmosfæreforsinkelsen, dt^s er klokkefeil i satellitten og ϵ_R er et støyledd [6, side 252–258].

Den atmosfæriske forsinkelsen kan i geodetisk sammenheng deles inn i to deler; ionosfæren og troposfæren. Fordi gassene i troposfæren er en miks av uladete partikler, kalles

denne en nøytral del av atmosfæren. Troposfæren i denne sammenhengen kan mer riktig kalles den nøytrale atmosfære og strekker seg opp til hele 50 km, som da også inkluderer tropospausen og stratosfæren [7]. Når det snakkes om troposfæreforsinkelsen senere er dette altså det samme som forsinkelsen gjennom den nøytrale atmosfæren. Ionosfæren på den annen side, inneholder store mengder elektroner og ioner, som oppstår som følge av blant annet stråling fra sola. Dette gir en forsinkelse gjennom atmosfæren som er frekvensavhengig. En kan dermed lage en ligning for koden der atmosfæreforsinkelsen er splittet:

$$P = R + c(dt_r + dt^s) + Z + I_i + \epsilon_R \quad (2.4)$$

der Z og I er henholdsvis troposfæreforsinkelse og ionosfæreforsinkelse.

En tilsvarende ligning er gitt for fasen

$$L = R + c(dt_r + dt^s) + Z - I_i + \epsilon_R \quad (2.5)$$

Det er vanlig å lage en ionosfærefri lineærkombinasjon for å kunne fjerne ionosfæreforsinkelsen. Dette kan gjøres fordi ionosfæren er ladet og derfor frekvensavhengig som gjør at vi, fordi vi har 2 frekvenser, kan fjerne denne støyen ved å lage en ionosfærefri lineærkombinasjon. Dersom man i tillegg antar at vi bruker samme mottaker- og satellittklokke for kode og fase får vi at den ionosfærefrie lineærkombinasjonen er gitt i ligningene

$$P_3 = R + \delta_{orb} + (c \cdot dt_r + b_{P_3}^r) - (c \cdot dt^s + b_{P_3}^s) + Z + \epsilon_{P_3}, \quad (2.6)$$

$$L_3 = R + \delta_{orb} + (c \cdot dt_r + b_{P_3}^r) - (c \cdot dt^s + b_{P_3}^s) + Z + B_3 + \epsilon_{L_3} m, \quad (2.7)$$

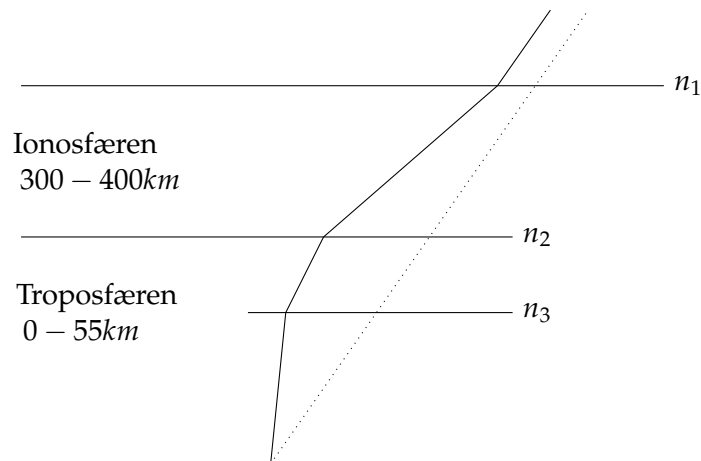
der

$$B_3 = -\lambda_3 N_3 + (b_{L_3}^r - b_{P_3}^r) - (b_{L_3}^s + b_{P_3}^s)$$

for henholdsvis kode og fase. Her har vi også inkludert satellittens banefeil, δ_{orb} [m] og bias-leddene $b_{P_3}^r$, $b_{P_3}^s$, $b_{L_3}^r$ og $b_{L_3}^s$. Disse bias-leddene kalles observasjonsavhengige mottaker- og satellittbiaser. For utledning av den ionosfærefrie lineærkombinasjonen, se tillegg B.

2.2. Troposfæreforsinkelse

Vi har nå sett at observasjonsligningen til beregning av posisjon ved bruk av GPS inneholder forsinkelsen gjennom atmosfæren. De elektromagnetiske signalene fra satellitten som forplanter seg gjennom atmosfæren antas å være en av feilene som oppstår i GPS



Figur 2.1.: Figuren viser hvordan en stråle brytes ulikt gjennom lag med ulik brytningsindeks, og følger av fermats prinsipp. Det er dette som skjer når signalene fra satellitten går gjennom atmosfæren. Den heltrukne linja er den egentlig veien til signalet, mens den stiplede linja gir geometrisk avstand [8].

posisjonering. Det er derfor viktig å ha gode korreksjoner for dette. Signalene forplanter seg ulikt gjennom de ulike atmosfærelagene, fordi de har ulik brytningsindeks, som vist i figur 2.1. Dette er kjent som Fermats prinsipp: En lysstråle som beveger seg fra et punkt til et annet vil alltid velge veien som gjør at den kommer raskest frem til målet.

Snells lov følger av dette prinsippet og gir:

$$n_1 \sin \theta_1 = n_2 \sin \theta_2.$$

Brytningsindeksen er gitt ved $n = \frac{v}{c}$ der v er hastigheten til signalet og c lyshastigheten. Snells lov kan derfor beskrives som

$$n_1 v_1 = n_2 v_2 = c.$$

Atmosfæriske forsinkelsesbidraget kan dermed beskrives på følgende måte,

$$\Delta Z_a = \int_S ds \cdot n(s) - \int_G ds \quad (2.8)$$

der S er den virkelige (optiske) veibanen, mens G er den geometriske avstanden (her er $n = 1$) [9].

En kan av praktiske årsaker introdusere følgende, fordi n er så nær 1;

$$N = (n - 1) \cdot 10^6$$

dette igjen som gitt av [10]

$$N = k_1 \frac{p_d}{T} + k_2 \frac{e}{T} + k_3 \frac{e}{T^2} \quad (2.9)$$

der p_d [hPa] er partielltrykket til tørre bidrag, $p_d = p(\text{total trykk}) - e(\text{vanndamptrykk})$ [8], e [hPa] er det partielle vanndamptrykket og T [K] er absolutt temperatur [11]. Koeffesientene k_1 , k_2 og k_3 løses empirisk fra laboratoriske eksperimenter. Tallet k_1 er avhengig a mengden CO_2 i atmosfæren som den 04. mai 2016 lå på 407 ppm. De mest vanlige verdiene for koeffesientene er gitt i tabell 2.1. Her er k_1 beregnet med utgangspunkt i 300 ppm og den vil dermed avvike noe fra dette [12].

Koeffisient	Beste tilgjengelige estimat
k_1 [K/hPa]	77.691 ± 0.013
k_2 [K/hPa]	71.97 ± 10.5
k_3 [K ² /hPa]	375406 ± 3000

Tabell 2.1.: Tabellen viser koeffesientene k_1 , k_2 og k_3 som er løst empirisk. Her har man regnet k_1 basert på en CO_2 -konsentrasjon på 300 ppm [11].

I den øvre delen av atmosfæren beveger signalet seg gjennom ionosfæren. Dette gir den største forsinkelsen. Denne forsinkelsen er imidlertid lett å korrigere for, da ionosfæren er frekvensavhengig. Som nevnt i seksjon 2.1 kan vi bruke to frekvenser til å lage en ionosfærefri lineærkombinasjon. Dette skal ikke diskuteres mer i denne oppgaven. Lengre ned i atmosfæren finner vi troposfæren. Det er dette laget som inneholdet vanndamp og derav været som vi oppfatter her på jorda. Troposfæren inneholder omlag 80% av all masse i atmosfæren og så og si all vanndampen [11]. Denne delen kan i geodetisk sammenheng deles inn i to deler: En hydrostatisk (*zenith hydrostatic delay*, ZHD) og en våt del, (*zenith wet delay*, ZWD). Det vil si

$$\Delta Z(\alpha, \epsilon) = \Delta Z_h^Z m(\epsilon)_h + \Delta Z_w^Z m(\epsilon)_w, \quad (2.10)$$

der $\Delta Z(\alpha, \epsilon)$ er totale forsinkelsen, ϵ er elevasjonsvinkelen, α er asimutvinkelen, ΔZ_h^Z og ΔZ_w^Z er forsinkelsen i senit for henholdsvis den hydrostatiske og våte delen. $m(\epsilon)_h$ og $m(\epsilon)_w$ er avbildningsfunksjoner (*mapping functions*), som avbilder forsinkelsen fra senit til riktig elevasjonsvinkel [9].

Hydrostatisk forsinkelse kommer hovedsaklig som følge av brytning med oksygen (O_2) og nitrogen (N), som utgjør største del av gassene i atmosfæren. Den ikke-dipolariserte vanndampen regnes også som en del av den hydrostatiske forsinkelsen, og er grunnen til at den kalles en hydrostatisk forsinkelse istedet for tørr forsinkelse [13].

På grunn av asymmetri i asimut bør gradientene også tas hensyn til:

$$\Delta Z(\alpha, \epsilon) = \Delta Z_h^Z m(\epsilon)_h + \Delta Z_w^Z m(\epsilon)_w + m_g(\epsilon) [G_N \cos \alpha + G_E \sin \alpha],$$

hvor α er asimut vinkelen, G_N og G_E er gradientene i nord- og østlig retning og m_g er gradientens avbildningsfunksjon. Denne er ofte definert

$$m_g(\epsilon) = \frac{1}{\sin \epsilon \tan \epsilon + C}.$$

Den hydrostatiske delen gir den største forsinkelsen, men igjen er denne også relativt lett å estimere og er gitt ved formel

$$\Delta Z_h^Z = 0.00227768 \frac{P}{(1 - 0.00266 \cos(2\varphi) - 0.00000028 \times H)}, \quad (2.11)$$

der P er det totale trykket gitt i hPa ved stasjonen, φ er breddegraden i radianer og H er ortometrisk høyde gitt i m [14].

Den våte delen er på den andre siden ikke veldig stor, omlag 10 % av den totale refraksjonen, men vanskelig å estimere fordi den varierer så mye [6, side 314–316]. Av formel (2.8) og (2.9) får vi følgende formel for beregning av våt forsinkelse i senit:

$$\Delta Z_w^Z = 10^{-6} \int_w \left(k_2 \frac{e}{T} + k_3 \frac{e}{T^2} \right). \quad (2.12)$$

Formelen vi bruker for å beregne a priori våt forsinkelse i senit (ZWD) med MET-data er:

$$\Delta Z_{0w}^Z = \left(\frac{1255}{T} + 0.05 \right) \frac{0.0022768 \times e}{(1 - 0.00266 \cos(2\varphi) - 0.00000028 \times H)} \quad (2.13)$$

der T er temperatur i Kelvin og e er partiell vanndamptrykk i hPa [15]. Nettopp på grunn av den store variasjonen kan den våte delen estimeres ved hjelp av et kalmanfilter, der den regnes som en stokastisk, altså tidsvarierende, parameter og går under betegnelsen integrert hvit støy [16]. Mer om hvordan denne estimeres ved hjelp av kalmanfiltrering i GIPSY står i avsnittet om PPP, seksjon 2.3.

Den totale forsinkelsen i senit (*Zenith Total Delay, ZTD*), er undersøkt mye i denne oppgaven her og er gitt ved

$$ZTD = ZHD + ZWD$$

Avbildningsfunksjonene m_h og m_w kan defineres på forskjellige måter, men den vanligste formen som benyttes i dag er

$$m(\epsilon) = \frac{1 + \frac{a}{1 + \frac{b}{1+c}}}{\sin(\epsilon) + \frac{a}{\sin(\epsilon) + \frac{b}{\sin(\epsilon)+c}}} \quad (2.14)$$

der a , b og c er konstanter som kan beregnes ved å bruke ulike avbildningsfunksjoner [9]. Som en ser i likning (2.14) er det 3 parametre som må beregnes for både hydrostatisk og våt del, og det er ulike metoder for å beregne disse. I tillegg til disse koeffisientene er det viktig med god a priori informasjon om trykk, temperatur og vanndamptrykk og det er interessant å se på data direkte fra MET, fordi disse dataene kan gi en mer nøyaktig informasjon.

2.2.1. Integrated water vapor

For å se på fuktighet i atmosfæren som kan brukes i meteorologisk sammenheng kan man regne om ZWD til integrated water vapor (IWV). Utregningen av IWV følger [17] og er gitt ved

$$\text{IWV} = \text{ZWD} \times \Pi \quad (2.15)$$

der Π er

$$\Pi^{-1} = 10^{-6} \rho_{H_2O} R_v \left(\frac{k_3}{T_m} + k'_2 \right),$$

og

$$k'_2 = k_2 - mk_1 = (22.1 \pm 2.2) \text{KhPa}^{-1},$$

$\rho_{H_2O} = 10^3 \text{kgm}^{-3}$ er tettheten til vann i væskeform, $R_v = 461.5 \text{Jkg}^{-1}\text{K}^{-1}$ er den spesifikke gasskonstanten for vanndamp og $m = \frac{M_w}{M_d}$ er forholdet mellom molar masse for henholdsvis tørr og våt luft. Koeffisientene k_1 , k_2 og k_3 er gitt i tabell 2.1.

Temperaturen T_m er et vektet gjennomsnitt for atmosfæren

$$T_m = \frac{\int \left(\frac{e}{T} \right) dz}{\int \left(\frac{e}{T^2} \right) dz}.$$

Tidligere har en antatt et tilnærmet lineært forhold til temperatur ved overflaten og er gitt av formel

$$T_m \approx 70.2 + 0.72T_s$$

Denne formelen har en RMS på 4.7 K, relativ feil på mindre enn 2% og kan derfor antas å være en grei antagelse [17].

Emardson og Derks(2000) [18] forbedret ligning (2.15) ved hjelp av kvadratregresjon for 120 000 radiosondeprofiler fra 30 stasjoner i Europa og kom frem til følgende formel

$$\text{IWV} = \frac{\text{ZWD}}{(a_0 + a_1\Delta T + a_2\Delta T^2)} \quad (2.16)$$

der $\Delta T = T_s - T_{av}$, T_s er temperatur ved overflaten, og er i denne oppgaven gitt fra AROME-dataene til MET. T_{av} , a_0 , a_1 og a_2 er konstantene [19]:

$$\begin{aligned} T_{av} &= 283.49K, \\ a_0 &= 6.548m^3kg^{-1}, \\ a_1 &= -1.78 \times 10^{-2}m^3kg^{-1}K^{-1}, \\ a_2 &= -2.2 \times 10^{-5}m^3kg^{-1}K^{-2}. \end{aligned}$$

Det er denne metoden som er brukt for å beregne IWV i denne oppgaven.

Data fra MET er ikke korrigert for høyde, det gjelder også for temperaturen. I vintermånedene november og desember kan det i tillegg skje inversjon, slik at dette kan påvirke resultatene for omregningen av våt forsinkelse til integrert vanddamp. Det vil ikke gi veldig store utslag, og en forskjell på 10 grader utgjør ca. 0.6 mm i IWV.

2.2.2. Vienna Mapping function (VMF1)

Vienna mapping function (VMF1) bruker den numeriske værmodellen European Centre for Medium-Range Weather Forecast (ECMWF) som grunnlag for å bestemme koeffesientene i avbildningsfunksjonen. Avbildningsfunksjonen brukt her er en oppdatert versjon, kalt VMF1. Den baserer seg på 7488 profiler - 156 gridpunkter i 12 måneder med målinger 4 ganger per dag, altså med et intervall på 6-timer [16]. Det brukes 10 start-elevasjonsvinkler og lages profiler med 23 nivåer fra 1000 hPa til 1 hPa. VMF1 er basert på full ray-tracing, som vil si at en følger signalet fra stasjonen til øvre del av troposfæren. Ved hvert nivå finner man verdier for høyde, total trykk, temperatur og vanddamp [16]. Koeffesientene a , b og c er tilpasset de 10 diskrete avbildningsfunksjonene for hver profil (de 10 ulike elevasjonsvinklene). Dette gir en gjennomsnittlig b -koeffesient på 0.0029. Denne blir så holdt fast og ut i fra dette blir a og c tilpasset ved hjelp av minste kvadratsmetode. Koeffesienten c blir estimert ved formel

$$c_h = c_0 + \left[\left(\cos \left(\frac{\text{doy} - 28}{365} \times 2\pi + \psi \right) + 1 \right) \times \frac{c_{11}}{2} + c_{10} \right] \times (1 - \cos \varphi), \quad (2.17)$$

der doy er day of year, ψ sier om det er nordlig eller sørlig halvkule ($\psi = 0$ er nordlig og $\psi = 1$ er sørlig), koeffesientene c_0 , c_{11} og c_{10} er gitt i tabell 2.2 og φ er breddegraden.

For den våte delen er koeffesientene b_w og c_w estimert på samme måte som i NMF, dette kan gjøres fordi forsinkelsen kun er rundt 10 cm. NMF, Niells avbildningsfunksjon, tar utgangspunkt i klimatiske forhold fremfor direkte meteorologi for stasjonen fordi den

Halvkule	c_0	c_{10}	c_{11}
Nordlig	0.062	0.001	0.005
Sørlig	0.062	0.002	0.007

Tabell 2.2.: Tabellen viser koeffisientene c_0 , c_{10} , c_{11} som brukes for å beregne den hydrostatisk c -koeffisienten til VMF1 for nordlig og sørlig halvkule [11].

braker globale modeller i sin beregning. Den våte koeffisienten er basert på en lineær interpolasjon som kun er avhengig av breddegraden [16]. Koeffisienten a_w blir bestemt for hver enkelt stasjon, på samme måte som VMF.

Det er i dag anbefalt å bruke avbildningsfunksjoner basert på numeriske værmodeller, slik som VMF1 [3]. I følge Boehm et. al.(2006) [3] er en global avbildningsfunksjon (GMF) et godt alternativ dersom man ikke har tilgang til VMF1, men den vil gi dårligere resultater på presisjon over et kort tidsrom. For studier gjort ved sammenligning av GPT som input fremfor ECMWF i VMF1 ble det funnet av Steigenberger et. al.(2009) [20] at ECMWF ga best resultater dersom man inkluderer atmosfærisk loading(*atmospheric loading*), som blir studert mer i avsnitt 2.2.5.

2.2.3. Global Pressure and Temperature model (GPT2)

Dersom VMF1 ikke er tilgjengelig for et sted er en global modell et alternativ. GPT2 er en global trykk- og temperaturmodell, i en oppdater versjon, som baserer seg på et gjennomsnitt av globale temperatur, trykk og vanndamptrykk over en viss tidsperiode.

GPT2 gir ut temperatur, trykk, vanndamptrykk, adiabatisk temperaturendring og avbildningsfunksjons koeffesienten a (for både hydrostatisk og våt) for alle stasjoner. Den har en horisontal oppløsning på 5 grader og bruker månedlige data, kalt ERA-interim, hentet fra ECMWF over en periode på 10 år (2001-2010) [4]. ERA- interim er en global atmosfærisk reanalyse fra 1979 som blir kontinuerlig oppdatert i nåtid. I studier gjort av Lagler et al. [4], vises det at GPT2 gir bedre resultater for årlig/semi-årlig amplitudforskjell for stasjonshøyden enn ved bruk av GMF/GPT.

Metoden som brukes for å beregne avbildningsfunksjonen a er den samme som for VMF1, men basert på annen informasjon. Konstantene b og c er de samme som i VMF1. Koeffisientene fra GPT2 kan dermed brukes direkte i VMF1 og sees således ikke på som en egen avbildningsfunksjon, men mer som en annen metode å hente input data fra.

Som nevnt tidligere har Steigenberger et. al.(2009) [20] vist at ZHD som baserer seg på data fra GPT og bruker VMF1 som avbildningsfunksjon kan gi bedre resultater enn VMF1 med ECMWF som input dersom en ser på en tidsserie for høyden for en periode og man ikke tar hensyn til atmosfærisk loading. Det er derfor interessant å se på forskjellen mellom GPT2 og VMF1 fordi de baserer seg på ulike data og om forskjellene forandres dersom atmosfærisk loading inkluderes. Informasjon om trykk, temperatur og vanndamptrykk kan man også hente fra MET-data.

2.2.4. MET-data: AROME-MetCoOp

For å få et tettere grid og en annen a priori informasjon om stasjonene er det foreslått å bruke data direkte fra meteorologisk institutt (MET). Disse dataene er gitt av modellen AROME-MetCoOp (Applications and Research to Operations at Mesoscale), som er en versjon av Harmonie-AROME som blir brukt i MetCoOp NWP (Meteorological Co-operation on Operational NWP (Numerical Weather Prediction)). MetCoOp er et svensk-norsk meteorologisk samarbeid for å ha en felles numerisk værmodell. Denne modellen bruker den ikke-hydrostatiske modellen Harmonie og fysikken til modellen AROME.

AROME-MetCoOp bruker en 2.5 km grid og er en relativt ny numerisk værprediksjon med oppstart i 2008 fra Frankrike [21]. Fysikken rundt overflatemodellen, kalt SURFEX, er som følger: Hver gridboks er delt inn i 4 mulige lag: land, by, hav og innsjøer (og elver). De ulike lagene har ulike mikroklimaer basert på hvilken av de overnevnte kategoriene overflaten er. På denne måten får man dekket atmosfæriske tilstander og tilstander ved havet. SURFEX estimerer temperatur ved 2 meter og relativ fuktighet ved 2 meter for hvert tidsskritt med en algoritme kalt surface boundary layer. Dataene som brukes i denne oppgaven er lufttemperatur ved 2 meter (*air temperature 2m*), relativ fuktighet ved 2 meter (*relative humidity 2m*) og trykk ved overflaten (*surface air pressure*) som kan settes inn i ligningene (2.11) for a priori hydrostatisk del og (2.12) for den våte delen. I ligning (2.12) brukes vanndamptrykket, ikke relativ fuktighet. Her er dette regnet om ved hjelp av formelen

$$e = SVP \times RH, \quad (2.18)$$

der *SVP* (*Saturated Vapor Pressure*) er metningstrykket og er gitt av

$$SVP = c_1 = \exp\left(\frac{c_2 \times T}{c_3 + T}\right).$$

Konstantene c_1 , c_2 og c_3 er gitt i tabell 2.3 og T er temperaturen i celsius [22]. I beregningene har vi antatt at vannet ikke er is, da det ikke er mulig å vite hvilken tilstand vannet befinner seg i, noe som kan være en mulig feilkilde.

Fase	Temperatur[C]	c_1 [hPa]	c_2	c_3 [K]
Vann	-50.9 – 0	6.10780	17.84362	245.425
Vann	0 – 100.9	6.10780	17.08085	234.175

Tabell 2.3.: Tabellen viser koeffisientene c_1 , c_2 , c_3 som brukes for å beregne metningsstrykket til vanndamp, *SVP*. *SVP* kan igjen brukes til å regne vanndamptrykket i ligning 2.18, som brukes til beregning av *ZWD* i ligning 2.12 [22].

I studier gjort av Koltzow et. al(2012) [23] der HARMONIE-AROME er sammenlignet med ECMWF, ble det funnet at trykket ved havnivå gir dårligere resultater for

HARMONIE-AROME sammenlignet med ECMWF. HARMONIE-AROME var på sin side best på temperatur og nedbør.

2.2.5. Atmosfærisk loading

Underveis i analysen viste det seg at VMF1 ga mye bedre resultater langs kysten i Sør-Norge, sammenlignet med andre steder. GPT2 ga også bedre resultater i høyderetningen for en del innlandsstasjoner. Tidligere studier gjort av Steigenberger et. al. [20] der VMF1 (ECMWF) og VMF1 (GPT2) er sammenlignet viser lignende resultater. Her gikk studiene også ut på å korrigere for såkalt atmosfærisk loading (*atmospheric loading*), noe som forbedret resultatene for stasjoner som ikke lå langs kysten. Derfor blir en del av oppgaven også å korrigere for atmosfærisk loading og se om resultatene blir bedre av det.

Atmosfærisk loading er en trykklast som følge av forflytning av luftmasse rundt på jorda. Denne loadingen vil, dersom det blir påvirket av store trykksystemer, kunne ha en påvirkning på jordas overflate og dermed føre til en vertikal endring av posisjoner for en del stasjoner. Denne atmosfæriske loadingen kan det derfor være lurt å korrigere for. Van Dam har tidligere vist at atmosfærisk loading ofte er systematisk høyere ved høyere til midlere breddegrader fordi det ofte er her de synoptiske trykksystemekstremene er størst [24]. Med synoptisk trykksystem menes systemer med en horisontal skala som er større en 1000 km. Ved kysten og havet blir atmosfærisk loading minimal på grunn av et fenomen kalt inverted barometric effekt. Denne effekten gjør at havet motvirker trykkvariasjoner ved å motbalansere den, slik at totaltrykket ved havets bunn forblir konstant. Fordi inverted barometric effekt fra havet ikke er så stor på innlandet blir derfor atmosfærisk loading ved disse stasjonene større [24]. I den atmosfæriske loadingen brukt i denne oppgaven som er hentet fra nettsiden International Mass Loading Service: Atmospheric Pressure Loading [25] tar de ikke hensyn til såkalt tide-atmosfære, som er en variasjon som skjer en gang i døgnet.

I oppgaven ble modellen GEOS-FPIT brukt til å finne den atmosfæriske loadingen for november-desember 2015. Modellen er en operasjonell semi-frosset numerisk værmodell og dekker tidsrommet siden 1. januar 2000 med oppdateringer for hver 3. time. Oppløsningen er 0.50 grader \times 0.625 grader \times 72 lag. Denne modellen passer best for analyser som krever kort ventetid - noe som ble tilfelle i denne oppgaven.

2.3. Presis enkeltpunkts stedfesting (PPP)

P्रेसis enkeltpunkts stedfesting med GPS er den enkleste metoden for stedfesting. I denne metoden kreves det kun målinger fra én GPS-mottaker og alle synlige satellitter. Fra GPS-satellittene sendes det i navigasjonsmeldingen ut informasjon om satellittens posisjon i rommet, altså baneparametre, og korreksjon for klokken ombord i satellitten.

Nøyaktigheten til disse er lave på grunn av begrensninger i dataformatet, oppdaterings-
tid og at GPS referansestasjonene er relativt spredd.

Det ble i 1994 etablert et sivilt bakkebasert nettverk av GPS referansestasjoner, som var
et samarbeid mellom geodetiske institusjoner fra hele verden. Denne tjenesten ble kalt
for IGS (International GPS Service) og hadde som formål å beregne presise baner til
alle GPS-satellitter. Disse beregningene ble foretatt av flere analysesentre med forskjel-
lige GPS-analyseprogrammer. IGS beregner i dag daglige presise satellittbaner, satellitt-
klokkekorreksjon og jordrotasjonsparametre.

Tilgjengeliggjøringen av presise satellittbaner og - klokkekorreksjoner åpnet for presis
punkt stedfesting. Denne målemetoden bruker alle synlige GPS-satellitter som tas imot
av en GPS-mottaker. Det mottas observasjoner på to frekvenser, altså to kode- og to
fasemålinger, noe som gir oss muligheten til å minimalisere det største feilbidraget i
en GPS-analyse, nemlig radiosignales forsinkelse gjennom ionosfæren. Ved å lage en io-
nosfærefri lineærkombinasjon av kode-og fasemålingen fjernes 1.ordensleddene i ionos-
færeforsinkelsen. Siden vi henter presise baneparametre og satelittklokkekorreksjoner
fra IGS, eller et av analysesentrene til IGS, er det kun mottakerens posisjon, mottakerens
klokkefeil, flerfasetvetydigheten og den våte delen av troposfæreforsinkelsen som må
estimeres [26].

Analysene i denne oppgaven er utført med å benytte programvaren GIPSY-OASIS (GPS-
inferred Positiong SYstem/ Orbit Analysis Simulation Software) som er utviklet ved
calteck/NASA Jet propulsion Labratory (JPL). Denne programvaren benytter udiffere-
nsierte målinger. JPL er et av IGS analysesentre og benytter GIPSY til å beregne presise
satellittbaneparametre, jordrotsajonsparametre og satellittklokkekorreksjoner.

For å være sikkert på at løsningen vår er konsistent med de modellene som er benyttet
under beregningen av satelittbanene og klokkekorreksjonene benyttes disse parametre-
ne fra JPL. De mest presise produktene er tilgjengelige etter ca 14 dager og kan derfor
benyttes til prosessering i ettertid. Ved nærsanntidsanalyse kan man benytte ultra rapid
produkter, som kommer etter ca. 2 timer [27]. I denne oppgaven har vi også undersøkt
kvaliteten på estimert posisjon og troposfære ved å benytte ultra rapid produkter.

Ligningen som blir igjen å estimere blir da

$$P_3 = \rho + (cdt_r + b_{P_3}^r) + Z + \epsilon_{P_3} \quad (2.19)$$

for koden og

$$L_3 = \rho + (cdt_r + b_{L_3}^r) + Z + B_3 + \epsilon_{L_3} \quad (2.20)$$

for fasen.

For å estimere disse parametere brukes det i GIPSY et kalmanfilter. Dette er en optimal
tilstandsestimator som gjelder for stokastiske prosesser. En stokastisk parameter kan

bestemmes gjennom statistiske metoder og tillates å variere med tiden [26]. Slike stokastiske parametre kan også tillates å variere fra tidspunkt til tidspunkt. Vanndampen regnes som en slik stokastisk prosess og modelleres som integrert hvit støy.

2.3.1. Kalmanfilter

Her gis det en kort innføring i algoritmene som brukes i kalmanfilteret med fokus på GPS og da spesielt estimeringen av troposfæreforsinkelsen. Teorien om kalmanfilteret er i hovedsak hentet fra Kristiansen(2015) [26]. Kalmanfilter er en metode å estimere parametre på i PPP og er en rekursiv algoritme. Figur 2.2 viser en enkel skisse som illustrerer rekursiviteten; en har en startverdi for tilstandene (initialisering) som tidsoppdateres (prediksjon), som deretter korrigeres ved hjelp av en måleoppdatering. Resultatene av måleoppdateringen blir så tidsoppdatert, slik at prosessen går i en løkke.

Man tar utgangspunkt i en observasjonsmodell der z_k er observert minus utregnet verdi, og er gitt ved

$$z_k = A_k x_k + v_k. \quad (2.21)$$

der observasjonsmodellen er gitt ved et tidspunkt t_k . Tilstandsvektoren x_k beskriver tilstanden til observasjonen. Det er denne vektoren som estimeres og optimaliseres ved et kalmanfilter. A_k kalles designmatrisa og er gitt ved $A_k = \frac{\partial z_k}{\partial x_k}$.

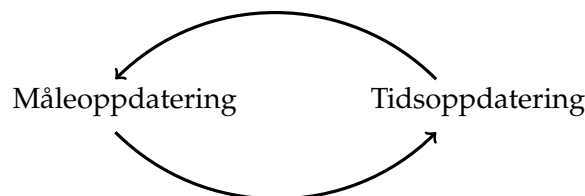
For å kunne ha en tidsoppdatering tas det utgangspunkt i den dynamiske ligningen

$$\dot{x} = F(t)x(t) + G(t)w(t) + L(t)u(t) \quad (2.22)$$

der $F(t)$ er dynamisk systemmatrise, $x(t)$ er tilstandsvektoren, $G(t)$ er prosessstøy-matrisen, $w(t)$ er en støy-vektor, $u(t)$ er et pådrag og $L(t)$ er pådragsmatrisen.

Tidsoppdatering er en prediksjon der prosessstøyen inkluderes og gitt ved den dynamiske modellen

$$\tilde{x}_{k+1} = \Phi(t_{(k+1,t_k)})\hat{x}_k + e_k$$



Figur 2.2.: Figuren viser hvordan kalmanfilteret er rekursivt. Resultater fra prediksjonen blir brukt i korreksjonen, som igjen blir brukt til en ny prediksjon osv.

der \hat{x} er tilstandsvektoren ved tidspunkt k , Φ_{t_{k+1}, t_k} er transisjonsmatrisa som overfører tilstandsvektoren fra tidspunkt k til $k + 1$ og e_k er prosessstøyen som er gitt ved likningen

$$e_k = \int_{\tau=t_k}^{t_{k+1}} \Phi_{t_{k+1}, \tau} G w(\tau) d\tau.$$

Varians-kovariansmatrisa ved tidsoppdatering er gitt ved ligning

$$\tilde{P}_{k+1} = \Phi_{(k+1, k)} \hat{P}_k \Phi_{(k+1, k)}^T + Q_e.$$

Q_e er varians-matrisa til prosess-støyen og er gitt ved

$$Q_e = [\delta e_k, \delta e_k^t] = \int_{\tau=t_k}^{t_{k+1}} \Phi(t_k, \tau) G Q(\tau) G^T \Phi^T(t_k, \tau) d\tau$$

G her er prosessstøy-matrisen gitt i den dynamiske ligningen, (2.22), som normalt er lik identitetsmatrisen. Transisjonsmatrisa $\Phi_{t_{k+1}, t_k} = e^{F\Delta t}$, der F er den dynamiske systemmatrisa gitt i ligning (2.22).

Når det er gjort en prediksjon foretar man deretter en korreksjon. Her finner vi et estimat \hat{x}_k og en kovariansmatrise \hat{P}_k basert på a priori informasjonen fra tidsoppdateringa. Dette kalles en måleoppdatering og er gitt ved

$$\hat{x}_k = \tilde{x}_k + K_k(z_k - A_k \tilde{x}_k)$$

og tilhørende kovariansematrise

$$\begin{aligned} \hat{P}_k &= (I - K_k A_k) \tilde{P}_k, \\ K_k &= \tilde{P}_k A_k^T S_K^{-1}, \\ S &= A_k \tilde{P}_k A_k^T + R_k \end{aligned}$$

der K er kalmanforsterkningen, S kalles innovasjonsmatrisa og er kovariansmatrisa til innovasjonen og R_k er kovariansmatrisa til observasjonene, altså målestøyen.

Det korreksjonen gjør er altså å si i hvilken grad estimatet skal vekte målingene fremfor modellen vår. Er K liten vil målingene være lite vektlagt, mens dersom K er stor vil modellen være vektlagt mindre. Kalmanforsterkningen varierer ettersom prosessstøyen, e_k , og målestøyen v_k varierer. Høy målestøy gir liten K som igjen gjør at modellen blir mest vektlagt og motsatt.

Når denne korreksjonen er gjort, tidsoppdateres den igjen, for så å måleoppdatere igjen, og så videre. Dette er da en rekursiv algoritme som vist i figur 2.2.

Med utgangspunkt i ligningene (2.19) og (2.20) blir tilstandsvektoren i PPP

$$x = \begin{pmatrix} \Delta x_r \\ \Delta y_r \\ \Delta z_r \\ cdt_r \\ dZ_w \\ B_1 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{pmatrix}$$

For tilstandsvektorer som inneholder variabler med ulike egenskaper, slik som kalmanfiltrering i GPS, kan det være mest hensiktsmessig og dele den opp i 3 ulike deler; deterministisk, stokastisk og konstant del for å gjøre estimeringen enklere.

Ligning (2.21) kan da skrives slik

$$z = A \begin{pmatrix} \epsilon \\ p \\ y \end{pmatrix} + v$$

der ϵ er den deterministiske delen, p den stokastiske og y den konstante.

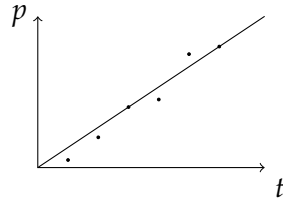
Tidsoppdateringen kan da også deles opp på følgende måte:

$$\begin{aligned} \text{Deterministiske:} & \quad \tilde{\epsilon}_{k+1} = \Phi_\epsilon \hat{\epsilon}_k + \Phi_\epsilon \hat{p}_k + \Phi_\epsilon \hat{y}_k \\ \text{Stokastiske:} & \quad \tilde{p}_{k+1} = M \hat{p}_k + w_k \\ \text{Konstante:} & \quad \tilde{y}_{k+1} = \hat{y}_k = y \end{aligned}$$

Dette, tillegg til innføring av epoch-tilstand gir følgende kalmanfilter for GPS:

$$\begin{pmatrix} \tilde{\epsilon} \\ \tilde{p} \\ \tilde{y} \end{pmatrix}_{k+1} = \begin{pmatrix} I_\epsilon & \Phi_p(t_k) & 0 \\ 0 & M(t_k) & 0 \\ 0 & 0 & I_y \end{pmatrix} \begin{pmatrix} \hat{\epsilon}' \\ \hat{p} \\ \hat{y} \end{pmatrix}_k + \begin{pmatrix} 0 \\ w \\ 0 \end{pmatrix}_k$$

Stasjonene som beregnes her er stasjonære og en kan derfor regne mottakerens koordinater som konstante. Klokkefeil og troposfære regnes som stokastiske, henholdsvis som hvit og integrert hvit støy. Biasene, som er en pr. satellitt som er med i beregningen, regnes også som konstante. I figur 2.3 kan en se hvordan den stokastiske modellen integrert hvit støy oppfører seg over tid. Integrert hvit støy er altså korrelert over tid og vil aldri nå en stasjonær tilstand. Prosessen blir dermed karakterisert av endringen i



Figur 2.3.: Figuren viser den stokastiske modellen integrert hvit støy. Her er parameteren korrelert med tid, da den har en tydelig lineær sammenheng. Vann-damp i atmosfæren regnes nettopp som en slik stokastisk parameter [26].

prosessstøy-variansen, $q\delta t$, i motsetning til hvit støy der variansene er langs diagonalen i kovariansmatrisa til prosessstøyen.

Beregningen av troposfæreforsinkelsen ved PPP gjøres som beskrevet i avsnittet om troposfæreforsinkelse, det er imidlertid ikke alt som kan beregnes her og en resterende forsinkelse blir igjen, da hovedsaklig for den våte delen som estimeres av kalmanfilteret [26].

Det troposfæriske bidraget til kalmanfilteret blir da:

$$dZ = \Delta Z \rightarrow A_K = \frac{\partial z}{\partial dZ} = m_w(\epsilon)$$

Ved hjelp av disse opplysningene kan en finne kalmanfilteret for PPP.

Designmatrisa vil bli følgende, når vi tar utgangspunkt i vektoren gitt for PPP tidligere i avsnittet:

$$A_k = \begin{pmatrix} \frac{(x_r - x^1)}{\rho_r^1} & \frac{(y_r - y^1)}{\rho_r^1} & \frac{(z_r - z^1)}{\rho_r^1} & 1 & m_w & 0 & \cdots & 0 & \cdots & 0 \\ \frac{(x_r - x^1)}{\rho_r^1} & \frac{(y_r - y^1)}{\rho_r^1} & \frac{(z_r - z^1)}{\rho_r^1} & 1 & m_w & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{(x_r - x^i)}{\rho_r^i} & \frac{(y_r - y^i)}{\rho_r^i} & \frac{(z_r - z^i)}{\rho_r^i} & 1 & m_w & 0 & \cdots & 0 & \cdots & 0 \\ \frac{(x_r - x^i)}{\rho_r^i} & \frac{(y_r - y^i)}{\rho_r^i} & \frac{(z_r - z^i)}{\rho_r^i} & 1 & m_w & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{(x_r - x^n)}{\rho_r^n} & \frac{(y_r - y^n)}{\rho_r^n} & \frac{(z_r - z^n)}{\rho_r^n} & 1 & m_w & 0 & \cdots & 0 & \cdots & 0 \\ \frac{(x_r - x^n)}{\rho_r^n} & \frac{(y_r - y^n)}{\rho_r^n} & \frac{(z_r - z^n)}{\rho_r^n} & 1 & m_w & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

Tilstanden, altså posisjon og hastighet til mottaker, regnes som konstante her og gir derfor :

$\tilde{\epsilon}_{k+1} = \hat{\epsilon}_k$ noe som betyr at transisjonsmatrise til posisjonen x, y, z til mottaker er gitt ved identitetsmatrisa og en prosess-støyvariansmatrise, $Q_{XYZ} = 0$. Det samme gjelder for Bias-leddene.

2. Teori

Klokkefeilen og troposfæreforsinkele regnes, som sagt, ut som stokastiske parametre, henholdsvis hvit og integrert hvit støy, og gir:

$$\begin{aligned} \text{Hvit støy:} \quad & \tilde{p}_{k+1} = w_k \\ \text{Integrert hvit støy:} \quad & \tilde{p}_{k+1} = M\hat{p}_k + w_k \end{aligned}$$

M blir her, der diagonalen blir 0 for hvit støy og 1 når parameteren er integrert hvit støy:

$$M = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Prosesstøy variansmatrisa blir her $Q_p = \begin{pmatrix} \sigma_{clk}^2 & 0 \\ 0 & q_{trop}\delta t \end{pmatrix}$.

Dette totalt gir en transisjonsmatrise:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & \dots & 1 \end{pmatrix}$$

Det slik de ulike parametrene i GIPSY estimeres. I denne analysen er integrert hvit støy parameteren $q_{trop}\delta t = 5 \cdot 10^{-8} \frac{\text{km}}{\sqrt{\text{s}}}$.

3. Metode

3.1. Troposfæremodellering med GIPSY PPP

I denne oppgaven er GIPSY-OASIS kjørt for 217 stasjoner fordelt over hele Norge for november og desember 2015. MET-dataene fra AROME-MetCoOp har ikke trykk, temperatur og vanndamptrykk for Svalbard, Færøyene og Bjørnøya, og fjernes derfor for analysen.

Navn	Bane-og klokkeprodukter	A priori	Avbildningsfunksjon
GPT2	JPL final	GPT2	GPT2
VMF1	JPL final	ECMWF (VMF1)	VMF1
VMF1-ultra	JPL Ultra Rapid	ECMWF (VMF1)	VMF1
MET-direkte	JPL final	AROME	VMF1
MET-standard	JPL final	AROME	VMF1
MET-mellom	JPL final	AROME	VMF1

Tabell 3.1.: Tabellen viser en oversikt over metodene som er brukt og hva som er forskjellen mellom dem. Her ser vi at de har ulik informasjon om a priori trykk (P), temperatur (T) og vanndamptrykk (e), avbildningsfunksjoner og JPLs baneprodukter.

Programvaren som blir brukt kalles GIPSY-OASIS og er utviklet av NASA. I tabell 3.1 kan vi se metodene som er brukt i denne oppgaven. For GPT2, VMF1 og MET-datene benyttes JPL final products, som er de mest presise banene. Disse kommer etter ca. 14 dager, og kan derfor kun benyttes til prosessering i ettertid. For VMF1-ultra benyttes JPL ultra-rapid products, som kommer etter ca 2 timer. Disse blir inkludert i analysen for å sjekke om den er god nok til meteorologenes analyse, for å se om dette også kan bli brukt i nær-sanntid, noe som er svært aktuelt for meteorologene. MET-dataene, VMF1 og VMF1-ultra bruker alle VMF1 som avbildningsfunksjon. MET-data bruker AROME-MetCoOp til beregning av a priori informasjon om ZHD og ZWD.

GIPSY skal være konsistent med International Earth Rotation Service (IERS). Det er viktig at modellen er konsistent med referanserammen som er nå, da dette setter standarden for hva som brukes generelt i verden. Uten denne standarden kan man ikke sammenligne resultater med andre, og det blir vanskelig å skulle gjenta forsøk ved senere tidspunkt.

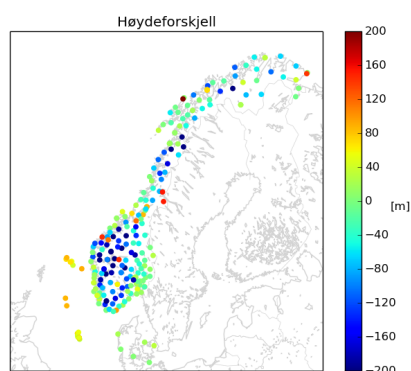
Tabell 3.2 viser en oversikt over modellene og standardene som er brukt i denne GIPSY-analysen.

3. Metode

Modell brukt	
GIPSY versjon	6.3
Referanseramme	IGb08
Elevation Angle Cutoff	7 grader
Samplingsfrekvens	5 min
Elevation dependent weighting	Ja, SQRTSIN: $\sigma^2 = \frac{1}{\sqrt{\sin(\epsilon)}}$
Antenna phase center (receivers/transmitters)	IGS standard e.g. igs08-1645.atx
2nd order ionosphere model:	Basert på IONEX filer
Ocean loading	FES2004
Ocean pole tide model	Ja
Flerfasetvetydighet	Basert på Bertiger et al. [28]

Tabell 3.2.: Tabellen viser hvilke modeller som er brukt i GIPSY-analysen.

3.1.1. MET-data: A priori informasjon om ZHD og ZWD



Figur 3.1.: Differansen mellom korrekt stasjonshøyde og modellhøyden til MET er her plottet for lengde og breddegrad. Blå farge betyr negativ høyde, altså at MET-høyden er større enn den faktiske høyden, grønn er forskjellen rundt 0, mens rød betyr at den korrekte høyden er høyere enn MET. Plottet viser at det er store forskjeller i fjellområdene i midt-Norge, mens flatere områder langs kyst og på østlandet gir mindre forskjeller.

Dataene fra MET er som sagt hentet fra modellen AROME-MetCoOp, og er brukt til beregning av a priori ZHD og ZWD. Under er det beskrevet 3 ulike måter for å komme frem til denne utregningen. I og med at modellen er en 2,5 km grid med en egen høydmodell vil ikke trykk, temperatur og vanddamptrykket være nøyaktig korrigeret for riktig høyde for stasjonene. Gridpunktene er for en gitt høyde, men denne høyden er ikke korrekt når en interpolerer for lengde og breddegrad. For en del stasjoner der det ikke er lokalt store høydevariasjoner vil høyden som trykk, temperatur og vanddamptrykket korresponderer til kun variere med et par meter iforhold til GPS-stasjonen. Derimot på steder som har bratte fjell, slik som Lofthus i Ullensvang vil høyden variere

mye. Jeg har kjørt analysen for 3 ulike metoder for beregning av a priori informasjon fra MET-dataene:

MET-direkte

Trykket er ikke bare avhengig av høyden, men også av andre parametre, slik som vind, underlag og lignende. Dette gjør at dataene fra MET-direkte antas å være gode i utgangspunktet. Jeg startet derfor først analysen med MET-direkte, men så etterhvert at det var nødvendig med en høydekorrigering. Derfor er også GIPSY kjørt for to ulike måter for høydekorrigering. Vedlagt i tillegg E.1 ligger python-scriptet brukt til å hente ut data fra MET, for så å gjøre det om til riktig tdp-format (*time-dependent-parameter*) slik at det kan kjøres i GIPSY.

MET-standard

En metode som kan brukes for å korrigere for høyden er å bruke en standard atmosfære modell. Det finnes ulike standard atmosfæremodeller som kan gjøre dette og en metode beskrevet i [14] er følgende formel for korrigerende av trykket:

$$P = P_{sl} \left(\frac{T_{ksl} - \alpha H}{T_{ksl}} \right)^{\left(\frac{g}{R\alpha} \right)} \quad (3.1)$$

der P er trykket ved stasjonshøyde, P_{sl} [hPa] er trykket ved havoverflaten, H [m] er ortometrisk høyde, T_{ksl} [K] er temperatur ved havoverflaten, α er adiabatisk temperaturendring gitt ved 4,5 K/km, g er gravitasjon ved havoverflaten gitt ved 9,7867m/s² og R er gasskonstanten gitt ved 0,286KJ/°/kg for tørr luft. Dette settes så inn i formel (2.11) for beregning av hydrostatisk forsinkelse i senit.

Trykket ved havnivå er hentet fra AROME. Det er derimot ingen parameter for temperaturen ved havnivået og ved hjelp av trykknivåer fra AROME blir temperaturen ved 2 meter gjort om til å gjelde for havnivået. Relativ fuktighet er de opprinnelige dataene, uten noe form for høydekorrigering.

Deretter regnes vanndamptrykket ut ved hjelp av formel (2.18) som igjen brukes til å regne a priori våt forsinkelse ved formel (2.12). Hydrostatisk og våt forsinkelse basert på MET-standard atmosfæremodell er brukt til å modellere a priori troposfæreforsinkelse i GIPSY PPP løsningen.

MET-mellom

Etter mer analyse av resultatene fant jeg ut at for enkelte stasjoner med store høydeforskjeller var både den hydrostatiske og den estimerte våte forsinkelsen ganske annerledes enn for de andre metodene for MET-direkte. Dataene så ut som det var forskjøvet. Jeg valgte

derfor å kjøre enda en metode som jeg fikk anbefalt av MET å teste ut, som mente at standard atmosfæremodell kanskje ikke var god nok. I denne metoden gjøres en mellomting, der man tar utgangspunkt i interpolert verdi for lengdegrad og breddegrad, og bruker standard atmosfærisk modell til å beregne verdien for en riktig høyde for å se om dette ga et mer riktig resultat. Formel (3.1) skrives om til

$$P = P_0 \left(\frac{T_k - \alpha H_d}{T_k} \right)^{\left(\frac{g}{R\alpha} \right)} \quad (3.2)$$

der P_0 er trykket ved stasjonen før høydekorrigering, altså fra rådatene, T_K [K] er temperaturen ved stasjonen og H_d [m] er høydeforskjellen mellom korrekt høyde og MET-høyde. Koeffisienten α er fortsatt adiabatisk temperaturendring gitt ved 4,5 K/km, g er gravitasjon ved havoverflaten gitt ved 9,7867 m/s² og R er gasskonstanten gitt ved 0,286 KJ/°/kg for tørr luft. Dette trykket settes så inn i formel (2.11) for beregning av a priori hydrostatisk forsinkelse.

Det er viktig å ta hensyn til at enkelte løsninger for data-assimileringen hos MET kan være påvirket av den store høydeforskjellen, da ZTD er basert på data direkte fra MET, uten noen form for høydekorreksjon. Dette bør sees nærmere på ved senere samarbeid mellom MET og Kartverket. I figur 3.1 er differansen mellom korrekt stasjonshøyde og MET-høyden plottet for lengde og breddegrad. I vedlegg C er metoden brukt for å regne om geopotensial fra modellen til geometrisk høyde beskrevet. Her ser man tydelig at stasjoner på innlandet, i fjellpartiene altså i svært kupertede områder, har en større høydeforskjell. I tillegg A er det gitt en full oversikt over differansen mellom korrekt høyde og modellhøyden til MET.

3.2. Data-assimilering MET

For meteorologene kan det være interessant å motta total forsinkelse i senit fra Kartverket for forbedring av værmodeller og kanskje også værprognoser. For denne data-assimileringen lages en COST-format fil. Se vedlegg E.2, for python-koden brukt for å lage denne. Figur 3.2 viser data for Ås den første timen 01. november. I COST-formatet hentes ZWD, ZHD fra GIPSY PPP-analysen og ved hjelp av formel (2.16) regnes våt forsinkelse om til integrated water vapor. COST-fila inneholder oppdatering for hver 15. min for 1 time hver 3. time.

I denne oppgaven er ikke slant delay, altså forsinkelsen for riktig elevasjonsvinkel (ikke i senit), inkludert og er derfor satt til 0. For nærmere formatbeskrivelse se [29].

På grunn av begrenset tid er kun en kort beskrivelse av data-assimileringa inkludert i denne oppgaven.

Observasjonene fra COST-filene blir kjørt i en data-assimilering. Data-assimileringen, HARMONIE-Norway, som er basert på ALADIN (Aire Limitée Adaptation Dynamique Développement International) regional data-assimilering, bruker her en 11 km oppløsning [30].

```

COST-716 V2.2          E-GVAP          OPER
AASC XXXXXXXXXX      As, Norway
TRIMBLE NETR9        TRM57971.00    TZGD
 59.660333   10.781722   133.500    38.888    0.000
01-NOV-2015 00:00:00  29-MAR-2016 12:06:06
nma_              GIPSY 6.3          JPLPRE          OBS/
  INTERPOLATED
  15  60  360
FFFFFFFFF
 5
00 00 00 FFFFFFFF 2431.2   2.2  132.6   20.5 1010.5  282.1  88.9  -0.00  0.00  0.00
      0.00 -99.999
 0
00 15 00 FFFFFFFF 2427.7   1.9  129.3   20.0 1010.3  282.2  88.7  -0.00  0.00  0.00
      0.00 -99.999
 0
00 30 00 FFFFFFFF 2425.1   1.6  126.9   19.6 1010.1  282.2  88.5  -0.00  0.00  0.00
      0.00 -99.999
 0
00 45 00 FFFFFFFF 2424.1   1.4  126.0   19.5 1009.9  282.3  88.3  -0.00  0.00  0.00
      0.00 -99.999
 0
00 59 00 FFFFFFFF 2424.5   1.4  126.5   19.5 1009.8  282.4  88.1  -0.00  0.00  0.00
      0.00 -99.999
 0
-----

```

Figur 3.2.: COST-format til Ås i 1 november den første timen. Som en her ser er det data for hvert 15.minutt for 1 time, hver 3 time. Totalt gir det 40(5x8) oppdateringer pr. dag.

3. Metode

En spektral 3-dimensjonal analyse blir brukt i upper-air analysen, der cost-funksjonen blir minimalisert:

$$J = \frac{1}{2}(x - x_b)^T \mathbf{B}^{-1}(x - x_b) + \frac{1}{2}(y - \mathbf{H}(x - x_b) - H(x_b))^T \mathbf{R}^{-1}(y - \mathbf{H}(x - x_b) - H(x_b)) \quad (3.3)$$

her er x tilstanden når J er minimum, x_b er a priori estimat av atmosfærisk tilstand ved 6 timers prognose, y er observasjonsvektoren, H er en ikke-lineær observasjonsoperator som projiserer atomsfæretilstanden til observasjonsrommet, \mathbf{H} er en tangent-lineær versjon av observasjonsoperatoren, \mathbf{B} er kovariansmatrisa til bakgrunnsfeilen (a priori) og \mathbf{R} er kovariansmatrisa til observasjonsfeilen. H er kun brukt en gang til initialisering, mens \mathbf{H} kun er brukt til oppdatering [30].

ZTD fra GPS gir en del biaser som må korrigeres for, blant annet fra avbildingsfunksjonene, utregning av hydrostatisk forsinkelse o.l. [31]. I denne oppgaven er det brukt såkalt adaptive variational scheme (VarBC) for denne korreksjonen. Disse biasene er estimert i data-assimileringen på samme tid som man utleder en assimilering kontrollvektor og minimaliserer kostfunksjonen. For å beregne denne biasen blir det brukt en lineær prediksjonmodell,

$$\mathbf{b}(\beta, \mathbf{x}) = \sum_{i=0}^{N_p} \beta_i \mathbf{p}_i(\mathbf{x}) \quad (3.4)$$

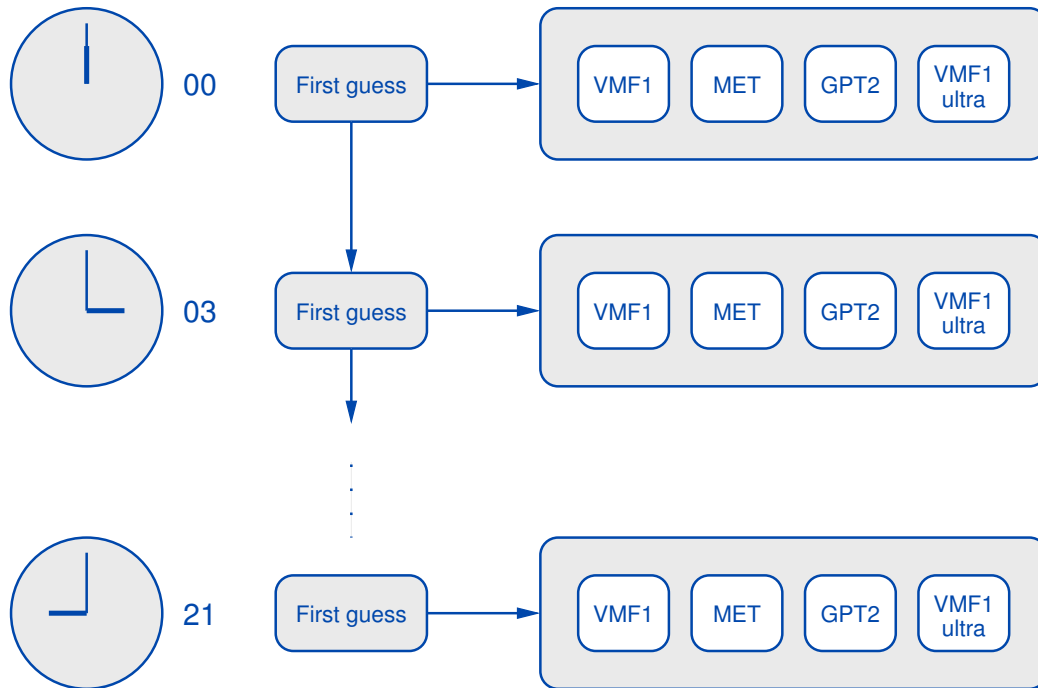
der \mathbf{p}_i er prediktorer, N_p er antall prediktorer og β_i er ukjente bias-parametere. For å estimere disse biasene inkluderes dem i kontrollvektorene.

I denne oppgaven er observasjonsdataene ZTD hentet fra de ulike metodene VMF1, GPT2 og MET-direkte beskrevet over for perioden 1. november - 10. desember. Grunnen til dette valget er at november/desember 2015 var spesielt fuktig i 2015, og kan dermed gi interessante resultater. På grunn av den korte tidsperioden jeg hadde til rådighet, samt problemer med data-assimileringen, fikk vi kun resultater fra 1. november - 12. november for VMF1-ultra.

Opgaven gikk ut på å sammenligne observasjonsdataene fra de ulike metodene med modellen. For å kunne gjøre dette må man ha modell-parametre. Dette gjøres ved å lage såkalte first-guess, eller bakgrunnsdata, som baserer seg på en cycling data-assimilering, for så å kjøre en 3-timers prognose. Denne modellen, som kun er basert på AROME- Met-CoOps konvensjonelle observasjoner, som vil si overflate-, radiosonde-, fly- og skips-data, ble kjørt på ECMWFs superdatamaskin. First-guess modellen blir deretter brukt til sammenligning (observert-minus-first-guess) for alle metodene, som er uavhengige data-assimileringer. Ved å gjøre det på denne måten blir det mulig å sammenligne alle metodene, for å se hvilken av dem som gir best resultat for ZTD for MET. I figur 3.3 er denne metoden illustrert. I assimileringen kjøres det en passiv-assimilering for alle tilgjengelige dataer. Det vil si at dataene bestod kvalitetssjekken og bidro til estimeringen

av bias-korrigeringskoeffisienten, men ikke til en full analyse [32]. Derfor undersøkes kun observasjon-minus-first guess, slik at analyse-resultatene her er irrelevante.

Data-assimileringen brukes til å sjekke kvaliteten på dataene, og i tillegg til å se om VarBC fungerer bra.



Figur 3.3.: Figuren viser hvordan de ulike metodene bruker samme bakgrunnsmodell, first-guess, for å se på om modell og observasjoner samsvarer. Her lages en first-guess assimilering, som baserer seg på first-guess for hver syklus og så en 3 timers prognose.

4. Resultater GIPSY PPP

I analysen har vi kjørt GIPSY for 217 stasjoner i Norge, og for fire forskjellige metoder: VMF1, VMF1-ultra, GPT2 og MET. I tillegg har vi kjørt analysen med ulike verdier av dataene hentet fra MET; MET-direkte, MET-standard og MET-mellom. VMF1-ultra blir ikke nevnt mye i denne delen av resultatene, da dette i hovedsak skal brukes for å se om man kan bruke nær-santids assimileringer for MET og vil inkluderes i kapittel 5.

Hydrostatisk, våt og total forsinkelse er plottet i perioden november-desember for å se hvordan disse varierer mellom metodene for de ulike stasjonene gitt i tabell 4.1. I denne delen er MET-standard og MET-mellom inkludert i tillegg til MET-direkte for å se på forskjellene de utgjør i våt, hydrostatisk og total forsinkelse.

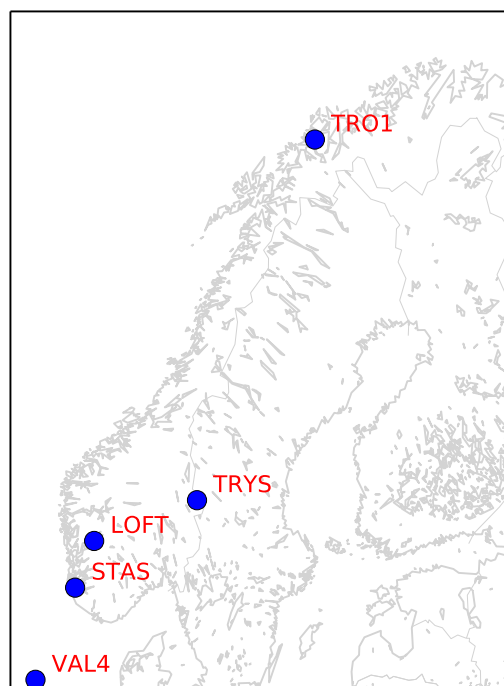
Siden det er så mange stasjoner er RMS for høyden til tidsserien, gjennomsnittlig våt, hydrostatisk og total forsinkelse plottet på et kart for å få et helhetlig bilde av Norge. Deretter er et representativ utvalg av stasjoner valgt ut for å se på en tidsserieanalyse over perioden november og desember 2015 for å se på forskjellene mellom metodene. Stasjonene som er valgt for perioden er gitt i tabell 4.1 og illustrert i figur 4.1.

For tidsserieanalysen er det fokusert på VMF1, GPT2 og MET-direkte, da forskjellene mellom disse metodene var størst.

Stasjon		Lengdegrad [°]	Breddegrad [°]	Ortometrisk høyde [m]
Stavanger	STAS	5.599	59.017	61.15
Trysil	TRYS	12.381	61.423	688.89
Tromsø	TRO1	18.939	69.662	106.93
Valhall	VAL4	3.393	56.275	101.60
Lofthus	LOFT	6.658	60.329	37.95

Tabell 4.1.: Tabellen viser et utvalg stasjoner som er sett nærmere på i oppgaven og ved hvilken ortometrisk høyde, lengde- og breddegrad de befinner seg.

Meteorologiske resultater er gitt i kapittel 5.



Figur 4.1.: Figuren viser hvor stasjonene i tabell 4.1 ligger plassert i Norge.

4.1. Hydrostatisk forsinkelse i senit

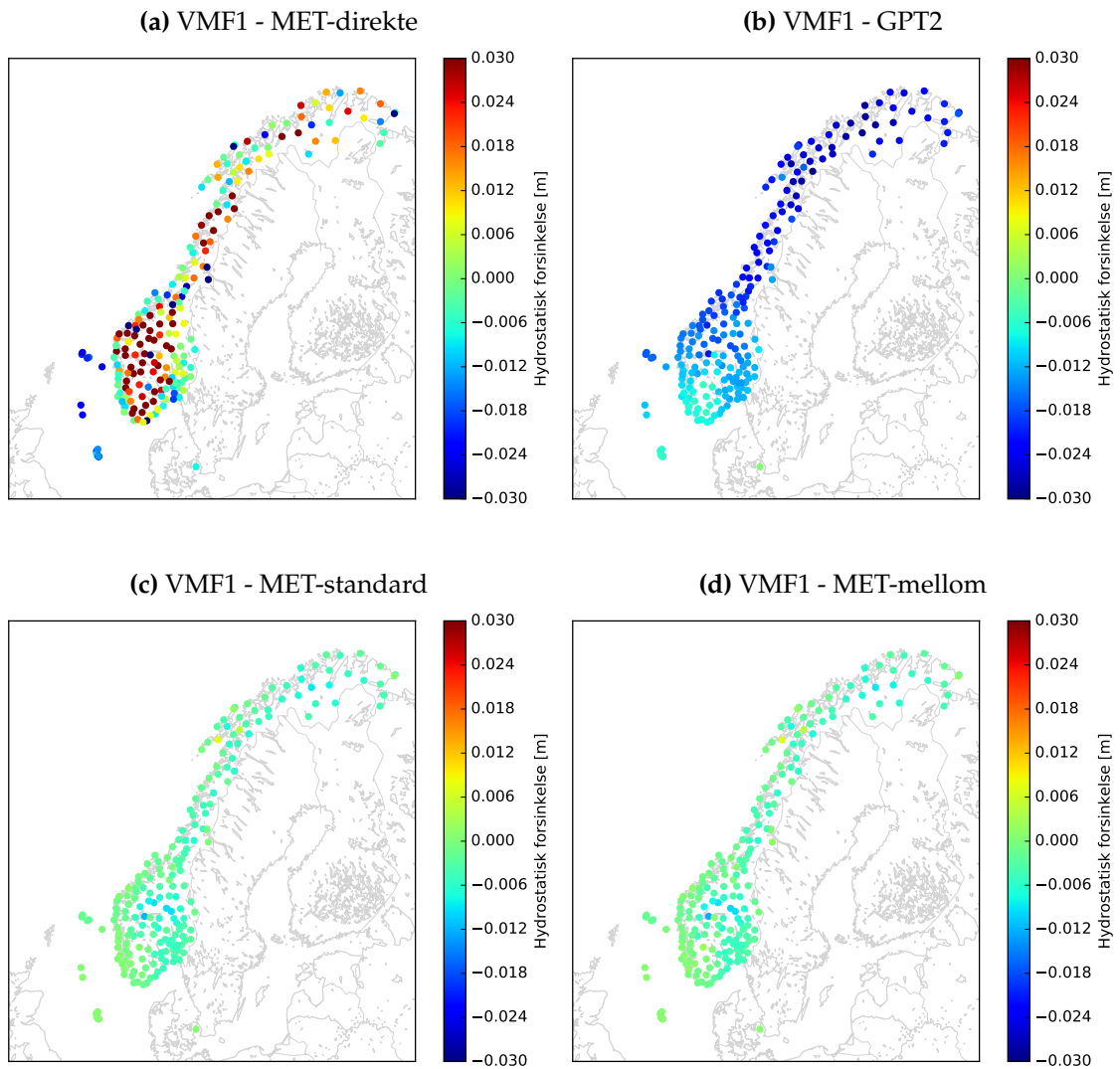
4.1.1. Met-direkte

De ulike metodene har, som diskutert i avsnitt 2.2, ulik informasjon for beregning av a priori hydrostatisk forsinkelse. Det kan dermed være interessant å studere forskjellene i a priori informasjon for de ulike metodene for å se hvilke forskjeller det er, og hva det eventuelt har å si for resultatene. I første del har vi sett på hydrostatisk forsinkelse i senit for VMF1, GPT2 og MET-direkte.

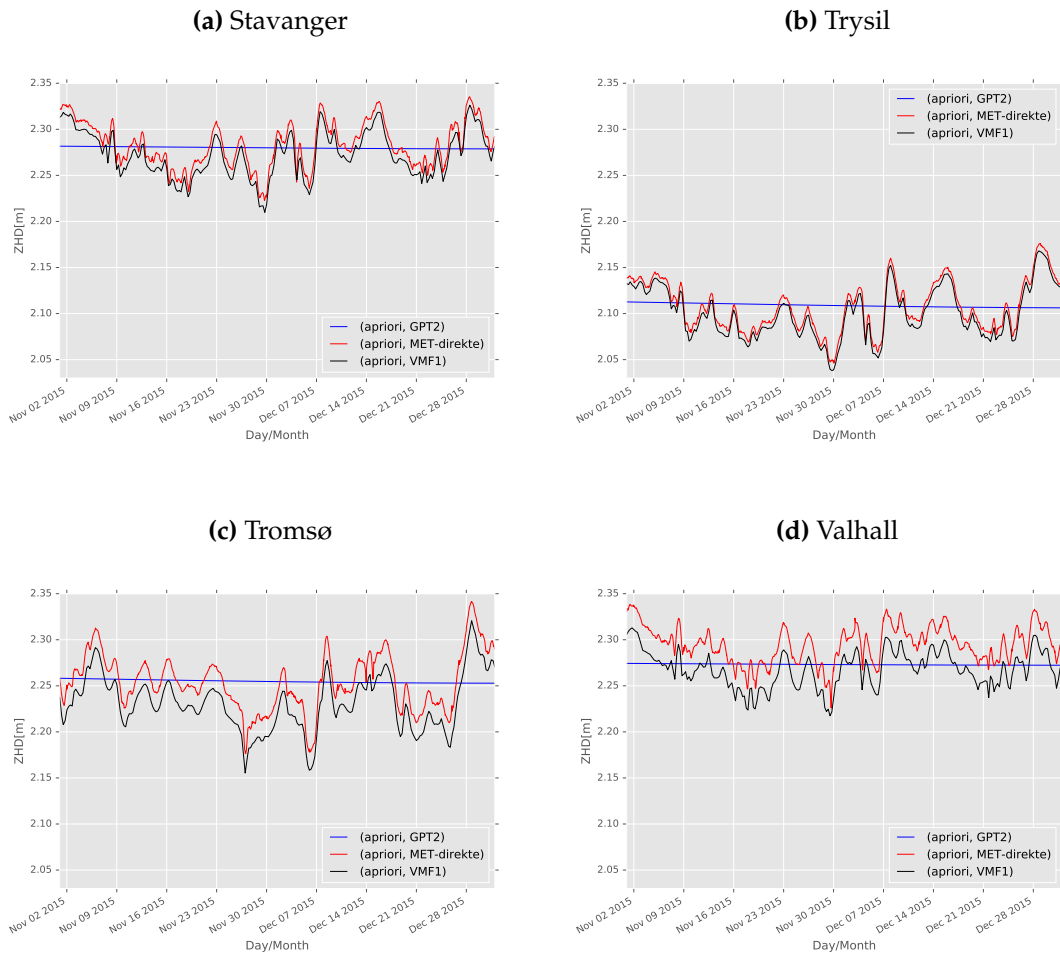
Siden det er mange stasjoner å forholde seg til kan det være en fordel å plote forskjellene for et gjennomsnitt av den hydrostatiske forsinkelsen i november - desember 2015 for de ulike metodene i et kart for å få et bilde av forskjellene fordelt over landet. I figur 4.2a er den gjennomsnittlige hydrostatisk forsinkelse for MET-direkte trukket fra den hydrostatiske forsinkelsen i VMF1. Her kan vi se at det er en forskjell som mest sannsynlig er avhengig av høydeforskjellen de ulike metodene operer med. Dersom vi sammenligner figuren med høydeforskjellen mellom MET-modellen og korrekt stasjonshøyde i figur 3.1, ser vi at disse ligner. En større forskjell mellom modellhøyde og korrekt høyde gir større utslag i forskjellen mellom hydrostatisk forsinkelse for MET-direkte og VMF1. I figur 4.2b er forskjellen mellom VMF1 og GPT2 plottet i et kart. Her ser vi at det er en tydelig breddegradsavhengighet mellom de to ulike metodene, der GPT2 gir en høyere hydrostatisk forsinkelse i nord, sammenlignet med VMF1.

For å se nærmere på disse forskjellene plottes hydrostatisk forsinkelse for alle metodene i perioden november og desember for alle stasjonene. Et utvalg stasjoner er vist i figur 4.3, der hydrostatisk forsinkelse er plottet for stasjonene Stavanger, Trysil, Tromsø og Valhall. MET-direkte ligger over VMF1 for alle stasjonene, men følger samme kurve. Dette kan skyldes manglende korrigerings av høyde. Her ser vi tydelig at GPT2 gir et grovt gjennomsnitt av a priori informasjon, noe som er hovedforskjellene på metodene. Vi ser at innlandsstasjonen Trysil generelt ligger på en hydrostatisk forsinkelse på 2.1 meter, mens Stavanger og Tromsø begge ligger på rundt 2.3 meter. For Valhall i figur 4.3d ser vi at det er stor daglig variasjon i den hydrostatiske forsinkelsen, noe som er typisk for de værutsatte oljeplattformene.

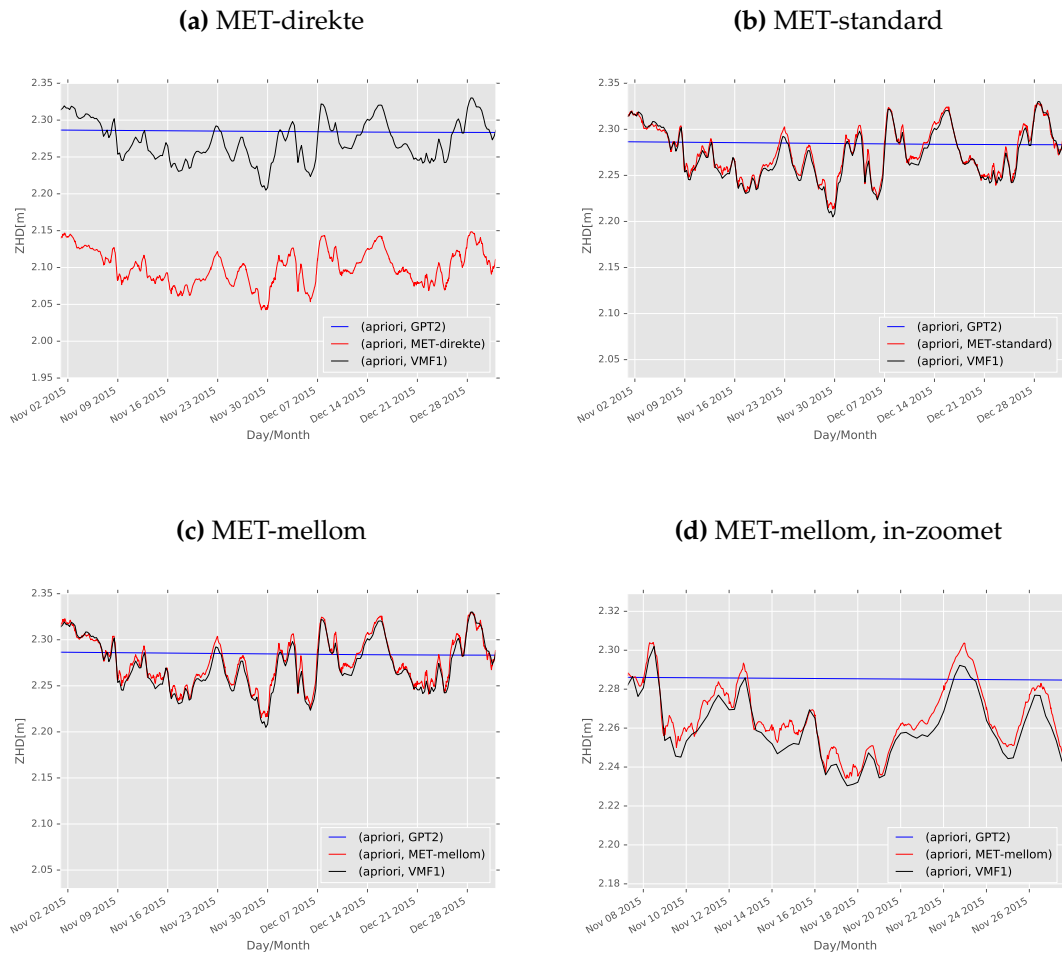
Forskjellen mellom hydrostatisk forsinkelse for MET-direkte og VMF1 kan, som nevnt, ligge i høyde-forskjellen. Dette kommer tydeligere frem for stasjonen Lofthus, som er stasjonen med størst forskjell mellom MET-dataene og korrekt-data, der høydeforskjellen er på hele 653.55 m forskjell. Ved å plote den hydrostatiske forsinkelsen for november og desember 2015 for Lofthus, i figur 4.4a, ser vi at forskjellen mellom MET-direkte og de andre metodene er opp mot 20 cm.



Figur 4.2.: Differansen mellom gjennomsnittlig hydrostatisk forsinkelse for a) VMF1-MET-direkte, b) VMF1-GPT2, c) VMF1-MET-standard og d) VMF1-MET-mellom plottet for lengde- og breddegrad. Fargeskalaen er gitt i meter.



Figur 4.3.: A priori hydrostatisk forsinkelse. Her for a) Stavanger, b) Trysil, c) Tromsø og d) Valhall, som er representative stasjoner for henholdsvis sørlig kyststasjon, sørlig innlandstasjon, en nordlig stasjon og en oljeplattform. Metodene sett på er VMF1, GPT2 og MET-direkte. Y-aksen er hydrostatisk forsinkelse gitt i meter.



Figur 4.4.: A priori hydrostatisk forsinkelse, her for Lofthus som er stasjonen med størst høydeforskjell mellom korrekt stasjonshøyde og MET-modell høyden. I a) brukes MET-direkte i b) med standard atmosfære modell, c) mellom-metoden og d) viser et kortere tidsrom av MET-mellom. Y-aksen er hydrostatisk forsinkelse gitt i meter, mens x-aksen viser dager i november-desember.

4.1.2. MET-modellene

Som vi nå har sett er det en feilmodellering av MET-direkte, der manglende høydekorreksjon gir feil a priori informasjon, spesielt på steder med stor høydendifferanse. Derfor er det testet to ulike metoder for høydekorrigering: MET-standard og MET-mellom, som er beskrevet i avsnitt 3.1.1.

Figur 4.2c viser differansen mellom VMF1 og MET-standard. Her er fordelingen annerledes enn for differansen mellom VMF1 og MET-direkte i figur 4.2a. Det er liten forskjell mellom hydrostatisk forsinkelse for VMF1 og MET-standard generelt, men det ser ut forskjellen mellom dem er avhengig av såkalte regnskygger. Steder som gir de største forskjellene i snitt er steder som er mye tørrer dersom man ser på nedbørsfordelingen i Norge, som er vedlagt i tillegg D for november og desember 2015.

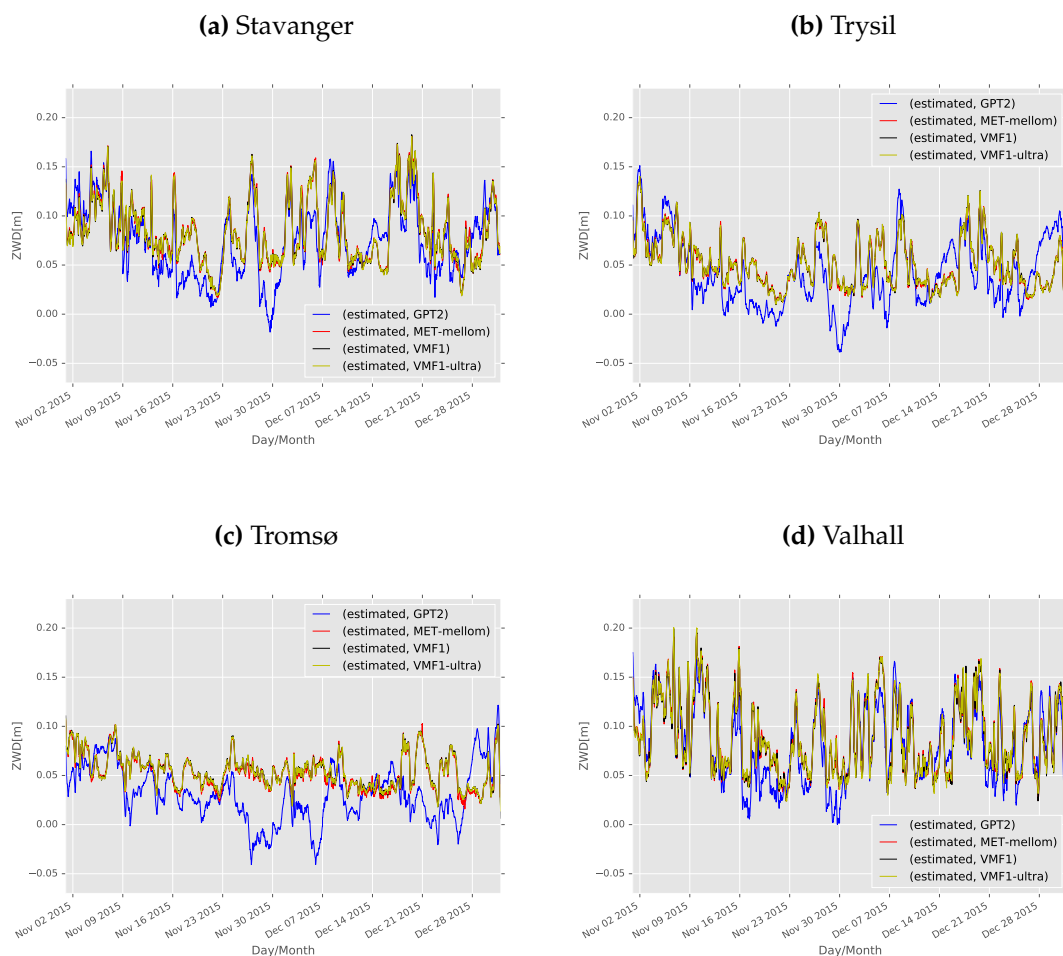
For å teste hvilket utslag det gjør å bruke standard atmosfæremodell blir hydrostatisk forsinkelse plottet for november og desember for GPT2, VMF1, MET-direkte, MET-standard og MET-mellom, som vist i figur 4.4 for stasjonen Lofthus. Her med MET-verdiene korrigeret for høyde ved hjelp av en standard atmosfæremodell, i figur b), og for MET-mellom i c). Figur 4.4b for MET-standard og figur 4.4c for MET-mellom viser at forskjellen mellom MET-metodene og VMF1 nå er liten, spesielt sammenlignet med MET-direkte i figur 4.4a. Det kan se ut som at MET-standard og MET-mellom har en mer nøyaktig beskrivelse av hydrostatisk forsinkelse sammenlignet med VMF1, som man ser dersom man zoomer inn for 18 dager i november for Lofthus og metoden MET-mellom i figur 4.4d. De følger samme grove kurve, men MET-mellom ser ut til å gi en større time-til-time variasjon, slik at VMF1 har en grovere tidsoppløsning enn MET-mellom. Forskjell mellom MET-standard og MET-mellom er på kun noen mm.

4.2. Estimert våt forsinkelse i senit

Da vi i forrige avsnitt så at MET-direkte har en feilmodellering av den hydrostatiske forsinkelsen har vi videre i denne delen valgt å se på metoden MET-mellom når metodene skal sammenlignes for den våte forsinkelsen. Hvilken av metodene MET-standard og MET-mellom som brukes er tilfeldig, da forskjellen mellom dem er veldig små.

Estimeringen av våt forsinkelse i perioden november - desember 2015 er plottet i figur 4.5 for de ulike metodene for stasjonene a) Stavanger, b) Trysil, c) Tromsø og d) Valhall. Metodene MET-mellom, VMF1 og VMF1-ultra ligger nære hverandre og viser at våt estimert forsinkelse ikke varierer veldig mye mellom metodene. At VMF1 og VMF1-ultra er like er naturlig, i og med at de har samme hydrostatiske forsinkelse. Den lille forskjellen mellom disse metodene gjelder generelt for alle stasjonene. Tromsø kan kanskje ha en antydning til litt større variasjon mellom MET-mellom og VMF1, der MET-mellom har en litt finere tidsoppløsning.

I figur 4.5a ser vi for Stavanger at det er store daglige variasjoner i våt forsinkelse og at den ligger på et snitt rundt 10 cm. Disse variasjonene skyldes nok at Stavan-

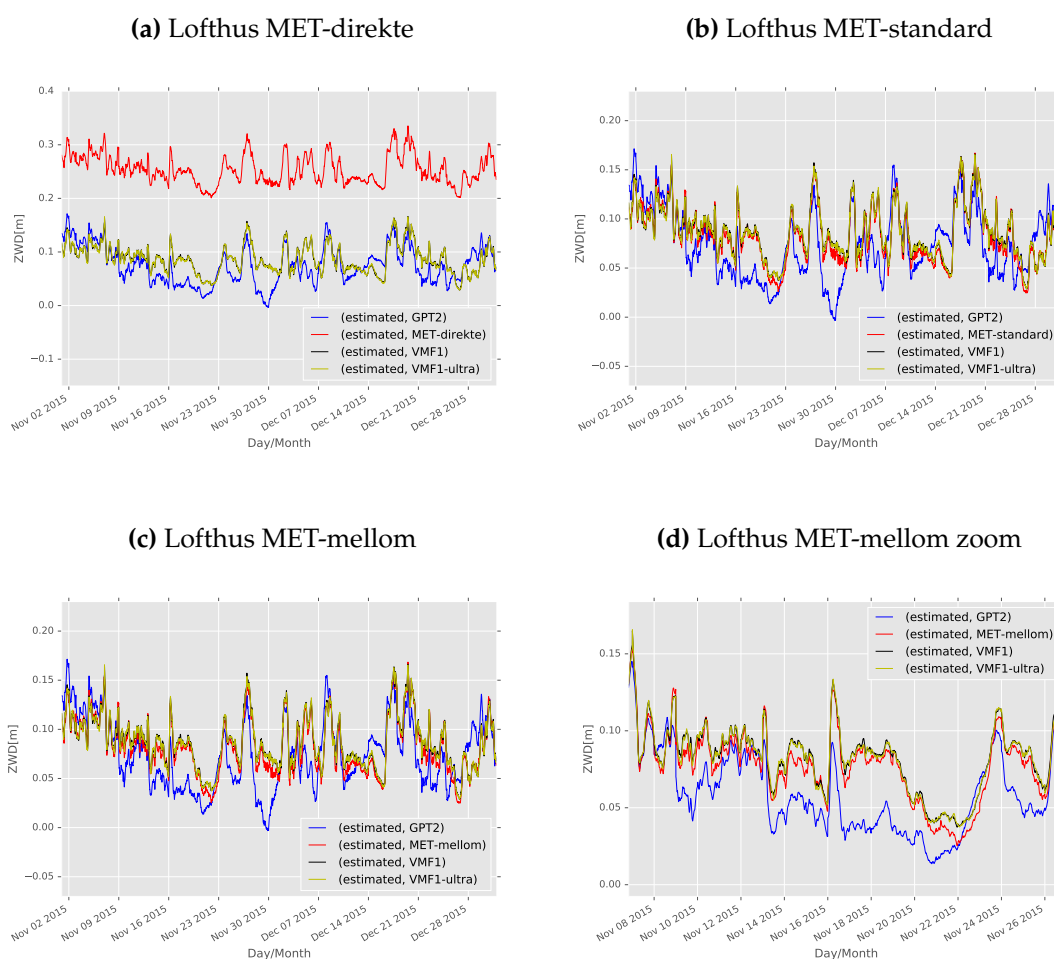


Figur 4.5.: Den estimerte våte forsinkelsen for VMF1, GPT2, VMF1-ultra og MET-direkte. Her for a) Stavanger, b) Trysil, c) Tromsø og d) Valhall som er en representativ stasjoner for henholdsvis sørlig kyststasjon, sørlig innlandstasjon, nordlig stasjon og en oljeplattform. Blå = GPT2, Rød = MET-mellom, grønn = VMF1-ultra og svart = VMF1. Y-aksen er våt forsinkelse i meter, mens x-aksen viser dager i november-desember.

ger er en kyst-stasjon med mye nedbør og vær-endringer, slik som en også kan se i nedbørskartene i tillegg D.

Trysil i figur 4.5b har en gjennomsnittlig våt forsinkelse på rundt 5 cm som er en del mindre enn for Stavanger. For Valhall er det en gjennomsnittlig våt forsinkelse på i overkant av 10 cm, med store dagsvariasjoner, som gjelder for mange værutsatte oljeplattformer.

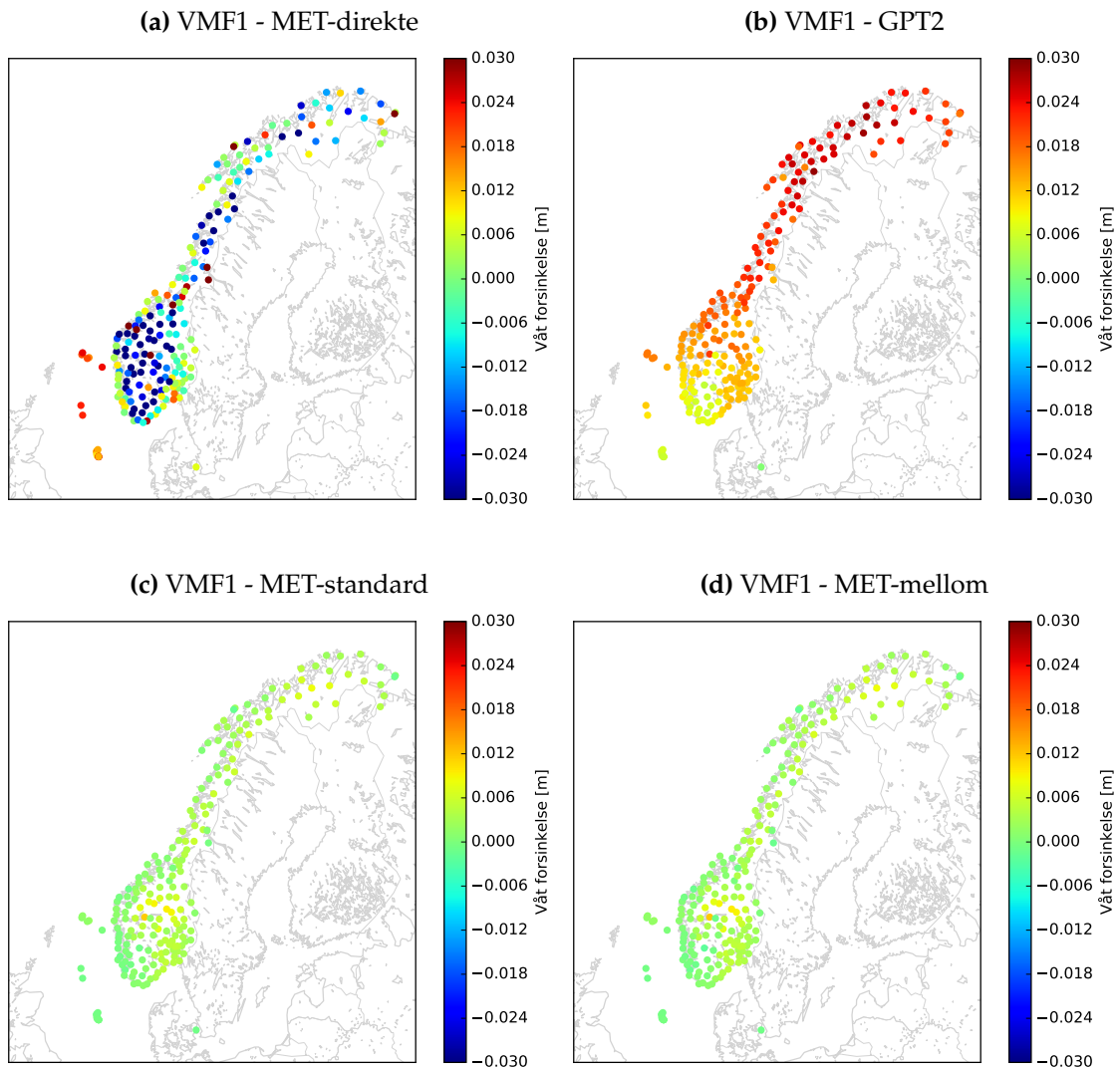
Ved estimeringen av GPT2 skjer det en kompensasjon for den grove modelleringa av



Figur 4.6.: Stasjonen Lofthus har en forskjell i høyden mellom MET-dataene og korrekt data på hele 653,66 meter og våt forsinkelse er her plottet for a) MET-direkte og b) MET-standard, c) MET-mellom og d) MET-mellom for en kortere tidsperiode. Y-aksen er våt forsinkelse i meter, mens x-aksen viser dager i november-desember.

hydrostatisk forsinkelse, det er derfor GPT2 er annerledes sammenlignet med de andre metodene. Dersom vi sammenligner våt forsinkelse for Tromsø i figur 4.5d med den hydrostatiske forsinkelsen i figur 4.3c ser vi at GPT2 korrigerer for den unøyaktige a priori informasjonen, og plottet er nesten identisk med formen som VMF1.

Dette er også tilfelle for MET-direkte, som vi kan se plottet for Lofthus i figur 4.6a. Sammenlignet med MET-mellom og MET-standard i henholdsvis figur 4.6c og 4.6b ser vi at korreksjon for høyde er viktig. Korrekt a priori informasjon for beregning av ZHD er med andre ord veldig viktig, spesielt dersom vi skal se på den våte forsinkelsen. Ved å regne om våt forsinkelse til integrated water vapor vil en få en stor feil for stasjoner som



Figur 4.7.: Differansen mellom gjennomsnittlig våt forsinkelse for a) VMF1-MET-direkte b) VMF1-GPT2 c) VMF1-MET-standard og d) VMF1-MET-mellom. Fargeskalaen er her i meter.

ikke er riktig høydekorrigert og også dersom GPT2 er brukt som modell. I figur 4.6c ser vi Lofthus for MET-mellom plottet for 17 dager i november. Her ser vi at det ikke er en like tydelig forskjell mellom tidsoppløsningen for VMF1 og MET-mellom som for den hydrostatiske delen.

Ved å studere forskjellene på gjennomsnittet av våt forsinkelse for de ulike metodene plottet på et kart i figur 4.7, ser vi også her at estimering av våt forsinkelse kompenserer for forskjellen i hydrostatisk forsinkelse. For VMF1- MET-direkte i figur 4.7a er det fortsatt den ukorrigerte høyden som skaper forskjellen mellom metodene, men her

er den våte forsinkelsen større for MET-direkte enn VMF1, disse forskjellene utjevner hverandre, som vi skal se i neste avsnitt. Det samme ser vi for VMF1-GPT2 i b), VMF1-MET-standard i c) og VMF1-MET-mellom i d).

4.3. Total forsinkelse i senit

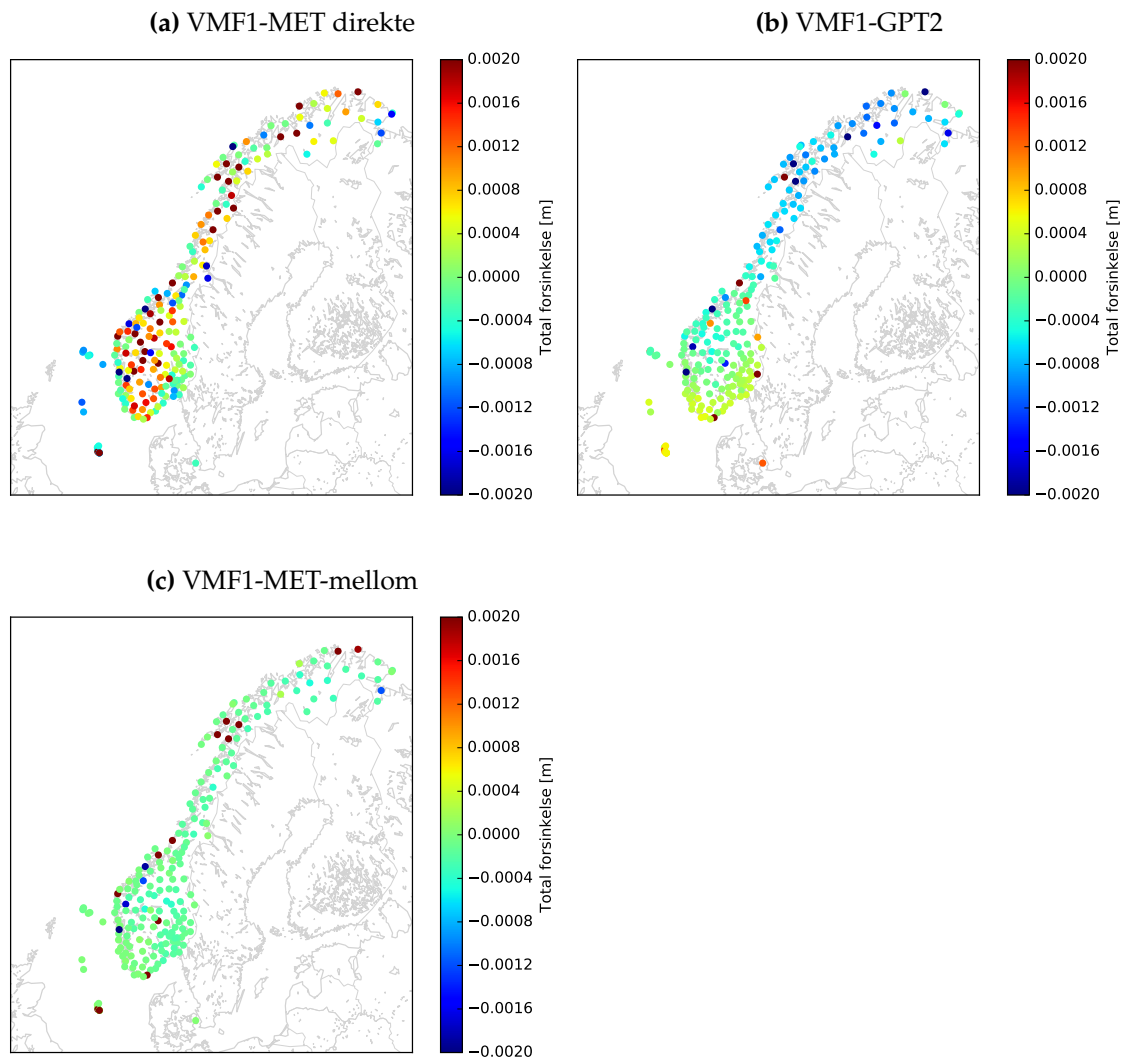
Det har nå blitt vist hvordan hydrostatisk og våt forsinkelse varierer med de ulike metodene. Det som nå er interessant er hvordan summen av dem ser ut, altså den totale forsinkelsen i senit. Dette plottes på samme måte som tidligere og en gjennomsnittlig oversikt over alle stasjonene, vist i figur 4.8, viser den samme trenden som tidligere, selv etter kompenseringen fra estimeringen. Her ser vi at den totale forsinkelsen for VMF1 ligger over MET-direkte for stasjoner med store høydeforskjeller. Skalaen er her satt til m, som vil si at forskjellene er på mm-nivå, noe som er en liten, men viktig forskjell, som kan gjøre en forskjell for posisjonsresultatene. Forskjellen mellom total forsinkelse for VMF1 og GPT2 ser fortsatt ut til å være breddegradsavhengig, der stasjoner i nord har en total forsinkelse som er større for metoden GPT2 enn VMF1 og omvendt i sør, selv etter korreksjonen. Dette er også på mm-nivå. VMF1-MET-mellom i figur 4.8c viser at forskjellene mellom disse metodene er veldig liten, og er rundt 0.1 mm.

Dersom vi ser på et utvalg stasjoner for perioden, vist i figur 4.9 ser vi at forskjellen mellom metodene er minimale. Her er MET-mellom igjen MET-metoden som er undersøkt. Vi kan også se at for Stavanger i perioden 4.-7. desember kan være antydning til mer forsinkelse, noe som samsvarer med stormen som herjet på den tiden. Det er i tillegg interessant å se på forskjellen mellom forsinkelsene for Stavanger i figur 4.9a, Trysil i figur 4.9b, og Tromsø i figur 4.9c. Den totale forsinkelsen til Trysil ligger på rundt 2.11 m, mens den for Stavanger er oppe i 2.38 m i snitt, Tromsø har en total forsinkelse på 2.28 m i snitt. Det er med andre ord store variasjoner i total forsinkelse avhengig av hvor i landet stasjonen befinner seg. Dersom vi ser på den våte delen er ikke forskjellene så store, noe som betyr at det er trykket som har mest og si for denne forskjellen. Total forsinkelse for Valhall i plot 4.9d viser, som for hydrostatisk og våt forsinkelse, en stor daglig variasjon.

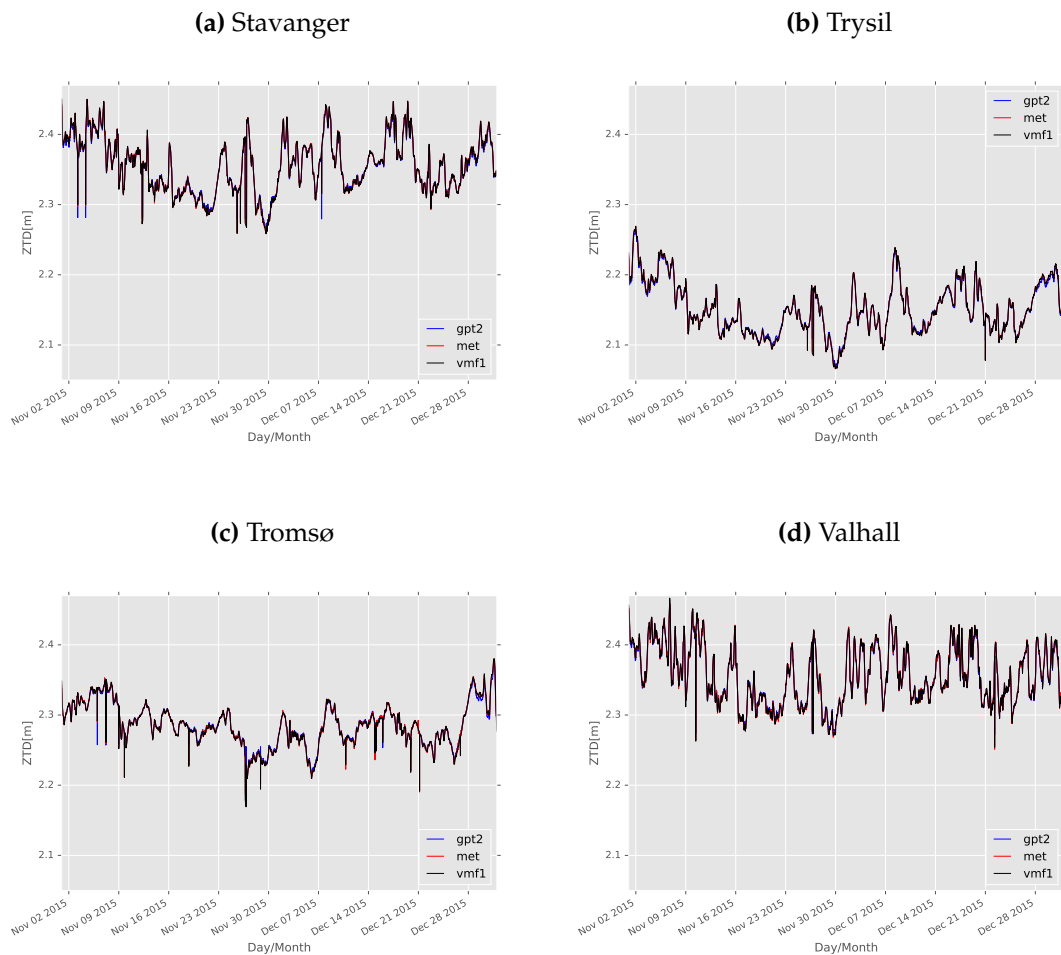
4.3.1. MET-metodene

Ved å korrigere for høyden får vi, som vist for Lofthus i figur 4.10b og 4.10c, en total forsinkelse som ikke har den store forskjellen for enkelte dager slik som MET-direkte har i figur 4.10a har. MET-direkte gir for Lofthus generelt en total forsinkelse litt under de andre metodene, og er på et punkt helt nede i en total forsinkelse på kun 2,10 m.

I figur 4.10d er MET-mellom plottet sammen med GPT2 og VMF1 for et kortere tidsrom for å se nærmere på forskjellene. Det er ikke en veldig tydelig tidsoppløsningsforskjell mellom MET-mellom og VMF1 her heller.

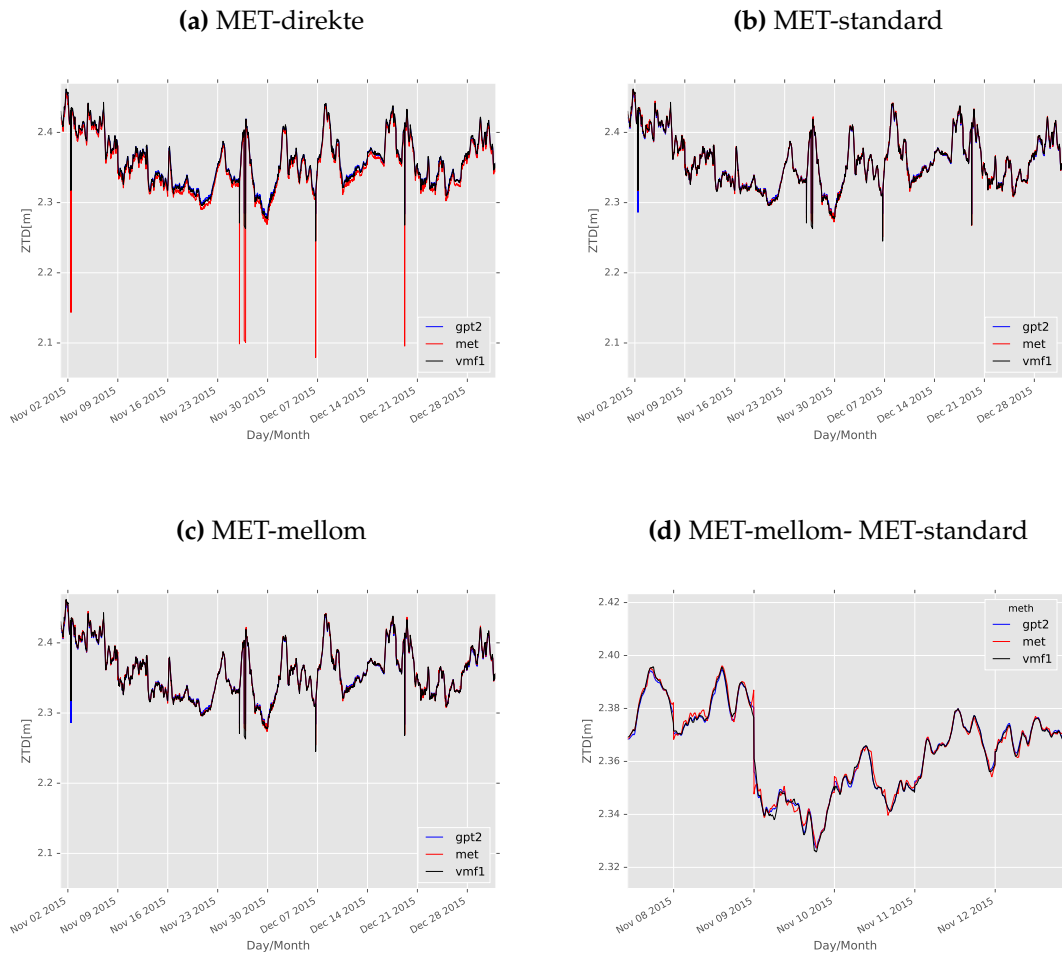


Figur 4.8.: Figuren illustrerer forskjellen mellom et snitt av total forsinkelse for perioden desember-november 2015 for a) VMF1 og MET-direkte, b) VMF1 og GPT2 og c) VMF1 og MET-mellom.

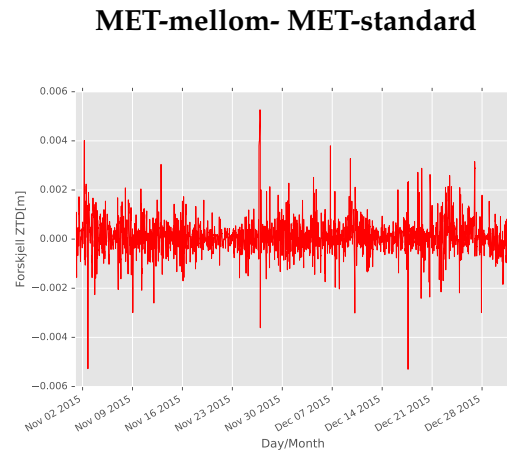


Figur 4.9.: Den totale forsinkelsen i senit for VMF1, GPT2, VMF1-ultra og MET-direkte. Her for a) Stavanger, b) Trysil, c) Tromsø og d) Valhall, som er en representativ stasjoner for henholdsvis sørlig kyststasjon, sørlig innlandstasjon, en nordlig stasjon og en oljeplattform. Her er ikke VMF1-ultra inkludert. Y-aksen er våt forsinkelse i meter, mens x-aksen viser dager i november-desember.

Lofthus



Figur 4.10.: Total forsinkelse i senit, her for Lofthus som er stasjonen med størst høydeforskjell mellom korrekt stasjonshøyde og MET-modell høyden. I a) brukes MET-direkte i b) MET-standard og c) MET-mellom. I d) er MET-mellom plottet for 4 dager. Y-aksen er total forsinkelse gitt i meter, mens x-aksen viser dager i november-desember.



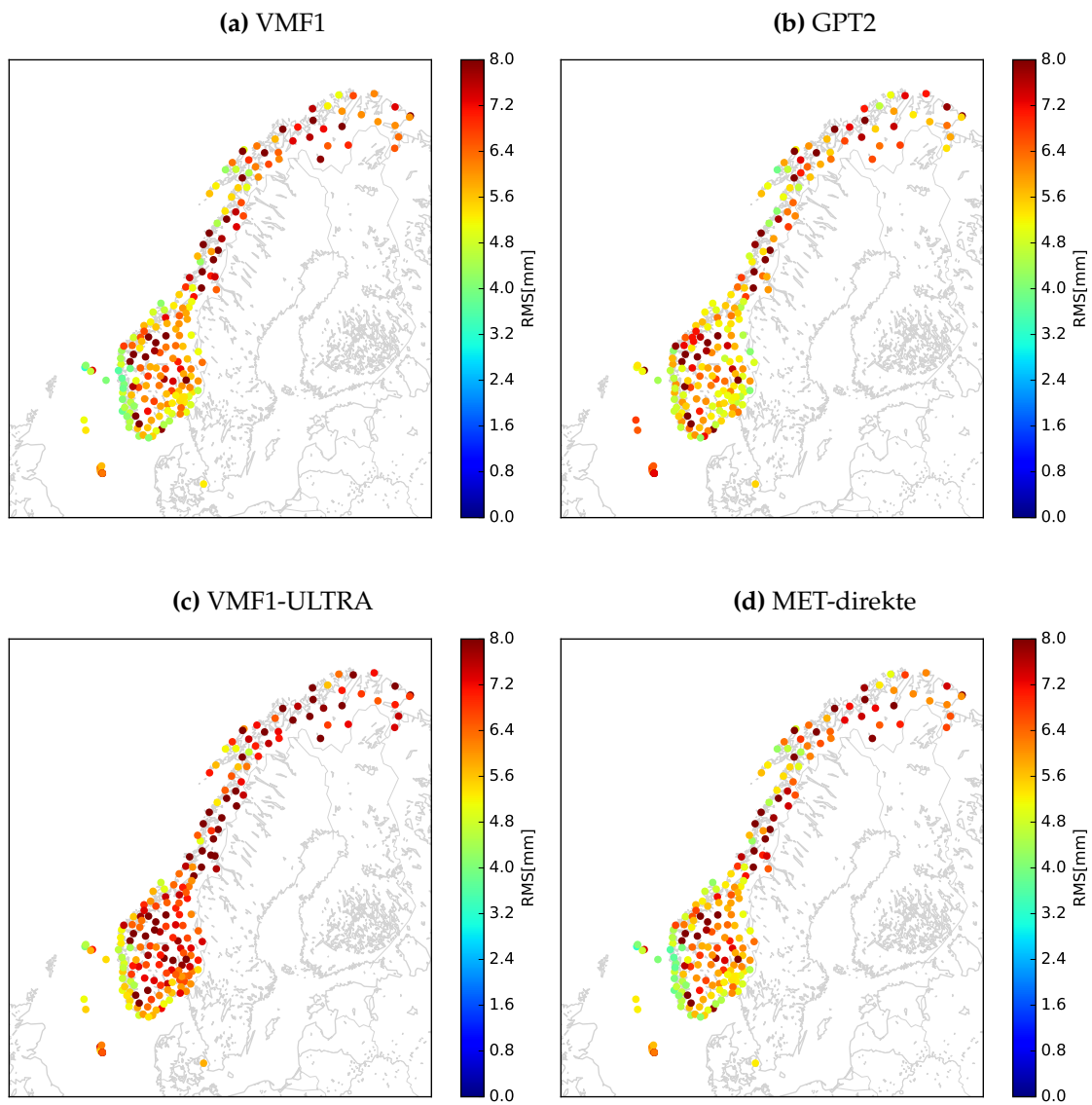
Figur 4.11.: Forskjellen mellom total forsinkelse for MET-standard og MET-mellom plottet for november-desember 2015.

Metodene MET-standard og MET-mellom gir også her veldig like resultater. I figur 4.11, der den totale forsinkelsen for MET-mellom er trukket fra MET-standard, ser vi imidlertid en at det er en forskjell mellom metodene. Denne forskjellen ligger på noen mm, men vil kunne utgjøre en liten forskjell på posisjonsresultatene.

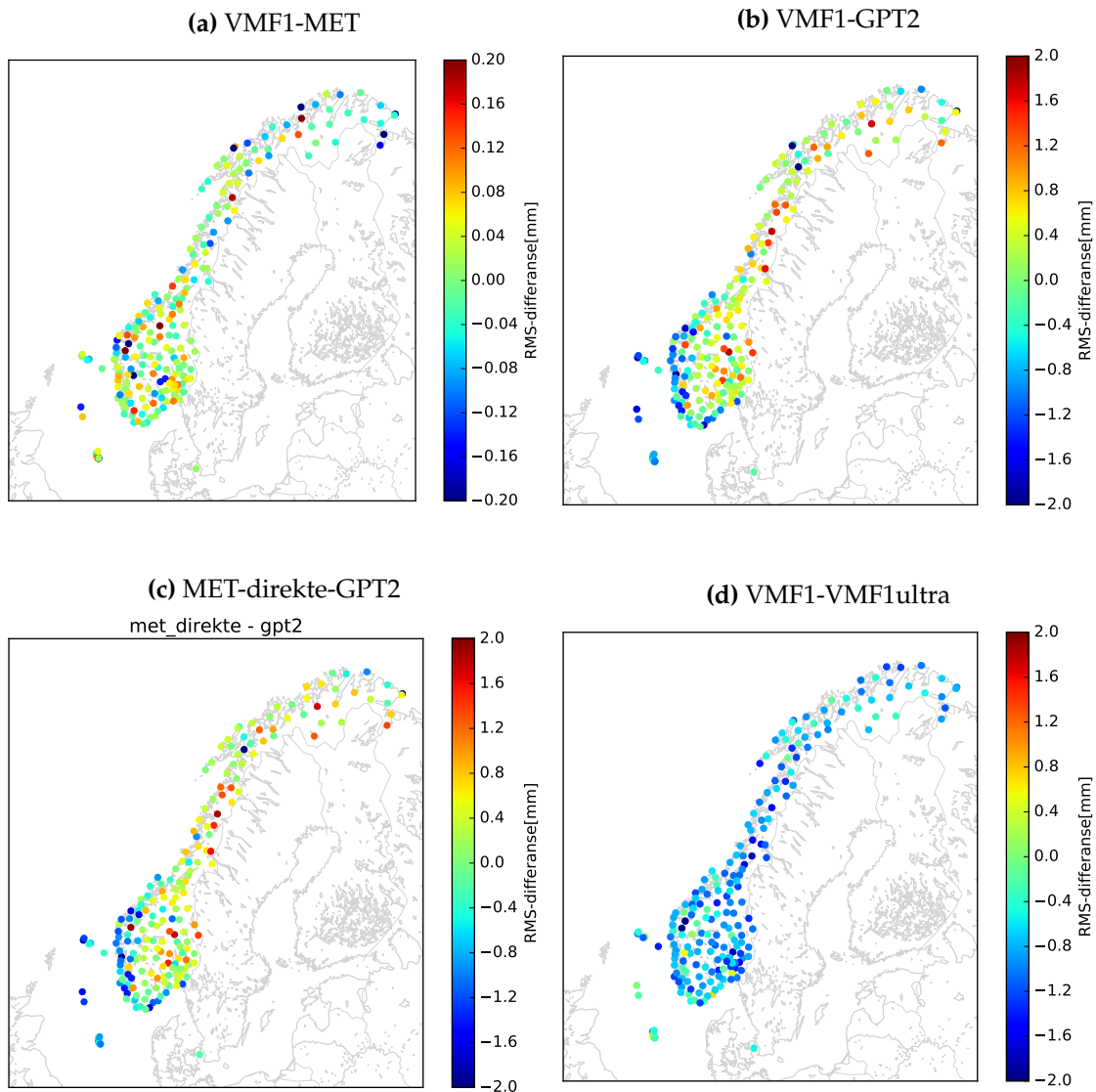
4.4. Tidsserieanalyse

Det er i oppgaven studert om det er forskjell på hvilke metode som gir den mest presise posisjonen. For alle stasjoner og alle metoder er det gjort en tidsserieanalyse hvor vi plottet variasjonen i øst, nord og høyde, slik som vist for Stavanger i figur 4.15. For hvert av plottene blir det laget en lineær og en harmonisk regresjonslinje. Når vi ser på troposfæren som eneste variabel mellom metodene er det naturlig å se på variasjonen i høyde, da det er denne som vil variere mest mellom de ulike metodene. Fordi det er et såpass kort tidsrom er det den lineære regresjonen som er mest nyttig å studere. Root-mean-square (RMS)-verdiene, i mm, står over hvert av plottene, og lavere RMS indikerer et bedre resultat.

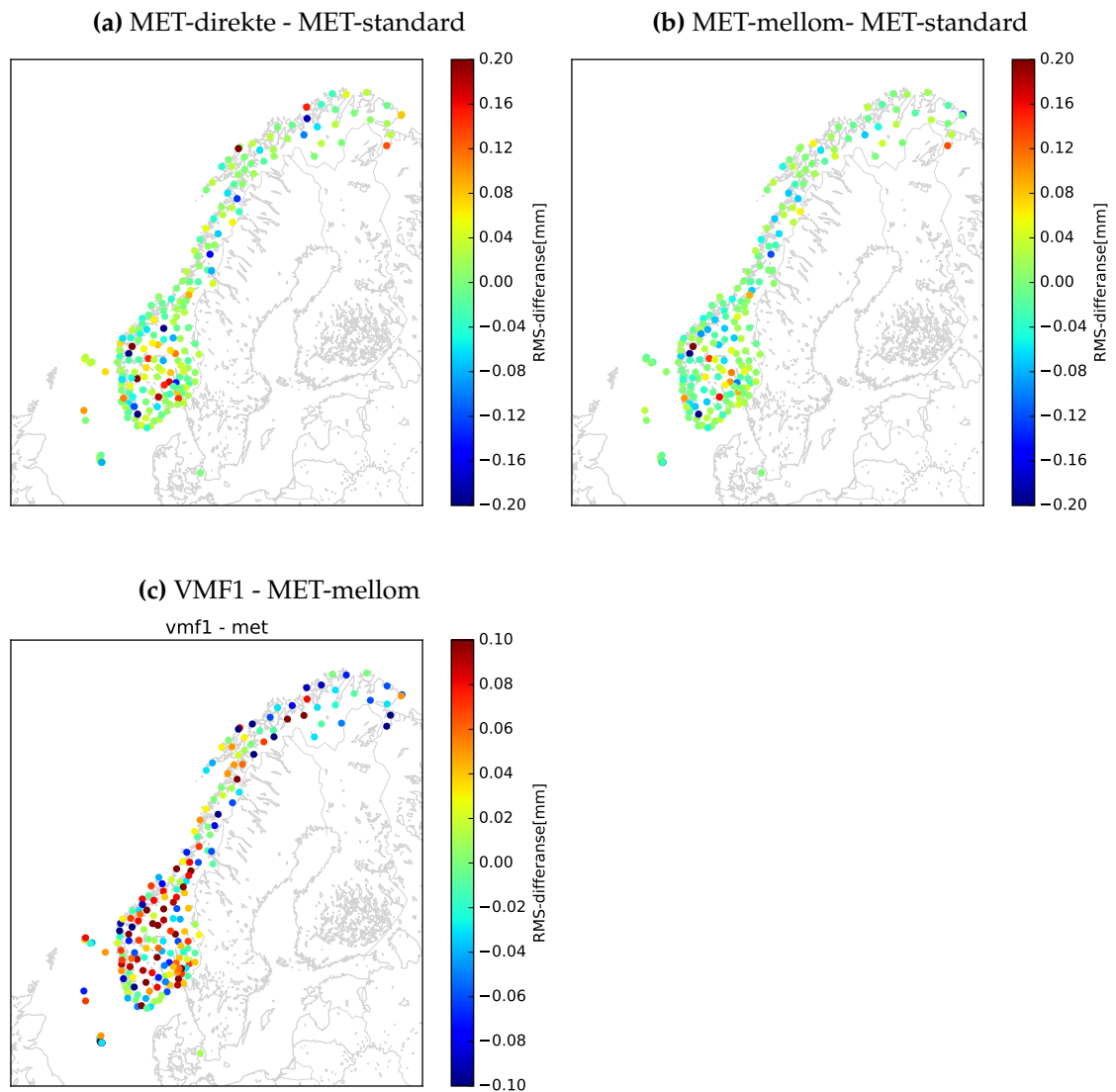
Ved å studere RMS-verdiene for høyden for alle stasjoner over hele Norge kan en få et bilde av hvordan RMS-verdien, og dermed hvor godt resultatet er, er fordelt over Norge. Figur 4.12 viser RMS-verdier for VMF1, GPT2, VMF1-ultra og MET-direkte, der ulike farger indikerer høy eller lav RMS. Høy RMS gir en mørk rød farge, mens blå er en lav RMS. Vi kan se at RMS ligger rundt 5 mm for de fleste stasjoner og metoder. For VMF1 og MET-direkte, henholdsvis figur 4.12a og 4.12d, kan det se ut som stasjoner nære kysten i Sør-Norge har lavere RMS enn stasjoner på innlandet.



Figur 4.12.: RMS-verdier til høyden for stasjoner i Norge for a) VMF1 b) GPT2 c) VMF1-ultra og d) MET-direkte. Fargeskalaen er i mm og rød indikerer en høy RMS, mens blå betyr at den er liten. RMS-verdiene er basert på en lineær tidsse-rieanalyse for perioden november-desember 2015.



Figur 4.13.: Differansen mellom RMS-verdiene til lineære tidsserieanalysen for høyden for a) MET-direkte-VMF1 b)VMF1-GPT2 c) MET-direkte-GPT2 og d) VMF1-VMF1-ultra. Fargeskalaen er i mm. Legg merke til at skalaen i a) er mindre enn for de andre metodene. Dette er fordi forskjellen mellom VMF1 og MET-direkte er mindre enn for de andre metodene, og får å se på forskjellene ble skalaen satt til 1/10 sammenlignet med de andre.



Figur 4.14.: Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember for a) MET-direkte - MET-standard, b) MET-mellom - MET-standard og c) VMF1-MET-mellom.

I tillegg til å se på RMS-verdien for hver metode kan de trekkes fra hverandre for å se hvilke metode som gir lavest RMS-verdi. I figur 4.13a er forskjellen mellom RMSen for VMF1 og MET-direkte plottet for lengde og breddegrad. Her er de blå verdiene negative, altså at RMS-verdiene til MET-direkte er større enn VMF1, mens for gul og rød er MET størst. Det er viktig å se på skalaen her, da denne er veldig liten for å få frem forskjellene. Det ser ut som hvilke metode som er best er litt tilfeldig.

Figur 4.13c viser forskjellen på RMS-høyde for MET-direkte og GPT2. Her er skalaen satt litt høyere enn for figur 4.13a fordi forskjellene er større her. En kan tydelig se at det er lavere RMS-verdier for MET-direkte langs kysten, da prikkene her er blå. På innlandet og ved høyere breddegrader kan det derimot se ut som GPT2 gir lavere RMS. Det samme sees i figur 4.13b, der VMF1 er sammenlignet med GPT2.

Figur 4.13d viser at VMF1-ultra, ikke overraskende, har mye høyere RMS enn VMF1 og bekrefter at bruk av baner og klokker som er beregnet i nær-sanntid gir en posisjonsbestemmelse med et større avvik.

Som en ser av figurene 4.12 og 4.13 er forskjellene mellom høyde-RMS mellom metodene fordelt forskjellig for ulike deler av landet. En kan tydelig se at VMF1 og MET-direkte gir et bedre resultat langs kysten i sør sammenlignet med GPT2. Det kan derfor være interessant å sammenligne sørlige kyststasjoner med innlandsstasjoner.

4.4.1. MET-metodene

I figur 4.14a ser vi forskjellen mellom RMS-verdiene for MET-direkte og MET-standard. En kan ut i fra dette se at det er litt tilfeldig hvilke av stasjonene som blir forbedret og forverret ved bruk av standard atmosfæremodell. Legg merke til at skalaen her er fra -0.2–0.2 mm. Det er ikke veldig tydelig hvilken av metodene som gir det beste resultatet. Forskjellene mellom MET-standard og MET-mellom, som vist i figur 4.14b, er heller ikke veldig stor, og det er ikke tydelig hvilken av dem som er best. I figur 4.14c er MET-melloms RMS trukket fra VMF1. Det kan se ut som det er en liten forbedring ved bruk av MET-mellom gir litt bedre resultater sammenlignet med VMF1, men dette er ikke veldig tydelig. Her er skalaen enda mindre, -0.1 – 0.1 mm for å få frem forskjellene.

Det er viktig å legge merke til at selvom RMS-verdien er liten, er det ikke nødvendigvis slik at resultatet er best, noe vi skal undersøke nærmere når en ser på tidsserieanalysen for ulike stasjoner og de ulike metodene.

4.4.2. Tidsserieanalyse

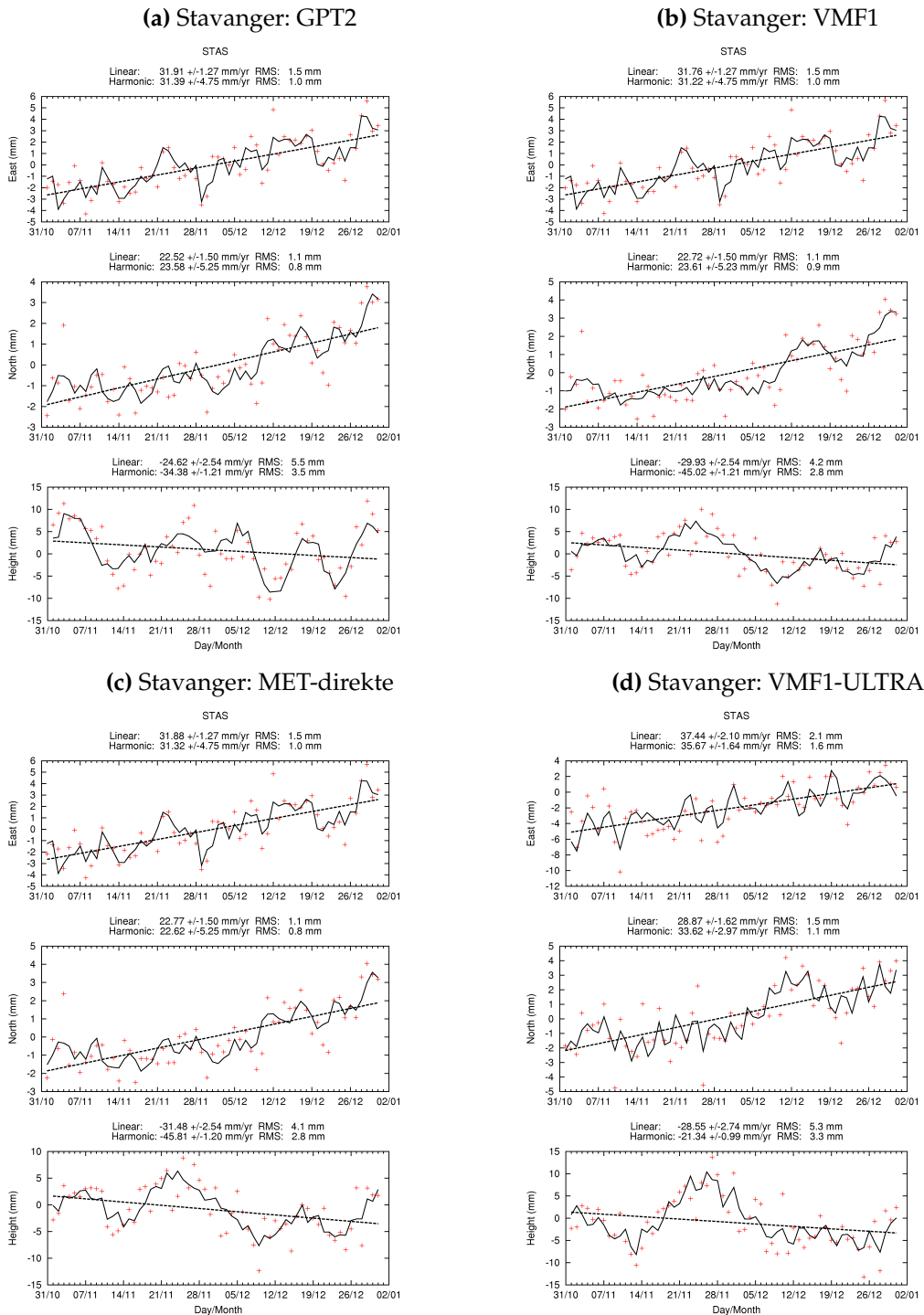
For nærmere analyse av metodene har jeg sett på tidsserien for de ulike metodene og på forskjeller mellom dem. Som tidligere er stasjonene gitt i tabell 4.1 sett nærmere på.

I figur 4.15 ser vi tidsserieanalysen for Stavanger i perioden november - desember 2015 for 4 ulike metoder, MET-direkte, GPT2, VMF1 og VMF1-ultra. Plottene for øst og nord er så og si like, bortsett fra for VMF1-ultra, som har litt større variasjoner, noe som er forventet. Dersom vi derimot ser på høyden kan vi se at MET-direkte og VMF1 er ganske like, mens GPT2 gir et ganske annerledes resultat. Alle metodene har et maksimum rundt 25. november, men variasjonene for GPT2 er mye større generelt i perioden. Dette gir en RMS som er mye høyere enn for VMF1, VMF1-ultra og MET-direkte. Rundt 4.-7. desember var det storm langs kysten i Sør-Norge, noe som kan ha påvirket resultatene. Vi kan se at MET-direkte og VMF1 har en outlier rundt 9 desember, men variasjonene er like store andre steder, slik at det er ikke nødvendigvis et tegn på at resultatet er spesielt påvirket av været. Stavanger er en representativ stasjon for stasjoner som ligger langs kysten i sør, der MET-direkte og VMF1 gir de beste resultatene, mens GPT2 og VMF1-ultra gir dårligere resultater.

I og med at MET-direkte, VMF1 og VMF1-ultra har ganske like utgangspunkt og derfor ikke vil være veldig forskjellige når en ser på posisjon, som en ser i 4.15, har jeg valgt å se på høyden til VMF1 og GPT2 for et utvalg stasjoner, da det er mest informasjon å hente ut av det. I tillegg har jeg valgt å se på differansen mellom VMF1 og GPT2, og forskjellene mellom VMF1 og MET-direkte. Det er interessant å se hvilken påvirkningen feilmodelleringa av høyden har å si for posisjonen når vi sammenligner MET-direkte og VMF1.

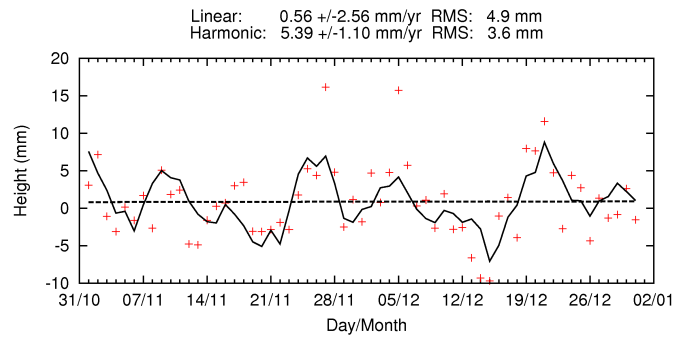
I 4.16 ser vi på høyden i Trysil for GPT2 og VMF1, henholdsvis i figur 4.16a) og 4.16b). I tillegg er forskjellen mellom dem plottet i 4.16c). Trysil er en typisk innlandsstasjon og er langt unna kysten, med omlag 200 km til nærmeste kyst. Her ser vi et ganske annet resultat enn for Stavanger. VMF1 har en mye større spredning her enn ved kystområdet, og gjør at GPT2 gir et bedre resultat enn VMF1. Forskjellen mellom dem er stor og strekker seg fra -10 mm til 10 mm. Om vi ser på forskjellen mellom VMF1 og MET-direkte er denne ganske liten og har et maksimum på 1 mm. Det som er interessant med denne sammenligningen er at vi kan se en liten forskyvning, og den lineære linja går ikke gjennom 0, men rundt 0.8 mm. Forskjellen mellom MET-modellhøyde og korrekt stasjonshøyde er 26.82 m, noe som kan være grunnen til denne forskyvningen.

For Tromsø i figur 4.17 a) og b) ser vi at både GPT2 og VMF1 har stor spredning, og har relativt like verdier, bortsett fra helt mot slutten av perioden. Dette er spesielt tydelig i figur c). Generelt er forskjellen mellom dem fra -5 mm til 5, men helt mot slutten er det en forskjell opp mot 20 mm. Det samme gjelder for mange stasjoner i nord. Grunnen til dette kan være knyttet til atmosfærisk loading, som vi skal se nærmere på i avsnitt 4.4.3. For sammenligningen mellom VMF1 og MET-direkte i figur 4.17d) ser vi også her en forskyvning, denne gangen på 2 mm. Her er forskjellen mellom MET-modellhøyde og korrekt stasjonshøyde 75.44 m, noe som igjen bekrefter at denne høydedifferansen kan

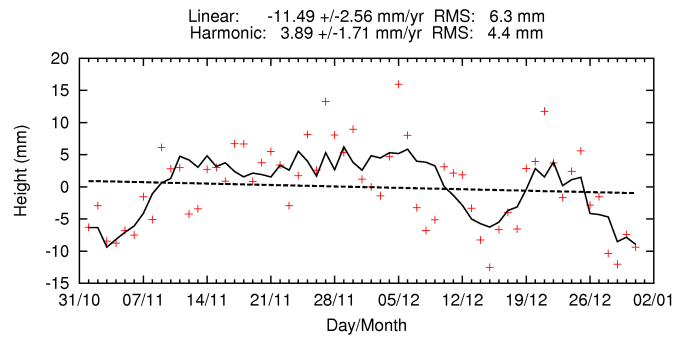


Figur 4.15.: Tidsserieanalyse for øst, nord og høyde for Stavanger for de ulike metodene i perioden november–desember 2015. RMS-verdien står over hvert av plottene; en for lineær og en for harmonisk regresjon. I studering av troposfæren er det mest hensiktsmessig å se å på den lineære RMS-verdien til høyden for de ulike metodene for hvert stasjon. Figuren viser for a) GPT2, b) VMF1, c) MET-direkte og d) VMF1-ultra.

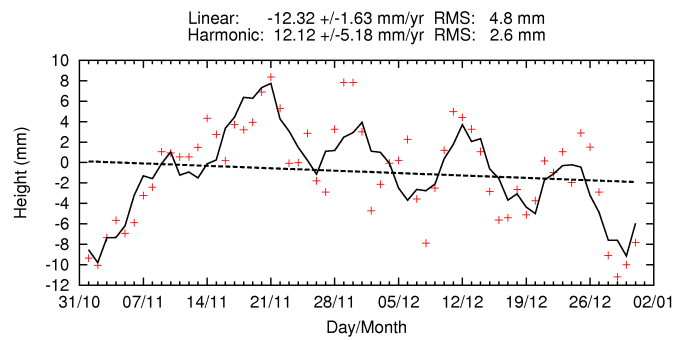
a) Trysil: GPT2



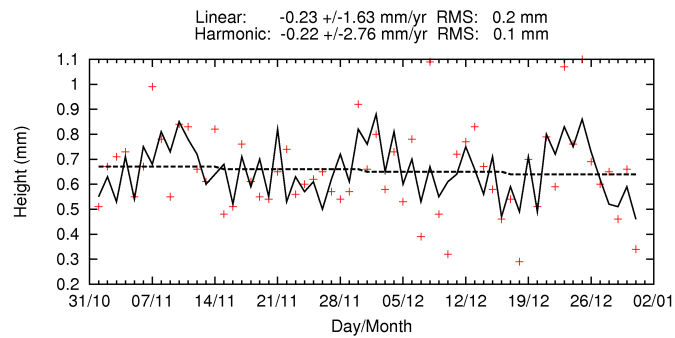
b) Trysil: VMF1



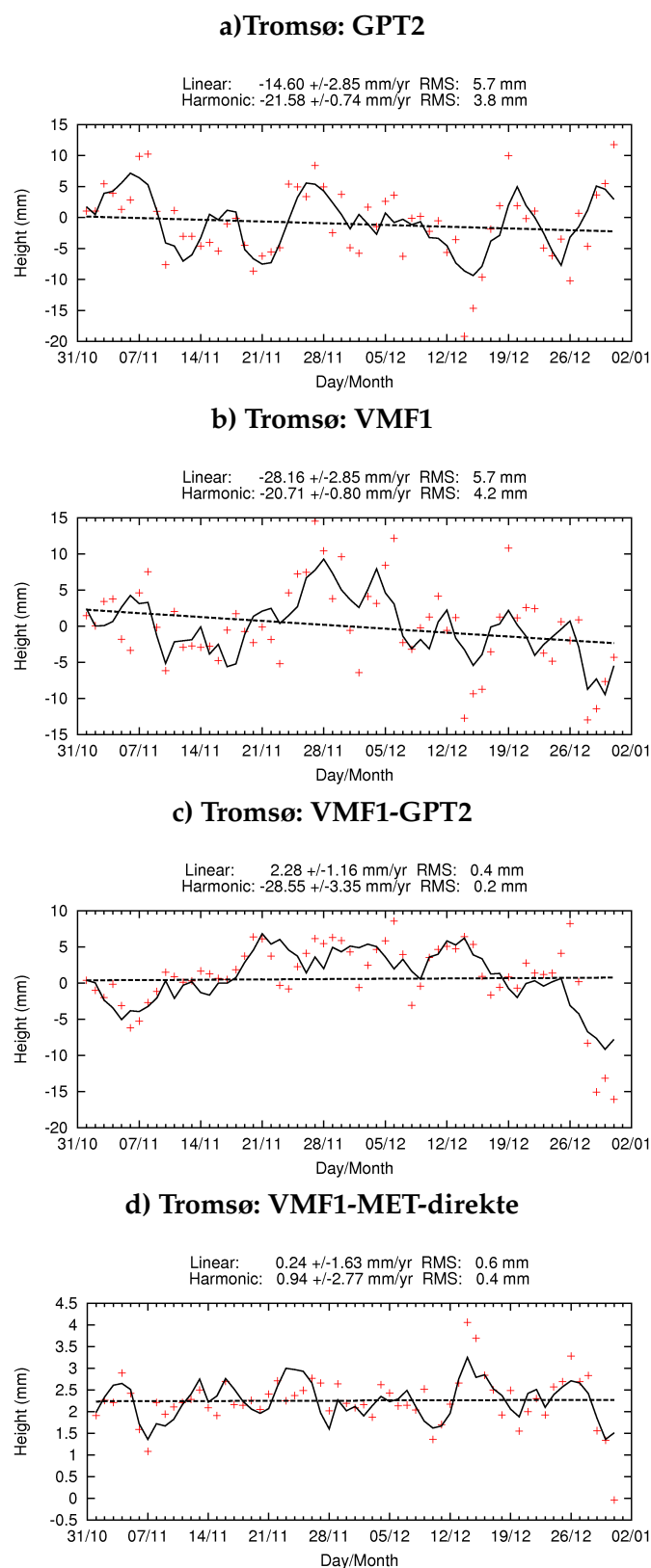
c) Trysil: VMF1-GPT2



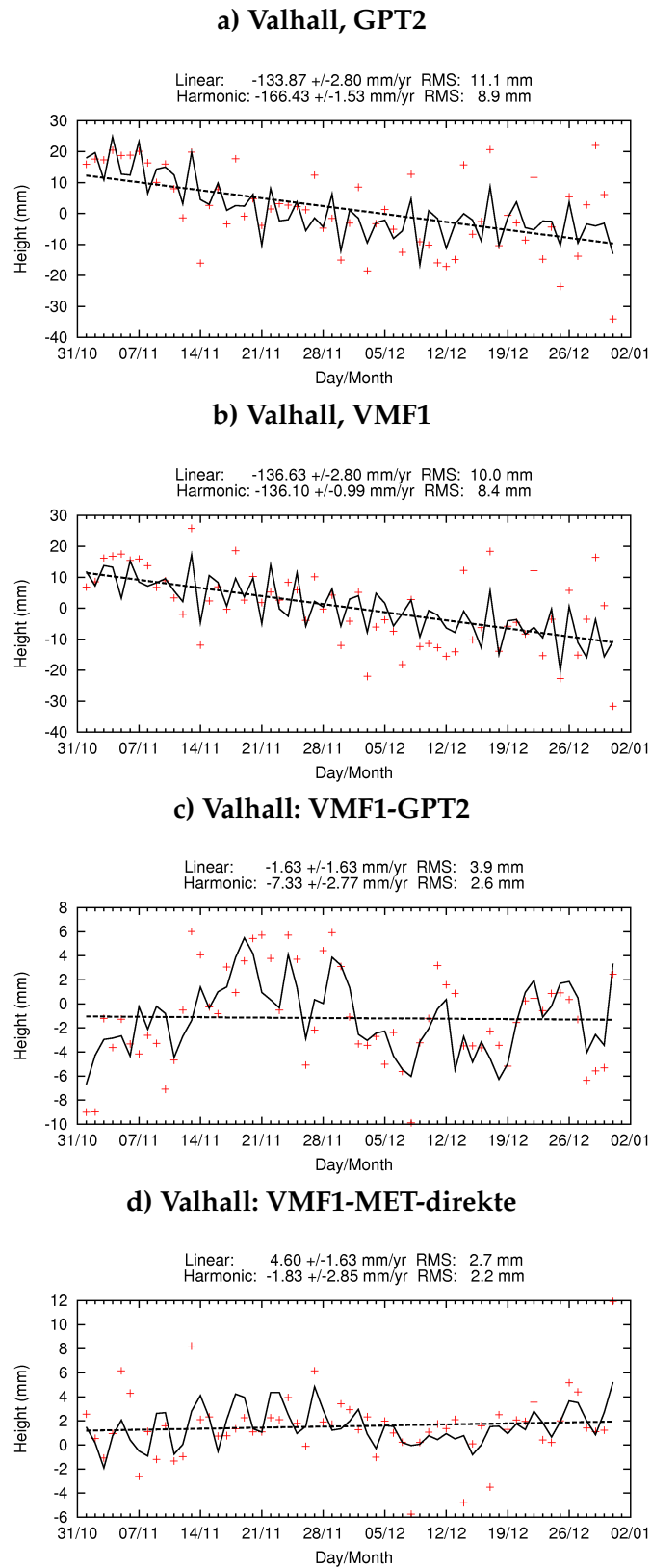
d) Trysil: VMF1-MET direkte



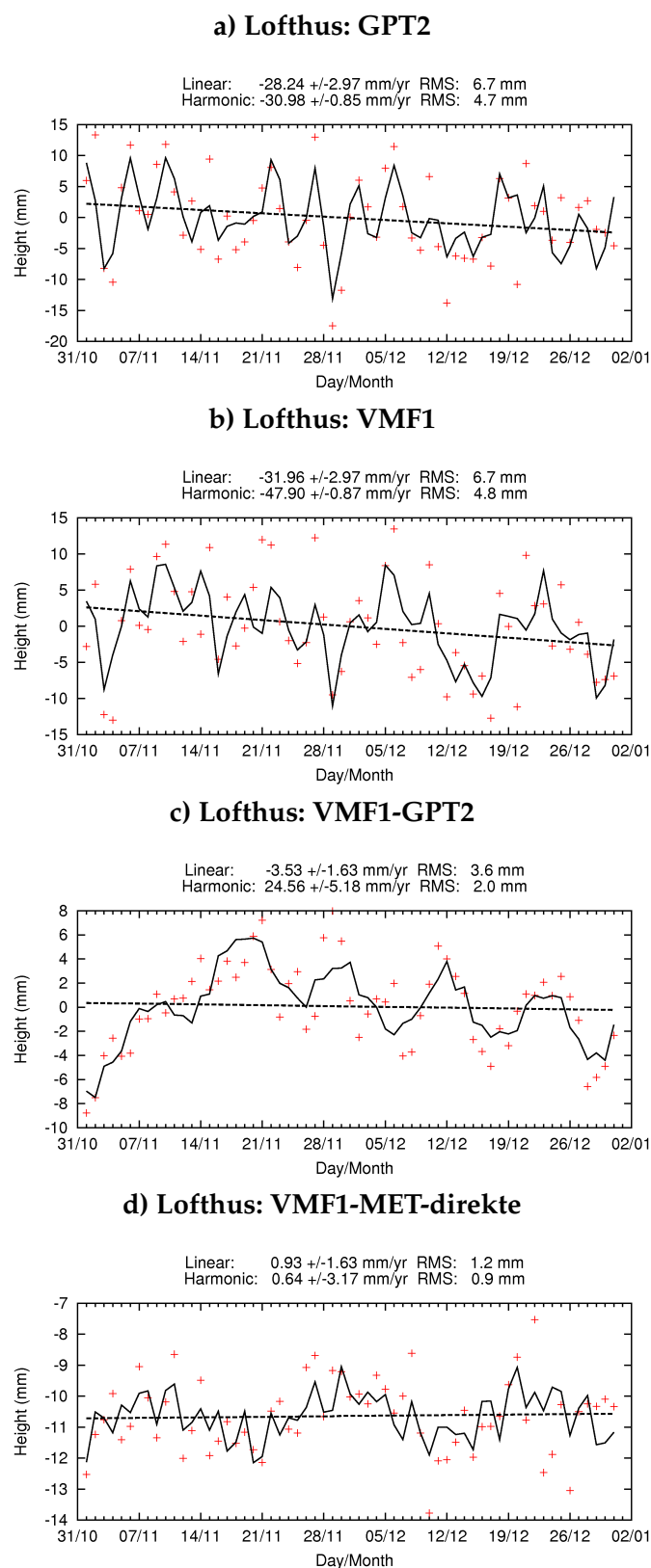
Figur 4.16.: Tidsserieanalyse for Trysil, for a) GPT2, b) VMF1, c) VMF1-GPT2 og d) VMF1-MET direkte.



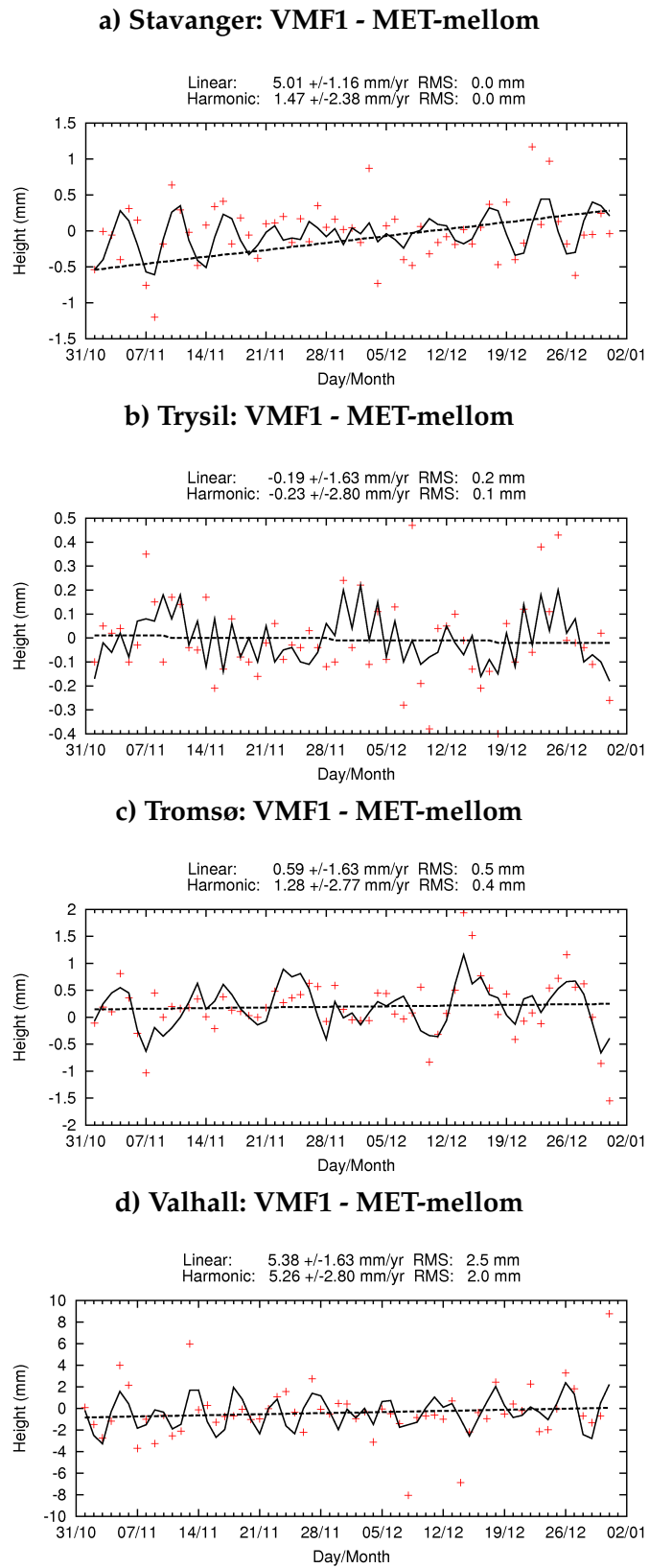
Figur 4.17.: Tidsserieanalyse for høyden til Tromsø for perioden november-desember 2015 for a) GPT2, b) VMF1, c) VMF1-GPT2 og d) VMF1-MET-direkte.



Figur 4.18.: Tidsserieanalyse for høyden til oljeplattformen Valhall for perioden november-desember 2015 for a) GPT2, b) VMF1, c) VMF1 - GPT2 og d) VMF1 - MET-direkte.



Figur 4.19.: Tidsserieanalyse for høiden til Lofthus for perioden november-desember 2015 for a) GPT2, b) VMF1 og c) VMF1 - GPT2 og d) VMF1-MET-direkte.



Figur 4.20.: Tidsserieanalyse for differansen mellom høyden til VMF1 og MET-mellom. a) Stavanger, b) Trysil, c) Tromsø og d) Valhall.

føre til posisjonsfeil.

Valhall i figur 4.18 viser en typisk høyde-tidsserie for oljeplattformene. Her er det store variasjoner i både GPT2 og VMF1, og høyden varierer mellom 20 mm og -20 mm. Differansen mellom dem viser at for de ulike metodene kan høyden variere med opp til 10 mm. Forskjellen mellom VMF1 og MET-direkte i figur 4.18d) er større her enn for andre stasjoner, med en forskjell opp mot 8 mm. Det er også en forskyvning på rundt 2 mm her, noe som kan korrespondere med en høydifferanse på 101 m for Valhall.

For å se nærmere på om manglende høydekorreksjon kan være grunnen til forskyvingen mellom MET-direkte og VMF1 kan vi igjen se på Lofthus i figur 4.19. Her er spredningen for GPT2 og VMF1 generelt store, i figur a) og b), men VMF1 ser ut til å gi det beste resultatet. Det som er mest interessant her er å se på forskjellen mellom MET-direkte og VMF1; forskyvningen nå er på -11 mm, noe som illustrerer hvor viktig det er med riktig korreksjon for høyde. RMS-verdien til slike plot vil ikke nødvendigvis være noe dårligere enn dersom den er korrigerert riktig, fordi forskyvningen gjør at regressjonslinja også forskyver seg. Dermed er det viktig å ikke ta RMS-verdiene som eneste kilde til at metoden gir et godt resultat.

Det er sett at manglende høydekorreksjon av a priori hydrostatisk forsinkelse her gir relativt store utslag for resultatene. MET-direkte er dermed ikke metodene som burde brukes. Om vi derimot ser på forskjellen til høydetidsserien mellom VMF1 og MET-mellom i figur 4.20, er det altså en forskjell mellom dem selv etter høydekorreksjonen. Størst forskjell er det for Valhall, med en forskjell rundt 2 mm. Deretter er det Tromsø og Stavanger som begge gir en forskjell på litt under 1 mm, mens Trysil kun ligger rundt 0.2 mm.

Av resultatene kan vi se at stasjoner nære kysten har en antydning til lavere RMS, og mindre spredning i tidsserieanalysen til høyden. Som nevnt tidligere kan en av grunnene skyldes atmosfærisk loading. Fenomenet inverted barometric effect kan forklare hvorfor stasjonen langs kysten var gode, fordi atmosfærisk loading her blir utjevnet av den effekten.

4.4.3. Korreksjon for atmosfærisk loading

Figur 4.21 viser RMS-verdiene for høydeposisjonen i perioden november-desember 2015 plottet for lengde og breddegrad for differansen mellom VMF1 korrigerert for atmosfærisk loading og VMF1 ikke-korrigerert. Blå betyr at VMF1-korrigerert er bedre enn VMF1-ukorrigerert. Det er tydelig at det gir en forbedring på innlandet og også i nord ved å korrigerer for atmosfærisk loading. En ser også at inverted barometric effect skjer ved kysten - og spesielt på oljeplattformene som gir dårligere resultat ved bruk av korreksjon.

Dersom en også korrigerer de andre metodene for atmosfærisk loading ser en i figur 4.23 forskjellene mellom a) VMF1 og MET-direkte, b) VMF1 og GPT2, c) VMF1 og VMF1-ultra og d) MET-direkte og GPT2. MET-direkte, i figur 4.23a, gir en litt dårligere RMS

enn VMF1 mot kyst og delvis innland i sør, mens det i Midt-og Nord-Norge er en trend mot litt lavere RMS.

I figur 4.23b ser en tydelig at VMF1 nå er bedre enn GPT2 for hele Norge. Forskjellen er litt mindre midt i landet, på kupertede områder. Det samme gjelder for MET-direkte og GPT2.

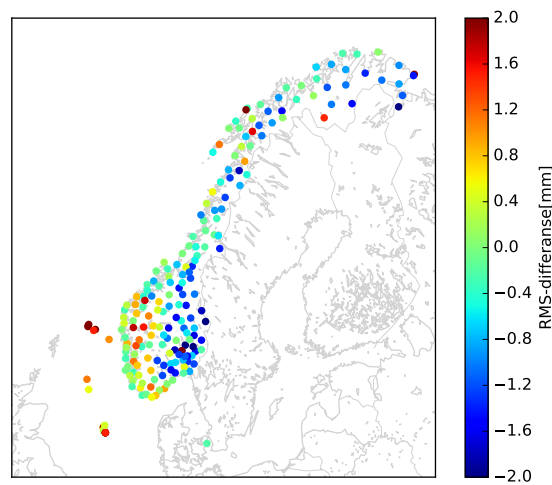
For å se på sammenhengen mellom høyden til tidsserien og atmosfærisk loading kan man lage plot for å se hvor godt de korrelerer med hverandre. Her er dette kun testet for VMF1. Dersom en ser på korrelasjonen mellom tidsserieanalysen til høyden mot verdiene for atmosfærisk loading for samme perioden får en for innlandstasjonen Trysil i figur 4.22a at de korrelerer bra, med en korrelasjon på 0.86, noe som betyr at mye av høydevariasjonen her kan skyldes atmosfærisk loading. I figur 4.22b er korrelasjonen for Tromsø, som er på 0.64. Dette er noe lavere enn for Trysil, og siden Tromsø ligger nære kysten er det logisk at inverted barometric effect virker noe mer her. Stavanger, i figur 4.22c, gir en korrelasjon på kun 0.25, som også er logisk da denne stasjonen ligger ved kysten. For å illustrere påvirkningen av inverted barometric effect kan en se på oljeplattformen Valhall som ligger midt i åpent hav. Denne stasjonen er plottet i figur 4.22d, og her er korrelasjonen på 0.03.

I figur 4.24 er tidsserien plottet for VMF1 der atmosfærisk loading er korrigerert for, sammen med den ukorrigererte tidsserien. Her ser vi lignende resultater; Stavanger i figur 4.24a gir relativt like plot med og uten korreksjon, der det stort sett enten gir like eller dårligere resultater. Ser en derimot på Trysil i figur 4.24b blir resultatene kraftig forbedret for nesten hele perioden. Tromsø i figur 4.24c blir også forbedret, spesielt fra midten og mot slutten av perioden. For disse stasjonene er det et interessant funn i slutten av desember. Her ser det ut som at det for Trysil og Tromsø er svært god forbedring med inkludering av atmosfærisk loading, mens Stavanger gir helt motsatt. Dette var også det samme som vi så når vi så på differansen mellom GPT2 og VMF1 for Tromsø og Trysil, i henholdsvis figur 4.17 og 4.16, som begge hadde en stor forskjell i slutten av desember. Helt i slutten av desember 2015 var det et stort høytrykk over Finland som preget nordlige og østlige områder av Norge. På andre siden, i England, var det et stort lavtrykk som påvirket vestlandet og førte til kraftig vind i dette område [33]. Dette kan kanskje forklare resultatene vi får her. Valhall i figur 4.24d ligger litt over den ukorrigererte tidsserien hele veien, og gir altså et dårligere resultat enn tidsserien som ikke er korrigerert for atmosfærisk loading.

MET-standard og MET-mellom med atmosfærisk loading

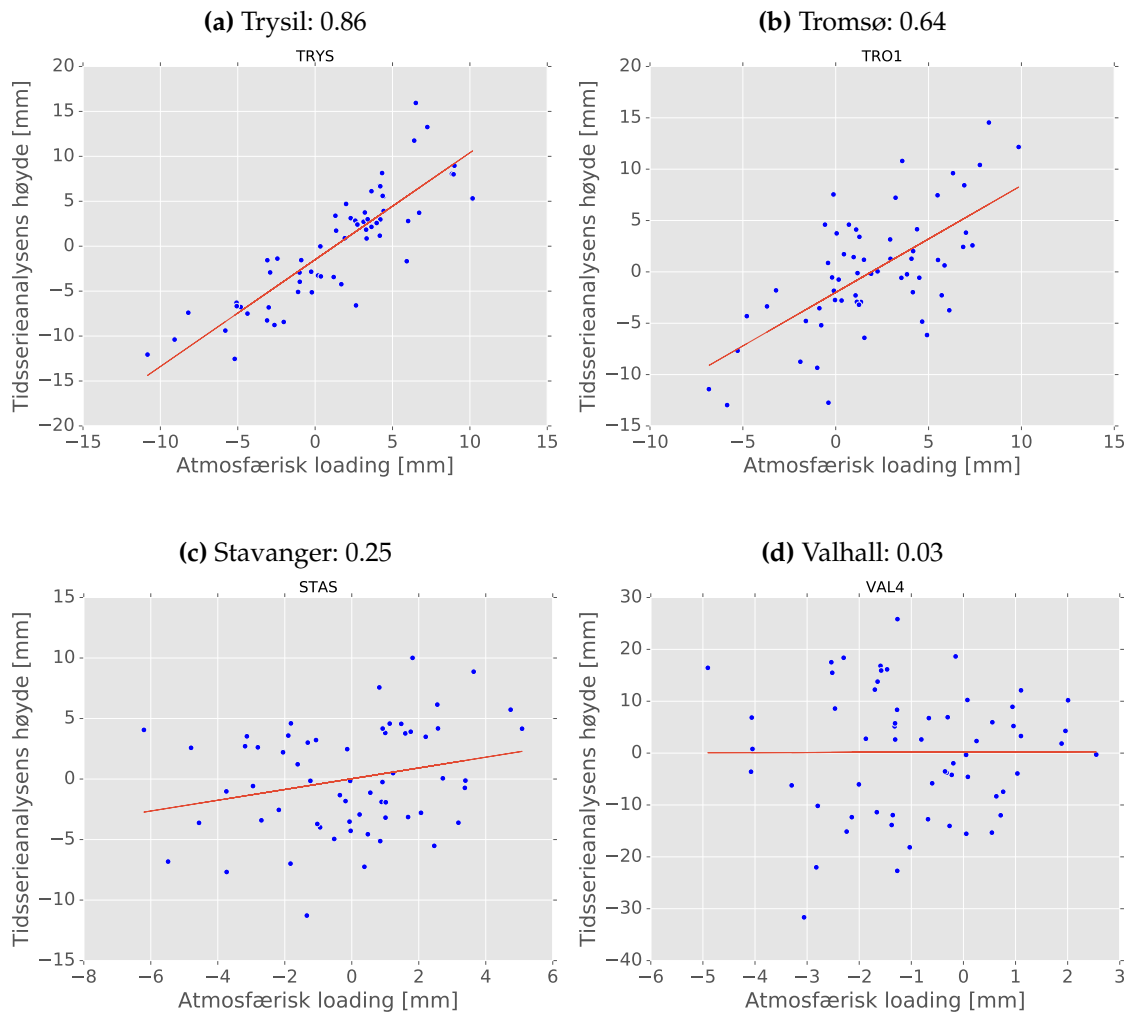
Dersom vi korrigerer alle MET-metodene for atmosfærisk loading får man også en endring i metoden som er best. I 4.25a er RMS-forskjellen til tidsserien der atmosfærisk loading er korrigerert for for MET-direkte - MET-standard plottet for lengde- og breddegrad. Rundt Røros-traktene er nå MET-direkte bedre enn MET-standard, samt for en del nordlige stasjoner. Sør-vest Norge viser at det for flere stasjoner her er MET-standard som gir bedre resultat. Det er ikke helt tydelig, men det kan kanskje se ut som at høydeforskjellen

VMF1-atmosfæriske loading - VMF1



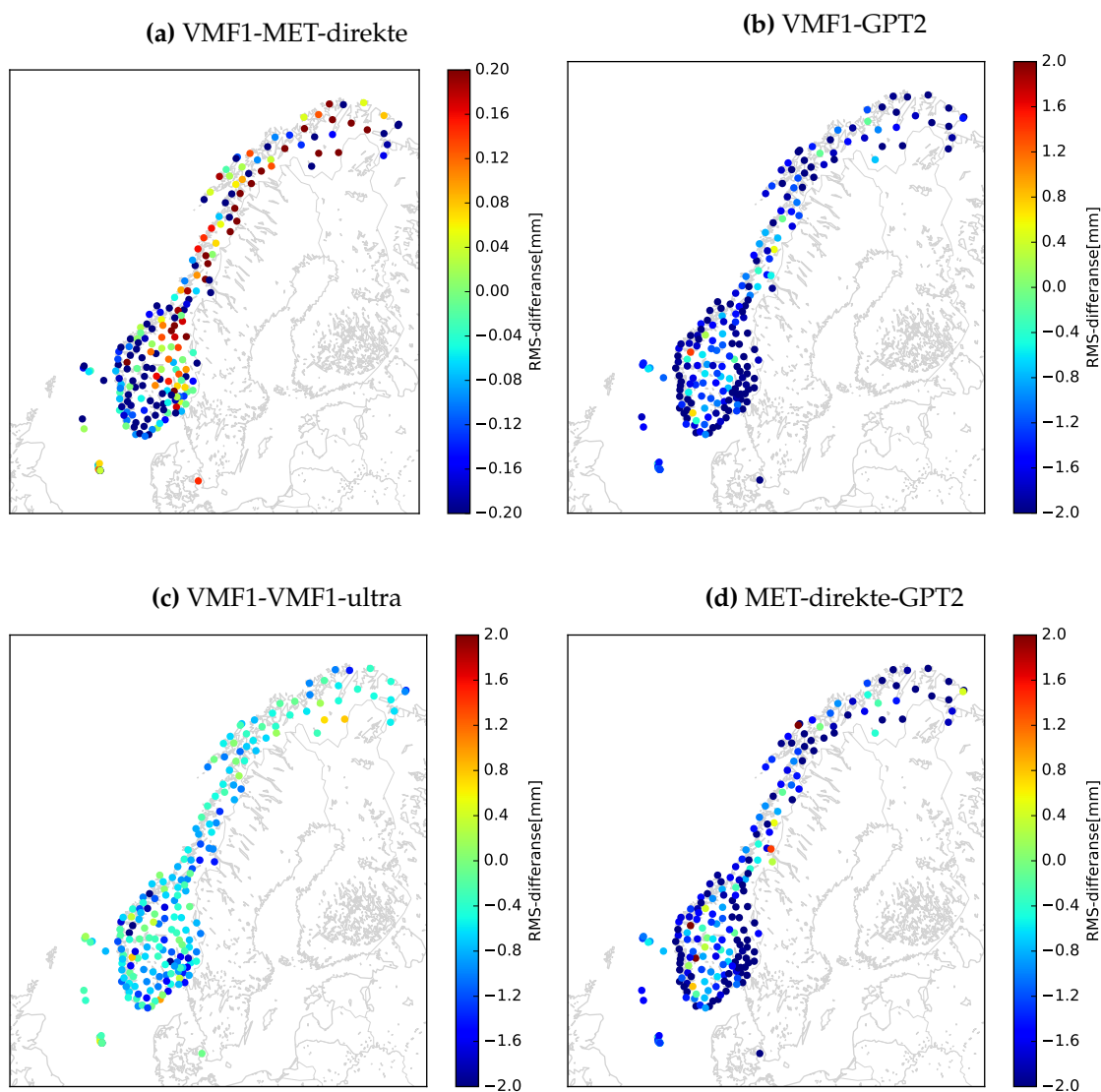
Figur 4.21.: RMS-verdien til høydeposisjonen for differansen mellom VMF1 med atmosfærisk loading korreksjon og VMF1, gitt i mm. En ser tydelig en forbedring på innlandet og i nord ved korreksjon av atmosfærisk loading. På oljeplattformene er det en helt motsatt effekt, her gir korreksjonen mye dårligere resultat.

nå spiller en større rolle når atmosfærisk loading er korrigert for. MET-mellom er også her veldig lik som MET-standard, som en ser i figur 4.25, men det kan være en antydning også her at MET-mellom gir et litt bedre resultat enn MET-standard. Dersom vi nå sammenligner VMF1 med MET-mellom ser vi at VMF1 gir det beste resultatet.

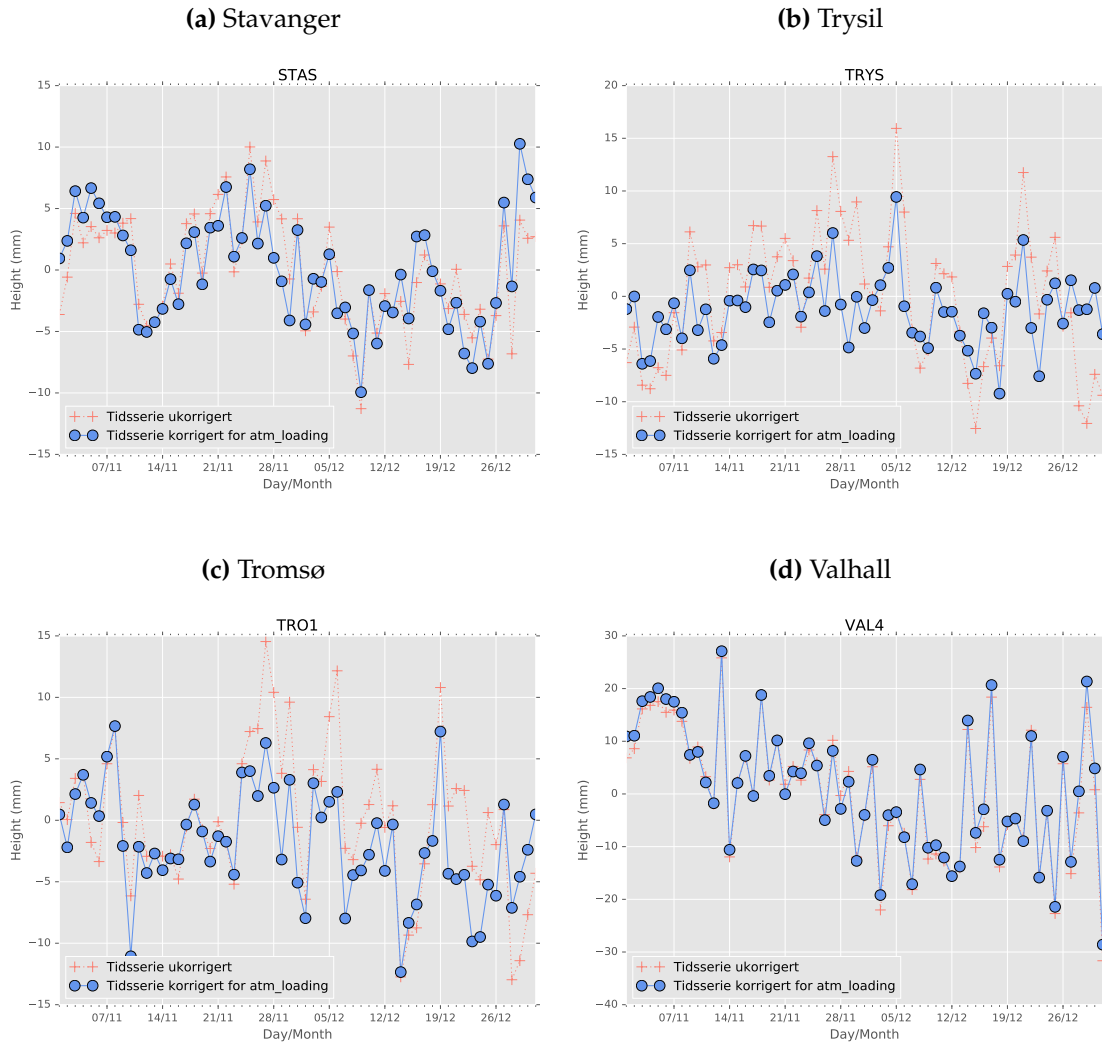


Figur 4.22.: Plottene viser korrelasjon mellom høyden fra tidsserieanalysen mot atmosfærisk loading for a) Trysil, b) Tromsø, c) Stavanger og d) Valhall. Trysil har høyest korrelasjon og Valhall har lavest.

Med atmosfærisk loading

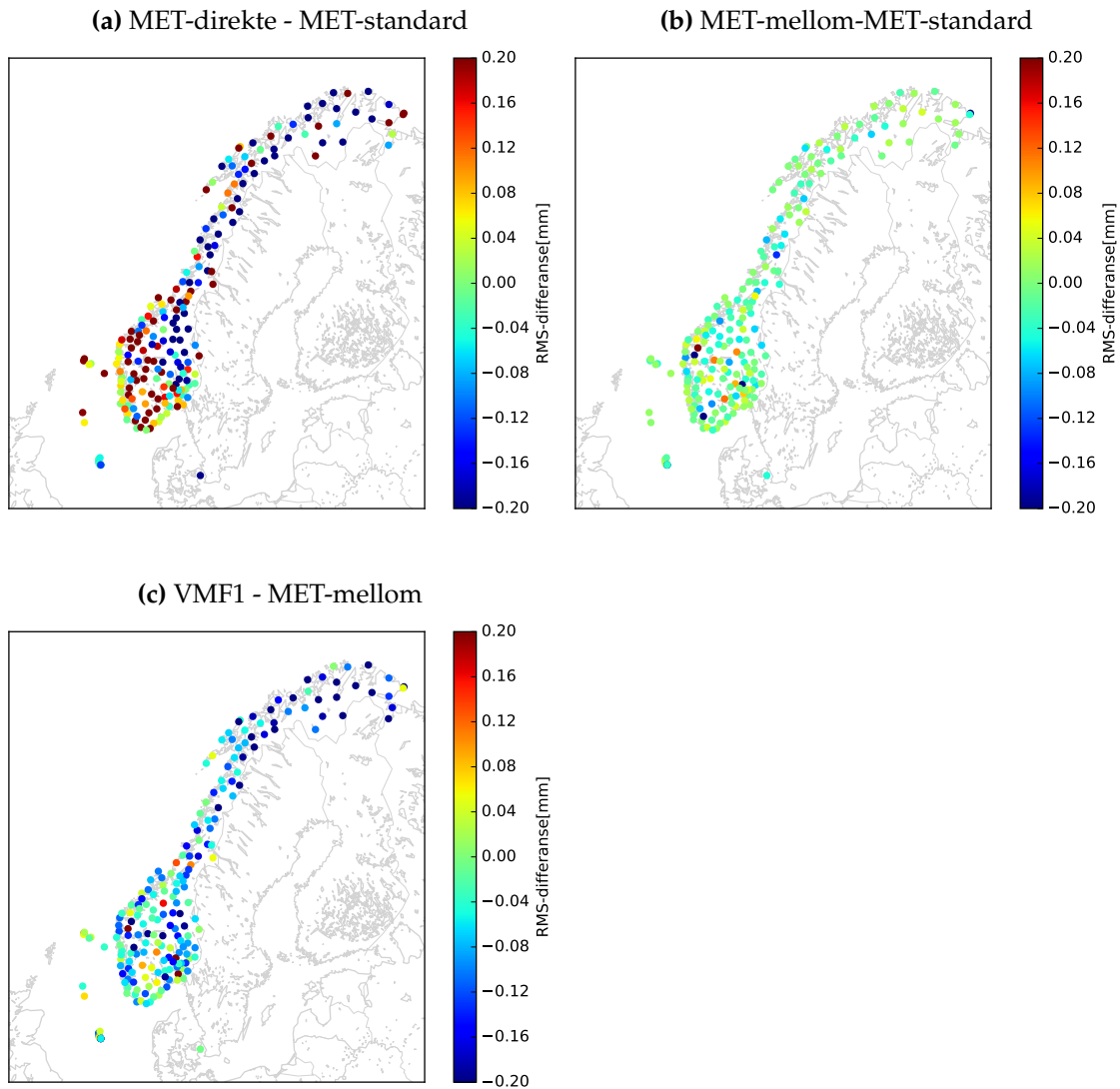


Figur 4.23.: Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember med korreksjon for atmosfærisk loading for a) VMF1 - MET-direkte og b) VMF1-GPT2, c) VMF1-VMF1-ultra og d) MET-direkte-GPT2. Begge fargeskalaene er i mm. Legg merke til at a) har skala fra -0,2 til 0,2 mm, mens b), c) og d) har -2 til 2 mm.



Figur 4.24.: Plottene viser tidsserieanalysen for VMF1 korrigert for atmosfærisk loading. Her er blå den korrigerte tidsserie, mens rød viser opprinnelig tidsserie.

Med atmosfærisk loading



Figur 4.25.: Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember med korreksjon for atmosfærisk loading for a) MET-direkte - MET-standard, b) MET-standard-MET-mellom og c) VMF1-MET-mellom. Begge fargeskalaene er i mm.

5. Resultater data-assimilering

Resultatene fra MET er for de fire metodene MET-direkte, VMF1, VMF1-ultra og GPT2. Her er altså ikke MET-mellom og MET-standard tatt med i resultatene. Grunnen til dette er at MET-standard og MET-mellom ble kjørt i GIPSY PPP etter MET-direkte, og rakk dermed ikke å bli en del av data-assimileringa. Da tiden er såpass begrenset for denne oppgaven, ble tidsperioden for assimileringen satt fra 1. november til 10. desember 2015 for GPT2, VMF1 og MET-direkte å få resultater fra alle metodene. På grunn av problemer med data-assimileringen for VMF1-ultra ble det kun resultater for perioden 1. november - 12. november for denne metoden.

I første del av resultatene fra MET sees det på brukbare data, altså hvilke stasjoner som regnes som godkjente. Dette vil variere ettersom hvor lenge data-assimileringen har pågått; flere stasjoner blir godkjente jo lengre tid det har gått. Det er interessant å se hvilke av metodene som gir raskest og flest godkjente stasjoner.

I siste del av resultatene lages en tidsserie for den totale forsinkelsen for hver stasjon. De samme stasjonene som nevnt i tabell 4.1 blir undersøkt nærmere. På denne måten kan også resultatene fra data-assimileringa sammenlignes med resultatene fra Kartverket. Det ideelle er om observasjonene etter kjøringa er mest mulig lik first-guess, fortest mulig, som beskrevet i avsnitt 3.2. Samt om observasjon-minus-first-guess følger en normalfordeling. Dette fordi det forteller oss at biasen blir korrigert fort, noe som er bra.

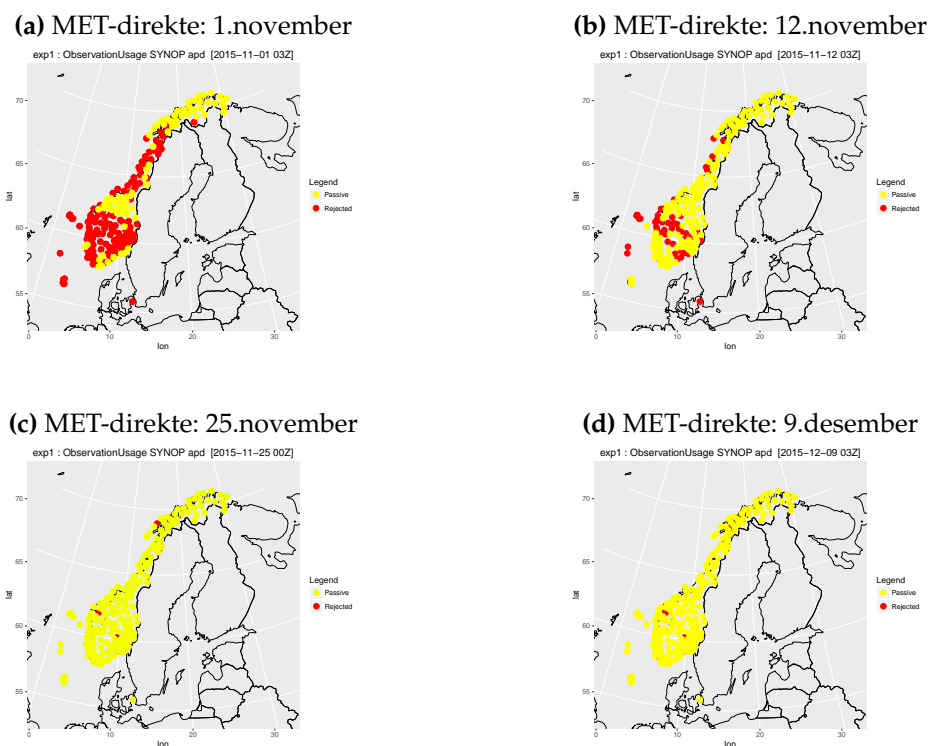
5.1. Brukbare data

For å se om kvaliteten på GPS-dataene sendt over til Meteorologisk institutt er av god kvalitet kan en blant annet se på observation usage, altså observasjoner som kan brukes. Stasjoner farget rødt kan ikke brukes, mens gule stasjoner kan det. Det blir flere og flere stasjoner som kan brukes utover perioden og det kan være interessant å se på utviklingen over tid for de ulike metodene for å se om det er en forskjell. I figur 5.1a ser vi godkjente stasjoner for metoden MET-direkte for dag 1. Her blir stasjoner i nord fort godkjent - allerede på dag 1, noe som er bra.

I figur 5.2a ser en godkjente stasjoner for GPT2 for første dag, altså 1 november 2015. En kan faktisk se at det for denne metoden er flere godkjente stasjoner enn for MET-direkte, VMF1 og VMF1-ultra. Stasjoner i Nord- og Midt-Norge ser ut til å bli fort godkjent her, samt noen oljeplattformer.

Som de andre metodene er det for VMF1, i figur 5.3a, mange godkjente Nord- og Midt

5. Resultater data-assimilering

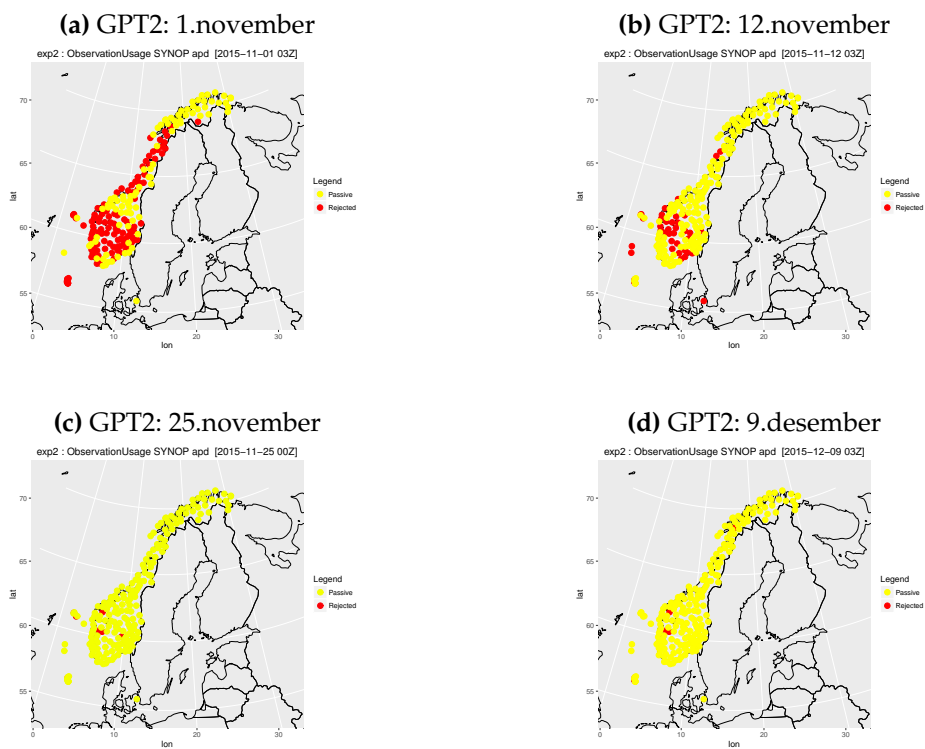


Figur 5.1.: Plottet viser stasjoner som er godkjent til bruk for MET-direkte i data-assimileringen etter a) 1.november, b) 12. november, c) 25.november og d) 9.desember. De røde stasjonene er ikke godkjente, mens gule stasjoner er godkjente. Metoden som er brukt her er MET-direkte.

Norge. MET-direkte og VMF1 gir ganske like resultater. VMF1-ultra i figur 5.4a er metoden som gir dårligst resultat den første dagen, men det er likevel mange godkjente stasjoner.

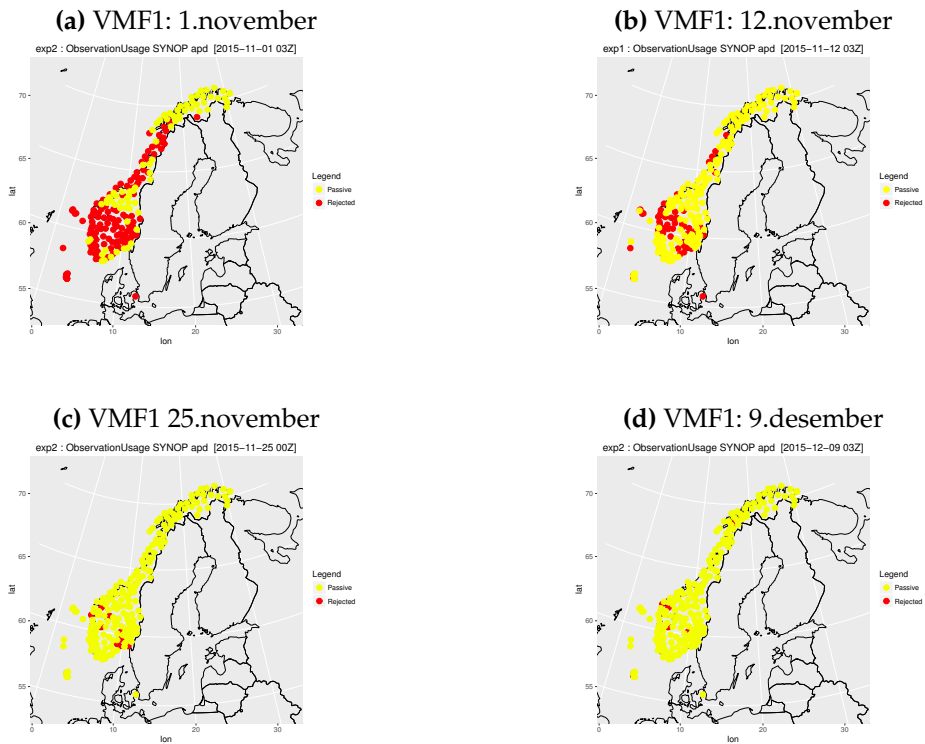
Etter 12 dager kan vi se at de fleste stasjoner er godkjente for alle metodene. GPT2 ser også ut til å gi best resultater etter 12. dager. VMF1 gir igjen litt flere godkjente stasjoner enn MET-direkte og VMF1-ultra i figur 5.4b.

25. november virker det som at antall godkjente stasjoner har stabilisert seg, og en ser at nå er også oljeplattformene godkjente for alle metodene. Det er nå ikke noe forskjell mellom metodene, som også gjelder for siste dag 10.desember.

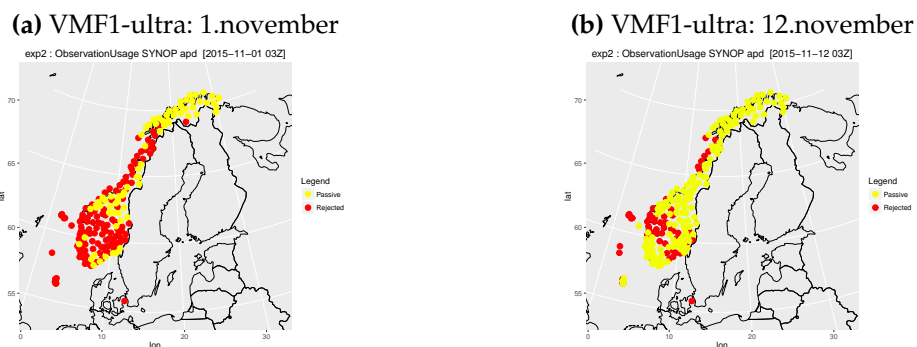


Figur 5.2.: Plottet viser stasjoner som er godkjent til bruk for GPT2 i data-assimileringen etter a) 1. november, b) 12.november, c) 25.november d) 9.desember alle for cycle 03. De røde stasjonene er ikke godkjente, mens gule stasjoner er godkjente. Metoden som er brukt her er GPT2.

5. Resultater data-assimilering



Figur 5.3.: Plottet viser stasjoner som er godkjent til bruk for VMF1 i data-assimileringen etter a) 1. november b) 12.november, c) 20.november d) 9.desember. De røde stasjonene er ikke godkjente, mens gule stasjoner er godkjente. Metoden som er brukt her er VMF1.



Figur 5.4.: Antall godkjente stasjoner i data-assimileringa for VMF1-ultra for a) 1. november og b) 12. november.

5.2. Tidsserieanalyse MET

For å se om bias-korrekasjonen fungerer bra på GNSS-dataene kan en se på en tidsserie over perioden. Er korrekasjonen god nærmer observasjonene seg first-guess modellen, noe som er ønskelig. I figur 5.5 kan en se en tidsserie samt fordeling av observasjoner minus first-guess for MET-direkte i 5.5a, VMF1 i 5.5b og GPT2 i 5.5c for Stavanger. Det er gunstig dersom fg-dep plottet følger en normalfordeling, noe en kan se at VMF1 i figur 5.5b gjør best av de ulike metodene her. En ser også at normalfordelinga ikke går rundt 0, men ca 0,015, noe som betyr at observasjonene systematisk ligger litt over first-guess. For alle metodene nærmer observasjonene seg first-guess rundt 25. november.

For Trysil i figur 5.6 ser en at GPT2 gir best normalfordeling, i 5.6c. Observasjonene for alle metodene nærmer seg, også her, first-guess modell rundt 25. november.

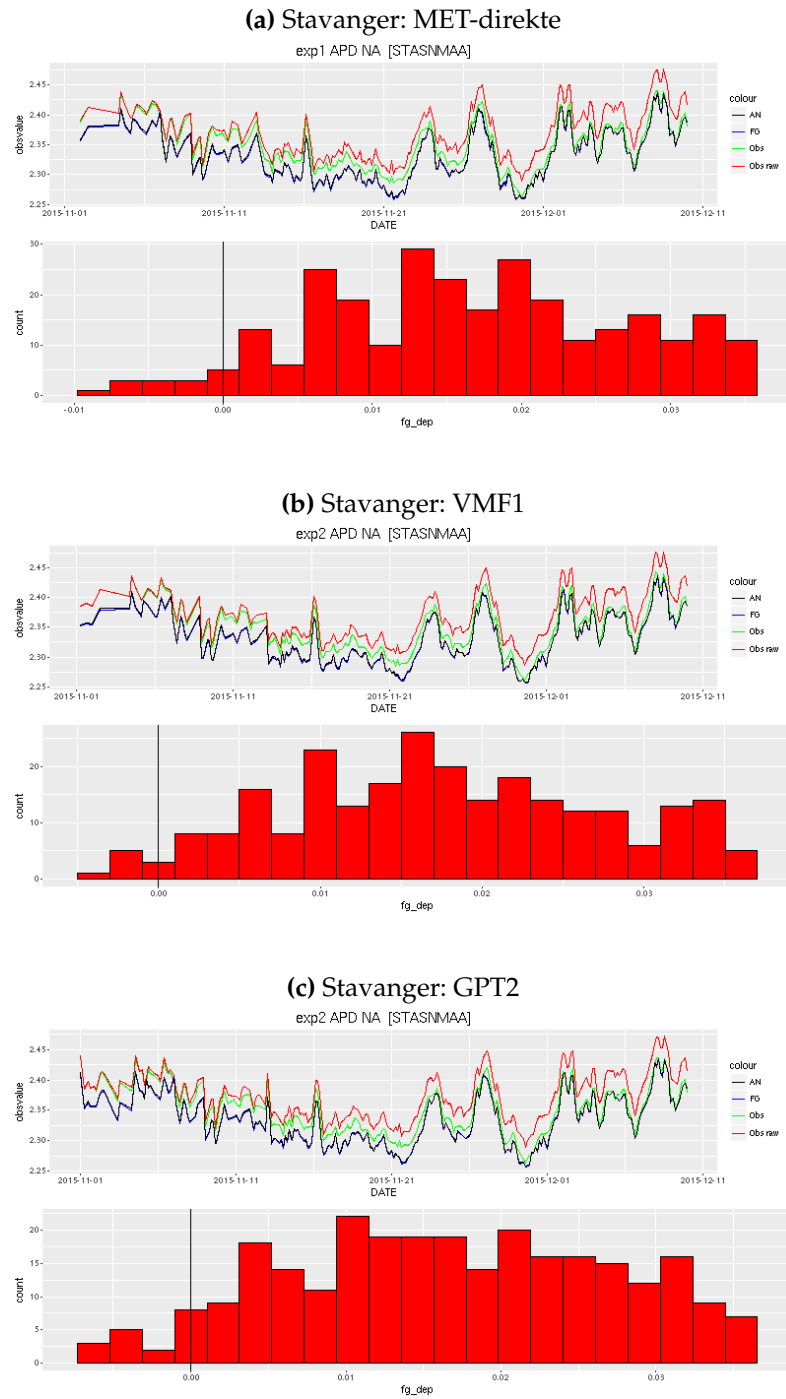
I figur 5.7 ser vi resultatene for Tromsø. Her ser en generelt at resultatene er gode, da alle metodene har en god normalfordeling for observasjon-minus-first-guess. VMF1, i plot 5.7b, er den som gir best resultat her. Det gode resultatet gjelder for mange stasjoner i nord, noe som også samsvarer med at mange nordlige stasjoner ble godkjente allerede 1. dagen, i figur 5.3a.

Meteorologene er generelt veldig interessert i data fra oljeplattformene og i figur 5.8 er et eksempel på en oljeplattform, Valhall. Her er det generelt stor variasjon i ZTD og resultatene er helt ok. Det er generelt litt større avstand mellom observasjonene og first-guess modellen. Valhall gir et relativt godt resultat sammenlignet med andre oljeplattformer, men det er fortsatt ikke spesielt bra. VMF1 i figur 5.8b gir det beste resultatet.

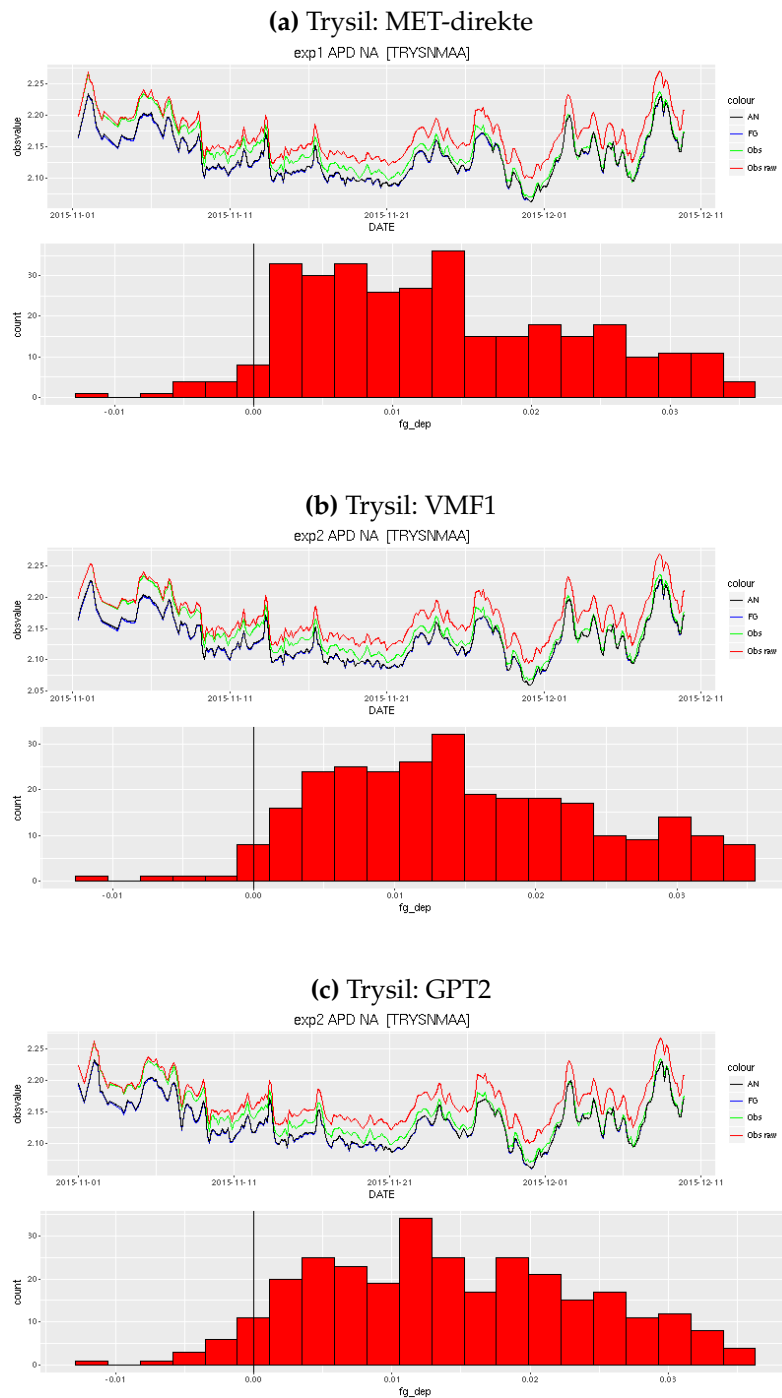
Et interessant funn er stasjonen Lofthus, i figur 5.9, som har vært preget av et høydeforskjellproblem. Dataene sendt til MET er helt uten høydekorrekasjon, og MET-direkte skulle i teorien gi det dårligste resultatene, da denne ikke er korrigerert for riktig høyde. Her får vi et helt motsatt resultat; både VMF1, 5.9b, og GPT2, 5.9c, gir svært dårlige resultater og observasjonene nærmer seg ikke first-guess i det hele tatt. MET-direkte gir heller ikke et godt resultat, men ligner mer på en normalfordeling enn de andre.

Resultatene for VMF1-ultra vises i figur 5.10 for perioden 1.-12. november, for a) Stavanger, b) Trysil og c) Tromsø. Også for VMF1-ultra nærmer observasjonene seg first-guess-modellen fort for Tromsø. Stavanger gir dårligst resultat for VMF1-ultra av stasjonene gitt i figur 5.10. Det er ikke lett å se hvor godt resultat VMF1-ultra gir, siden analysen er kjørt for en så kort periode. Det kan se ut som, dersom man sammenligner med resultatene for de andre stasjonene for de samme antall dagene for de andre metodene, at det ikke er veldig stor forskjell mellom dem. Det kan likevel ikke konkluderes med noe i denne oppgaven, da perioden er for kort til å kunne si noe særlig om metoden.

5. Resultater data-assimilering

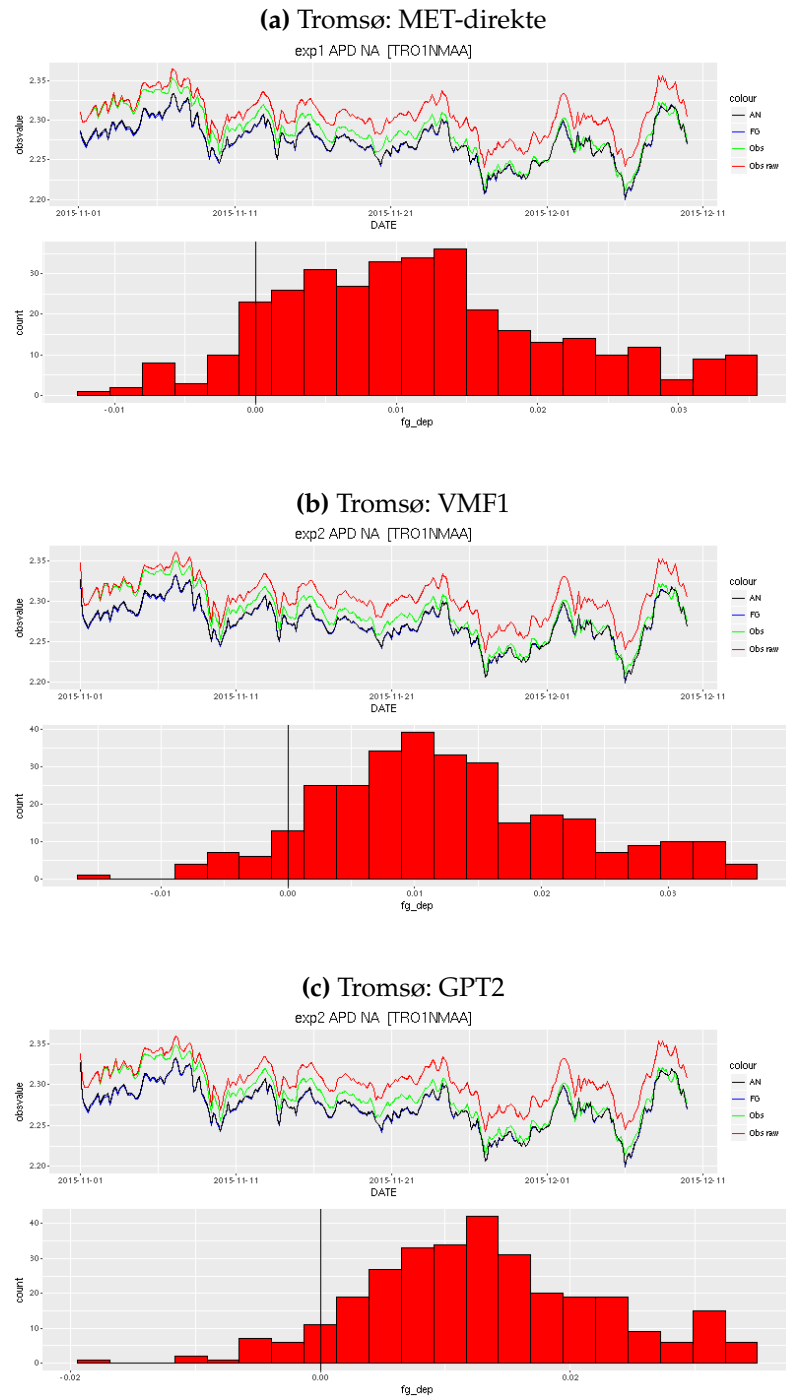


Figur 5.5.: Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Stavanger i perioden 1.november-10.desember, nedre del fg-dep, er first-guess-observasjoner. a) Viser MET-direkte, b) VMF1 og c) GPT2. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerede observasjoner og blå er first-guess.

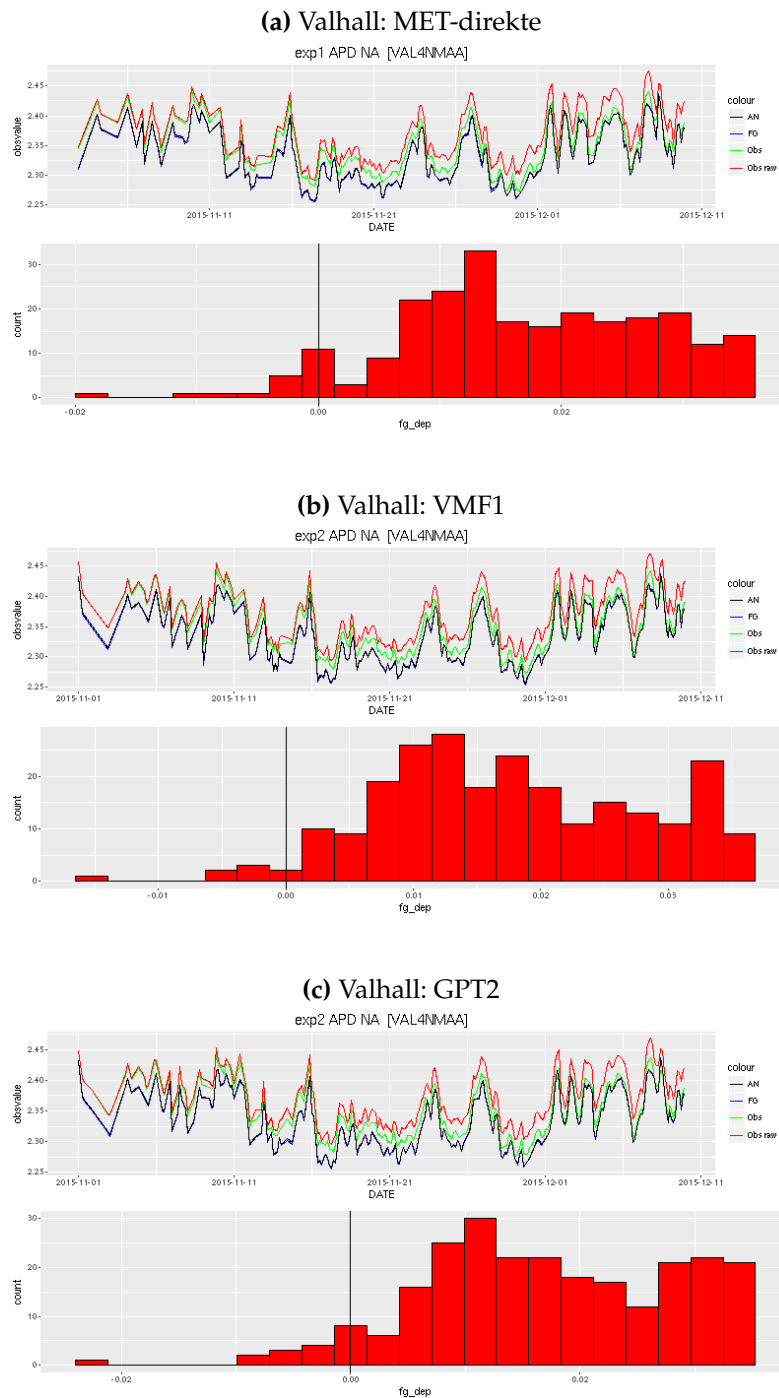


Figur 5.6.: Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GNSS-dataene for Trysil i perioden 1.november-10.desember, nedre del fg_dep, er first-guess-observasjoner. a) Viser MET-direkte, b) VMF1 og c) GPT2. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerte observasjoner og blå er first-guess.

5. Resultater data-assimilering

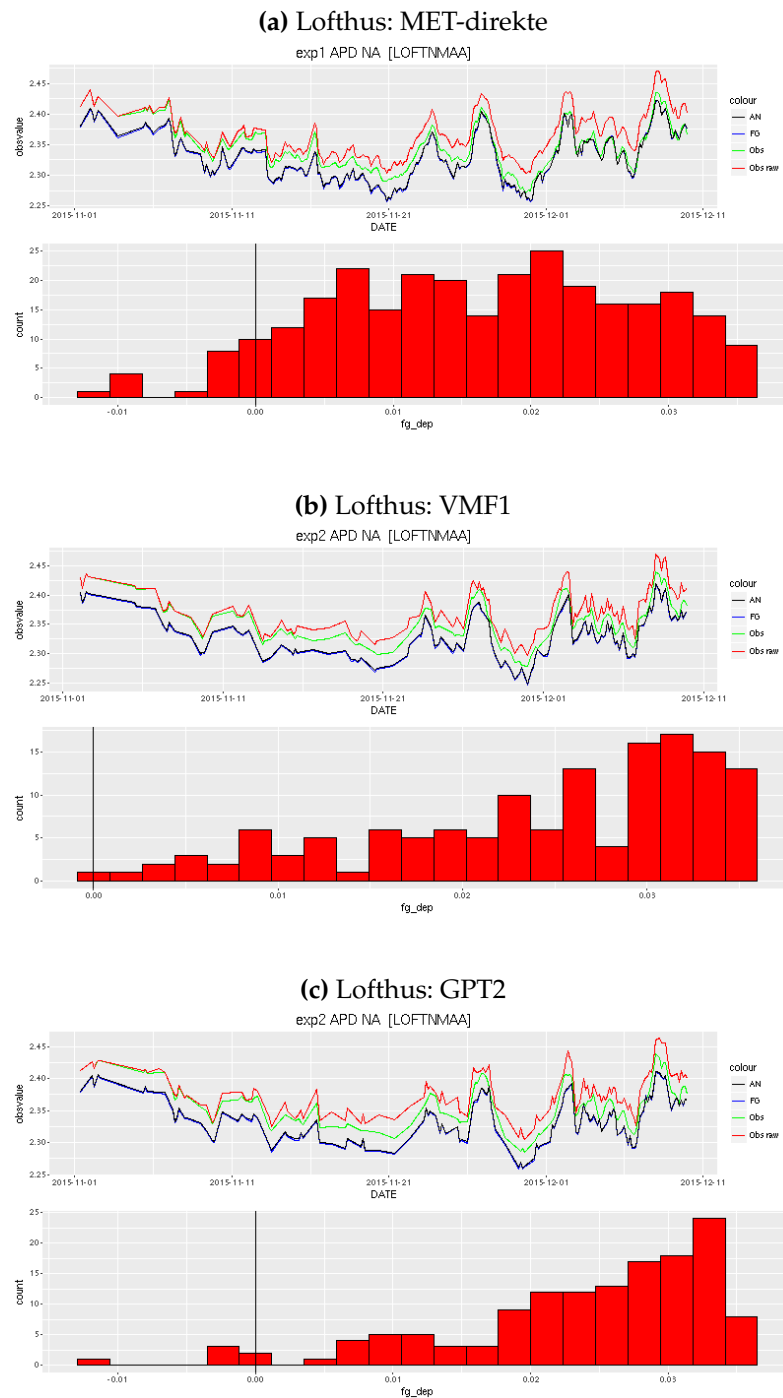


Figur 5.7.: Plottet øverst viser tidsserien for hvor godt VarBC korrejerer GNSS-dataene for Tromsø i perioden 1.november-10.desember, nedre del fg_dep, er first-guess-observasjoner. a) Viser MET-direkte, b) VMF1 og c) GPT2. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerte observasjoner og blå er first-guess.

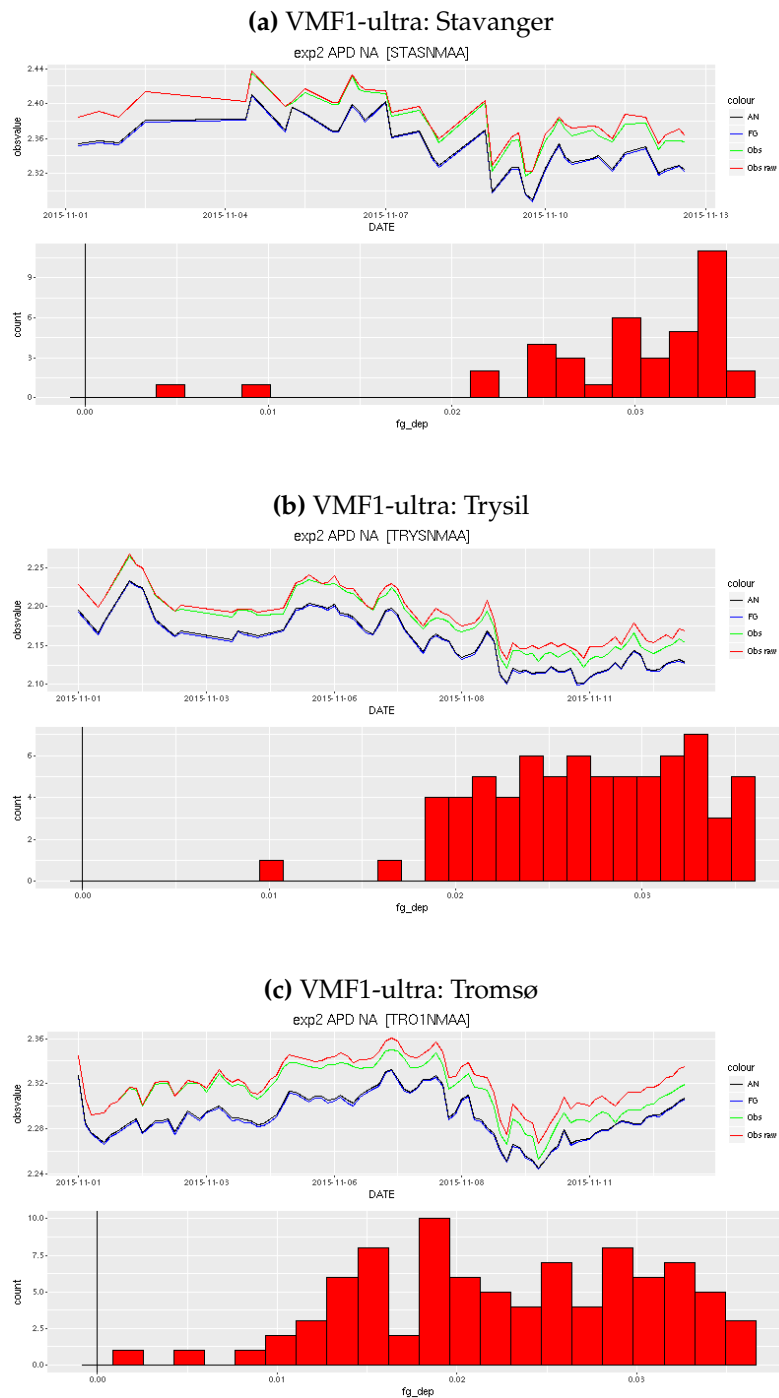


Figur 5.8.: Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for oljeplattformen Valhall i perioden 1.november-10.desember, nedre del fg_dep, er first-guess-observasjoner. a) Viser MET-direkte, b) VMF1 og c) GPT2. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerte observasjoner og blå er first-guess.

5. Resultater data-assimilering



Figur 5.9.: Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Lofthus i perioden 1.november-10.desember, nedre del fg-dep, er first-guess-observasjoner. a) Viser MET-direkte, b) VMF1 og c) GPT2. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerte observasjoner og blå er first-guess.



Figur 5.10.: Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GPS-dataene metoden VMF1-ultra for perioden 1.november-12.november. a) Viser Stavanger, b) Trysil og c) Tromsø. Y-aksen i øverste plot er i meter. Rød er opprinnelige observasjoner, grønn er korrigerte observasjoner og blå er first-guess.

6. Diskusjon

Forskjellene mellom metodene når det kommer til hydrostatisk, våt og total forsinkelse viser at det tydelig er forskjell på de ulike metodene. For dataene hentet fra MET ser vi at forsinkelsene er avhengig av høyden brukt på stasjonen og det bør derfor for videre studier korrigeres for dette. Spesielt dersom en skal se på integrated water vapor, som baserer seg på våt forsinkelse. Her vil IWV bli ganske mye feil på grunn av dette. Den totale forsinkelsen er mindre påvirket av høydeforskjellen, men den er stor nok til å utgjøre en forskjell, både for data-assimilering og posisjonsberegning. MET-mellom og MET-standard viste begge at de hadde en finere tidsoppløsning for a priori ZHD. Dette var imidlertid ikke like tydelig for ZWD, eller ZTD. Det kan være interessant ved videre studier å se på om endringen av integrert hvit støy paramteren gjør at forskjellene blir større, da den stokastiske prosessen vil kunne tillates å variere mer. Det er av Selle og Desai(2016) [34] sagt at integrert hvit støy-paramteren på $5 \cdot 10^{-8} \frac{\text{km}}{\sqrt{\text{s}}}$ brukt ved kvadratrot-sinus-vekting (SQRTSIN), som er brukt i denne oppgaven, kanskje ikke er den beste å bruke, slik at det senere kan være interessant å se om en endring av denne det gir noe utslag for resultatene.

Total forsinkelse er mindre ved Trysil enn ved Stavanger, mens den våte forsinkelsen ikke viser spesielt stor forskjell. Det er med andre ord trykkforskjellene som utgjør den store forskjellen mellom forsinkelsen til stasjonene. Forskjellen mellom forsinkelsen til VMF1 og GPT2 ser ut til å være breddegradsavhengig, der GPT2 generelt har en større total forsinkelse i nord enn i sør, sammenlignet med VMF1. Dette kan kanskje være grunnen til at GPT2 er bedre i nord enn i sør når atmosfærisk loading ikke er inkludert.

Stasjoner nær kysten og spesielt oljeplattformer har større daglig variasjon i forsinkelsene, noe som er naturlig fordi det på slike steder er større variasjon i forsinkelsen grunnet stor variasjon i trykk og vandampinnholdet. Det er generelt mye og raskt endrende vær-situasjoner på slike steder, noe som forklarer dette godt. Stormen Synne kom innover Sør-Norge i perioden 4.-7. desember, og ble spesielt god merket i Rogaland. Dette kunne man tenke seg at ga et utslag i forsinkelsene, men som vi ser for Stavanger ga det kun en liten antydning til høyere forsinkelse i denne perioden.

Utslagene a priori informasjon hadde på resultatene i tidsserieanalysen viser at VMF1 generelt gir et bedre resultat langs kysten, når RMS-verdiene til høyden ble sett på. Når korrigeringa for atmosfærisk loading ble inkludert forandret resultatene seg, og VMF1 ble forbedret på innlandet og mot nord, med en reduksjon i RMS på 30-40%. GPT2 ble på motsatt side dårligere ved atmosfærisk loading korreksjon, som bekrefter lignende observasjoner gjort av Kouba (2009) [35] og Steigenberger (2009) [20]. Forklaringen på dette kan være, i følge Kouba(2009) [35] at GPT2 har et gjennomsnittlig trykk for en

periode og ikke har inkludert hyppige variasjoner. Denne modellfeilen av trykket har vist seg å kompensere for atmosfærisk loading, spesielt ved høyere breddegrader, og en tilleggskorreksjon av denne vil dermed føre til et dårligere resultat, som vi også ser her. Dette kan også forklare hvorfor GPT2 på en del steder ga bedre resultat enn VMF1 på innland og i nord når atmosfærisk loading ikke var inkludert. Sammenligningen av VMF1 og MET-direkte er nå interessant - fra at forskjellen mellom dem var liten og så ut til å være relativt tilfeldig i figur 4.13a ble den i figur 4.23a mye større. Vi kan i tillegg se at VMF1 blir bedre langs kysten, mens MET-direkte blir litt bedre på innland og mot nord. I Sør-Norge kan vi også se at stasjoner der VMF1 er mye bedre enn MET-direkte er mange av de samme stasjonene som har store modell-høydeforskjeller i figur 3.1. Grunnen til de små forskjellene tidligere kan med andre ord skyldes at en større feil, atmosfærisk loading, påvirket resultatene. Det kan være interessant å se hvordan resultatene blir dersom en også inkluderer inverted barometric effect langs kysten.

Når vi også ser på de ulike metodene for a priori informasjon fra MET-direkte, MET-standard og MET-mellom kan vi her se at forskjellene ble mye større når atmosfærisk loading ble korrigert for; MET-direkte ble bedre på innland og mot nord, mens de andre metodene ble bedre mot sør-vestlandet. Det kan kanskje se ut som at manglende høydekorreksjon gjør at MET-standard og MET-mellom gir bedre resultater enn MET-direkte. Fordi metodene viste et generelt dårligere resultat ut mot kysten, grunnet inverted barometric effect kan det være interessant dersom man i tillegg inkluderer inverted barometric effect på kyst-stasjoner, for å se hva forskjellene blir da. Sammenligner man MET-standard og MET-mellom er forskjellene som en ser i figur 4.25b minimale, og det er snakk om en forskjell på 10^{-2} mm. Det kan likevel se ut som MET-mellom er litt bedre enn MET-standard en del steder. Som sagt er likevel RMS-verdien ikke en fasit på god metode, og høydekorrigeringa for MET-direkte bør skje, da vi så at høydeposisjonen i tidsserieanalysen fikk en forskyvning. Her ser det altså ut som MET-mellom er den beste metoden å bruke for høydekorrigering, men til videre samarbeid mellom MET og Kartverket bør dette testes nærmere. Sammenligningen av MET-mellom med VMF1 før korreksjon for atmosfærisk loading viste at MET-mellom ga en antydning til litt bedre RMS-resultater, mens det etter korreksjonen var motsatt.

Fra tidsseriene kan vi se at VMF1, GPT2 og MET-direkte gir en forskjell i høyde-residualene. Forskjellen mellom VMF1 og MET-direkte ligger rundt 2 mm, mens det for VMF1 og GPT2 kan være opp mot 10 mm, noe som er ganske mye. Viktigheten av god a priori informasjon om den hydrostatiske forsinkelsen blir tydelig når vi ser på MET-direkte sammenlignet med VMF1: MET-verdiene er forskjøvet, og for Lofthus er det en forskyvning på hele 11 mm. Feilmodelleringa av MET-direktes hydrostatiske forsinkelse gir med andre ord store utslag på høyde-residualene, og bør derfor korrigeres for. Når vi ser på forskjellene mellom VMF1 og MET-mellom er disse mye mindre, og ligger rundt 0.5 – 1.0 mm, avhengig av hvilken stasjon man ser på. Valhall ga foreksempel en forskjell på rundt 2 mm.

Når man ser på vertikal posisjon bør man på innlandet og i nord ta hensyn til atmosfærisk loading for å få optimale resultater. van Dam et. al.[24] fikk for Tromsø en korrelasjon på 0.42 mellom GPS stasjonshøyde-residualer og atmosfærisk loading, mens dette

forsøket ga en korrelasjon på 0.64. Analysen gjort her baserer seg på et kortere tidsrom enn van Dam et. al. og modellen for atmosfærisk loading er heller ikke den samme, noe som kan være grunner til forskjellene. Perioden jeg har sett på, høsten, er ofte preget av mer vær og dermed også større trykksystemer, som kan være årsaken til at atmosfærisk loading påvirker mer her enn i van Dams forsøk. Det at det kom et stort høytrykk over Skandinavia i slutten av desember, mens det var et lavtrykk over England, kan kanskje se ut til å gi utslag når man ser på tidsserien for høydeposisjonen når atmosfærisk loading er korrigert for. Det interessante var også at for tidsseiredifferansen mellom VMF1 og GPT2 får vi store forskjeller for de samme periodene som atmosfærisk loading påvirker mye. Dette kan kanskje vise hvordan GPT2 delvis korrigerer for atmosfærisk loading, som gjør at forskjellene blir større i denne perioden. Til senere studier kan det være lurt å se på de andre modellene for atmosfærisk loading som er tilgjengelig, for å se om de gir bedre korreksjon. Det er også slik at det i denne oppgaven ble laget et gjennomsnitt pr.dag over atmosfærisk loading, og så korrigert i høyde-tidsserien pr. dag. For et enda mer nøyaktig resultat kan det med fordel bli korrigert mer nøyaktig enn fra dag til dag. I og med at atmosfærisk loading kommer for hver 3.time, kan en korreksjon for hver 3. time gi enda bedre resultater, dette kan kanskje også endre forskjellene mellom VMF1 og MET-mellom.

Av resultatene fra meteorologisk institutt ser vi at de ulike metodene har en relativ lik utvikling av godkjente stasjoner, der stasjoner i Nord-og Midt-Norge fort blir godkjent. En grunn til dette kan være at det i nord er mer stabilt trykk. Dersom en ser på total forsinkelse i senit for Tromsø, i figur 4.9c, ser en at de daglige variasjonene her er mindre enn for Trysil og Stavanger. Dette kan være grunnen til at data-assimileringa hos MET fungerer bra her. Stasjoner i nord har også en bra gauss-kurve, som betyr at stasjonene gir bra kriterier for data-assimilering. En annen grunn til dette kan kanskje også være at modellen som den totale forsinkelsen sammenlignes med blir dårligere jo lengre nord du kommer [32]. Fordi modellen kan være dårligere, blir forsinkelsen fortere lik modellen enn dersom modellen er god. Det kan altså være at det ikke er den totale forsinkelsen mot nord som nødvendigvis er bedre, men at det er modellen som er dårligere.

Det er generelt VMF1 som ser ut til å gi best resultat når vi ser på normalfordelinga. Stasjonen Lofthus gir svært dårlige resultater for VMF1 og GPT2, mens MET-direkte gir noe bedre. Dette kan tenkes at skyldes mangel av høydekorrsjon, selvom min intuitivitet sa meg at VMF1 og GPT2 skulle gi bedre resultat. Dette bør sees videre på ved et videre samarbeid mellom Meteorologisk institutt og Kartverket. En kan faktisk også muligens se en trend i at for en del innlandsstasjoner, her representert ved Trysil, gir GPT2 et bedre resultat enn de andre metodene - noe som kanskje kan skyldes at dataene ikke er korrigert for atmosfærisk loading.

Generelt kan det se ut stasjoner med store variasjoner fra dag til dag, slik som på oljeplattformene, gir dårligere resultat i data-assimileringen, mens Tromsø, med mindre variasjon gir bra resultater for alle metodene. Det ser også ut som at MET-direkte ikke gir de beste resultatene for noen av de utvalgte stasjonene. Dette kan muligens endres dersom en tar hensyn til høydeforskjellen.

Data-assimileringa for VMF1-ultra ble for kort til å si noe om hvor godt denne metoden

6. Diskusjon

fungerte. Det ser likevel lovende ut, da metoden virker nesten like god som de andre metodene var etter 12 dager, dersom vi ser på brukbare data og tidsserieanalysen. Dette er noe som bør sees videre på i et samarbeid mellom MET og Kartverket.

Til videre studier anbefaler jeg å se nærmere på: høydekorrigering og korrigering for atmosfærisk loading som en del av programvare for beregning av ZTD for å se hvilket utslag dette gjør.

7. Konklusjon

Etter å nå ha sett på våt, hydrostatisk og total forsinkelse for metodene kan vi se at trykkforskjellene mellom stasjonene er det som gir den reelle forskjellen. Det er trykkelige og de store værssystemene som påvirker forsinkelsene mest, lokale vær-variasjoner har mindre betydning. På grunn av dette er korrigering for atmosfærisk loading viktig, for så å eventuelt kunne se disse mindre variasjonene. Dette ble tydelig for VMF1 og MET-direkte etter at atmosfærisk loading ble fjernet fra tidsserien. Det burde testes å implementere denne korrigeringa som en del av programvaren GIPSY, på samme måte som for eksempel ocean loading, da ZTD også blir påvirket av dette. Da bør også invertet barometric effekt vurderes som en del av dette, siden denne effekten er stor langs kysten. Et klart svar på de ulike metodenes forskjeller i forsinkelse kan ikke helt la seg diskutere i denne oppgaven, men vi kan se at MET-direktes høydeforskjell har en direkte påvirkning på forsinkelsene og posisjon. GPT2s forsinkelse sammenlignet med VMF1 er breddegradsavhengig, der det er større total forsinkelse i nord enn i sør. Forskjellen mellom MET-standard og MET-mellom er minimale og hvilken av metodene som brukes for korrigering av høyden har ikke veldig mye å si. Likevel har MET-mellom en svak tendens til å gi litt bedre resultater enn MET-standard. Det ser også ut som MET-mellom og MET-standard har en litt finere tidsoppløsning enn VMF1 for ZHD, men at denne ikke er veldig tydelig på den totale forsinkelsen.

Uten atmosfærisk loading kan det se ut som at VMF1 blir de beste metoden langs kysten, mens det på innlandet er mer tilfeldig. MET-direkte ser ut til å gi et totalt det beste resultat på innlandet etter atmosfærisk loading er korrigert for, mens VMF1 gir lavest RMS langs kysten og steder der MET-direkte er sterkt påvirket av at høyden ikke er korrigert for. MET-direktes RMS-verdi er basert på en forskjøvet regressjonslinje og vil derfor være misvisende, da modellen i utgangspunktet er feil. MET-dataene bør med andre ord korrigeres for høyden. MET-mellom sammenlignet med VMF1 gir litt bedre resultater før korreksjon av atmosfærisk loading, mens det etter korreksjonen var VMF1 som ga et bedre resultat. GPT2 gir bedre resultater før korreksjon for atmosfærisk loading, og er metoden som gir høyest RMS-verdier dersom den inkluderes.

Data-assimilering for ZTD på MET viser samme trend som resultatene fra Kartverket. GPT2 syntes å virke bedre på stasjoner som også gir lavest RMS-resultat for denne metoden, slik som innlandsstasjonen Trysil. Langs kysten er resultatene dårligere, sannsynligvis grunnet store daglige variasjoner i trykk og vær. De nordlige stasjonene gir bedre resultater en sørlige, noe som kan skyldes et mer stabilt trykk i nord i forhold til sør, men også at modellen som den totale forsinkelsen sammenlignes med er dårligere i nord enn lenger sør. Alt i alt ser det lovende ut med bruk av ZTD i dataassimileringen, og mange stasjoner gir gode resultater for bias-korrigering. Bruk av VMF1-ultra

7. Konklusjon

kan også se ut til å gi liknende resultater som de andre metodene, men det er vanskelig å se på det korte tidsrommet analysen ble kjørt. Det anbefales å se nærmere på dette for MET.

Dersom et samarbeid skal fortsette bør en altså se nærmere på atmosfærisk loading og høydekorreksjon for å oppnå best mulig resultat - både for Meteorologisk institutt og Kartverket.

Figurer

1.1.	Figuren viser en skjematisk oversikt over oppgaven.	2
2.1.	Figuren viser hvordan en stråle brytes ulikt gjennom lag med ulike brytningsindeks, og følger av Fermats prinsipp.	5
2.2.	Figuren viser hvordan Kalmanfilteret er rekursivt. Resultater fra prediksjonen blir brukt i korreksjonen, som igjen blir brukt til en ny prediksjon osv.	14
2.3.	Figuren viser den stokastiske modellen integrert hvit støy.	17
3.1.	Differansen mellom korrekt stasjonshøyde og modellhøyden til MET er her plottet for lengde og breddegrad.	20
3.2.	COST-format til Ås i 1. november de den første timen.	23
3.3.	Figuren viser hvordan de ulike metodene bruker samme bakgrunnsmodell, first-guess, for å se om modell og observasjoner samsvarer.	25
4.1.	Figuren viser hvor stasjonene i tabell 4.1 ligger plassert i Norge.	28
4.2.	Differansen mellom gjennomsnittlig hydrostatisk forsinkelse for a) VMF1-MET-direkte, b) VMF1-GPT2, c) VMF1-MET-standard og d) VMF1-MET-mellom plottet for lengde- og breddegrad	30
4.3.	A priori hydrostatisk forsinkelse for a) Stavanger, b) Trysil, c) Tromsø og d) Valhall og metodene VMF1, GPT2 og MET-direkte	31
4.4.	A priori hydrostatisk forsinkelse for Lofthus. I a) brukes MET-direkte i b) MET-standard, c) MET-mellom og d) MET-mellom for et kortere tidsrom.	32
4.5.	Den estimerte våte forsinkelsen for VMF1, GPT2, VMF1-ultra og MET-direkte. Her for a) Stavanger, b) Trysil, c) Tromsø og d) Valhall. Perioden er november-desember 2015.	34
4.6.	Våt forsinkelse for Lofthus. I a) brukes MET-direkte i b) med standard atmosfære modell, c) mellom-metoden og d) MET-mellom for en kort tidsperiode. VMF1, GPT2 og VMF1-ultra er også plottet for perioden november-desember 2015.	35
4.7.	Differansen mellom gjennomsnittlig våt forsinkelse for a) VMF1-MET-direkte b) VMF1-GPT2 c) VMF1-MET-standard og d) VMF1-MET-mellom.	36
4.8.	Forskjellen mellom gjennomsnittlig total forsinkelse plottet i et kart for a) VMF1-MET-direkte b) VMF1-GPT2 og c) VMF1-MET-mellom.	38
4.9.	Total forsinkelse i senit plottet for GPT2, VMF1 og MET-standard for stasjonene Stavanger, Trysil, Tromsø og Valhall i perioden november-desember 2015.	39

4.10. Total forsinkelse for Lofthus for november-desember 2015 for GPT2, VMF1 samt, for a) MET-direkte, B) MET-standard c) MET-mellom. I d) er forskjellen mellom MET-standard og MET-mellom plottet.	40
4.11. Forskjellen mellom total forsinkelse for MET-standard og MET-mellom plottet for november-desember 2015.	41
4.12. RMS-verdier til høyden for stasjoner i Norge for a) VMF1 b) GPT2 c) VMF1-ultra og d) MET-direkte.	42
4.13. Differansen mellom RMS-verdiene til lineære tidsserieanalysen for høyden for a)VMF1-MET-direkte b)VMF1-GPT2 c) MET-direkte-GPT2 og d) VMF1-VMF1-ultra.	43
4.14. Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember for a) MET-direkte - MET-standard, b) MET-mellom - MET-standard og c) VMF1-MET-mellom.	44
4.15. Tidsserieanalyse for øst, nord og høyde for Stavanger for de ulike metodene i perioden november-desember 2015. I a) GPT2, b) VMF1, c) MET-direkte og d) VMF1-ULTRA.	47
4.16. Tidsserieanalyse for Trysil, for a) GPT2, b) VMF1, c) VMF1-GPT2 og d) VMF1-MET direkte.	48
4.17. Tidsserieanalyse for høyden til Tromsø for perioden november-desember 2015 for a) GPT2, b) VMF1, c) VMF1-GPT2 og d) VMF1-MET-direkte. . . .	49
4.18. Tidsserieanalyse for høyden til oljeplattformen Valhall for perioden november-desember 2015 for a) GPT2, b) VMF1, c) VMF1 - GPT2 og d) VMF1 - MET-direkte.	50
4.19. Tidsserieanalyse for høyden til Lofthus for perioden november-desember 2015 for a) GPT2, b) VMF1 og c) VMF1 - GPT2 og d) VMF1-MET-direkte.	51
4.20. Tidsserieanalyse for differansen mellom høyden til VMF1 og MET-mellom. a) Stavanger, b) Trysil, c) Tromsø og d) Valhall.	52
4.21. RMS-verdien til høydeposisjonen for differansen mellom VMF1 med atmosfærisk loading korreksjon og VMF1	55
4.22. Plottene viser korrelasjon mellom høyden fra tidsserieanalysen mot atmosfærisk loading for a) Trysil, b) Tromsø, c) Stavanger og d) Valhall. Trysil har høyest korrelasjon og Valhall har lavest.	56
4.23. Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember med korreksjon for atmosfærisk loading for a) VMF1 - MET-direkte og b) VMF1-GPT2, c) VMF1-VMF1-ultra og d) MET-direkte-GPT2.	57
4.24. Plottene viser tidsserieanalysen for VMF1 korrigeret for atmosfærisk loading.	58
4.25. Plottene viser forskjellen i tidsseries lineære RMS for høyden for perioden november-desember med korreksjon for atmosfærisk loading for a) MET-direkte - MET-standard, b) MET-mellom - MET-standard og c) VMF1-MET-mellom.	59
5.1. Plottet viser stasjoner som er godkjent til bruk for MET-direkte i data-assimileringen etter a) 1.november, b) 12. november, c) 25.november og d) 9.desember	62

5.2. Plottet viser stasjoner som er godkjent til bruk for GPT2 i data-assimileringen etter a) 1. november, b) 12. november, c) 25. november d) 9. desember	63
5.3. Plottet viser stasjoner som er godkjent til bruk for VMF1 i data-assimileringen etter a) 1. november b) 12. november, c) 25. november d) 9. desember.	64
5.4. Antall godkjente stasjoner i data-assimileringa for VMF1-ultra for a) 1. november og b) 12. november.	64
5.5. Plottene viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Stavanger i perioden 1. november-10. desember og fg-dep, er first-guess-minus-observasjoner	66
5.6. Plottene viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Trysil i perioden 1. november-10. desember og fg-dep, er first-guess- minus-observasjoner.	67
5.7. Plottene viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Tromsø i perioden 1. november-10. desember og fg-dep, er first-guess- minus-observasjoner.	68
5.8. Plottene viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for 'Valhall i perioden 1. november-10. desember og fg-dep, er first-guess- minus-observasjoner.	69
5.9. Plottene viser tidsserien for hvor godt VarBC korrigerer GPS-dataene for Lofthus i perioden 1. november-10. desember og fg-dep, er first-guess- minus-observasjoner.	70
5.10. Plottet øverst viser tidsserien for hvor godt VarBC korrigerer GPS-dataene metoden VMF1-ultra for perioden 1. november-12. november og fg-dep, er first-guess- minus-observasjoner. a) Viser Stavanger, b) Trysil og c) Tromsø.	71

Tabeller

2.1.	Tabellen viser koeffisientene k_1 , k_2 og k_3 som er løst empirisk. Her har man regnet k_1 basert på en CO ₂ -konsentrasjon på 300 ppm [11].	6
2.2.	Tabellen viser koeffisientene c_0 , c_{10} , c_{11} som brukes for å beregne den hydrostatisk c -koeffisienten til VMF1 for nordlig og sørlig halvkule [11].	10
2.3.	Tabellen viser koeffisientene c_1 , c_2 , c_3 som brukes for å beregne metningstrykket til vanndamp, SVP.	11
3.1.	Tabellen viser en oversikt over metodene som er brukt og hva som er forskjellen mellom dem.	19
3.2.	Tabellen viser hvilke modeller som er brukt i GIPSY-analysen.	20
4.1.	Tabellen viser et utvalg stasjoner som er sett nærmere på i oppgaven og ved hvilken ortometrisk høyde, lengde- og breddegrad de befinner seg. .	27
A.1.	Tabellen viser en oversikt over høydeforskjellen for de ulike stasjonene. Her er met-høyden trukket fra den korrekte høyden for stasjonen.	89

Bibliografi

- [1] A. Niell, "Global mapping functions for the atmosphere delay at radio wavelengths," *Journal of Geophysical Research: Solid Earth*, vol. 101, no. B2, pp. 3227–3246, 1996.
- [2] J. Böhm and H. Schuh, *Vienna mapping functions*. na, 2003.
- [3] J. Böhm, A. Niell, P. Tregoning, and H. Schuh, "Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data," *Geophysical Research Letters*, vol. 33, no. 7, 2006.
- [4] K. Lagler, M. Schindelegger, J. Böhm, H. Krásná, and T. Nilsson, "GPT2: Empirical slant delay model for radio space geodetic techniques," *Geophysical research letters*, vol. 40, no. 6, pp. 1069–1073, 2013.
- [5] B. W. Parkinson, *Progress in Astronautics and Aeronautics: Global Positioning System: Theory and Applications*, vol. 2. Aiaa, 1996.
- [6] G. Seeber, *Satellite geodesy: foundations, methods, and applications*. 2003.
- [7] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS—global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.
- [8] M. Troller, *GPS based determination of the integrated and spatially distributed water vapor in the troposphere*, vol. 67. 2004.
- [9] T. Herring, "Modeling atmospheric delays in the analysis of space geodetic data," *Proceedings of Refraction of Transatmospheric signals in Geodesy*, eds. JC De Munck and TA Spoelstra, Netherlands Geodetic Commission Publications on Geodesy, vol. 36, 1992.
- [10] E. K. Smith Jr and S. Weintraub, "The constants in the equation for atmospheric refractive index at radio frequencies," *Proceedings of the IRE*, vol. 41, no. 8, pp. 1035–1037, 1953.
- [11] E. J. Orliac, *Development of azimuth dependent tropospheric mapping functions, based on a high resolution mesoscale numerical weather model, for GNSS data processing*. PhD thesis, University of Nottingham, 2009.
- [12] O. Kristiansen, "Posisjonering med GPS- teori og anvendelser," 2015.
- [13] G. Petit and B. Luzum, "IERS conventions (2010)," tech. rep., DTIC Document, 2010.

- [14] P. Tregoning and T. Herring, "Impact of a priori zenith hydrostatic delay errors on GPS estimates of station heights and zenith total delays," *Geophysical Research Letters*, vol. 33, no. 23, 2006.
- [15] I. Martin, "Gnss precise point positioning: the enhancement with glonass," 2013.
- [16] J. Boehm, B. Werl, and H. Schuh, "Troposphere mapping functions for GPS and very long baseline interferometry from European Centre for Medium-Range Weather Forecasts operational analysis data," *Journal of Geophysical Research: Solid Earth (1978–2012)*, vol. 111, no. B2, 2006.
- [17] M. Bevis, S. Businger, S. Chiswell, T. A. Herring, R. A. Anthes, C. Rocken, and R. H. Ware, "GPS meteorology: Mapping zenith wet delays onto precipitable water," *Journal of applied meteorology*, vol. 33, no. 3, pp. 379–386, 1994.
- [18] T. R. Emardson and H. J. Derks, "On the relation between the wet delay and the integrated precipitable water vapour in the European atmosphere," *Meteorological Applications*, vol. 7, no. 1, pp. 61–68, 2000.
- [19] J. Haase, M. Ge, H. Vedel, and E. Calais, "Accuracy and variability of GPS tropospheric delay measurements of water vapor in the western Mediterranean," *Journal of Applied Meteorology*, vol. 42, no. 11, pp. 1547–1568, 2003.
- [20] P. Steigenberger, J. Boehm, and V. Tesmer, "Comparison of GMF/GPT with VMF1/ECMWF and implications for atmospheric loading," *Journal of Geodesy*, vol. 83, no. 10, pp. 943–951, 2009.
- [21] Y. Seity, P. Brousseau, S. Malardel, G. Hello, P. Bénard, F. Bouttier, C. Lac, and V. Masson, "The AROME-France convective-scale operational model," *Monthly Weather Review*, vol. 139, no. 3, pp. 976–991, 2011.
- [22] S. S. Rønning, "Vapor Pressure og geopotential hight." personal communication, Februar 2016.
- [23] M. A. Køltzow, K.-I. Ivarsson, D. Bjørge, S. Agersten, M. Lars, O. Vignes, B. Eriksen, P. Dahlgren, M. Ridal, and R. Rudsar, "Verification study - HARMONIE AROME compared with HIRLAM, UM and ECMWF," 2012.
- [24] T. M. van Dam, G. Blewitt, and M. B. Heflin, "Atmospheric pressure loading effects on Global Positioning System coordinate determinations," *Journal of Geophysical Research: Solid Earth*, vol. 99, no. B12, pp. 23939–23950, 1994.
- [25] L. Petrov, "International Mass Loading Service: Atmospheric Pressure Loading," 2015.
- [26] O. Kristiansen, "Introduksjon til minste kvadraters metode og kalmanfiltering," 2015.
- [27] NASA-Jet Propulsion Laboratory, "ORBIT DATA AND CLOCK PRODUCTS," 2016.

-
- [28] W. Bertiger, S. D. Desai, B. Haines, N. Harvey, A. W. Moore, S. Owen, and J. P. Weiss, "Single receiver phase ambiguity resolution with gps data," *Journal of Geodesy*, vol. 84, no. 5, pp. 327–337, 2010.
- [29] Met office, "EIG EUMETNET GNSS Water Vapour Programme(E-GVAP-II) - 'COST-Format' File Specification for Ground-based GNSS Delay and Water vapour Data," 2012.
- [30] A. Storto and R. Randriamampianina, "A new bias correction scheme for assimilating GPS zenith tropospheric delay estimates," *Idojárás*, vol. 114, no. 4, pp. 237–250, 2010.
- [31] J. S. Arriola, M. Lindskog, S. Thorsteinsson, and J. Bojarova, "Variational bias correction of GNSS ZTD in the HARMONIE modeling system," *Journal of Applied Meteorology and Climatology*, no. 2016, 2016.
- [32] R. Randriamampianina, "Data-assimilation MET." personlig kommunikasjon, Mai 2016.
- [33] Meteorologisk institutt, "Høytrykk over skandinavia," 2015.
- [34] C. Selle and S. Desai, "Optimalization of tropospheric delay estimation parameters by comparison of GPS-based precipitable water vapor estimated with microwave radiometer measurements," Februar 2016.
- [35] J. Kouba, "Testing of global pressure/temperature (GPT) model and global mapping function (GMF) in GPS analyses," *Journal of Geodesy*, vol. 83, no. 3-4, pp. 199–208, 2009.

A. Høydeforskjell

Tabell A.1.: Tabellen viser en oversikt over høydeforskjellen for de ulike stasjonene. Her er met-høyden trukket fra den korrekte høyden for stasjonen.

STA	Høydediff	STA	Høydediff	STA	Høydediff	STA	Høydediff
aasc	-7.30	floc	3.65	lalc	-9.23	renc	-15.13
adac	-106.33	folc	87.79	lauc	-24.00	rind	-21.18
akrc	6.48	fold	-31.31	leic	-326.32	roac	25.55
ales	144.76	forc	-193.83	leik	-99.71	rorc	-41.42
altc	40.08	frec	14.18	lekc	1.53	rosc	-184.30
alvc	-53.17	froc	47.90	leks	-74.66	royc	154.57
ande	7.99	fyha	3.83	liec	152.35	selc	-138.59
ando	272.25	gesr	2.76	lilc	-24.63	sfja	60.81
arda	-428.16	gjec	-303.43	linc	34.30	sfjb	68.66
arec	17.71	gjes	38.86	lisc	0.35	sfjc	86.57
arnc	5.38	gjoc	-397.24	lodc	-23.91	sirc	-2.67
asac	-50.90	gloc	-38.06	lofs	32.52	skac	-103.03
askv	-118.84	golec	-199.29	loft	-653.66	skan	-20.71
aust	18.07	gran	-38.80	lomc	-243.48	skja	-140.11
balc	-17.02	groc	-83.71	lonc	-62.07	skjc	-17.64
barc	-52.61	gudr	87.35	lopc	-88.31	skoc	1.62
batc	-70.15	haby	18.84	lotc	16.69	skrc	-23.44
berc	-43.82	hama	20.72	lstc	1.15	slpn	87.34
birk	-38.50	hamc	-34.78	lund	-134.56	smoc	14.57
bjac	-10.78	hanc	-14.90	lurc	-109.18	smol	12.07
vale	67.77	harc	9.23	lyng	-66.23	smrc	-51.75
bjoc	-131.23	hauc	-80.66	lysc	-55.33	sort	-8.16
vegc	-120.04	havc	-55.58	malo	-70.30	stac	-120.70
bjuc	69.48	hedc	-64.84	mare	3.27	stas	40.79
blec	-78.59	hegc	-66.27	mauc	-182.94	stec	12.28
bmlc	6.36	helc	179.69	mazc	-75.79	stgc	12.28
bod2	1.94	hemc	-92.05	mebc	-39.24	stjc	-6.14
bod3	1.94	hfs2	18.80	mehc	-66.56	stoe	-159.33
bods	17.10	hfs3	18.80	mfol	87.59	stra	113.05
boec	-57.30	hfs4	18.70	mldc	-65.86	sulc	-258.49
bomc	37.50	hfss	18.70	mnak	28.23	svec	-112.14
brec	-112.91	hita	6.36	moec	-39.22	svic	-8.50

A. Høydeforskjell

brgs	15.52	hodd	33.44	moic	-166.88	svoc	-23.21
broc	15.53	holc	84.85	molc	-233.45	tanc	-53.84
budp	30.88	hon2	29.86	mosj	-130.05	terc	-41.41
bygc	-164.80	hons	29.86	mran	46.39	tgde	-30.14
dags	-96.82	horc	49.69	msim	-121.56	tinc	-187.80
dglc	-95.82	huld	40.77	msta	3.41	tjmc	51.91
doms	-104.90	husc	2.06	mste	-61.31	tjoc	10.70
donc	-69.90	ibec	19.04	mtrm	75.54	tnsc	-189.95
drec	-3.05	innc	-271.77	myrc	-5.33	tonc	-347.90
eids	-41.74	jan2	5.14	musc	45.94	tore	53.90
ekoa	45.71	jans	5.14	myse	-24.88	trds	116.05
ekob	36.08	josc	-304.15	nams	-42.48	trec	-128.00
ekoh	61.37	karc	-65.10	narc	-82.42	trms	75.44
ekoj	70.48	kau2	23.27	nask	-76.62	tro1	75.44
ekol	71.33	kaus	23.27	norh	-66.34	tro2	69.74
elda	48.28	kirc	1.53	noto	-110.98	troc	-98.10
eldb	46.55	kjoc	-50.58	oldc	-132.36	trol	90.19
embl	39.09	kobc	-209.49	opcc	6.18	trom	69.74
esbc	3.16	konc	-10.55	ornc	-117	trys	26.82
etne	-136.98	kopc	-22.70	orsk	-84.52	tved	-44.99
faus	17.27	krod	-74.13	osls	21.14	tyic	145.21
fava	-184.73	krss	97.95	ostc	-390.32	tysv	22.55
ferr	21.13	krsu	27.59	ovec	-47.46	ulec	17.71
fetc	-0.85	kvac	-295.93	pasc	-5.34	ulsc	-110.52
finc	6.85	kvbj	53.35	porc	8.88	vadc	39.88
fioy	-10.21	kvic	-131.73	prec	-75.17	val1	79.38
flic	9.73	kyrc	-207.21	prgc	-34.50	val2	79.18
vegs	1.85	vold	-151.45	rauc	60.23	val3	87.18
vikc	-14.06	vosc	-163.74	vard	9.91	val4	101.68
vike	-179.97	vssc	-163.64	vars	141.02	valc	-97.48
vinc	-164.34	valh	51.08				

B. Ionosfærefri lineærkombinasjon

Utleiding av ionosfærefri lineærkombinasjon basert på [26]

$$L_3 = \alpha L_1 + \beta L_2 = \frac{f_1^2}{f_1^2 - f_2^2} L_1 - \frac{f_2^2}{f_1^2 - f_2^2} L_2 = 2.546 L_1 - 1.546 L_2 \quad (\text{B.1})$$

$$P_3 = \alpha P_1 + \beta P_2 = \frac{f_1^2}{f_1^2 - f_2^2} P_1 - \frac{f_2^2}{f_1^2 - f_2^2} P_2 = 2.546 P_1 - 1.546 P_2 \quad (\text{B.2})$$

der $f_1 = 154$ og $f_2 = 120$.

Ligning 2.4 settes så inn i B.2 og 2.5 i B.1.

Den ionosfærefrie-biasen for p-koden blir da

$$b_{p_3}^r = \frac{f_1^2}{f_1^2 - f_2^2} b_{p_1}^r - \frac{f_2^2}{f_1^2 - f_2^2} b_{p_2}^r = \alpha_3 b_{p_1}^r + \beta b_{p_2}^r \quad (\text{B.3})$$

$$b_{p_3}^s = \frac{f_1^2}{f_1^2 - f_2^2} b_{p_1}^s - \frac{f_2^2}{f_1^2 - f_2^2} b_{p_2}^s = \alpha_3 b_{p_1}^s + \beta b_{p_2}^s \quad (\text{B.4})$$

total gir det

$$P_3 = \rho + \delta_{orb} + (cdt_r + b_{p_3}^r) - (cdt^s + b_{p_3}^s) + Z + \epsilon_{P_3} \quad (\text{B.5})$$

som er det samme som 2.6.

For fasen blir ting litt annerledes fordi et tvetydighetsledd er inkludert,

$$\lambda_3 N_3 = \alpha_3 \lambda_1 N_1 - \beta_3 \lambda_2 N_2 \quad (\text{B.6})$$

α og β er her de samme som i B.1. λ er bølgelengden og N er en heltallstvetydighet for L. Fordi $\lambda = \frac{c}{f}$ blir

$$\lambda_3 N_3 = \frac{c}{f_1^2 - f_2^2} (f_1 N_1 - f_2 N_2) = \frac{2cf_0}{f_1^2 - f_2^2} (77N_1 - 60N_2) \quad (\text{B.7})$$

Bias-leddene for fasen finner man på lignende måte som for koden:

$$b_{L_3}^r = \frac{f_1^2}{f_1^2 - f_2^2} b_{L_1}^r - \frac{f_2^2}{f_1^2 - f_2^2} b_{L_2}^r = \alpha_3 b_{L_1}^r + \beta b_{L_2}^r \quad (\text{B.8})$$

$$b_{L_3}^s = \frac{f_1^2}{f_1^2 - f_2^2} b_{L_1}^s - \frac{f_2^2}{f_1^2 - f_2^2} b_{L_2}^s = \alpha_3 b_{L_1}^s + \beta b_{L_2}^s \quad (\text{B.9})$$

Fase-ligningen blir dermed

$$L_3 = \rho + \delta_{orb} + (cdt_r + b_{L_3}^r) - (cdt^s + b_{L_3}^s) + Z + B_3 + \epsilon_{L_3} \quad (\text{B.10})$$

der B_3 er

$$B_3 = \lambda_3 N_3 = \frac{2cf_0}{f_1^2 - f_2^2} (77N_1 - 60N_2) \quad (\text{B.11})$$

C. Fra geopotensial til geomterisk høyde

For å finne høyden til en stasjon som blir brukt i modellen AROME-MetCoOp brukes geopotensial verdi for en gitt lengde- og breddegrad som så regnes om til geometrisk høyde ved hjelp av formlene under. Fremgangsmåte er fått fra Snorre Stavik Rønning [22], som er ansatt på MET.

$$Z = \frac{R \times \Phi}{g_\phi \times R - \Phi} \quad (\text{C.1})$$

der Z er geometrisk høyde over ellipsoiden(mean sea level), R er riktig verdi for jordas radius ved en gitt breddegrad i m, Φ er geopotensial og g_ϕ faktisk gravitasjon ved geoiden ved en gitt breddegrad. g_ϕ er gjort ved å bruke WGS 1984 Ellipsoidal Gravity formula.

$$R = \frac{2 \times g_\phi}{-(\frac{\delta g}{\delta Z})_{Z=0}}. \quad (\text{C.2})$$

$$-(\frac{\delta g}{\delta Z})_{Z=0} = 3,085462 \times 10^{-6} + 2,27 \times 10^{-9} \cos(2\phi) - 2 \times 10^{-12} \cos(4\phi). \quad (\text{C.3})$$

$$g_\phi = 9,7803267714 \times \frac{1 + 0,0019318513138639 \times \sin^2(\phi) \text{ m}}{\sqrt{1 - 0,00669437999013 \times \sin^2(\phi)} \text{ s}^2} \quad (\text{C.4})$$

der ϕ er breddegraden gitt i radianer.

D. Nedbør 2015

D.0.1. November

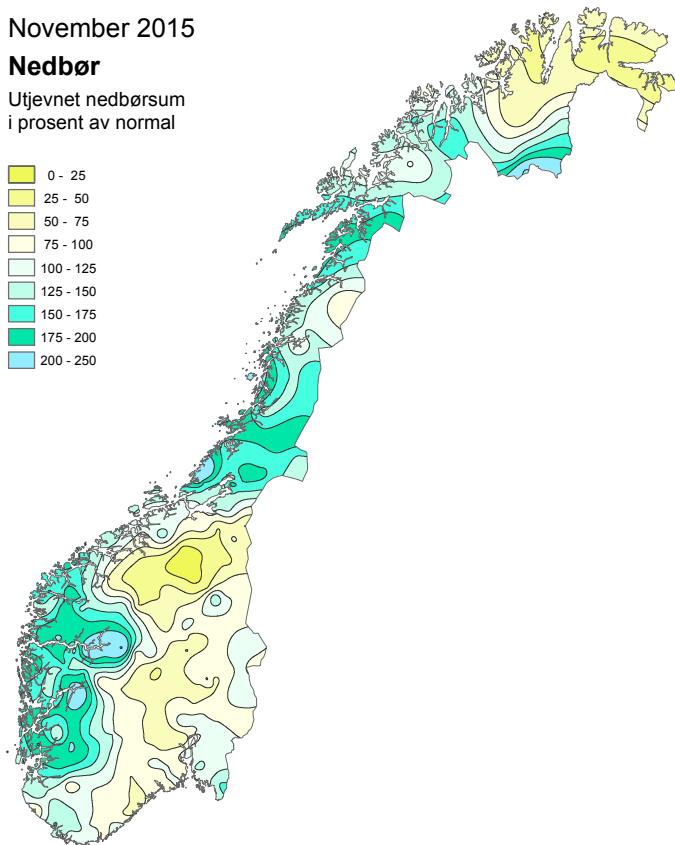
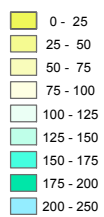


Klimatologisk månedsoversikt

November 2015

Nedbør

Utjevnet nedbørsum
i prosent av normal



Normalperioden er 1961 - 1990

Utgitt: 01.12.2015

Ved bruk skal Meteorologisk institutt oppgis som kilde.
http://met.no/Klima/Klimastatistikk/Varet_L_Norge/

D.0.2. Desember

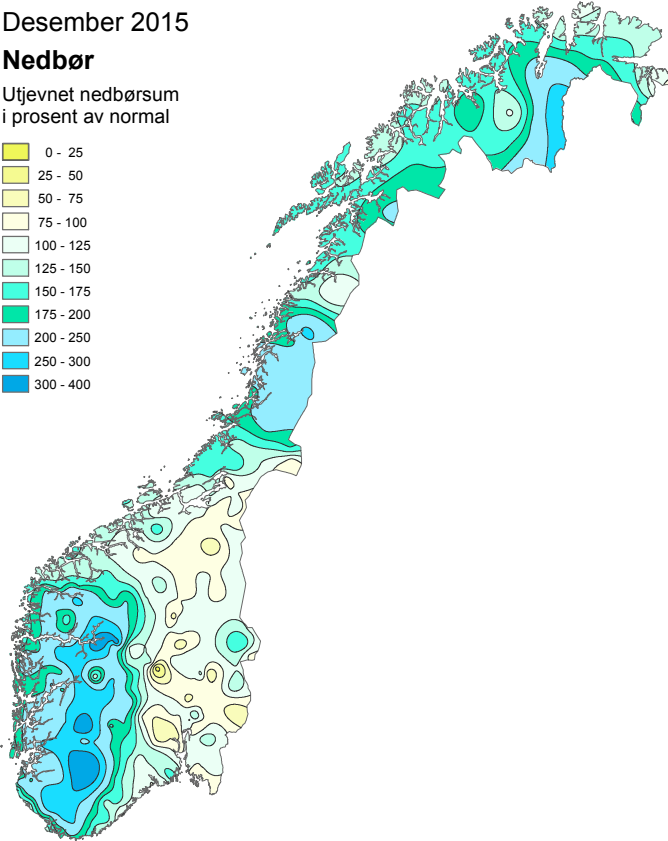
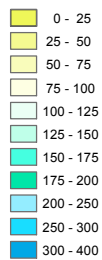


Klimatologisk månedsoversikt

Desember 2015

Nedbør

Utjevnet nedbørsum
i prosent av normal



Normalperioden er 1961 - 1990

Utgitt: 04.01.2015
Ved bruk skal Meteorologisk institutt oppgis som kilde.
http://met.no/Klima/Klimastatistikk/Varet_i_Norge/

E. Kildekode

MET2TDP: Programmet henter AROME-MetCoOP dataene trykk, temperatur og relativ fuktighet og bruker dette til å beregne ZHD og ZWD, som forklart i avsnitt 2.2. Det lages så en tdp(time-dependent-parameter)-fil slik at GIPSY PPP kan kjøre VMF1 med a priori informasjon om ZHD og ZWD fra AROME-MetCoOp.

COST-format: Programmet henter ZTD, ZWD og gradientene med tilhørende error fra GIPSY PPP løsningen, samt temperatur, trykk og relativ fuktighet fra AROME-dataene. IWV er også regnet ut her ved hjelp av formel (2.16). Dette settes opp slik at det kan kjøres i METs data-assimilering.

E.1. MET2TDP

```
#!/usr/bin/env python3

import subprocess
import os.path
import netCDF4 as nc
from datetime import datetime
import sys
import numpy as np
import math
from scipy import interpolate

BASE_DIR = os.path.join('/home', 'hanne', 'Master')
INPUT_FILE = os.path.join(BASE_DIR, 'input', '{yyyy}', '{mm}', 'AROME_MetCoOp_00_fp.nc_{yyyy}
    {mm}{dd}')
OUTPUT_FILE = os.path.join(BASE_DIR, 'output_direkte', '{STATION}', '{station}{doy}{yy}
    _met_drywet.tdp')

DATE_VARS = dict()

def main():
    create_date_vars(int(sys.argv[1]), int(sys.argv[2]), int(sys.argv[3]))
    print (INPUT_FILE.format(**DATE_VARS))
    lat_long, name, lat, height = read_station_positions()
    extract_data(lat_long)
    data, time = read_MET_data(lat,height)
    write_tdp(name, time, data, ['DRYZTROP', 'WETZTROP'])

def create_date_vars(year, month, day):
    date = datetime(year, month, day)
    DATE_VARS.update({
        'yyyy': date.strftime('%Y'),
        'ce': date.strftime('%Y')[:2],
```

E. Kildekode

```
'yy':      date.strftime('%y'),
'mm':      date.strftime('%m'),
'mmm':     date.strftime('%b').lower(),
'MMM':     date.strftime('%B').upper(),
'dd':      date.strftime('%d'),
'doy':     date.strftime('%j'),
})

def read_station_positions():
    """Read latitude and longitude of all stations.

    Read names, latitudes and longitudes for the stations from the file "
    norwegian_station_coordinates.txt".

    Returns:
        Two lists, one with latitude-longitude-tuples, one with station codes.
    """
    lat_long = list()
    name = list()
    lat = list()
    height = list()
    with open(os.path.join(BASE_DIR, 'norwegian_station_coordinates.txt')) as fid:
        for line in fid:
            if line.startswith('#') or not line.strip():
                continue
            fields = line.split()
            lat_long.append((fields[6], fields[7]))
            lat.append(float(fields[4]))
            name.append(fields[0])
            height.append(float(fields[10]) - float(fields[11]))

    return lat_long, name, np.array(lat), np.array(height)

def extract_data(lat_long):
    """ Extracts temprature, pressure and relative humidity from METdata and interpolate
    to the coordinates for the stations.

    Use fimex to extract temprature, pressure and relative humidity for the
    AROME_MetCoOp_00_fp* data.
    Fimex also interpolates the parameters to the longitude and latitude for the different
    GPS stations.

    Returns: MET2TDP.nc
    """
    fimex_cmd = [
        'fimex',
        INPUT_FILE.format(**DATE_VARS),
        '--input.type=nc',
        '--extract.reduceDimension.name=time',
        '--extract.reduceDimension.start=0',
        '--extract.reduceDimension.end=24',
        '--extract.selectVariables=x',
        '--extract.selectVariables=y',
        '--extract.selectVariables=air_temperature_2m',
        '--extract.selectVariables=relative_humidity_2m',
        '--extract.selectVariables=surface_air_pressure',
        '--interpolate.latitudeValues=' + ','.join([ll[0] for ll in lat_long]),
        '--interpolate.longitudeValues=' + ','.join([ll[1] for ll in lat_long]),
        '--output.file', 'MET2TDP.nc',
    ]
    subprocess.call(fimex_cmd)
```

```

def read_MET_data(lat, height):
    """Read the data from MET2TDP.nc

    Extract the different coloums from the dataset; time,air_pressure_at_sea_level
    relative_humidity_2m and air_temperature_pl. Converts temprature from Kelvin to
    celsius
    and the time from seconds after 1970 to J200. Interpolates to 5 miunutes(300 seconds)
    data for 24 hours.

    Calculates the saturated water pressure with different values of the coeffisients
    depending on the temprature, and from that calculates the
    partial water vapor pressure, e.

    Based on that calculates the zenit hydrostatic delay, ZHD, and zenit wet delay, ZWD.

    Returns output file with ZHD and ZWD.

    """
    data = nc.Dataset('MET2TDP.nc')
    delta=(datetime(2000, 1, 1, 12, 0) - datetime(1970, 1, 1)).total_seconds()
    time_indata = data['time'][:] - delta
    time = np.arange(time_indata[0], time_indata[-1], 300)
    pressure = interpolate_data(data, 'surface_air_pressure', time_indata, time)[: , 0, :]/
        100
    tempk = interpolate_data(data, 'air_temperature_2m', time_indata, time)[: , 0, :]
    tempc = tempk - 273.15
    r_humidity = interpolate_data(data, 'relative_humidity_2m', time_indata, time)[: , 0,
        : ]
    coeffs = np.zeros(tempk.shape + (3,))
    coeffs[tempc<0, :] = np.array((6.10718, 17.84362, 245.425))
    coeffs[0<=tempc, :] = np.array((6.10718, 17.08085, 234.175))
    sat_water = coeffs[:, :, 0] * (np.exp((coeffs[:, :, 1]*tempc)/(coeffs[:, :, 2] + tempc
        )))
    water_vapor = r_humidity * sat_water
    f = 1 - 0.00266 * np.cos(2 * lat) - 0.00000028 * height
    ZHD = (0.0022768) * pressure / f
    ZWD = ((1255 / tempk) + 0.05) * (0.0022768 * water_vapor) / f
    output = np.zeros(2,) + tempk.shape
    output[0] = ZHD
    output[1] = ZWD
    return output, time

def interpolate_data(data, fieldname, time_from, time_to):
    field_data_from = data[fieldname][:,:,0,:]
    return interpolate.interp1d(time_from, field_data_from, axis=0)(time_to)

def write_tdp(station_names, times, data, data_names):
    """ Writes the tdp-file in the correct format, so it can run in GIPSY.

    """
    for col, station in enumerate(station_names):
        if np.max(data[:, :, col]) > 10:
            print('No availibale data, station: ' + station)
            continue
        output_file = OUTPUT_FILE.format(station=station, STATION=station.upper(), **
            DATE_VARS)
        print('Writing to ' + output_file)
        os.makedirs(os.path.dirname(output_file), exist_ok=True)
        with open(output_file, mode='w') as fid:
            for idx, data_name in enumerate(data_names):
                for time, datum in zip(times, data[idx, :, col]):
                    fid.write(' {:14.4f} {:22.15e} {:22.15e} {:10.3e}  {:<8s}{:<4s}\n'.format(time

```

```
, 0, datum, 1, data_name, station.upper())

if __name__ == '__main__':
    main()
```

E.2. COST-format

```
#!/usr/bin/env python3

import subprocess
import glob
import gzip
import os.path
import netCDF4 as nc
from datetime import datetime, timedelta
import sys
import numpy as np
from scipy import interpolate
import math

BASE_DIR = os.path.join('/home', 'kihhan', 'OGIPSY', 'RES')
INPUT_FILES = {'TDP': os.path.join(BASE_DIR, '{METHOD}', 'PPP', '{STATION}', '{yyyy}-{mm}-{dd}{STATION}.tdp'),
               'TDPgzip': os.path.join(BASE_DIR, '{METHOD}', 'PPP', '{STATION}', '{yyyy}-{mm}-{dd}{STATION}.tdp.gz'),
               'ZHD': os.path.join(BASE_DIR, '{METHOD_VMF1}', 'TROPNOMINAL', '{STATION}', '{station}{doy}{yy}_{method_vmfl}_drywet.tdp'),
               'site': os.path.join('/home', 'kihhan', 'OGIPSY', 'MOD', 'sesite.snx'),
               'height': os.path.join('/home', 'kihhan', 'Master', 'norwegian_station_coordinates.txt'),
               'vmf_tmean': os.path.join('/home', 'kihhan', 'VMF_TMEAN', 't{yy}{doy}.h{HH:02d}'),
               'met_temp': os.path.join('/home', 'kihhan', 'Master', 'met_temp_riktig', '{STATION}', '{station}{doy}{yy}_met_temp.tdp')}
OUTPUT_FILE = os.path.join(BASE_DIR, '{METHOD}', 'COST', 'cost_{first:%Y%m%d%H%M}_mult_{processingcenter}.dat')
DATE_VARS = dict()

def create_date_vars(year, month, day):
    date = datetime(year, month, day)
    DATE_VARS.update({
        'yyyy': date.strftime('%Y'),
        'ce': date.strftime('%Y')[2],
        'yy': date.strftime('%y'),
        'mm': date.strftime('%m'),
        'mmm': date.strftime('%b').lower(),
        'MMM': date.strftime('%B').upper(),
        'dd': date.strftime('%d'),
        'doy': date.strftime('%j'),
        'HH': date.strftime('%H'),
        'MM': date.strftime('%M'),
        'SS': date.strftime('%S'),
    })

def read_station_heights():
    """Read heights of all stations.

    Reads norwegian_station_coordinates.txt.
```



```

Returns:
    Dict with heights and coordinates
"""
data = dict()
with open (INPUT_FILES['height']) as fid:
    for line in fid:
        if line.startswith('#') or not line.strip():
            continue
        fields = line.split()
        station = fields[0]
        data[station] = dict (lat=float (fields[6]),
                             lon=float (fields[7]),
                             ell_ht=float (fields[10]),
                             geo_ht=float (fields[11]))

return data

def read_station_info():
    data = dict()
    with open (INPUT_FILES['site']) as fid:
        for line in fid:
            if line.startswith('+'):
                parse_func = SNX_FUNCS[line.strip()[1:]]
            elif line.startswith(' '):
                parse_func(data, line)

    return data

def parse_snx_ignore(data, line):
    pass

def parse_snx_site(data, line):
    station = line[1:5].lower()
    domes = line[9:18]
    domes = domes.replace('-----', 'XXXXXXXXXX')
    domes = domes.replace('_____', 'XXXXXXXXXX')
    station_name = line[21:45].title()

    data.setdefault(station, dict()).update(dict (domes=domes, station_name=station_name)
)

def parse_snx_receiver(data, line):
    station = line[1:5].lower()
    receiver = line[42:62]

    data.setdefault(station, dict()).update(dict (receiver=receiver))

def parse_snx_antenna(data, line):
    station = line[1:5].lower()
    antenna = line[42:62]

    data.setdefault(station, dict()).update(dict (antenna=antenna))

def read_observation_data(station):
    """Reads observation data from TDP-files in TROPNOMINAL files for ZHD and PPP-solutions
    for the ZWD """
    vars = dict (station=station, STATION=station.upper(), **DATE_VARS)
    obs_dict = dict()

```

```

if os.path.exists(INPUT_FILES['TDPgzip'].format(**vars)):
    tdp_open = gzip.open
    tdp_file = INPUT_FILES['TDPgzip'].format(**vars)
elif os.path.exists(INPUT_FILES['TDP'].format(**vars)):
    tdp_open = open
    tdp_file = INPUT_FILES['TDP'].format(**vars)
else:
    return list()

with tdp_open(tdp_file, mode='rt') as fid:
    for line in fid:
        parse_tdp_line(obs_dict, line, ('WETZTROP', 'TRPAZCOS', 'TRPAZSIN'))

with open(INPUT_FILES['ZHD'].format(**vars), mode='rt') as fid:
    for line in fid:
        parse_tdp_line(obs_dict, line, ('DRYZTROP'))

with open(INPUT_FILES['met_temp'].format(**vars), mode='rt') as fid:
    for line in fid:
        parse_tdp_line(obs_dict, line, ('TEMP', 'PRESSURE', 'RH'))

for variable in ('WETZTROP', 'TRPAZCOS', 'TRPAZSIN', 'DRYZTROP'):
    for date_dict in obs_dict.values():
        if variable in date_dict:
            date_dict[variable] *= 1000
            date_dict[variable + '_err'] *= 1000

obs_list = list()
for time in sorted(obs_dict.keys()):
    obs = obs_dict[time]
    if 'WETZTROP' not in obs:
        continue

    obs['pcd'] = 2**32 - 1
    obs['number_of_slants'] = 0
    obs['ztd'] = obs['DRYZTROP'] + obs['WETZTROP']
    obs['ztd_err'] = obs['WETZTROP_err']

    obs['iwv'] = (obs['WETZTROP'] / (6.458 + ((-1.78e-2)*(obs['TEMP'] - 283.49)) +
        ((-2.2e-5)*(obs['TEMP'] - 283.49)**2)))
    obs['rh'] = obs['RH'] * 100 #in %
    obs['tec'] = -99.999 #BEWARE
    obs_list.append(obs)
return obs_list

def parse_tdp_line(data, line, variables):
    timestamp = float(line[1:15])
    timestamp = timestamp - timestamp % 300 # BEWARE, is it ok to consider 17 seconds
        different as the same time?
    timestamp_dt = datetime(2000, 1, 1, 12, 0) + timedelta(seconds=timestamp)
    variable = line[74:82].strip()
    if variable in variables:
        data.setdefault(timestamp_dt, dict()).update({'HH': timestamp_dt.hour,
            'MM': timestamp_dt.minute,
            'SS': timestamp_dt.second,
            variable: float(line[39:61]),
            variable + '_err': float(line
                [62:72])})

```

```

def write_cost_file(data):
    """ Writes the cost-format file, with date, ZTD,ZTD-error, ZWD, IWV, pressure,
    temperature, relative humidity, and the gradients. """
    stations = sorted([os.path.basename(d).lower() for d in glob.glob(os.path.join(
        BASE_DIR, data['METHOD'],'PPP', '*'))])
    heights = read_station_heights()
    station_info = read_station_info()
    vmf_tmean = read_vmf_tmean()
    first = data.pop('date_first_sample')
    last = data.pop('date_last_sample')
    cost = [None] * 8
    with open(OUTPUT_FILE.format(first=first, last=last, **data), mode='w') as cost[0],
        \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=3), last=last +
            timedelta(hours=3), **data), mode='w') as cost[1], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=6), last=last +
            timedelta(hours=6), **data), mode='w') as cost[2], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=9), last=last +
            timedelta(hours=9), **data), mode='w') as cost[3], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=12), last=last +
            timedelta(hours=12), **data), mode='w') as cost[4], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=15), last=last +
            timedelta(hours=15), **data), mode='w') as cost[5], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=18), last=last +
            timedelta(hours=18), **data), mode='w') as cost[6], \
        open(OUTPUT_FILE.format(first=first + timedelta(hours=21), last=last +
            timedelta(hours=21), **data), mode='w') as cost[7]:
    for station in stations:
        print(station)
        obs_data = read_observation_data(station)
        filtered_obs = [list(), list(), list(), list(), list(), list(), list(), list
            ()] # Lag 8 tomme lister
        for obs in obs_data:
            if (obs['HH'] % 3) == 1 and obs['MM'] == 0: #BEWARE, is this OK?
                obs['MM'] = 59
                obs['HH'] = obs['HH'] - 1
            if (obs['MM'] not in [00, 15, 30, 45, 59]) or (obs['HH'] % 3 > 0):
                continue
            idx = int(obs['HH'] / 3)
            filtered_obs[idx].append(obs)
        data['station'] = station
        data['STATION'] = station.upper()
        data.update(heights.get(station, dict())) # BEWARE: wrong height if
            station missing
        data.update(station_info.get(station, dict())) # BEWARE: wrong info if
            station missing
        # Header
        for idx in range(8):
            data['num_obs'] = len(filtered_obs[idx])
            data['first_sample'] = '{:%d-%b-%Y %H:%M:%S}'.format(first + timedelta(
                hours=3*idx)).upper()
            cost[idx].write("""{format_name:20}      {project_name:20}      {
                file_status:20}
{STATION:4} {domes:9}          {station_name:60}
{receiver:20}      {antenna:20}
{lat:12.6f}{lon:12.6f}{ell_ht:12.3f}{geo_ht:12.3f}{benchmark_ht:12.3f}
{first_sample:20}          {processing_date:20}
{processingcenter:20}      {processing_method:20}      {orbit_type:20}      {source_metdata
:20}
{time_increment:5d}{update_interval:5d}{batch_length:5d}
{pcdh:08X}
{num_obs:4d}
""").format(**data)
    """ .format(**data)

```

```

# Observations
for idx in range(8):
    for obs in filtered_obs[idx]:
        cost[idx].write(""" {HH:02d} {MM:02d} {SS:02d} {pcd:08X}{ztd:7.1f}{
            ztd_err:7.1f}{WETZTROP:7.1f}{iwv:7.1f}{PRESSURE:7.1f}{TEMP:7.1f
            }{rh:7.1f}{TRPAZCOS:7.2f}{TRPAZSIN:7.2f}{TRPAZCOS_err:7.2f}{
            TRPAZSIN_err:7.2f}{tec:8.3f}

(number_of_slants:4d)
""".format(**obs))

# Footer
for idx in range(8):
    cost[idx].write('-' * 100 + '\n\n')

def main():
    method_map = dict(vmf1='vmf1grid', vmf1_ultra='vmf1grid')
    METHOD_map = dict(VMF1_ULTRA='VMF1')

    create_date_vars(int(sys.argv[2]), int(sys.argv[3]), int(sys.argv[4]))
    DATE_VARS['method'] = sys.argv[1].lower()
    DATE_VARS['METHOD'] = sys.argv[1].upper()
    DATE_VARS['method_vmf1'] = method_map.get(DATE_VARS['method'], DATE_VARS['method'])
    DATE_VARS['METHOD_VMF1'] = METHOD_map.get(DATE_VARS['METHOD'], DATE_VARS['METHOD'])
    data = dict(processingcenter='nma_',
                format_name='COST-716 V2.2',
                project_name='E-GVAP',
                file_status='OPER',
                processing_date=datetime.strftime(datetime.now(), '%d-%b-%Y %H:%M:%S').
                    upper(),
                first_sample='{dd}-{MMM}-{yyyy} {HH}:{MM}:{SS}'.format(**DATE_VARS),
                benchmark_ht=0,
                pcdh=2*32 - 1,
                date_first_sample=datetime(int(sys.argv[2]), int(sys.argv[3]), int(sys.
                    argv[4])),
                date_last_sample=datetime(int(sys.argv[2]), int(sys.argv[3]), int(sys.
                    argv[4]), 0, 59, 0),
                **DATE_VARS,
                processing_method='GIPSY 6.3',
                orbit_type='JPLPRE',
                source_metdata='OBS/INTERPOLATED',
                time_increment=15, #BEWARE
                update_interval=60, #BEWARE, don't know what this is
                batch_length=360, #BEWARE, don't know what this is
                )
    write_cost_file(data)

SNX_FUNCS = {'FILE/REFERENCE': parse_snx_ignore,
             'SITE/ID': parse_snx_site,
             'SITE/RECEIVER': parse_snx_receiver,
             'SITE/ANTENNA': parse_snx_antenna,
             'SITE/ECCENTRICITY': parse_snx_ignore}

if __name__ == '__main__':
    main()

```




Norges miljø- og biovitenskapelig universitet
Noregs miljø- og biovitenskapelige universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway