ANNEXE 2

Report on the work with Kalman filtering of DACFOS

J. Jacobsen

Rapport over arbejde med Kalmanfiltrering af DACFOS

Joachim Jacobsen
DMI d. 17. september 1997

Contents

1 Indledning

Som baggrund for nedenstående henvises til Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter" (GW & GB) , samt Christian Ødum Jensens note "Kalman filtrering af DACFOS modellen" (CJ).

I forbindelse med reference til forskellige filer i det nedenstaaende, vil for overskuelighedens skyld "$Kalman" stå for direktoriet /net/marvin/priv_3/marvin_disperse/DACFOS/kalman_filter/ . For at følge eksemplerne start da med kommandoen:

setenv Kalman /net/marvin/priv_3/marvin_disperse/DACFOS/kalman_filter

2 Status pr. 1/8/97:

2.1 Den specifikke implementering af Kalman filteret

Kalman filteret anvendes her til parameter bestemmelse - det vil sige, man antager at ozonkoncentrationen i Jægersborg kan skrives som en linear kombination,

$$C_{O3}(t) = \sum_i h_i(t) x_i, \tag{1}$$

af forskellige variable h_i, der skal kendes i en indlæringsperiode, og kunne forudsiges et stykke ud i fremtiden. Kalmanfilteret anvendes så til at bestemme de konstante koefficienter, x_i, i denne linear kombination. Kalmanfilteret sœger at finde disse koefficienter i en iterativ process i en indlæringsperiode hvor baade $C_{O3}(t)$ og alle h_i(t) er kendte. Efter indlæringsperioden har man så et bud paa hvad koefficienterne x_i er, og dette bruges til at forudsige $C_{O3}(t)$ udfra forudsigelserne af $h_i(t)$, som kommer andetsteds fra.
Ovenstående ligning (1) svarer til ligning (6) i CJ - $C_{O3}(t)$ er z_k, x_i er komponenterne i vektoren x_k; da z_k her er en vektor med kun et element, bliver matricen H_k en $1 \times N$ matrix med elementerne $h_i(t)$, hvor N er antallet af variable h_i, der inkluderes. Indekset k giver tiden t i det diskrete Kalman Filter, og ligning (6) i CJ skriver x_k, fordi koefficienterne x_i varierer i indlæringsperioden. Det er dog nyttigt at tænke på det på den måde, at Kalman filteret i indlæringsperioden forsøger at lære de bedste værdier for de konstante koefficienter. Således er i ligning (1) i CJ, matricen A enhedsmatricen, og B

nul – vores model siger $x_{k+1} = x_k$, mens indæringen af værdien af x_k sker i ligning (4) i CJ.
For et givet valg af hvilke variable h_i, der inkluderes i Kalman filteret, er 3 parametre (i filterets nuværende implementering), der kan justeres for at optimere kvaliteten af Kalman filterets prognose: Q, R og τ. Q og R er definerede i lignigerne (1.3) og (1.4) i GW & GB. Ledet Bu_k i ligning (1.1) er lempet ind i Q, som således angiver usikkerhed og fejl på antagelsen om at "den rigtige" værdi for koefficienterne x_i er konstant. R angiver den totale fejl til et givet tidspunkt af ligning (1), og indeholder saaledes både måleusikkerheden på $C_{O3}(t)$ samt fejlen i antagelsen om, at $C_{O3}(t)$ skulle være en linear kombination af de valgte variable h_i. τ er indført i ligningerne (3) og (5) i CJ, og er en tidskonstant for hvor hurtigt Kalman filteret glemmer den tidlige indlæringsperiode.

2.2 Valg af uafhængige variable h_i, og af parametre.

Kalman filteret kørte semi-operationelt i to versioner for brugeren disperse i direktorierne: $Kalman/test_bin og: $Kalman/bin. Hver af disse versioner kørte flere forskellige kalmanfiltre.
Som de uafhængige variable h_i anvendtes den "gamle" DACFOS models estimat af ozonkoncentrationen i Jaegersborg - eller rettere i nogle filtre anvendtes DACFOS modellens 10 niveauer (som 10 h_i'er), i andre filtre anvendtes medianen af de 10 niveauer (som 1 h_i). I nogle filtre anvendtes Hirlam 10 meter temperatur (C) og vindstyrke (m/s). Desuden anvendtes i nogle filtre en sinus-funktion af tiden og en konstant.
Alle disse filtre kørte paa to måder: En version havde i indlæringsperioden kendskab til de direkte målte ozonkoncentrationer $C_{O3}(t)$ for hvert 10. minut. En anden version så ikke disse rå data, men derimod en løbende middelværdi af disse med midlingsperiode paa 3 timer.
Alle filtre kørte med samme valg af parametrene Q,R og τ, nemlig $Q = R = 10$ og $\tau = 200$.

3 Arbejdsopgaver med Kalmanfilteret pr. 1/8/97:

- Man ønsker at bestemme hvilket Kalman filter, der giver den bedste mulige ozon-prognose, herunder flere delopgaver:

  a1) hvad menes med "den bedste mulige ozon-prognose"?
  a2) der er brug for tools til at sammenligne forskellige filtre.

b) hvad betyder valget af, hvilke uafhængige variable (h_i) der inkluderes i filteret. Er det afgørende at inkludere DACFOS, eller er det ligeså godt at bruge hirlam temperatur, vind og evt. sinus og en konstant?

c) hvordan skal parametrene Q,R og tau vaelges?

- DACFOS koerer nu i en ny version, med kun 5 niveauer, og Kalman filteret skal op at koere paa denne nye version.
- Der foreligger DACFOS resultater for en række stationer udover Jaegersborg. Man kan taenke sig at koere kalmanfiltre paa flere stationer.

4 Det har jeg lavet siden 1/8/97:

4.1 Kalman filtering af den nye DACFOS model:

Kalman filteret er nu oppe at køre på den nye DACFOS model. I denne forbindelse er der foretaget ændringer i følgende gamle kode:

- emep biblioteket (i /net/marvin/priv_3/marvin_disperse/DACFOS/libraries/emep/) . Dette bibliotek bruges til at læse DACFOS output–filer.
- Koden der henter data til brug af Kalmanfilteret: get_kalman_data.c
- Selve Kalmanfilteret.
- scriptet der starter Kalmanfilteret — tidligere $Kalman/bin/forecast.scr og $Kalman/test_bin/test_forecast.scr.

Endvidere er der tilføjet følgende nye shell–scripts:

- get_kal_Tv.sh

4.1.1 Ændringer i emep biblioteket

I emep.h er EMEP_ANALYSIS_LENGTH nu sat til 6 (før 12), fordi DACFOS nu kører hver 6. time, og analyse perioden dermed bliver forlænget med 6 timer for hver ny DACFOS kørsel.

4.1.2 Ændringer i get_kalman_data

Nyeste source kode ligger i $Kalman/joj_test/joj_get_kalman_data/ i filen get_kalman_data.c, som kopileres med kommandoen
‘ “gmake -f makefile.get_kalman_data all”.
Der er ændringer i, hvordan get_kalman_data læser DACFOS-filer og hirlam data fra trajektorie filer.
get_kalman_data har fået en ny option -s. Med denne option kan man angive koordinaterne for den station man ønsker at hente DACFOS data for. Default er koordinaterne for stationen i Jægersborg, bredde 55.76 og længde 12.53. Ændringerne i get_kalman_data.c gør, at get_kalman_data nu forstår at læse antallet af stationer, og højder for hver station, i DACFOS-filen, dernæst at finde data for den rigtige station, og skrive disse data i en fil “kf.DACFOS” som før. På denne måde kan get_kalman_data læse gamle som nye DACFOS-filer, dog er det en betingelse, at antallet stationer og højder ikke ændres i de DACFOS-filer der læses (DACFOS filerne for KALMAN-indlæringsperioden).

4.1.3 get_kal_Tv.sh

Der er lidt mere omfattende ændringer i hvordan HIRLAM data for temperaturer og vinde hentes, så de kan læses af kalmanfilteret. Dette skyldes, at der til forskel fra tidligere nu kun ligger trajektorie filer for det seneste døgn i det operationelle $DACFOSHOME (/net/dale/data/dispopr/SMOG/DACFOS/dat). Derfor kører der nu operationelt et script, get_kal_Tv.sh, i /net/dale/data/dispopr/SMOG/DACFOS/bin/get_kal_Tv.sh, der for hver trajektorie fil laver en fil eksempelvis 97090906.hirlam.tv, som indeholder temperaturer (C) og vinde (m/s) for ankomstpunktet Jægersborg for en analyse periode 6 timer tilbage, og for en prognose 48 timer frem. *.hirlam.tv filerne bliver lagt i et direktorie $KALMANTRAJDATA, som er sat til $DACFOS-HOME.
Endvidere ligger der i /net/dale/data/dispopr/SMOG/DACFOS/bin/ to scripts readdatabase.sh og get_kal_Tv.database.sh, som blev brugt til at generere *.hirlam.tv filer for juni, juli og august 1997 udfra databasen af trajektorie filer på unitree.

4.1.4 Ændringer i get_kalman_data — fortsat

get_kalman_data læser *.hirlam.tv filerne og skriver data for hirlam temperaturer og vinde for den rigtige indlæringsperiode og prognose periode i filen

"kf.hirlam", som før. Med en environment variable KALMANTRAJDATA, kan man angive det direktorie, hvor *.hirlam.tv filerne findes.

4.1.5 Ændringer selve Kalman filteret

Der er ikke ændringer i Kalman biblioteket, derimod er der ændringer i den sourcekode, der definerer hvorledes biblioteket anvendes specifikt. Den nyeste version pr. 1/8/97 af denne kode ligger i $Kalman/src/filter_tests.* . Den ændrede version ligger i $Kalman/joj_test/src/ .

- Hvor det før var "hard coded", at der var 10 DACFOS niveauer, er dette nu sat til 5.
- Der er indført en ny option -m, således at man på kommando linien kan angive en værdi for tidskonstanten τ for filterets hukommelse.
- Ændret output til Standard Error: Der bliver nu udskrevet samhørige værdier af filtrede, DACFOS, og observerde ozonkoncentrationer for både indlæringsperiode og prognoseperiode til forskel fra før, hvor det kun blev udskrevet for prognoseperioden. Denne ændring er gjort for at have disse data et sted samlet, så det let kan plottes med eksempelvis xmgr.
- Der er foretaget de nødvendige ændringer i forbindelse med at prognoseperioden nu er 48 timer, mod 36 timer før.
- Det er tilføjet til koden, at den nu finder de forudsagte peak-værdier for ozonkoncentrationen, og sammenligner med observationerne, hvis disse foreligger. Resultatet af dette bliver skrevet til Standard Error.
- Outputtet til *update* og *predict* filerne er ændret, således at disse nu indeholder coefficienterne x_i, samt bidragene $h_i x_i$, som funktion af tiden.

4.1.6 Scriptet der starter Kalmanfilteret

Det filter, der nu kører operationelt, startes af scriptet forecast_joj.scr i direktoriet $Kalman/joj_test/Kalman_Dacfos5_01/, via scriptet crontab.scr, der ligger samme sted. "forecast_joj.scr" kører 10 forskellige Kalmanfiltre, der afviger i valget af hvilke variable h_i, der inkluderes (dette gøres med options "-s i", se CJ). Bemærk, at det kører med $Q = 0.0001$, $R = 1000$ og $\tau = 200$.

Bemærk endvidere, at der ikke foretages nogen midling af observationerne, før de læses af Kalmanfilteret — en sådan midling er uden betydning med parametervalget $Q = 0.0001$ og $R = 1000$ — mere herom nedenfor.

4.1.7 Kørsel af Kalmanfiltre for en historisk periode

Skulle man for eksempel ønske at teste et nyt sæt parametre Q, R, og τ, ligger der model data og observationer, så man umiddelbart kan køre Kalmanfiltre på den nye DACFOS model for juni, juli og august. Direktorierne Kalman_Dacfos5_02/, Kalman_Dacfos5_03/, Kalman_Dacfos5_04/ og Kalman_Dacfos5_05/ i $Kalman/joj_test/ er eksempler hvor dette er gjort. I sidste afsnit i denne rapport vender jeg tilbage til en sammenligning af disse testkørsler. Det er lige til at lave et teste endnu et sæt parameter værdier: Dan for eksempel et direktorie $Kalman/joj_test/Kalman_Dacfos5_06/, med et under direktorie $Kalman/joj_test/Kalman_Dacfos5_06/kalman_data/. Udfør nu følgende kommandoer:

```
cd $Kalman/joj_test/Kalman_Dacfos5_05/
cp forecast_joj ../Kalman_Dacfos5_06
cp forecast_joj.scr ../Kalman_Dacfos5_06
cp get_kalman_data ../Kalman_Dacfos5_06
cp run_kalman.sh ../Kalman_Dacfos5_06
cd ../Kalman_Dacfos5_06
ln -s kalman_data/ data
```

Editer så forecast_joj.scr, og indtast de ønskede værdier for Q_value (Q), R_value (R) og Memory (τ). Kør Kalmanfilrene for hele perioden med:

```
run_kalman.sh > & run_kalman.log &
```

4.2 Hvad er et fornuftigt område for parametrene Q, R og τ?

Figurerne 1, 2, 3 og 4 viser data fra ialt 4 kørsler af Kalman filteret alle med origo for prognose den 14/8/97 kl. 00.00 UTC. Der er 2 forskellige valg af variable h_i: Et filter bruger de 5 DACFOS niveauer, og finder 5 koefficienter, og et andet filter bruger kun middelværdien af de 5 DACFOS niveauer, og finder 1 koefficient. De to filtre er kørt med to valg af parametre:

Det valg som Kalmanfiltret har kørt semi-operationelt med indtil 1/8/97 er $Q = R = 10$. Det andet sæt er $Q = 0.0001$ og $R = 1000$. I alle tilfælde er $\tau = 200$.

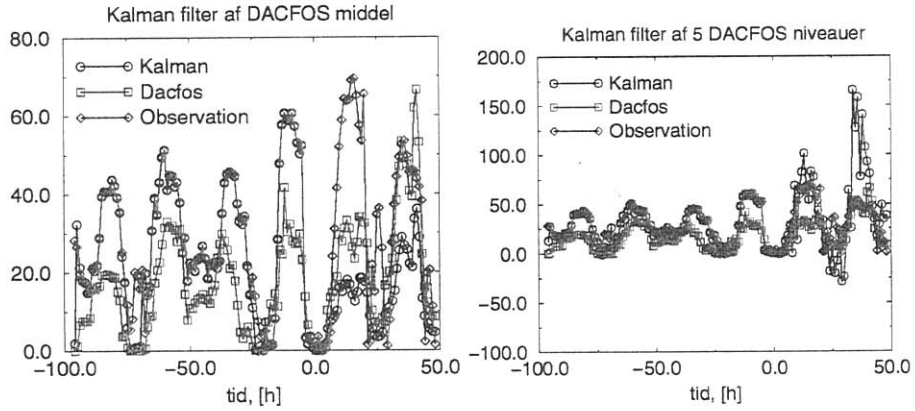

Figure 1: Q=R=10

Figur 1 viser observeret ozonkoncentration sammen med DACFOS resultatet og det Kalman filtrerede resultat, for indlæringsperioden på 4 dage, og prognoseperioden på 2 dage, for begge filtre med parameter valget $Q = R = 10$. Med disse parametre ser man, at begge filtre giver et resultat, der helt følger observationerne i indlæringsperioden, men på ingen måde er en forbedring af DACFOS resultatet i prognoseperioden - tværtimod! Man har en situation, hvor observationerne generelt ligger højere end DACFOS resultatet i indlæringsperioden, og da dette også er tilfældet den første dag i prognoseperioden, burde et fornuftigt parametriseret Kalman filter netop her give en god prognose. $Q = R = 10$ er altså et dårligt parameter valg.
Figur 2 giver et indblik i, hvad der går galt. Figuren viser, hvordan Kalman koefficienterne udvikler sig i de to filtre. Det ene filter har altså 1 koefficient, der bliver ganget på DACFOS middel resultatet, det andet har 5 koefficienter, der bliver ganget på de 5 DACFOS niveauer. Ved at se på den venstre graf på figur 1, vurderer man at koefficienten til DACFOS middel resultatet skal være omtrent 1.5, for nogenlunde at ramme observationerne. Venstre graf på figur 2 viser, at med $Q = R = 10$ varierer denne koefficient voldsomt, og antager ekstreme værdier over 100. Dette skyldes, at Q er for stor. Q angiver fejlen på at koefficienten er konstant fra iteration til iteration i Kalman filteret. Vi ved dens værdi skal være ca. 1.5. Med $Q = 10$ fortæller

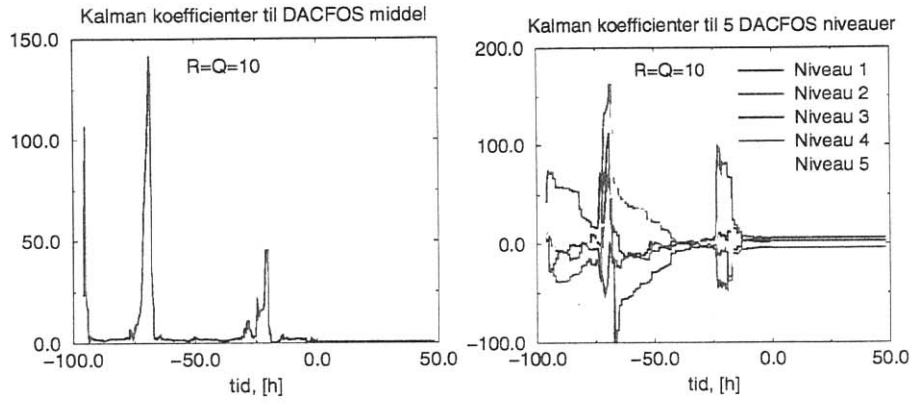

Figure 2: Q=R=10

vi filteret, at den "rigtige" værdi af koefficienten forventes variere med 10 – derfor tillader filteret denne meget store variation af koefficienten. Med $Q = R = 10$ vil filtret hver gang der foreligger en ny observation, kunne sætte koefficienten, så DACFOS middel gange koefficienten rammer observationen godt. Undtagen hvis DACFOS middel resultatet bliver 0 i en kort periode, her vil filteret lade koefficienten vokse til de meget store værdier på kort tid. Værdien af koefficienten ved origo for prognosen bliver dermed meget tilfædig: den afhænger kun af forholdet mellem de aller seneste observationer og DACFOS resultater – ikke af det generelle forløb i indlæringsperioden. Af samme årsag bliver Kalman filterets prognose dårlig.
Højre graf på figur 2 viser udviklingen af de 5 koefficienter til DACFOS niveauerne. Også her ser man, for $Q = R = 10$, en voldsom variation. Her burde filteret være i stand til at finde den rigtige vægtning af de 5 niveauer. Det er derfor ufysisk at have koefficienter på plus og minus 50.
Figur 3 og 4 viser de samme data som figur 1 og 2, nu bare med $Q = 0.0001$ og $R = 1000$. Den lave Q tvinger filteret til at holde koefficienten (evt. koefficienterne) næsten konstant(e). Den høje R fortæller filteret, en ny observation godt kan ligge langt fra de nuværende Koefficienter gange DACFOS resultatet. Som det fremgår af figur 4, tvinger denne kombination af parametre filtret til langsomt at justere koefficienterne ind på en fornuftig værdi igennem indlæringsperioden. Og som det fremgår af figur 3, lærer kalman filtrene at DACFOS undervurderer ozonkoncentrationerne i indlæringsperioden, og kalman prognosen bliver dermed bedre end DACFOS den første dag, hvor

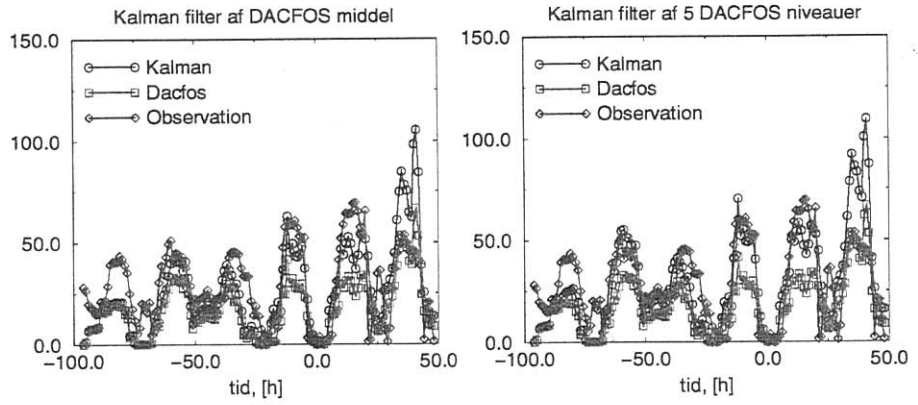

Figure 3: Q=0.0001,R=1000

tendensen holder stik. På anden prognosedag ser man, at mønsteret ændres, og at observationerne ikke overstiger DACFOS prognosen. Dermed kommer kalman filtrene til at skyde for højt. Det sådanne ændringer i mønsteret kan Kalman filteret ikke forventes at kunne forudsige korrekt – med mindre de er velbeskrevet ved en anden variabel, eksempelvis temperatur eller grænselaget tykkelse, som kan inkluderes i filteret.

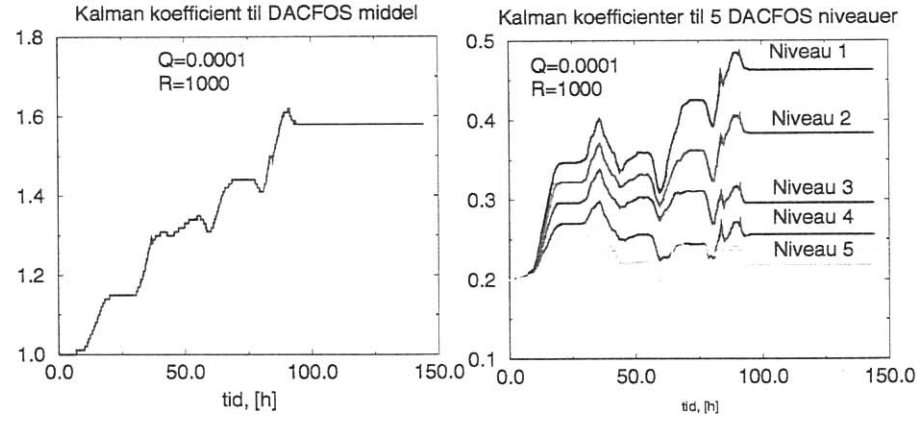

Figure 4: Q=0.0001,R=1000

4.3 Tools til at teste kvaliteten af Kalman filtre

Ovenstående afsnit var en argumentation for, at det rigtige parameter område er lav Q og høj R. Men der er brug for mere omfattende tests af hvilke variable der bør inkluderes i Kalman filteret, og en optimering af Q og R. I denne forbindelse må man køre mange filtre over en længere periode, og så sammenligne kvaliteten af filtrenes prognoser. Derfor er der brug for redskaber til at foretage denne sammenligning. Pr. den 1/8/97 forelå nogle sådanne redskaber, se noten CJ.

4.3.1 plot_pre.sh

Som beskrevet ovenfor, udskriver Kalman filteret nu samhørige værdier af filtrede, DACFOS, og observerde ozonkoncentrationer for både indlæringsperiode og prognoseperiode. "plot_pre.sh" er et shell-script, der bruger "xmgr"-grafik programmet til at plotte disse data, enten for en serie forskelige filtre, hvis output ligger i samme "log-fil", eller et bestemt filter for en serie kørsler, hvor outputtet ligger i en serie "log-filer". Prøv at log ind som brugeren "disperse", og giv kommandoen "plot_pre.sh -h". Eksempelvis vil
cd $Kalman/joj_test/Kalman_test2/data
plot_pre.sh -d 97081400_log
lave en graf for hvert filter i filen 97081400_log. Figur 3 er lavet lavet udfra disse grafer. Et andet eksempel kunne være:
cd $Kalman/joj_test/Kalman_DACFOS5_01/data
plot_pre.sh -f 0 1 0 0 0 0 97071*00_log
som vil lave en graf for det filter, der bruger de 5 DACFOS niveauer som variablene h_i, for hver dag fra den 10/7 til og med den 19/7.
plot_pre.sh ligger i direktoriet $Kalman/joj_test/scripts.

4.3.2 readlog.sh og readlogstr

"readlog.sh" og source koden til "readlogstr" ligger i $Kalman/joj_test/scripts og sidst nævnte kompileres med:
cc -o readlogstr readlogstr.c
"readlog.sh" og "readlogstr" bruges, når man har kørt en serie Kalman filtre med forskellige valg af variable og eller parametre, til at sammenligne disse. "readlog.sh" er et shell-script, der læser en serie af "log-filer", finder data for ozonkoncentrationerne i prognoseperioden, og sender til standard output. Serien af "log-filer" skal indeholde data fra de samme filtre kørt med forskellige origo for prognosen. Endvidere skal filtrene i hver "log-fil"

alle være kørt med samme parameter værdier, men antages at bruge forskellige kombinationer af variablene h_i. Typisk vil man lade "readlog.sh" læse "log-filer" for en eller flere måneder. "readlogstr" er et c-program, der læser outputtet fra "readlog.sh" fra standard input, foretager forskellige statistiske beregner og skriver output i filerne "merr.dat", "mabserr.dat", "rms.dat", "sum.dat" og "ndata.dat".
Lad mig gennemgå "readlog.sh" og "readlogstr" med et eksempel - prøv følgende:
cd $Kalman/joj_test/Kalman_DACFOS5_01/data
readlog.sh 9706*log 9707*log 9708*log | less
Hver linie i outputtet starter med et bogstav, der angiver hvilken type data linien indeholder. 'h' står for 'header' - denne linie indeholder antallet af "log-filer", der er læst, og antallet af forskellige filtre i hver "log-fil". 'v' angiver, at linien indeholder det følgende filters værdier for Q, R og τ. 'u' angiver, at linien indeholder indeks u1, u2, u3, u4, u5, u6 for hvilke variable h_i, det følgende filter anvender. Såfremt "log-filen" indeholder meningsfyldt data for et givet filter, følger nu linier der begynder med 'a' og senere 'f'. 'a'-linierne indeholder obsereverede ozon-koncentrationer for det sidste døgn i indlæringsperioden. Disse data kan senere anvendes til at lave en prognose, under antagelse af persistens. 'f' linierne indeholder så filtrerede, DACFOS og observerede ozon-koncentrationer i prognoseperioden.

4.3.3 "sum.dat" filen

Prøv nu kommandoerne:
cd $Kalman/joj_test/Kalman_DACFOS5_01/data
readlog.sh 9706*log 9707*log 9708*log | readlogstr less sum.dat
Det er en fordel at trække vinduet ud til at være meget bredt. "sum.dat" indeholder et godt grundlag, for at sammenligne kvaliteten af de forskellige filtres prognoser med DACFOS og persistens antagelsen. Alle koncentraioner, fejl i koncentrationer er i ppb. Der er gjort lidt statistik for observationerne: observeret middel ozon-koncentration og dennes spredning, samt middel peak-højde og dennes spredning. Endvidere indeholder sum.dat følgende for DACFOS, persistensantagelsens og filtrenes prognoser:
Total Average: middelværdier for prognoserne over hele den inkluderede periode
$< \text{err} >$: middelfejl eller bias.
$< |\text{err}| >$: middelfejl absolut fejl.
rms : root-mean-square af fejlen.

Peaks for forste og andet døgn i prognose perioden: < err > : middelfejl eller bias på peak–forudsigelserne.
rms : root–mean–square af fejlen på peak–forudsigelserne.

Best performers: I hvor stor en del af alle prognoserne klarede den givne metode sig bedst, når kriteriet for dette er:

< |err| > : Mindst middel absolut fejl for en prognoseperiode.
Peak_1: Tættest på første–døgns peaken.
Peak_2: Tættest på andet–døgns peaken.
high_1: Tættest på første–døgns peaken, når denne var over 60 ppb.
high_2: Tættest på andet–døgns peaken, når denne var over 60 ppb.

4.3.4 "merr.dat", "mabserr.dat" og "rms.dat" filerne

"readlogstr" danner også outputfilerne "merr.dat", "mabserr.dat" og "rms.dat". Disse indeholder henholdsvis fejlen, absolut fejlen og root–mean–square fejlen somfuntkion af tiden efter origo for prognosen, og midlet over alle de indlæste log–filer. Første kolonne i "merr.dat", "mabserr.dat" og "rms.dat" angiver timetallet efter origo, derefter følger data for DACFOS prognosen og persistens prognosen, og derefter for Kalman filtrene i den samme rækkefølge som de står i "sum.dat" filen. Prøv nu kommandoerne:
cd $Kalman/joj_test/Kalman_DACFOS5_01/data
xmgr -nxy rms.dat
Der skulle gerne poppe et vindue med grafer af root–mean–square fejlen op. Ud af x-aksen har vi timetallet efter origo. Den sorte graf er for DACFOS, den røde er for persistens prognosen. Hver data kolonne i rms.dat er blevet til et datasæt i xmgr. Datasættene er nummereret i samme rækkefølge som kolonnerne i filen. Man kan se hvilken graf, der svarer til hvilket datasæt, ved at poppe vinduet "Symbols/legends" op. Det kan bl. a. gøres ved at dobbeltklikke på en af graferne. "merr.dat" og "mabserr.dat" ses på samme måde. Prøv
cd $Kalman/joj_test/Kalman_DACFOS5_01/data
xmgr -nxy merr.dat
Alle prognoser pånær persistens, har en tendens til at undervurdere observationerne om dagen, og måske overvurdere den lidt om natten. Persistens (rød graf) har meget lille bias.

5 Foreløbig performance.

Lad mig tilsidst sammenligne testkørslerne under direktorierne Kalman_Dacfos5_01/, Kalman_Dacfos5_02/, Kalman_Dacfos5_03/, Kalman_Dacfos5_04/ og Kalman_Dacfos5_05/ i $Kalman/joj_test/ . Under hvert af disse direktorier ligger der testkørslerne for en række valg af variable h_i. Forskellen mellem direktorierne er valget af parametre – dette valg fremgår af tabel 1.

Direktorie	Q	R	τ
$Kalman/joj_test/Kalman_Dacfos5_01	0.0001	1000	200
$Kalman/joj_test/Kalman_Dacfos5_02	0.001	100	400
$Kalman/joj_test/Kalman_Dacfos5_03	0.0003	300	400
$Kalman/joj_test/Kalman_Dacfos5_04	0.00003	3000	400
$Kalman/joj_test/Kalman_Dacfos5_05	10	10	200

Table 1: parameter værdier for testkørsler.

Jeg vil foretage sammenligningen ved at kigge på "sum.dat" filerne, der ligger i underdirektorierne "kalman_data/". Indholdet af disse "sum.dat" filer er gennemgået ovenfor.
Lad os kigge lidt på filen kalman_data/sum.dat under Kalman_Dacfos5_01 . Vi ser, at den er dannet på baggrund af 92 logfiler – det er imidlertid ikke alle disse, der indeholder data. Den observerede middel ozon–koncentration har været 28.44 ppb. Middel peak-værdien har været 43.49 ppb, og spredningen på denne har været 10.73 ppb. Et ikke urimeligt ønske til kvaliteten af en prognose må være, at den kan forudsige den følgende dags peak med en root–mean–square fejl mindre en den naturlige variation på 10.73 ppb. Umiddelbart må man forvente, at en persistensprognose må kan gøre det i hvert fald så godt. Vi ser, at DACFOS ikke har levet op til dette, med en rms på første peak på 15.92 ppb. Endvidere har ingen af disse Kalmanfiltre levet op til ønsket. Kun persistens prognosen har naturligt nok netop opfyldt kriteriet. De bedste Filtre til bestemmelse af første peaks højde er nummer 7 og 8, som har u1=u2=u3=1 (DACFOS middelresultatet samt hirlam temperatur (C)) og nummer 8 har endvidere u4=1 (hirlam vind). De to filtre for rms på første peak på henholdsvis 12.57 og 12.70 ppb.
Kigger vi på den total root–mean–square fejl for hele perioden, ser vi, at den bedste prognose stammer fra filter nummer 8 med u1=u2=u3=u4=1 og

u5=u6=0, som har en rms-fejl på 12.21 ppb. Vi ser, at også hvad rms-fejlen angår, er persistensprognosen bedre end DACFOS.
Hvem kommer oftest tættest på første peak? Igen persistens, som kommer tættest i 34 % af prognoserne. Hvem kommer så tættest på første peak, når denne er over 60 ppb? Her er det interessant nok det første filter (u2=1), som kommer tættest i 3 ud af de ialt 7 sådanne tilfælde i perioden. Dette filter bruger de 5 DACFOS niveauer, som 5 h_i, og er altså bedst til at forudsige hvornår der kommer høje peaks.
Hvordan så med de andre parameter valg? Altså testkørslerne under direktorierne Kalman_Dacfos5_02/, Kalman_Dacfos5_03/, Kalman_Dacfos5_04/ og Kalman_Dacfos5_05/ i $Kalman/joj_test/ ? Ved at gennemgå sum.dat filerne, på lignende måde som ovenfor, ser man at ingen af disse filtre kommer med bedre prognoser end dem under Kalman_Dacfos5_01/ . Næstbedst bliver nok filtrene under Kalman_Dacfos5_04/ . Helt galt går det for filtrene under Kalman_Dacfos5_05/ .