

Data-assimilaatiomenetelmistä ja niiden  
soveltamisesta ROSE-kemiakuljetusmallin

*Pro Gradu-tutkielma*

Janne Hakkarainen

Ohjaajat

Marko Laine ja Heikki Haario

HELSINGIN YLIOPISTO  
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA  
MATEMATIIKAN JA TILASTOTIETEEN LAITOS

Marraskuu 2009





Tiedekunta/Osasto – Fakultet/Sektion – Faculty Matemaattis-luonnontieteellinen tiedekunta		Laitos – Institution – Department Matematiikan ja tilastotieteen laitos	
Tekijä – Författare – Author Janne Hakkarainen			
Työn nimi – Arbetets titel – Title Data-assimilaatiomenetelmistä ja niiden soveltamisesta ROSE-kemiakuljetusmalliin			
Oppiaine – Läroämne – Subject Matematiikka			
Työn laji – Arbetets art – Level Pro gradu		Aika – Datum – Month and year Marraskuu 2009	Sivumäärä – Sidoantal – Number of pages 63
<p>Tiivistelmä – Referat – Abstract</p> <p>Data-assimilaatio on tekniikka, jossa havaintoja yhdistetään dynaamisiin numeerisiin malleihin tarkoituksena tuottaa optimaalista esitystä esimerkiksi ilmankehän muuttuvasta tilasta. Data-assimilaatiota käytetään muun muassa operaatiivisessa sään ennustamisessa.</p> <p>Tässä työssä esitellään eri data-assimilaatiomenetelmiä, jotka jakautuvat pääpiirteittäin Kalmanin suotimiin ja variaatioanaalisiin menetelmiin. Lisäksi esitellään erilaisia data-assimilaatiossa tarvittavia apuvälineitä kuten optimointimenetelmiä. Eri data-assimilaatiomenetelmien toimintaa havainnollistetaan esimerkkien avulla.</p> <p>Tässä työssä data-assimilaatiota sovelletaan muun muassa Lorenz95-malliin. Käytännön data-assimilaatio-ongelmana on GOMOS-instrumentista saatavan otsonin assimiloiminen käyttäen hyväksi ROSE-kemiakuljetusmallia.</p>			
Avainsanat – Nyckelord – Keywords Data-assimilaatio, Kalmanin suodin, Kemiakuljetusmalli			
Säilytyspaikka – Förvaringställe – Where deposited Kumpulan tiedekirjasto			
Muita tietoja – Övriga uppgifter – Additional information			



# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
1.1	Notaatiosta . . . . .	1
1.2	Kiitokset . . . . .	2
<b>2</b>	<b>Data-assimilaatio</b>	<b>3</b>
2.1	Mitä data-assimilaatio on? . . . . .	3
2.2	Data-assimilaatio osana ilmakehän kaukokartoitusta . . . . .	5
2.3	Data-assimilaation hyödyt . . . . .	6
<b>3</b>	<b>Data-assimilaatiomenetelmiä</b>	<b>7</b>
3.1	Kalmanin suotimet . . . . .	13
3.1.1	Tavallinen lineaarinen Kalmanin suodin . . . . .	13
3.1.2	Kalmanin silotin . . . . .	15
3.1.3	Laajennettu Kalmanin suodin . . . . .	17
3.1.4	Ensemble Kalmanin suodin . . . . .	21
3.1.5	Paikallinen ensemble-muunnos Kalmanin suodin . . . . .	23
3.1.6	Muista Kalmanin suotimista . . . . .	27
3.2	Variaationaaliset data-assimilaatiomenetelmät . . . . .	28
3.2.1	4D-Var . . . . .	30
3.2.2	Optimointimenetelmiä . . . . .	32
3.2.3	Adjungaattimenetelmistä . . . . .	36
3.2.4	Variaationaalinen Kalmanin suodin . . . . .	38
<b>4</b>	<b>Numeeriset esimerkit</b>	<b>41</b>
4.1	Brusselator . . . . .	41
4.2	Lorenz95 . . . . .	45
<b>5</b>	<b>Data-assimilaatio ROSE-kemiakuljetusmallissa</b>	<b>49</b>
5.1	Malli ja data . . . . .	49
5.2	Assimilaatiomenetelmä . . . . .	50
5.3	Tuloksia ja johtopäätöksiä . . . . .	52



# Luku 1

## Johdanto

Tämä työ käsittelee data-assimilaatiota[4] ja siihen liittyviä menetelmiä ja sovelluksia. Työ rakentuu tämän johdantoluvun lisäksi neljään erilliseen lukuun, joista kaksi ensimmäistä käsittelevät data-assimilaatiota ja menetelmiä ja kaksi seuraavaa niiden sovelluksia. Liitteessä A esittelemme yleisiä matemaattisia ja tilastollisia käsitteitä, sekä näihin liittyvää notaatiota. Osan käsitteistä ja ominaisuuksista oletamme kuitenkin tunnetuiksi, kuten esimerkiksi tavalliset matriisioperaatiot.

Tämän työn teoriaosuudessa tulemme esittelemään keskeisimpiä data-assimilaatio-tekniikoita pyrkien esittämään menetelmien taustalla olevan idean ja mahdollisesti havainnollistamaan menetelmän toimintaa esimerkkien valossa. Useassa tapauksessa tulemme myös johtamaan tekniikat kirjallisuudesta tutuilla tavoilla. Kaikkia kaavoja emme ole kuitenkaan johtaneet ja silloin viittaammekin lähdekirjallisuuteen. Tulemme käsittelemään lyhyesti myös data-assimilaatiossa käytettyjä apuvälineitä kuten optimointimenetelmiä.

Tämä työ on tehty yhteistyössä Ilmatieteen laitoksen kanssa ja varsinaisessa sovellusosuudessa käytetty data on oikeata satelliiteista havaittua dataa ja käytetty malli on oikea ilmakehän kemialla kuvaava malli. Tällä työllä onkin kaksi tarkoitusta: Toisaalta esitellä kattavasti eri tekniikoita ja toisaalta rakentaa toimiva data-assimilaatiosysteemi Ilmatieteen laitoksen tarpeisiin. Joitain käytännön data-assimilaatiossa esiintyviä ongelmia kuten esimerkiksi kovarianssimatriisien arviointia ja havainto-operaattorin luontia ei ole kuitenkaan käsitelty.

### 1.1 Notatiosta

Tässä työssä merkitsemme vektoria pienellä kursiivilla kirjaimella (esim.  $x$ ) ja matriisia isolla kursiivilla kirjaimella (esim.  $C$ ). Epälineaarista funktio-

ta merkitsemme kaunokirjoitussymbolilla (esim.  $\mathcal{H}$ ). Alaindeksiä käytämme merkitsemään vektorin alkion ja yläindeksiä kuvaamaan ensemblen elementtiä. Taulukossa 1.1 on esitetty keskeisimmät lyhenteet ja symbolit. Huomautamme, että lähteistä riippuen notaatio data-assimilaatiossa vaihtelee suuresti. Data-assimilaatiolle on luotu standardinotaatio [11], jota emme kuitenkaan käytä tässä työssä. Termillä analyysi tarkoitamme data-assimilaatiossa tehtyä analyysiä emme perinteisessä mielessä matematiikasta tunnettua analyysiä.

Taulukko 1.1: Symbolit ja lyhenteet.

Määrite	Selite
$est$	Estimaatti, analyysi
$a$	A priori, tausta
$true$	Totuus
$obs$	Havainto
$t$	Aika
$B, C_a$	Taustavirheen kovarianssimatriisi $n \times n$
$R, C_\varepsilon$	Havaintovirheen kovarianssimatriisi $p \times p$
$A, C_{est}$	Analyysivirheen kovarianssimatriisi $n \times n$
$C_\xi$	Mallivirheen kovarianssimatriisi $n \times n$
$x$	Tilavektori $n \times 1$
$y$	Havaintovektori $p \times 1$
$\varepsilon$	Virhe(vektori)
$G$	Vahvistusmatriisi
$M, \mathcal{M}$	Mallioperaattori
$H, \mathcal{H}$	Havainto-operaattori
$J$	Sakkofunktio
$I$	Identtinen matriisi

## 1.2 Kiitokset

Erityiskiitos Marko Laineelle ja Heikki Haariolle erinomaisesta ohjauksesta. Kiitos myös kaikille niille, jotka antoivat vinkkejä, kommentteja ja tukea. Lisäksi kiitos koko Ilmatieteen laitoksen kaukokartoitusryhmälle; erityisesti Simolle jatkuvista hyvistä vinkeistä ja Johannalle mahdollisuuksien antamisesta.



# Luku 2

## Data-assimilaatio

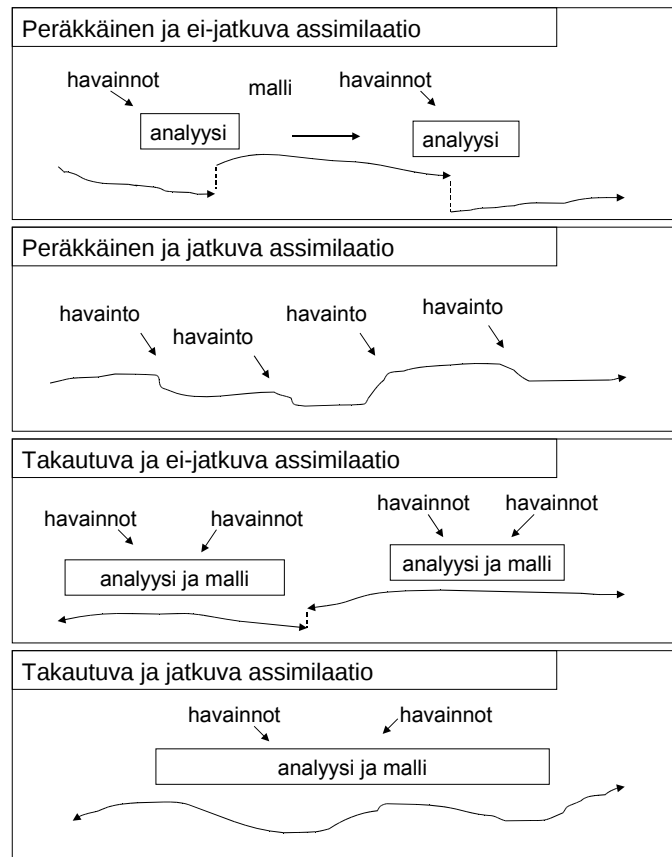
Data-assimilaatiota on perinteisesti käytetty sään ennustamisessa, mutta sitä on sovellettu menestyksekkäästi myös monilla muilla tieteen aloilla. Historiallisesti data-assimilaation voidaankin katsoa alkaneen 1950-luvulla kun menetelmiä numeeriseen sään ennustamiseen ruvettiin kehittämään. Satelliiteista saatavat mittaukset ja tietokoneiden kehittyminen ovat vaikuttaneet data-assimilaation suosioon merkittävästi, ja se on vielä tänäkin päivänä aktiivisen tutkimuksen kohteena. Tässä luvussa käsitellään erityisesti data-assimilaation soveltamisesta ilmakehän kaukokartoitukseen sekä data-assimilaation hyötyjä ja peruseriaatteita.

### 2.1 Mitä data-assimilaatio on?

Data-assimilaatio on tekniikka, jossa havainnot yhdistetään dynaamisiin numeerisiin malleihin tarkoituksena tuottaa *optimaalista* esitystä esimerkiksi ilmankehän muuttuvasta tilasta [20]. Optimaalisella tarkoitetaan sitä, että malleista ja havainnoista saatavaa informaatiota yhdistetään tavalla, jossa estimoidun tilan (mahdollisesti tunnettu) virhearvio on mahdollisimman pieni. Malli tuottaa yhtenäisyyttä havaintoihin ja interpoloi niitä johdonmukaisella tavalla myös alueille, joissa havainnot ei ole tehty. Havainnot korjaavat epätäydellisestä mallista saatavaa liikekaarta pitäen sen raiteillaan. Laskentatehon kasvaessa yhä enemmän huomiota on kiinnitetty optimaalisiin menetelmiin.

Tiivistäen voidaan sanoa, että data-assimilaatiolla on seuraavia käyttötapoja [6]. Data-assimilaation avulla

1. voidaan *suodattamalla* erottaa signaali kohinaisista havainnoista,
2. havainnot voidaan *interpoloida* tilassa ja ajassa ja



Kuva 2.1: Eri lähtökohtia data-assimilaatioon. Huomaa nuolten suunnat.

3. saadaan *täydentämällä* tietoa myös muista tilamuuttujista, joista ei ole havaintoja.

Data-assimilaatioon on kaksi perusteeltaan erilaista lähestymistapaa: *peräkkäinen* ja *takautuva*. Peräkkäisessä huomioidaan (vain) mittaushetkeen asti tehty analyysi. Takautuvassa voidaan käyttää hyväksi sekä analysointihetken ennen että jälkeen tehtyjä havaintoja. Peräkkäisiä menetelmiä käytetään esimerkiksi reaaliaikaisissa assimilaatiosysteemeissä. Lähestymistapoja on esitelty kuvassa 2.1. Menetelmiä, jotka toimivat peräkkäin kutsutaan usein *suotimiksi* (eng. *filter*) ja vastaavasti takautuvia menetelmiä kutsutaan usein *silottajiksi* (eng. *smoother*). Tässä työssä keskitymme lähinnä peräkkäisiin menetelmiin.

## 2.2 Data-assimilaatio osana ilmakehän kaukokartoitusta

Satelliiteista tehtyä ilmakehän kaukokartoitusta käytetään sään ennustamiseen ja globaalin ilmakehän tutkimiseen. Ilmakehän havaintodataa saadaan useista eri lähteistä ja se on usein varsin heterogeenista. Data voi olla useista eri satelliiteista saatavia radianssi-mittauksia, sondeista saatavia mittauksia ja niin edelleen. Saadaksemme maksimaalisen hyödyn datasta meidän täytyy löytää tapa yhdistää mittaukset. Usein tavoitteena on muodostaa globaali kuva tutkittavasta ilmiöstä. Haluamme myös, että analyysimme on johdonmukainen ilmakehän fysiikan ja kemian lainalaisuuksien kanssa. Moderni tapa saavuttaa tämä päämäärä on data-assimilaatio.

Ilmakehän muutokset ovat fysikaalisessa mielessä jatkuvia, joten haluamme, että tekemämme analyysikin on. Perusteeltaan ilmakehämallit ovat monimutkaisia ja ylimääräisenä hankaluutena – vaikkakin ilmakehän dynamiikkaa kuvaavat kaavat ovatkin periaatteessa tunnettuja – on, että tila muuttuu ajan funktiona. Käytännössä vaatimuksena on, että kaikki havainnot käytetään hyödyksi – ainoastaan virheellinen data hylätään. Kaukokartoituksessa data-assimilaatiota onkin käytetty optimoimaan kallista satelliittidataa. Data-assimilaatiota on alettu soveltamaan myös ilmakehän kemiankuljetusmalleihin.

Globaali analyysiongelma on samanlainen kuin kaukokartoituksessa usein esiintyvä (esimerkiksi otsoni-)profiilien hankintaongelma, mutta huomattavasti laajempi [22]. Tilavektori on annetussa hilassa koko ilmakehän tila, toisin kuin yksittäinen profiili. Viime aikoihin asti monet analyysimenetelmät pystyivät ainoastaan assimiloimaan mallimuuttujia itsessään, niin että satelliittidatasta oli ensin hankittava pystyprofiilit ennen kuin niitä pystyttiin käyttämään. Kuitenkin suoria assimilointimenetelmiä – ilman kaukomittauksista välissä olevaa retrieval-askelta – on nyt ryhdytty kehittämään.

Data-assimilaatiossa on tarpeen löytää paras tilaestimaatti, joka on sopu-suinnassa havaintojen ja saatavilla olevan priori-informaation kanssa. Kalmanin suodin (3.1) on suunniteltu täsmälleen näihin olosuhteisiin, mutta globaali analyysi on niin laaja ongelma, että Kalmanin suodimen suora käyttö ilman approksimaatioita on vielä osin mahdotonta toteuttaa. Tällä hetkellä nopeampia ja yksinkertaisempia menetelmiä on yleisesti käytössä, jotka perustuvat muun muassa optimointiteoriaan.

Ilmakehän tila voidaan määritellä joukolla mallimuuttujia kuten lämpötila, paine, tuulten nopeudet ja ilmankosteus. Nämä esitetään yleensä yhdessä korkeuden kanssa karteesisessa pituus-leveyspiiri-hilassa, vaikkakin on monia yhtäpitäviä tapoja esittää tila, kuten ikosahedrisessä hilassa. Erityisesti sään

ennustuksessa lopputuotteen tarkkuus riippuu suuresti siitä mikä globaali kenttä esittää yksittäisellä ajanhetkellä ilmakehän todellista tilaa.

## 2.3 Data-assimilaation hyödyt

Tässä kappaleessa kootaan yhteen jo aiemmissa kappaleissa esiintyneitä data-assimilaation hyötyjä. Seuraavassa listassa [20] on data-assimilaation keskeisimmät edut:

**Laadunvalvonta:** Priori-informaatio antaa menetelmän laadunvalvontaan. Vertailemalla analyysiämme priori-informaatioon voimme määrittää ovatko assimilaationsysteemin lähtötiedot virheellisiä.

**Havaintojen yhdistäminen:** Data-assimilaation avulla voidaan yhdistää ja sovittaa havaintoja ja prioritietoa eri lähteistä saadaksemme parhaan tilaestimaatin.

**Virheet datassa ja mallissa:** Vertailemalla havaintoja ja priori-informaatiota tilaestimaattiin voimme muodostaa virhearvioita, joilla voimme kehittää instrumentteja ja menetelmiä.

**Puuttuvien havaintojen täydentäminen:** Malli mahdollistaa informaation levittämisen myös alueille, joista havaintoja ei ole tai niitä on vähän.

**Fysikaalisen ja kemiallisen tasapainon säilyminen:** Malli varmistaa, että fysikaaliset ja kemialliset suhteet muuttujissa säilyvät huolimatta havainnoissa olevista virheistä, jotka muuten saattaisivat johtaa epä johdonmukaisuuksiin.

**Havaitsemattomien suureiden estimointi:** Mallissa olevien muuttujien suhteiden avulla voimme saada informaatiota myös muuttujista, joita ei ole havaittu.

## Luku 3

# Data-assimilaatiomenetelmiä

Tässä luvussa esitellään eri data-assimilaatiomenetelmiä. Pääpiirteittäin menetelmät voidaan jakaa kahteen eri luokkaan: Kalmanin suotimiin ja variaationaalisiin menetelmiin. Aluksi menetelmiä käsitellään hyvin yleisellä tasolla. Kappaleissa (3.1) ja (3.2) käsitellään eri menetelmiä yksityiskohtaisemmin. Aluksi kiinnitetään notaatiota ja määritellään data-assimilaatioon liittyviä yleisiä käsitteitä. Taustalla pyritään pitämään ajatusta mahdollisista sovelluskohteista ilmakehän mallinnukseen.

Lähdettäessä muodostamaan ongelman matemaattista formulointia on tärkeää määritellä niin sanottu tila-avaruus (eng. state space). Esimerkiksi sääennustusmallissa ilmakehän tilaa kuvataan tietyllä hetkellä<sup>1</sup>  $t$  tilavektorigilla  $x \in \mathbb{R}^n$ . Diskretointi eli kannan valinta vaikuttaa siihen miten tilavektorin komponentit suhtautuvat todelliseen tilaan; usein kaukokartoituksessa tilavektori esitetään esimerkiksi pituus–leveyspiiri–hilassa. Tilavektoria, joka parhaiten annetuissa kannassa kuvaa todellisuutta merkitsemme symbolilla  $x_{true}$ . Yleensä käytössämme on myös jotain tausta- eli *priori*-informaatiota, joka voi perustua esimerkiksi edelliseen analyysiin. Vektoria, joka kuvaa ennakkokäsitystämme tilasta merkitsemme symbolilla  $x_a$ . Ja analyysin avulla saatua estimaattia, joka kuvaa käsitystämme todellista tilasta merkitsemme symbolilla  $x_{est}$ .

Mittausten avulla saatua havaintotietoa merkitsemme *havaintovektorilla*  $y \in \mathbb{R}^p$ . On hyvä huomata, että havaintovektorin ja tilavektorin dimensio ei useinkaan ole sama. Itse asiassa systeemimme ovat lähes poikkeuksetta *alimääräytyjä* eli mittauksia on (paljon) vähemmän kuin mallissa on parametreja;  $\dim(x) \gg \dim(y)$ . Funktiota joka liittyy mallin tila-avaruuden havaintoavaruuteen kutsutaan havainto-operaattoriksi  $\mathcal{H}$ . Käytännössä  $\mathcal{H}: \mathbb{R}^n \rightarrow \mathbb{R}^p$  on kokoelma interpolaatio-operaattoreita mallidiskretoinnista havaintopistei-

---

<sup>1</sup>Mikäli on tarpeen, niin merkitsemme ajanhetken näkyviin alaindeksinä  $x_t$ .

siin ja käännöksiä mallimuuttujista havaintoparametreihin. Yksinkertaisimmillaan  $\mathcal{H}$  voisi olla identtinen matriisi.

Käytännössä havaintoihin, priori-informaatioon ja tehtyyn analyysiin liittyy aina epävarmuutta, jolloin on tarvetta tilastollisiin tarkasteluihin. Tällöin jokaiselle virhetyypille oletetaan jokin todennäköisyys-, usein esimerkiksi gaussinen, jakauma. Kirjoitetaan muuttujat ja niihin liittyvät virheet seuraavasti

$$\begin{aligned}x_a &= x_{true} + \varepsilon_a \\y &= \mathcal{H}(x_{true}) + \varepsilon_{obs} \\x_{est} &= x_{true} + \varepsilon_{est}.\end{aligned}$$

Tällöin virheet taustalla ja havainnossa mallinnetaan seuraavasti:

**priorivirhe:**  $\varepsilon_a = x_a - x_{true}$ , jolla on kovarianssimatriisi  $B$ .

**havaintovirhe:**  $\varepsilon_{obs} = y - \mathcal{H}(x_{true})$ , jolla on kovarianssimatriisi  $R$ .

**analyysivirhe:**  $\varepsilon_{est} = x_{est} - x_{true}$ , jolla on kovarianssimatriisi  $A$ .

Yleensä oletetaan, että virheiden odotusarvot ovat nolliä. Tällöin esimerkiksi priorivirheen kovarianssimatriisi voidaan laskea kaavalla  $B = E(\varepsilon_a \varepsilon_a^T)$ .

Haluamme, että estimaattimme  $x_{est}$  on lineaarinen pienimmän varianssin harhaton estimaatti eli niin kutsuttu BLUE-estimaatti (eng. Best Linear Unbiased Estimate). Merkitään  $\varepsilon_{est} = (\varepsilon_{est,1}, \dots, \varepsilon_{est,n})^T$ . Tarkasteellaan matriisin  $A$  jälkeä. Kovarianssimatriisin diagonaalilla ovat varianssit eli

$$\text{diag}(A) = (\text{var}(\varepsilon_{est,1}), \dots, \text{var}(\varepsilon_{est,n}))^T,$$

joten  $A$ :n jälki on

$$\begin{aligned}\text{Tr}(A) &= \sum_{i=1}^n \text{var}(\varepsilon_{est,i}) \\&= E\left(\sum_{i=1}^n (\varepsilon_{est,i} - \overline{\varepsilon_{est}})^2\right) = E(\|\varepsilon_{est} - \overline{\varepsilon_{est}}\|^2).\end{aligned}$$

Edellisessä kaavassa oleva summa on se, jonka haluamme minimoida. Haluamme lisäksi, että analyysivirheen odotusarvo on nolla eli estimaattimme on harhaton.

Olkoot priori- ja havaintovirheet harhattomia ja keskenään korreloimattomia ja olkoot niiden kovarianssimatriisit positiivisesti definiittejä. Oletetaan

ensin, että havainto-operaattori  $\mathcal{H}$  on lineaarinen taustatilan läheisyydessä jolloin

$$\mathcal{H}(x) - \mathcal{H}(x_a) = H(x - x_a), \quad (3.1)$$

missä  $H \in \mathbb{R}^{n \times p}$  on lineaarinen operaattori. Nyt voimme muotoilla seuraavan lauseen [4]:

**Lause 3.0.1.** *Olkoot muuttujat kuten edellä. Tällöin lineaarinen pienimmän varianssin harhaton estimaatti saadaan seuraavista yhtälöistä:*

$$x_{est} = x_a + G(y - \mathcal{H}(x_a)), \quad (3.2)$$

missä

$$G = BH^T(HBH^T + R)^{-1}. \quad (3.3)$$

Lineaarista operaattoria  $G$  kutsutaan vahvistusmatriisiksi (eng. gain matrix). Analyysivirheen kovarianssimatriisi  $A = \text{cov}(\varepsilon_{est})$  voidaan laskea seuraavasti

$$A = (I - GH)B. \quad (3.4)$$

*Todistus.* Olkoon  $G$  mielivaltainen matriisi ja määritellään  $x_{est}$  kaavalla 3.2. Tällöin

$$\begin{aligned} \varepsilon_{est} - \varepsilon_a &= x_{est} - x_a = G(y - \mathcal{H}(x_a)) \\ &= G(\varepsilon_{obs} + \mathcal{H}(x_{true}) - \mathcal{H}(x_a)) \\ &= G(\varepsilon_{obs} + H(x_{true} - x_a)) \\ &= G(\varepsilon_{obs} - H(\varepsilon_a)), \end{aligned}$$

joten

$$\varepsilon_{est} = G\varepsilon_{obs} - (I + GH)\varepsilon_a. \quad (3.5)$$

Nyt

$$\begin{aligned} \varepsilon_{est}\varepsilon_{est}^T &= (G\varepsilon_{obs} + (I - GH)\varepsilon_a)(G\varepsilon_{obs} + (I - GH)\varepsilon_a)^T \\ &= G\varepsilon_{obs}\varepsilon_{obs}^T G^T + (I - GH)\varepsilon_a\varepsilon_a^T(I - GH)^T \\ &\quad + G\varepsilon_{obs}\varepsilon_a^T(I - GH)^T + (I - GH)\varepsilon_a\varepsilon_{obs}^T G^T \end{aligned}$$

Koska oletimme, että  $E(\varepsilon_a) = E(\varepsilon_{obs}) = 0$ , niin yhtälöstä (3.5) seuraa suoraan, että  $E(\varepsilon_{est}) = 0$  eli estimaattimme on harhaton. Nyt siis ottamalla puolittain odotusarvon ja käyttämällä oletusta siitä, että ristikovarianssi on

nolla ja huomioimalla  $B$ :n ja  $R$ :n määritelmät saamme analyysivirheen kovarianssimatriisiksi

$$A = E(\varepsilon_{est}\varepsilon_{est}^T) = G E(\varepsilon_{obs}\varepsilon_{obs}^T)G^T + (I - GH) E(\varepsilon_a\varepsilon_a^T)(I - GH)^T \quad (3.6)$$

$$+ G E(\varepsilon_{obs}\varepsilon_a^T)(I - GH)^T + (I - GH) E(\varepsilon_a\varepsilon_{obs}^T)G^T \quad (3.7)$$

$$= (I - GH)B(I - GH)^T + GRG^T. \quad (3.8)$$

Koska haluamme, että estimaattimme  $x_{est}$  on lineaarinen pienimmän varianssin estimaatti, niin meidän tulee minimoida sen kovarianssimatriisin jälki  $\text{Tr}(A)$ . Nyt  $A$ :n yleisestä muodosta (3.8) saamme

$$\text{Tr}(A) = \text{Tr}(B) - 2 \text{Tr}(BH^T G^T) + \text{Tr}(GHBH^T G^T) + \text{Tr}(GRG^T).$$

Olkoon  $L$  nyt mielivaltainen testimatriisi. Lasketaan matriisiderivaatta (katso liite A).

$$\begin{aligned} \frac{d}{dG}(\text{Tr}(A))L &= 2 \text{Tr}(GHBH^T L^T) - 2 \text{Tr}(BH^T L^T) + 2 \text{Tr}(GRL^T) \\ &= 2 \text{Tr}(GHBH^T L^T - BH^T L^T + GRL^T) \\ &= 2 \text{Tr}((G(HBH^T + R) - BH^T)L^T) \end{aligned}$$

Viimeiseltä riviltä saamme, että derivaatta on nolla kaikilla  $L$ , jos ja vain jos  $G = BH^T(HBH^T + R)^{-1}$ .

Olkoon  $G$  nyt kuten edellä. Näytetään, että tällöin

$$-(I - GH)BH^T G^T + GRG^T = 0.$$

Tällöin (3.8) sievenee haluttuun muotoon (3.4). Suoraan laskemalla saamme

$$\begin{aligned} -(I - GH)BH^T G^T + GRG^T &= -((I - GH)BH^T - GR)G^T \\ &= -((BH^T - G(HBH^T + R))G^T, \end{aligned}$$

mikä on nolla kun huomioidaan  $G$ :n määritelmä. □

**Huomautus.** Tarvitsimme  $\mathcal{H}$  lineaarisuuden taustatilan läheisyydessä johtaaksemme tarkat algebralliset yhtälöt. Käytännössä  $\mathcal{H}$  ei välttämättä ole lineaarinen, mutta usein voimme kuitenkin olettaa, että taustatilan läheisyydessä

$$\mathcal{H}(x) - \mathcal{H}(x_a) \approx H(x - x_a),$$

eli hieman täsmällisemmin

$$\mathcal{H}(x + h) = \mathcal{H}(x) + Hh + \varepsilon(h),$$



missä  $\varepsilon(h) \rightarrow 0$ , kun  $h \rightarrow 0$ .

**Ehdolliset jakaumat ja Bayesin kaava.** Tarkastellaan tilannetta hie-  
man toisella tavalla käyttäen ehdollisia todennäköisyysjakaumia. Kiinnitetään  
aluksi hiukan merkintöjä. Merkinnällä  $P(A|B)$  tarkoitetaan tapahtuman  $A$   
todennäköisyyttä, kun tiedetään tapahtuman  $B$  tapahtuneen. Vastaavasti  
merkinnällä  $p(x|y)$  tarkoitamme satunnaismuuttujan  $x$ :n tiheysfunktioita eh-  
dolla, että muuttujan  $y$  arvot ovat tunnettuja.

**Bayesin kaava.** Jos  $p(y) \neq 0$ , niin tällöin niin sanottu posterioritiheys-  
funktio  $p(x|y)$  voidaan laskea seuraavasti:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \quad (3.9)$$

missä  $p(x)$  on priorijakauma,  $p(y|x)$  uskottavuusfunktio ja  $p(y) = \int p(y|x)p(x) dx$   
normeeraustekijä.

Oletetaan nyt, että  $x \sim N(x_a, B)$  ja  $y|x \sim N(\mathcal{H}(x), R)$ . Tällöin

$$p(x) = \frac{1}{(2\pi)^{n/2} \det(B)^{1/2}} \exp\left(-\frac{1}{2}(x_a - x)^T B^{-1}(x_a - x)\right)$$

$$p(y|x) = \frac{1}{(2\pi)^{m/2} \det(R)^{1/2}} \exp\left(-\frac{1}{2}(y - \mathcal{H}(x))^T R^{-1}(y - \mathcal{H}(x))\right).$$

Bayesin kaavaan (3.9) sijoittamalla saamme

$$p(x|y) = c \exp\left(-\frac{1}{2}((x_a - x)^T B^{-1}(x_a - x) + (y - \mathcal{H}(x))^T R^{-1}(y - \mathcal{H}(x)))\right),$$

missä  $c$  on vakio, joka ei riipu  $x$ :stä. Ratkaisua joka maksimoi edellisen funk-  
tion  $x$ :n suhteen kutsumme suurimman posteriorin estimaatiksi (eng. Maxi-  
mum A Posteriori estimate, MAP-estimate) eli MAP-estimaatiksi. Tämän  
kanssa on yhtäpitävää minimoida seuraavaa funktiota. Ottamalla puolittain  
logaritmin saamme

$$-2 \ln(p(x|y)) - \ln c = (x_a - x)^T B^{-1}(x_a - x) + (y - \mathcal{H}(x))^T R^{-1}(y - \mathcal{H}(x)).$$

Variaationaalissa data-assimilaatiossa on tarkoitus minimoida *sakko-*  
*funktio* (eng. *penalty function* tai *cost function*):

$$J(x) = (x_a - x)^T B^{-1}(x_a - x) + (y - \mathcal{H}(x))^T R^{-1}(y - \mathcal{H}(x)) \quad (3.10)$$

$$= J_a(x) + J_{obs}(x), \quad (3.11)$$

missä  $J_a(x)$  on *prioritermi* ja  $J_{obs}(x)$  on *havaintotermi*. Tällöin optimointion-  
gelman ratkaisu on siis

$$\tilde{x}_{est} = \arg \min_x J(x).$$

Todistakaamme seuraavaksi, että lauseen 3.0.1 ratkaisu  $x_{est}$  ja  $\tilde{x}_{est}$  yhty-  
vät. Samoilla oletuksilla kuin ensimmäisessä lauseessa 3.0.1 pätee tulos:

**Lause 3.0.2.** Lineaarinen pienimmän varianssin harhaton estimaatti on ekvivalentti optimointiongelma (3.11) saadun ratkaisun kanssa eli kun  $G$  on kuten kaavassa (3.3), niin

$$\arg \min_x J(x) = x_a + G(y - \mathcal{H}(x_a)). \quad (3.12)$$

Jos lisäksi priori- ja havaintovirheiden tiheysfunktiot ovat gaussisia, niin silloin  $x_{est}$  on myös vektorin  $x_{true}$  MAP-estimaatti.

*Todistus.* Koska  $\tilde{x}_{est}$  on optimointiongelman ratkaisu, niin sakkofunktion gradientti kohdassa  $\tilde{x}_{est}$  on nolla. Derivoimalla saamme

$$0 = \nabla J(\tilde{x}_{est}) = 2B^{-1}(\tilde{x}_{est} - x_a) - 2H^T R^{-1}(y - \mathcal{H}(\tilde{x}_{est})).$$

Jakamalla puolittain luvulla 2 ja käyttämällä hyväksi kaavaa (3.1) saamme

$$0 = B^{-1}(\tilde{x}_{est} - x_a) - H^T R^{-1}(y - \mathcal{H}(x_a) - H^T R^{-1} H(\tilde{x}_{est} - x_a))$$

ja edelleen

$$\tilde{x}_{est} = x_a + (B^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1}(y - \mathcal{H}(x_a)).$$

Seuraava identiteetti pätee

$$H^T R^{-1}(HBH^T + R) = HR^{-1}HBH^T + H^T \quad (3.13)$$

$$= (B^{-1} + HR^{-1}H)BH^T, \quad (3.14)$$

joten kertomalla  $(B^{-1} + HR^{-1}H)^{-1}$  vasemmalta ja  $(HBH^T + R)^{-1}$  oikealta saamme

$$(B^{-1} + HR^{-1}H)^{-1} H^T R^{-1} = BH^T (HBH^T + R)^{-1}.$$

Sijoittamalla tämän yhtälöön (3.14) saamme

$$\tilde{x}_{est} = x_a + BH^T (HBH^T + R)^{-1}(y - \mathcal{H}(x_a)). \quad (3.15)$$

Huomioimalla nyt  $G$ :n määritelmän saamme

$$\tilde{x}_{est} = x_a + G(y - \mathcal{H}(x_a)) = x_{est},$$

mikä on haluttu yhtäsuuruus.

Se, että gaussisessa tapauksessa ratkaisu on myös MAP-estimaatti on selvää, sillä tällöin sakkofunktio ja uskottavuusfunktion logaritmin negatio lisättynä vakiolla määritellään vakiota vaille samalla kaavalla.  $\square$

## 3.1 Kalmanin suotimet

Tässä kappaleessa tarkastelemme eri Kalmanin suodatustekniikoita<sup>2</sup>. Yleinen asettelu ongelmalle diskreetteinä ajanhetkinä  $t_i = t_0, t_1, \dots$  annetaan parina

$$\begin{cases} x_{t_i} &= \mathcal{M}_{t_i}(x_{t_{i-1}}) + \xi_{t_i} \\ y_{t_i} &= \mathcal{H}_{t_i}(x_{t_i}) + \varepsilon_{t_i}, \end{cases} \quad (3.16)$$

missä  $x_{t_i}$  on tila ajanhetkellä  $t_i$ ,  $\mathcal{M}_{t_i}$  on tunnettu *systeemioperaattori*, joka vie vektorin  $x_{t_{i-1}}$  vektorille  $x_{t_i}$  ja  $\xi_{t_i}$  on mallista johtuva prosessointivirhe, joka johtuu siitä, että malli ei kuvaa ilmiötä täydellisesti. Toisessa yhtälössä olevat termit ovat samat kuin edellisessä kappaleessa ja siinä kuvataan havaintoja  $y_{t_i}$  tilasta  $x_{t_i}$ , poikkeuksena kuitenkin, että nyt niiden ajatella riippuvan myös ajanhetkestä  $t_i$ . Virheiden  $\xi_{t_i}$  ja  $\varepsilon_{t_i}$  kovarianssimatriisit oletetaan tunnetuiksi ja niitä merkitään symboleilla  $C_{\xi, t_i}$  ja  $C_{\varepsilon, t_i}$ . Ongelmana on löytää  $x_{t_i}$ :n paras estimaatti, kun havainnot  $y_{t_1}, \dots, y_{t_i}$  on saatu.

### 3.1.1 Tavallinen lineaarinen Kalmanin suodin

Tavallisessa lineaarisessa Kalmanin suodatuksessa (eng. basic linear Kalman Filter, KF) lähdetään liikkeelle ongelman (3.16) lineaarisesta versiosta

$$\begin{cases} x_{t_i} &= M_{t_i}x_{t_{i-1}} + \xi_{t_i} \\ y_{t_i} &= H_{t_i}x_{t_i} + \varepsilon_{t_i}, \end{cases} \quad (3.17)$$

missä  $M_{t_i}$  ja  $H_{t_i}$  ovat ajasta riippuvia tunnettuja matriiseja. Haluamme käyttää hyväksemme lausetta 3.0.1. Käytetään prioritietona edellisestä vaiheesta saatua estimaattia, jota on siirretty eteenpäin mallilla  $M_{t_i}$ . Nyt siis

$$x_{a, t_i} = M_{t_i}x_{est, t_{i-1}}. \quad (3.18)$$

Olettamalla, että  $x_{est, t_{i-1}}$  ja  $\xi_{t_i}$  ovat riippumattomia, niin priorivirheen kovarianssimatriisi  $C_{a, t_i} = \text{cov}(x_{a, t_i})$  voidaan laskea seuraavasti:

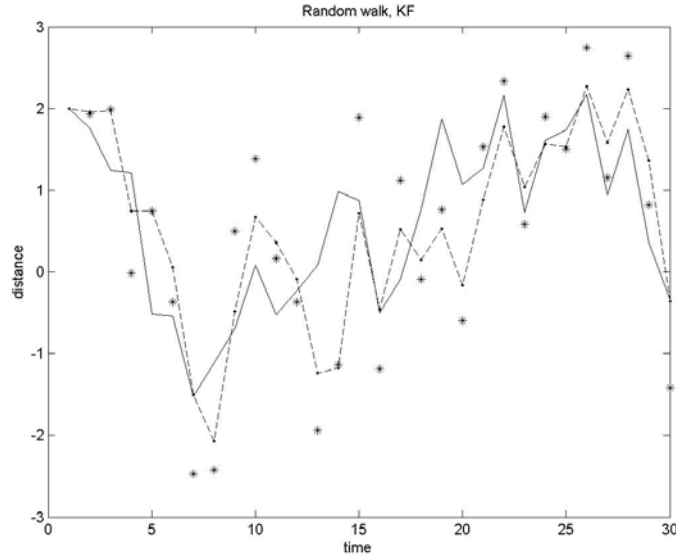
$$C_{a, t_i} = \text{cov}(M_{t_i}x_{est, t_{i-1}} + \xi_{t_i}) = M_{t_i}C_{est, t_{i-1}}M_{t_i}^T + C_{\xi, t_i}. \quad (3.19)$$

Sijoittamalla nämä lauseen 3.0.1 kaavoihin (3.2)–(3.3) saamme

$$x_{est, t_i} = x_{a, t_i} + G_{t_i}(y_{t_i} - H_{t_i}x_{a, t_i}), \quad (3.20)$$

---

<sup>2</sup>Termi Kalmanin suodin ei ole täysin vakiintunut, vaan saatetaan esimerkiksi kirjoittaa Kalman-suodin. Lisäksi suodin, voi olla esimerkiksi suodatin tai filteri.



Kuva 3.1: Simuloitu satunaiskulku. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä ja KF:stä saatua estimaattia katkoviivalla.

missä

$$G_{t_i} = C_{a,t_i} H_{t_i}^T (H C_{a,t_i} H_{t_i}^T + C_{\varepsilon_{t_i}})^{-1}. \quad (3.21)$$

Samoin  $x_{est,t}$ :n kovarianssimatriisi on

$$C_{est,t_i} = (I - G_{t_i} H_{t_i}) C_{a,t_i}. \quad (3.22)$$

Yhtälöt (3.18)–(3.22) muodostavat rekursiivisen Kalmanin suodinmallin, jossa matriisia  $G_{t_i}$  kutsutaan *Kalmanin vahvistusmatriisiksi* (eng. Kalman gain matrix). Nyt voimme luoda seuraavan Kalmanin suodinalgoritmin:

**Askel 0:** Valitse alkuarvaus  $x_{est,t_0}$ , kovarianssimatriisi  $C_{est,t_0}$  ja aseta  $i = 1$ .

**Askel 1:** Laske  $x_{a,t_i} = M_{t_i} x_{est,t_{i-1}}$  ja  $C_{a,t_i} = M_{t_i} C_{est,t_{i-1}} M_{t_i}^T + C_{\xi_{t_i}}$ .

**Askel 2:** Laske  $G_{t_i} = C_{a,t_i} H_{t_i}^T (H C_{a,t_i} H_{t_i}^T + C_{\varepsilon_{t_i}})^{-1}$ ,  $x_{est,t_i} = x_{a,t_i} + G_{t_i} (y_{t_i} - H_{t_i} x_{a,t_i})$  ja  $C_{est,t_i} = (I - G_{t_i} H_{t_i}) C_{a,t_i}$ .

**Askel 3:** Päivitä  $i = i + 1$  ja palaa askeleeseen 1.

Tarkastellaan Kalmanin suodimen toimintaa yksinkertaisen esimerkin valossa.

**Esimerkki 3.1.1** (Satunnaiskulku). *Tarkastellaan reaaliarvoista prosessia  $x_{t_i}$  ja havaintoa  $y_{t_i}$ , jotka saadaan seuraavista yhtälöistä*

$$\begin{cases} x_{t_i} &= x_{t_{i-1}} + \xi_{t_i} \\ y_{t_i} &= x_{t_i} + \varepsilon_{t_i}, \end{cases} \quad (3.23)$$

missä  $\xi_{t_i}$ :n varianssi on  $\sigma_\xi^2$  ja  $\varepsilon_{t_i}$ :n varianssi on  $\sigma_\varepsilon^2$ .

Käyttämällä yhtälöparin (3.17) notaatiota havaitsemme, että  $M_{t_i} = H_{t_i} = 1$  kaikilla  $t_i$ . Kaavoista (3.18)–(3.19) saamme

$$\begin{aligned} x_{a,t_i} &= x_{est,t_{i-1}} \\ \sigma_{a,t_i}^2 &= \sigma_{est,t_{i-1}}^2 + \sigma_\xi^2. \end{aligned}$$

Kalmanin vahvistusmatriisiksi saamme

$$G_{t_i} = \frac{\sigma_{est,t_{i-1}}^2 + \sigma_\xi^2}{\sigma_{est,t_{i-1}}^2 + \sigma_\xi^2 + \sigma_\varepsilon^2}$$

ja estimaatiksi

$$x_{est,t_i} = x_{a,t_i} + G_{t_i}(y_{t_i} - x_{a,t_i}).$$

Päivitettävä varianssia saa muodon

$$\sigma_{est,t_i}^2 = \sigma_{a,t_i}^2 - G_{t_i}\sigma_{a,t_i}^2.$$

Simuloimme satunnaiskulun alkuarvosta  $x_{t_0} = 2$ . Variansseille käytimme arvoja  $\sigma_\xi^2 = 1$  ja  $\sigma_\varepsilon^2 = 1$ . Kalmanin suodinta varten käyttämämme alkuarvot olivat  $\sigma_{est,t_0}^2 = 1$  ja  $x_{est,t_0} = 2$ . Simuloinnin tulos on esitetty kuvassa 3.1.

### 3.1.2 Kalmanin silotin

Kiinteän viiveen Kalmanin silotin (eng. Fixed-Lag Kalman Smoother, FLKS) on takautuva data-assimilaatio menetelmä, jonka tarkoituksena on silottaa Kalmanin suotimesta saatuja tuloksia. Ideana on, että sekä analyysihetken ennen että jälkeen tehtyjä havaintoja voidaan käyttää hyödyksi. Esittelemme menetelmän ainoastaan silottaaksemme tavallisesta lineaarisesta Kalmanin suotimesta saatuja estimaatteja. Menetelmän pystyy myös yleistämään muihin Kalmanin suotimiin. Yleisesti Kalmanin silotin parantaa analyysin laatua, mutta hintana on sen soveltumattomuus reaaliaikaiseen data-assimilaatioon. Menetelmä on kuitenkin käyttökelpoinen niin sanotussa offline-tilanteessa.

Tarkastelkaamme kiinteitä ajanhetkiä  $t_i = t_0, \dots, t_N$  ja käyttäkäämme kiinteästä viiveestämme merkintää  $L$ . Viive voi olla enintään  $N$ . Kalmanin silottimesta saatua estimaattia hetkellä  $t_{i-l}$  merkitsemme  $x_{t_{i-l}|t_i}^{est}$  kun analyysi on suoritettu ajanhetkeen  $t_i$  asti. Nyt voimme esittää Kalmanin silottimen yhtälön seuraavasti

$$x_{t_{i-l}|t_i}^{est} = x_{t_{i-l}|t_{i-1}}^{est} + G_{t_{i-l}|t_i}(y_{t_i} - H_{t_i}x_{a,t_i}), \quad (3.24)$$

missä  $G_{t_{i-l}|t_i}$  on hetkeä  $t_{i-l}$  vastaava vahvistusmatriisi kun analyysi on suoritettu hetkeen  $t_i$  asti.

Yhtälö 3.24 on käyttökelpoinen vain kun  $l \leq i$ . Kun  $L \leq i$ , niin  $l = 1, \dots, i$  ja muulloin  $l = 1, \dots, L$ . Vahvistusmatriisi  $G_{t_{i-l}|t_i}$  saadaan seuraavasti

$$G_{t_{i-l}|t_i} = C_{t_i, t_{i-l}|t_{i-1}}^{a, est} H_{t_i}^T (H_{t_i} C_{a, t_i} H_{t_i}^T + C_{\varepsilon, t_i})^{-1}. \quad (3.25)$$

Yhtälössä oleva ristikovarianssimatriisi  $C_{t_i, t_{i-l}|t_{i-1}}^{a, est}$  saadaan seuraavista yhtälöistä

$$C_{t_i, t_{i-l}|t_{i-1}}^{a, est} = M_{t_i} C_{t_{i-1}, t_{i-l}|t_{i-1}}^{est, est} \quad (3.26)$$

$$C_{t_i, t_{i-l}|t_i}^{est, est} = (I - G_{t_i} H_{t_i}) C_{t_i, t_{i-l}|t_{i-1}}^{a, est}. \quad (3.27)$$

Silottimen estimaatin kovarianssimatriisi saadaan kaavasta

$$C_{t_{i-l}|t_i}^{est} = C_{t_{i-l}, t_{i-1}}^{est} - G_{t_{i-l}|t_i} H_{t_i} C_{t_i, t_{i-l}|t_{i-1}}^{a, est}. \quad (3.28)$$

**Huomautus.** Kun  $l = 1$ , niin

$$C_{t_{i-1}, t_{i-1}|t_{i-1}}^{est, est} = C_{est, t_{i-1}}$$

ja

$$C_{t_{i-1}, t_{i-1}}^{est} = C_{est, t_{i-1}}.$$

Nyt voimme koota yhtälöt (3.24)-(3.28) seuraavaksi Kalmanin silottinalgoritmiksi:

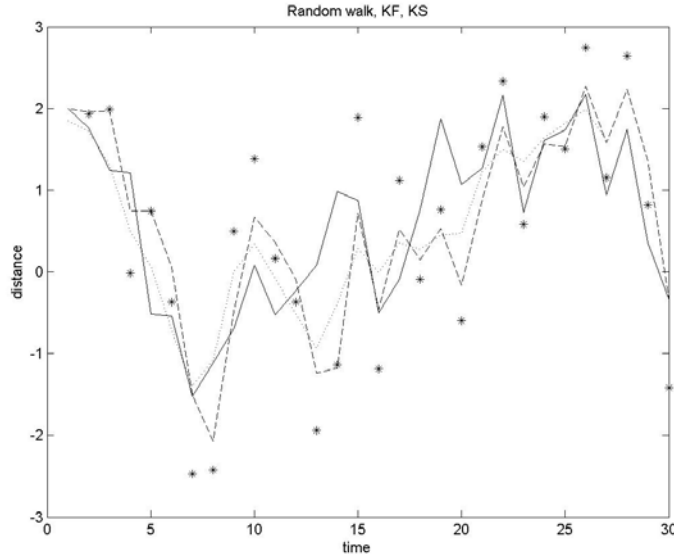
**Askel 0:** Määritä  $L$  ja aseta  $i = 1$ .

**Askel 1:** Aseta  $l = 1$ .

**Askel 2:** Laske  $C_{t_i, t_{i-l}|t_{i-1}}^{a, est} = M_{t_i} C_{t_{i-1}, t_{i-l}|t_{i-1}}^{est, est}$ .

**Askel 3:** Laske  $G_{t_{i-l}|t_i}$  kuten yhtälössä (3.25) ja  $x_{t_{i-l}|t_i}^{est}$  kuten yhtälössä (3.24).

**Askel 4:** Laske  $C_{t_i, t_{i-l}|t_{i-1}}^{a, est} = M_{t_i} C_{t_{i-1}, t_{i-l}|t_{i-1}}^{est, est}$  ja  $C_{t_i, t_{i-l}|t_i}^{est, est} = (I - G_{t_i} H_{t_i}) C_{t_i, t_{i-l}|t_{i-1}}^{a, est}$ .



Kuva 3.2: Simuloitu satunaiskulku. Muuten sama kuin kuva 3.1, mutta Kalmanin silottimesta saatua estimaattia on merkitty pisteviivalla. Kiinteänä viiveenä on käytetty arvoa kolme.

**Askel 5:** Laske  $C_{t_{i-l}|t_i}^{est} = C_{t_{i-l},t_{i-1}}^{est} - G_{t_{i-l}|t_i} H_{t_i} C_{t_i,t_{i-l}|t_{i-1}}^{a,est}$ .

**Askel 6:** Päivitä  $l = l + 1$  ja palaa askeleeseen 2 niin kauan kunnes  $l < \min(L, i)$ .

**Askel 7:** Päivitä  $i = i + 1$  ja palaa askeleeseen 1.

Simuloidaan silottimen toimintaa esimerkin 3.1.1 avulla. Simuloinnin tulos viiveen arvolla kolme on esitetty kuvassa 3.2. Havaitsemme, että silottimesta saatava estimaatti on parempi kuin Kalmanin suotimesta saatu.

Kalmanin silotinta on käsitelty tarkemmin artikkeleissa [21] ja yhtälöt on johdettu tarkasti artikkelissa [5]. Periaatteessa yhtälöt voidaan johtaa kuten lauseessa 3.0.1.

### 3.1.3 Laajennettu Kalmanin suodin

Laajennetussa Kalmanin suodatuksessa (eng. Extended Kalman Filter, EKF) on tarkoitus ratkaista yleinen suodatusongelma (3.16). Kun systeemi- ja/tai havainto-operaattori eivät ole lineaarisia, niin tavallista lineaarista Kalmanin suodinmallia ei enää voida suoraan käyttää. Tällöin voimme kuitenkin

Jacobin matriisin avulla tehdä seuraavat linearisoidut approksimaatiot

$$\begin{cases} M_{t_i} &= \frac{\partial \mathcal{M}_{t_i}(x_{est,t_{i-1}})}{\partial x}, \\ H_{t_i} &= \frac{\partial \mathcal{H}_{t_i}(x_{a,t_i})}{\partial x}, \end{cases} \quad (3.29)$$

joita tarvitaan kovarianssien laskemiseen. Derivointi suoritetaan siis muuttujan  $x$  suhteen pisteessä  $x_{est,t_{i-1}}$  ja malli evaluoidaan aikapisteessä  $t_i$ . Toisella laisella notaatiolla tämän voi kirjoittaa seuraavasti:

$$M_{t_i} = \left. \frac{\partial \mathcal{M}_{t_i}(x)}{\partial x} \right|_{x=x_{est,t_{i-1}}}. \quad (3.30)$$

Triviaalisti on syytä huomauttaa, että yhtälöissä (3.29) olevaa derivointia ei suoriteta ajan suhteen, mutta mikäli malli annetaan esimerkiksi differentiaaliyhtälönä ajan suhteen, niin differentiaaliyhtälön ratkaisijassa käytetty aika-askel vaikuttaa approksimaatioon. Havainnollistetaan yhtälöä (3.30) esimerkin avulla.

**Esimerkki 3.1.2.** *Tarkastellaan hetkellä  $t_i$  epälineaarista mallia*

$$\mathcal{M}_{t_i}(x) = (x_1^2, x_1 x_2)^T$$

*ja oletetaan, että edellisessä vaiheessa on saatu estimaatiksi  $x_{est,t_{i-1}} = (3, 5)^T$ .*

Laskemalla Jacobin matriisin saamme

$$M_{t_i} = \begin{pmatrix} 2x_{est,t_{i-1},1} & 0 \\ x_{est,t_{i-1},1} & x_{est,t_{i-1},2} \end{pmatrix} = \begin{pmatrix} 6 & 0 \\ 3 & 5 \end{pmatrix}.$$

Kun approksimaatiot (3.29) on suoritettu voimme ratkaista ongelmanme kuten tavallisen lineaarisen Kalmanin suotimen tapauksessa seuraavin yhtälöin

$$x_{a,t_i} = \mathcal{M}_{t_i}(x_{est,t_{i-1}}), \quad (3.31)$$

$$C_{a,t_i} = M_{t_i} C_{est,t_1} M_{t_i}^T + C_{\xi,t_i}, \quad (3.32)$$

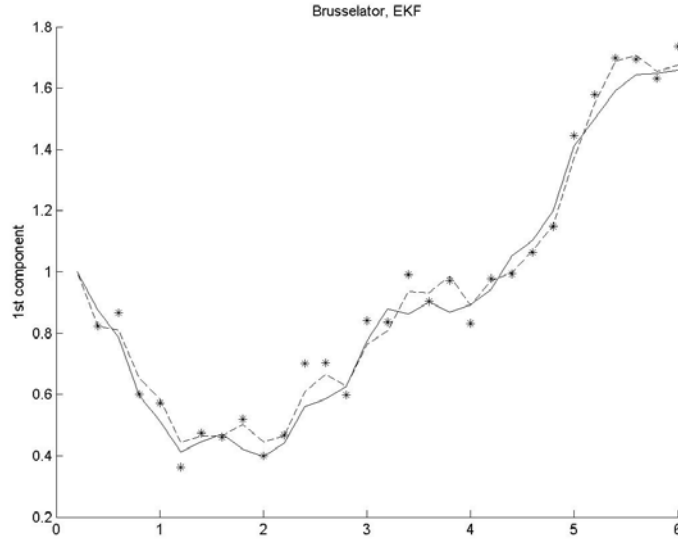
$$x_{est,t_i} = x_{a,t_i} + G_{t_i}(y_{t_i} - \mathcal{H}_{t_i}(x_{a,t_i})), \quad (3.33)$$

$$G_{t_i} = C_{a,t_i} H_{t_i}^T (H C_{a,t_i} H_{t_i}^T + C_{\varepsilon,t_i})^{-1}, \quad (3.34)$$

$$C_{est,t_i} = (I - G_{t_i} H_{t_i}) C_{a,t_i}, \quad (3.35)$$

missä kaikki symbolit ovat määritelty kuten lineaarisen Kalmanin suotimen tapauksessa. Kuitenkin on syytä huomata, että yhtälössä (3.31)  $x_{a,t_i}$  on saatu käyttämällä epälineaarista mallia  $\mathcal{M}_{t_i}$ , eikä sen linearisoitua versiota  $M_{t_i}$ . Nyt voimme luoda seuraavan laajennetun Kalmanin suodinalgoritmin:





Kuva 3.3: Brusselatorin ensimmäinen komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä ja EKF:stä saatua estimaattia katkoviivalla.

**Askel 0:** Valitse alkuarvaus  $x_{est,t_0}$ , kovarianssimatriisi  $C_{est,t_0}$  ja aseta  $i = 1$ .

**Askel 1:** Laske  $x_{a,t_i} = \mathcal{M}_{t_i}(x_{est,t_{i-1}})$

**Askel 2:** Laske  $M_{t_i}$  ja  $H_{t_i}$  kuten kaavassa (3.29).

**Askel 3:** Laske  $C_{a,t_i} = M_{t_i}C_{est,t_{i-1}}M_{t_i}^T + C_{\xi,t_i}$ .

**Askel 4:** Laske  $G_{t_i} = C_{a,t_i}H_{t_i}^T(HC_{a,t_i}H_{t_i}^T + C_{\varepsilon,t_i})^{-1}$ ,  $x_{est,t_i} = x_{a,t_i} + G_{t_i}(y_{t_i} - \mathcal{H}_{t_i}(x_{a,t_i}))$  ja  $C_{est,t_i} = (I - G_{t_i}H_{t_i})C_{a,t_i}$ .

**Askel 5:** Päivitä  $i = i + 1$  ja palaa Askeleeseen 1.

Askelessa 2  $M_{t_i}$  ja  $H_{t_i}$  voidaan laskea tai estimoida usealla eri tavalla esimerkiksi numeerisesti differenssien avulla. Tämä voi olla laskennallisesti erittäin kallista, sillä muodostamiseen tarvittavia mallievaluointeja on  $n$  kappaletta.

Tarkastellaan laajennetun Kalmanin suotimen toimintaa niin kutsuttujen Brusselator-yhtälöiden avulla, jotka mallintavat autokatalyyttisiä reaktioita. Tulemme käyttämään tätä esimerkkiä jatkossakin.

**Esimerkki 3.1.3** (Brusselator). *Tarkastellaan seuraavaa differentiaaliyhtälöryhmää*

$$\begin{cases} u' &= 1 + au^2v - (b+1)u \\ v' &= bu - au^2v, \end{cases} \quad (3.36)$$

missä  $a$  ja  $b$  ovat vakioita. Alkuarvoina ovat  $u(0) = u_0$  ja  $v(0) = v_0$ .

Aikaisemman käyttämämme notaation valossa  $u = x_1$  ja  $v = x_2$ . Nyt voimme kirjoittaa mallimme seuraavasti

$$\mathcal{M}_{t_i}(u_{t_{i-1}}, v_{t_{i-1}}) = (u_{t_i}, v_{t_i}). \quad (3.37)$$

Laskemme simulaatiota varten todellisen tilan Matlabin differentiaaliyhtälöratkaisijalla, joka käyttää Runge-Kutta-menetelmää, simuloimme tähän virhettä ja simuloimme tästä havainnon. Alkupisteenä käytämme  $(u_{t_0}, v_{t_0}) = (1, 1)$ . Havainto-operaattorina käytämme identtistä matriisiä eli havainnot osuvat samoihin hilapisteisiin kuin tilamuuttujat. Käytämme vakiodien arvoja  $a = 1$  ja  $b = 2$  ja havainnon aika-askeleeksi otimme  $\Delta t = 0,2$ . Kovarianssimatriisit oletimme kaikilla  $t_i$  seuraaviksi

$$C_{\xi, t_i} = C_{\varepsilon, t_i} = 0,005I.$$

EKF:n alkuarvauksena käytimme oikeita arvoja eli  $(u_{est, t_0}, v_{est, t_0}) = (1, 1)$  ja tälle oletimme kovarianssimatriisiksi  $C_{est, t_0} = I$ . Mallin linearisoinnin laskimme differenssien avulla. Kuvassa 3.3 on havainnollistettu simuloinnin tulos ensimmäisessä komponentissa.

Seuraavaksi esitämme joitakin huomioita edellä käytettyihin Kalmanin suotimiin. Siinä missä tavallinen KF on lauseen 3.0.1 nojalla optimaalinen, niin sama ei välttämättä päde laajennetun Kalmanin suotimen osalta. Tämä johtuu siitä, että mallin ollessa hyvin epälineaarinen ei algoritmin kolmannessa askeleessa käytetty kovarianssimatriisi ole välttämättä oikea. Usein kun systeemin koko on pieni ja aika-askel riittävän lyhyt ovat käytetyt approksimaatiot riittävän hyviä.

Mikäli systeemin koko on iso syntyy approksimaatioiden tarve. Erityisesti tämä koskee vahvistusmatriisin laskemista, ja siinä tarvittavan käänteismatriisin muodostamista. Myös kovarianssimatriisin koko kasvaa nopeasti liian suureksi, eikä matriiseja välttämättä pystytä edes tallentamaan muistiin. Lisäksi mikäli matriisin  $M$  laskenta suoritetaan käyttäen differenssejä, niin laskennasta tulee huomattavan hidasta. Nämä seikat johtavat siihen, että perinteisestä Kalmanin suodatuksesta tulee ongelman koon kasvaessa toteuttamiskelvoton. Seuraavaksi esittelemme muutaman eri vaihtoehdon approksimoida perinteistä Kalmanin suodatusta.

### 3.1.4 Ensemble Kalmanin suodin

Ensemble Kalmanin suodatus (eng. Ensemble Kalman Filter, EnKF) lähtee samoista ongelmanasettelusta (3.16), kuin edellisetkin Kalmanin suotimet, mutta siinä on tarkoitus välttää laajennetussa Kalmanin suodatuksessa tarvittavaa mallin linearisointia. Perusideana on luoda mallitilojen joukko eli niin sanottu ensemble  $X_{est} = (x^1, \dots, x^N)^T$ , jota viedään mallilla eteenpäin, ja jonka jälkeen siitä voidaan laskea otoskeskiarvo ja otoskovarianssi. Näitä voidaan käyttää prioritietona tuotettaessa analyysia.

Suodatus lähtee käyntiin luomalla alkuensemble  $X_{est,t_0}$ . Sen luomiseen on eri vaihtoehtoja, mutta kun alkuarvauksen  $x_{est,t_0}$  jakauma ja kovarianssimatriisi  $C_{est,t_0}$  oletetaan tunnetuiksi, niin alkuensemblen voi tuottaa esimerkiksi satunnaislukugeneraattorilla. Seuraavaksi ensemble viedään mallilla seuraavaan ajanhetkeen ja siihen lisätään satunnaislukugeneraattorilla tunnettua mallivirhettä  $\xi_{t_i}$  vastaavaa häiriötä, jolloin saamme

$$x_{a,t_i}^k = \mathcal{M}_{t_i}(x_{est,t_{i-1}}^k) + \xi_t^k. \quad (3.38)$$

Laskemalla tästä siirretystä otoksesta keskiarvon ja kovarianssimatriisin saamme tarvittavat priori-informaatiot

$$x_{a,t_i} = \frac{1}{N} \sum_{k=1}^N x_{a,t_i}^k, \quad (3.39)$$

$$(C_{a,t_i})_{lj} = \frac{1}{N-1} \sum_{k=1}^N (x_{l,t_i}^k - \overline{x_{l,t_i}})(x_{j,t_i}^k - \overline{x_{j,t_i}}). \quad (3.40)$$

Nyt voimme laskea vahvistusmatriisin kuten aiemminkin

$$G_{t_i} = C_{a,t_i} H_{t_i}^T (H C_{a,t_i} H_{t_i}^T + C_{\varepsilon,t_i})^{-1} \quad (3.41)$$

ja sen avulla voimme päivittää ensemblen, kun havainnot  $y_{t_i}$  on tehty:

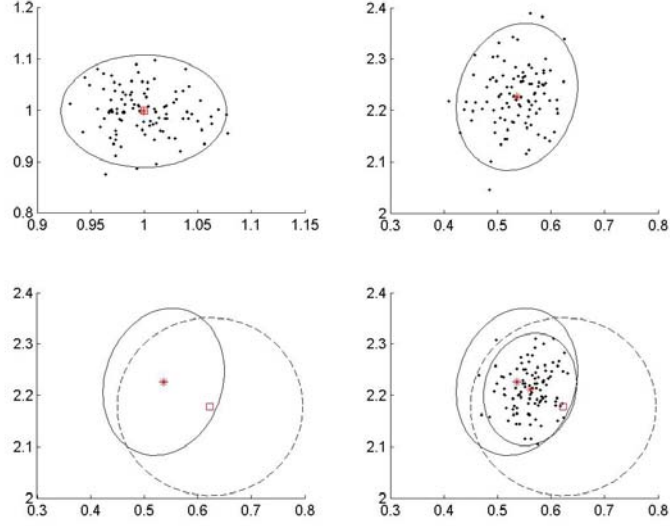
$$x_{est,t_i}^k = x_{a,t_i}^k + G_{t_i}((y_{t_i} - \mathcal{H}_{t_i}(x_{a,t_i}^k))). \quad (3.42)$$

Laskemalla tästä keskiarvon ja kovarianssin saamme estimaatit

$$x_{est,t_i} = \frac{1}{N} \sum_{k=1}^N x_{est,t_i}^k, \quad (3.43)$$

$$(C_{est,t_i})_{lj} = \frac{1}{N-1} \sum_{k=1}^N (x_{est,l,t_i}^k - \overline{x_{est,l,t_i}})(x_{est,j,t_i}^k - \overline{x_{est,j,t_i}}). \quad (3.44)$$

Nyt voimme koota yhtälöt (3.38)-(3.44) yhteen ensemble Kalmanin suodinalgoritmiiksi:



Kuva 3.4: EnKF:n toiminta. Ensimmäisessä kuvassa on simuloitu alkuensemble ja 95 prosentin luottamusalue. Ensemblen keskiarvoa on merkitty tähdellä ja alkuarvausta neliöllä. Toisessa kuvassa ensemble on viety mallilla eteenpäin ja pisteistä on laskettu keskiarvo ja 95 prosentin luottamusalue. Kolmannessa kuvassa on piirretty ensemblen keskiarvo (tähti) ja 95 prosentin luottamusalue, sekä tehty havainto (neliö) ja piirretty tämän luottamusalue. Neljännessä kuvassa on piirretty lisäksi päivitetty ensemble ja tästä laskettu keskiarvo ja luottamusalue. Mallina on käytetty Brusselator-yhtälöitä.

**Askel 0:** Luo alkuensemble  $X_{est,t_0}$  ja aseta  $i = 1$ .

**Askel 1:** Vie ensemble eteenpäin mallilla ja häiriöitä  $x_{a,t_i}^k = \mathcal{M}_{t_i}(x_{est,t_{i-1}}^k) + \xi_{t_i}^k$ .

**Askel 2:** Laske  $H_{t_i}$  kuten kaavassa (3.29).

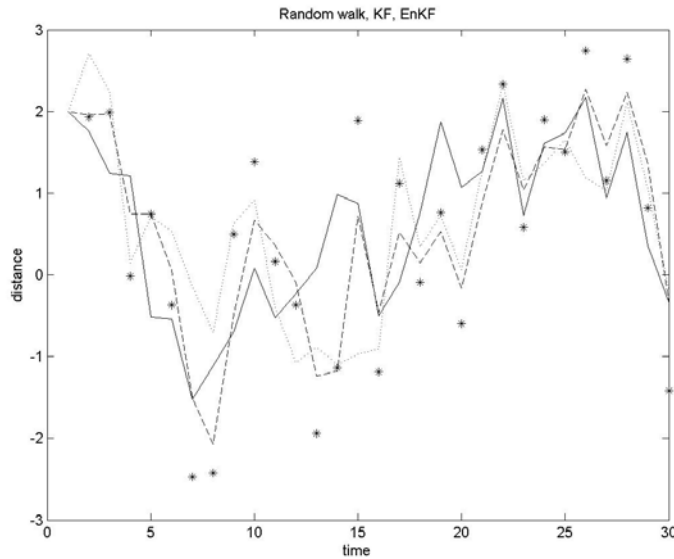
**Askel 3:** Laske  $x_{a,t_i} = \text{mean}(X_{a,t_i})$  ja  $C_{a,t_i} = \text{cov}(X_{a,t_i})$  ja  $G_{t_i}$ .

**Askel 4:** Päivitä ensemble  $x_{est,t_i}^k = x_{a,t_i}^k + G_{t_i}((y_{t_i} - \mathcal{H}_{t_i}(x_{a,t_i}^k) + \epsilon_{t_i}^k)$ .

**Askel 5:** Laske  $x_{est,t_i} = \text{mean}(X_{est,t_i})$  ja  $C_{est,t_i} = \text{cov}(X_{est,t_i})$ .

**Askel 6:** Päivitä  $i = i + 1$  ja palaa Askeleeseen 1.

Algoritmin vaiheita on esitelty kuvassa 3.4. Havainnollistetaan vielä ensemble Kalmanin suotimen toimintaa jo käytetyn esimerkin 3.1.1 valossa.



Kuva 3.5: Simuloitu satunaiskulku. Muuten sama kuin kuva 3.1, mutta EnKF:stä saatua estimaattia on merkitty pisteviivalla.

Simuloinnissa on ensemblen kokona käytetty kymmentä. Simuloinnin tulos on esitetty kuvassa 3.5. Havaitaan, että EnKF:n antama estimaatti on lähes sama, kuin KF:stäkin saatu.

On selvää, että alkuensemble pitäisi ideaalisesti esittää alkuarvauksen  $x_{est,t_0}$  virheiden tilastolliset ominaisuudet hyvin. Vaatimattomat virhetulkinat alkuensemblen kanssa eivät kuitenkaan vaikuta tuloksiin merkittävästi ajan kuluessa. Nyrkkisääntönä voitaisiin pitää, että alkuensemblen voi luoda lisäten *jonkinlaista* häiriötä parhaaseen arvaukseen ja sen jälkeen viedä ensemblea eteenpäin mallilla, jonka jälkeen siihen lisätään tunnettua mallivirhettä vastaavaa häiriötä. Tämä riittää takaamaan, että systeemi pysyy tasapainossa ja että riittävät korrelaatiot ovat syntyneet.

Ensemble Kalmanin suodatusta on myös yritetty ilman, että mallilla eteenpäin vietyä ensemblea häiriöitetään. Tämä menetelmä johtaa kuitenkin liian pieniin virhearvioihin. Kun ensemblen kokoa kasvatetaan, niin EnKF konvergoituu kohti (laajennettua) Kalmanin suodinta [7].

### 3.1.5 Paikallinen ensemble-muunnos Kalmanin suodin

Paikallinen ensemble-muunnos Kalmanin suodin (eng. Local Ensemble Transform Kalman Filter, LETKF) on ensemble Kalmanin suodatusta muistuttava menetelmä. Siinä ideana on muuntaa taustaensemble analyysiensembleksi

käyttäen painovektoreita (eng. weight vector), jotka tarjoavat 'parhaan' sovituksen ensemblehäiriöiden virittämässä aliavaruudessa.

Kirjoitetaan EnKF-mentelmässä käytetty taustavirheen kovarianssimatriisi muodossa

$$C_a = \frac{1}{N-1} X^a (X^a)^T,$$

missä  $X^a$  on  $n \times N$  -ulotteinen matriisi, jonka sarakkeina ovat erotukset  $x_a^k - \bar{x}_a$ . Tällöin matriisin  $X^a$  sarakkeiden summa on nolla. Korvataan nyt sakkofunktiossa (3.11) ollut matriisi  $B$  matriisilla  $C_a$ . Formaalisti saamme

$$J(x) = (\bar{x}_a - x)^T C_a^{-1} (\bar{x}_a - x) + (y - \mathcal{H}(x))^T R^{-1} (y - \mathcal{H}(x)).$$

Oletetaan nyt, että  $x$  saadaan lineaarikombinaationa

$$x = \bar{x}_a + X^a w,$$

missä  $w$  on painovektori. Mikäli  $w$  on nollakeskinen gaussisesti jakautunut satunnaismuuttuja kovarianssimatriisilla  $1/(N-1)I$ , niin tällöin  $x$  on gaussisesti jakautunut kovarianssimatriisilla  $C_a$  ja odotusarvolla  $\bar{x}_a$ . Tämä motivoi meitä määrittelemään seuraavan sakkofunktion

$$\tilde{J}(w) = (N-1)w^T w + (y - \mathcal{H}(\bar{x}_a + X^a w))^T R^{-1} (y - \mathcal{H}(\bar{x}_a + X^a w)). \quad (3.45)$$

Tällöin mikäli  $w$  minimoi sakkofunktion  $\tilde{J}$ , niin  $\bar{x}_{est} = \bar{x}_a + X^a w$  minimoi sakkofunktion  $J$  [10]. Mikäli  $\mathcal{H}$  on epälineaarinen, niin muodostetaan ensemble:

$$y_a^k = \mathcal{H}(x_a^k)$$

ja vastaavasti  $Y^a$  kuten  $X^a$ . Tehdään approksimaatio

$$\mathcal{H}(x_a + X^a w) \approx \bar{y}_a + Y^a w,$$

jolloin voimme kirjoittaa sakkofunktion (3.45) muodossa

$$\tilde{J}^*(w) = (N-1)w^T w + (y - \bar{y}_a + Y^a w)^T R^{-1} (y - \bar{y}_a + Y^a w).$$

Tämä sakkofunktio minimoituu<sup>3</sup> kohdassa

$$\bar{w}_{est} = \tilde{C}_{est} (Y^a)^T R^{-1} (y - \bar{y}_a),$$

missä

$$\tilde{C}_{est} = ((N-1)I + (Y^a)^T R^{-1} Y^a)^{-1},$$

---

<sup>3</sup>Havaitaan esimerkiksi derivoimalla kuten aiemminkin.

Analyysipuolella estimaatti ja sen kovarianssi saadaan tällöin kaavoista:

$$\begin{aligned}\bar{x}_{est} &= \bar{x}_a + X^a \bar{w}_{est}, \\ C_{est} &= X^a \tilde{C}_{est} (X^a)^T.\end{aligned}$$

Tarvitsemme nyt analyysiensemblen (ja sitä vastaavan matriisin  $X^{est}$ ), jonka keskiarvo on  $\bar{x}_{est}$  ja kovarianssi  $C_{est}$ . Valitaan nyt

$$X^{est} = X^a W^{est},$$

missä

$$W^{est} = ((N-1)\tilde{C}_{est})^{\frac{1}{2}}, \quad (3.46)$$

jossa neliöjuurella tarkoitamme symmetristä neliöjuurta. Tällöin

$$\tilde{C}_{est} = \frac{1}{N-1} W^{est} (W^{est})^T$$

ja edelleen

$$\begin{aligned}C_{est} &= X^a \tilde{C}_{est} (X^a)^T = X^a \frac{1}{N-1} W^{est} (W^{est})^T (X^a)^T \\ &= \frac{1}{N-1} X^{est} (X^{est})^T\end{aligned}$$

eli  $C_{est}$  on matriisin  $X^{est}$  määrämän enssemblen kovarianssimatriisi. Näytetään vielä, että matriisin  $X^{est}$  sarakkeiden summa on nolla. Tästä seuraa, että enssemblen keskiarvo on haluamamme  $\bar{x}_{est}$ .

Olkoon  $v = (1, \dots, 1)^T$ . Mikäli  $X^{est}$  sarakkeiden summa on nolla, niin

$$X^{est} v = 0.$$

Koska matriisin  $Y^a$  sarakkeiden summa on nolla, niin

$$\tilde{C}_{est}^{-1} v = ((N-1)I + (Y^a)^T R^{-1} Y^a) v = (N-1)v$$

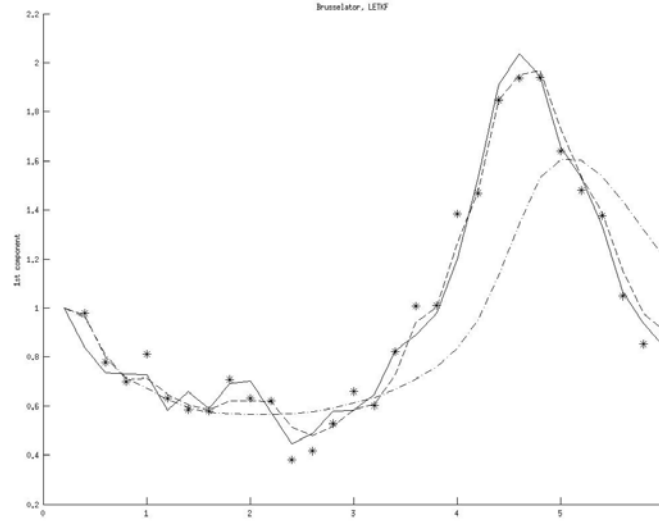
eli  $v$  on matriisin  $\tilde{C}_{est}$  ominaisvektori ominaisarvolla  $1/(N-1)$ , jolloin yhtälöstä (3.46) seuraa, että  $v$  on myös matriisin  $W^{est}$  ominaisvektori ominaisarvolla yksi. Nyt koska matriisin  $X^a$  sarakkeiden summa on nolla, niin

$$X^{est} v = X^a W^{est} v = X^a v = 0.$$

Lisäämällä matriisiin  $W^{est}$  sarakkeisiin vektorin  $\bar{w}_{est}$  voimme luoda analyysiensemblen seuraavasti

$$x_{est}^k = \bar{x}_a + X^a w_{est}^i.$$

Kootkaamme LETKF-menetelmä algoritmiksi:



Kuva 3.6: Brusselatorin ensimmäinen komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä, LETKF:stä saatua estimaattia pisteviivalla ja LETKF-estimaattia, joka on saatu kovarianssikorjauksen avulla ( $r = 1$ ) katkoviivalla.

**Askel 0:** Luo alkuensemble ja aseta  $i = 1$ .

**Askel 1:** Vie ensemble eteenpäin mallilla, laske siitä keskiarvo  $\bar{x}_{a,t_i}$  ja muodosta matriisi  $X_{t_i}^a$  ja laske ensemble  $y_{a,t_i}^k$  ja matriisi  $Y_{t_i}^a$ .

**Askel 2:** Laske kovarianssimatriisi  $\tilde{C}_{est,t_i}$  ja painovektori  $\bar{w}_{est,t_i}$ .

**Askel 3:** Laske estimaatti  $\bar{x}_{est,t_i} = \bar{x}_{a,t_i} + X_{t_i}^{a,t_i} \bar{w}_{est,t_i}$  ja sen kovarianssimatriisi  $C_{est,t_i} = X_{t_i}^a \tilde{C}_{est,t_i} (X_{t_i}^a)^T$ .

**Askel 4:** Laske matriisi  $W_{t_i}^{est}$  ja muodosta ensemble  $w_{est,t_i}^k$  lisäämällä matriisiin  $W_{t_i}^{est}$  sarakkeisiin vektori  $\bar{w}_{est,t_i}$ .

**Askel 5:** Päivitä analyysi-ensemble  $x_{est,t_i}^k = \bar{x}_{a,t_i} + X_{t_i}^a w_{est,t_i}^k$ .

**Askel 6:** Päivitä  $i = i + 1$  ja palaa Askeleeseen 1.

**Kovarianssi-inflaatio.** Eräs LETKF-menetelmän heikkouksista on, että menetelmässä – toisin kuin muissa Kalmanin suotimissa – ei käytetä lainkaan mallivirhettä. Tämä voi käytännössä johtaa liian pieniin virhearvioihin.



Virhearvioiden pienuus taas johtaa siihen, että analyysissä ruvetaan painotamaan (mahdollisesti virheellistä) taustatilaa havaintojen kustannuksella. Tätä voi yrittää korjata kasvattamalla keinotekoisesti kovariansseja assimi-laatiosykliissä. Eräs käytetty korjaus on

$$\tilde{C}_{est} = \left( \frac{N-1}{r} I + (Y^a)^T R^{-1} Y^a \right)^{-1},$$

missä  $r > 0$  on vapaasti valittava viritysparametri [10].

Tarkastellaan korjauksen vaikutusta jo esitetyn Brusselator-esimerkin avulla. Käytetyt parametrit ovat samat kuin aikaisemminkin. Kuvassa 3.6 on esitetty ajon tulos ilman korjausta ja korjauksen kanssa ( $r = 1$ ). Havaitsemme, että korjauksella on merkittävä vaikutus.

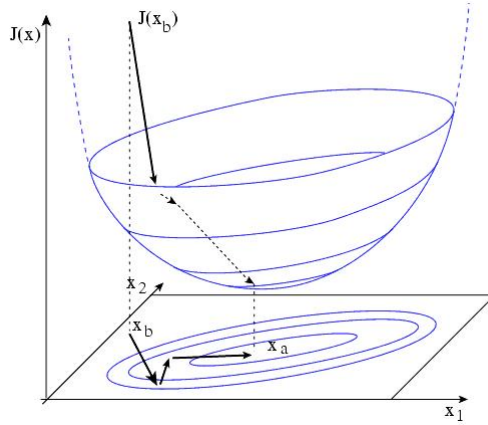
LETKF-menetelmää ja sen paikallista käyttöä on tarkasteltu tarkemmin artikkeleissa [10] [8]. Lisäksi siitä on tehty neliulotteinen versio 4D-LETKF, joka muistuttaa hieman 4D-Var-menetelmää (katso kappale 3.2.1). Siinä käytetty sakkofunktio on korvattu 4D-Var-menetelmän vastaavalla. Menetelmän johto ei ole oleellisesti hankalampi ja sitä on hahmoteltu edellä mainituissa artikkeleissa.

### 3.1.6 Muista Kalmanin suotimista

Edellisissä kappaleissa mainittujen Kalmanin suotimien lisäksi on kehitelty eri tarkoituksia varten iso joukko muita Kalmanin suotimia. Näille menetelmille yhteistä on, että niissä pyritään välttämään kovarianssimatriisin suoraa laskemista ja/tai mallin linearisointia. Nämä menetelmät juontavat siitä, että perinteistä Kalmanin suodatusta ei jostain syystä – usein ongelman suuren koon takia – voida käyttää, vaan sitä pyritään approksimoimaan käytettävällä menetelmällä. Approksimaatioissa usein kuitenkin menetämme riittävän informaation tilaestimaatista ja virhearvioista.

Muita Kalmanin suotimia ovat muun muassa hajustamaton Kalmanin suodin (eng. Unscented Kalman Filter, UKF) [1][25] ja nopea Kalmanin suodin (eng. Fast Kalman Filter, FKF) [16].

Tässä työssä käsittelemme kuitenkin vielä niin sanotun variaationaalisen Kalmanin suotimen (3.2.4), jossa yhdistetään variaationaalisia data-assimi-laatiomenetelmiä ja Kalmanin suodatusta.



Kuva 3.7: Havaintokuva sakkofunktion minimoimisesta [4].

## 3.2 Variaationaaliset data-assimilaatiomenetelmät

Kolmiulotteisessa variaationaalisessa data-assimilaatiossa (lyh. 3D-Var)<sup>4</sup>, toisin kuin Kalmanin suotimien tapauksissa, on tarkoitus välttää vahvistusmatriisin (3.3) laskemista. Siinä on tarkoitus ratkaista jollain numeerisella algoritmilla pienimmän varianssin ratkaisun kanssa ekvivalentti (lause 3.0.2) minimointiongelma

$$x_{est} = \arg \min_x J(x),$$

missä  $J$  on kuten aiemminkin,

$$J(x) = (x_a - x)^T B^{-1} (x_a - x) + (y - \mathcal{H}(x))^T R^{-1} (y - \mathcal{H}(x))$$

esitetty sakkofunktio. Approksimointi perustuu siihen, että vain pieni määrä iteraatioita suoritetaan. Käytännössä minimointi voidaan pysäyttää kun 'riittävä' määrä iteraatioita on suoritettu tai kun sakkofunktion gradientin

$$\nabla J(x) = 2B^{-1}(x - x_a) - 2H^T R^{-1}(y - \mathcal{H}(x))$$

normi  $\|\nabla J(x)\|$  vähenee 'tarpeeksi'. Edellisessä kaavassa  $H$  on havainto-operaattorin  $\mathcal{H}$  derivaatta. Kuvassa 3.7 on havainnollistettu sakkofunktion minimoimista.

<sup>4</sup>Ongelman ei välttämättä tarvitse olla kolmiulotteinen, sillä lyhenteet 1D-Var ja 2D-Var eivät ole vakiintuneessa käytössä. Ongelman ei kuitenkaan oleteta riippuvan ajasta. Ajasta riippuville ongelmille käytetään (dimensiosta huolimatta) lyhennettä 4D-var.

Variaationaalissa data-assimilaatiossa analyysivirheen kovarianssimatriisi  $A = E((x_{est} - x_{true})(x_{est} - x_{true})^T)$  saadaan seuraavan lauseen [4] avulla.

**Lause 3.2.1** (Hessen informaatiolause). *Analyysivirheen kovarianssimatriisi  $A$  saadaan sakkofunktion  $J$  Hessen matriisin käänteismatriisin avulla kaavalla*

$$A = \left(\frac{1}{2} \text{Hess}(J)\right)^{-1}. \quad (3.47)$$

*Todistus.* Hessen matriisi saadaan derivoimalla kaksi kertaa sakkofunktiota  $J(x)$ . Näin saamme

$$\nabla J(x) = 2B^{-1}(x - x_a) - 2H^T R^{-1}(y - \mathcal{H}(x)) \quad (3.48)$$

$$\nabla \nabla J(x) = 2(B^{-1} + H^T R^{-1} H). \quad (3.49)$$

Nyt käyttäen hyväksi tietoa, että  $\nabla J(x_{est}) = 0$  ja lisäämällä  $x_{true}$  saamme

$$\begin{aligned} 0 &= B^{-1}(x_{est} - x_a) - H^T R^{-1}(y - \mathcal{H}(x_{est})) \\ &= B^{-1}(x_{est} - x_{true} + x_{true} - x_a) - H^T R^{-1}(y - \mathcal{H}(x_{true}) - H(x_{true} - x_{est})) \\ &= B^{-1}(x_{est} - x_{true}) - H^T R^{-1} H(x_{true} - x_{est}) - B^{-1}(x_a - x_{true}) - H^T R^{-1}(y - \mathcal{H}(x_{true})), \end{aligned}$$

joten

$$(B^{-1} + H^T R^{-1} H)(x_{est} - x_{true}) = B^{-1}(x_a - x_{true}) + H^T R^{-1}(y - \mathcal{H}(x_{true})).$$

Kertomalla tätä oikealta transpoosillaan ja ottamalla puolittain odotusarvon saamme

$$\begin{aligned} &E((B^{-1} + H^T R^{-1} H)(x_{est} - x_{true})(x_{est} - x_{true})^T (B^{-1} + H^T R^{-1} H)^T) \\ &= E(B^{-1}(x_a - x_{true})(x_a - x_{true})^T (B^{-1})^T) \\ &+ E(B^{-1}(x_a - x_{true})(y - \mathcal{H}(x_{true}))^T (H^T R^{-1})^T) \\ &+ E(H^T R^{-1}(y - \mathcal{H}(x_{true}))(y - \mathcal{H}(x_{true}))^T (H^T R^{-1})^T) \\ &+ E(H^T R^{-1}(y - \mathcal{H}(x_{true}))(x_a - x_{true})^T (B^{-1})^T). \end{aligned}$$

Koska oletamme, että prioritilat ja mittaukset ovat korreloimattomia saamme huomioimalla matriisien  $A$  ja  $R$  kaavat

$$\begin{aligned} (B^{-1} + H^T R^{-1} H)A(B^{-1} + H^T R^{-1} H)^T &= B^{-1}B(B^{-1})^T + H^T R^{-1}R(H^T R^{-1})^T \\ &= (B^{-1} + H^T R^{-1} H)^T, \end{aligned}$$

joten kertomalla matriisin  $A$  oikealla ja vasemmalla puolella olevien matriisien käänteismatriiseilla saamme

$$A = (B^{-1} + H^T R^{-1} H)^{-1} = \left(\frac{1}{2} \text{Hess}(J)\right)^{-1}.$$

□

**Huomautus.** Mikäli  $H$  voidaan muodostaa tarkasti, esimerkiksi jos  $H$  on lineaarinen, niin sakkofunktion Hessen matriisi ei riipu analyysin arvosta. Itse asiassa kovarianssimatriisi  $A$  voidaan määrittää ennen kuin analyysia on edes tehty.

### 3.2.1 4D-Var

Neliulotteinen variaationaalinen assimilaatio (lyh. 4D-Var) on 3D-Var-menetelmän yleistys, joka on riippuvainen ajasta. Se on operaatiivisessa käytössä säännestämisen. Yhtälöt 4D-Var-menetelmässä ova pääpiirteittäin samoja kuin 3D-Var-menetelmässäkin, mutta nyt ne riippuvat diskreetteistä ajanhetkistä  $t_i = t_0, \dots, t_N$ . 4D-Var-menetelmässä käytettyä aikaa kutsutaan *assimilaatioikkunaksi*. Ideana on, että hetkellä  $t = t_0$  viedään edellisestä vaiheesta saatua estimaatti mallilla eteenpäin assimilaatioikkunan viimeiseen ajanhetkeen asti, jonka jälkeen suoritetaan analyysi, jossa verrataan tulevaisuuden mittauksia mallilla saatuun tietoon. Tämän analyysin pohjalta muutamme arviotamme tilaestimaatista ajanhetkellä  $t_i = t_0$ . Kuvassa 3.8 on havainnollistettu 4D-Var-menetelmän toimintaa.

4D-Var-menetelmässä sakkofunktiomme saa seuraavan muodon

$$\begin{aligned} J(x_{t_0}) &= (x_a - x_{t_0})^T B^{-1} (x_a - x_{t_0}) + \sum_{i=0}^N (y_{t_i} - \mathcal{H}_{t_i}(x_{t_i}))^T R_{t_i}^{-1} (y_{t_i} - \mathcal{H}_{t_i}(x_{t_i})) \\ &= J_a + J_{obs}, \end{aligned}$$

missä  $x_{t_i}$  on tilaestimaatti,  $y_{t_i}$  havainto,  $\mathcal{H}_{t_i}$  havainto-operaattori ja  $R_{t_i}$  havaintovirheen kovarianssimatriisi hetkellä  $t_i$ . Sakkofunktion ensimmäinen termi on sama kuin aikaisemminkin. Lisäksi, kuten Kalmanin suotimien kanssa, meillä on käytössämme malli, joka vie tilaa eteenpäin

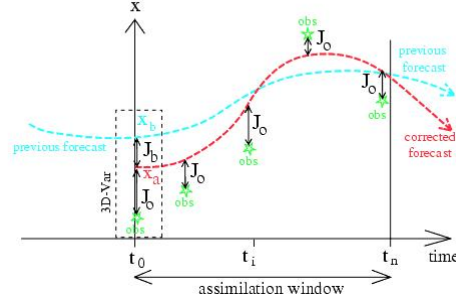
$$x_{t_i} = \mathcal{M}_{t_i}(x_{t_{i-1}}) = \mathcal{M}_{t_i}(\mathcal{M}_{t_{i-1}}(\dots(\mathcal{M}_{t_1}(x_{t_0}))\dots)).$$

Asettakaamme jälleen optimointiongelma

$$x_{est} = \arg \min_{x_{t_0}} J(x_{t_0}).$$

Kuten aikaisemmassa tapauksessa, voidaan osoittaa, että mikäli virheet oletetaan gaussisiksi, niin minimointi on ekvivalentti posteriorifunktion maksimoinnin kanssa.

Yleisesti optimointimenetelmää varten tarvittava sakkofunktion gradientin laskeminen voi olla hankalaa ja numeerisesti kallista. Seuraava esitellään menetelmä sakkofunktion gradientin tehokkaaseen laskemiseen.



Kuva 3.8: Havaintokuva 4D-Varin toiminnasta [4].

Sakkofunktion prioritermi on sama kuin 3D-Var-menetelmässä, joten jätetään se merkintöjen helpottamiseksi käsittelemättä ja merkitään sakkofunktion havaintotermejä symbolilla  $J_{obs,t_i}$ . Lasketaan mallin avulla tilat  $x_{t_i}$  muistiin. Lasketaan myös seuraava skaalattu residuaali muistiin

$$d_{t_i} = R_{t_i}^{-1}(y_{t_i} - \mathcal{H}_{t_i}(x_{t_i})).$$

Derivoimalla saamme

$$\begin{aligned} -\frac{1}{2}\nabla J_{obs}(x) &= -\frac{1}{2}\sum_{i=0}^N \nabla J_{obs,t_i}(x) \\ &= \sum_{i=0}^N \nabla(\mathcal{H}(\mathcal{M}_{t_i}(\mathcal{M}_{t_{i-1}}(\cdots(\mathcal{M}_{t_1}(x))\cdots))))R_{t_i}^{-1}(y_{t_i} - \mathcal{H}_{t_i}(x_{t_i})) \\ &= \sum_{i=0}^N M_{t_0}^T M_{t_1}^T \cdots M_{t_i}^T H_{t_i}^T d_{t_i} \\ &= M_{t_0}^T (H_{t_0}^T d_{t_0} + \underbrace{M_{t_1}^T (H_{t_1}^T d_{t_1} + \cdots + M_{t_N}^T (H_{t_N}^T + \tilde{x}_{t_{N+1}})) \cdots)}_{=\tilde{x}_{t_1}}), \end{aligned}$$

missä  $M_{t_0}^T$  on identtinen matriisi ja  $\tilde{x}_{t_{N+1}} = 0$ . Laskemalla nyt niin sanotut liittomuuttujat  $\tilde{x}_{t_{i-1}} = M_{t_i}^T (H_{t_i}^T d_{t_i} + \tilde{x}_{t_i})$  vaiheittain saamme lopuksi, että

$$\nabla J_{obs}(x) = -2\tilde{x}_{t_0}.$$

Lisäämällä tähän prioritermin gradientin saamme koko sakkofunktion gradientin

$$\nabla J_{obs}(x) = -2B^{-1}(x_a - x) + \nabla J_{obs}(x).$$

Mikäli ongelman koko on suuri, niin lineaarisen operaattorin  $\frac{\partial \mathcal{M}_{t_i}(x_{t_{i-1}})}{\partial x}$  esitysmatriisia  $M_{t_i}$  ei välttämättä pystytä edes tallentamaan muistiin. Edellä olevassa esityksessä esitysmatriisia ei kuitenkaan tarvita matriisimuodossa

vaan kuvauksena  $x \mapsto M_{t_i}x$ . Tälle kuvaukselle voidaan kuitenkin tietokoneella tehdä tangenttilineaarinen koodi, jota yleensä kirjallisuudessa merkitään symbolilla  $M_{t_i}^{tl}$ , jossa matriisia  $M_{t_i}$  ei tarvitse tallentaa muistiin. Tälle ohjelmakoodille voidaan taas tehdä sitä vastaava liittokoodi  $M_{t_i}^*$  (katso kappale 3.2.3). Olennaista on, että  $M_{t_i}^{tl}$  ja  $M_{t_i}^*$  eivät ole enää matriiseja, joten niiden välillä ei voi esimerkiksi suorittaa matriisioperaatioita.  $M_{t_i}^*$  voidaan kuitenkin ajatella  $M_{t_i}^{tl}$ :n transpoosina.

Nyt voimme muodostaa seuraavan algoritmin sakkofunktion havaintotermien gradientin laskemiseksi

**Askel 0:** Määritä assimilaatioikkunan pituus  $N$  ja laske kaikilla ajanhetkillä

$$d_{t_i} = R_{t_i}^{-1}(y_{t_i} - \mathcal{H}_{t_i}(x_{t_i})) \text{ muistiin. Aseta } i = N \text{ ja } \tilde{x} = 0.$$

**Askel 1:** Laske  $\tilde{x} = M_{t_i}^*(H_{t_i}^T d_{t_i} + \tilde{x})$ .

**Askel 2:** Päivitä  $i = i - 1$  ja palaa askeleeseen 1, kunnes  $i = 0$ .

**Askel 3:** Laske lopuksi gradientin arvo  $\nabla J_{obs}(x) = -2\tilde{x}$ .

Kun sakkofunktion gradientti on käytössä, niin periaatteessa 4D-Var-menetelmässä voidaan käyttää mitä tahansa gradientti-pohjaista optimointimenetelmää. Usein käytetään LBFGS-menetelmää. Eri menetelmiä käsitellään seuraavassa kappaleessa.

Yleisesti 4D-Var – toisin kuin Kalmanin suotimet – toimii sen oletuksen alla, että malli on täydellinen. Mikäli mallin oletetaan olevan täydellinen, niin 4D-Var on yhtäpitävä Kalmanin suotimien kanssa, mutta laskennaltaan paljon kevyempi. Pitkällä assimilaatio ikkunalla 4D-Varin on sanottu olevan silottava algoritmi ja se soveltuu erityisen hyvin esimerkiksi sään ennustamiseen.

Ennen siirtymistä seuraavaan data-assimilaatiomenetelmään hankkikaamme hiukan optimointityökaluja ja tarkastelkaamme hiukan tarkemmin edellä esiintynyttä liittokoodia.

### 3.2.2 Optimointimenetelmiä

Tässä kappaleessa käsitellemme lyhyesti optimointimenetelmiä joita tarvitsemme muun muassa sakkofunktion minimoimiseen. Numeerinen kustannus on tärkeä ominaisuus. Lisäksi on tärkeää löytää kompromissi numeerisen nopeuden, tilastollisen optimaalisuuden ja fysikaalisen realismin välillä.

Optimointia varten on kehitelty paljon erilaisia, eri käyttötarkoitukseen soveltuvia menetelmiä. Eräs yksinkertaisimmista menetelmistä on gradientti-pohjainen jyrkimmän laskun menetelmä (eng. the method of steepest descent).

Menetelmistä missä tarvitaan Hessen matriisia suosituin lienee Levenbergin ja Marquardtin menetelmä. Hessen matriisin laskeminen ei kuitenkaan ole käytännöllistä suurikokoisissa data-assimilaatio-ongelmissa, joten sen sijaan esittelemme niin kutsutun BFGS-menetelmän ja sen rajoitetun muistin version. Siinä missä menetelmiä, joissa tarvitaan Hessen matriisia kutsutaan Newtonin menetelmiksi, niin menetelmiä joissa Hessen matriisia approksimoidaan iteratiivisesti kutsutaan kvasi-Newtonin menetelmiksi.

**Jyrkimmän laskun menetelmä.** Olkoon  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  funktio, jonka haluamme minimoida. Jyrkimmän laskun menetelmässä lähdemme ratkaisemaan tätä iteratiivisesti. Lähdemme liikkeelle mielivaltaisesta pisteestä  $x^0$  ja otamme sarjan askelia  $x^1, x^2, \dots$  kunnes olemme 'riittävän' lähellä ratkaisua  $x$ . Kun otamme askeleen valitsemme suunnan johon  $f$  vähenee voimakkaimmin. Tämä suunta on vastakkainen gradienttiin nähden, joten tämä suunta  $r$  on

$$r = -\nabla f(x).$$

Kun iteraatti  $x^i$  on saatu, niin seuraavassa vaiheessa otamme siis pisteen

$$x^{i+1} = x^i + \alpha^i r^i,$$

missä  $\alpha$  on askeleen pituus. Seuraava kysymys kuuluukin, kuinka suuren askeleen  $\alpha$  otamme. Välttämätön, mutta yksinään riittämätön vaatimus on, että  $f$  vähenee eli

$$f(x^i + \alpha^i r^i) < f(x^i).$$

Parametrin  $\alpha$  valinnalle 'riittävät' ehdot saamme esimerkiksi niin kutsutuista Wolfen ehdoista

$$\begin{cases} f(x^i + \alpha^i r^i) & \leq f(x^i) + c_1 \alpha^i \nabla f_i^T r^i, \\ \nabla f(x^i + \alpha^i r^i)^T r^i & \geq c_2 \nabla f_i^T r^i, \end{cases} \quad (3.50)$$

missä  $0 < c_1 < c_2 < 1$ . Näistä ensimmäinen on vähenevyys- ja toinen kaarevuusehto. Voimakkaissa Wolfen ehdoissa jälkimmäinen epäyhtälö korvataan ehdolla

$$|\nabla f(x^i + \alpha^i r^i)^T r^i| \geq c_2 |\nabla f_i^T r^i|.$$

Kuten totesimme, niin vähenevyysehto ei yksinään ole riittävä. Mikäli emme kuitenkaan halua käyttää kaarevuusehtoa, niin voimme turvautua niin kutsuttuun perääntyvään (eng. backtracking) lähestymistapaan. Yksinkertaisimmillaan tämä voidaan esittää seuraavana algoritmia:

**Askel 0:** Valitse  $\alpha > 0$ ,  $\rho \in (0, 1)$  ja  $c \in (0, 1)$ .

**Askel 1:** Tarkista ehto  $f(x^i + \alpha r^i) \leq f(x^i) + c\alpha \nabla f_i^T r^i$ . Mikäli ehto on voimassa, niin lopeta, muulloin siirry askeleeseen 2.

**Askel 2:** Päivitä  $\alpha = \rho\alpha$  ja palaa askeleeseen 1.

Päätöksen siitä milloin olemme 'riittävän' lähellä ratkaisua  $x$  ei ole yhtä oikeaa ratkaisua. Voimme esimerkiksi päättää keinotekoisesti iteraatioiden lukumäärälle ylärajan  $i_{max}$  tai suorittaa iteraatiota kunnes  $\|\nabla f(x^i)\| \leq \varepsilon \|\nabla f(x^0)\|$ , missä  $\varepsilon$  on ennalta määrätty toleranssi. Nyt voimme kirjoittaa jyrkimmän laskun menetelmälle seuraavan algoritmin:

**Askel 0:** Aseta  $i = 0$  ja alkuarvaus  $x$ , toleranssi  $\varepsilon$  ja iteraatiolle yläraja  $i_{max}$ .

**Askel 1:** Laske  $r = -\nabla f(x)$  ja  $\delta = r^T r$ , sekä aseta  $\delta_0 = \delta$ .

**Askel 2:** Valitse  $\alpha$  esimerkiksi, niin että se toteuttaa Wolfen ehdot ja päivitä  $x = x + \alpha r$ .

**Askel 3:** Laske  $r = -\nabla f(x)$  ja  $\delta = r^T r$ .

**Askel 5:** Päivitä  $i = i + 1$ . Palaa askeleeseen 2 kunnes  $i > i_{max}$  tai kunnes  $\delta > \varepsilon^2 \delta_0$ .

**BFGS-menetelmä.** Seuraavaksi esittelemme kuuluisan BFGS-menetelmän eli Broydenin, Fletcherin, Goldfarbin ja Shannon menetelmän, joka on niin kutsuttu kvasi-Newtonin menetelmä. Esittelemme lisäksi sen rajoitetun muistin version LBFGS-menetelmän (eng. Limited memory BFGS-method).

Olettakaamme minimoitavalle funktiolle  $f$  toisen asteen Taylorin sarjan approksimaatio

$$f(x + \Delta x) \approx f(x) + (\nabla f(x))^T \Delta x + \frac{1}{2}(\Delta x)^T B \Delta x,$$

missä  $B$  on funktion  $f$  Hessen matriisi. Tämän approksimaation gradientti  $\Delta x$ :n suhteen on

$$\nabla f(x + \Delta x) \approx \nabla f(x) + B \Delta x.$$

Jatkossa merkitsemme tehdyt approksimaatiot tarkaksi. Asettamalla gradientin nolaksi saamme Newtonin askeleen

$$\Delta x = -B^{-1} \nabla f(x).$$

Käyttämällä samaa logiikkaa kuin jyrkimmän laskun menetelmässä voimme kirjoittaa

$$\Delta x^i = -\alpha^i B_i^{-1} \nabla f(x^i)$$

ja edelleen

$$x^{i+1} = x^i + \Delta x^i.$$



Merkitään

$$z^i = \nabla f(x^{i+1}) - \nabla f(x^i).$$

BFGS-menetelmässä edellä olevat yhtälöt ja minimointiongelmassa

$$\min_B \|B - B_k\|_F$$

käytetty Frobeniuksen normi johtavat siihen, että Hessen matriisin käänteismatriisin päivytyksessä käytetään seuraavaa kaavaa [19]

$$B_{i+1}^{-1} = \left( I - \frac{z^i(\Delta x^i)^T}{(z^i)^T \Delta x^i} \right)^T B_i^{-1} \left( I - \frac{z^i(\Delta x^i)^T}{(z^i)^T \Delta x^i} \right) + \frac{\Delta x^i(\Delta x^i)^T}{(z^i)^T \Delta x^i}.$$

Lisäksi tarvittaessa voimme laskea

$$B_{i+1} = B_i + \frac{z^i(z^i)^T}{(z^i)^T \Delta x^i} - \frac{B_i \Delta x^i (B_i \Delta x^i)^T}{(\Delta x^i)^T B_i \Delta x^i}.$$

Nyt olemme valmiit kirjoittamaan seuraavan BFGS-algoritmin:

**Askel 0:** Aseta  $i = 0$ , alkuarvaus  $x_0$ , alkuarvaus kovarianssimatriisin käänteismatriisille  $B_0^{-1}$ , toleranssi  $\varepsilon$  ja iteraatiolle yläraja  $i_{max}$ .

**Askel 1:** Laske  $g_0 = \nabla f(x_0)$  ja  $\delta = g_0^T g_0$ , sekä aseta  $\delta_0 = \delta$ .

**Askel 2:** Laske  $r_i = -B_i^{-1} g_i$  ja etsi  $\alpha_i$ , joka toteuttaa voimakkaat Wolfen ehdot.

**Askel 3:** Laske  $x_{i+1} = x_i + \alpha_i r_i$ .

**Askel 4:** Laske  $g_{i+1} = \nabla f(x_{i+1})$  ja  $\delta = g_{i+1}^T g_{i+1}$ .

**Askel 5:** Laske  $z_i = x_{i+1} - x_i$ ,  $d_i = g_{i+1} - g_i$ ,  $\rho_i = 1/(d_i^T z_i)$ ,  $V_i = I - \rho_i d_i z_i^T$  ja  $B_{i+1}^{-1} = V_i^T B_i^{-1} V_i + \rho_i z_i z_i^T$ .

**Askel 6:** Päivitä  $i = i + 1$ . Palaa askeleeseen 2 kunnes  $i > i_{max}$  tai kunnes  $\delta > \varepsilon^2 \delta_0$ .

**Rajoitetun muistin versio.** Mikäli ongelmamme on laaja, niin matriisin  $B^{-1}$  tallentaminen kokonaan muistiin ei enää onnistu. Tämä motivoi käyttämään rajoitetun muistin versiota algoritmista. Oletetaan, että iteraation hetkellä  $i$  on tallennettu  $j$  kappaletta vektoripareja  $\{z_i, d_i\}_{k=i-j}^{i-1}$ . Tällöin voimme laskea Hessen käänteismatriisille seuraavasti

$$\begin{aligned} B_i^{-1} &= (V_{i-1}^T \cdots V_{i-j}^T) B_0^{-1} (V_{i-j+1} \cdots V_{i-1}) \\ &\quad + \rho_{i-j} (V_{i-1}^T \cdots V_{i-j+1}^T) z_{i-j} z_{i-j}^T (V_{i-j+1} \cdots V_{i-1}) \\ &\quad + \dots \\ &\quad + \rho_{i-1} z_{i-1} z_{i-1}^T. \end{aligned}$$

Tästä voimme laskea tulon  $B_i^{-1}g_i$  tehokkaasti.

Mikäli laskut ovat tarkkoja ja  $j$  on yhtä suuri kuin ongelman koko  $n$ , niin  $x^i$  konvergoi kohti  $f$ :n yksikäsitteistä minimoijaa enintään  $n$ :llä iteraatilla. Mikäli  $n$  iteraattia suoritetaan, niin  $B_{n+1}^{-1} = B^{-1}$ . Eri minimointimenetelmiä on käsitelty muun muassa teoksissa [19][13][24].

### 3.2.3 Adjungaattimenetelmistä

Kuten todettua, niin aina emme voi edes tallentaa mallimme  $M$  linearisaatioita matriisimuodossa. Kuitenkin voimme tulkita matriisin lineaarikuvauksena  $M: \mathbb{R}^n \rightarrow \mathbb{R}^n$ , jolloin voimme kirjoittaa tämän kuvauksen tangenttilineaarisena ohjelmakoodina, jota tarvitsimme muun muassa 4D-Var-menetelmän yhteydessä. Tälle ohjelmakoodille voimme kirjoittaa sitä vastaavan liittokoodin, joka on tulkittavissa matriisin  $M$  transpoosiksi. Yleisesti adjungaatti määritellään seuraavasti.

**Määritelmä 3.2.1.** *Olkoon  $A$  lineaarikuvaus  $E \rightarrow F$ . Ja olkoot  $\langle \cdot, \cdot \rangle_E$  ja  $\langle \cdot, \cdot \rangle_F$  sisätuloja. Tällöin  $A$ :n adjungaatti saadaan kaavasta*

$$\langle Ax, y \rangle_F = \langle x, A^*y \rangle_E. \quad (3.51)$$

**Huomautus** Meidän tapauksessamme  $E$  ja  $F$  ovat aina äärellisulotteisia, joten adjungaatin olemassa olon kanssa ei tule ongelmia, mikäli vain  $A$  on olemassa.

Määritelmä 3.2.1 antaa myös mahdollisuuden testata onko adjungaatille tehty liittokoodi oikea. Tyypillisesti ratkaistu malli on differentiaaliyhtälön numeerinen ratkaisu. Mikäli malli annetaan differentiaaliyhtälöinä, niin myös ratkaisija joudutaan ottamaan mukaan kaavoihin. Havainnollistetaan liittokoodin laskemista Brusselator-esimerkin valossa.

**Esimerkki 3.2.1.** *Palautetaan mieliin Brusselator-yhtälöt*

$$\begin{cases} u' &= 1 + au^2v - (b+1)u \\ v' &= bu - au^2v, \end{cases} \quad (3.52)$$

*joista Eulerin menetelmällä saatu diskretoitu ratkaisu on*

$$\begin{cases} u_{t_i} &= u_{t_{i-1}} + \Delta t u'_{t_{i-1}} \\ v_{t_i} &= v_{t_{i-1}} + \Delta t v'_{t_{i-1}}, \end{cases} \quad (3.53)$$

*missä  $\Delta t$  on käytetty aika-askel.*

Nyt voimme kirjoittaa diskretoidun mallimme muodossa

$$\mathcal{M}_{t_i}(u_{t_{i-1}}, v_{t_{i-1}}) = (u_{t_i}, v_{t_i}).$$

Tässä esimerkissä oletamme, että Eulerin menetelmässä käytetty aika-askel on sama kuin data-assimilaatiossakin. Mikäli näin ei haluttaisi toimia voilla olevan mallin sisään kirjoittaa useamman Euler-askleen. Esimerkiksi, jos välin keskipisteessä halutaan suorittaa Euler-askel, niin kirjoitamme

$$\mathcal{M}_{t_i}(u_{t_{i-1}}, v_{t_{i-1}}) = (u_k + k(u_{t_{i-1}} + ku'_{t_{i-1}}), v_k + k(v_{t_{i-1}} + kv'_{t_{i-1}})),$$

missä  $k = (t_i - t_{i-1})/2$ .

Emme jatkossa merkitse enää indeksiä  $t$  näkyviin. Lasketaan  $\mathcal{M}$ :n Jacobin matriisi yhtälöiden (3.52) ja (3.53) avulla ja saadaan

$$M = \begin{pmatrix} 1 + \Delta t(2auv - (b+1)) & \Delta t(au^2) \\ \Delta t(b - 2auv) & 1 + \Delta t(-au^2) \end{pmatrix}.$$

Kun matriisi tulkitaan lineaarikuvaukseksi  $(z, h) \in \mathbb{R}^2 \rightarrow \mathbb{R}^2$  eli

$$\begin{pmatrix} \partial u \\ \partial v \end{pmatrix} = M \begin{pmatrix} z \\ h \end{pmatrix}, \quad (3.54)$$

niin voimme laskea seuraavat tangenttilineaariset yhtälöt

$$\begin{cases} \partial u &= z + \Delta t((2auv - (b+1))z + (au^2)h), \\ \partial v &= h + \Delta t((b - 2auv)z + (-au^2)h). \end{cases}$$

Vastaavasti ottamalla matriisista  $M$  transpoosin ja tulkitsemalla tämän lineaarikuvaukseksi saamme liittoyhtälöt

$$\begin{cases} \partial \tilde{u} &= z + \Delta t((2auv - (b+1))z + (b - 2auv)h), \\ \partial \tilde{v} &= h + \Delta t((au^2)z + (-au^2)h). \end{cases}$$

Tangenttilineaaristen yhtälöiden avulla voimme kirjoittaa ohjelmakoodin  $M^{tl}$  ja liittoyhtälöiden avulla ohjelmakoodin  $M^*$  ilman, että vastaavia matriiseja tarvitsee tallentaa muistiin.

Yleisesti liittokoodin tekeminen on työlästä, mutta usein monien käytännön data-assimilaatiosysteemien kannalta välttämätöntä. Liittokoodin teoreettista muodostamista ja käytännön koodin tekoa on käsitelty esimerkiksi [12][9].

### 3.2.4 Variaationaalinen Kalmanin suodin

Kuten jo todettua, mikäli assimilaatio-ongelman koko on iso, niin perinteistä Kalmanin suodatuksesta tulee epäkäytännöllinen tai jopa toteuttamiskelvoton. Variaationaalisessa Kalmanin suotimessa (eng. Variational Kalman Filter, VKF) hyödynnetään sekä variaationaalisia data-assimilaatiomenetelmiä, sekä Kalmanin suodatusta. Esittelemme ainoastaan epälineaarisen version – lineaarinen versio menee vastaavasti. Menetelmän ideana on hyödyntää LBFGS-menetelmästä saatua kovarianssimatriisien ja niiden käänteismatriisin rajoitetun muistin approksimaatiota ja käyttää niitä minimoitaessa sakkofunktioita. Jatkossa ajattelemmekin matriiseja  $C_{est}$  ja  $C_a^{-1}$  niiden rajoitetun muistin versioina.

Ensimmäisessä vaiheessa lasketaan prioriestimaatti kuten aiemminkin viemällä edellisessä vaiheessa saatua estimaattia eteenpäin mallilla

$$x_{a,t_i} = \mathcal{M}_{t_i}(x_{est,t_{i-1}}).$$

Tarvittava priorikovarianssin  $C_{a,t_i}$  käänteismatriisin saamme minimoimalla funktiota

$$J_C(x) = x^T M_{t_i} C_{est,t_{i-1}} M_{t_i}^T x + x^T C_{\xi_{t_i}} x$$

LBFGS-algoritmillä. Tässä oletetaan, että lineaarinen operaattori  $M_{t_i}$  on voitu muodostaa kuten kaavassa (3.29). Mikäli tangenttilineaarinen koodi ja sitä vastaava liittokoodi on käytössä, niin voimme kirjoittaa minimoitavan funktion muodossa

$$J_C(x) = x^T (M_{t_i}^{tl} C_{est,t_{i-1}} M_{t_{i-1}}^* x + C_{\xi_{t_i}} x).$$

Tämän jälkeen minimoimme sakkofunktiota, jossa on käytetty edellä saatuja estimaatteja

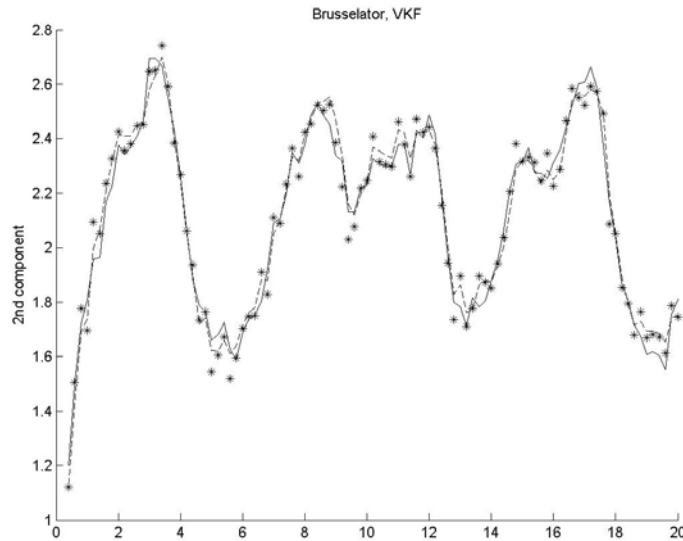
$$J(x) = (x - x_{a,t_i})^T C_{a,t_i}^{-1} (x - x_{a,t_i}) + (y_{t_i} - \mathcal{H}_{t_i}(x))^T C_{\varepsilon_{t_i}}^{-1} (y_{t_i} - \mathcal{H}_{t_i}(x)),$$

käyttäen LFBGS-menetelmää. Tämän minimoinnin seurauksena saamme estimaatit  $x_{est,t_i}$  ja  $C_{est,t_i}$ . Nyt voimme muodostaa seuraavan VKF-algoritmin:

**Askel 0:** Valitse alkuarvaus  $x_{est,t_0}$ , kovarianssimatriisi  $C_{est,t_0}$  ja aseta  $i = 1$ .

**Askel 1:** Minimoi LBFGS-algoritmillä  $J_C(x) = x^T M_{t_i} C_{est,t_{i-1}} M_{t_i}^T x + x^T C_{\xi_{t_i}} x$  tai vaihtoehtoisesti  $J_C(x) = x^T (M_{t_i}^{tl} C_{est,t_{i-1}} M_{t_i}^* x + C_{\xi_{t_i}} x)$  ja talleta saatu kovarianssimatriisin käänteismatriisi  $C_{a,t_i}^{-1}$ .

**Askel 2:** Minimoi LBFGS-algoritmillä sakkofunktio  $J(x) = (x - x_{a,t_i})^T C_{a,t_i}^{-1} (x - x_{a,t_i}) + (y_{t_i} - \mathcal{H}_{t_i}(x))^T C_{\varepsilon_{t_i}}^{-1} (y_{t_i} - \mathcal{H}_{t_i}(x))$  ja talleta saadut estimaatit  $x_{est,t_i}$  ja  $C_{est,t_i}$ .



Kuva 3.9: Brusselatorin toinen komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä ja VKF:stä saatua estimaattia katkovii-valla.

**Askel 3:** Päivitä  $i = i + 1$  ja palaa askeleeseen 1.

Kuvassa 3.9 on simuloitu VKF:n toimintaa Brusselator-esimerkin avulla. Liitto- ja tangenttilineaarinen koodi on saatu kuten esimerkissä 3.2.1. Minimoointialgoritmina käytettiin tavallista BFGS-algoritmia.

VKF on suhteellisen uusi menetelmä, eikä se ole vielä laajalti käytössä. Kuitenkin sitä pystyy käyttämään huomattavasti isompiin assimilaatio-ongelmiin kun mihin tavallisilla Kalmanin suotimilla voidaan toteuttaa. Eräs keskeinen hyöty verrattuna 4D-Var-menetelmään on, että VKF:ssä voidaan käyttää myös mallivirhettä, kun taas 4D-Var-menetelmässä mallin oletetaan olevan täydellinen. Laskennallisesti VKF on samankaltainen 4D-Var-menetelmän kanssa. Pullonkaulana voidaan pitää  $C_{a,t_i}^{-1}$  laskemista, koska sen laskemiseen tarvitaan liittokoodia, jonka tekeminen on työlästä. VKF-menetelmää on esitetty tarkemmin artikkelissa [2].



# Luku 4

## Numeeriset esimerkit

Tässä luvussa tarkastelemme erästä data-assimilaatiossa usein käytettyä niin kutsuttua benchmark-mallia. Käytämme sitä vertaillaksemme eri data-assimilaatiomenetelmiä. Hyödynnamme vertailussa myös jo aikaisemmin esiintynyttä Brusselator-esimerkkiä. On tärkeää verrata eri data-assimilaatiomenetelmiä benchmark-malleilla, jotta meillä on luottamus ja käytönnön tuntuma assimilaatiomenetelmäämme. Se, että menetelmä toimii benchmark-mallissa, ei vielä takaa sitä, että menetelmä toimii 'oikeassa' data-assimilaatio-ongelmassa. Ennen siirtymistä eri menetelmien vertailuun määritellään muutama suure, jolla assimilaation 'hyvyyttä' voidaan mitata.

*Suhteellinen virhe* (eng. *relative error*) on suure

$$[\text{suhteellinen\_virhe}]_{t_i} = \frac{\|x_{est,t_i} - x_{true,t_i}\|}{\|x_{true,t_i}\|}$$

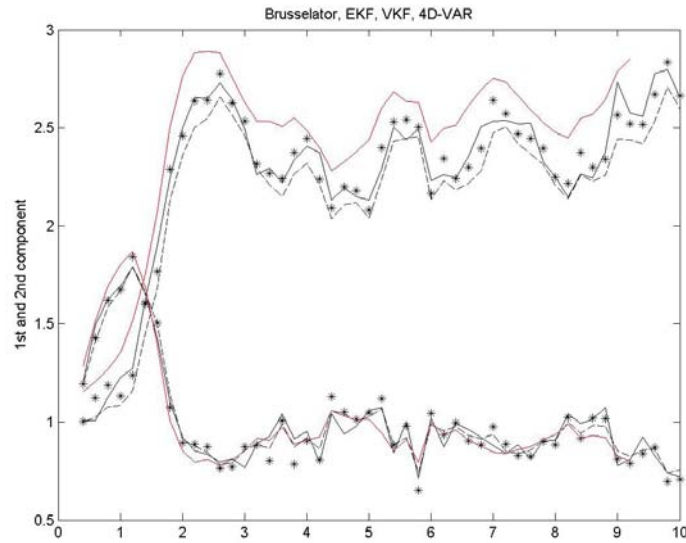
ja *neliöllinen keskiarvo* (eng. *root mean square, rms*) on

$$[\text{rms}]_{t_i} = \sqrt{\frac{1}{n} \|x_{est,t_i} - x_{true,t_i}\|^2}.$$

### 4.1 Brusselator

Brusselator yhtälöt on esitetty jo esimerkissä 3.1.3 ja niille on laskettu tangenttilineaarinen koodi ja liittokoodi esimerkissä 3.2.1. Simuloinnissa käytetyt virhearviot ja alkuarvot ovat samat kuin aikaisemmissakin esimerkeissä.

Näin pienessä mallissa ei menetelmien toiminnassa ole juurikaan eroa, mikäli malliin ei oleteta systemaattista virhettä eli harhaa (eng. *bias*). Yleensä pienissä malleissa voidaan EKF:n ajatella olevan perusmenetelmä ja muut sen approksimaatioita.



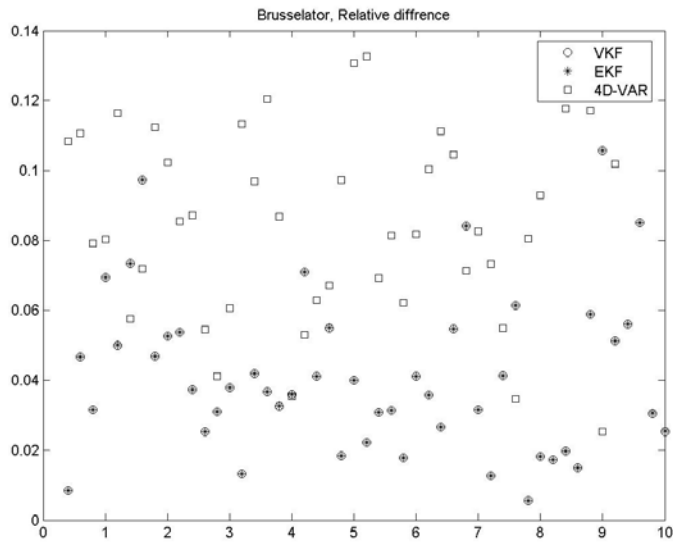
Kuva 4.1: Brusselator-mallin ensimmäinen ja toinen komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä, EKF:stä saatua estimaattia katkoviivalla, VKF:stä saatua estimaattia pisteviivalla ja 4D-Var-menetelmästä saatua estimaattia punaisella viivalla.

Seuraavaksi testataan harhan vaikutusta eri menetelmiin. Oletetaan, että Brusselator-yhtälöissä käytetyt parametrien arvot ovat tuntemattomia ja simuloidaan oikea arvo käyttäen parametrien arvoja  $a = 1,2$  ja  $b = 2,05$ , mutta käyttäen assimilaatiomallissamme arvoja  $a = 1$  ja  $b = 2$ . Lisäksi ensimmäiseen komponenttiin lisätään arvo 0,2.

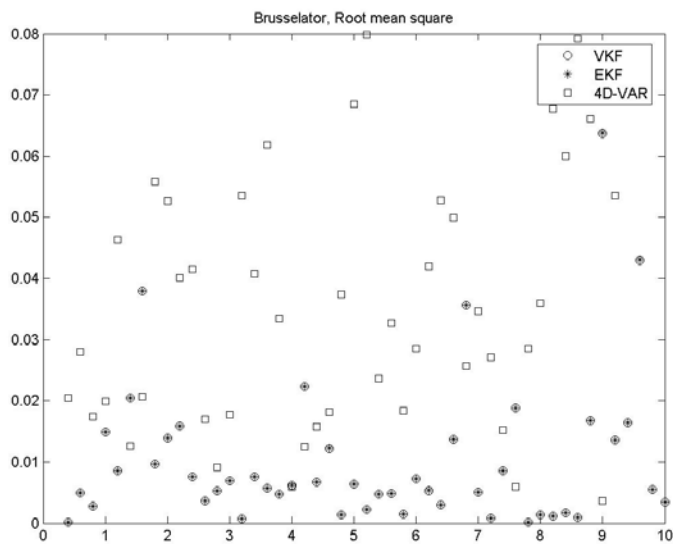
Tarkastellaan EKF:n, VKF:n ja 4D-Var-menetelmän eroja. Kuvassa 4.1 on piirretty assimilaation tulos molemmissa komponentissa. Kuvassa 4.2 on laskettu menetelmän tekemä suhteellinen virhe ja kuvassa 4.3 neliöllinen keskiarvo. Tässä esimerkissä EKF ja VKF tuottavat täsmälleen saman tuloksen. Mikäli ongelman koko olisi niin suuri, että käytettäisiin aidosti LBFGS-optimointimenetelmää, niin menetelmien erot tulisivat esiin. Havaitaan myös, että 4D-Var-menetelmä ( $N = 4$ ) toimii heikommin. Mikäli valitaan  $N = 1$ , niin tulokset paranevat (ei näytetty). Tämä johtuu nimenomaan harhan vaikutuksesta.

Jatketaan testausta käyttäen LETKF- ja EKF-menetelmiä. Käytetään LETKF-menetelmässä viritysparameetrin arvoa  $r = 3$ . Ajon tulos on esitetty kuvassa 4.4. Havaitsemme, että kovarianssikorjattu LETKF antaa paremman estimaatin kuin EKF. Mikäli kovarianssimatriisia ei korjata, niin LETKF-menetelmä ei toimi lainkaan. Huomautetaan vielä, että mikäli viritysparame-

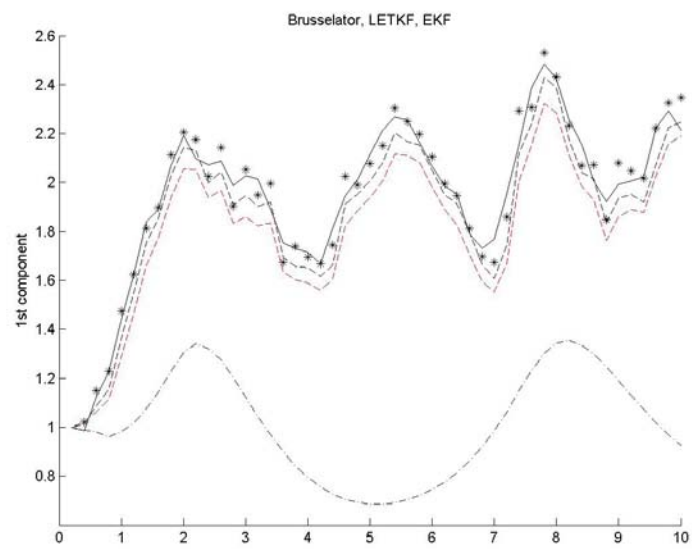




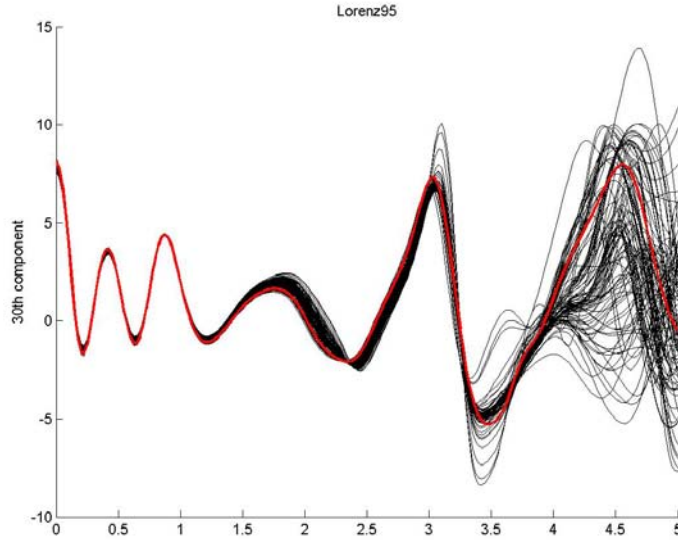
Kuva 4.2: Brusselator-simulaatiossa muodostuneet estimaattien tekemät suhteelliset virheet.



Kuva 4.3: Brusselator-simulaatiossa muodostuneet neliölliset keskiarvot.



Kuva 4.4: Brusselatorin ensimmäinen komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä, LETKF:stä saatua estimaattia pisteviivalla, EKF:stä saatua estimaattia punaisella katkoviivalla ja LETKF-estimaattia, joka on saatu kovarianssikorjauksen avulla ( $r = 3$ ) katkoviivalla.



Kuva 4.5: Lorenz95. Kolmaskymmenes komponentti ( $N = 40$ ). Viivat ovat eri ennustuksia lähdetessä vähän häiriötetystä tilasta. Punainen viiva kuvaa ennustusta lähettäen oikeasta tilasta.

metrina käytetään 'liian' suurta arvoa, niin LETKF rupeaa painottamaan pelkkää mittausta.

## 4.2 Lorenz95

Lorenz95 on Edward Lorenzin vuonna 1995 esittelemä dynaaminen systeemi, jolla on samanlainen virheiden kasvu kuin operatiivisessa käytössä olevilla numeerisen sään ennustussysteemeillä [17]. Systeemi muodostuu yhtälöistä

$$\frac{dx_k}{dt} = -x_{k-2}x_{k-1} + x_{k-1}x_{k+1} - x_k + F, \quad (4.1)$$

missä  $k = 1, \dots, N$  ja reunaehtoina on  $x_0 = x_N$ ,  $x_{-1} = x_{N-1}$  ja  $x_{N+1} = x_1$ . Esimerkiksi arvolla  $F = 8$  systeemistä tulee kaoottinen. Systeemin kaoottisuutta on havainnollistettu kuvassa 4.5. Systeemin voi esimerkiksi ajatella kuvaavan säätilaa kiinnitetyllä pituuspiirillä. Systeemin käyttämä aikayksikkö vastaa viittä vuorokautta.

Tutkimme EKF:n ja EnKF:n eroja Lorenz95-systeemin avulla. Teimme simulaatiota varten kaksi koetta eri ensemble-koon arvolla. Käytännön simulointi varten asetimme seuraavat kovarianssimatriisit alkuarvaukselle, malli-

virheelle ja havaintovirheelle:

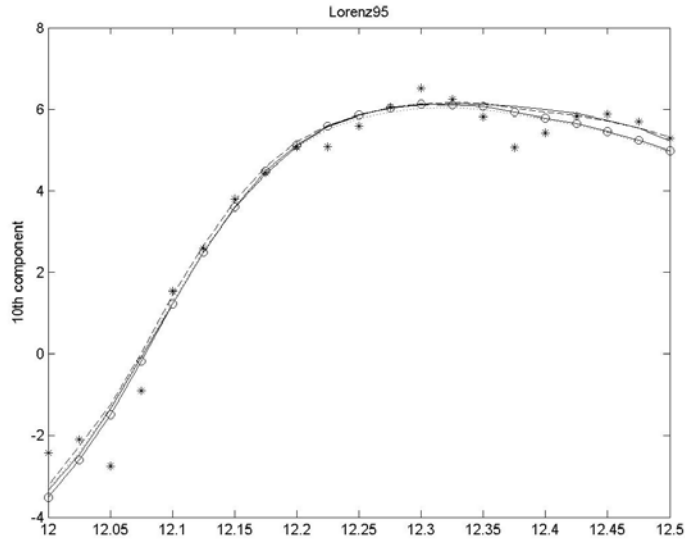
$$\begin{aligned} C_{est,t_0} &= (0,3\sigma_{clim})^2 I \\ C_{\xi,t_i} &= (0,005\sigma_{clim})^2 I \\ C_{\varepsilon,t_i} &= (0,15\sigma_{clim})^2 I \end{aligned}$$

missä  $\sigma_{clim} = 3,6414723$  on eräs ilmastollisista simulaatioista saatu approksimatiivinen keskihajonta. Alkutilan systeemille saimme ajamalla mallia ilman assimilaatiota pitkään alkutilasta, missä asetimme  $x_i = 8$  kaikilla  $i = 1, \dots, 40$  paitsi  $x(30) = 8,008$ . Näin saatua alkutilaa käytimme myös alkuarvauksena kaikille assimilaatiomenetelmille. Käytimme suoria havaintoja, mutta havaintoja ei tehty jokaisessa hilapisteessä. Havaintoverkkomme koostui seuraavista pisteistä  $3, 4, 5, 8, 9, 10, \dots, 38, 39, 40$ . Ensemblen kokona käytimme arvoja  $N_{ens} = 30$  ja  $N_{ens} = 80$ . Assimilaation suoritimme  $\Delta t = 0,025$  aikayksikön (eli kolmen tunnin) välein.

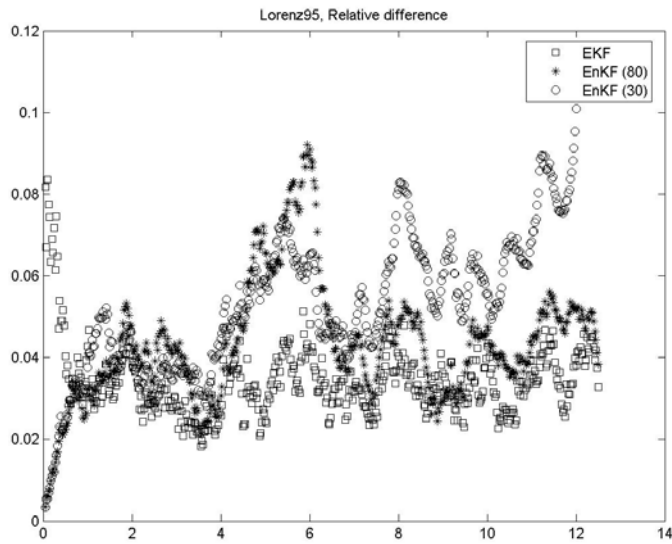
Simulointimme suoritimme seuraavasti: Veimme tilan  $x_{true,t_i}$  Lorenz95-mallilla eteenpäin ja generoimme tähän mallivirheen satunaislukugeneraattorilla. Mittauksen saimme generoimalla näin saatuun oikeaan arvoon mittavirhettä. Mallissa tarvittavan differentiaaliyhtälöryhmän ratkaisimme Matlabin differentiaaliyhtälöratkaisijalla. Mallin linearisoinnin laajennettua Kalmanin suodinta varten laskimme Eulerin menetelmän avulla kuten kappaleessa 3.2.3.

Kuvassa 4.6 on piirretty assimilaation tulos kymmenennessä komponentissa molemmilla käyttämillämme ensemblen koilla. Kuvassa 4.7 on laskettu menetelmän tekemä suhteellinen virhe ja kuvassa 4.8 neliölliset keskiarvot samoin molemmilla ensemblen koilla.

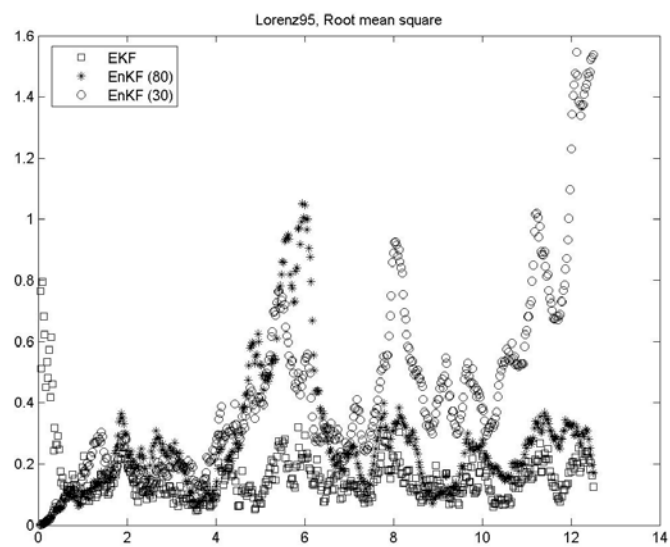
Havaitsemme, että kun ensemblen koko on pieni, niin virhearviot alkavat kasvaa ajan kuluessa. Kun ensemblen kokoa kasvatetaan, niin EnKF yhtyy EKF:ään. Tulos on odotettu kappaleessa (3.1.4) esitetyn tuloksen perusteella.



Kuva 4.6: Lorenz95:n kymmenes komponentti. Oikeata arvoa on merkitty suoralla viivalla, 'mittauksia' tähdellä, EKF:stä saatua estimaattia katkoviivalla ja EnKF:stä saatuja estimaatteja palloviivalla ( $N_{ens} = 80$ ) ja pisteviivalla ( $N_{ens} = 30$ ).



Kuva 4.7: Lorenz95-simulaatiossa muodostuneet estimaattien tekemät suhteelliset virheet.



Kuva 4.8: Lorenz95-simulaatiossa muodostuneet neliölliset keskiarvot.

# Luku 5

## Data-assimilaatio ROSE-kemiakuljetusmallissa

Tässä luvussa käsittelemme keski-ilmakehän kemiallisia prosesseja kuvaavan ROSE-kemiakuljetusmallia ja satelliitti-instrumenteista havaitun otsonin assimiloimista. Tavoitteenamme on assimiloida GOMOS-instrumentista saatua otsonia.

### 5.1 Malli ja data

ROSE-kemiakuljetusmalli (Research for Ozone in the Stratosphere and its Evolution, ROSE) [23] on Klaus Rosen (dynamiikka) ja Guy Brasseur'n (kemian) alunperin 1980 luvulla kehittämä kemiakuljetusmalli.

Tässä työssä käytämme NCAR:n (the National Center for Atmospheric Research, NCAR) muunnelmaa ROSE-mallista, joka fokusoi ilmakehän mesosfäärin kemiaan ja dynamiikkaan [26]. ROSE:ssä on mallinnettuna 29 kemiallista ainetta. Se on hilapistemalli, jonka dimensioidut ovat:

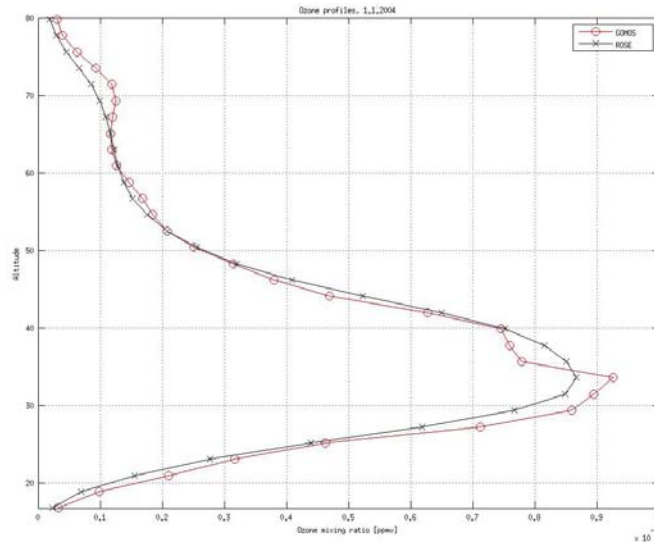
**nx:** 32 pituuspiiriä 11,25 asteen resoluutiolla välillä  $0^{\circ}$ – $348,75^{\circ}$ .

**ny:** 36 leveyspiiriä 5 asteen resoluutiolla välillä  $-87,5^{\circ}$ – $87,5^{\circ}$ .

**nz:** 64 kerrosta 0,3 skaalauskerroimen resoluutiolla välillä 2,4–21,3.

Mallin tilavektorin koko on siis  $32 \times 36 \times 64$  eli 73728 pistettä. Malli ratkaistaan lyhytkestoissa kemiallisissa aineissa implisiittisellä Eulerin menetelmällä ja pitkäkestoissa semi-implisiittisellä backward Eulerin menetelmällä. Se käyttää kiintää 7,5 minuutin aika-askelta.

Korkeus saadaan skaalauskerroimesta kertomalla se seitsemällä kilometrillä eli esimerkiksi mallin alin kerros on  $2,4 \times 7 = 16,8$  kilometriä.



Kuva 5.1: Kaksi otsonin pystyprofiilia ekvaattorilla 1.1.2004. GOMOS-instrumentista saatua profiilia on merkitty ympyrällä ja ROSE-mallin prioritilaa ristillä.

**Havainnot.** Havaintoina käytämme GOMOS-instrumentista (Global Ozone Monitoring by Occultation of Stars, GOMOS) [15] saatuja otsonin pystyprofiileja. Otsoniprofiilien 15–100 kilometrin välillä oleva pystyresoluutio on parempi kuin 1,7 kilometriä ja niillä on hyvä globaali peitto. GOMOS-instrumentti laukaistiin Envisat-satelliitin mukana maaliskuussa 2002. Envisat (Environmental satellite) on Euroopan avaruusjärjestö ESA:n ympäristöä mittaava kaukokartoitussatelliitti.

GOMOS-instrumentti mittaa valon intensiteettiä kolmella eri aallonpituuskaistalla. Instrumentin mittaus perustuu tähtiokkultaatioon eli mittauksissa käytetään valon lähteenä eri tähtiä. Seurattavien tähtien magnitudi eli kirkkaus vaikuttaa mittaustuloksen tarkkuuteen. Mittaustulos on sitä luotettavampi mitä kirkkaampi tähti on mittauksen kohteena [3].

Kuvassa 5.1 on havainnollistettuna GOMOS-instrumentista saatu otsoniprofiili, sekä ROSE-mallin lähtötila ekvaattorilla 1.1.2004.

## 5.2 Assimilaatiomenetelmä

Ensimmäisen kerran Kalmanin suotimia sovellettiin satelliittihavaintojen assimilaatioon globaalin ilmakehän kuljetusmallissa vuonna 1997 [14]. Tässä



työssä käytämme lähtökohtaa, jota on käytetty muun muassa MLS-instrumentista (Microwave Limb Sounder, MLS) saatavan otsonin assimilointiin. Assimilaatio perustuu Kalmanin suotimiin ja sitä on esitelty muun muassa töissä [18][14].

Koska mallin hila on huomattavan suuri, niin meillä tarvetta approksimaatioihin. Ensimmäinen approksimaatio on, että assimilatio suoritetaan joka kerrokselle erikseen. Tämä tarkoittaa sitä, että priori- ja havaintovirheiden kovarianssit asetetaan pystysuunnassa nolliksi. GOMOS-instrumentista saamamme otsonin interpoloimme mallikorkeuksille. Kerroksittain suoritettu assimilatio vähentää kerralla ratkaistavan ongelman koon 73728 pisteestä  $(32 \times 36 \times 64)$  1152 pisteeseen  $(32 \times 36)$ , joka on mahdollista ratkaista nykypäivän tietokoneilla.

Menetelmämme perustuu siis Kalmanin suotimiin eli käytämme seuraavia yhtälöitä (merkinnät kuten sivulla 18)

$$\begin{aligned} x_{a,t_i} &= \mathcal{M}_{t_i}(x_{est,t_{i-1}}), \\ x_{est,t_i} &= x_{a,t_i} + G_{t_i}(y_{t_i} - H_{t_i}(x_{a,t_i})), \\ G_{t_i} &= C_{a,t_i} H_{t_i}^T (H C_{a,t_i} H_{t_i}^T + C_{\varepsilon,t_i})^{-1}, \\ C_{est,t_i} &= (I - G_{t_i} H_{t_i}) C_{a,t_i}. \end{aligned}$$

Koska satelliittimittaukset eivät osu mallimme hilapisteisiin ei havainto-operaattori  $H$  ole identtinen matriisi. Muodostomme matriisin  $H$  siten, että sillä operoiminen vastaa etäisyyden neliöllä kääntäen painotettua hilanurkien malliarvojen keskiarvon laskemista. Tällöin  $H$  on matriisi, jonka yhdellä rivillä on enintään neljä nollasta eroavaa alkioita.

Havaintovirheen kovarianssimatriisin  $C_{\varepsilon,t_i}$  oletamme diagonaalimatriisiksi eli

$$C_{\varepsilon,t_i,j,j} = \sigma_{t_i,j,j}^2,$$

missä  $\sigma_{t_i,j,j}$  on GOMOS-tuotteen keskihajonta.

Taustavirheen kovarianssin  $C_{a,t_i}$  muodon oletamme pysyvän ajan suhteen vakiona [18][14]. Matriisin  $C_{a,t_i}$  diagonaali-alkioiksi oletamme 10 prosenttia mallin ennustamasta otsonin sekoitussuhteesta. Tämä voi olla huono arvio kun pitoisuus on pieni. Diagonaalien ulkopuoliset elementit saadaan gaussisesta funktiosta, jonka korrelaatiopituus on kiinnitetty. Korrelaatiopituutena  $L$  käytämme kymmentä astetta, joka vastaa noin 1000 kilometriä. Lisäksi haluamme, että matriisi  $C_{a,t_i}$  on harva eli siinä on paljon nolla-alkioita. Tämän vuoksi asetamme elementin arvon nolliksi, jos sen korrelaatiokertoimen arvo

on pienempi kuin  $\delta = 0,0001$ . Nyt siis saamme, että

$$C_{a,t_i,j,k} = \begin{cases} (0,1O_3(j))^2, & \text{kun } j = k \\ \sqrt{C_{a,t_i,j,j}C_{a,t_i,k,k}}e^{-1/2(d_{j,k}/L)^2}, & \text{kun } j \neq k \text{ ja } e^{-1/2(d_{j,k}/L_{j,k})^2} \geq \delta \\ 0, & \text{muulloin.} \end{cases}$$

Kaavassa  $d_{j,k}$  on mallipisteiden  $j$  ja  $k$  välinen etäisyys.

Analyysin suoritamme vuorokauden välein kello 12 UTC-aikaa. Ja käytämme havaintoina analyysihetkestä 12 tuntia ennen ja jälkeen tehtyjä havaintoja. Välttääksemme mahdollisia ongelmia otsonin yö-päivä-vaihtelun kanssa suoritamme assimiloinnin vain 17. kerrokseen eli 50,4 kilometriin asti. GOMOS-mittauksien käyttämiselle asetimme hyvin konservatiiviset ehdot eli kaikista havainnoista valitsimme vain ne havainnot, jossa seurattava tähti oli ollut riittävän kirkas. Assimilaatiossa käytettiin 1025:tä otsoniprofilia.

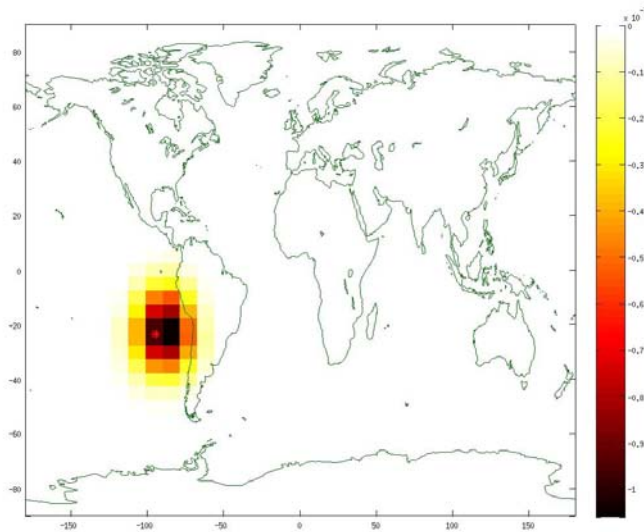
### 5.3 Tuloksia ja johtopäätöksiä

Teimme aluksi kuukauden kontrolliajon mallilla ilman data-assimilaatiota lähtien päivämäärästä 1.1.2004. Malliajon tulos kaikilta mallikerroksilta laskettuna totaaliotsonina Dobsonin yksikössä on esitetty kuvassa 5.4. Tämän jälkeen suoritimme saman ajon uudestaan käyttäen edellisessä kappaleessa esitettyä data-assimilaatiomenetelmää. Kuvassa 5.5 on esitetty ajon tulos. Kuvassa 5.6 on esitetty näiden ajojen erotus ja kuvassa 5.3 käyttämiemme GOMOS-havaintojen paikat.

Havaitsemme, että esimerkiksi Etelämantereella, josta havaintoja ei ole laisinkaan, niin ero assimiloitun ja assimiloimattoman totaaliotsonin välillä on pieni (alle 5 prosenttia). Kuitenkin niillä eteläisen pallonpuoliskon alueilla joista havaintoja on paljon erot tulosten välillä näkyvät selvästi. Pohjoisella pallonpuoliskolla, myös alueilla joissa mittauksia ei ole, havaitsemme assimiloitusta ajosta selvästi enemmän otsonia kuin ilman assimilaatiota tehdystä ajosta. Tämä selittyyneen mallin talviajan dynamiikalla pohjoisella pallonpuoliskolla sekä isolla korrelaatiopituudella. Kuvasta 5.7 havaitsemme, että ajojen suhteellinen ero on varsin pieni lukuunottamatta yhtä Sipeariassa olevaa aluetta.

Johtuen taustavirheen kovarianssimatriisin muodon pysymisenä vakiona ja approksimaation karkeudesta analyysimme alkaa painottua havaintoihin. Näin ollen GOMOS-datan laatu onkin tärkeä tekijä määriteltäessä tulosten laatua. Yksittäisen GOMOS-havainnon vaikutusala on esitetty kuvassa 5.2.

On selvää, että tekemämme yksinkertaistukset assimilatio-ongelmaan ovat varsin karkeita. Erityisesti se, että suoritamme assimilatio-kerroksit-

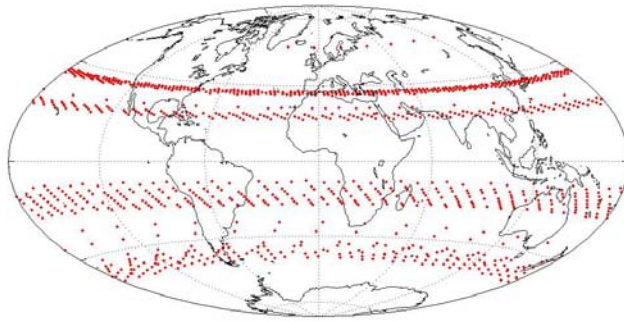


Kuva 5.2: Yksittäisen GOMOS-havainnon vaikutus assimilaatiosysteemissä viidennellä kerroksella.

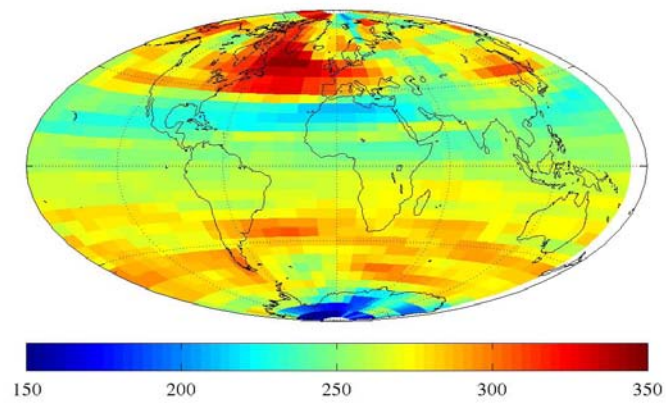
tain aiheuttaa ongelmia, sillä mallissa esiintyy myös kuljetusta korkeussuunnassa, eikä myöskään GOMOS-mittauksissa olevien virheiden voida olettaa olevan täysin riippumattomia eri kerroksilla. Laskennallisesti seuraava askel olisikin priorivirheen kovarianssimatriisin laskeminen tavalla, joka huomioi mallin vaikutuksen. Käytännössä tämä vaatisi ensemble-menetelmän tai liitokoodin implementoimisen. Myöskään vuorokauden kestävä analyysiväli ei ole ideaali johtuen muun muassa otsonin yö-päivä-vaihtelusta ylemmillä kerroksilla.

Vaikka käyttämämme assimilaatiomenetelmä onkin varsin yksinkertainen, niin se antaa kuitenkin tekniikan johdonmukaisiin tarkasteluihin. Sen avulla on muun muassa mahdollista levittää havainnoista saatua informaatiota alueille, joissa mittauksia on vähän, ja pystymme hyödyntämään myös niitä profileja joiden laatu ei ole hyvä. Lisäksi systeemiin on helppo lisätä myös muista instrumenteista saatua otsonia, jolla voidaan parantaa analyysin laatua.

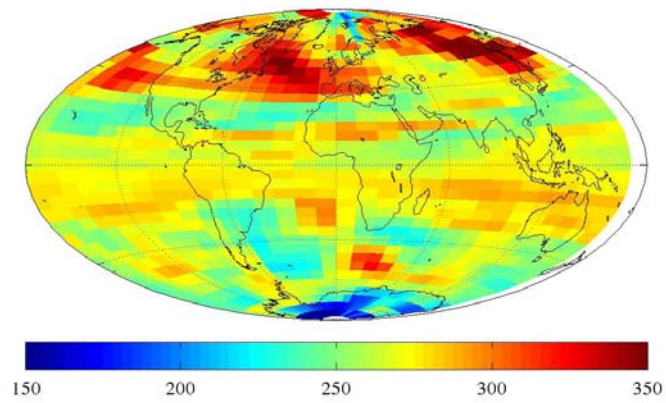
Tulevaisuudessa tarkoituksena on lisätä data-assimilaatiosysteemiin myös OSIRIS-instrumentista hankittuja otsoniprofiileja ja kiinnittää enemmän huomiota validointiin. Lisäksi tavoitteena on siirtyä käyttämään Ilmatieteen laitoksessa kehitettyä FinROSE-mallia, joka käyttää dynamiikan tuottamiseen Euroopan keskuksen (ECMWF) tuulitietoja ja siten vastaa paremmin todellista dynamiikkaa.



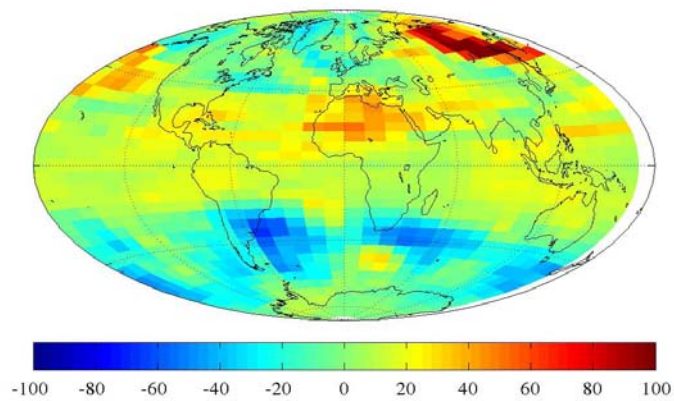
Kuva 5.3: Data-assimilaatiossa käytetyt GOMOS-havainnot.



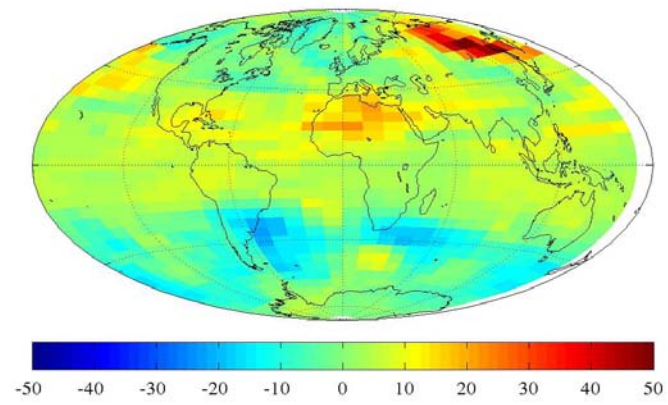
Kuva 5.4: Mallin antama totaaliotsoni Dobsonin yksiköissä 1.2.2004.



Kuva 5.5: Data-assimilaatiosysteemistä saatu totaaliotsoni Dobsonin yksiköissä 1.2.2004.



Kuva 5.6: Mallista saadun ja data-assimilaatiosysteemistä saadun totaaliotsonin erotus Dobsonin yksikössä.



Kuva 5.7: Mallista saadun ja data-assimilaatiosysteemistä saadun totaaliotsonin suhteellinen ero.

# Kirjallisuutta

- [1] Luis Antonio Aguirre, Bruno Otávio S. Teixeira, ja Leonardo Antônio B. Tôrres. Using data-driven discrete-time models and the unscented Kalman filter to estimate unobserved variables of nonlinear systems. *Phys. Rev. E*, 72(2):026226, 2005.
- [2] H. Auvinen, J. M. Bardsley, H. Haario, ja T. Kauranne. The variational Kalman filtering and an efficient implementation using limited memory BFGS. *International Journal for Numerical Methods in Fluids*, 2009.
- [3] Harri Auvinen. Adaptiivinen Markov chain Monte Carlo–menetelmä korkeaulotteisissa parametriavaruuksissa. Pro gradu, Helsingin yliopisto, 2000.
- [4] F. Bouttier ja P. Courtier. Data assimilation concepts and methods, 1999. ECMWF. [http://www.ecmwf.int/newsevents/training/rcourse\\_notes/DATA\\_ASSIMILATION/ASSIM\\_CONCEPTS/Assim\\_concepts.html](http://www.ecmwf.int/newsevents/training/rcourse_notes/DATA_ASSIMILATION/ASSIM_CONCEPTS/Assim_concepts.html).
- [5] Stephen E. Cohn, N. S. Sivakumaran, ja Ricardo Todling. A fixed-lag Kalman smoother for retrospective data assimilation. *Monthly Weather Review*, 122, 1994.
- [6] Roger Daley. *Atmospheric Data Analysis*. Cambridge University Press, 1993.
- [7] Geir Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.
- [8] Elana J. Fertig, John Harlim, ja Brian R. Hunt. A comparative study of 4D-Var and a 4D ensemble Kalman Filter: perfect model simulations with Lorenz-96. *Tellus*, 2007.
- [9] Ralf Giering ja Thomas Kaminski. Recipes for adjoint code construction. *ACM Trans. Math. Softw.*, 24(4):437–474, 1998.

- [10] Brian R. Hunt, Eric J. Kostelich, ja Istvan Szunyogh. Efficient Data Assimilation for Spatiotemporal Chaos: a Local Ensemble Transform Kalman Filter. *Physica D: Nonlinear Phenomena*, 2006.
- [11] K. Ide, P. Courtier, M. Ghil, ja A. C. Lorenc. Unified notation for data assimilation: Operational, sequential and variational. *Journal of the Meteorological Society of Japan*, 75(1B):181–189, 1997.
- [12] Heikki Järvinen. A direct conversion of model algorithms to their adjoints exemplified with barotropic atmospheric model, 1993. Lisensiaattityö.
- [13] C. T. Kelley. *Iterative Methods for Optimization*. The Society for Industrial and Applied Mathematics, 1999.
- [14] Boris V. Khatatov, Jean-Francois Lamarque, Lawrence V. Lyjak, Richard Menard, Pieter Levelt, Xuexi Tie, Guy P. Brasseur, ja John C. Gille. Assimilation of satellite observations of long-lived chemical species in global chemistry transport models. *Journal of Geophysical Research*, 2000.
- [15] E. Kyrölä, J. Tamminen, G. W. Leppelmeier, V. Sofieva, S. Hassinen, J. L. Bertaux, A. Hauchecorne, F. Dalaudier, C. Cot, O. Korablev, O. Fanton d’Andon, G. Barrot, A. Mangin, B. Théodore, M. Guirlet, F. Etanchaud, P. Snoeij, R. Koopman, L. Saavedra, R. Fraisse, D. Fussen, ja F. Vanhellemont. GOMOS on Envisat: An overview. *Advances in Space Research*, 33:1020–1028, 2004.
- [16] Antti A. I. Lange. Fast kalman processing of the gps carrier-phases for mobile positioning and atmospheric tomography, 2009. [http://www.fig.net/pub/fig2009/papers/ts06c/ts06c\\_lange\\_3497.pdf](http://www.fig.net/pub/fig2009/papers/ts06c/ts06c_lange_3497.pdf).
- [17] Martin Leutbecher. A data assimilation tutorial based on the Lorenz-95 system, 2008. ECMWF. [http://www.ecmwf.int/newsevents/training/meteorological\\_presentations/pdf/DA/ToyModel\\_notes.pdf](http://www.ecmwf.int/newsevents/training/meteorological_presentations/pdf/DA/ToyModel_notes.pdf).
- [18] P. F. Levelt, B. V. Khatatov, J. C. Gille, G. P. Brassaeur, X. X. Tie, ja J. W. Waters. Assimilation of MLS ozone measurements in the global three-dimensional chemistry transport model ROSE. *Geophysical Research Letters*, 25(24):4493–4496, 1998.
- [19] Jorge Nocedal ja Stephen J. Wright. *Numerical Optimaziation*. Springer, 2006.



- [20] Alan O’Neil. *Remote Sensing and Climate Change: The Role of Earth Observation*, chapter Atmospheric data assimilation, ss. 97–106. Springer–Praxis, 2001.
- [21] S. E. Cohn R. Todling ja N. S. Sivakumaran. Suboptimal schemes for retrospective data assimilation based on the fixed-lag Kalman smoother. *Monthly Weather Review*, 126, 1997.
- [22] Clive D. Rodgers. *Inverse Methods For Atmospheric Sounding: Theory and Practice*. World Scientific, 2000.
- [23] Klaus Rose ja Guy Brasseur. A three-dimensional model of chemically active trace species in the middle atmosphere during disturbed winter conditions. *Journal of Geophysical Research*, 94, 1989.
- [24] Jonathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Tekninen raportti, Pittsburgh, PA, USA, 1994.
- [25] A. Sitz, U. Schwarz, ja Jürgen Kurths. The unscented Kalman filter, a powerful tool for data analysis. *I. J. Bifurcation and Chaos*, 14(6):2093–2105, 2004.
- [26] Anne K. Smith ja Daniel Marsh. *Description of the ROSE Model*. National Center for Atmospheric Research, 2006.



# Liite A

## Käsitteitä ja määritelmiä

Olkoon  $y_i \in \mathbb{R}$ ,  $i = 1, \dots, p$  tehdyt havainnot ja  $f: \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$  ilmiötä kuvaava malli. Vektori  $x \in \mathbb{R}^n$  on tunnettu mallin tilamuuttuja. Tällöin tilannetta kuvaa yhtälö

$$y_i = f(x) + \varepsilon_i$$

missä  $\varepsilon_i$  on havaintovirhe. Estimaattia  $x_{est} = \min_{x \in \Omega} S(x)$ , joka minimoi neliösumman

$$S(x) = \sum_{i=1}^p (y_i - f(x))^2$$

kutsutaan *pienimmän neliösumman estimaatiksi* (eng. *least squares estimate*). Virhetermi  $\varepsilon$  oletetaan usein satunnaismuuttujaksi ja sen statistisia ominaisuuksia käsittelemme todennäköisyysjakaumien avulla. Usein oletamme, että virhe on normaalisti eli gaussisesti jakautunut. Satunnaismuuttuja  $\varepsilon \in \mathbb{R}^n$  on normaalijakautunut (merkitään  $\varepsilon \sim N(\mu, C)$ ), jos sillä on tiheysfunktio

$$p(\varepsilon) = \frac{1}{(2\pi)^{n/2} \det(C)^{1/2}} \exp\left(-\frac{1}{2}(\varepsilon - \mu)^T C^{-1}(\varepsilon - \mu)\right),$$

missä vektori  $\mu \in \mathbb{R}^n$  on satunnaismuuttujan  $\varepsilon$  odotusarvo  $E(\varepsilon)$  ja  $C \in \mathbb{R}^{n \times n}$  sen kovarianssimatriisi. Kovarianssimatriisi määritellään kaavalla

$$\text{cov}(\varepsilon) = E((\varepsilon - \mu)(\varepsilon - \mu)^T).$$

Auki kirjoitettuna kovarianssimatriisi näyttää seuraavalta

$$\text{cov}(\varepsilon) = \begin{pmatrix} E((\varepsilon_1 - \mu_1)(\varepsilon_1 - \mu_1)) & \cdots & E((\varepsilon_1 - \mu_1)(\varepsilon_n - \mu_n)) \\ \vdots & \ddots & \vdots \\ E((\varepsilon_n - \mu_n)(\varepsilon_1 - \mu_1)) & \cdots & E((\varepsilon_n - \mu_n)(\varepsilon_n - \mu_n)) \end{pmatrix}.$$

Huomaamme, että kovarianssimatriisi on aina symmetrinen. Sanomme, että kovarianssimatriisi  $C$  on positiivisesti definiitti, jos

$$\gamma^T C \gamma > 0$$

kaikilla  $\gamma \in \mathbb{R}^n$ .

Satunnaismuuttujien  $\varepsilon$  ja  $\gamma$  välinen ristikovarianssimatriisi määritellään kaavalla

$$\text{cov}(\varepsilon, \gamma) = E((\varepsilon - E(\varepsilon))(\gamma - E(\gamma))^T).$$

Sanomme, että satunnaismuuttujat ovat *korreloimattomia* (eng. *uncorrelated*), jos niiden välinen ristikovarianssi on nollamatriisi.

Mittausarjasta  $y^i \in \mathbb{R}^p$ ,  $k = 1, \dots, N$  voimme laskea otoskeskiarvon

$$\text{mean}(Y) = \bar{Y} = \frac{1}{N} \sum_{k=1}^N y^k$$

ja otoskovarianssimatriisin  $S$

$$S_{ij} = \frac{1}{N-1} \sum_{k=1}^N (y_i^k - \bar{y}_i)(y_j^k - \bar{y}_j).$$

Funktion  $f: \mathbb{R}^n \rightarrow \mathbb{R}^p$  Jacobin matriisi määritellään kaavalla

$$\text{Jacob}(f(x)) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_p(x)}{\partial x_1} & \dots & \frac{\partial f_p(x)}{\partial x_n} \end{pmatrix}.$$

Funktion  $g: \mathbb{R}^n \rightarrow \mathbb{R}$  gradientti määritellään kaavalla

$$\nabla g(x) = \left( \frac{\partial g(x)}{\partial x_1}, \dots, \frac{\partial g(x)}{\partial x_n} \right)^T$$

ja Hessen matriisi kaavalla

$$\text{Hess}(g(x)) = \begin{pmatrix} \frac{\partial^2 g(x)}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 g(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 g(x)}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 g(x)}{\partial x_n \partial x_n} \end{pmatrix}.$$

Funktion  $h: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}$  matriisiderivaattaa määritellään kaavalla

$$\frac{\partial h(X)}{\partial X} = \begin{pmatrix} \frac{\partial h(X)}{\partial X_{1,1}} & \dots & \frac{\partial h(X)}{\partial X_{n,1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h(X)}{\partial X_{1,m}} & \dots & \frac{\partial h(X)}{\partial X_{n,m}} \end{pmatrix}.$$

Vektorin  $\gamma$  euklidista normia merkitsemme  $\|\gamma\|$ . Sen voi laskea kaavalla

$$\|\gamma\| = \sqrt{\gamma_1^2 + \cdots + \gamma_n^2}.$$