

Otsoniaineiston analysointi lineaarisella tila-avaruusmallilla

Niilo Latva-Pukkila



Tilastotieteen pro gradu -tutkielma

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
6. tammikuuta 2014

Tiivistelmä: Niilo Latva-Pukkila: *Otsoniaineiston analysointi lineaarisella tila-avaruusmallilla* (engl. *Gaussian state space models with an application in ozone modelling*), tilastotieteen pro gradu -tutkielma, 44 s. + liitteitä 16 s., Jyväskylän yliopisto, Matematiikan ja tilastotieteen laitos, 6. tammikuuta 2014.

Tutkielma käsittelee yläilmakehän otsonimäärän mallintamista lineaarisella tila-avaruusmallilla. Ilmakehän otsonimäärä vaihtelee vuodenajan mukaan ja lisäksi tunnetaan joitakin luonnollisia tekijöitä, jotka vaikuttavat otsonimäärään. Tämän lisäksi erilaiset ihmisen toiminnan vuoksi ilmakehässä lisääntyneet aineet aiheuttavat muutoksia otsonimäärässä.

Aiemmin otsonimäärää on mallinnettu yleistettyjen lineaaristen mallien avulla. Tila-avaruusmallit ovat kuitenkin huomattavasti monipuolisempia malleja, joilla voidaan hyvin mallintaa ajassa tapahtuvaa kehitystä. Mallin sovittamisessa käytetään modernia MCMC-algoritmia.

Aineistona käytetään kahden eri satelliitti-instrumentin havaintoja otsonimäärästä. Nämä kaksi aikasarjaa on yhdistetty yhdeksi aikasarjaksi, joka kattaa aikavälin 1984–2011.

Kun tunnettujen luonnollisten tekijöiden vaikutus on huomioitu, arvioidaan otsonimäärän laskeneen noin vuoteen 1998 asti. Tällöin lasku loppui ja määrän kehityksen estimoidaan kääntyneen jopa positiiviseksi. Muutokset ovat kuitenkin varsin pieniä ja lisätutkimusta tarvitaan, ennen kuin voidaan puhua otsonimäärän palautumisesta.

Avainsanat: aikasarja-analyysi, tila-avaruusmalli, viivästetyn hylkäyksen adaptiivinen Metropolis-algoritmi, otsonikato, ilmakehä

Kansilehden kuva: ENVISAT-satelliitti auringon valon taittuessa ilmakehän läpi kulkiessaan. (Kuvan lähde: ESA)

Kiitokset

Haluan kiittää Ilmatieteen laitoksen Ilmakehän kaukokartoitus -ryhmää kiehtovan ja haastavan aiheen ja aineiston antamisesta käyttöni sekä kesäharjoittelujaksojen tarjoamisesta. Erityiskiitokset FT Marko Laineelle arvokkaista ideoista ja palautteesta sekä inspiroivista keskusteluista. Lisäksi haluan kiittää työni ohjaajaa professori Jukka Nyblomia neuvoista ja palautteesta tämän prosessin aikana. Kiitokset myös Annalle ja Juholle tekstin oikolukemisesta.

Helsingissä loppiaisenä 6.1.2014
Niilo Latva-Pukkila

Sisältö

Johdanto	1
Merkinnöistä	2
Luku 1. Tila-avaruusmallien teoriaa	3
1.1. Dynaaminen lineaarinen regressio	3
1.2. Mallin määrittäminen	4
1.2.1. Trendin mallintaminen	4
1.2.2. Taustamuuttujat	5
1.2.3. Kausivaihtelu	5
1.2.4. Mallikomponenttien yhdistäminen	7
1.3. Tilojen estimointi ja ennustaminen	8
1.3.1. Kalmanin suodin	9
1.3.2. Tilojen tasoitus	10
1.3.3. Ennustaminen	10
1.3.4. Puuttuvan tiedon käsittely	11
1.3.5. Alustus	11
1.4. Tilojen simulointi	11
1.5. Tuntemattomien parametrien estimointi	12
Luku 2. Metropolis-algoritmin laajennuksia	14
2.1. Metropolis-algoritmi	15
2.2. AM-algoritmi	16
2.3. DR-algoritmi	17
2.4. DRAM-algoritmi	18
2.5. DRAM-algoritmin sovellus tila-avaruusmalleihin	19
Luku 3. Otsonimäärän mallintaminen tila-avaruusmallilla	21
3.1. Otsoniaineisto	21
3.2. Taustamuuttujat	23
3.3. Otsonimalli	25
3.4. Tulokset	27
3.5. Mallidiagnostiikka	34
Luku 4. Pohdintaa	39
Lähdeluettelo	41
Liite A. R-koodi	45

Johdanto

Otsoni on kolmesta happiatomista muodostuva molekyyli, jota esiintyy luonnollisesti ilmakehässä. Otsoniin viitataan usein sen kemiallisella kaavalla O_3 . Noin 90 prosenttia ilmakehän otsonista sijaitsee stratosfäärissä, eli yläilmakehässä, joka on noin 15-80 kilometrin korkeudella maan pinnasta. Loput 10 prosenttia sijaitsee ilmakehän alimmassa kerroksessa, troposfäärissä, joka ulottuu maan pinnan tasosta stratosfäärin alarajaan. Niitä stratosfäärin alueita joissa otsonin esiintyminen on kaikkein suurinta, kutsutaan yleisesti otsonikerrokseksi. (World Meteorological Organization, 2011, s. 4)

Otsonikerros torjuu Auringosta peräisin olevaa haitallista B-tyypin ultraviolettisäteilyä (UVB). UVB-säteilyn tiedetään lisäävän ihmisten ihosyövän riskiä ja heikentävän ihmisten immuunijärjestelmää (Matsumura & Ananthaswamy, 2004). Otsonimäärän tiedetään vähentyneen vuosien 1979 ja 1997 välillä, mutta useissa tutkimuksissa on havaittu viitteitä siitä, että otsonimäärän kehityksessä on tapahtunut muutos noin vuonna 1997 (Newchurch et al., 2003; Jones et al., 2009; Kyrölä et al., 2013). Arveltu syy tälle muutokselle on vuonna 1987 voimaan astunut Montrealin pöytäkirja, jolla pyrittiin rajoittamaan otsonikatoa aiheuttavien aineiden, erityisesti kloorin ja bromin, joutumista ilmakehään (Kyrölä et al., 2013, s. 10645–10646). Tämän hetken tiedon perusteella uskotaan, että otsonikatoa aiheuttavien aineiden määrä ilmakehässä oli huipussaan vuosien 1995 ja 2000 välillä, mutta ennusteiden perusteella näiden aineiden määrä laskee vuotta 1980 edeltäneelle tasolle vasta vuosien 2050 ja 2060 välillä (Jones et al., 2009, s. 6055–6056).

Otsonimäärän tiedetään muuttuvan luonnollisesti Auringon aktiivisuuden ja Singaporen tuulen vaikutuksesta (Angell & Korshover, 1964; Angell, 1989). Näiden vaikutus halutaan yleensä huomioida mallinnusprosessin aikana, sillä muutoin nämä vaikuttavat tarpeettomasti trendiin.

Tämän tutkielman tavoittena on satelliittihavaintoja käyttämällä selvittää, miten otsonimäärä on kehittynyt ilmakehässä leveyspiirien 10 astetta pohjoista leveyttä ($10^\circ N$) ja 20 astetta pohjoista leveyttä ($20^\circ N$) välissä korkeusalueella 35.5 – 45 km vuosien 1984 ja 2011 välillä, kun on otettu huomioon tunnetut luonnolliset tekijät. Tätä mallinnetaan tila-avaruusmalleilla, jotka ovat suosittuja aikasarja-analyysin menetelmiä.

Erityisen hyödyllisiä tila-avaruusmallit ovat tilanteissa, joissa malli voidaan jakaa osiin, kuten trendiin, kausivaihteluun ja regressiokomponentteihin. Laskennallisesti tehokkailla menetelmillä, kuten Kalmanin suotimella ja tasoituksella, voidaan laskea tehokkaasti mallin eri osien vaikutukset (Durbin & Koopman, 2012, s. 1–2). Toisinaan mallin tuntemattomien parametrien arviointi ei kuitenkaan ole helppoa. Vaikka Kalmanin suotimella voidaan laskea tilat helposti, saattaa malli sisältää muita tuntemattomia parametreja. Näiden laskemiseen voidaan käyttää viime aikoina hyvin suosituksi nousseita MCMC-pohjaisia simulointimenetelmiä (Särkkä, 2013, luku 12).

Ensimmäisessä luvussa käydään läpi yleistä tila-avaruusmallien teoriaa. Siinä näytetään, kuinka tila-avaruusmalleja voidaan käyttää esittämään useita erilaisia aikasarja-analyysin malleja ja esitetään menetelmiä tilojen ja mallin määrittämisessä mahdollisesti olevien tuntemattomien parametrien estimointiin. Toisessa luvussa esitellään kaksi Metropolis-algoritmin laajennusta sekä niiden yhdistelmä ja näytetään, kuinka niitä voidaan hyödyntää tila-avaruusmallien sovittamisessa. Kolmannessa luvussa esitellään käytössä oleva aineisto ja näytetään, kuinka kahdessa ensimmäisessä luvussa esiteltyjä menetelmiä voidaan käyttää otsonimäärän analysointiin. Lopuksi esitellään keskeisimmät tulokset ja tarkastellaan mallin sopivuutta.

Merkinnöistä

Tässä tutkielmassa käytetään toistuvasti seuraavia merkintöjä. Muut merkinnät esitellään asiayhteydessä.

- Y_t on vastemuuttuja hetkellä t ja y_t vastaava tehty havainto
- $y_{1:t}$ tarkoittaa kaikkia havaintoja ajanhetkien 1 ja t välillä
- T on viimeisimmän havainnon ajanhetki
- A^{-1} on matriisin A käänteismatriisi
- A^T on matriisin A transpoosi
- $\text{diag}(x)$ tarkoittaa diagonaalimatriisia, jonka diagonaalilla on vektorin x alkiot
- $\text{blockdiag}(A_1, \dots, A_n)$ tarkoittaa lohkodeagonaalista matriisia, jonka lohkoina ovat matriisit A_1, \dots, A_n
- $\mathbb{E}[X]$ on satunnaismuuttujan X odotusarvo
- $\text{Var}(X)$ on satunnaismuuttujan X varianssi
- $X|Y$ on satunnaismuuttuja X ehdolla Y
- \mathcal{N}_p on p -ulotteinen normaalijakauma
- \mathbb{R}^p on p -ulotteinen euklidinen avaruus
- Jakaumien yhteydessä isot kirjaimet (esim. P, Q) viittaavat jakaumiin ja pienet kirjaimet (esim. p, q) niiden tiheys- tai pistetodennäköisyysfunktioihin

LUKU 1

Tila-avaruusmallien teoriaa

Tässä tila-avaruusmallien teoriaa käsittelevässä luvussa esitellään tila-avaruusmallien peruskäsitteitä ja näytetään, kuinka muutamia erilaisia aikasarja-analyysin malleja voidaan esittää tila-avaruusmalleina. Tämän luvun pääasiallinen lähde on kirja *Dynamic Linear Models with R* (Petrus et al., 2009).

1.1. Dynaaminen lineaarinen regressio

Dynaaminen lineaarinen regressio on tila-avaruusmallien erikoistapaus, jossa tilojen kehitys ja tilojen kuvautuminen havainnoiksi oletetaan lineaarisiksi ja kaikki satunnaisvaihtelu oletetaan normaalijakautuneeksi. Yksiulotteinen dynaaminen lineaarinen malli voidaan kirjoittaa

$$(1.1) \quad Y_t = F_t \theta_t + v_t, \quad v_t \sim \mathcal{N}_1(0, V_t),$$

$$(1.2) \quad \theta_t = G_t \theta_{t-1} + w_t, \quad w_t \sim \mathcal{N}_p(0, W_t),$$

$t = 1, \dots, T$. Satunnaismuuttujat v_t ja w_t oletetaan keskenään riippumattomiksi.

Yhtälöllä (1.1) kuvataan sitä, miten tilat kuvautuvat havainnoiksi ja yhtälöllä (1.2) sitä, miten tilat kehittyvät ajan suhteen. Yhtälöä (1.1) kutsutaan havaintoyhtälöksi ja yhtälöä (1.2) tilayhtälöksi. Mallissa esiintyvien matriisien ja vektoreiden koot esitetään taulukossa 1.1.

TAULUKKO 1.1. Dynaamisen lineaarisen regression matriisien ja vektorien koot ja tulkinnat

Merkintä	koko	tarkoitus
Y_t	skalaari	vastemuuttuja hetkellä t
θ_t	$p \times 1$	tilavektori
F_t	$1 \times p$	liittää tilavektorin havaintoon
G_t	$p \times p$	määrittää tilojen kehittymisen
V_t	skalaari	satunnaismuuttujan v_t varianssi
W_t	$p \times p$	tilojen kehityksen kovarianssimatriisi

1.2. Mallin määrittäminen

Tila-avaruusmallit rakentuvat erilaisista mallikomponenteista. Näitä ovat muun muassa trendi, taustamuuttujat ja kausivaihtelu. Eri malleja voidaan myös yhdistellä.

1.2.1. Trendin mallintaminen. Aikasarjan trendillä tarkoitetaan aikasarjan keskimääräisen tason muutosta pitkällä aikavälillä. Se, mitä pitkällä aikavälillä tarkoitetaan, riippuu tietysti aikasarjan pituudesta. Lyhyissä aikasarjoissa trendiltä näyttävä ilmiö voi olla osa pitkäjaksoista syklistä vaihtelua, mutta lyhyen aikasarjan osalta sitä on kuitenkin usein järkevää mallintaa trendinä. (Chatfield, 2003, s. 12)

Kaikkein yksinkertaisimmillaan aikasarjan tasoa voidaan mallintaa lokaalilla tasomallilla. Määritellään $G = 1$ ja $F = 1$, jonka seurauksena yhtälöt (1.1) ja (1.2) yksinkertaistuvat muotoon

$$\begin{aligned} Y_t &= \mu_t + v_t, \quad v_t \sim \mathcal{N}_1(0, V_t), \\ \mu_t &= \mu_{t-1} + w_{t,\mu}, \quad w_{t,\mu} \sim \mathcal{N}_1(0, \sigma_\mu^2). \end{aligned}$$

Tässä tilavektori on normaalisti jakautunut skalaariarvoinen satunnaiskulku. Mikäli $\sigma_\mu^2 = 0$, ei tila muutu ajan suhteen lainkaan, eli $\mu_t = \mu$ kaikilla $t = 1, \dots, T$. Parametri μ kuvaa koko aikasarjan keskimääräistä tasoa. Tällaista mallia kutsutaan deterministiseksi lokaaliksi tasomalliksi.

Mallia voidaan laajentaa sisällyttämällä siihen trendikomponentti. Tällaisessa mallissa tilavektori on kaksiulotteinen. Malli voidaan kirjoittaa

$$\begin{aligned} Y_t &= \mu_t + v_t, \quad v_t \sim \mathcal{N}_1(0, V_t), \\ \mu_t &= \mu_{t-1} + \alpha_{t-1} + w_{t,\mu}, \quad w_{t,\mu} \sim \mathcal{N}_1(0, \sigma_\mu^2), \\ \alpha_t &= \alpha_{t-1} + w_{t,\alpha}, \quad w_{t,\alpha} \sim \mathcal{N}_1(0, \sigma_\alpha^2). \end{aligned}$$

Tässä μ_t kuvaa aikasarjan tasoa hetkellä t ja α_t kulmakerrointa hetkellä t . Havainnot muodostuvat pelkästään ensimmäisestä tilakomponentista (μ_t) ja kohinasta (v_t). Toinen tilakomponentti ei vaikuta suoraan havaintoihin, mutta ensimmäinen tilakomponentti saadaan molempien tilakomponenttien summana, eli toinen tilakomponentti vaikuttaa ensimmäisen tilakomponentin kehittymiseen. Ensimmäisen tilakomponentin arvo voi siis muuttua kahdella tavalla: suoraan satunnaismuuttujan $w_{t,\mu}$ välityksellä ja sen lisäksi toisen tilakomponentin välityksellä. Mikäli $\sigma_\mu^2 = \sigma_\alpha^2 = 0$, niin kyseessä on normaali lineaarinen regressiomalli, jota voidaan sanoa lokaaliksi lineaariseksi trendimalliksi, jossa on deterministinen taso ja kulmakerroin. Mikäli tilakomponentti α_t on stokastinen ($\sigma_\alpha^2 > 0$), voidaan sitä ajatella regressiokertoimena, jonka arvo voi muuttua ajan suhteen.

Samaan tapaan voidaan kirjoittaa myös yleinen asteen n trendimalli, jossa tilavektori on n -ulotteinen vektori $(\mu_t, \alpha_{t,1}, \dots, \alpha_{t,n-1})^T$. Tässä

ensimmäinen tilakomponentti μ kuvaa aikasarjan tasoa ja tilakomponentti α_j sitä, kuinka tilakomponentti α_{j-1} kehittyy seuraavalla aikavälillä. Yksityiskohtaista toteuttamista varten katso Petris et al. (2009, s. 99–100).

1.2.2. Taustamuuttujat. Usein ollaan kiinnostuneita siitä, miten aikasarjan keskimääräinen taso muuttuu, kun jonkin toisen muuttujan arvo muuttuu. Lisäksi joissain sovelluksissa ollaan kiinnostuneita siitä, millaista trendiä jää jäljelle, kun tunnettujen taustamuuttujien vaikutus on eliminoitu pois. Molempiin kysymyksiin voidaan vastata sisällyttämällä malliin selittäjäksi halutut taustamuuttujat.

Dynaamiseen lineaariseen malliin on helppo sisällyttää taustamuuttujia. Tällainen malli, eli dynaaminen regressiomalli, voidaan kirjoittaa

$$Y_t = \beta_{0,t} + \sum_{i=1}^k \beta_{i,t} x_{i,t} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}_1(0, \sigma^2).$$

Mikäli kaikki parametrit β oletetaan vakioiksi, vastaa tämä normaalia lineaarista regressiomallia. Tila-avaruusmallien merkinnöillä tämä toteutetaan asettamalla vektori $\beta = (\beta_0, \dots, \beta_k)^T$ tilavektoriksi ja F_t vastaamaan taustamuuttujien arvoja. Mikäli tilojen satunnaiskulku sallitaan, voidaan tämä kirjoittaa

$$\begin{aligned} Y_t &= \beta_{0,t} + \sum_{i=1}^k \beta_{i,t} x_{i,t} + v_t, \quad v_t \sim \mathcal{N}_1(0, V_t), \\ \beta_t &= \beta_{t-1} + w_{t,\beta}, \quad w_t \sim \mathcal{N}_{k+1}(0, W_t). \end{aligned}$$

Dynaamista regressiomallia voidaankin ajatella tavallisen lineaarisen regressiomallin yleistyksenä, jossa regressiokertoimet voivat muuttua ajan suhteen.

1.2.3. Kausivaihtelu. Kausivaihtelu on aikasarjassa esiintyvää jaksollista vaihtelua. Kausivaihtelu voi olla esimerkiksi sitä, että aikasarjan taso on järjestelmällisesti erilainen eri vuodenaikoina tai vaikkapa eri vuorokaudenaikoina. (Chatfield, 2003, s. 12)

Kausivaihtelua voidaan mallintaa tila-avaruusmalleilla useilla eri tavoilla. Kaikkein yksinkertaisin tapa toteuttaa tämä on estimoida oma parametri kullekin jakson hetkelle, esimerkiksi kuukausittaisen aineiston tapauksessa jokaiselle kuukaudelle. Tässä tutkielmassa käytetään kuitenkin Fourier-muotoa, jolla on erinäisiä hyviä puolia yksinkertaisempiin menetelmiin verrattuna.

Merkitään kirjaimella s sitä, kuinka monta ajanhetkeä jaksossa on. Mikäli esimerkiksi aineisto on kuukausittaista ja mallinnetaan sitä, miten kukin kuukausi poikkeaa keskimäärin koko vuoden keskiarvosta, niin $s = 12$, koska vuodessa on 12 kuukautta. Vastaavasti jos mittauksia on neljännesvuosittain, niin $s = 4$.

Fourier-menetelmä perustuu siihen, että s -ulotteinen avaruus viritetään s kappaleella vektoreita, jotka muodostuvat trigonometristen funktioiden arvoista niin sanottuilla Fourier-taajuuksilla. Oletetaan ensin, että s on parillinen. Määritellään Fourier-taajuudet

$$\omega_j = \frac{2\pi j}{s}, \quad j = 1, \dots, \frac{s}{2}.$$

Näiden avulla kausivaihtelun vaikutus hetkellä t voidaan kirjoittaa

$$\gamma_t = \sum_{j=1}^{\frac{s}{2}} (a_j \cos(\omega_j t) + b_j \sin(\omega_j t))$$

joillain $a_j, b_j \in \mathbb{R}$ kaikilla $j = 1, \dots, \frac{s}{2}$. Koska s on parillinen, pätee $\omega_{\frac{s}{2}} = \pi$ ja edelleen $\sin(\omega_{\frac{s}{2}} t) = 0$ kaikilla $t = 1, \dots, T$. Näin ollen viimeinen sin-termi voidaan jättää huomioimatta.

Trigonometriset funktiot vaihtavat merkkiään sitä tiheämmän, mitä suurempi j on. Tätä voidaan havainnollistaa piirtämällä trigonometristen funktioiden arvoja eri taajuuksilla (ks. esim. Petris et al., 2009, s. 104).

Mikäli kausivaihtelun halutaan voivan muuttua ajan suhteen, kirjoitetaan kausivaihtelun vaikutus hetkellä t

$$\gamma_t = \sum_{j=1}^{\frac{s}{2}} \zeta_{t,j},$$

jossa parametri ζ_j saadaan päivityskaavojen

$$\zeta_{t,j} = \zeta_{t-1,j} \cos \omega_j + \zeta_{t-1,j}^* \sin \omega_j + w_{t,\zeta_j}, \quad w_{t,\zeta_j} \sim \mathcal{N}_1(0, \sigma_{\zeta_j}^2)$$

ja

$$\zeta_{t,j}^* = -\zeta_{t-1,j} \sin \omega_j + \zeta_{t-1,j}^* \cos \omega_j + w_{t,\zeta_j^*}, \quad w_{t,\zeta_j^*} \sim \mathcal{N}_1(0, \sigma_{\zeta_j^*}^2)$$

avulla kaikille $j = 1, \dots, \frac{s}{2}$. Huomaa, että päivityskaavoissa tarvitaan myös tilat ζ_j^* , jotka eivät ole mukana komponentissa γ_t .

Edellä oletettiin, että s on parillinen. Tämä oletus onkin hyvin usein voimassa, sillä usein kausivaihtelun ajatellaan muodostuvan esim. 12 kuukaudesta tai neljästä kvartaalista. Jos s ei ole parillinen, niin edelliset kaavat vaativat pieniä muutoksia (ks. esim. Petris et al., 2009, s. 108–109).

Usein satunnaiskulkuun liittyvät varianssit oletetaan samaksi kaikille tilakomponenteille. Kuten kirjassa *Forecasting, Structural Time Series Models and the Kalman Filter* (Harvey, 1990, s. 43) todetaan, ei varianssikomponenttien asettaminen samaksi yleensä huononna mallia paljoo, mutta tekee numeerisesta optimoinnista huomattavasti helpompaa.

Monesti kausivaihtelu on luonteeltaan esimerkiksi vain vuosittaista tai puolivuositista. Tämä onkin yksi Fourier-muodon eduista, sillä

Fourier-muodon käyttö antaa mahdollisuuden käyttää vain osaa taajuuksista. Jos vaihtelu on täysin harmonista, aineisto on kuukausittaista ja pisin ajateltu jakson pituus on yksi vuosi, vastaa ensimmäinen taajuus vuosivaihtelua, toinen puolivuosisivaihtelua jne. Jos käytetään kaikkia taajuuksia, niin Fourier-muodon käyttö vastaa tilannetta, jossa kullekin jakson hetkelle estimoidaan oma parametrinsa.

1.2.4. Mallikomponenttien yhdistäminen. Dynaamiset lineaariset mallit koostuvat usein useammasta eri komponentista. Malliin voi kuulua esimerkiksi sekä trendi- että kausivaihtelukomponentit.

Oletetaan, että haluttu malli koostuu S eri komponentista, joiden tilat havainnoiksi kuvaavat matriisit ovat $F_{1,t}, \dots, F_{S,t}$, tilavektorit $\bar{\Theta}_{1,t}, \dots, \bar{\Theta}_{S,t}$, tilojen kehittymistä kuvaavat matriisit $G_{1,t}, \dots, G_{S,t}$ ja tilojen kehittymisen satunnaisuuteen liittyvät kovarianssimatriisit $W_{1,t}, \dots, W_{S,t}$. Nyt olettamalla että eri mallikomponenttien kehitys ja kehitykseen liittyvä satunnaisuus ovat riippumattomia, koko mallin määrittävät matriisit saadaan kirjoittamalla

$$\theta_t = (\bar{\Theta}_{1,t}, \dots, \bar{\Theta}_{S,t})^T,$$

$$F_t = (F_{1,t}, \dots, F_{S,t}),$$

$$G_t = \text{blockdiag}(G_{1,t}, \dots, G_{S,t})$$

ja

$$W_t = \text{blockdiag}(W_{1,t}, \dots, W_{S,t}).$$

Edellä esitettiin, kuinka voidaan mallintaa trendiä, taustamuuttujien vaikutusta ja kausivaihtelua. Jos halutaan tehdä malli, jossa esiintyy nämä kaikki, voidaan nämä yhdistää samaan malliin. Esimerkiksi havaintoyhtälö malliin, joka sisältää lineaarisen trendin, jossa on stokastinen kulmakerroin, kolme taustamuuttujaa ja kuukausittainen kausivaihtelu, jossa on yhteinen varianssiparametri, voidaan kirjoittaa

$$\begin{aligned} Y_t &= \mu_t + \beta_{t,1}X_1(t) + \beta_{t,2}X_2(t) + \beta_{t,3}X_3(t) \\ &\quad + \zeta_{t,1} + \zeta_{t,2} + \zeta_{t,3} + \zeta_{t,4} + \zeta_{t,5} + \zeta_{t,6} + v_t, \quad v_t \sim \mathcal{N}_1(0, \sigma_\epsilon^2), \end{aligned}$$

ja tilayhtälöt

$$\mu_t = \mu_{t-1} + \alpha_{t-1}$$

$$\alpha_t = \alpha_{t-1} + w_{t,\alpha}, \quad w_{t,\alpha} \sim \mathcal{N}_1(0, \sigma_\alpha^2)$$

$$\beta_{t,1} = \beta_{t-1,1} + w_{t,\beta_1}, \quad w_{t,\beta_1} \sim \mathcal{N}_1(0, \sigma_{\beta_1}^2)$$

$$\beta_{t,2} = \beta_{t-1,2} + w_{t,\beta_2}, \quad w_{t,\beta_2} \sim \mathcal{N}_1(0, \sigma_{\beta_2}^2)$$

$$\beta_{t,3} = \beta_{t-1,3} + w_{t,\beta_3}, \quad w_{t,\beta_3} \sim \mathcal{N}_1(0, \sigma_{\beta_3}^2)$$

$$\zeta_{t,i} = \zeta_{t-1,i} \cos\left(\frac{\pi i}{6}\right) + \zeta_{t-1,i}^* \sin\left(\frac{\pi i}{6}\right) + w_{t,\zeta_i}, \quad w_{t,\zeta_i} \sim \mathcal{N}_1(0, \sigma_\zeta^2)$$

$$\zeta_{t,i}^* = -\zeta_{t-1,i} \sin\left(\frac{\pi i}{6}\right) + \zeta_{t-1,i}^* \cos\left(\frac{\pi i}{6}\right) + w_{t,\zeta_i^*}, \quad w_{t,\zeta_i^*} \sim \mathcal{N}_1(0, \sigma_\zeta^2)$$

$$\zeta_{t,6} = -\zeta_{t-1,6} + w_{t,\zeta_6}, \quad w_{t,\zeta_6} \sim \mathcal{N}_1(0, \sigma_\zeta^2),$$

$i = 1, \dots, 5$.

Merkitsemällä kunkin mallin komponentin tilojen kehittymisen määrittäviä matriiseja

$$G_{\text{trendi}} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad G_{\text{reg}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$G_{\text{kausi},i} = \begin{pmatrix} \cos\left(\frac{\pi i}{6}\right) & \sin\left(\frac{\pi i}{6}\right) \\ -\sin\left(\frac{\pi i}{6}\right) & \cos\left(\frac{\pi i}{6}\right) \end{pmatrix},$$

$i = 1, \dots, 5$ ja

$$G_{\text{kausi},6} = (-1),$$

koko malli voidaan esittää matriisimuodossa kirjoittamalla

$$\begin{aligned} \theta_t &= (\mu_t, \alpha_t, \beta_{t,1}, \beta_{t,2}, \beta_{t,3}, \\ &\quad \zeta_{t,1}, \zeta_{t,1}^*, \zeta_{t,2}, \zeta_{t,2}^*, \zeta_{t,3}, \zeta_{t,3}^*, \zeta_{t,4}, \zeta_{t,4}^*, \zeta_{t,5}, \zeta_{t,5}^*, \zeta_{t,6})^T, \\ G &= \text{blockdiag}(G_{\text{trendi}}, G_{\text{reg}}, \\ &\quad G_{\text{kausi},1}, G_{\text{kausi},2}, G_{\text{kausi},3}, G_{\text{kausi},4}, G_{\text{kausi},5}, G_{\text{kausi},6}), \\ F_t &= (1, 0, \mathbf{X}_1(t), \mathbf{X}_2(t), \mathbf{X}_3(t), \\ &\quad 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1), \\ W &= \text{diag}(0, \sigma_\alpha^2, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \sigma_{\beta_3}^2, \\ &\quad \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2, \sigma_\zeta^2). \end{aligned}$$

1.3. Tilojen estimointi ja ennustaminen

Tilojen estimoinnilla tarkoitetaan tilojen ehdollisten jakaumien laskeamista ehdolla havainnot. Tilojen ehdollisia jakaumia voidaan merkitä yleisesti $\theta_s | y_{1:t}$. Tämä voidaan jakaa kolmeen eri tapaukseen.

Kun $s = t$, puhutaan tilojen suodatuksesta (engl. *filtering*). Tässä tapauksessa havainnot tunnetaan tarkasteltavaan ajanhetkeen asti. Tämä on tilanne useassa käytännön sovelluksessa, kun aineistoa saadaan

käyttöön ajanjaksoittain ja ollaan kiinnostuneita systeemin tämän hetkisestä tilasta.

Tilojen estimointia tilanteessa $s < t$ kutsutaan tilojen tasoitukseksi (engl. *smoothing*). Tässä tapauksessa tunnetaan havaintoja myös tarkasteltavasta ajanhetkestä eteenpäin. Tämä on kiinnostavaa esimerkiksi tilanteissa, joissa tunnetaan koko tarkasteltavan ajanjakson havainnot ja halutaan retrospektiivisesti tutkia ilmiötä.

Tilannetta $s > t$ kutsutaan ennustamiseksi (engl. *forecasting*). Ennustamisessa halutaan laskea tilojen jakaumat tulevilla ajanhetkillä. Etenkin aikasarja-analyysissä ennustaminen on monesti pääasiallinen kiinnostuksen kohde.

1.3.1. Kalmanin suodin. Kalmanin suotimen tarkoituksena on laskea tilojen jakaumat hetkellä t ehdolla havainnot samaan hetkeen asti. Linearisessa ja gaussisessa tila-avaruusmallissa tilat ovat normaalisti jakautuneita, joten voidaan merkitä $\theta_t|y_{1:t} \sim \mathcal{N}_p(m_t, C_t)$. Näin ollen ehdollisen jakauman määrittämiseen riittää ehdollisen odotusarvon $m_t = \mathbb{E}[\theta_t|y_{1:t}]$ ja ehdollisen kovarianssimatriisin $C_t = \text{Var}(\theta_t|y_{1:t})$ laskeminen. Nämä saadaan laskettua seuraavien vaiheiden kautta (ks. todistus Petris et al., 2009, s. 54–55).

- (1) Oletetaan, että $\theta_{t-1}|y_{1:t-1} \sim \mathcal{N}_p(m_{t-1}, C_{t-1})$, jossa m_{t-1} ja C_{t-1} tunnetaan.
- (2) Lasketaan yhden askeleen ennustejakauma tilavektorille θ_t ehdolla havainnot $y_{1:t-1}$. $\theta_t|y_{1:t-1} \sim \mathcal{N}_p(a_t, R_t)$, jossa

$$a_t = \mathbb{E}[\theta_t|y_{1:t-1}] = G_t m_{t-1}$$

$$R_t = \text{Var}(\theta_t|y_{1:t-1}) = G_t C_{t-1} G_t^T + W_t$$

- (3) Lasketaan yhden askeleen ennustejakauma vastemuuttujalle Y_t ehdolla havainnot $y_{1:t-1}$. Vastemuuttuja $Y_t|y_{1:t-1}$ noudattaa normaalijakaumaa keskiarvolla f_t ja varianssilla Q_t .

$$f_t = \mathbb{E}[Y_t|y_{1:t-1}] = F_t a_t$$

$$Q_t = \text{Var}(Y_t|y_{1:t-1}) = F_t R_t F_t^T + V_t$$

- (4) Lasketaan ennustevirhe $e_t = y_t - f_t$. Tämän avulla saadaan laskettua tilavektorin ehdollinen odotusarvo ja kovarianssimatriisi

$$m_t = \mathbb{E}[\theta_t|y_{1:t}] = a_t + R_t F_t^T Q_t^{-1} e_t$$

ja

$$C_t = \text{Var}(\theta_t|y_{1:t}) = R_t - R_t F_t^T Q_t^{-1} F_t R_t.$$

Tilavektorin ehdollinen odotusarvo m_t saadaan siis korjaamalla tilojen ennusteen odotusarvovektoria ennustevirheellä painotetulla vektorilla $R_t F_t^T Q_t^{-1}$. Näin ollen yksittäisen havainnon y_t vaikutuksen suuruus riippuu luvun Q_t kautta vastemuuttujan varianssista V_t ja tilojen kovarianssimatriisista R_t .

Tässä ratkaisussa ongelmaksi voi nousta laskennallinen epästabiiliisuus matriisia C_t laskettaessa. Tätä varten on kehitetty useita algoritmeja, jotka parantavat laskennallista vakautta. Tässä tutkielmassa käytetty R-paketti käyttää matriisin C_t singulaariarvohajotelmaan (engl. *singular value decomposition, SVD*) perustuvaa algoritmia.

1.3.2. Tilojen tasoitus. Tilojen tasoituksessa halutaan laskea tilojen ehdolliset jakaumat hetkellä $t < T$, kun kaikki havainnot ovat tiedossa. Myös tässä tapauksessa tilat ovat normaalisti jakautuneita, joten voidaan merkitä $\theta_t|y_{1:T} \sim \mathcal{N}_p(s_t, S_t)$. Tilavektorin ehdollinen odotusarvo s_t ja ehdollinen kovarianssimatriisi S_t voidaan laskea hyödyntämällä Kalmanin suotimen laskukaavoissa esiintyviä matriiseja. Laskeminen tapahtuu laskemalla ensin Kalmanin suotimella ehdollinen jakauma $\theta_T|y_{1:T}$ ja tämän avulla rekursiivisesti jakaumat $\theta_t|y_{1:T}$ kaikilla $t = T - 1, T - 2, \dots, 2, 1$.

Oletetaan, että $\theta_{t+1}|y_{1:T} \sim \mathcal{N}_p(s_{t+1}, S_{t+1})$. Nyt $\theta_t|y_{1:T} \sim \mathcal{N}_p(s_t, S_t)$, jossa

$$s_t = m_t + C_t G_{t+1}^T R_{t+1}^{-1} (s_{t+1} - a_{t+1})$$

ja

$$S_t = C_t - C_t G_{t+1}^T R_{t+1}^{-1} (R_{t+1} - S_{t+1}) R_{t+1}^{-1} G_{t+1} C_t.$$

Todistus Petris et al. (2009, s. 61–62).

1.3.3. Ennustaminen. Ennustamisessa ollaan kiinnostuneita tilojen ja vastemuuttujan ehdollisista jakaumista ajanhetkellä, joilta ei ole havaintoja. Merkitään tilojen ennustejakaumaa $\theta_{t+k}|y_t$ ja vastemuuttujan ennustejakaumaa $Y_{t+k}|y_t$ kaikilla $k \in \mathbb{N}$. Samoin kuin Kalmanin suotimen ja tasoituksen tapauksissa, myös nämä ehdolliset jakaumat ovat normaalijakaumia ja näin ollen riittää laskea näiden odotusarvot ja kovarianssimatriisit. Tapaus $k = 1$ saadaan Kalmanin suotimen sivutuotteena, mutta usein ollaan kuitenkin kiinnostuneita ennustamaan tilojen tai vastemuuttujan arvoja myös pidemmälle.

Määritellään aluksi neljä uutta merkintää.

$$a_t(k) = \mathbb{E}[\theta_{t+k}|y_{1:t}]$$

$$R_t(k) = \text{Var}(\theta_{t+k}|y_{1:t})$$

$$f_t(k) = \mathbb{E}[Y_{t+k}|y_{1:t}]$$

$$Q_t(k) = \text{Var}(Y_{t+k}|y_{1:t})$$

Lisäksi asetetaan $a_t(0) = m_t$ ja $R_t(0) = C_t$. Näiden avulla saadaan laskettua jakaumien $\theta_{t+k}|y_t$ ja $Y_{t+k}|y_t$ keskiarvot ja kovarianssimatriisit seuraavasti:

$$a_t(k) = G_{t+k} a_t(k-1)$$

$$R_t(k) = G_{t+k} R_t(k-1) G_{t+k}^T + W_{t+k}$$

$$f_t(k) = F_{t+k} a_t(k)$$

$$Q_t(k) = F_{t+k} R_t(k) F_{t+k}^T + V_t$$

Todistus Petris et al. (2009, s. 71).

1.3.4. Puuttuvan tiedon käsittely. Havaitusta aikasarjasta puuttuu usein yksi tai useampia havaintoja. Kalmanin suodin perustuu yhden askeleen ennusteeseen, jota korjataan havaitun arvon perusteella lasketun ennustevirheen avulla. Mikäli havaintoa ei ole tehty, ei korjausta tehdä ja suodatetuksi odotusarvoksi asetetaan suoraan ennuste ja suodatetuksi varianssiksi ennusteen varianssi. Tilanteessa, jossa useita peräkkäisiä havaintoja puuttuu, ennustetaan niin monta arvoa eteenpäin kuin mitä havaintoja puuttuu. Tämä on teknisesti helppo toteuttaa asettamalla $F_t = 0$ kaikille t , joilla havainto puuttuu. (Petris et al., 2009, s. 59)

Koska tilojen tasoitus perustuu Kalmanin suotimen rekursiiviseen käyttöön, ei havaintojen puuttuminen vaadi tasoituskaavojen muokkaamista.

1.3.5. Alustus. Edellä esitetty teoria edellyttää, että tilojen jakauma hetkellä $t = 1$ tunnetaan kokonaan. Usein käytännön sovelluksissa tämä ei kuitenkaan toteudu. Yksinkertainen menetelmä tämän ongelman ratkaisemiseksi on niin sanottu diffuusi alustus. Diffuusissa alustuksessa asetetaan $\theta_1 \sim \mathcal{N}_p(0, \text{diag}(\infty, \dots, \infty))$. Käytännössä varianssiksi ei kuitenkaan voida asettaa ääretöntä, joten varianssiksi asetetaan jokin riittävän iso luku. Liian ison varianssin valitseminen voi johtaa helposti isoihin pyöristysvirheisiin, joten diffuusin alustuksen käyttö edellyttää huolellisuutta. On kehitetty myös menetelmiä, joilla tästä ongelmasta päästään eroon, mutta tässä tutkielmassa käytetään diffuusialustusta menetelmän yksinkertaisuuden vuoksi. (Durbin & Koopman, 2012, s. 124–125)

1.4. Tilojen simulointi

Edellä on näytetty, kuinka tiloille voidaan laskea estimaatteja Kalmanin suotimella ja tasoituksella. Toisinaan on kuitenkin hyödyllistä voida simuloida arvoja tilojen jakaumasta $\theta_{0:T}|y_{1:T}$. Tätä tarvitaan muun muassa silloin, kun tiloista halutaan laskea epälineaarisia tunnuslukuja. Toinen sovelluskohde on bayesläinen mallinnus, jota esitellään luvussa 2. Yksi tapa simuloida tiloja on FFBS-algoritmi, joka on akronyymi sanoista *Forward Filtering Backward Sampling*.

FFBS-algoritmin ideana on ensin laskea Kalmanin suotimella jakauman $\theta_T|y_{1:T}$ parametrit ja sen jälkeen rekursiivisesti takaperin edeten kaikki loput jakaumat. Voidaan näyttää (esim. Petris et al., 2009, s. 162), että $\theta_t|\theta_{t+1:T}, y_{1:T}$ on sama jakauma kuin $\theta_t|\theta_{t+1}, y_{1:T}$ ja tämä on moniulotteinen normaalijakauma $\mathcal{N}_p(h_t, H_t)$, jossa

$$h_t = m_t + C_t G_{t+1}^T R_{t+1}^{-1} (\theta_{t+1} - a_{t+1})$$

ja

$$H_t = C_t - C_t G_{t+1}^T R_{t+1}^{-1} G_{t+1} C_t.$$

Tässä m_t , C_t , R_{t+1} ja a_{t+1} saadaan Kalmanin suotimesta ja G_{t+1} on määritelty mallissa.

Algoritmi etenee siis seuraavasti:

- (1) Aja Kalmanin suodin
- (2) Arvo θ_T jakaumasta $\mathcal{N}_p(m_T, C_T)$
- (3) Toista kaikilla $t = T - 1, \dots, 0$:
 - (a) Laske parametrit h_t ja H_t
 - (b) Arvo θ_t jakaumasta $\mathcal{N}_p(h_t, H_t)$

1.5. Tuntemattomien parametrien estimointi

Kalmanin suotimen ja tasoituksen käyttö edellyttää, että kaikki mallin määrittävät matriisit F_t , G_t , V_t ja W_t tunnetaan täysin. Näin ei kuitenkaan ole useissa käytännön tilanteissa. Erityisesti havaintovarianssia V_t ja tilojen kehityksen kovarianssimatriisia W_t ei usein tunneta.

Kootaan kaikki matriisien F_t , G_t , V_t ja W_t tuntemattomat parametrit yhteen vektoriin ψ . Kun ψ on asetettu johonkin arvoon, niin kaikkien havaintojen yhteisjakauma voidaan kirjoittaa $p(y_1, \dots, y_n; \psi)$. Uskottavuusfunktio on vakiokerrointa lukuun ottamatta tämä yhteistihyysfunktio tulkittuna parametrin ψ funktiona.

$$(1.3) \quad L(\psi) = c \cdot p(y_1, \dots, y_n; \psi) = c \cdot \prod_{t=1}^n p(y_t | y_{1:t-1}; \psi),$$

jossa $c \in \mathbb{R}$. Lausekkeen oikealla puolella olevat jakaumat ovat lineaarisen ja gaussisen mallin tapauksessa normaalijakaumia, joiden tiheysfunktiot voidaan kirjoittaa suljetussa muodossa. Näin ollen uskottavuusfunktion logaritmi voidaan kirjoittaa muodossa

$$(1.4) \quad l(\psi) = -\frac{1}{2} \sum_{t=1}^n \log |Q_{t,\psi}| - \frac{1}{2} \sum_{t=1}^n (y_t - f_{t,\psi})^T Q_{t,\psi}^{-1} (y_t - f_{t,\psi}),$$

jossa $f_{t,\psi}$ ja $Q_{t,\psi}$ saadaan Kalmanin suotimesta. Suurimman uskottavuuden estimaattori on vektori $\hat{\psi}$, joka maksimoi funktion $l(\psi)$.

$$\hat{\psi} = \operatorname{argmax}(l(\psi))$$

Käytännössä funktiota $l(\psi)$ optimoidaan aina numeerisesti.

Dynaamisten lineaaristen mallien uskottavuusfunktiolla voi olla useita lokaaleja maksimeja, joten numeeristen optimointialgoritmien käyttö edellyttää huolellisuutta. On suositeltavaa aloittaa optimointi useista eri alkupisteistä ja tarkastella, että konvergoiko algoritmi aina samaan pisteeseen. (Petris et al., 2009, s. 144)

Toinen mahdollinen ongelma muodostuu, jos uskottavuusfunktio on hyvin latteaa. Tällöin suurimman uskottavuuden estimaattorin varianssi

on suuri ja uskottavuusfunktion arvo on likimain sama useilla eri vektoreilla ψ . Tämä tilanne esiintyy muun muassa silloin, kun malli on yliparametrisoitu. Suurimman uskottavuuden estimaattorin varianssia voidaan arvioida logaritmisen uskottavuusfunktion Hessen matriisin avulla, sillä tietyin lievillä ehdoin suurimman uskottavuuden estimaattori on asympotoottisesti normaalisti jakautunut vektorin $\hat{\psi}$ ympäristössä. Tällöin logaritmisen uskottavuusfunktion Hessen matriisin käänteismatriisin H^{-1} arvo pisteessä $\hat{\psi}$ estimoi suurimman uskottavuuden estimaattorin varianssia. (Petris et al., 2009, s. 144)

Optimoimisen sijasta uskottavuusfunktiota voidaan myös simuloida. Simuloinnin etuna on, että siinä paljastuu helposti, mikäli uskottavuusfunktio on hyvin lattea. Näin ollen parametrien estimoinnissa oleva epävarmuus tulee huomioiduksi. Simulointialgoritmeja ja niiden soveltamista tila-avaruusmalleihin esitellään luvussa 2.

LUKU 2

Metropolis-algoritmin laajennuksia

Simulointiin perustuvat menetelmät ovat yleistyneet tilastotieteessä huomasti viime vuosikymmenien aikana. Sen sijaan, että laskettaisiin satunnaismuuttujan jakauma ja siitä johdettuja tunnuslukuja analyytisesti, simuloinnissa pyritään saamaan satunnaisotos satunnaismuuttujan jakaumasta ja lasketaan halutut tunnusluvut siitä. Mikäli esimerkiksi haluttaisiin laskea piste, jonka alapuolelle jää 95 % havainnoista, voidaan simuloida lukuja kyseisestä jakaumasta ja laskea, minkä pisteen alapuolelle jää 95 % simuloituista arvoista. (Gelman et al., 2003, s. 25)

Simulointia voidaan hyödyntää tila-avaruusmalleissa muun muassa tuntemattomien parametrien estimoinnissa. Tässä tutkielmassa käytetään menetelmää, jossa Kalmanin suotimesta laskettavaa uskottavuusfunktioita (funktio (1.3) sivulla 12) simuloidaan. Tämä tekniikka on esitetty kirjassa *Bayesian Filtering and Smoothing* (Särkkä, 2013).

Joissain tapauksissa halutusta jakaumasta voidaan simuloida lukuja suoraan. Esimerkiksi normaalijakauman tapaukseen on kehitetty useita menetelmiä, jolla voidaan simuloida toisistaan riippumattomia lukuja normaalijakaumasta (esim. Box & Muller, 1958). Usein ei kuitenkaan ole mahdollista tai laskennallisesti kannattavaa pyrkiä simuloimaan toisistaan riippumattomia lukuja. Metropolis et al. (1953) esittivät menetelmän, jossa seuraava simuloitu luku voi riippua edellisestä simuloidusta luvusta. Vaikka tämä nykyisin Metropolis-algoritmina tunnettu menetelmä esitettiin alunperin fysiikan numeerisena integrointimenetelmänä, käytetään sitä nykyisin hyvin laajasti satunnaislukujen simulointiin eri todennäköisyysjakaumista.

Metropolis-algoritmi vaatii jonkin verran säätötoimenpiteitä, joten sen käyttö voi olla hyvin työlästä erityisesti moniulotteisissa tapauksissa. Tätä varten on kehitetty adaptiivisia Metropolis-algoritmeja (engl. *Adaptive Metropolis, AM*), jotka säätävät itse itsensä. Tässä tutkielmassa esitellään niistä yksi (Haario et al., 2001). Toinen varsin suosittu Metropolis-algoritmin parannus on viivästetyn hylkäyksen menetelmä (engl. *Delayed Rejection, DR*), jonka esittivät nykymuodossaan Tierney & Mira (1999). Siinä ideana on hyödyntää useita eri ehdotusjakaumia samassa algoritmossa. Haario et al. (2006) näyttivät, että nämä kaksi tehokasta menetelmää voidaan myös yhdistää viivästetyn hylkäyksen adaptiiviseksi Metropolis-algoritmiksi (engl. *Delayed Rejection Adaptive Metropolis, DRAM*).

Tässä luvussa esitellään aluksi tavanomainen Metropolis-algoritmi. Tämän jälkeen esitellään adaptiivinen Metropolis-algoritmi ja viivästetyn hylkäyksen Metropolis-algoritmi ja näytetään, kuinka nämä kaksi menetelmää voidaan yhdistää viivästetyn hylkäyksen adaptiiviseksi Metropolis-algoritmiksi. Lopuksi näytetään konkreettisesti, miten tätä DRAM-menetelmää voidaan hyödyntää tila-avaruusmallien sovittamisessa.

Vaikka tässä tutkielmassa hyödynnetäänkin simulointimenetelmiä vain tila-avaruusmalleissa, esitetään kaikki simulointiin liittyvä teoria yleistä notaatiota käyttäen, koska tässä tutkielmassa esiteltävät simulointimenetelmät eivät sinänsä ole suunniteltu mihinkään yksittäiseen sovelluskohteeseen. Myös liitteessä A esitetty funktio DRAM on kirjoitettu siten yleiskäyttöiseksi, että sitä voidaan hyödyntää myös muiden jakaumien simuloinnissa.

2.1. Metropolis-algoritmi

Metropolis-algoritmin tavoitteena on poimia satunnaisotos todennäköisyysjakaumasta, jonka tiheysfunktion arvot voidaan laskea skaalauskestijää lukuunottamatta. Metropolis-algoritmi ei tuota toisistaan riippumattomia lukuja, sillä seuraava arvottu luku riippuu aina edellisestä luvusta. Tämä ei kuitenkaan ole välttämättä ongelma, sillä tunnuslukuja voidaan estimoida harhattomasti myös toisistaan riippuvia havaintoja käyttäen (Lunn et al., 2012, s. 63). Jos kuitenkin halutaan riippumattomampia lukuja, voidaan ketjua harventaa (engl. *thinning*). Tällöin ketjusta käytetään vain esimerkiksi joka n :ttä arvoa (Lunn et al., 2012, s. 77).

Metropolis-algoritmin ideana on, että uutta lukua ehdotetaan aina edellisen luvun perusteella. Tämä uusi luku joko hyväksytään tai hylätään tietyllä todennäköisyydellä. Näin satunnaisluvusta muodostuu ketju, joka riittävän monen simuloinnin jälkeen on edustava otos halutusta jakaumasta. Alla esitetään Metropolis-algoritmi kuten kirjassa Gelman et al. (2003, s. 289–290).

Merkitään kirjaimella x satunnaismuuttujaa, jonka jakaumaa $P(x)$ halutaan simuloida. Satunnaismuuttuja x voi olla joko skalaari- tai vektoriarvoinen muuttuja. Tässä merkitään satunnaismuuttujan x dimensiota kirjaimella d . Ehdotusjakauma on symmetrinen jakauma $Q(x|x^{t-1})$, jolle siis pätee $Q(x|x^{t-1}) = Q(x^{t-1}|x)$

- (1) Valitse aloituspiste $x^0 \in \mathbb{R}^d$ siten, että tiheysfunktion arvo $p(x^0) > 0$
- (2) Toista $t = 1, 2, \dots$:
 - (a) Arvo ehdokaspiste x^* symmetrisestä ehdotusjakaumasta $Q(x|x^{t-1})$

(b) Laske tiheysfunktion arvojen suhde:

$$r = \frac{p(x^*)}{p(x^{t-1})}$$

(c) Laske hyväksymistodennäköisyys $\alpha = \min(r, 1)$

(d) Aseta

$$x^t = \begin{cases} x^* & \text{todennäköisyydellä } \alpha \\ x^{t-1} & \text{muutoin} \end{cases}$$

Hastings (1970) kehitti Metropolis-algoritmia siten, ettei ehdotusjakauman Q tarvitse olla symmetrinen. Tämä Metropolis-Hastings-algoritmi on muutoin vastaava kuin Metropolis-algoritmi, mutta ehdotuksen x^* hyväksymistodennäköisyyden laskemisessa käytetään myös ehdotusjakauman tiheysfunktion arvoja. Katso yksityiskohdat esim. Gelman et al. (2003, s. 297).

Ehdotusjakauma $Q(x|x^{t-1})$ voi teoriassa olla melkein mikä tahansa jakauma, mutta käytännön kannalta on erittäin suuri merkitys sillä, miten se valitaan. Tyypillinen valinta on esimerkiksi edellisen arvon ympärille keskittynyt tasajakauma tai normaalijakauma, mutta toisinaan käytetään myös esimerkiksi nollan suhteen peilattua tasajakamaa. Huomaa, että sekä tasajakauma että normaalijakauma ovat symmetrisiä jakaumia, joten Metropolis-algoritmia voidaan käyttää.

Mikäli aloituspiste x^0 on huono, voi ketjulla kestää jonkin aikaa, ennen kuin se pääsee alueelle, jossa $p(x)$ on suuri. Tätä kutsutaan ketjun lämpenemiseksi. Usein onkin tapana, ettei ensimmäisiä simulointeja (ns. burn-in-jaksoa) käytetä, kun ketjusta tehdään päättelyä. (Lunn et al., 2012, s. 71–72)

Hastings (1970) näytti, että Metropolis-Hastings-algoritmissa vektoriarvoisen satunnaismuuttujan x komponentit voidaan päivittää joko kaikki kerralla tai vain yksi kerrallaan. Tässä tutkielmassa rajoitutaan tarkastelemaan algoritmin versiota, jossa kaikki komponentit päivitetään kerralla.

2.2. AM-algoritmi

Metropolis-algoritmia toteuttaessa tulee määrittää ehdotusjakauma $Q(x|x^{t-1})$. Tyypillinen valinta on normaalijakauma, jonka keskiarvo on ketjun edeltävän hetken sijainti x^{t-1} ja kovarianssimatriisi C . Kovarianssimatriisi C tulee valita huolellisesti, sillä jos varianssit ovat liian suuria, suurin osa ehdotuksista on hyvin kaukana edellisestä pisteestä ja hylkäyksiä tulee usein, mutta jos taas varianssit ovat liian pieniä, kestää kohtuuttoman pitkään, ennen kuin ketju on ehtinyt kulkea kaikilla alueilla, joissa $p(x) > 0$ (Gelman et al., 2003, s. 292). Lisäksi kovarianssimatriisin C ei-diagonaaliset alkiot, eli parametrien väliset kovarianssit, tulisi määrittää siten, että parametrien välinen riippuvuus-rakenne huomioidaan tehokkaalla tavalla. Tämä ei ole aina helppoa,

etenkään korkeaulotteisissa tapauksissa. Tätä varten on kehitetty menetelmiä, jotka säätävät kovarianssimatriisia C aiemmin simuloitujen arvojen perusteella.

Haario et al. (2001) esittivät adaptiivisen Metropolis-algoritmin (lyh. AM), jossa kovarianssimatriisi on aina aiemmin simuloitujen arvojen empiirinen kovarianssimatriisi kerrottuna vakiolla. Ennen adaptoinnin aloittamista kuitenkin käytetään hetki käsin määriteltyä kovarianssimatriisia C_0 . Ehdotusjakauma on siis

$$x|x^{t-1} \sim \mathcal{N}_d(x^{t-1}, C_t),$$

jossa

$$C_t = \begin{cases} C_0, & t \leq t_0 \\ s_d \text{Cov}(x^0, \dots, x^{t-1}), & \text{muutoin.} \end{cases}$$

Tässä t_0 määrittää, että kuinka pitkään ketjua ajetaan ennen adaptoinnin aloittamista. Vakiokerroin s_d tulee valita käsin, mutta on näytetty, että tietyssä mielessä optimaalinen valinta on

$$s_d = \frac{2.4^2}{d}.$$

Vakiokerroimen s_d arvo riippuu siis satunnaismuuttujan x dimensiosta d . Tässä tutkielmassa käytetään tätä arvoa vakiokerroimelle s_d .

Huomattava on, ettei tällä tavalla simuloitu ketju enää ole Markovin ketju, sillä ehdotusjakauma riippuu koko historiasta, eikä vain edeltävästä arvosta. Haario et al. (2001) näyttivät kuitenkin, että konvergenssitulokset ovat voimassa tästä huolimatta.

Haario et al. (2001) ehdottivat myös, että laskennan nopeuttamiseksi voidaan tehdä niin, että kovarianssimatriisia ei lasketa jokaisella ajanhetkellä vaan ainoastaan tietyin väliajoin.

2.3. DR-algoritmi

Toinen yleinen muunnos Metropolis-algoritmista on viivästetyn hylkäyksen Metropolis-algoritmi (lyh. DR), jonka kehittivät Tierney & Mira (1999). Viivästetyn hylkäyksen Metropolis-algoritmista on ideaa, että kun tavallisessa Metropolis-algoritmista tulee hylkäys, niin paikallaan pysymisen sijaan yritetään uutta ehdotusta eri ehdotusjakauksella. Tämä mahdollistaa erilaisten ehdotusjakauksien yhdistämisen samaan algoritmiin. Haario et al. (2006) vertaavat tätä tenniksen syöttöstrategiaan, jossa ensimmäisellä syöttöyrityksellä yritetään rohkeaa ässäsyöttöä, mutta toisella syöttöyrityksellä tyydytään varmempaan suoritukseen. Jakauman simuloinnissa tämä tarkoittaa, että ensimmäisellä ehdotuksella voidaan yrittää suurta, globaalia siirtymää, mutta toisella ehdotuksella pyritään varmistamaan, että ketju liikkuu edes johonkin.

Tässä esitellään melko yksinkertainen versio viivästetyn hylkäyksen Metropolis-algoritmista. Tämän menetelmän on esitellyt Mira (2001).

Oletetaan, että Metropolis-algoritmissa on edetty vaiheeseen, jossa ehdokaspiste x^* on poimittu ehdotusjakaumasta $Q_1(x|x^{t-1})$, hyväksymistodennäköisyydeksi on laskettu α_1 , mutta ehdotus päättyy hylkäykseen. Nyt poimitaan uusi ehdokaspiste x^{**} jakaumasta $Q_2(x|x^{t-1}, x^*)$. Tämä uusi ehdokaspiste hyväksytään todennäköisyydellä

$$\alpha_2 = \min \left(1, \frac{p(x^{**})q_1(x^*|x^{**})q_2(x^{i-1}|x^{**}, x^*) \left[1 - \min \left(1, \frac{p(x^*)}{p(x^{**})} \right) \right]}{p(x^{i-1})q_1(x^*|x^{i-1})q_2(x^{**}|x^{i-1}, x^*)(1 - \alpha_1)} \right).$$

Jos ehdotusjakauma Q_2 ei riipu ensimmäisestä ehdotetusta arvosta ja se on symmetrinen, yksinkertaistuu edellinen kaava muotoon

$$\alpha_2 = \min \left(1, \frac{p(x^{**})q_1(x^*|x^{**}) \left[1 - \min \left(1, \frac{p(x^*)}{p(x^{**})} \right) \right]}{p(x^{i-1})q_1(x^*|x^{i-1})(1 - \alpha_1)} \right).$$

On kehitetty useita eri tapoja määrittellä toinen ehdotusjakauma Q_2 . Tässä tutkielmassa käytetään hyvin yksinkertaista keinoa, jossa molemmat ehdotusjakaumat Q_1 ja Q_2 ovat normaalijakaumia, joiden molempien keskiarvovektori on ketjun nykyinen sijainti, x^{t-1} . Toisen ehdotusjakauman kovarianssimatriisi on skaalaustekijää lukuunottamatta sama kuin ensimmäisen ehdotusjakauman kovarianssimatriisi. Tässä siis molemmat ehdotusjakaumat ovat symmetrisiä ja toinen ehdotusjakauma ei riipu ensimmäisestä ehdotuksesta. Näin ollen voidaan käyttää yksinkertaisempaa muotoa hyväksymistodennäköisyydestä. Ensimmäinen ehdotusjakauma on siis

$$Q_1 = \mathcal{N}_d(x^{t-1}, C)$$

ja toinen ehdotusjakauma

$$Q_2 = \mathcal{N}_d(x^{t-1}, \gamma C),$$

jossa $\gamma > 0$. Parametri γ voidaan valita vapaasti, mutta useat simulointikokeet (Green & Mira, 2001; Haario et al., 2006) ovat osoittaneet, että useimmiten on hyödyllistä valita $\gamma < 1$.

Olisi mahdollista rakentaa myös useamman askeleen viivästetyn hylkäyksen Metropolis-algoritmi (Haario et al., 2006). Tämä tarkoittaisi sitä, että mikäli toinenkin ehdotus päättyy hylkäykseen, otettaisiin seuraava ehdotus ehdotusjakaumasta Q_3 ja niin edelleen. Tässä kuitenkin rajoitutaan tarkastelemaan vain kahden ehdotuksen versiota.

2.4. DRAM-algoritmi

Toisinaan käy niin, ettei kumpikaan edellä esitetyistä Metropolis-algoritmin parannelluista versioista yksinään riitä tuottamaan hyvää tulosta. Adaptiivisen Metropolis-algoritmin heikkous on se, että mikäli ehdotusjakauman kovarianssimatriisin alkuarvo C_0 on liian suuri, ei ehdotuksia hyväksytä juuri lainkaan ja algoritmilla on vaikeuksia päästä alkuun. Mikäli taas C_0 on liian pieni, voi kestää kauan ennen kuin ketju

on ehtinyt kulkea laajan alueen läpi ja C_t on saatu adaptoitua riittävän suureksi. Viivästetyn hylkäyksen Metropolis-algoritmin heikkous taas on siinä, että mikäli kaikki ehdotusjakaumat ovat määritelty huonosti, ei algoritmi tuota tuloksia järkevissä ajassa. Haario et al. (2006) näyttivät, että nämä kaksi menetelmää voidaan yhdistää viivästetyn hylkäyksen adaptiiviseksi Metropolis-algoritmiksi (lyh. DRAM).

Viivästetyn hylkäyksen adaptiivisen Metropolis-algoritmin idea on, että varsinaista ehdotusjakaumaa Q_1 adaptoidaan kuten normaalisti AM-algoritmissa. Sen lisäksi käytetään toista ehdotusjakaumaa Q_2 , jonka kovarianssimatriisi on skaalaustekijää lukuunottamatta sama kuin ensimmäisen ehdotusjakauman. Algoritmi etenee siis hetkellä t seuraavasti:

- (1) Lasketaan ensimmäisen ehdotusjakauman kovarianssimatriisi C_t kuten luvussa 2.2 näytettiin
- (2) Ehdotetaan uutta pistettä x^* jakaumasta $\mathcal{N}_d(x^{t-1}, C_t)$
- (3) Todennäköisyydellä α (ks. luku 2.1) asetetaan $x^t = x^*$, muutoin:
 - (a) Ehdotetaan uutta pistettä x^{**} jakaumasta $\mathcal{N}_d(x^{t-1}, \gamma C_t)$
 - (b) Todennäköisyydellä α_2 (ks. luku 2.3) asetetaan $x^t = x^{**}$, muutoin asetetaan $x^t = x^{t-1}$.

Haario et al. (2006) näyttivät, että DRAM-algoritmi ratkaisee useita ongelmia, joita esiintyy AM- ja DR-algoritmeissa. Liitteessä A on R-ohjelmoinnilla toteutettu funktio DRAM, joka suorittaa DRAM-simulointia.

2.5. DRAM-algoritmin sovellus tila-avaruusmalleihin

Adaptiiviset MCMC-menetelmät soveltuvat hyvin tila-avaruusmallien analysointiin. Luvussa 1.5 esitettiin, että mallin uskottavuusfunktioita (1.3) tai logaritmistista uskottavuusfunktioita (1.4) voidaan optimoida numeerisilla menetelmillä. Numeeriseen optimointiin liittyy kuitenkin usein epävarmuutta, koska uskottavuusfunktioilla voi olla useampia lokaaleja maksimeita, joihin optimointialgoritmi jää jumiin. Toinen mahdollinen ongelma on, että toisinaan uskottavuusfunktio on melko tasainen, eli sen arvo on useassa eri pisteessä liki sama. Tällöin eri alkuarvoista aloitetut optimoinnit voivat päätyä hyvinkin erilaisiin arvoihin. Numeerisen optimoinnin käyttö edellyttääkin siis parhaimmillaankin suurta huolellisuutta. (Petris et al., 2009, s. 144)

Suurimman uskottavuuden menetelmässä (luku 1.5) toimitaan niin, että uskottavuusfunktion maksimoiva vektori $\hat{\psi}$ sijoitetaan suoraan malliin ikään kuin tunnettuna tosiasiana. Petris et al. (2009, s. 148) esittävätkin, että parametreja ψ käsiteltäisiin satunnaismuuttujina, joiden estimoinnin epävarmuus on huomioitava analyysissä. Yksi vaihtoehto

olisi käyttää konjugaattiprioreihin nojaavaa Gibbs-otantaa (esim. Petris et al., 2009, luku 4.1.), mutta tässä tutkielmassa käytetään uskottavuusfunktion simulointia Metropolis-algoritmillä, kuten Särkkä (2013, luku 12) ohjeistaa.

Kalmanin suotimella laskettu uskottavuusfunktio $L(\psi)$ on havaintojen ja tuntemattomien parametrien yhteisjakauman uskottavuusfunktio. Käytännössä useimmin lasketaan kuitenkin sen logaritmi $l(\psi)$. Särkkä (2013, s. 188) kertoo, että tätä uskottavuusfunktiota voidaan simuloida Metropolis-Hastings-pohjaisia simulointialgoritmeja käyttämällä, jotta parametrejen ψ yhteisjakauma saadaan selville. Mikäli ollaan kiinnostuneita myös tiloista, niin niitä voidaan simuloida käyttämällä luvussa 1.4 esiteltyä FFBS-algoritmia. Tällöin siis poimitaan vuorotellen realisaatio varianssiparametrien jakaumasta käyttäen Metropolis-algoritmia (tässä tutkielmassa DRAM-algoritmia) ja tilojen jakaumasta käyttäen FFBS-algoritmia, kuten Petris et al. (2009, s. 163) ohjeistaa.

Parametreille ψ voidaan asettaa myös priorijakauma. Simuloitava jakauma on tällöin priorijakauman ja uskottavuusfunktion tulo. Kuten Gamerman (1997, s. 58) kertoo, dynaamiset lineaariset mallit mahdollistavat ennakkotiedon hyödyntämisen tehokkaasti priorien avulla. Tässä tutkielmassa käytetään kuitenkin hyvin yksinkertaista prioria, joka saa vakioarvon kun $\psi_i \geq 0$ kaikilla i ja muutoin arvon 0. Näin ollen posteriorijakauman tiheysfunktion arvo on $L(\psi)$ kun $\psi_i \geq 0$ kaikilla i ja 0 muutoin. Tällöin posteriorin maksimi (MAP, *maximum a posteriori*) on myös uskottavuusfunktion maksimi.

On huomattava, ettei tällä tavalla määritelty prior ole aito jakauma, sillä sen tiheysfunktion integraali ei suppene. Posteriorijakauma on kuitenkin aito jakauma, sillä uskottavuusfunktio määrittää aidon jakauman, eikä positiiviseksi rajatun tasajakauman tiheysfunktiolla kertominen vaikuta tiheysfunktion integraalin suppenemiseen.

Viivästetyn hylkäyksen adaptiivinen Metropolis-algoritmi on hyvin käytännöllinen valinta tuntemattomien parametrien posteriorijakauman simulointiin tila-avaruusmalleissa, sillä hyvä ehdotusjakauma on usein vaikea määritellä. Adaptiivisuus pitää huolen siitä, että ehdokasjakaumaksi saadaan sopiva jakauma ja viivästetty hylkäys taas auttaa adaptointia pääsemään alkuun.

LUKU 3

Otsonimäärän mallintaminen tila-avaruusmallilla

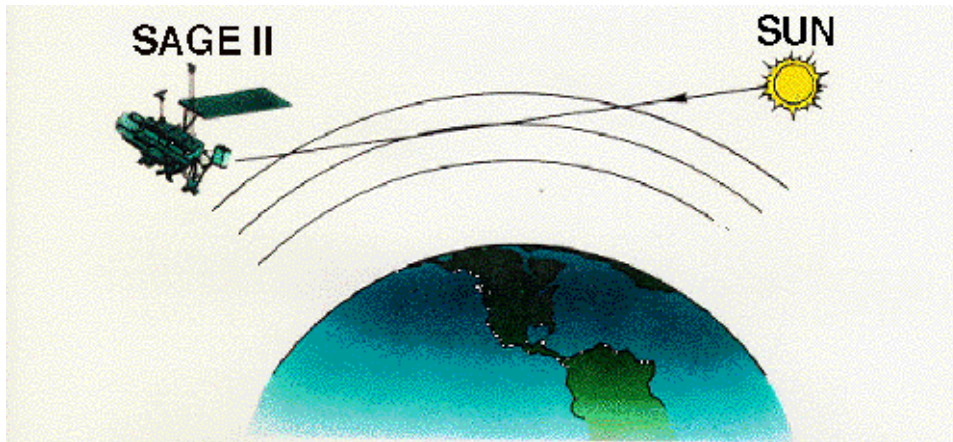
Tässä luvussa käytetään edellä esitettyjä menetelmiä otsonimäärän mallintamiseen. Aluksi esitellään käytössä oleva aineisto ja kuvataan lyhyesti, miten se on muokattu käytössä olevaan muotoon. Tämän jälkeen tarkastellaan, millä tavalla otsonimäärää on mallinnettu aiemmin ja esitetään perustelut, miksi tätä mallia tulee parantaa. Tila-avaruusmallien avulla rakennetaan paranneltu otsonimalli, jonka yhteensopivuutta ja tuloksia tarkastellaan yhden leveyspiirin yhdellä korkeusalueella.

3.1. Otsoniaineisto

Tässä tutkielmassa käytetty otsoniaineisto on saatu Ilmatieteen laitoksen Uudet havaintomenetelmät -yksikön Ilmakehän kaukokartoitus -ryhmältä. Vastemuuttujana käytetty aikasarja sisältää kahden eri satelliitti-instrumentin tuottamia havaintoja otsonin määrästä yläilmakehässä.

Yhdysvaltain ilmailu- ja avaruushallinnon (*National Aeronautics and Space Administration, NASA*) satelliitti-instrumentti Stratospheric Aerosol and Gas Experiment II (SAGE II) kiersi maapalloa ympäri ERBS-satelliitin mukana vuodesta 1984 vuoteen 2005 saakka. SAGE II -instrumentin toiminta perustui siihen, että se mittasi ilmakehän läpi kulkevan auringonvalon auringonnousun ja -laskun aikoina. Auringon valon spektrin taittuminen riippuu ilmakehän koostumuksesta. Näin ollen mitatun spektrin perusteella voidaan matemaattista inversiota käyttämällä tehdä päätelmiä ilmakehän koostumuksesta. SAGE II -instrumentin toimintaperiaatetta havainnollistaa kuva 3.1. SAGE II keräsi tietoa muun muassa otsonin (O_3), typpidioksidin (NO_2) ja veden (H_2O) tiheyksistä ilmakehässä 0,5 kilometrin ja 70 kilometrin korkeuksien väliltä. (National Aeronautics and Space Administration, 2012) Tässä tutkielmassa käytetty versio SAGE II -aineistosta on saatu käyttämällä artikkelissa Chu et al. (1989) esiteltyä inversioalgoritmia.

Toinen tässä tutkielmassa käytetty satelliitti-instrumentti on GOMOS. GOMOS, eli Global Ozone Monitoring by Occultation of Stars, oli ENVISAT-satelliittiin kiinnitetty satelliitti-instrumentti, joka tuotti tietoa ilmakehän tilasta vuosien 2002 ja 2012 välillä. GOMOS oli Euroopan avaruusjärjestön (*European Space Agency, ESA*) projekti. GOMOS-instrumentti on osittain Suomessa suunniteltu ja tehty, ja siihen liittyvää algoritmikehitystä tehdään edelleenkin Ilmatieteen laitoksella.



KUVA 3.1. SAGE II -instrumentin toimintaperiaate
(Kuvan lähde: NASA)



KUVA 3.2. GOMOS-instrumentin toimintaperiaate
(Kuvan lähde: Ilmatieteen laitos)

GOMOS-instrumentin perusideana oli mitata ilmakehän läpi kulkevien tähtien valon spektri. GOMOS siis toimi vastaavalla periaatteella kuin SAGE II, mutta Auringon valon sijaan se pystyi hyödyntämään noin 180 kirkkaimman tähden valoa (Tamminen et al., 2010). Tätä havainnollistetaan kuvassa 3.2. Tällä tavalla saatiin tietoa muun muassa otsonin (O_3), typpidioksidin (NO_2), nitraatin (NO_3), veden (H_2O) ja hapen (O_2) määristä ilmakehässä. (European Space Agency, 2007) GOMOS-instrumentin tekemien havaintojen, eli tähtien valon spektrin, muuntamista ilmankehän koostumusta kuvaaviksi tunnusluvuiksi on käsitelty laajemmin artikkelissa Kyrölä et al. (2010) ja varsinainen inversioalgoritmi esitellään artikkelissa Bertaux et al. (2010).

SAGE II ja GOMOS -aikasarjat on yhdistetty yhdeksi pitkäksi aikasarjaksi käyttäen hyväksi yhteistä toiminta-aikaa vuosien 2002 ja 2005 välillä. Aikasarjojen yhdistämistä käsitellään laajemmin artikkelissa Kyrölä et al. (2013).

Tässä tutkielmassa käytetyssä otsoniaikasarjassa kutakin kuukautta edustaa vain yksi arvo. Tämä arvo on saatu laskemalla keskiarvo kunkin korkeusalueen havainnoista kunkin kuukauden ajalta. Tämä luku on saatu jakamalla aineisto leveyspiirin, korkeuden ja ajan suhteen osiin ja laskemalla keskiarvo tämän osion kaikista havainnoista. Havainnon epävarmuutta kuvaa samasta osiosta laskettu keskiarvon keskihajonta. Otsonimäärän yksikkö tässä aineistossa on molekyyli/cm³.

Otsoniaikasarja alkaa lokakuusta 1984 ja loppuu lokakuuhun 2011. Näin ollen aikasarja kattaa yhteensä 324 kuukautta, mutta näistä 67 kuukaudelta havainto puuttuu. Havaintojen puuttuminen johtuu mitaustekniikan epävarmuudesta. Tätä käsitellään laajemmin artikkelissa Kyrölä et al. (2013).

3.2. Taustamuuttujat

On olemassa luonnollisia tekijöitä, jotka tunnetusti vaikuttavat otsonin määrään ilmakehässä. Näihin kuuluu sekä Auringon aktiivisuus että Singaporen tuuli.

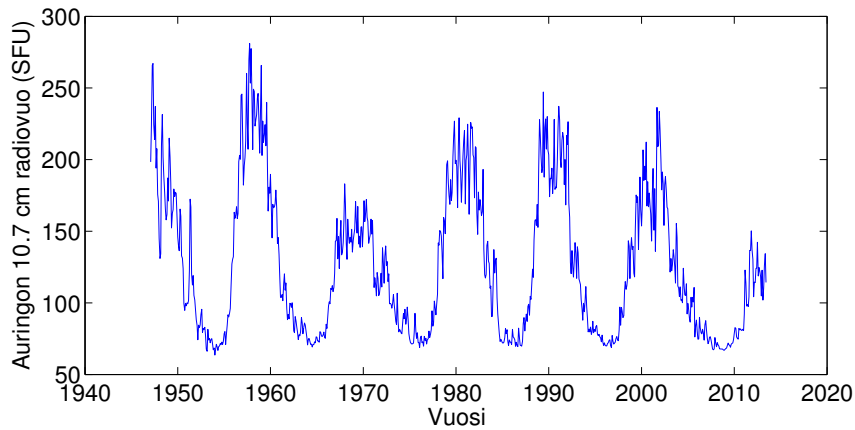
Auringon magneettinen aktiivisuus vaikuttaa muun muassa auringonpurkausten ja auringonpilkkujen määrään. Auringon aktiivisuus vaikuttaa myös maapalloon, sillä Auringon ollessa aktiivisimmillaan ultraviolettisäteilyn ja sähkömagneettisen säteilyn määrä maapallon yläilmakehässä on huomattavasti normaalia korkeampi. Auringon magneettinen aktiivisuus vaihtelee noin 11 vuoden mittaisissa sykleissä. (Hathaway, 2010)

Auringonpilkkujen lukumäärä korreloi hyvin voimakkaasti Auringon magneettisen aktiivisuuden kanssa, jonka vuoksi sitä on käytetty pitkään kuvaamaan Auringon magneettista aktiivisuutta. Auringonpilkkujen lukumäärää voi tarkastella yksinkertaisesti kaukoputkea käyttämällä, joten siitä on voitu kerätä aineistoa jo 1700-luvulta lähtien. Toisaalta juuri tästä syystä auringonpilkkujen laskeminen on myös ongelmallista, sillä havaittu auringonpilkkujen lukumäärä voi riippua havainnoijasta ja havainnointi ei onnistu kaikissa sääolosuhteissa. Tästä syystä Auringon magneettista aktiivisuutta on viime aikoina tarkasteltu Auringon 10.7 cm radiovuon (engl. *solar radio flux*) avulla (Hathaway, 2010). Auringon radiovuolla on oma yksikkö, SFU (*solar flux unit*).

$$1 \text{ SFU} = 10^{-22} \text{ W m}^{-2} \text{ Hz}^{-1}$$

(Lee et al., 1995).

Auringon aktiivisuuden vaikutusta ilmakehän otsonimäärään on tutkittu jo pitkään. Angell (1989) näytti, että otsonimäärällä ja auringonpilkkujen lukumäärällä on voimakas yhteys. Angell (1989) käytti aineistona maan pinnalta tehtyjä mittauksia otsonimäärästä. Koska Auringon aktiivisuuden jaksot ovat melko pitkiä ja ilmakehän kaukokartoitus on uudehko tieteenala, ei Auringon aktiivisuuden vaikutusta pystytty



KUVA 3.3. Auringon aktiivisuus vaihtelee noin 11 vuoden sykleissä

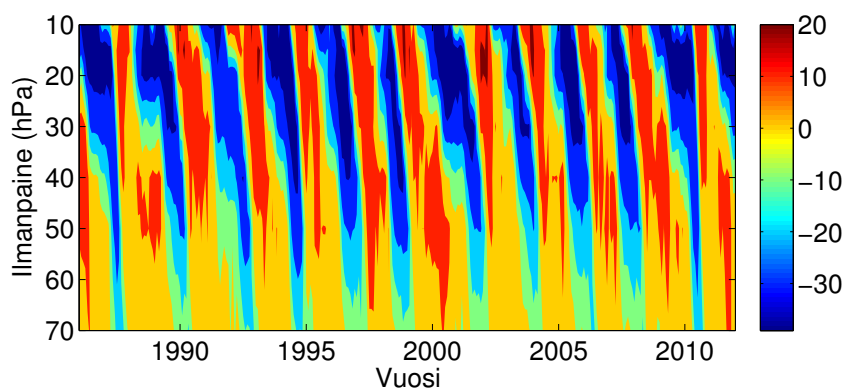
estimoimaan kunnolla satelliittimittauksista ennen kuin 1990-luvulla (Miller et al., 1996).

Canadian Solar Radio Monitoring Programme on kerännyt Auringon 10.7 cm radiovuon arvoja vuodesta jo 1946 (Hathaway, 2010). Tämä aikasarja esitetään kuvassa 3.3. Aineisto on vapaasti ladattavissa Internet-sivustolta http://www.spaceweather.ca/data-donnee/sol_flux/sx-5-mavg-eng.php.

Singaporen tuuli (engl. *quasi-biennial oscillation*, QBO) on päiväntasaajan lähialueilla vaikuttava ilmiö, jolla on suuri merkitys moniin muihin stratosfäärissä tapahtuviin ilmiöihin (Baldwin et al., 2001). Singaporen tuuli puhaltaa joko idästä tai lännestä ja tämä vaihtelee noin kahden vuoden mittaisissa jaksoissa. Tuulen voimakkuus ja suunta ilmoitetaan yleensä metreinä sekunnissa siten, että positiiviset arvot tarkoittavat länsituulta ja negatiiviset itätuulta. Tuulen voimakkuus ja jakson vaihe riippuvat korkeusalueesta, kuten kuvasta 3.4 voidaan havaita.

Erityisesti Singaporen tuulen vaikutus otsonimäärään on tunnettu jo pitkään (esim. Angell & Korshover, 1964). Vaikka Singaporen tuuli puhaltaakin vain päiväntasaajan lähetyillä, on sillä vaikutusta otsonimäärään myös muilla leveyspiireillä, kuten esimerkiksi artikkeleissa Bowman (1989) ja Randel & Cobb (1994) on havaittu.

Singaporen tuulen vaikutusta otsonimäärään tutkitaan käyttämällä mittaustuloksia kahdelta eri korkeusalueelta, 10 hPa ja 30 hPa. Ilmanpaineen muuntaminen korkeudeksi merenpinnan yläpuolella riippuu hieman sääoloista, mutta 10 hPa vastaa noin 26 km korkeutta ja 30 hPa noin 22 km korkeutta merenpinnan yläpuolella. Nämä kaksi korkeusalueita on valittu, koska ne kuvaavat tuulen eri vaiheita. Korkeusalueiden valintaa tarkastellaan artikkelissa Hauchecorne et al. (2010) ja näitä valittuja korkeusalueita on käytetty vastaavassa mallintamisessa myös muissa alan tutkimuksissa (esim. Kyrölä et al., 2013).



KUVA 3.4. Singaporen tuulen voimakkuus (m/s) ja suunta vaihtelevat korkeuden ja ajanhetken mukaan. Positiiviset arvot tarkoittavat länsituulta ja negatiiviset itätuulta.

Aineisto on vapaasti saatavilla Internet-sivustolta <http://www.geo.fu-berlin.de/en/met/ag/strat/produkte/qbo/>.

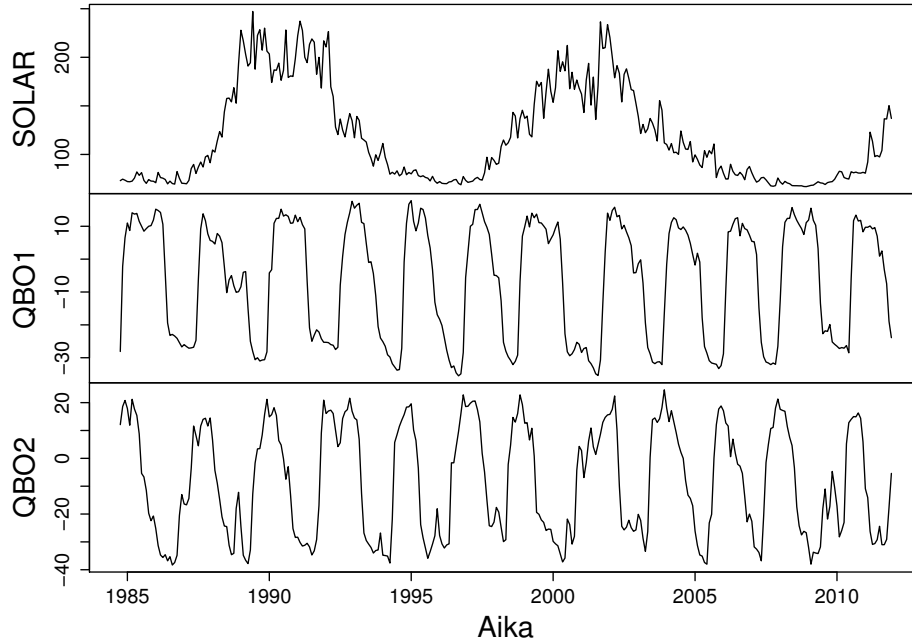
3.3. Otsonimalli

Aineiston analysoinnissa käytetään pohjana mallia, jota on käytetty aiemmin vastaavassa otsoniaineiston analysoinnissa (Newchurch et al., 2003). Aiemmin käytetty malli on

$$(3.1) \quad O_3(t) = \beta_1 + \beta_2 t + \beta_3 \sin(2\pi t) + \beta_4 \cos(2\pi t) \\ + \beta_5 \sin(2\pi 2t) + \beta_6 \cos(2\pi 2t) + \beta_7 \text{QBO}_1(t) \\ + \beta_8 \text{QBO}_2(t) + \beta_9 \text{SOLAR}(t) + \varepsilon(t),$$

jossa trigonometriset funktiot kuvaavat vuosittaista ja puolivuositista kausivaihtelua, $\text{QBO}_1(t)$ ja $\text{QBO}_2(t)$ ovat Singaporen tuulen vaikutusta kuvaavia indeksejä ja $\text{SOLAR}(t)$ on auringon aktiivisuutta kuvaava indeksi. Tässä t on ajanhetki vuosina. Virheterminä on $\varepsilon(t)$. Mallissa oletetaan, että virhetermit ovat toisistaan riippumattomia ja normaalisti jakautuneita.

Otsonin määrän kehityksessä uskotaan tapahtuneen muutos noin vuonna 1997 (Newchurch et al., 2003) ja mallin (3.1) jäännöksiä tutkimalla selviää, että trendi ei ole täysin lineaarinen myöskään sitä ennen. Kyrölä et al. (2013) laajensivat mallia siten, että malli mahdollistaa lineaarisen trendin muuttumisen yhdessä ennalta asetettussa käännoispisteessä. Ei kuitenkaan ole perusteltua syytä olettaa, että trendi muuttuu vain tasan yhdessä pisteessä ja lisäksi vaikka näin olisi, niin tämä käännoispiste tulisi tietää etukäteen. Tämän vuoksi halutaan mahdollistaa trendin vapaampi muuttuminen. Tässä tutkielmassa tämä toteutetaan tila-avaruusmallilla.



KUVA 3.5. Taustamuuttujien aikasarjat tarkasteltavalla aikavälillä

Mallissa käytetään hyväksi kuhunkin havaintoon liittyvää keskiarvon keskivirhettä. Näiden lukujen ajatellaan estimoivan havaintojen keskihajontoja yhteistä skaalaustekijää lukuunottamatta. Merkitään kutakin keskiarvon keskivirhettä ϵ_t ja yhteistä skaalaustekijää σ_ϵ . Lu-
vut ϵ_t siis tunnetaan kaikilla $t = 1, \dots, N$, mutta σ_ϵ tulee estimoida. Näin ollen mallissa havaintovirhe $v_t \sim \mathcal{N}_1(0, \sigma_\epsilon^2 \epsilon_t^2)$

Trendi rakennetaan käyttämällä lokaalia lineaarista trendimallia. Aiemmissa tutkimuksissa trendi on ollut lineaarinen (esim. Newchurch et al., 2003; Jones et al., 2009) tai paloittain lineaarinen (Kyrölä et al., 2013), mutta tässä tutkielmassa halutaan mahdollistaa trendin vapaampi muuttuminen, joten trendin varianssiparametri estimoidaan.

Mallissa on kolme taustamuuttujaa: Singaporen tuuli kahdella eri korkeusalueella (QBO_1 ja QBO_2) ja Auringon aktiivisuus (SOLAR). Näiden aikasarjat tarkasteltavalta aikaväliltä esitetään kuvassa 3.5. Näihin liittyville regressiokertoimille sallitaan satunnaiskulku.

Malliin kuuluu myös kausivaihtelu, jonka mallintaminen toteutetaan Fourier-taajuuksilla. Aineisto on kuukausittaista, joten kausivaihtelun mallintamiseen tarvitaan 11 tilakomponenttia. Kausivaihtelulle sallitaan satunnaiskulku, mutta satunnaiskulkuun liittyvät varianssit oletetaan samaksi kaikille tilakomponenteille.

TAULUKKO 3.1. Varianssiparametrien tunnuslukuja

parametri	5% kvantiili	mediaani	95% kvantiili	MAP-estimaatti
σ_α^2	$1.519 \cdot 10^{-6}$	$2.344 \cdot 10^{-5}$	$1.668 \cdot 10^{-4}$	$3.465 \cdot 10^{-6}$
σ_ζ^2	$1.597 \cdot 10^{-4}$	$4.214 \cdot 10^{-4}$	$8.580 \cdot 10^{-4}$	$3.987 \cdot 10^{-4}$
$\sigma_{\beta_1}^2$	$6.634 \cdot 10^{-4}$	$9.378 \cdot 10^{-3}$	$4.536 \cdot 10^{-2}$	$1.021 \cdot 10^{-4}$
$\sigma_{\beta_2}^2$	$2.281 \cdot 10^{-5}$	$4.842 \cdot 10^{-4}$	$5.124 \cdot 10^{-3}$	$3.056 \cdot 10^{-3}$
$\sigma_{\beta_3}^2$	$2.641 \cdot 10^{-4}$	$5.809 \cdot 10^{-3}$	$2.651 \cdot 10^{-2}$	$9.990 \cdot 10^{-3}$
σ_ϵ^2	17.68	26.76	38.81	29.7

Kun nämä komponentit yhdistetään samaan malliin, saadaan samanlainen malli kuin aliluvussa 1.2.4. Näin ollen tuntemattomat parametrit mallin määrittämisessä ovat varianssit σ_α^2 , $\sigma_{\beta_1}^2$, $\sigma_{\beta_2}^2$, $\sigma_{\beta_3}^2$, σ_ζ^2 ja havaintovirhematriisin yhteinen tekijä σ_ϵ^2 . Myös muita mahdollisia varianssien määrittämiä kokeiltiin, mutta tämä malli oli diagnostiikaltaan selvästi paras. Tuntemattomien parametrien yhteisjakaumaa simuloitiin luvussa 2 esiteltyllä DRAM-algoritilla. Ketjun alkupisteeksi asetettiin suurimman uskottavuuden estimaatit ja ehdotusjakauman kovarianssimatriisin alkuarvoksi diagonaalimatriisi, jonka diagonaalilla on vastaavat suurimman uskottavuuden estimaatit kerrottuna vakiolla 0.1. Tilat simuloitiin käyttäen harvennettua varianssiparametrien ketjua ja FFBS-algoritmia.

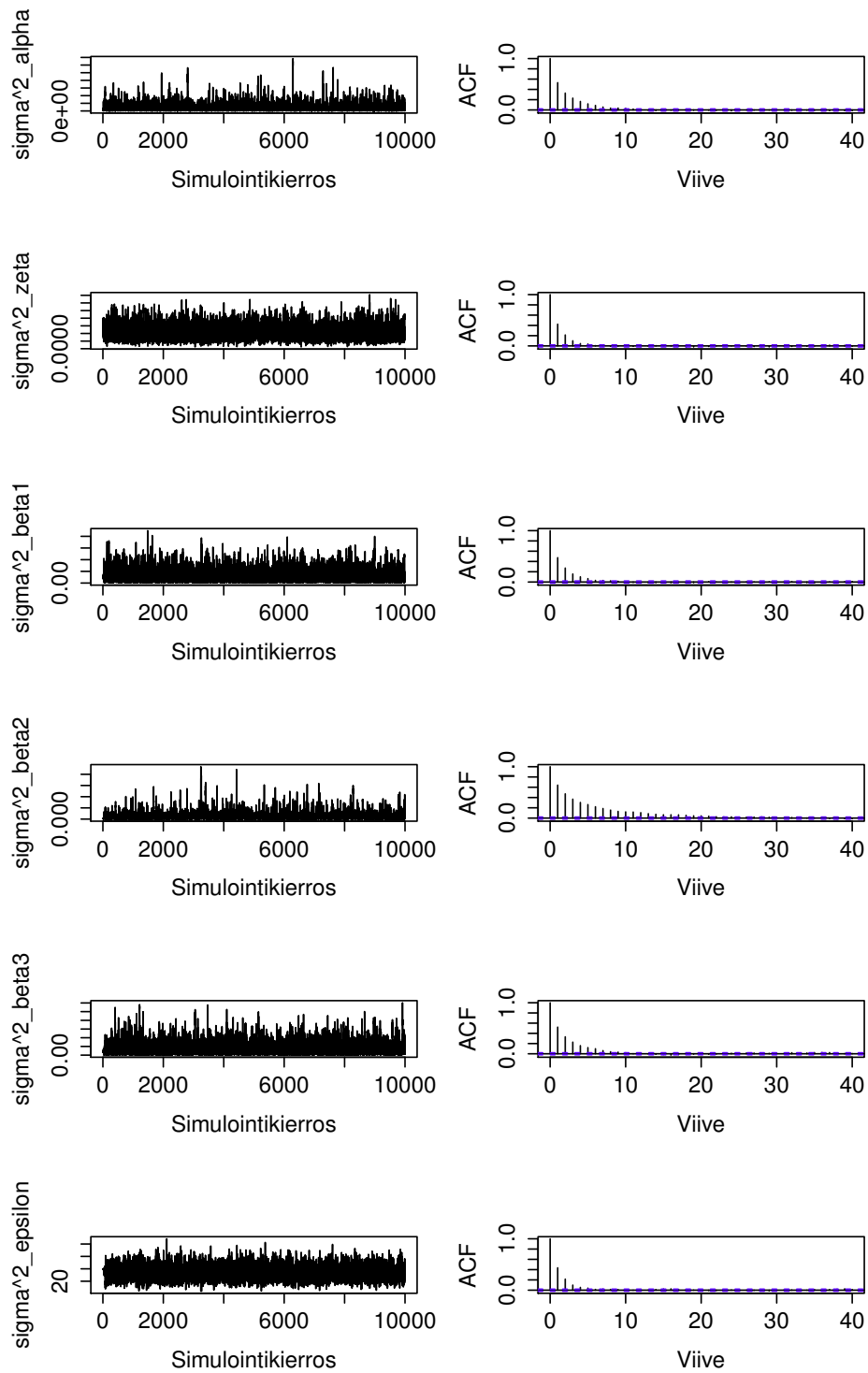
Kaikki laskeminen tehtiin käyttäen vapaan lähdekoodin ohjelmistoa R (R Development Core Team, 2011) ja sen toimintaa laajentavaa pakettia dlm (Petris, 2010) sekä tätä tarkoitusta varten kirjoitettuja uusia funktioita (liite A). Aineiston käsittelyyn ja joidenkin kuvien piirtämiseen käytettiin ohjelmistoa MATLAB (The MathWorks Inc., 2013).

3.4. Tulokset

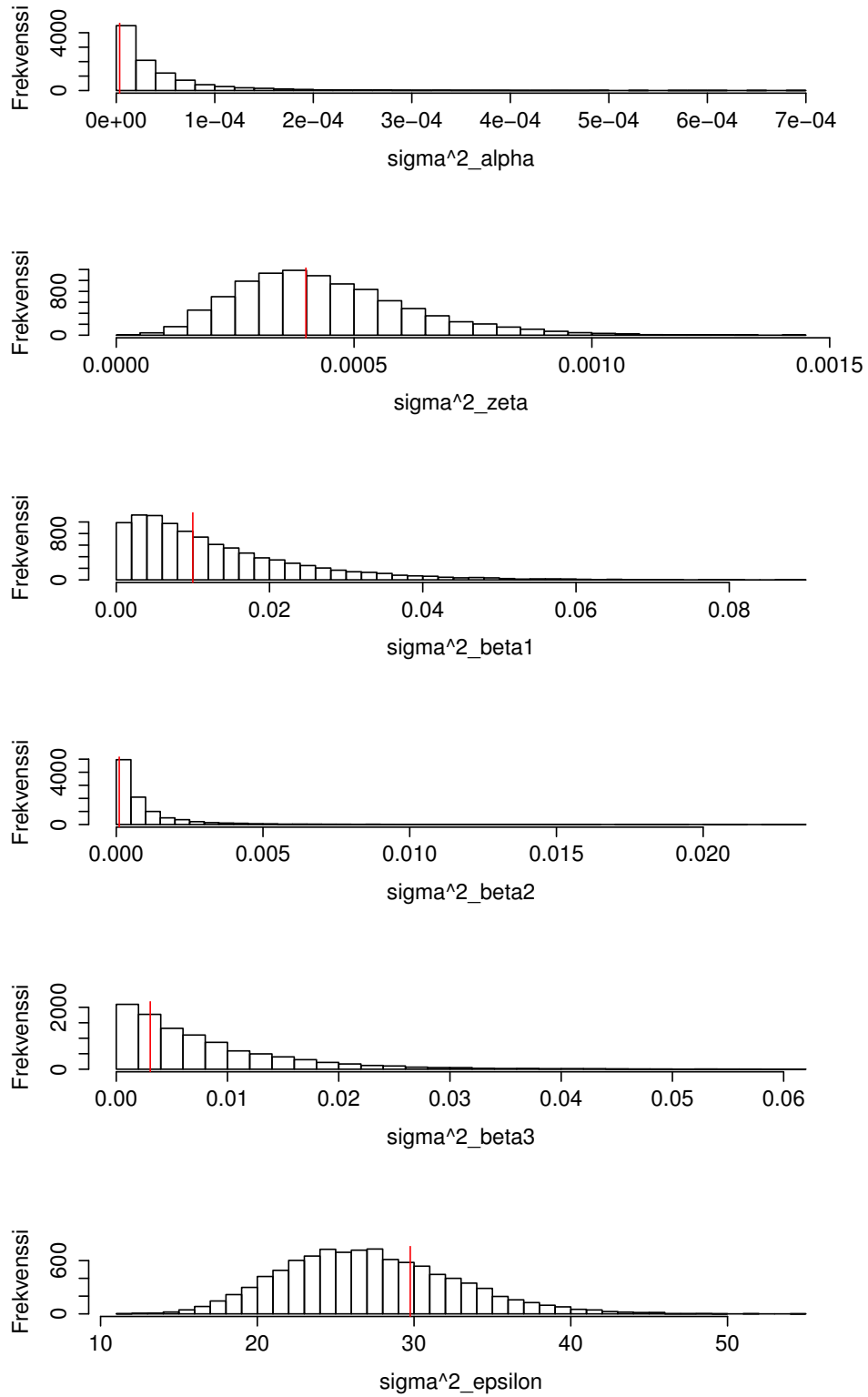
DRAM-algoritmia ajettiin 500000 kierrosta, josta poistettiin alusta 10000 ensimmäistä kierrosta burn-in-jaksona. Tämän jälkeen ketju harvennettiin siten, että joka viideskymmenes iteraatio tallennettiin. Varianssiparametrien harvennetut MCMC-ketjut esitetään kuvassa 3.6. Kuten siitä voidaan havaita, löytyi hyvä ehdotusjakauma nopeasti ja ketju on silmämääräisen tarkastelun perusteella hyvin konvergoinut. Hyväksytyjen ehdotusten osuus oli 66.7 %.

Kuvassa 3.7 esitetään varianssiparametrien histogrammit. Tästä voidaan nähdä, että jokaisen varianssiparametrin arvot voivat vaihdella jokseenkin leveällä välillä. Tämä havainnollistaa sitä, minkä vuoksi analyysissä halutaan huomioida varianssiparametrien epävarmuus. Varianssiparametrien tunnuslukuja esitetään taulukossa 3.1.

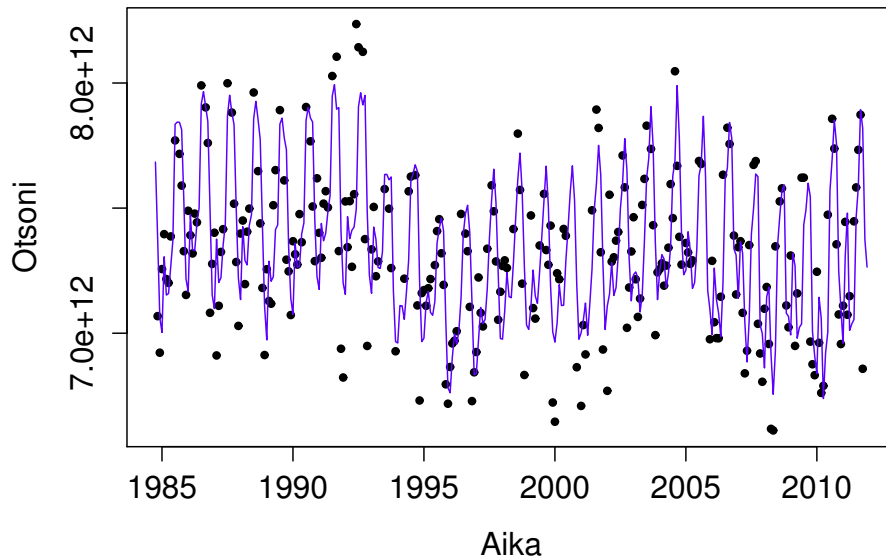
Kuvassa 3.8 esitetään havainnot ja MAP-varianssiestimaateilla sovitettujen mallien Kalmanin tasoituksella lasketut ennusteet. Kuten tästä kuvasta voi havaita, ei malli pysty selittämään kaikkea vaihtelua.



KUVA 3.6. Simuloidut MCMC-ketjut harvennuksen jälkeen ja vastaavat autokorrelaatiofunktion kuvaajat



KUVA 3.7. Varianssiparametrien histogrammit. Punainen viiva merkitsee MAP-estimaattia

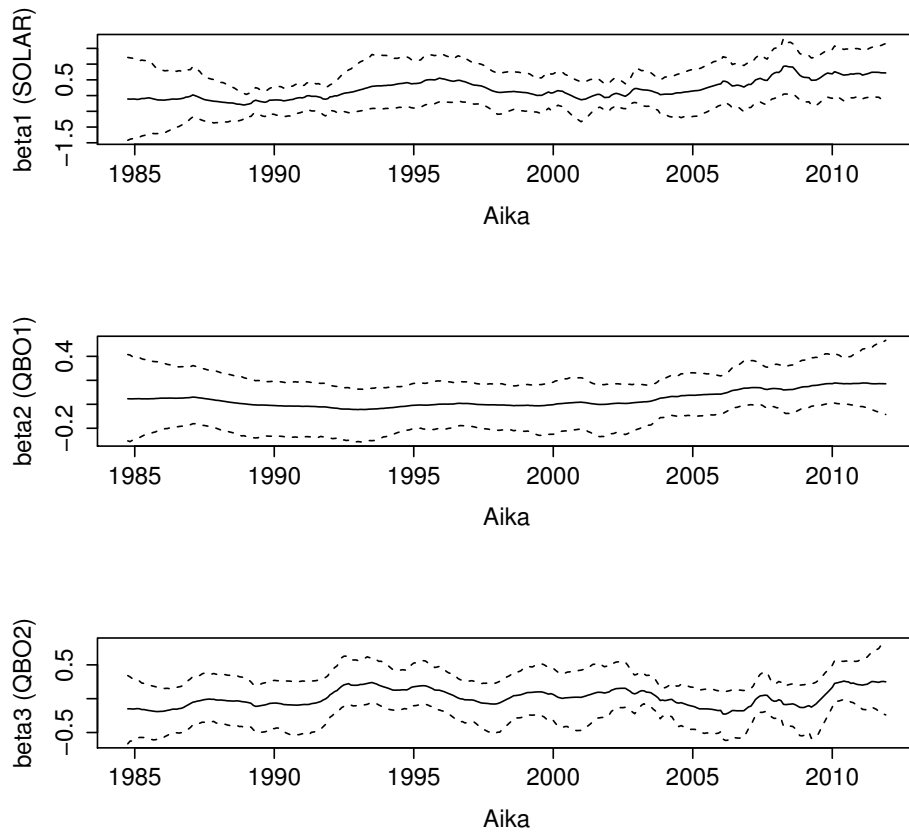


KUVA 3.8. Havainnot (mustat pisteet) ja Kalmanin ta-soituksella lasketut ennusteet (violetti viiva), kun va-rianssiksi on asetettu MAP-estimaatit

Taustamuuttujien vaikutuksen arvioinnissa on paljon epävarmuutta. Kuten kuvasta 3.9 voi havaita, ovat kaikki regressiokertoimet varsin lähellä nollaa ja käytännössä kaikilla ajanhetkillä nolla kuuluu 95 % luottovälille. Tässä aineistossa taustamuuttujilla ei siis näy kovin selvää vaikutusta. Kuva 3.10 näyttää kunkin taustamuuttujan vaikutuksen, eli simuloidulla regressiokertoimella kerrotun taustamuuttujan arvon. Siitä näkee hyvin, kuinka paljon näiden taustamuuttujien estimointiin liittyy epävarmuutta. Erityisesti auringon aktiivisuuden vaikutus voi saada monia eri suuruisia arvoja. Tämä korostaa sitä, min-ikä takia on tärkeää ottaa huomioon varianssiparametrien estimoinnin epävarmuus.

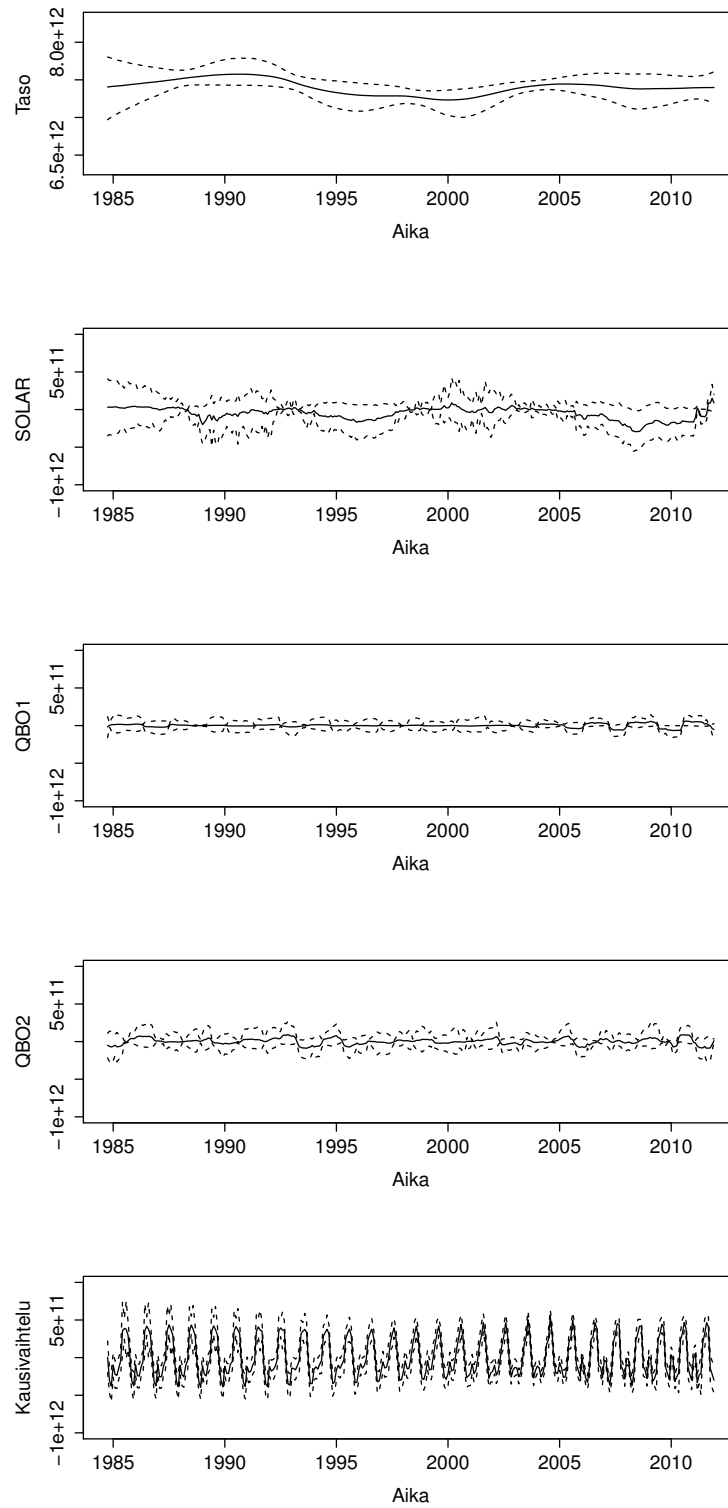
Kuvassa 3.10 esitetään kaikki mallin komponentit. Mallisovite on siis näiden komponenttien summa. Tästä voidaan havaita, että kaikkein eniten mallissa on vaikutusta kausivaihtelulla.

Tutkimuskysymyksenä oli selvittää, miten otsonimäärä on kehitty-nyt, kun tunnetut luonnolliset tekijät ja kausivaihtelu on huomioitu. Tätä analysoidaan tutkimalla mallin ensimmäistä tilaa μ_t . Otsonimää-
rän estimoidaan laskeneen noin 1990-luvun alusta aina noin vuoteen 1995 saakka, jolloin lasku alkoi hidastua ja otsonimäärän estimointiin

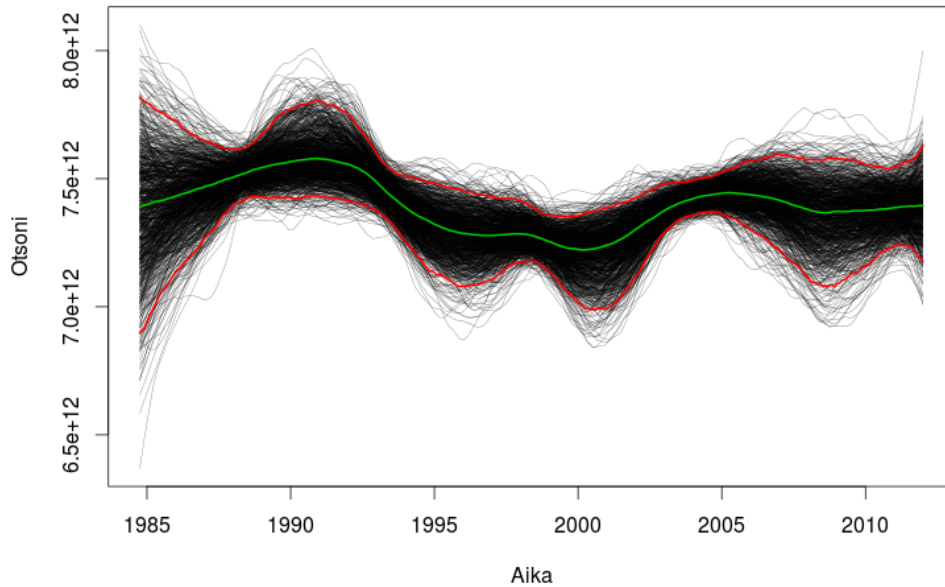


KUVA 3.9. Simuloitujen regressiokertoimien mediaani kullakin ajanhetkellä (yhtenäinen viiva) ja sen 95 % luottoväli (katkoviivat).

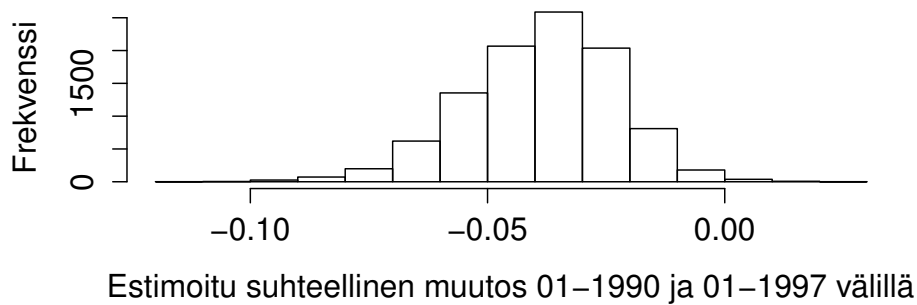
kääntyneen jopa nousuun (kuva 3.11). Muutokset ovat kuitenkin varsin pieniä ja etenkin otsonimäärän kääntymisestä nousuun ei ole kovin voimakasta näyttöä. Tätä voidaan tarkastella myös kuvista 3.12 ja 3.13, jotka esittävät aikasarjan keskimääräisen tason (tila μ_t) suhteellista muutosta ajankohtien tammikuu 1990 ja tammikuu 1997 (kuva 3.12) ja tammikuu 1997 ja joulukuu 2011 (kuva 3.13) väleillä. Nämä histogrammit on saatu laskemalla simuloituista 1. tiloista kiinnostuksen kohteena olevat muutokset. Kuten kuva 3.12 osoittaa, on hyvin voimakasta näyttöä siitä, että keskimääräinen otsonimäärä laskenut vuosien 1990 ja 1997 välillä. Arvoista 99.5 % on negatiivisia ja 95 % arvoista kuuluu välille $[-0.071; -0.011]$. Kuvasta 3.13 taas nähdään, että vuosien 1997 ja 2011 välillä muutos ei ole näin selvä. Suuri osa arvoista (81.3 %) on positiivisia, mutta tämä ei ole vielä kovin voimakasta näyttöä otsonimäärän noususta. Arvoista 95 % on välillä $[-0.020; 0.054]$.



KUVA 3.10. Mallin komponenttien mediaanit (yhtenäiset viivat) ja niiden 95 % luottovälit (katkoviivat).

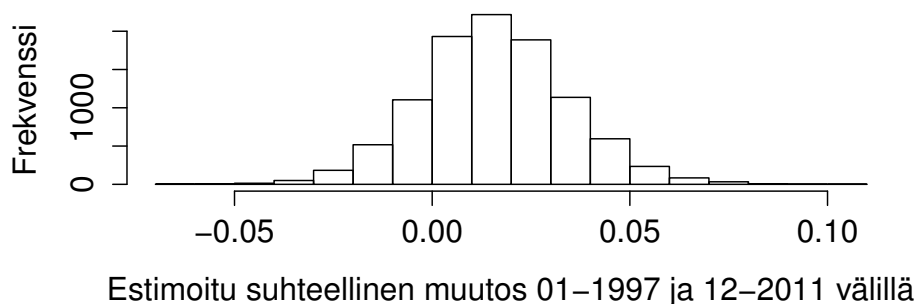


KUVA 3.11. 1000 simuloitua realisaatiota 1. tiloista, eli aikasarjan tasosta (mustat viivat), 95 % luottoväli (punaiset viivat) ja mediaani (vihreä viiva). Tässä ollaan huomioitu varianssiparametrien estimoinnin mukanaan tuoma epävarmuus.



KUVA 3.12. Ensimmäisen tilan suhteellinen muutos tammikuun 1990 ja tammikuun 1997 välillä

Nämä tulokset ovat hyvin linjassa aiemman aihetta koskeneen tutkimuksen (esim. Newchurch et al., 2003; Jones et al., 2009; Kyrölä et al., 2013) kanssa, mutta tässä tapauksessa trendiä ei ole rajoitettu lineaarisiksi, ja näin ollen päästään tarkastelemaan paremmin trendin mahdollisia käännepeisteitä.



KUVA 3.13. Ensimmäisen tilan suhteellinen muutos tammikuun 1997 ja joulukuun 2011 välillä

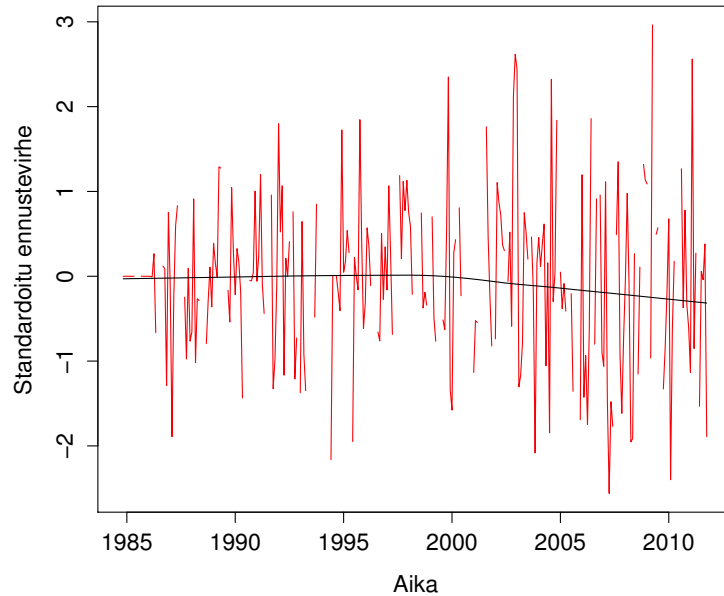
3.5. Mallidiagnostiikka

Mallidiagnostiikka ollaan tehty niin, että malliin ollaan sijoitettu varianssiparametrien MAP-estimaatit, jotka vastaavat tässä tapauksessa suurimman uskottavuuden estimaatteja. Bayesläisessä päätelyssä mallidiagnostiikka hoidetaan usein esimerkiksi poistamalla havainnot ja tarkastelemalla mallin ennustuskykyä, mutta tässä päädyttiin käyttämään varianssiparametrien MAP-estimaatteja menetelmän yksinkertaisuuden vuoksi.

Mallin yhteensopivuuden tarkasteluun käytettiin standardoituja ennustevirheitä $\bar{e}_t = e_t/\sqrt{Q_t}$, jossa e_t ja Q_t on kuten on määritelty luvussa 1.3.1 sivulla 9. Mikäli mallin oletukset ovat voimassa, ovat nämä likimäärin riippumattomia satunnaisotoksia normaalijakaumasta $\mathcal{N}_1(0, 1)$.

Kuvaan 3.14 on piirretty standardoidut ennustevirheet ajan suhteen sekä LOESS-käyrä. LOESS-käyrä on eräänlainen lokaali regressiomalli, jota käytetään tässä yhteydessä tarkastelemaan, onko ennustevirheissä säännönmukaista sovittamatonta vaihtelua. Yksityiskohtaisempaa kuvausta varten katso esim. Cleveland (1979). Kuten tästä kuvasta huomataan, ei ennustevirheissä ole sovittamatta jäänyttä trendin vaihtelua. Jäännösten varianssi tosin näyttää kasvavan hieman ajan suhteen, mikä herättää epäilyksiä aineiston heteroskedastisuudesta. Heteroskedastisuus ei kuitenkaan vaikuta kovin suurelta, joten sitä ei käsitellä tässä tarkemmin.

Ennustevirheissä ei ole juurikaan autokorrelaatiota, kuten käy ilmi kuvasta 3.15. Kuvaan 3.15 on piirretty autokorrelaatiofunktion arvoja eri viiveillä. Autokorrelaatiofunktion arvoja on estimoitu vain niissä tapauksissa, kun kumpikaan tarkasteltavista ennustevirheistä ei puuttu. Lisäksi laskettiin Ljung-Box-testin p -arvoja viiveeseen 20 saakka. Ljung-Box-testi testaa, esiintyykö aineistossa autokorrelaatiota viiveen

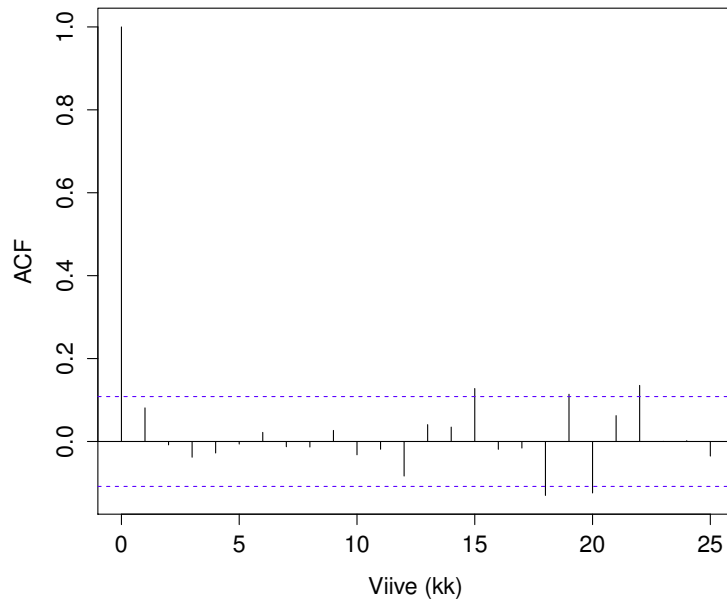


KUVA 3.14. Standardoidut ennustevirheet (punainen viiva) ja LOESS-käyrä (musta viiva)

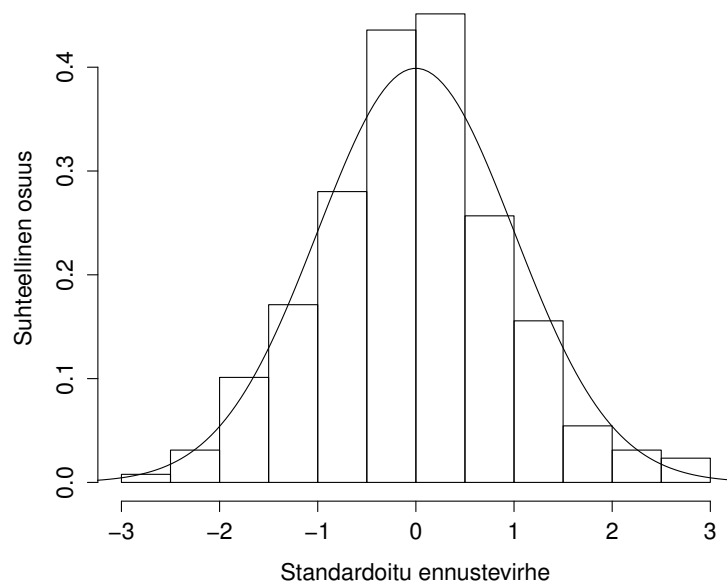
k tai pienemmän viiveen autokorrelaatiota, ks. Ljung & Box (1978). Kaikki Ljung-Box-testin p -arvot olivat suurempia kuin 0.05 kaikille tarkastelluille viiveille, joten autokorrelaatiosta ei ole näyttöä.

Ennustevirheet ovat varsin hyvin normaalijakautuneita, kuten kuvan 3.16 histogrammista voi havaita. Myöskään jäännösten graafinen tarkastelu teoreettisten ja empiiristen kvantiilien avulla (kuva 3.17) ei osoita selviä ongelmia.

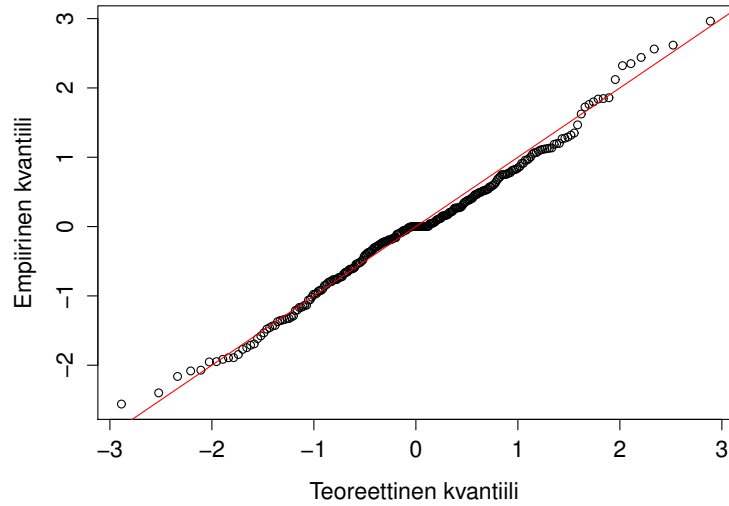
Kuvassa 3.18 on standardoidut ennustevirheet kuukausittain. Standardoitujen ennustevirheiden kuukausikohtaiset keskiarvot, keskihajonnot ja varianssit esitetään taulukossa 3.2. Näistä nähdään, että minäkään kuukauden ennustevirheet eivät ole säännöllisesti nollasta poikkeavia.



KUVA 3.15. Standardoitujen ennustevirheiden autokorrelaatiofunktion estimaatteja eri viiveillä



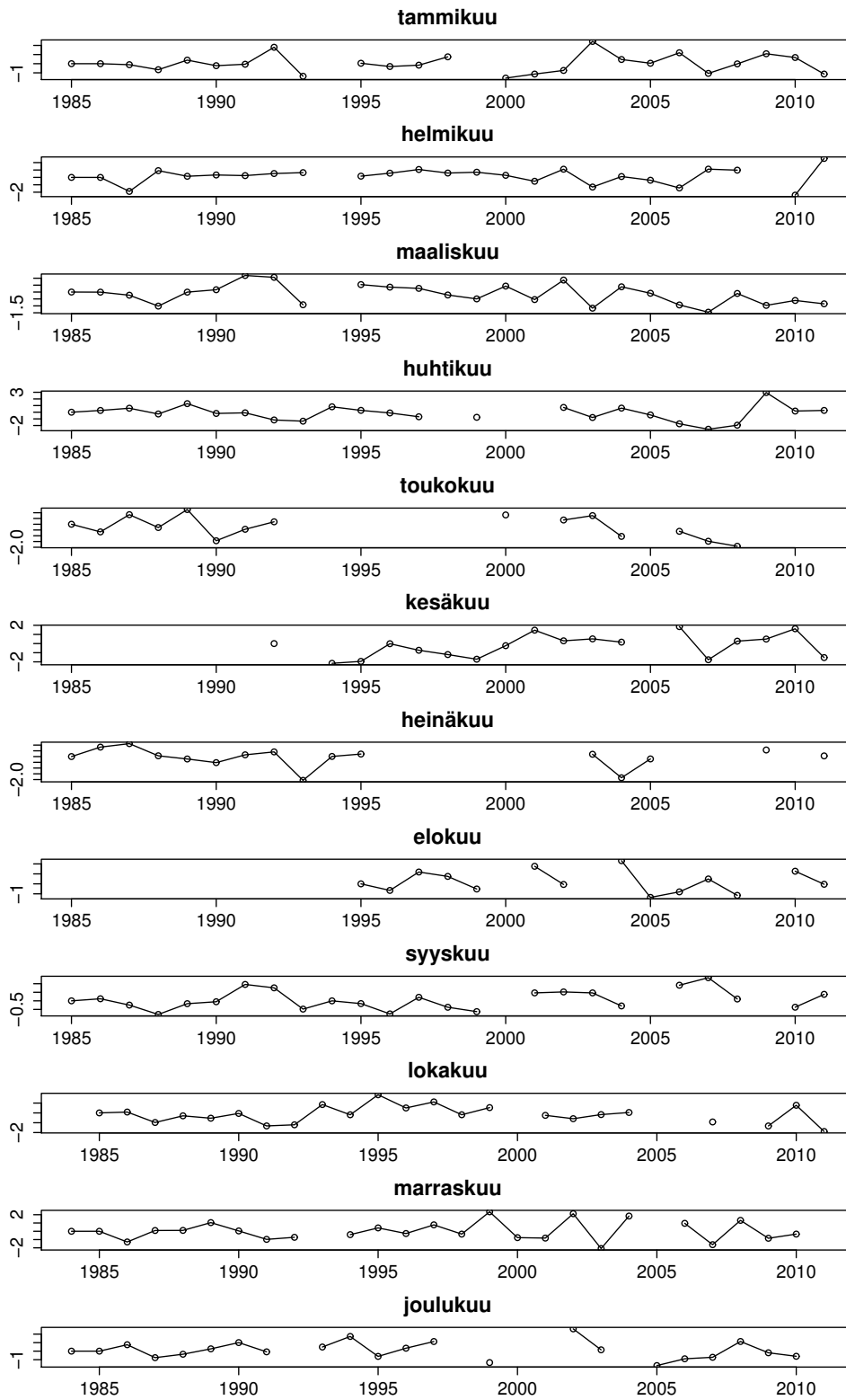
KUVA 3.16. Standardoitujen ennustevirheiden histogrammi ja $N(0,1)$ -normaalijakauman tiheysfunktio



KUVA 3.17. Teorettiset ja empiiriset kvantiilit ja $y = x$ -suora (punainen viiva)

TAULUKKO 3.2. Standardoidut ennustevirheet kuukausittain

kuukausi	keskiarvo	keskihajonta	varianssi
tammikuu	0.014	0.966	0.934
helmikuu	0.165	1.059	1.121
maaliskuu	-0.166	0.704	0.495
huhtikuu	-0.170	1.152	1.326
toukokuu	-0.243	0.962	0.925
kesäkuu	-0.257	1.253	1.569
heinäkuu	-0.077	0.838	0.701
elokuu	0.228	1.113	1.238
syyskuu	0.080	0.561	0.314
lokakuu	-0.181	0.880	0.775
marraskuu	0.023	1.127	1.270
joulukuu	0.107	1.015	1.031



KUVA 3.18. Standardoitut ennustevirheet kuukausittain

LUKU 4

Pohdintaa

Tässä tutkielmassa selvitettiin otsonin kokonaismäärän muutoksia ilmakehässä. Paranneltu malli esitettiin tila-avaruusmallien avulla. Malli sopii aineistoon hyvin, ja tulokset tukevat ja täydentävät hyvin aiempaa aiheesta tehtyä tutkimusta. Myös tässä tutkimuksessa havaittiin, että otsonimäärän laskeneen 1990-luvun alun ja vuoden 1997 välillä. Sen sijaan tammikuun 1997 ja joulukuun 2011 keskimääräisten tasojen välillä ei havaittu tilastollisesti merkitsevää eroa.

Jakauman simulointi paljasti, että varianssiparametrien suurimman uskottavuuden estimoinnissa jää huomioimatta paljon eri vaihtoehtoja, jotka ovat melkein yhtä hyviä kuin suurimman uskottavuuden estimaatit. Varianssiparametrien posteriorijakaumat olivat melko leveitä ja tämä näkyi tuloksissa suurehkona epävarmuutena. Tämä johtuu ainakin osittain siitä, että priorijakaumana käytettiin hyvin yksinkertaista, positiiviseksi rajoitettua tasajakaumaa. Muuttamalla tätä informatiivisemmaksi voitaisiin mahdollisesti saada tarkempia tuloksia.

Olisi kiinnostavaa tietää, miten ihmisen toiminta on vaikuttanut otsonimäärän kehittymiseen. Vaikka tässäkin tutkielmassa havaittiin otsonimäärän ensin laskeneen ja sittemmin nousseen, ei sillä voida suoraan tehdä kausaalipäätelyä siitä, että Montrealin pöytäkirjan perusteella tehdyt toimet olisivat pysäyttäneet otsonikadon. Kausaalipäätelyn tekeminen on haasteellista ja vaatisi huomattavan syvällistä ymmärrystä ilmakehätieteistä. Siksi tässä tutkielmassa keskityttiin tarkastelemaan, onko muutoksia ylipäätään tapahtunut.

Eräs mallin laajennusmahdollisuus olisi siirtyminen moniulotteisiin aikasarjoihin. Tällöin voitaisiin analysoida useampia korkeusalueita ja leveyspiirejä samalla mallilla sekä rakentaa riippuvuusrakente näiden väleille. Moniulotteisen aikasarjan rakentaminen on teknisesti varsin helppoa, mutta käytännön ongelmaiseksi tulee laskennan vaativuus. Jo vain kolmen aikasarjan analysointi samanaikaisesti kolminkertaistaisi uskottavuusfunktion ulottuvuuden ja mahdolliset riippuvuusrakenteet kasvattaisivat sitä entisestään, joten moniulotteisten aikasarjojen analysointi vaatisi monimutkaisempia menetelmiä. Koska otsonialueet ovat järjestyneet tilassa 2-ulotteisesti, on kyseessä spatiotemporaalinen ongelma. Mallin määrittävät matriisit ovat harvoja, joten laskentaa voidaan suorittaa tehokkaasti (ks. esim. Gelfand et al., 2010).

Tässä tutkielmassa esitetty otsonimäärän tila-avaruusmalli sovitaan kaikille yhdistetyn SAGE II / GOMOS -aikasarjan korkeusalueille ja leveyspiireille artikkelissa Laine, Latva-Pukkila & Kyrölä (2013).

Lähdeluettelo

- Angell, J. K. (1989), "On the Relation between Atmospheric Ozone and Sunspot Number", *Journal of Climate*, 2, 1404–1416.
- Angell, J. K. & Korshover, J. (1964), "Quasi-Biennial Variations in Temperature, Total Ozone, and Tropopause Height", *Journal of the Atmospheric Sciences*, 21, 479–492.
- Baldwin, M. P., Gray, L. J., Dunkerton, T. J., Hamilton, K., Haynes, P. H., Randel, W. J., Holton, J. R., Alexander, M. J., Hirota, I., Horinouchi, T., Jones, D. B. A., Kinniersley, J. S., Marquardt, C., Sato, K. & Takahashi, M. (2001), "The quasi-biennial oscillation", *Reviews of Geophysics*, 39, 179–229.
- Bertaux, J. L., Kyrölä, E., Fussen, D., Hauchecorne, A., Dalaudier, F., Sofieva, V., Tamminen, J., Vanhellemont, F., Fanton d'Andon, O., Barrot, G., Mangin, A., Blanot, L., Lebrun, J. C., Pérot, K., Fehr, T., Saavedra, L., Leppelmeier, G. W. & Fraisse, R. (2010), "Global ozone monitoring by occultation of stars: an overview of GOMOS measurements on ENVISAT", *Atmospheric Chemistry and Physics*, 10, 12091–12148.
- Bowman, K. P. (1989), "Global Patterns of the Quasi-biennial Oscillation in Total Ozone", *Journal of the Atmospheric Sciences*, 46, 3328–3343.
- Box, G. & Muller, M. (1958), "A Note on the Generation of Random Normal Deviates", *Annals of Mathematical Statistics*, 29, 610–611.
- Chatfield, C. (2003), *The Analysis of Time Series: An Introduction* (6th ed.), Chapman & Hall/CRC Texts in Statistical Science.
- Chu, W. P., McCormick, M. P., Lenoble, J., Brogniez, C. & Pruvost, P. (1989), "SAGE II inversion algorithm", *Journal of Geophysical Research: Atmospheres*, 94, 8339–8351.
- Cleveland, W. (1979), "Robust Locally Weighted Regression and Smoothing Scatterplots", *Journal of the American Statistical Association*, 74, 829–836.
- Durbin, J. & Koopman, S. J. (2012), *Time Series Analysis by State Space Methods* (2nd ed.), Oxford University Press.
- European Space Agency (2007), "GOMOS Product Handbook", Saataavilla Internetistä: <http://envisat.esa.int/handbooks/gomos/>, luettu 21.8.2012.
- Gamerman, D. (1997), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference* (2nd ed.), Chapman & Hall/CRC Texts

- in Statistical Science.
- Gelfand, A., Diggle, P., Guttorp, P. & Montserrat, F. (2010), *Handbook of Spatial Statistics*, Chapman & Hall/CRC.
- Gelman, A., Carlin, J., Stern, H. & Rubin, D. (2003), *Bayesian Data Analysis, Second Edition* (Chapman & Hall/CRC Texts in Statistical Science, 2nd ed.), Taylor & Francis.
- Green, P. & Mira, A. (2001), "Delayed Rejection in Reversible Jump Metropolis-Hastings", *Biometrika*, 88, 1035–1053.
- Haario, H., Laine, M., Mira, A. & Saksman, E. (2006), "DRAM: Efficient adaptive MCMC", *Statistics and Computing*, 16, 339–354.
- Haario, H., Saksman, E. & Tamminen, J. (2001), "An adaptive Metropolis algorithm", *Bernoulli*, 7, 97–109.
- Harvey, A. (1990), *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.
- Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications", *Biometrika*, 27, 97–109.
- Hathaway, D. H. (2010), "The Solar Cycle", *Living Reviews in Solar Physics*, 7, 1–65.
- Hauchecorne, A., Bertaux, J. L., Dalaudier, F., Keckhut, P., Lemenais, P., Bekki, S., Marchand, M., Lebrun, J. C., Kyrölä, E., Tamminen, J., Sofieva, V., Fussen, D., Vanhellemont, F., Fanton d'Andon, O., Barrot, G., Blanot, L., Fehr, T. & Saavedra de Miguel, L. (2010), "Response of tropical stratospheric O₃, NO₂ and NO₃ to the equatorial Quasi-Biennial Oscillation and to temperature as seen from GOMOS/ENVISAT", *Atmospheric Chemistry and Physics*, 10, 8873–8879.
- Jones, A., Urban, J., Murtagh, D. P., Eriksson, P., Brohede, S., Haley, C., Degenstein, D., Bourassa, A., von Savigny, C., Sonkaew, T., Rozanov, A., Bovensmann, H. & Burrows, J. (2009), "Evolution of stratospheric ozone and water vapour time series studied with satellite measurements", *Atmospheric Chemistry and Physics*, 9, 6055–6075.
- Kyrölä, E., Laine, M., Sofieva, V., Tamminen, J., Päivärinta, S.-M., Tukiainen, S., Zawodny, J. & Thomason, L. (2013), "Combined SAGE II-GOMOS ozone profile data set for 1984–2011 and trend analysis of the vertical distribution of ozone", *Atmospheric Chemistry and Physics*, 13, 10645–10658.
- Kyrölä, E., Tamminen, J., Sofieva, V., Bertaux, J. L., Hauchecorne, A., Dalaudier, F., Fussen, D., Vanhellemont, F., Fanton d'Andon, O., Barrot, G., Guirlet, M., Mangin, A., Blanot, L., Fehr, T., Saavedra de Miguel, L. & Fraisse, R. (2010), "Retrieval of atmospheric parameters from GOMOS data", *Atmospheric Chemistry and Physics*, 10, 11881–11903.
- Laine, M., Latva-Pukkila, N. & Kyrölä, E. (2013), "Analyzing time varying trends in stratospheric ozone time series using state

- space approach”, *Atmospheric Chemistry and Physics Discussions*, 13, 20503–20530.
- Lee, R. B., Gibson, M. A., Wilson, R. S. & Thomas, S. (1995), ”Long-term total solar irradiance variability during sunspot cycle 22”, *Journal of Geophysical Research: Space Physics*, 100, 1667–1675.
- Ljung, G. M. & Box, G. E. P. (1978), ”On a measure of lack of fit in time series models”, *Biometrika*, 65, 297–303.
- Lunn, D., Jackson, C., Spiegelhalter, D. J., Best, N. & Thomas, A. (2012), *The BUGS book: A practical introduction to Bayesian analysis*, CRC Press.
- Matsumura, Y. & Ananthaswamy, H. N. (2004), ”Toxic effects of ultraviolet radiation on the skin”, *Toxicology and Applied Pharmacology*, 195, 298–308.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953), ”Equation of State Calculations by Fast Computing Machines”, *Journal of Chemical Physics*, 21, 1087–1092.
- Miller, A. J., Hollandsworth, S. M., Flynn, L. E., Tiao, G. C., Reinsel, G. C., Bishop, L., McPeters, R. D., Planet, W. G., DeLuisi, J. J., Mateer, C. L., Wuebbles, D., Kerr, J. & Nagatani, R. M. (1996), ”Comparisons of observed ozone trends and solar effects in the stratosphere through examination of ground-based Umkehr and combined solar backscattered ultraviolet (SBUV) and SBUV 2 satellite data”, *Journal of Geophysical Research: Atmospheres*, 101, 9017–9021.
- Mira, A. (2001), ”On Metropolis-Hastings algorithms with delayed rejection”, *Metron*, 59, 3–4.
- National Aeronautics and Space Administration (2012), ”SAGE II website”, Saatavilla Internetistä: <http://www-sage2.larc.nasa.gov/>, luettu 21.8.2012.
- Newchurch, M. J., Yang, E.-S., Cunnold, D. M., Reinsel, G. C., Zawodny, J. M. & Russell, J. M. (2003), ”Evidence for slowdown in stratospheric ozone loss: First stage of ozone recovery”, *Journal of Geophysical Research: Atmospheres*, 108, 12:1–12:13.
- Petris, G. (2010), ”An R Package for Dynamic Linear Models”, *Journal of Statistical Software*, 36, 1–16.
- Petris, G., Petrone, S. & Campagnoli, P. (2009), *Dynamic Linear Models with R*, Springer.
- R Development Core Team (2011), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. Saatavilla Internetistä: <http://www.R-project.org/>.
- Randel, W. J. & Cobb, J. B. (1994), ”Coherent variations of monthly mean total ozone and lower stratospheric temperature”, *Journal of Geophysical Research: Atmospheres*, 99, 5433–5447.
- Särkkä, S. (2013), *Bayesian Filtering and Smoothing*, Cambridge University Press.

- Tamminen, J., Kyrölä, E., Sofieva, V. F., Laine, M., Bertaux, J.-L., Hauchecorne, A., Dalaudier, F., Fussen, D., Vanhellemont, F., Fanton-d'Andon, O., Barrot, G., Mangin, A., Guirlet, M., Blanot, L., Fehr, T., Saavedra de Miguel, L. & Fraisse, R. (2010), "GOMOS data characterisation and error estimation", *Atmospheric Chemistry and Physics*, 10, 9505–9519.
- The MathWorks Inc. (2013), *MATLAB version 8.1.0.604*, Natick, Massachusetts.
- Tierney, L. & Mira, A. (1999), "Some adaptive Monte Carlo methods for Bayesian inference", *Statistics in Medicine*, 18, 2507–2515.
- World Meteorological Organization (2011), "Twenty Questions and Answers about the Ozone Layer: 2010 Update", teoksessa *Scientific Assessment of Ozone Depletion: 2010*, toim. D. W. Fahey & M. I. Hegglin, World Meteorological Organization.

LIITE A

R-koodi

Tässä esitetään oleelliset tutkimuksessa käytetyt R-koodit. R-ohjelmisto on vapaasti saatavilla (R Development Core Team, 2011).

```
#mvtnorm library is necessary for the DRAM function
library("mvtnorm")

DRAM<-function(logDens,X0,C0=diag(length(X0)),nSim=2000,
               n0=500,t0=300,gamma=0.01,printResults=T){
  #This is a general-purpose simulation function implementing
  #the DRAM algorithm as proposed by
  #Haario, H., Laine, M., Mira, A. & Saksman, E. in
  #"DRAM: Efficient adaptive MCMC", published in 2006 by
  #Statistics and Computing, Vol. 16, pages 339-354
  #
  #Parameters:
  #logDens:      logarithmic likelihood function
  #X0:          starting point for the algorithm
  #C0:          initial covariance matrix for the proposals
  #nSim:        number of simulations
  #n0:          how often the adaptation will take place
  #t0:          number of simulations before the first adaptation
  #gamma:       scaling factor for the DR part
  #printResults: should a summary be printed?
  #
  #Value:
  #A named list containing:
  #X:           The simulated chain
  #dens:        Values of the density function
  #acc:         Logical matrix containing information about acceptances.
  #             The first column means acceptance on the first
  #             trial (TRUE/FALSE) and the second column
  #             acceptance on the second trial (TRUE/FALSE)
  #C:           The final covariance matrix of proposals
  #
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 2nd, 2013
```

```

#

#Simulate all probabilities for acceptance at once
accLimits<-runif(nSim)
#d is the dimension of parameter space
d<-length(X0)
#X is the chain
X<-matrix(mat.or.vec(nSim,d),ncol=d)
X[1,]<-X0
#densValues includes the values of the density function
densValues<-mat.or.vec(nSim,1)
#Adaptation has a index of it's own
iOfAdaptation<-1
#Initialize the acceptance counters for
#total acceptance and DR acceptances
nAcc<-0
nAccDR<-0
#This is a logical matrix to include information
#about acceptances. The first column means acceptance
#on the first trial and the second column acceptance
#on the second (DR) trial.
accMatrix<-mat.or.vec(nSim,2)

#Initialize the DR counter
nDR<-0

#Calculate the log density of the initial value
llOld<-logDens(X0)
#and save the density:
densValues[1]<-exp(llOld)
#Simulate the first increases
Ct<-C0
propIncreases<-rmvnorm(n=max(n0,t0),mean=rep(0,d),sigma=Ct)

#Open a text progress bar
pb<-txtProgressBar(min=1,max=nSim,style=3)

#Here begins the actual simulation loop
for (i in 2:nSim){
  #Update the progress bar only every 100th iteration
  if (i%%100==0){
    setTxtProgressBar(pb,i)
  }
  if ( (iOfAdaptation == n0 && i>t0)|(i==max(n0,t0))){
    #Adapt

```

```

Ct<-cov(matrix(X[1:i,],ncol=d))*(2.4^2)/d
#Simulate all increases at once
propIncreases<-rmvnorm(n=n0,mean=rep(0,d),sigma=Ct)
#A new loop of adaptation begins:
iOfAdaptation<-1
}
#Pick the old value
old<-X[i-1,]
#Calculate the proposal value
prop<-old+propIncreases[iOfAdaptation,]
#Calculate the log density of the proposal value
llProp<-logDens(prop)
#Calculate the acceptance probability
rxy<-exp(llProp-llOld)
alpha<-min(1,rxy)

if (accLimits[i]<alpha){
  #Accepted, update the location
  X[i,]<-prop
  #Update also llOld
  llOld<-llProp
  #Update the acceptance matrix
  accMatrix[i,1]<-1
  #save the density
  densValues[i]<-exp(llProp)
}
else {
  #Rejection, here begins the DR part

  #Pick a new proposal
  secondProp<-rmvnorm(n=1,mean=old,sigma=gamma*Ct)
  llSecondProp<-logDens(secondProp)

  if(llSecondProp==-Inf){
    #The new proposal impossible, never accept
    rxy2<-0
  }
  else {
    #Calculate the acceptance probability
    alphaTemp<-min(1,exp(llProp-llSecondProp))

    rxy2<-exp(llSecondProp-llOld)*
      (dmvnorm(x=prop,mean=secondProp,sigma=Ct)/
       dmvnorm(x=prop,mean=old,sigma=Ct))*
      ((1-alphaTemp)/(1-alpha))
  }
}

```

```

    }
    alpha2<-min(1,rx2)
    if (runif(1)<alpha2){
      #Accepted on the second attempt
      X[i,]<-secondProp
      #Update also llOld
      llOld<-llSecondProp
      #Update the acceptance matrix
      accMatrix[i,2]<-1
      #save the density
      densValues[i]<-exp(llSecondProp)
    }
    else {
      #Rejection, the chain stays where it was
      X[i,]<-old
      #llOld stays as it was
      #save the density
      densValues[i]<-exp(llOld)
    }
  }
  #Update the index within this adaptation loop
  iOfAdaptation<-iOfAdaptation+1
}
if (printResults){
  #Print the results
  cat("\nDRAM simulation complete!")
  cat("\n",nSim,
      " simulations ran with an acceptance rate of",
      sum(accMatrix[,1]|accMatrix[,2])/nSim)
  cat("\nDR was necessary ",sum(!accMatrix[,1]),
      " times and it was successful ",sum(accMatrix[,2])," times")
  cat("\nThe final covariance matrix for proposals was \n")
  print(Ct)
}
#return the key results
return(list(X=X,dens=densValues,acc=accMatrix,C=Ct))
}

thinChain<-function(chain,nThin){
  #This function extracts every nThin'th
  #element of a matrix and returns
  #this thinned chain
  #
  #Parameters:
  #chain:      the chain to be thinned

```

```

#nThin:      pick every nThin'th element
#
#Value:
#The thinned chain
#
#Author: Niilo Latva-Pukkila
#Version: Dec 17th, 2013
#
return(chain[seq(1,length(chain[,1]),nThin),])
}

modifyDlm<-function(par,winds,mod){
  #This function modifies a dlm object
  #and returns the modified object
  #
  #Parameters:
  #par:      New variance parameters for the dlm object
  #winds:    indices of the diagonal parameters of W
  #          (see example below)
  #mod:      the dlm object to be modified
  #
  #Value:
  #The modified model
  #
  #Example:
  #Consider you want to optimize the parameters
  #at W[2,2],W[3,3],W[4,4] and V and that W[3,3] and W[4,4]
  #must be the same.
  #Now par would include the parameters in the same order
  #and winds should be a vector including the indices of
  #the diagonal parameters, i.e. in this case
  #winds<-c(0,1,2,2).
  #If par is longer than max(winds), the last value of par
  #is interpreted as V (or a multiplier of time-varying V,
  #if JV is defined).
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 2nd, 2013
  #

  #Place the variance parameters of W:
  for (i in 1:length(winds)){
    if (winds[i]>0){
      mod$W[i,i]<-par[winds[i]]
    }
  }
}

```

```

}
#Place the observation variance parameter (if asked for)
if (max(winds)<length(par)){
  if (is.null(mod$JV)){
    #The observation variances do equal
    mod$V<-matrix(par[max(winds)+1])
  }
  else {
    #The obs variances do not equal
    mod$X[,mod$JV]<-par[max(winds)+1]*mod$X[,mod$JV]
  }
}
#Return the modified dlm object
return(mod)
}

```

```

#dlm library is necessary for the logDensKF function
library("dlm")

```

```

logDensKF<-function(par,winds,y,mod){
  #This function is meant to be a link between
  #the dlm package and DRAM function.
  #The dlm object is first modified
  #using modifyDlm and then the log likelihood
  #of the modified model is calculated using
  #the dlm package.
  #
  #Parameters:
  #par:          New variance parameters for the dlm object
  #winds:        indices of the diagonal parameters of W
  #              (see example in modifyDlm function)
  #y:            observations
  #mod:          the dlm object to be modified
  #
  #Value:
  #Log-likelihood of the modified model
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 2nd, 2013
  #
  #Never accept negative variances
  if (min(par)<0){
    return(-Inf)
  }
}

```

```

mod<-modifyDlm(par,winds,mod)
return(-1*dmlLL(y,mod))
}

sampleStates<-function(nSamples=100,burnin=0,
                       nThin=1,chain,winds,y,mod){
  #This function simulates samples from the conditional
  #distribution (cond. under the variance parameters)
  #of the states using FFBS algorithm.
  #
  #Parameters:
  #nSamples:   how many samples to simulate
  #burnin:    where to start using the chain
  #nThin:     use only nThin'th value of the chain
  #chain:     MCMC chain of the variance parameters
  #winds:     indices of the diagonal parameters of W
  #           (see example in modifyDlm function)
  #y:         observations
  #mod:       the dlm object to be modified
  #
  #Value:
  #A 3-d array of simulated states
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 2nd, 2013
  #

  #check that the chain is long enough:
  if(burnin+nSamples*nThin>length(chain[,1])){
    errorMsg<-c("The chain is not long enough! ",
               "Consider choosing smaller nSamples, burnin or nThin.")
    stop(errorMsg)
  }

  #initialize the sample array
  samples<-array(0,dim=c(length(y)+1,length(mod$m0),nSamples))
  #Open a text progress bar
  pb<-txtProgressBar(min=1,max=nSamples,style=3)

  for (i in 1:nSamples){
    #Update the progress bar only every 20th iteration
    if (i%%20==0){
      setTxtProgressBar(pb,i)
    }
    #modify the model

```

```

    modTemp<-modifyDlm(chainOut[burnin+i*nThin,],winds,mod)
    #filter
    filteredTemp<-dlmFilter(y,modTemp)
    #apply FFBS to simulate the states
    samples[, ,i]<-dlmBSample(filteredTemp)
  }
  #return the samples
  return(samples[-1, ,])
}

plotSamples<-function(samples,firstTime,ylab="",
                      nToPlot=dim(samples[2])){
  #This function is used to plot the samples
  #with median level and 95% CI
  #
  #Parameters:
  #samples:      samples of state
  #firstTime:    when the ts starts
  #ylab:         label of the y axis (optional)
  #nToPlot:     how many realizations to
  #              plot (optional, default = all)
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 20th, 2013
  #
  nSamples<-dim(samples)[2]
  if (nToPlot<=0){
    #requested number of states was not-positive, so use
    #a different function
    plotOnlyQuantiles(samples,firstTime,ylab,-1)
    return()
  }
  samplesThinned<-samples[,seq(1,nSamples,floor(nSamples/nToPlot))]
  #Plot the first sample
  plot(ts(samplesThinned[,1],start=firstTime,freq=12),
       type="l",ylim=range(samplesThinned[,]),
       ylab=ylab,xlab="Aika",lwd=0.25)
  #...and all the rest
  for (i in 2:nToPlot){
    lines(ts(samplesThinned[,i],start=firstTime,freq=12,
            ),type="l",lwd=0.25)
  }
  #Calculate quantiles of the state:
  sampleQuantiles<-mat.or.vec(length(samples[,1]),3)
  quantileProbs=c(0.025,0.5,0.975)

```



```

for (i in 1:length(samples[,1])){
  sampleQuantiles[i,]<-quantile(samples[i,],
                                probs=quantileProbs)
}
#Plot the median level and 95% credible interval
lines(ts(sampleQuantiles[,1],start=firstTime,freq=12),
      col=2,lwd=2)
lines(ts(sampleQuantiles[,2],start=firstTime,freq=12),
      col=3,lwd=2)
lines(ts(sampleQuantiles[,3],start=firstTime,freq=12)
      ,col=2,lwd=2)
}
plotOnlyQuantiles<-function(samples,firstTime,ylab="",ylim=-1){
  #This function is used to plot only
  #the quantiles and the median of
  #the sampled states
  #
  #Parameters:
  #samples:      samples of state
  #firstTime:    when the ts starts
  #ylab:         label of the y axis (optional)
  #
  #Author: Niilo Latva-Pukkila
  #Version: Dec 20th, 2013
  #
  nSamples<-dim(samples)[2]

  #Calculate quantiles of the state:
  sampleQuantiles<-mat.or.vec(length(samples[,1]),3)
  quantileProbs=c(0.025,0.5,0.975)
  for (i in 1:length(samples[,1])){
    sampleQuantiles[i,]<-quantile(samples[i,],
                                  probs=quantileProbs)
  }
  if (ylim[1]==-1){
    ylim=range(sampleQuantiles[,])
  }
  #Plot the median level and 95% credible interval
  plot(ts(sampleQuantiles[,1],start=firstTime,freq=12),
       ,ylim=ylim,
       ylab=ylab,xlab="Aika",lty=2,lwd=1)
  lines(ts(sampleQuantiles[,2],start=firstTime,freq=12),
        lty=1,lwd=1)
  lines(ts(sampleQuantiles[,3],start=firstTime,freq=12)
        ,lty=2,lwd=1)
}

```

```

}

#####
# Everything below is problem specific!  #
# Here we apply the DRAM algorithm in    #
# ozone modelling.                       #
#####

library(mvtnorm)
library(dlm)
library(zoo)

#Get the proxies (solar and qbo at 10 and 30 HPa)
proxies<-read.csv("graduProxies19Sep2013.csv",header=F)

#Standardize the proxies (for numerical stability)
proxiesScaled<-array(dim=dim(proxies))
proxiesScaled[,1]<-(proxies[,1]-mean(proxies[,1]))/
  (sd(proxies[,1]))
proxiesScaled[,2]<-(proxies[,2]-mean(proxies[,2]))/
  (sd(proxies[,2]))
proxiesScaled[,3]<-(proxies[,3]-mean(proxies[,3]))/
  (sd(proxies[,3]))
proxiesScaled<-as.matrix(proxiesScaled)

#The data start from Oct 1984:
firstTime<-c(1984,10)

#Create time series of the proxies
proxiesTsScaled<-ts(proxiesScaled,start=firstTime,freq=12)

#Get the ozone data and uncertainty estimates
y<-read.csv("graduY.csv")$x
s<-read.csv("graduS.csv")$x

#Create time series of the ozone data and the uncertainties
yTs<-ts(y,start=firstTime,freq=12)
sTs<-ts(s,start=firstTime,freq=12)

#Standardize the ozone data and the uncertainties
yMean<-mean(y,na.rm=T)
ySd<-sd(y,na.rm=T)
yTsScaled<-(yTs-yMean)/ySd
sTsScaled<-sTs/ySd

```

```

#Create the model
O3Mod <- dlmModPoly(order=2)+dlmModTrig(s=12,q=6)+
  dlmModReg(proxiesTsScaled,addInt=FALSE)
#Input the uncertainties
X(O3Mod)<-cbind(proxiesTsScaled,sTsScaled^2)
O3Mod$JV<-matrix(c(4))

#Define a function for ML estimation
buildFunOzone <- function(x){
  diag(O3Mod$W) [2]<-exp(x[1])
  diag(O3Mod$W) [3:13]<-exp(x[2])
  diag(O3Mod$W) [14:16]<-exp(x[3:5])
  O3Mod$X[,4]<-O3Mod$X[,4]*exp(x[6])
  return(O3Mod)
}

#Find ML estimates
fit <- dlmMLE(yTsScaled, parm = rep(-4.5,6),
             build = buildFunOzone,
             control=list(trace=1))
MLE<-exp(fit$par)

#Define the variables for DRAM:
windsOzone<-c(0,1,rep(2,11),3,4,5)
logDensOzone<-function(par){
  return(logDensKF(par,windsOzone,yTsScaled,O3Mod))
}

#Use the ML estimates as initial value
X0Ozone<-MLE
#The initial proposal variance
C0Ozone<-diag(X0Ozone)*0.1

#Run the simulations
dramOutOzone<-DRAM(logDens=logDensOzone,X0=X0Ozone,
                  CO=C0Ozone,nSim=5e5,n0=2000,t0=500)

#Pick the chain
chainOut<-dramOutOzone$X

#Name the parameters
parNames<-c("sigma^2_alpha","sigma^2_zeta",
            "sigma^2_beta1","sigma^2_beta2","sigma^2_beta3",
            "sigma^2_epsilon")
d<-dim(chainOut) [2]

```

```

par(mfrow=c(d,2))
#\textheight=50.5pc \topskip=10pt
#\textwidth=30pc
#Plot the chains and acf's
for (i in 1:d){
  plot(chainOut[,i],type="l",
        #cex.axis=1.2, cex.main=1.5,cex.lab=1.3,
        ylab=parNames[i],
        xlab="Simulointikierros")
  acf(chainOut[,i],
        xlab="Viive",main="")
}

#Thin the chain
chainThinned<-thinChain(chainOut,50)
par(mfrow=c(6,2))
#Plot the chains and acf's
for (i in 1:d){
  plot(chainThinned[,i],type="l",
        #cex.axis=1.2, cex.main=1.5,cex.lab=1.3,
        ylab=parNames[i],
        xlab="Simulointikierros")
  acf(chainThinned[,i],
        xlab="Viive",main="")
}

par(mfrow=c(d,1))
#Define burn-in period (based on visual inspection)
burnin<-100
#Plot histograms
par(mar=c(5.1,4.5,4.1,2.1))
for (i in 1:d){
  hist(chainThinned[-(1:burnin),i],40,
        #cex.axis=1.2, cex.main=1.5,cex.lab=1.3,
        xlab=parNames[i],
        ylab="Frekvenssi",
        main="")
  abline(v=X00zone[i],col=2)
}
par(mar=c(5.1,4.1,4.1,2.1))
#Calculate mean values
colMeans(chainThinned[-(1:burnin),])

#Calculate quantiles
quantileProbs<-c(0.025,0.5,0.975)

```

```

for (i in 1:d){
  print(quantile(chainOut[-(1:burnin),i],
                probs=quantileProbs))
}

#Sample states using the chain
nSamples<-10000
sampled<-sampleStates(nSamples=nSamples,
                     burnin=0,nThin=1,
                     chain=chainThinned,
                     winds=windsOzone,
                     y=yTsScaled,
                     mod=O3Mod)

#Unscale:
levelSamples<-sampled[,1,]*ySd+yMean
#Plot the samples of the first state (level)
plotSamples(levelSamples,firstTime,ylab="Otsoni",500)

#Plot the regression coefficients:
par(mfrow=c(3,1))
plotSamples(sampled[,14,],firstTime,"beta1 (SOLAR)",0)
plotSamples(sampled[,15,],firstTime,"beta2 (QB01)",0)
plotSamples(sampled[,16,],firstTime,"beta3 (QB02)",0)

#Calculate the seasonal effect:
seasInd<-c(3,5,7,9,11,13)
seasTotal<-array(dim=dim(sampled[,3,]))
seasTotal[,]<-0
for (i in 1:length(seasInd)){
  seasTotal<-seasTotal+sampled[,seasInd[i],]
}
#Plot all model components:
par(mfrow=c(5,1))
plotOnlyQuantiles(sampled[,1,]*ySd+yMean,firstTime,
                  "Taso",ylim=yMean+c(-1e12,1e12))
plotOnlyQuantiles(sampled[,14,]*O3Mod$X[,1]*ySd,firstTime,
                  "SOLAR",ylim=c(-1e12,1e12))
plotOnlyQuantiles(sampled[,15,]*O3Mod$X[,2]*ySd,firstTime,
                  "QB01",ylim=c(-1e12,1e12))
plotOnlyQuantiles(sampled[,16,]*O3Mod$X[,3]*ySd,firstTime,
                  "QB02",ylim=c(-1e12,1e12))
plotOnlyQuantiles(seasTotal*ySd,firstTime,
                  "Kausivaihtelu",ylim=c(-1e12,1e12))

#Calculate statistics of interest:

```

```

#Jan 1990 is the 74th observation
#Jan 1997 is the 148th observation
#difference in mean level between 1997 and 1991:
diff1<-(levelSamples[148,]-levelSamples[64,])/
  (levelSamples[64,])
hist(diff1,
      xlab="Estimoitu suhteellinen muutos 01-1990 ja 01-1997 välillä",
      ylab="Frekvenssi",main="")
diff2<-(levelSamples[327,]-levelSamples[148,])/(levelSamples[148,])
hist(diff2,
      xlab="Estimoitu suhteellinen muutos 01-1997 ja 12-2011 välillä",
      ylab="Frekvenssi",main="")

#Probability that the first trend was not negative:
mean(diff1>=0)
quantile(diff1,probs=c(0.025,0.975))*100
#Probability that the second trend was not positive:
mean(diff2>0)
quantile(diff2,probs=c(0.025,0.975))

#Find MAP estimates:
(MAP<-chainOut[which.max(dramOutOzone$dens),])
#Define a model with MAP parameters
O3ModMAP<-modifyDlm(MAP,windsOzone,O3Mod)

#Filter and smooth
fil<-dlmFilter(yTsScaled,O3ModMAP)
smo<-dlmSmooth(yTsScaled,O3ModMAP)

#Length of the time series
n<-length(yTsScaled)

#Initialize vectors
levelsScaled<-mat.or.vec(n,1)
upperLimitsScaled<-mat.or.vec(n,1)
lowerLimitsScaled<-mat.or.vec(n,1)
smoothedValuesScaled<-mat.or.vec(n,1)
stdOflevels<-mat.or.vec(n,1)

#Calculate the variances of states
variances <- dlmSvd2var(smo$U.S, smo$D.S)

#Calculate the smoothed states
Fmat <- FF(O3Mod)
for (i in 1:n) {

```

```

Fmat[JFF(O3Mod)>0]<-
  X(O3Mod)[i,JFF(O3Mod)]
smoothedValuesScaled[i]<-Fmat%*(smo$s[i,])
levelsScaled[i]<-smo$s[i,1]
stdOflevels[i] <- sqrt(variances[[i]][1,1])
}
#Unscale the values
smoothedValues<-smoothedValuesScaled*ySd+yMean
levels<-levelsScaled*ySd+yMean
lowerLimits<-levels-ySd*qnorm(0.975)*stdOflevels
upperLimits<-levels+ySd*qnorm(0.975)*stdOflevels

#Make time series to help plotting
smoothedValuesTs<-ts(smoothedValues,freq=12,start=firstTime)
levelsTs<-ts(levels,freq=12,start=firstTime)
lowerLimitsTs<-ts(lowerLimits,freq=12,start=firstTime)
upperLimitsTs<-ts(upperLimits,freq=12,start=firstTime)

#Plot the observations and the fitted values from
#the smoother
plot(yTs,,type="p",ylab="Otsoni",pch=20,
      xlab="Aika",)
lines(smoothedValuesTs,col="blue")

#####
#Model diagnostics:
#####

#Extract the filter residuals
filRes<-residuals(fil,type="standardized")$res
#Plot them with the LOESS curve
scatter.smooth(filRes,
               ylab="Standardoitu ennustevirhe",
               xlab="Aika",type="l",col=2)
#Plot a histogram and a density curve of
#the standard gaussian
hist(filRes,15,
     main="",
     xlab="Standardoitu ennustevirhe",
     ylab="Suhteellinen osuus",freq=F)
xGrid<-seq(-5,5,length.out=200)
lines(xGrid,dnorm(xGrid))

#QQ plot
qqnorm(filRes,

```

```

        xlab="Teoreettinen kvantiili",
        ylab="Empiirinen kvantiili",main="")
lines(xGrid,xGrid,col=2)

#ACF
acf(as.numeric(filRes),
    na.action=na.pass,xlab="Viive (kk)",main="")

monthNames<-c("tammikuu","helmikuu",
              "maaliskuu","huhtikuu",
              "toukokuu","kesäkuu",
              "heinäkuu","elokuu",
              "syyskuu","lokakuu",
              "marraskuu","joulukuu")
monthlyMeans<-mat.or.vec(12,1)
monthlyVariances<-mat.or.vec(12,1)

for (i in 1:12){
  monthTemp<-subset(filRes,cycle(filRes)==i)
  monthlyMeans[i]<-mean(monthTemp,na.rm=T)
  monthlyVariances[i]<-var(monthTemp,na.rm=T)
}

#Display monthly means, sd's and variances
round(cbind(monthlyMeans,
            sqrt(monthlyVariances),
            monthlyVariances),3)

#Plot the filter residuals by month
opar <- par()
par(mfrow=c(6,2))
par(mar=c(2,2,3,1))
for (i in 1:12){
  if (i>9){
    xAxis<-1984:2011
  }
  else {
    xAxis<-1985:2011
  }
  monthTemp<-subset(filRes,cycle(filRes)==i)
  plot(xAxis,monthTemp,xlab="Vuosi",ylab="",main=monthNames[i])
  lines(xAxis,monthTemp)
}
par(opar)

```