

Laajennettujen Kalman-suotimien soveltaminen epäkoherentin sironnan spektritiheysfunktion estimoinnissa

Pro Gradu -tutkielma

Karjalainen Joni

2112129

Matemaattisten tieteiden laitos

Oulun yliopisto

Syksy 2016

Sisältö

Johdanto	2
1 Stokastiset prosessit	3
2 Lineaarinen Kalman-suodin	7
2.1 Tila-avaruuusmallit	8
2.2 Kalman-suotimen johto	9
3 Kalman-suotimen laajennuksia	12
3.1 Laajennettu Kalman-suodin	12
3.2 Hajustamaton Kalman-suodin	18
3.3 Hiukkassuodin	21
4 Spektritiheysfunktion estimointi	24
4.1 Ongelman muotoilu	24
4.2 Levenberg-Marquardt -menetelmä	27
4.3 Tulokset	29
Lähdeluettelo	40
Liitteet	41
A EKF	41
B EKF2	44
C UKF	48
D BF	54
E LM	57

Johdanto

Tutkimusongelmana tälle työlle on soveltaa laajennettuja Kalman-suotimia ionosfääriin liittyvän käänteisen sirontaongelman parametrien estimointiin.

Ionosfääristä antennilla vastaanotetusta signaalista voidaan muodostaa autokovarianssi- ja spektritiheysfunktiot. Näistä voidaan saada selville erilaisia ionosfäärin ominaisuuksia. Tässä työssä rajoitutaan tarkastelemaan vain spektritiheysfunktioita.

Vastaanotetun signaalin spektritiheysfunktion plasmaparametrit muuttuvat eri ionosfäärin korkeuksilla. Tutkimusaiheena on käyttää lineaarisen Kalman-suotimen laajennuksia näiden parametrien estimointiin. Spektritiheysfunktioita simuloidaan aidon datan sijaan Lorentz-käyrien avulla. Suotimien toimivuutta verrataan aiemmin estimoinnissa käytettyyn Levenberg-Marquardt-menetelmään.

Työn runkona on käytetty lähdettä [6]. Matemaattiset merkinnät on pyritty pitämään pääosin samoina tämän lähteen kanssa. Liitteenä on estimoinnin tarpeisiin muokattujen menetelmien R-koodit.

Kiitokset Sodankylän Geofysikaalisen Observatorion väelle, joka auttoi tämän Pro Gradu -tutkielman tekemisessä.

1 Stokastiset prosessit

Tässä kappaleessa käydään läpi stokastisiin prosesseihin, erityisesti Markovin ketjuihin liittyvää teoriaa. Tämä kappale sisältää esitietoja eri Kalman-suotimiin liittyen. Kappale on koottu lähteistä [2] [5], ja [6].

Määritelmä 1.1. Olkoon annettu todennäköisyysavaruus $(\Omega, \mathcal{F}, \mathbb{P})$ ja mitallinen avaruuus (S, Σ) . Olkoon T täysin järjestetty indeksijoukko. *S-arvoisen stokastisen prosessi \mathbf{X}* on perhe $\{\mathbf{X}_k : k \in T\}$ S -arvoisia satunnaismuuttuja avaruudessa Ω . Joukkoa S kutsutaan stokastisen prosessin \mathbf{X} *tila-avaruudeksi* ja sen alkioita $\mathbf{x} \in \mathbf{X}$ *tiloiksi*.

Mikäli indeksijoukko T on diskreetti, sanotaan stokastisen prosessin olevan *diskreettiaikainen*, ja jos T on jatkuva, sanotaan stokastisen prosessin olevan *jatkuva-aikainen*. Stokastisen prosessin realisaatiota $\{\mathbf{x}_{k_0}, \mathbf{x}_{k_1}, \dots\}$, kutsutaan *aikasarjaksi*.

Joukko T tulkitaan usein aikapisteiden jouokksi. Merkitään jatkossa äärellistä aikasarjaa $\mathbf{x}_{k_0:k_n} := \{\mathbf{x}_{k_0}, \dots, \mathbf{x}_{k_n}, k_n < \infty\}$ yksinkertaisuuden vuoksi. Käytetään merkintää

$$p(\mathbf{x}_{k_0:k_n}) := p(\mathbf{X}_{k_0} = \mathbf{x}_{k_0}, \dots, \mathbf{X}_{k_n} = \mathbf{x}_{k_n})$$

tiheysfunktion p arvolle tiloissa $\mathbf{x}_{k_0}, \dots, \mathbf{x}_{k_n}$, ja merkintää

$$p(\mathbf{x} \mid \mathbf{x}_{k_0:k_n}) := p(\mathbf{X} = \mathbf{x} \mid \mathbf{X}_{k_0} = \mathbf{x}_{k_0}, \dots, \mathbf{X}_{k_n} = \mathbf{x}_{k_n})$$

ehdoliselle tiheysfunktiolle.

Tässä työssä keskitytään jatkossa tarkastelemaan diskreettiaikaisia stokastisia prosesseja.

Määritelmä 1.2. Olkoon \mathbf{X} stokastinen prosessi, T siihen liittyvä diskreetti indeksijoukko ja $k, l \in T$. Stokastisen prosessin \mathbf{X} *autokovarianssfunktio* on

$$C(k, l) := \text{Cov}(\mathbf{X}_k, \mathbf{X}_l) = \mathbb{E}[(\mathbf{X}_k - \mathbb{E}[\mathbf{X}_k])(\mathbf{X}_l - \mathbb{E}[\mathbf{X}_l])]$$

olettuen, että vaaditut odotusarvot ovat olemassa.

Kyseessä on siis stokastisen prosessin \mathbf{X} kovarianssi itsensä kanssa kaikilla ajanhetkillä.

Määritelmä 1.3. Olkoon diskreettiaikaisen stokastisen prosessin \mathbf{X} autokovarianssifunktio $C(k, l)$ hyvin määritelty ja

$$\sum_{k,l \in T} |C(k, l)| < \infty.$$

Tällöin voidaan määritellä stokastisen prosessin \mathbf{X} *spektritiheysfunktio* muunoksena

$$S(f) = \frac{1}{2\pi} \sum_{k,l \in T} C(k, l) e^{-i(k-l)f},$$

missä $-\infty < f < \infty$ on taajuus ja i on imaginaariyksikkö.

Kyseessä on siis autokovarianssifunktion diskreetti Fourier-muunnos. Autokovarianssifunktion avulla aikasarjaa tarkastellaan aikaulottuvuuden kautta, kun taas spektritiheysfunktiota tarkastellaan taajuusulottuvuuden kautta. Kyseessä ovat siis eri tavat karakterisoida aikasarja. Lisää tietoa spektritiheysfunktioista saa lähteestä [2].

Spektritiheysfunktioita tarkastellaan tarkemmin kappaleessa 4, missä esitoidaan plasmaparametrejä laajennetuilla Kalman-suotimilla.

Lause 1.4. Olkoot $\mathbf{x}_0, \dots, \mathbf{x}_k$ stokastisen prosessin \mathbf{X} tiloja. Tällöin

$$p(\mathbf{x}_{0:k}) = p(\mathbf{x}_0)p(\mathbf{x}_2 \mid \mathbf{x}_1) \cdots p(\mathbf{x}_k \mid \mathbf{x}_{1:k-1}), \quad (1)$$

oleettaen että ylläolevat ehdolliset todennäköisyydet on hyvin määritelty.

Yhtälöä (1) kutsutaan *kertolaskukaavaksi*.

Määritelmä 1.5. Diskreettiaikaista stokastista prosessia $\{\mathbf{X}_k : k = 0, 1, \dots\}$ kutsutaan *1. kertaluvun Markovin ketjuksi* mikäli kaikilla $k \in \mathbb{N}$ ja tiloilla $(\mathbf{x}_0, \dots, \mathbf{x}_k) \in \mathbf{S}^{k+1}$ pätee

$$p(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}), \quad (2)$$

mikäli kyseiset ehdolliset todennäköisyydet ovat olemassa.

Jatkuva-aikaista stokastista prosessia $\{\mathbf{X}_k : k \geq 0\}$, joka toteuttaa ehdon (2) kutsutaan *Markovin jonoksi*. Sitä, että Markovin ketju unohtaa kaikki muut tilat paitsi viimeisen kaavan (2) mukaisesti, kutsutaan *Markovin ominaisuudeksi*.

Markovin ketjut ovat usein tärkeitä sovellusten kannalta, koska ne yksinkertaistavat laskentaa, jossa tarvitaan kertolaskukaavaa (1). Johdetaan seuraavaksi eräs tärkeä tulos tilojen laskemiseksi.

Lause 1.6. Olkoon \mathbf{X} Markovin ketju. Markovin ketjun tiheysfunktioille pätee

$$p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (3)$$

missä $k \in \mathbb{N}_+$.

Todistus. Tilojen \mathbf{x}_k ja \mathbf{x}_{k-1} yhteisjakauma, kun $\mathbf{y}_{1:k-1}$ on annettu, on

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_{1:k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}) \\ &= p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{y}_{1:k-1}), \end{aligned}$$

missä on käytetty Markovin ominaisuutta mittauksen $\mathbf{y}_{1:k-1}$ poistamiseen. Marginaalijakaumaan (3) päästään tästä, kun integroidaan viimeinen yhtälö muuttujan \mathbf{x}_{k-1} suhteen.

□

Yhtälöä (3) kutsutaan *Chapman-Kolmogorovin yhtälöksi*.

Lisää tietoa Markovin ketjuista voi saada lähteestä [5].

Määritelmä 1.7. Satunnaisvektori $\mathbf{x} \in \mathbb{R}^n$ noudattaa *multinormaalijakumaa* odotusarvolla $\mathbf{m} \in \mathbb{R}^n$ ja kovarianssilla $\mathbf{P} \in \mathbb{R}^{n \times n}$, jos sen tiheysfunktio on muotoa

$$\mathcal{N}(\mathbf{x} | \mathbf{m}, \mathbf{P}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{P}|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \right). \quad (4)$$

Tässä $|\mathbf{P}|$ on kovarianssimatriisin \mathbf{P} determinantti. Satunnaisvektori \mathbf{x} osittaan usein kahteen osaan:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix},$$

jolloin myös odotusarvo \mathbf{m} ja kovarianssimatriisi \mathbf{P} jakaumassa (4) jaetaan osiin:

$$\mathbf{m} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{pmatrix}, \mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix}.$$

Teoriasta tiedetään, että kovarianssimatriisiin \mathbf{P} täytyy olla symmetrinen ja positiivisesti definiitti. Kovarianssimatriisiin komponenteille pätee siis $\mathbf{P}_{12} = \mathbf{P}_{21}^T$. Satunnaisvektorin osituksen tarkoituksesta on Kalman-suotimiin liittyen ilmaista dynaamisen systeemin tilat ja mittaukset saman jakauman avulla.

Kalman-suotimien johtamiseksi tarvitaan johtaa muutama multinormaalijakaumaan liittyvä aputulos. Käydään nämä läpi seuraavaksi.

Lemma 1.8. *Jos satunnaisvektorit \mathbf{x} ja \mathbf{y} noudattavat yhteistodennäköisyysjakaumana multinormaalijakaumaa*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \begin{pmatrix} \mathbf{A} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{B} \end{pmatrix}\right),$$

niin reunajakaumat noudattavat multinormaalijakaumaa seuraavasti:

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\mathbf{a}, \mathbf{A}), \\ \mathbf{y} &\sim \mathcal{N}(\mathbf{b}, \mathbf{B}). \end{aligned}$$

Ehdolliset jakaumat satunnaisvektoreille \mathbf{x} ja \mathbf{y} noudattavat multinormaalijakaumaa seuraavasti:

$$\begin{aligned} \mathbf{x} \mid \mathbf{y} &\sim \mathcal{N}(\mathbf{a} + \mathbf{CB}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{CB}^{-1}\mathbf{C}^T), \\ \mathbf{y} \mid \mathbf{x} &\sim \mathcal{N}(\mathbf{b} + \mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x} - \mathbf{a}), \mathbf{B} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}). \end{aligned}$$

Lemma 1.9. *Olkoon $\mathbf{x} \in \mathbb{R}^n$ ja $\mathbf{y} \in \mathbb{R}^m$ satunnaisvektoreita ja*

$$\begin{aligned} \mathbf{x} &\sim \mathcal{N}(\mathbf{m}, \mathbf{P}), \\ \mathbf{y} \mid \mathbf{x} &\sim \mathcal{N}(\mathbf{Hx} + \mathbf{u}, \mathbf{R}). \end{aligned}$$

Tällöin satunnaisvektorien \mathbf{x} ja \mathbf{y} yhteistihesfunktio on

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{Hm} + \mathbf{u} \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{PH}^T \\ \mathbf{HP} & \mathbf{HPH}^T + \mathbf{R} \end{pmatrix}\right),$$

ja satunnaisvektorin \mathbf{y} reunajakauma on

$$\mathbf{y} \sim \mathcal{N}(\mathbf{Hm} + \mathbf{u}, \mathbf{PH}^T + \mathbf{R}).$$

Määritelmä 1.10. Olkoon $\mathbf{x} \in \mathbb{R}^n$ ja $\mathbf{y} \in \mathbb{R}^m$, missä $n, m \in \mathbb{N}_+$, satunnaisvektoreita. Oletetaan, että yhtälö

$$p(\mathbf{x} | \mathbf{y}) = \frac{p(\mathbf{y} | \mathbf{x})p(\mathbf{x})}{p(\mathbf{y})} \quad (5)$$

on hyvin määritelty. Yhtälöä (5) kutsutaan *Bayesin kaavaksi* tiheysfunktioille. Ehdollista tiheysfunktiota $p(\mathbf{x} | \mathbf{y})$ kutsutaan *posterioritodennäköisyystiheysfunktioksi*, ehdollista tiheysfunktiota $p(\mathbf{y} | \mathbf{x})$ *uskottavuusfunktioksi* ja tiheysfunktiota $p(\mathbf{y})$ *prioritodennäköisyystiheysfunktioksi*. Yhtälön (5) oikean puolen nimitäjässä oleva lauseke

$$p(\mathbf{y}) = \int p(\mathbf{y} | \mathbf{x})p(\mathbf{x})d\mathbf{x}$$

on nollasta eroava normitusvakio.

Bayesin kaavaa (5) käytetään Kalman-suotimien johtamisessa rekursiivisesti siten, että päivittämällä datavektori \mathbf{y} käytetään uutta prioritodenäkösyystiheysfunktiota ja uskottavuusfunktiota posterioritodennäköisyystiheysfunktion laskemiseksi.

Määritelmä 1.11. *Gaussisessa approksimaatiossa* posteriorijakaumaa arvioidaan multinormaalijakaumalla. Merkitään tästä

$$p(\mathbf{x} | \mathbf{y}) \simeq \mathcal{N}(\mathbf{x} | \mathbf{m}, \mathbf{P}), \quad (6)$$

missä $\mathbf{x} \in \mathbb{R}^n$ ja $\mathbf{y} \in \mathbb{R}^m$; $n, m \in \mathbb{N}_+$, ja $\mathbf{m} \in \mathbb{R}^n$ on satunnaisvektorin \mathbf{x} odotusarvo ja \mathbf{P} kovarianssi.

2 Lineaarinен Kalman-suodin

Tässä kappaleessa johdetaan lineaarinen Kalman-suodin, joka on rekursiivinen ratkaisu tietyn tyypilliselle satunnaiselle dynaamiselle systeemille.

2.1 Tila-avaruuusmallit

Kalman-suodinten tapauksessa käytetään niin sanottuja *Markovin pilomalleja*. Tällöin estimoitavia systeemin tiloja $\{\mathbf{x}_k, k \in T\}$ ei voida suoraan mitata, vaan niitä on tarkasteltava epäsuorien mittausten $\{\mathbf{y}_k, k \in T\}$ avulla.

Bayesilaisessa suodatuksessa tarkoituksesta on saada laskettua jakauma $p(\mathbf{x}_k | \mathbf{y}_{1:k})$, eli nykyisen tilan \mathbf{x}_k jakauma, kun aiempien ajanhetkien mittausten arvot tunnetaan. Tämän jakauman laskeminen Bayesin kaavalla ei ole usein käytännössä mahdollista laskennallisen monimutkaisuuden kasvaessa mittausten määränpäähän vuoksi.

Rajoittamalla laskenta reunajakaumiin ja olettamalla dynaamisten malleiden luokan olevan Markovin jonoja laskenta muuttuu kevyemmäksi. Tältä pohjalta voidaan kehittää seuraava *satunnainen tila-avaruuusmalli*:

$$\begin{aligned} \mathbf{x}_0 &\sim p(\mathbf{x}_0), \\ \mathbf{x}_k &\sim p(\mathbf{x}_k | \mathbf{x}_{k-1}), \\ \mathbf{y}_k &\sim p(\mathbf{y}_k | \mathbf{x}_k), \end{aligned} \tag{7}$$

missä jakaumaa $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ kutsutaan *evoluutiomalliksi* ja jakaumaa $p(\mathbf{y}_k | \mathbf{x}_k)$ *havaintomalliksi*. Näitä jakaumia voidaan käyttää apuna Bayesilaisen suodatuksen jakauman laskemiseen.

Merkitään jatkossa parametrivektoria $\boldsymbol{\theta}$:lla. Parametrien estimointi tapahtuu Bayesilaisessa suodatuksessa siten, että asetetaan parametreille priorijakauma $p(\boldsymbol{\theta})$ ja pidetään parametreja ylimääräisesti satunnaismuuttujina tilamallissa (7). Näin saadaan parametrien estimoihin seuraava satunnainen tila-avaruuusmalli:

$$\begin{aligned} \boldsymbol{\theta} &\sim p(\boldsymbol{\theta}), \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0 | \boldsymbol{\theta}), \\ \mathbf{x}_k &\sim p(\mathbf{x}_k | \mathbf{x}_{k-1}, \boldsymbol{\theta}), \\ \mathbf{y}_k &\sim p(\mathbf{y}_k | \mathbf{x}_k, \boldsymbol{\theta}). \end{aligned}$$

Tila-avaruuusmallia (7) voidaan käyttää jakauman $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ laskemiseen. Seuraava algoritmi antaa keinon täähän.

Algoritmi 2.1.1. Tiheysfunktiot $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ ja $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ saadaan laskettua seuraavien vaiheiden avulla:

1. Aloitetaan rekursio alustamalla priorijakauma $p(\mathbf{x}_0)$.
2. Chapman-Kolmogorovin yhtälöistä (3) saadaan

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}.$$

3. Lasketaan

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) = \frac{1}{Z_k} p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}),$$

missä

$$Z_k = \int p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) d\mathbf{x}_k$$

on normitusvakio.

2.2 Kalman-suotimen johto

Määritelmä 2.2.1. Lineaarinen Kalman-suodin on suljettu ratkaisu Bayesilaisiin suodatusyhtälöihin, kun evoluutiomalli ja havaintomalli ovat lineaarisia, eli muotoa

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1} \mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{r}_k, \end{aligned} \tag{8}$$

missä $\mathbf{x}_k \in \mathbb{R}^n$ on tila, $\mathbf{y}_k \in \mathbb{R}^m$ on mittaus, $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ on prosesin kohina, $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ on mittausvirhe ja priorijakauma on gaussinen $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{n}_0, \mathbf{P}_0)$. Matriisi \mathbf{A}_{k-1} on evoluutiomallin siirtomatriisi ja \mathbf{H}_k mittausmallin matriisi.

Käyttämällä Gaussista approksimaatiota evoluutiomalliin ja havaintomalliin, voidaan tila-avaruusmalli (8) kirjoittaa jakaumamallina

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{A}_{k-1} \mathbf{x}_{k-1}, \mathbf{Q}_{k-1}), \\ p(\mathbf{y}_k \mid \mathbf{x}_k) &= \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k). \end{aligned}$$

Esitetään seuraavaksi lineaarinen Kalman-suodin.

Lause 2.2.2. Ratkaisuna yhtälöihin (8) saadaan Bayesilaisen suodatuksen posterioritodennäköisyydet

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-), \\ p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \\ p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k \mathbf{m}_k^-, \mathbf{S}_k), \end{aligned}$$

missä

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}_{k-1} \mathbf{m}_{k-1}, \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}, \end{aligned} \tag{9}$$

ja

$$\begin{aligned} \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} [\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_k^-], \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^-. \end{aligned} \tag{10}$$

Todistus. Ideana todistuksessa on käyttää algoritmia 2.1.1. Johdetaan ensin parametrien (9) yhtälöt. Lemmasta 1.9 saadaan

$$\begin{aligned} p(\mathbf{x}_{k-1}, \mathbf{x}_k \mid \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{A}_{k-1} \mathbf{x}_{k-1}, \mathbf{Q}_{k-1}) \mathcal{N}(\mathbf{x}_{k-1} \mid \mathbf{m}_{k-1}, \mathbf{P}_{k-1}) \\ &= \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{pmatrix} \mid \mathbf{m}', \mathbf{P}'\right). \end{aligned}$$

Tässä

$$\begin{aligned} \mathbf{m}' &= \begin{pmatrix} \mathbf{m}_{k-1} \\ \mathbf{A}_{k-1} \mathbf{m}_{k-1} \end{pmatrix}, \\ \mathbf{P}' &= \begin{pmatrix} \mathbf{P}_{k-1} & \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T \\ \mathbf{A}_{k-1} \mathbf{P}_{k-1} & \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{pmatrix}. \end{aligned}$$

Vektorin \mathbf{x}_k reunajakauma on Lemman 1.8 mukaan

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-).$$

Tässä

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}_{k-1} \mathbf{m}_{k-1}, \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}. \end{aligned}$$

Johdetaan seuraavaksi parametrien (10) yhtälöt. Lemmasta 1.9 saadaan vektorien \mathbf{x}_k ja \mathbf{y}_k yhteistodennäköisyys:

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{y}_k \mid \mathbf{y}_{1:k-1}) &= p(\mathbf{y}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \\ &= \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k \mathbf{x}_k, \mathbf{R}_k) \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-) \\ &= \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{pmatrix} \mid \mathbf{m}'', \mathbf{P}''\right). \end{aligned}$$

Tässä

$$\mathbf{m}'' = \begin{pmatrix} \mathbf{m}_k^- \\ \mathbf{H}_k \mathbf{m}_k^- \end{pmatrix}, \mathbf{P}'' = \begin{pmatrix} \mathbf{P}_k^- & \mathbf{P}_k^- \mathbf{H}_k^T \\ \mathbf{H}_k \mathbf{P}_k^- & \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \end{pmatrix}.$$

Lemmasta 1.8 saadaan

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{y}_k, \mathbf{y}_{1:k-1}) &= p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \\ &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \end{aligned}$$

missä

$$\begin{aligned} \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} [\mathbf{y}_k - \mathbf{H}_k \mathbf{m}_k^-], \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^-. \end{aligned}$$

□

Algoritmi 2.2.3. Lineaarinen Kalman-suodin jaetaan algoritmiltaan usein kahteen vaiheeseen.

1. Ennustevaiheessa lasketaan ennusteparametrit

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{A}_{k-1} \mathbf{m}_{k-1}, \\ \mathbf{P}_k^- &= \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}. \end{aligned}$$

2. Päivitysvaiheessa lasketaan matriisit

$$\begin{aligned} \mathbf{v}_k &= \mathbf{y}_k - \mathbf{H}_k \mathbf{m}_k^-, \\ \mathbf{S}_k &= \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k, \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k \mathbf{v}_k, \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T, \end{aligned}$$

jotka saadaan hajottamalla matriisit (10) sopivasti.

3. Aloitetaan seuraava kierros.

Lineaarisen Kalman-suotimen idea on siis laskea ensin ennustevaiheessa tilan estimaatti. Tämän jälkeen tästä estimaattia verrataan mittauksiin ja korjataan sopivasti päivitysvaiheessa. Korjattua estimaattia käytetään tämän jälkeen uutena ennusteena. Kalman-suodin toimii näin ollen rekursiivisena estimaattorina. Matriisia \mathbf{K}_k kutsutaan *vahvistusmatriisiksi*, ja se kertoo, miten paljon odotusarvoa ja kovarianssia pitää päivitysvaiheessa skaalata.

Epälineaarisille suodatusyhtälöille lineaarisella Kalman-suotimella ei saa yleensä tarkkoja ratkaisuja lineaarisuudesta johtuvan epätarkkuuden vuoksi. Tätä tarkoitusta varten on kehitetty erilaisia Kalman-suotimen laajennuksia.

3 Kalman-suotimen laajennuksia

Tässä kappaleessa käydään läpi laajennettu Kalman-suodin, hajustamaton Kalman-suodin ja hiukkassuodin.

3.1 Laajennettu Kalman-suodin

Johdetaan ensin laajennettu Kalman-suodin. Tämä menetelmä saadaan arvioimalla epälineaaristen suodatusyhtälöiden funktioita Taylorin kehitelmällä, jolloin voidaan käyttää lineaarista Kalman-suodinta tavalliseen tapaan. Tarkastellaan satunnaisvektorin \mathbf{x} muunnosta

$$\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{P}),$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}),$$

missä $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ ja $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ on jokin epälineaarinen funktio. Tätä muunnosta tarvitaan laajennetun Kalman-suotimen johtamisessa. Johdetaan seuraavaksi pari tulosta.

Määritelmä 3.1.1. Olkoon $\mathbf{x} = \mathbf{m} + \delta\mathbf{x}$, missä $\delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$. Oletetaan, että funktion $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ n :nnen asteen osittaisderivaatat ovat olemassa.

Tällöin

$$\mathbf{g}(\mathbf{m} + \delta\mathbf{x}) \approx \mathbf{g}(\mathbf{m}) + \mathbf{G}_{\mathbf{x}}(\mathbf{m})\delta\mathbf{x} + \sum_i \frac{1}{2} \delta\mathbf{x}^T \mathbf{G}_{xx}^{(i)}(\mathbf{m}) \delta\mathbf{x} \mathbf{e}_i + \dots, \quad (11)$$

missä

$$[\mathbf{G}_{\mathbf{x}}(\mathbf{m})]_{j,j'} = \left. \frac{\partial g_j(\mathbf{x})}{\partial x_{j'}} \right|_{\mathbf{x}=\mathbf{m}},$$

$$[\mathbf{G}_{\mathbf{x}\mathbf{x}}^{(i)}(\mathbf{m})]_{j,j'} = \left. \frac{\partial^2 g_i(\mathbf{x})}{\partial x_j \partial x_{j'}} \right|_{\mathbf{x}=\mathbf{m}}$$

ja $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ on vektori, jonka i :s komponentti on 1. Yhtälön (11) oikean puolen sarjaa kutsutaan funktion $\mathbf{g}(\mathbf{x})$ *Taylorin kehitelmäksi*.

Lemma 3.1.2. *Olkoon $\mathbf{x} = \mathbf{m} + \delta\mathbf{x}$, missä $\delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$. Olkoon funktio $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ensimmäisen kertaluvun osittaisderivaatat olemassa. Funktion \mathbf{g} ensimmäisen kertaluvun Taylorin kehitelmän*

$$\mathbf{g}(\mathbf{x}) \simeq \mathbf{g}(\mathbf{m}) + \mathbf{G}_{\mathbf{x}}(\mathbf{m})\delta\mathbf{x},$$

missä merkintä \simeq tarkoittaa, että funktio $\mathbf{g}(\mathbf{x})$ oletetaan olevan likimääräisesti oikean puolen lauseke, odotusarvo on

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] \simeq \mathbf{g}(\mathbf{m}).$$

Toisaalta kovarianssille pätee arvio

$$\text{Cov}[\mathbf{g}(\mathbf{x})] \simeq \mathbf{G}_{\mathbf{x}}(\mathbf{m}) \mathbf{P} \mathbf{G}_{\mathbf{x}}^T(\mathbf{m}).$$

Todistus. Odotusarvo on nyt

$$\begin{aligned} \mathbb{E}[\mathbf{g}(\mathbf{x})] &\simeq \mathbb{E}[\mathbf{g}(\mathbf{m}) + \mathbf{G}_{\mathbf{x}}(\mathbf{m})\delta\mathbf{x}] \\ &= \mathbf{g}(\mathbf{m}) + \mathbf{G}_{\mathbf{x}}(\mathbf{m}) \mathbb{E}[\delta\mathbf{x}] \\ &= \mathbf{g}(\mathbf{m}). \end{aligned}$$

Funktion $\mathbf{g}(\mathbf{x})$ kovarianssiksi saadaan

$$\begin{aligned}
\text{Cov}[\mathbf{g}(\mathbf{x})] &= \mathbb{E}[(\mathbf{g}(\mathbf{x}) - \mathbb{E}[\mathbf{g}(\mathbf{x})])(\mathbf{g}(\mathbf{x}) - \mathbb{E}[\mathbf{g}(\mathbf{x})])^T] \\
&\simeq \mathbb{E}[(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{m}))(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{m}))^T] \\
&\simeq \mathbb{E}[(\mathbf{g}(\mathbf{m}) + \mathbf{G}_x(\mathbf{m})\delta\mathbf{x} - \mathbf{g}(\mathbf{m}))(\mathbf{g}(\mathbf{m}) + \mathbf{G}_x(\mathbf{m})\delta\mathbf{x} - \mathbf{g}(\mathbf{m}))^T] \\
&= \mathbb{E}[(\mathbf{G}_x(\mathbf{m})\delta\mathbf{x})(\mathbf{G}_x(\mathbf{m})\delta\mathbf{x})^T] \\
&= \mathbf{G}_x(\mathbf{m}) \mathbb{E}[\delta\mathbf{x}\delta\mathbf{x}^T] \mathbf{G}_x^T(\mathbf{m}) \\
&= \mathbf{G}_x(\mathbf{m}) \mathbf{P} \mathbf{G}_x^T(\mathbf{m}).
\end{aligned}$$

□

Laajennetun Kalman-suotimen johdossa tarvitaan muunnosta

$$\begin{aligned}
\mathbf{x} &\sim \mathcal{N}(\mathbf{m}, \mathbf{P}), \\
\mathbf{q} &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \\
\mathbf{y} &= \mathbf{g}(\mathbf{x}) + \mathbf{q},
\end{aligned} \tag{12}$$

missä \mathbf{q} on riippumaton muuttujasta \mathbf{x} . Satunnaismuuttujien \mathbf{x} ja \mathbf{y} yhteisjakauma saadaan seuraavan lemmän avulla.

Lemma 3.1.3. *Muunnoksessa (12) muuttujien \mathbf{x} ja \mathbf{y} yhteistodennäköisyysjakauma on*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \boldsymbol{\mu}_L \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{C}_L \\ \mathbf{C}_L^T & \mathbf{S}_L \end{pmatrix} \right),$$

missä

$$\begin{aligned}
\boldsymbol{\mu}_L &= \mathbf{g}(\mathbf{m}), \\
\mathbf{S}_L &= \mathbf{G}_x(\mathbf{m}) \mathbf{P} \mathbf{G}_x^T(\mathbf{m}) + \mathbf{Q}, \\
\mathbf{C}_L &= \mathbf{P} \mathbf{G}_x^T(\mathbf{m}).
\end{aligned}$$

Toisen asteen laajennetun Kalman-suotimen johtamiseen tarvitaan seuraavaa lemmaa.

Lemma 3.1.4. *Toisen kertaluvun Taylorin approksimaatio muunnokselle (12) on*

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \boldsymbol{\mu}_Q \end{pmatrix}, \begin{pmatrix} \mathbf{P} & \mathbf{C}_Q \\ \mathbf{C}_Q^T & \mathbf{S}_Q \end{pmatrix} \right),$$

missä

$$\begin{aligned}\boldsymbol{\mu}_Q &= \mathbf{g}(\mathbf{m}) + \frac{1}{2} \sum_i \mathbf{e}_i \operatorname{tr} (\mathbf{G}_{\mathbf{xx}}^{(i)}(\mathbf{m}) \mathbf{P}), \\ \mathbf{S}_Q &= \mathbf{G}_{\mathbf{x}}(\mathbf{m}) \mathbf{P} \mathbf{G}_{\mathbf{x}}^T(\mathbf{m}) + \frac{1}{2} \sum_{i,i'} \mathbf{e}_i \mathbf{e}_{i'}^T \operatorname{tr} (\mathbf{G}_{\mathbf{xx}}^{(i)}(\mathbf{m}) \mathbf{P} \mathbf{G}_{\mathbf{xx}}^{(i')}(\mathbf{m}) \mathbf{P}), \\ \mathbf{C}_Q &= \mathbf{P} \mathbf{G}_{\mathbf{x}}^T(\mathbf{m}).\end{aligned}$$

Tässä $\mathbf{G}_{\mathbf{x}}(\mathbf{m})$ on funktio \mathbf{g} Jacobin matriisi pisteessä \mathbf{m} , $\mathbf{G}_{\mathbf{xx}}^{(i)}(\mathbf{m})$ on funktio g_i Hessian matriisi pisteessä \mathbf{m} , \mathbf{e}_i on yksikkävektori, jonka i :s komponentti on 1, ja operaatio $\operatorname{tr}()$ on matriisin jälki.

Määritelmä 3.1.5. Oletetaan, että prosessin ja mittauksen kohinat ovat aditiivisia. Laajennettu Kalman-suodatin on ratkaisu Bayesilaisiin suodatusyhdistelöihin, kun tila-avaruuusmalli on muotoa

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k,\end{aligned}\tag{13}$$

missä $\mathbf{x}_k \in \mathbb{R}^n$ on tila, $\mathbf{y}_k \in \mathbb{R}^m$ on mittaus, $\mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ on e luutiomallin virhe, $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ on havaintomallin virhe ja \mathbf{f} ja \mathbf{h} ovat epälineaarisia funktioita.

Kuten lineaarisen Kalman-suotimenkin tapauksessa, tarkoituksesta on laskea Bayesilaisen suodatuksen jakaumat. Esitetään seuraavaksi laajennetun Kalman-suotimen ratkaisu.

Lause 3.1.6. Suodatusjakaumat ovat laajennetun Kalman-suotimen tapauksessa muotoa

$$\begin{aligned}p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k^-, \mathbf{P}_k^-), \\ p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) &= \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \\ p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{y}_k \mid \mathbf{H}_k \mathbf{m}_k^-, \mathbf{S}_k).\end{aligned}\tag{14}$$

Tarvittavat parametrit yhtälöihin (14) ovat

$$\begin{aligned}\mathbf{m}_k^- &= \mathbf{f}(\mathbf{m}_{k-1}), \\ \mathbf{P}_k^- &= \mathbf{F}_{\mathbf{x}}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{x}}^T(\mathbf{m}_{k-1}) + \mathbf{Q}_{k-1},\end{aligned}\tag{15}$$

ja

$$\begin{aligned}
\mathbf{v}_k &= \mathbf{y}_k - \mathbf{h}(\mathbf{m}_k^-), \\
\mathbf{S}_k &= \mathbf{H}_{\mathbf{x}}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}_{\mathbf{x}}^T(\mathbf{m}_k^-) + \mathbf{R}_k, \\
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_{\mathbf{x}}^T(\mathbf{m}_k^-) \mathbf{S}_k^{-1}, \\
\mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k \mathbf{v}_k, \\
\mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T,
\end{aligned} \tag{16}$$

missä $\mathbf{F}_{\mathbf{x}}(\mathbf{m}_{k-1})$ on funktio \mathbf{f} Jacobin matriisi pisteessä \mathbf{m}_{k-1} ja $\mathbf{H}_{\mathbf{x}}(\mathbf{m}_k^-)$ on funktio \mathbf{h} Jacobin matriisi pisteessä \mathbf{m}_k^- .

Todistus. Sovelletaan lemmaa 3.1.3 funktioon

$$\mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}.$$

Tästä saadaan Gaussinen approksimaatio

$$f(\mathbf{x}_{k-1}, \mathbf{x}_k \mid \mathbf{y}_{1:k-1}) \simeq \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{pmatrix} \mid \mathbf{m}', \mathbf{P}'\right),$$

missä

$$\begin{aligned}
\mathbf{m}' &= \begin{pmatrix} \mathbf{x}_{k-1} \\ \mathbf{x}_k \end{pmatrix}, \\
\mathbf{P}' &= \begin{pmatrix} \mathbf{P}_{k-1} & \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{x}}^T \\ \mathbf{F}_{\mathbf{x}} \mathbf{P}_{k-1} & \mathbf{F}_{\mathbf{x}} \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{x}}^T + \mathbf{Q}_{k-1} \end{pmatrix}.
\end{aligned}$$

Jacobian matriisi $\mathbf{F}_{\mathbf{x}}$ lasketaan pisteessä $\mathbf{x} = \mathbf{m}_{k-1}$. Nyt

$$\begin{aligned}
\mathbf{m}_k^- &= \mathbf{f}(\mathbf{m}_{k-1}), \\
\mathbf{P}_k^- &= \mathbf{F}_{\mathbf{x}} \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{x}}^T + \mathbf{Q}_{k-1}.
\end{aligned}$$

Sovelletaan nyt lemmaa 3.1.3 funktioon

$$\mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k.$$

Tästä saadaan Gaussinen approksimaatio

$$f(\mathbf{x}_k, \mathbf{y}_k \mid \mathbf{y}_{1:k-1}) \simeq \mathcal{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{pmatrix} \mid \mathbf{m}'', \mathbf{P}''\right),$$

missä

$$\begin{aligned} \mathbf{m}'' &= \begin{pmatrix} \mathbf{m}_k^- \\ \mathbf{h}(\mathbf{m}_k^-) \end{pmatrix}, \\ \mathbf{P}'' &= \begin{pmatrix} \mathbf{P}_k^- & \mathbf{P}_k^- \mathbf{H}_x^T \\ \mathbf{H}_x \mathbf{P}_k^- & \mathbf{H}_x \mathbf{P}_k^- \mathbf{H}_x^T + \mathbf{R}_k \end{pmatrix}. \end{aligned}$$

Jacobin matriisi \mathbf{H}_x lasketaan pisteessä $\mathbf{x} = \mathbf{m}_k^-$

Lemman (1.8) perusteella saadaan

$$f(\mathbf{x}_k \mid \mathbf{y}_k, \mathbf{y}_{1:k-1}) \simeq \mathcal{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k),$$

missä

$$\begin{aligned} \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{P}_k^- \mathbf{H}_x^T (\mathbf{H}_x \mathbf{P}_k^- \mathbf{H}_x^T + \mathbf{R}_k)^{-1} [\mathbf{y}_k - \mathbf{h}(\mathbf{m}_k^-) \mathbf{m}_k^-], \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_x^T (\mathbf{H}_x \mathbf{P}_k^- \mathbf{H}_x^T + \mathbf{R}_k)^{-1} \mathbf{H}_x \mathbf{P}_k^-. \end{aligned} \quad (17)$$

□

Yhtälöiden (17) laskenta voidaan hajoittaa vastaavasti kuin lineaarisen Kalman-suotimen tapauksessa. Laajennetun Kalman-suotimen algoritmi voidaan siis kirjoittaa vastaavasti kuin lineaarinen versio 2.2.3.

Sovellettaessa toisen asteen Taylorin approksimaatiota epälineaariseen tila-avaruuusmalliin (13) lemmän 3.1.4 mukaisesti, saadaan muodostettua niin kutsuttu toisen asteen laajennettu Kalman-suodin. Esitetään kyseinen algoritmi seuraavaksi.

Algoritmi 3.1.7. Toisen asteen laajennettu Kalman-suodin saadaan ratkaistua seuraavien ennustus-ja päivitysvaiheiden avulla.

1. Ennustetaan odotusarvo ja kovarianssi:

$$\begin{aligned} \mathbf{m}_k^- &= \mathbf{f}(\mathbf{m}_{k-1}) + \frac{1}{2} \sum_i \mathbf{e}_i \operatorname{tr}(\mathbf{F}_{\mathbf{xx}}^{(i)}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1}), \\ \mathbf{P}_k^- &= \mathbf{F}_{\mathbf{x}}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{x}}^T(\mathbf{m}_{k-1}) \\ &\quad + \frac{1}{2} \sum_{i,i'} \mathbf{e}_i \mathbf{e}_{i'}^T \operatorname{tr}(\mathbf{F}_{\mathbf{xx}}^{(i)}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1} \mathbf{F}_{\mathbf{xx}}^{(i')}(\mathbf{m}_{k-1}) \mathbf{P}_{k-1}) + \mathbf{Q}_{k-1}. \end{aligned}$$

2. Päivitetään odotusarvo ja kovarianssi mittausten avulla:

$$\mathbf{v}_k = \mathbf{y}_k - \mathbf{h}(\mathbf{m}_k^-) - \frac{1}{2} \sum_i \mathbf{e}_i \operatorname{tr}(\mathbf{H}_{\mathbf{xx}}^{(i)}(\mathbf{m}_k^-) \mathbf{P}_k^-),$$

$$\begin{aligned} \mathbf{S}_k &= \mathbf{H}_{\mathbf{x}}(\mathbf{m}_k^-) \mathbf{P}_k^- \mathbf{H}_{\mathbf{x}}^T(\mathbf{m}_k^-) \\ &\quad + \frac{1}{2} \sum_{i,i'} \mathbf{e}_i \mathbf{e}_{i'}^T \operatorname{tr}(\mathbf{H}_{\mathbf{xx}}^{(i)}(\mathbf{m}_k^-) \mathbf{P}_{k-1} \mathbf{H}_{\mathbf{xx}}^{(i')}(\mathbf{m}_k^-) \mathbf{P}_{k-1}) + \mathbf{R}_k, \end{aligned}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_{\mathbf{x}}^T(\mathbf{m}_k^-) \mathbf{S}_k^{-1},$$

$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k \mathbf{v}_k,$$

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T.$$

3. Aloitetaan seuraava kierros.

Tässä $\mathbf{F}_{\mathbf{xx}}^{(i)}$ on funktion \mathbf{f} Hessian matriisi ja $\mathbf{H}_{\mathbf{xx}}^{(i)}$ on funktion \mathbf{h} Hessian matriisi.

3.2 Hajustamaton Kalman-suodin

Hajustamaton Kalman-suodin on eräs menetelmä epälineaarisen suodatusyhälön (13) ratkaisemiseksi. Ideana on arvioida posteriorijakauman odotusarvoa ja kovarianssia kiinteiden pisteiden avulla, jotka ovat sijoitettu kohdejakaumaan nähden sopivasti. Pisteiden sijoittelu voidaan muodostaa niin kutsuttulla hajustamattomalla muunnoksella. Määritellään tämä seuraavaksi.

Määritelmä 3.2.1. *Hajustamaton muunnos* muodostaa Gaussisen approksimaation (6) seuraavien vaiheiden kautta. Ensinnäkin muodostetaan $2n+1$ kapipaletta *sigmapisteitä*

$$\begin{aligned} \mathbf{X}^{(0)} &= \mathbf{m}, \\ \mathbf{X}^{(i)} &= \mathbf{m} + \sqrt{n+\lambda} \left[\sqrt{\mathbf{P}} \right]_i, \\ \mathbf{X}^{(i+n)} &= \mathbf{m} - \sqrt{n+\lambda} \left[\sqrt{\mathbf{P}} \right]_i, \quad i = 1, \dots, n, \end{aligned}$$

missä merkintä $[\cdot]_i$ tarkoittaa matriisin saraketta i , \mathbf{m} on odotusarvon esimaatti, n on odotusarvovektorin dimensio, matriisin P neliöjuuri määritelään

$$\sqrt{\mathbf{P}} \sqrt{\mathbf{P}} = \mathbf{P},$$

ja λ on skaalausparametri

$$\lambda = \alpha^2(n + \kappa) - n.$$

Tässä parametrit α ja κ määrittävät sigmapisteiden hajonnan keskiarvon ympärillä.

Seuraavaksi monistetaan sigmapisteet epälineaarisen funktion avulla:

$$\mathbf{Y}^{(i)} = \mathbf{g}(\mathbf{X}^{(i)}), \quad i = 0, \dots, 2n.$$

Muunnetun muuttujan odotusarvo ja kovarianssi lasketaan seuraavasti:

$$\begin{aligned} \mathbb{E}[\mathbf{g}(\mathbf{x})] &\simeq \boldsymbol{\mu}_U = \sum_{i=0}^{2n} W_i^{(m)} \mathbf{Y}^{(i)} \\ \text{Cov}[\mathbf{g}(\mathbf{x})] &\simeq \mathbf{S}_U = \sum_{i=0}^{2n} W_i^{(c)} (\mathbf{Y}^{(i)} - \boldsymbol{\mu}_U)(\mathbf{Y}^{(i)} - \boldsymbol{\mu}_U)^T, \end{aligned}$$

missä painot $W_i^{(m)}$ ja $W_i^{(c)}$ saadaan seuraavasti:

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{n + \lambda}, \\ W_0^{(c)} &= \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta), \\ W_i^{(m)} &= \frac{1}{2(n + \lambda)}, \quad i = 1, \dots, 2n, \\ W_i^{(c)} &= \frac{1}{2(n + \lambda)}, \quad i = 1, \dots, 2n. \end{aligned} \tag{18}$$

Algoritmi 3.2.2. Bayesilaiset suodatusyhtälöt voidaan nyt muodostaa käytämällä hajustamatonta muunnosta. Tuloksena saadaan seuraava hajustamattoman Kalman-suotimen algoritmi.

1. Lasketaan ennuste:

$$\begin{aligned} \mathbf{X}_{k-1}^{(0)} &= \mathbf{m}_{k-1}, \\ \mathbf{X}_{k-1}^{(i)} &= \mathbf{m}_{k-1} + \sqrt{n + \lambda} \left[\sqrt{\mathbf{P}_{k-1}} \right]_i, \\ \mathbf{X}_{k-1}^{(i+n)} &= \mathbf{m}_{k-1} - \sqrt{n + \lambda} \left[\sqrt{\mathbf{P}_{k-1}} \right]_i, \quad i = 1, \dots, n, \end{aligned}$$

2. Monistetaan sigmapisteet evoluutiomallilla:

$$\hat{\mathbf{X}}_k^{(i)} = \mathbf{f}(\mathbf{X}_{k-1}^{(i)}), \quad i = 0, \dots, 2n.$$

3. Lasketaan ennustettu keskiarvo ja ennustettu kovarianssi:

$$\begin{aligned}\mathbf{m}_k^- &= \sum_{i=0}^{2n} W_i^{(m)} \hat{\mathbf{X}}_k^{(i)}, \\ \mathbf{P}_k^- &= \sum_{i=0}^{2n} W_i^{(c)} (\hat{\mathbf{X}}_k^{(i)} - \mathbf{m}_k^-) (\hat{\mathbf{X}}_k^{(i)} - \mathbf{m}_k^-)^T + \mathbf{Q}_{k-1},\end{aligned}$$

missä painot $W_i^{(m)}$ ja $W_i^{(c)}$ määriteltiin yhtälöissä (18).

4. Päivitetään estimaatit: Muodostetaan sigmapisteet:

$$\begin{aligned}\mathbf{X}_k^{-(0)} &= \mathbf{m}_k^-, \\ \mathbf{X}_k^{-(i)} &= \mathbf{m}_k^- + \sqrt{n+\lambda} \left[\sqrt{\mathbf{P}_k^-} \right]_i, \\ \mathbf{X}_k^{-(i+n)} &= \mathbf{m}_k^- - \sqrt{n+\lambda} \left[\sqrt{\mathbf{P}_k^-} \right]_i, \quad i = 1, \dots, n,\end{aligned}$$

5. Monistetaan sigmapisteet mittausmallilla:

$$\hat{\mathbf{Y}}_k^{(i)} = \mathbf{h}(\mathbf{X}_k^{-(i)}), \quad i = 0, \dots, 2n.$$

6. Lasketaan odotettu keskiarvo, mittauksen odotettu kovarianssi ja mittauksen tilan ristikovarianssi:

$$\begin{aligned}\boldsymbol{\mu}_k &= \sum_{i=0}^{2n} W_i^{(m)} \hat{\mathbf{Y}}_k^{(i)}, \\ \mathbf{S}_k &= \sum_{i=0}^{2n} W_i^{(c)} (\hat{\mathbf{Y}}_k^{(i)} - \boldsymbol{\mu}_k) (\hat{\mathbf{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^T + \mathbf{R}_k, \\ \mathbf{C}_k &= \sum_{i=0}^{2n} W_i^{(c)} (\hat{\mathbf{X}}_k^{-(i)} - \mathbf{m}_k^-) (\hat{\mathbf{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^T.\end{aligned}$$

7. Lasketaan vahvistusmatriisi, tilan odotusarvo ja kovarianssi ehdollisesti tennä mittaukselle:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k (\mathbf{y}_k - \boldsymbol{\mu}_k), \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T.\end{aligned}$$

8. Aloitetaan seuraava kierros.

Laajennetuista Kalman-suotimista poiketen hajustamattomassa versiossa ei tarvitse laskea epälineaaristen funktioiden derivaattoja. Erona hiukkas-suotimiin hajustamattomassa Kalman-suotimessa odotusarvoa ja kovariansia estimoidaan kiinteillä ennaltamäärätyillä pisteillä satunnaisten otosten sijaan.

Hajustamaton Kalman-suodin on erikoistapaus niin kutsutusta *Gaussian suotimista*. Lisää näistä voi lukea lähteestä [6].

3.3 Hiukkassuodin

Tietyissä tilanteissa Bayesilaisia suodatusyhtälöitä $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$, $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$ ja $p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1})$ ei kannata ratkaista Gaussisen approksimaation avulla. Esimerkiksi jakaumien ollessa monihuippuisia menetetään tarkkuutta jakauman valinnasta johtuen. Tällaisissa tilanteissa *Monte Carlo -approksimaatio* toimii usein paremmin. Tarkoituksena on laskea odotusarvo

$$E[\mathbf{g}(\mathbf{x}) \mid \mathbf{y}_{1:T}] = \int \mathbf{g}(\mathbf{x}) p(\mathbf{x} \mid \mathbf{y}_{1:T}) d\mathbf{x}, \quad (19)$$

missä $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ on mielivaltainen funktio ja $p(\mathbf{x} \mid \mathbf{y}_{1:T})$ on posteriorijakauman tiheysfunktio. Monte Carlo -menetelmän ideana on ottaa N kappaletta riippumattomia satunnaisotoksia $\mathbf{x}^{(i)} \sim p(\mathbf{x} \mid \mathbf{y}_{1:T})$, $i = 1, \dots, N$ ja arviodaan odotusarvoa (19) summalla

$$\frac{1}{N} \sum_{i=1}^N \mathbf{g}(\mathbf{x}^{(i)}).$$

Käytännössä ei ole kuitenkaan usein mahdollista ottaa otoksia suoraan posteriorijakaumasta $p(\mathbf{x} \mid \mathbf{y}_{1:T})$. Tämän vuoksi pyritään ottamaan otoksia niin sanotusta *tärkeysjakaumasta* $\pi(\mathbf{x} \mid \mathbf{y}_{1:T})$, jota on helpompi käsitellä. Ehtonaa tärkeysjakauman valinnalle on, että sen tukialue on suurempi tai yhtäsuuri kuin posteriorijakaumalla. Odotusarvoa (19) voidaan arvioida tärkeysjakaumien avulla seuraavasti. Nyt

$$\int \mathbf{g}(\mathbf{x}) p(\mathbf{x} \mid \mathbf{y}_{1:T}) d\mathbf{x} = \int \left[\mathbf{g}(\mathbf{x}) \frac{p(\mathbf{x} \mid \mathbf{y}_{1:T})}{\pi(\mathbf{x} \mid \mathbf{y}_{1:T})} \right] \pi(\mathbf{x} \mid \mathbf{y}_{1:T}) d\mathbf{x}.$$

Ottamalla N otosta tärkeysjakaumasta voidaan muodostaa arvio

$$\begin{aligned} \text{E}[\mathbf{g}(\mathbf{x}) \mid \mathbf{y}_{1:T}] &\approx \frac{1}{N} \sum_{i=1}^N \frac{p(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})}{\pi(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})} \mathbf{g}(\mathbf{x}^{(i)}) \\ &= \sum_{i=1}^N \tilde{w}^{(i)} \mathbf{g}(\mathbf{x}^{(i)}), \end{aligned}$$

missä painot määritellään

$$\tilde{w}^{(i)} = \frac{1}{N} \frac{p(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})}{\pi(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})}. \quad (20)$$

Tätä menettelyä kutsutaan *tärkeysotannaksi*.

Jotta painot (20) voitaisiin laskea, täytyy pystyä laskemaan posterioritodennäköisyystiheysfunktion arvo $p(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})$. Tämä voidaan kirjoittaa Bayesin kaavan avulla muotoon

$$p(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T}) = \frac{p(\mathbf{y}_{1:T} \mid \mathbf{x}^{(i)}) p(\mathbf{x}^{(i)})}{\int p(\mathbf{y}_{1:T} \mid \mathbf{x}) p(\mathbf{x}) d\mathbf{x}}.$$

Nimittäjässä olevaa vakiota ei usein pystytä laskemaan. Muodostamalla tärkeystanta myös tälle vakiolle voidaan odotusarvo (19) saada muotoon

$$\begin{aligned} \text{E}[\mathbf{g}(\mathbf{x}) \mid \mathbf{y}_{1:T}] &= \int \mathbf{g}(\mathbf{x}) p(\mathbf{x} \mid \mathbf{y}_{1:T}) \\ &\approx \sum_{i=1}^N w^{(i)} \mathbf{g}(\mathbf{x}^{(i)}), \end{aligned}$$

missä

$$w^{(i)} = \frac{\frac{p(\mathbf{y}_{1:T} \mid \mathbf{x}^{(i)}) p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)} \mid \mathbf{y}_{1:T})}}{\sum_{j=1}^N \frac{p(\mathbf{y}_{1:T} \mid \mathbf{x}^{(j)}) p(\mathbf{x}^{(j)})}{\pi(\mathbf{x}^{(j)} \mid \mathbf{y}_{1:T})}}.$$

Johto painojen kaavalle löytyy lähteestä [6].

Jaksollisessa tärkeysotannassa muodostetaan tärkeystannan avulla approksimaatiot Bayesilaisille suodatusyhtälöille, kun tila-avaruusmalli voidaan kirjoittaa muodossa

$$\begin{aligned} \mathbf{x}_k &\sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}), \\ \mathbf{y}_k &\sim p(\mathbf{y}_k \mid \mathbf{x}_k). \end{aligned}$$

Suodatusyhtälöt estimoidaan niin kutsuttujen *hiukkasten* $\{(w_k^{(i)}, \mathbf{x}_k^{(i)}) : i = 1, \dots, N\}$ avulla. Ideana on ottaa otoksia $\mathbf{x}_k^{(i)}$ tärkeysjakaumasta ja laskea näille painot $w_k^{(i)}$ ajanhettellä k . Tällä tavoin voidaan laskea odotusarvo

$$\mathbb{E}[\mathbf{g}(\mathbf{x}_k) \mid \mathbf{y}_{1:k}] \approx \sum_{i=1}^N w_k^{(i)} \mathbf{g}(\mathbf{x}_k^{(i)})$$

jokaisella ajanhettellä k .

Jaksollinen tärkeysuudelleenotanta, eli *hiukkassuodin* on muuten samalainen menetelmä hiukkasten laskemiseksi kuin jaksollinen tärkeystointa, mutta tietyillä kierroksilla suoritetaan uudelleenotanta. Tämä pyrkii estämään tilanteen, joissa vain muutamalla hiukkasella on merkittävät painot. Tällöin lisätään suurien painojen omaavien hiukkasten lukumäärää otoksessa pitämällä tiloja $\mathbf{x}_k^{(i)}$ diskreetin satunnaismuuttujan arvoina ja painoja $w_k^{(i)}$ vastaavina todennäköisyyksinä.

Se, millä kierroksella uudelleenotanta suoritetaan, päätetään usein laskemalla niin kutsuttu *hiukkasten efektiivinen määrä* n_{eff} kaavalla

$$n_{\text{eff}} \approx \frac{1}{\sum_{i=1}^N (w_k^{(i)})^2}$$

ja vertaamalla tätä vakioon $n \leq N$. Mikäli efektiivinen määrä on pienempi, suoritetaan uudelleenotanta.

Esitetään seuraavaksi hiukkassuotimen ja uudelleenotannan algoritmit.

Algoritmi 3.3.1. Hiukkassuodin voidaan kirjoittaa seuraavien vaiheiden avulla:

1. Otetaan N otosta priorijakaumasta

$$\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0), \quad i = 1, \dots, N,$$

ja asetetaan $w_0^{(i)} = \frac{1}{N}$ kaikille $i = 1, \dots, N$.

2. Jokaiselle ajanhettuelle ($k = 1, \dots, T$) suoritetaan seuraavat vaiheet:

A Otetaan otoksia $\mathbf{x}_k^{(i)}$ tärkeysjakaumista

$$\mathbf{x}_k^{(i)} = \pi(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k}), \quad i = 1, \dots, N.$$

B Lasketaan uudet painot seuraavasti:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(\mathbf{y}_k \mid \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})}{\pi(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{y}_{1:k})}.$$

Nämä normalisoidaan.

C Mikäli hiukkasten efektiivinen määrä on vähäinen, niin suoritetaan uudelleenotanta.

Algoritmi 3.3.2. Uudelleenotanta suoritetaan seuraavien vaiheiden avulla:

1. Muodostetaan painot $w_k^{(i)}$ todennäköisyysinä saada indeksi i joukosta $\{\mathbf{x}_k^{(i)} : i = 1, \dots, N\}$.
2. Otetaan N otosta tästä diskreetistä jakaumasta ja korvataan vanha otosjoukko tällä.
3. Asetetaan kaikki painot vakioksi: $w_k^{(i)} = \frac{1}{N}$.

Hiukkassuotimen algoritmin johto löytyy lähteestä [6].

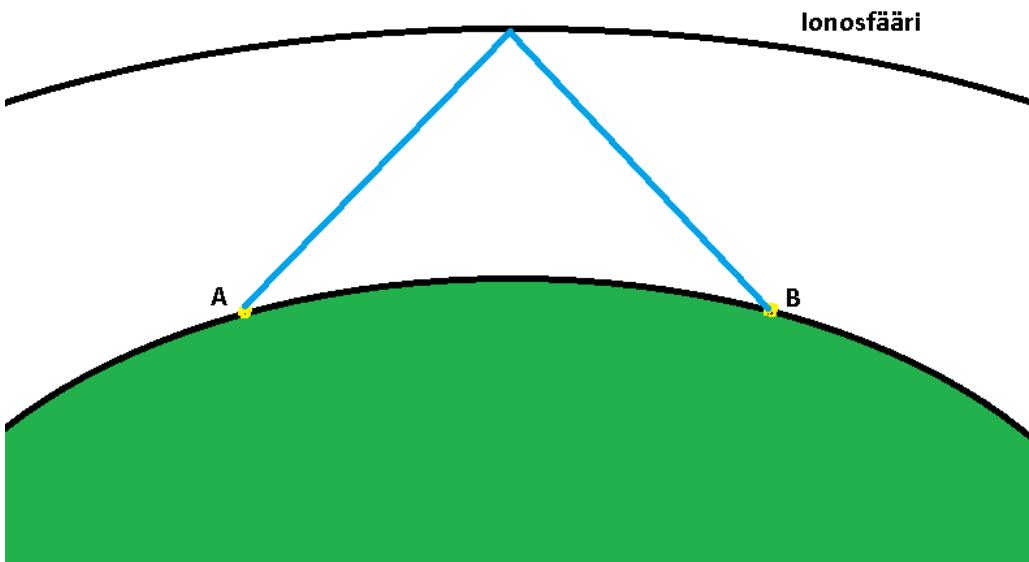
Niin kutsutussa *bootstrap-suotimessa* tärkeysjakaumana $\pi(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{y}_{1:k})$ käytetään evoluutiomallia $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. Tämän jakauman tehottomuuden vuoksi joudutaan yleensä käyttämään suurempaa määriä hiukkasia kuin tehokkaamman tärkeysjakauman tapauksessa. Bootstrap-suotimessa uudelleenotanta suoritetaan yleensä joka kierroksella.

4 Spektritiheysfunktion estimointi

Tässä kappaleessa sovelletaan laajennettuja Kalman-suotimia plasmaparametrien estimoimiseen. Kappale on koottu lähteistä [1], [3] ja [4].

4.1 Ongelman muotoilu

Ionosfääristä saadaan havaittua epäkoherenttia sirontaa, kun siihen lähettää signaali sirontatutkalla. Muuttunut signaali vastaanotetaan toisella asemalla. Tästä vastaanotetusta signaalista voidaan muodostaa autokovarianssi-



Kuva 1: Asema A lähettilä signaalini, joka vastaanotetaan asemalla B. Signaalissa tapahtuneet muutokset kertovat ionosfäärin ominaisuuksista.

ja spektritiheysfunktioit [3]. Niiden avulla saadaan tietoa ionisoituneen plasman ominaisuuksista, kuten elektronitiheydestä ja lämpötilasta [4]. Keskeytää jatkossa pelkästään spektritiheysfunktioiden tarkasteluun. Nämä riippuvat niin kutsutusta plasmaparametreista, joita pyritään estimoimaan. Tuttakalla mittaukset tehdään usein usealle eri korkeudelle.

Olkoon annettu spektritiheysfunktioparvi $S(f, \boldsymbol{\theta}(h))$, missä f on taajuus ja $\boldsymbol{\theta}(h)$ on parametrikektori, joka riippuu korkeudesta h . Ongelmana on löytää parametrien $\boldsymbol{\theta}(h)$ estimaatit tietyillä korkeuksilla Kalman-suotimen laajennuksien avulla.

Käytetään spektritiheysfunktion simulointiin *Lorentz-käyrää*. Se on muotoa

$$\frac{1}{1 + \left(\frac{f_0 - f}{\omega/2}\right)^2},$$

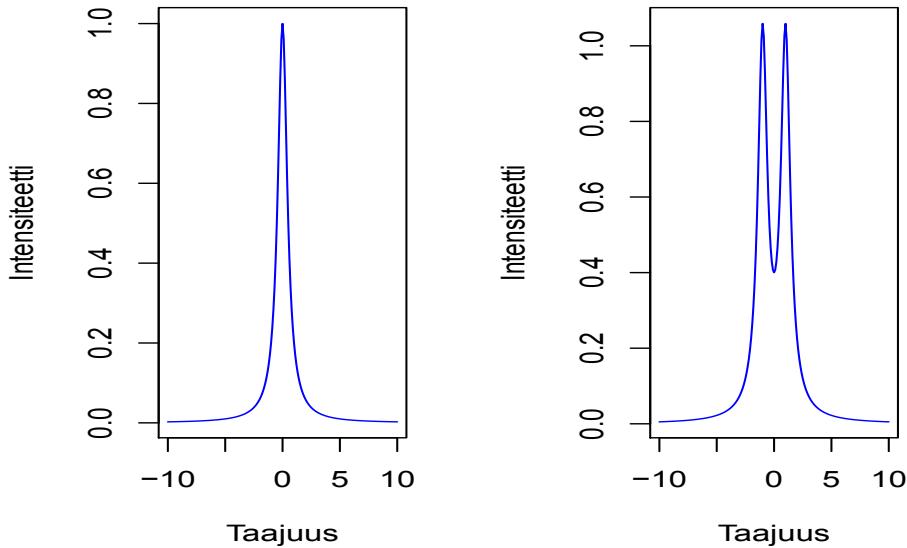
missä f_0 on huipun kohta, f on taajuus ja ω on puoliarvoleveys, eli sellaisien pisteiden välinen välimatka, jotka ovat arvoltaan puolet käyrän maksimiarvosta. Laskemalla yhteen kaksi Lorentz-käyrämuotoa voidaan simuloida

kaksihuippuista spektritiheysfunktiota. Tällöin saadaan funktio muotoa

$$L(f, \boldsymbol{\theta}(h)) := \frac{1}{1 + \left(\frac{f_{1,h} - f}{\omega_{1,h}/2}\right)^2} + \frac{1}{1 + \left(\frac{f_{2,h} - f}{\omega_{2,h}/2}\right)^2}, \quad (21)$$

missä $f_{1,h}$ ja $f_{2,h}$ ovat huippujen kohdat, $\omega_{1,h}$ on ensimmäisen ja $\omega_{2,h}$ toisen käyrän puoliarvoleveys. Parametrivektori on nyt muotoa

$$\boldsymbol{\theta}(h) = (f_{1,h}, \omega_{1,h}, f_{2,h}, \omega_{2,h}).$$



Kuva 2: Lorentz-käyrä (yksihuippuinen käyrä) ja kaksi eri Lorentz-käyrää summaamalla muodostettu kaksihuippuisen spektritiheysfunktion simulaatio.

Tarkastellaan yhtä korkeutta. Käytetään ongelman mallintamiseen tila-avaruusmallia

$$\begin{aligned} \boldsymbol{\theta}_k &= \boldsymbol{\theta}_{k-1} + \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1}), \\ y_{f,k} &= L(f, \boldsymbol{\theta}_k) + \mathcal{N}(0, R_k), \end{aligned}$$

missä $\boldsymbol{\theta}_k = (f_{1,k}, \omega_{1,k}, f_{2,k}, \omega_{2,k})$ on parametrivektori, $y_{f,k}$ on mittaus, L on määritelty kaavalla (21), $\mathbf{Q}_{k-1} = \sigma_{k-1} \mathbf{I}$, missä σ_{k-1} on vakio ja \mathbf{I} on yksikkö-matriisi, ja R_k on havaintomallin virheen varianssi.

Spektritiheysfunktion plasmaparametreja on tähän mennessä estimoitu Levenberg-Marquardt (LM) -menetelmällä. Johdetaan tämä seuraavaksi.

4.2 Levenberg-Marquardt -menetelmä

Olkoon annettu epälineaarinen yhtälöryhmä $\mathbf{G}(\boldsymbol{\theta}) = \mathbf{d}$, missä $\boldsymbol{\theta}$ on n -dimensioiden parametrivektori, vektori \mathbf{d} on m -dimensioinen datavektori ja \mathbf{G} on epälineaarinen kuvaus. Pyrkimyksenä on löytää sopiva parametrivektori $\boldsymbol{\theta}$ niin, että residuaalien normi minimoituu. Oletetaan, että mittauksen virheet mit ovat normaalijakautuneita keskivirheillä $\sigma_i, i = 1 \dots m$. Tarkoituksena olisi minimoida lauseke

$$f(\boldsymbol{\theta}) = \sum_{i=1}^m f_i^2(\boldsymbol{\theta}), \quad (22)$$

missä

$$f_i(\boldsymbol{\theta}) = \frac{G(\boldsymbol{\theta})_i - d_i}{\sigma_i}, \quad i = 1 \dots m,$$

on keskivirheillä painotettu residuaali. Käytetään minimoimiseen Newtonin menetelmää. Newtonin menetelmä on nyt

$$\mathbf{H}(f(\boldsymbol{\theta}^k))\Delta\boldsymbol{\theta} = -\nabla f(\boldsymbol{\theta}^k), \quad (23)$$

missä k on iteraatiokierros, $\mathbf{H}(f(\boldsymbol{\theta}^k))$ on funktion f Hessian matriisi pisteessä $\boldsymbol{\theta}^k$, $\Delta\boldsymbol{\theta}$ on tarkan ja likimäääräisen ratkaisun erotus iteraatiokierroksella k : $\Delta\boldsymbol{\theta} = \boldsymbol{\theta}^* - \boldsymbol{\theta}^k$ ja $\nabla f(\boldsymbol{\theta}^k)$ on funktion f gradientti pisteessä $\boldsymbol{\theta}^k$. Johdon Newtonin menetelmälle voi lukea lähteestä [1].

Funktion $f(\boldsymbol{\theta})$ gradientti voidaan kirjoittaa

$$\nabla f(\boldsymbol{\theta}) = \sum_{i=1}^m \nabla(f_i(\boldsymbol{\theta})^2).$$

Gradientin $\nabla f(\boldsymbol{\theta})$ komponentit voidaan kirjoittaa

$$(\nabla f(\boldsymbol{\theta}))_j = \sum_{i=1}^m 2f_i(\boldsymbol{\theta})(\nabla f_i(\boldsymbol{\theta}))_j,$$

mikä voidaan kirjoittaa matriisimerkinnöin muodossa

$$\nabla f(\boldsymbol{\theta}) = 2\mathbf{J}(\boldsymbol{\theta})^T \mathbf{F}(\boldsymbol{\theta}), \quad (24)$$

missä $\mathbf{J}(\boldsymbol{\theta})$ on Jacobin matriisi

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_1} & \dots & \frac{\partial f_1(\boldsymbol{\theta})}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\boldsymbol{\theta})}{\partial \theta_1} & \dots & \frac{\partial f_m(\boldsymbol{\theta})}{\partial \theta_n} \end{pmatrix},$$

ja matriisi $\mathbf{F}(\boldsymbol{\theta})$ on muotoa

$$\mathbf{F}(\boldsymbol{\theta}) = \begin{pmatrix} f_1(\boldsymbol{\theta}) \\ \vdots \\ f_m(\boldsymbol{\theta}) \end{pmatrix}.$$

Vastaavasti voidaan funktion $f(\boldsymbol{\theta})$ Hessen matriisi kirjoittaa kaavaa (22) käyttääen

$$\begin{aligned} \mathbf{H}(f(\boldsymbol{\theta})) &= \sum_{i=1}^m \mathbf{H}(f_i(\boldsymbol{\theta})^2) \\ &= \sum_{i=1}^m \mathbf{H}^i(\boldsymbol{\theta}), \end{aligned}$$

missä matriisi $\mathbf{H}^i(\boldsymbol{\theta})$ on funktion $f_i(\boldsymbol{\theta})^2$ Hessen matriisi.

Hessen matriisin $\mathbf{H}^i(\boldsymbol{\theta})$ j :nnen rivin ja k :nnen sarakkeen alkio on

$$\begin{aligned} H_{j,k}^i(\boldsymbol{\theta}) &= \frac{\partial^2(f_i(\boldsymbol{\theta})^2)}{\partial \theta_j \partial \theta_k} \\ &= \frac{\partial}{\partial \theta_j} \left(2f_i(\boldsymbol{\theta}) \frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_k} \right) \\ &= 2 \left(\frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_j} \frac{\partial f_i(\boldsymbol{\theta})}{\partial \theta_k} + f_i(\boldsymbol{\theta}) \frac{\partial^2 f_i(\boldsymbol{\theta})}{\partial \theta_j \partial \theta_k} \right). \end{aligned}$$

Näin ollen

$$\mathbf{H}(f(\boldsymbol{\theta})) = 2\mathbf{J}(\boldsymbol{\theta})^T \mathbf{J}(\boldsymbol{\theta}) + \mathbf{Q}(\boldsymbol{\theta}), \quad (25)$$

missä

$$\mathbf{Q}(\boldsymbol{\theta}) = 2 \sum_{i=1}^m f_i(\boldsymbol{\theta}) \mathbf{H}(f_i(\boldsymbol{\theta})).$$

Niin sanotussa *Gauss-Newton menetelmässä* Hessin matriisia arvioidaan jättämällä termi $\mathbf{Q}(\boldsymbol{\theta})$ pois yhtälöstä (25), jolloin saadaan

$$\mathbf{H}(f(\boldsymbol{\theta})) \approx 2\mathbf{J}(\boldsymbol{\theta})^T\mathbf{J}(\boldsymbol{\theta}). \quad (26)$$

Tämä arvio voidaan tehdä, koska kertoimien $f_i(\boldsymbol{\theta})$ voidaan olettaa olevan pieniä, kun lähestytään optimaalisia parametreiden $\boldsymbol{\theta}$ arvoja.

Sijoittamalla gradientti (24) ja Hessin matriisi (26) Newtonin menetelmään (23) saadaan iteraatioyhtälö

$$\mathbf{J}(\boldsymbol{\theta}^k)^T\mathbf{J}(\boldsymbol{\theta}^k)\Delta\boldsymbol{\theta} = -\mathbf{J}(\boldsymbol{\theta}^k)^T\mathbf{F}(\boldsymbol{\theta}^k), \quad (27)$$

missä k on iteraatioindeksi.

Levenberg-Marquardt -menetelmä (LM) saadaan, kun muokataan yhtälö (27) muotoon

$$(\mathbf{J}(\boldsymbol{\theta}^k)^T\mathbf{J}(\boldsymbol{\theta}^k) + \lambda\mathbf{I})\Delta\boldsymbol{\theta} = -\mathbf{J}(\boldsymbol{\theta}^k)^T\mathbf{F}(\boldsymbol{\theta}^k), \quad (28)$$

missä \mathbf{I} on yksikkämatriisi ja parametri $\lambda \in \mathbb{R}_+$ on positiivinen reaaliluku, joka päivitetään joka kierroksella iteraatiojonon $(\boldsymbol{\theta})_k$ suppenemisen varmistamiseksi. Parametreiden $\boldsymbol{\theta}$ päivittäminen tapahtuu ratkaisemalla muutos $\Delta\boldsymbol{\theta}$ kaavasta (28) sopivalla numeerisella menetelmällä ja laskemalla uuden kierroksen parametrit kaavalla $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \Delta\boldsymbol{\theta}$.

Termin $\lambda\mathbf{I}$ lisääminen pyrkii takaamaan, että matriisi $\mathbf{J}(\boldsymbol{\theta}^k)^T\mathbf{J}(\boldsymbol{\theta}^k) + \lambda\mathbf{I}$ olisi säädöllinen. Menetelmän toimivuus riippuu näin ollen keskeisesti parametrin λ valinnasta. Eräs idea on päivittää kyseistä parametria joka kierroksella riippuen residuaalin euklidisen normin

$$\left\| \frac{G(\boldsymbol{\theta})_i - d_i}{\sigma_i} \right\|, \quad i = 1 \dots m$$

suuruudesta. Mikäli kyseinen normi pienenee, pienennetään parametrin arvoa ja päinvastoin.

Tarkastellaan seuraavaksi estimointiongelmasta saatuja tuloksia.

4.3 Tulokset

Lasketaan spektritiheysfunktion parametreiden arvot yhdellä korkeudella laajennetulla, hajustamattomalla- ja bootstrap-suotimella. Verrataan näiden

menetelmien tarkkuutta LM -menetelmästä saatuihin tuloksiin. Estimointia varten muokatut algoritmit on toteutettu R-ohjelmointikielellä. Koodit ovat liitteinä.

Tarkastellaan edellä määriteltyä spektritiheysfunktion simulaatiota L , joka on määritelty kaavalla (21). Olkoon parametrien oikeat arvot

$$\boldsymbol{\theta} = (f_1, \omega_1, f_2, \omega_2) = (1, 2, -1, 2).$$

Olkoon alkuarvaus parametreille $\widehat{\boldsymbol{\theta}}_0 = (0.5, 1.5, -0.5, 1.5)$. Asetetaan evoluutiomallin ja priorijakauman kovarianssimatriisiin alkuarvoksi $\mathbf{Q}_0 = \mathbf{P}_0 = 0.01 \times \mathbf{I}$ ja havaintomallin varianssin alkuarvoksi $\mathbf{R}_0 = 0.01$. Asetetaan hajustamattomaan muunnokseen tarvittavat parametrit $\alpha = 0.001$, $\beta = 1$ ja $\kappa = 0.1$. Olkoon bootstrap-suotimen hiukkasten lukumäärä $N = 5000$. Olkoon LM-menetelmässä $\lambda_0 = 1$. Käytetään viimeiseltä kierrokselta saatua arvoa estimaattina kyseiselle korkeudelle.

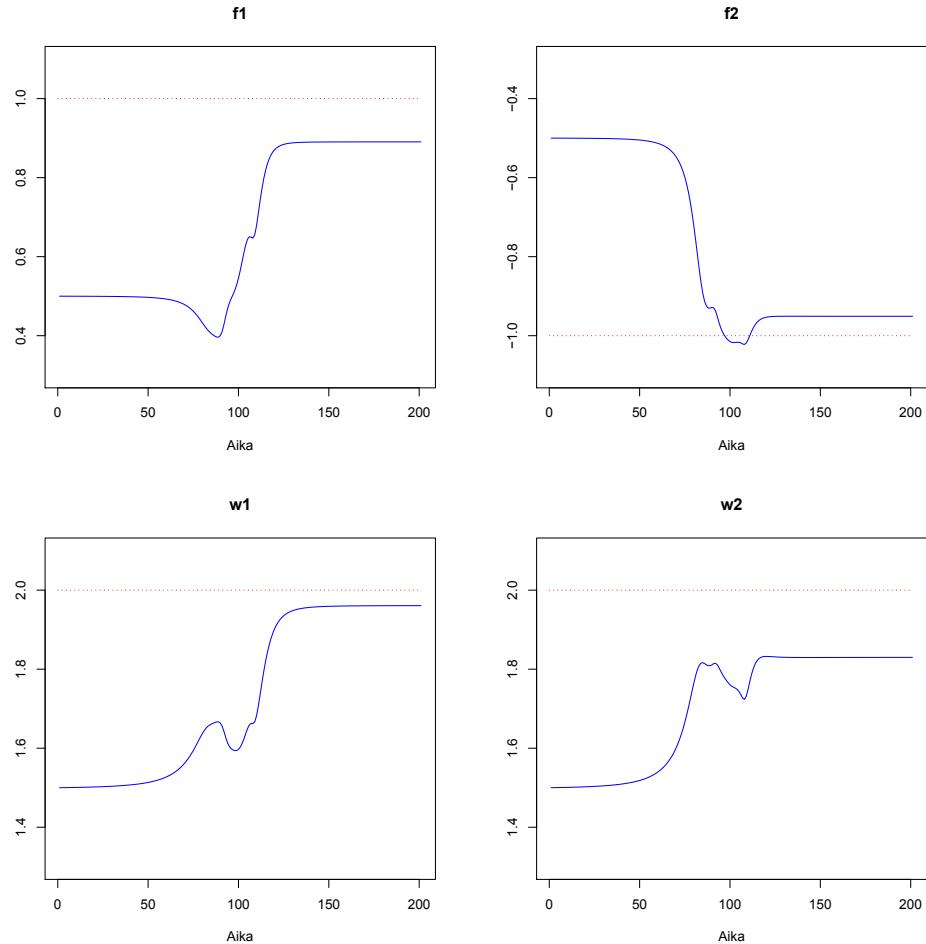
Ensimmäisen asteen laajennetulla Kalman-suotimella saadut estimaatit olivat pyöristettyinä $\widehat{\boldsymbol{\theta}}_{201} = (0.8905, 1.9608, -0.9509, 1.8299)$. Parametrien absoluuttiset virheet olivat

$$\begin{aligned} |\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{201}| &= (|f_1 - \widehat{f}_1|, |\omega_1 - \widehat{\omega}_1|, |f_2 - \widehat{f}_2|, |\omega_2 - \widehat{\omega}_2|) \\ &= (0.1095, 0.0392, 0.0491, 0.1701). \end{aligned}$$

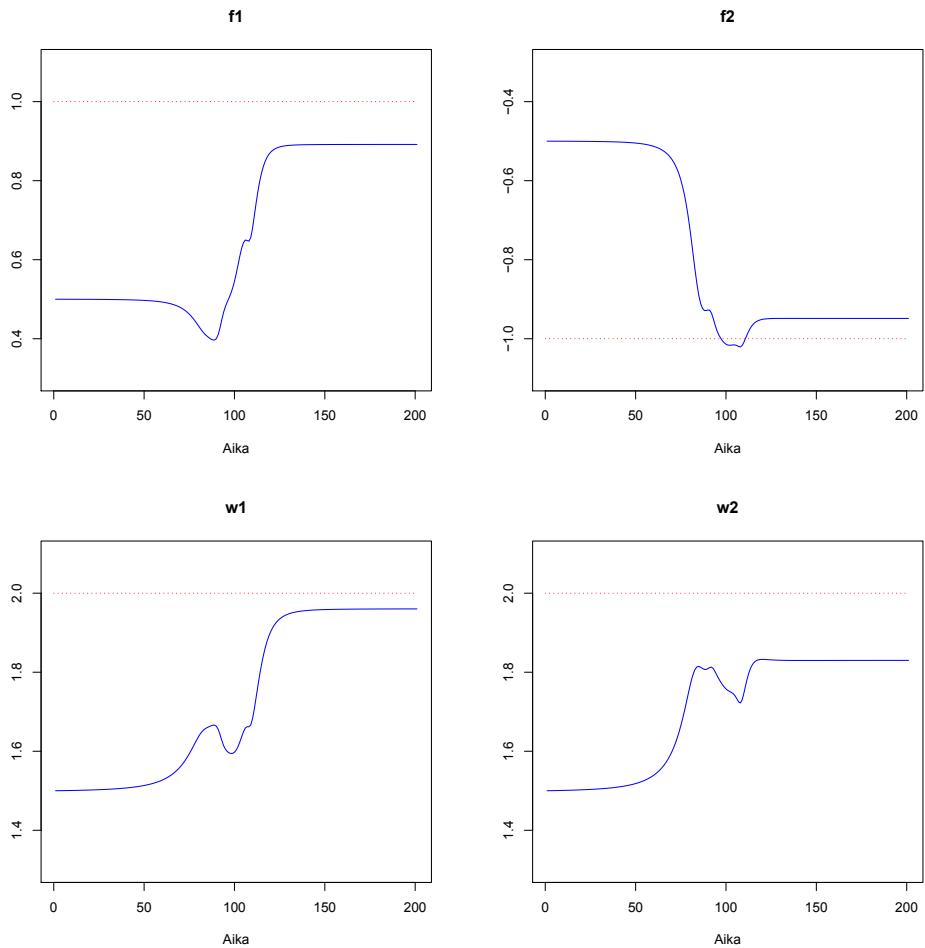
Aikaa menetelmän algoritmin suorittamiseen kului 0.01 sekuntia. Vastaavasti toisen asteen laajennetulla Kalman-suotimella saadut estimaatit olivat $\boldsymbol{\theta}_{201} = (0.8917, 1.9602, -0.9486, 1.830)$. Menetelmästä saadut virheet olivat $(0.1083, 0.0398, 0.0514, 0.1700)$. Algoritmin suorittaminen kesti 0.03 sekuntia.

Hajustamattomalla Kalman-suotimella saadut estimaatit olivat $\widehat{\boldsymbol{\theta}}_{201} = (0.8917, 1.9603, -0.9487, 1.8300)$. Tästä menetelmästä saadut virheet olivat $(0.1083, 0.0397, 0.0513, 0.1700)$. Aikaa algoritmin suorittamiseen kului 1.40 sekuntia. Bootstrap-suotimella ratkaisuna saatuiin estimaattien arvot $\widehat{\boldsymbol{\theta}}_{200} = (1.1474, 1.8576, -1.0868, 2.1525)$. Estimaattien virheet täällä menetelmällä olivat $(0.1474, 0.1424, 0.0868, 0.1525)$. Aikaa suorittamiseen kului 12.75 sekuntia.

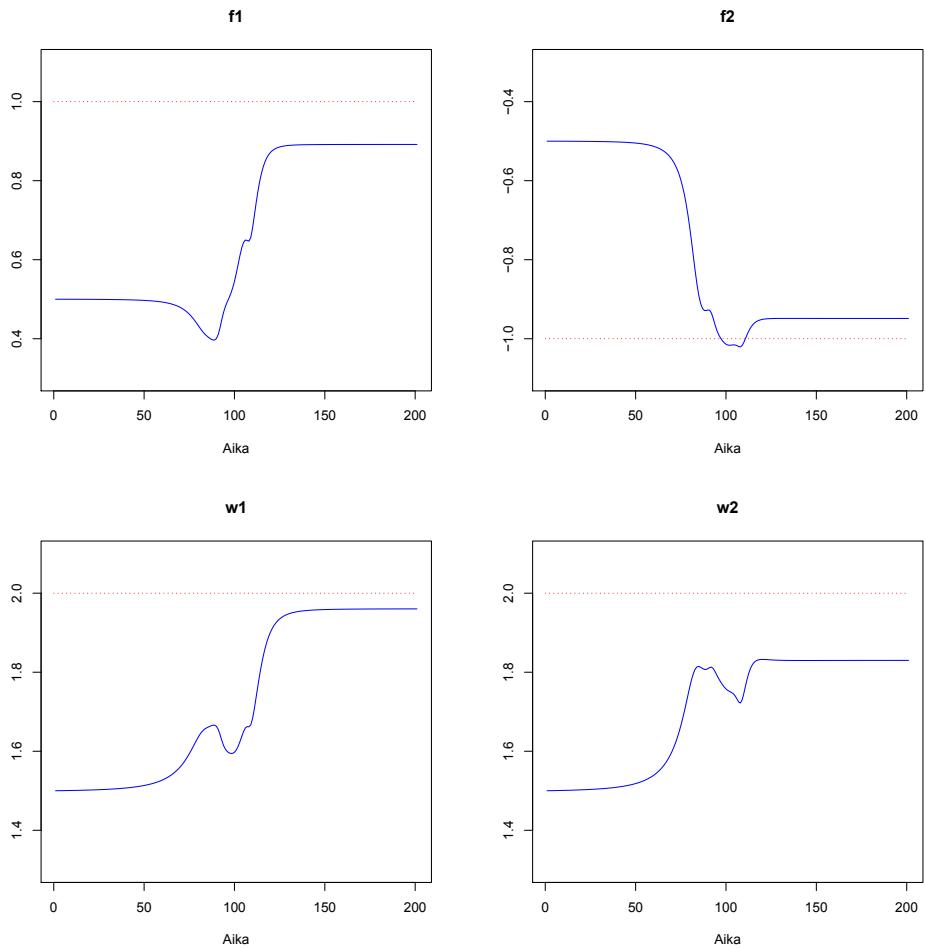
LM-menetelmää käyttäen oikeat parametrien arvot saavutettiin viidennellä kierroksella. Aikaa algoritmin suorittamiseen kului 0.11 sekuntia. Estimoinnin kulu eri menetelmillä voidaan nähdä seuraavista kuvista.



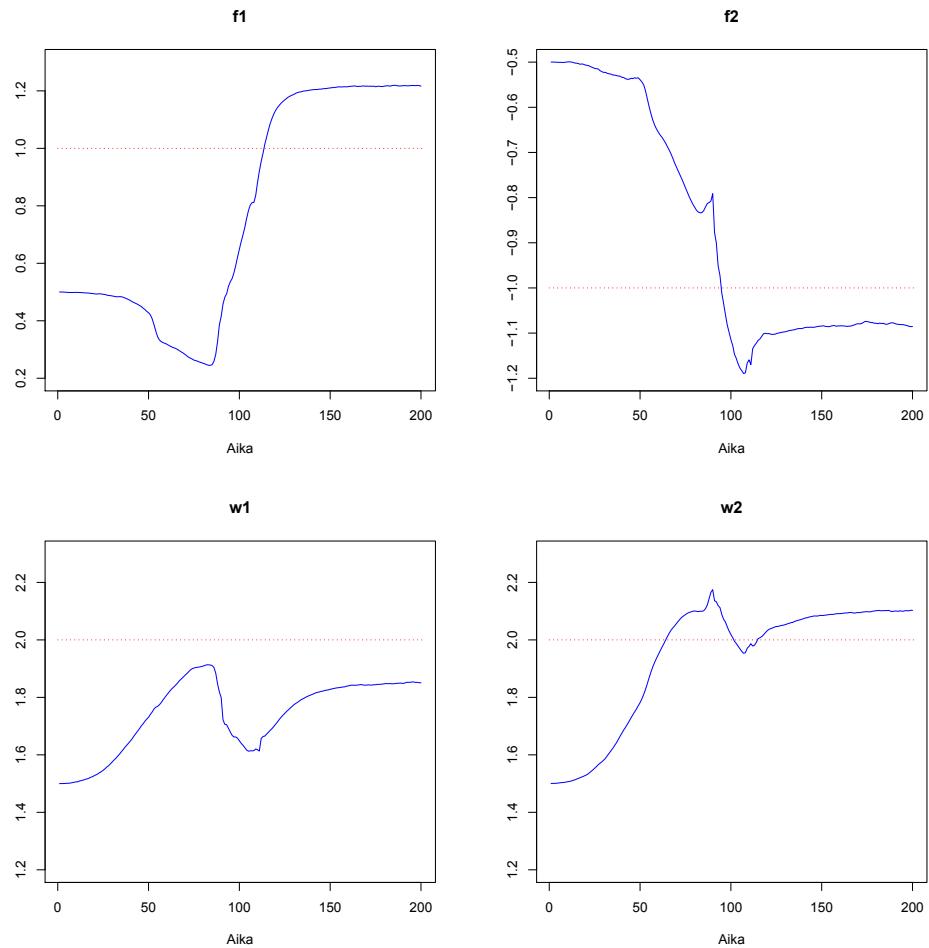
Kuva 3: Parametrien estimaatit ensimmäisen asteen laajennetulla Kalman-suotimella. Oikea parametrin arvo punaisella ja estimaatti sinisellä viivalla.



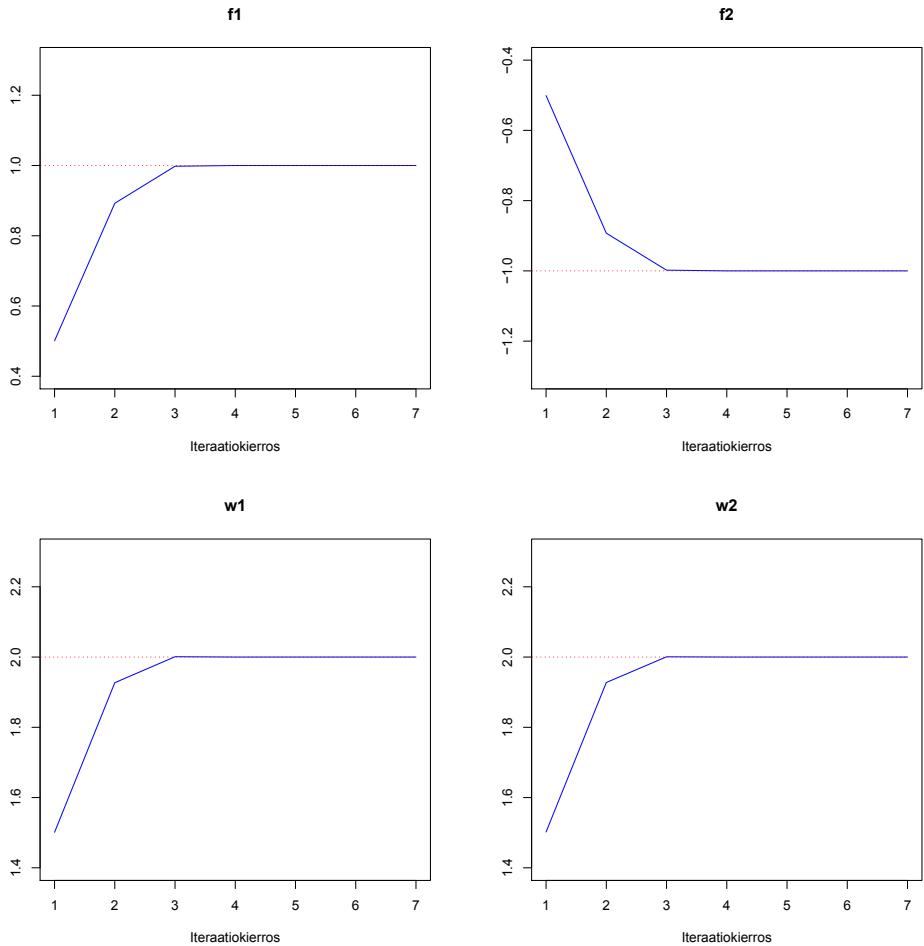
Kuva 4: Estimaatit toisen asteen laajennetulla Kalman-suotimella. Oikea parametrin arvo punaisella ja estimaatti sinisellä viivalla.



Kuva 5: Parametrien estimaatit hajustamattomalla Kalman-suotimella. Oikea parametrin arvo punaisella ja estimaatti sinisellä viivalla.



Kuva 6: Parametrien estimaatit bootstrap-suotimella. Oikea parametrin arvo punaisella ja estimaatti sinisellä viivalla.



Kuva 7: Parametrien estimaatit Levenberg-Marquardt -menetelmällä. Oikea parametrin arvo punaisella ja estimaatti sinisellä viivalla.

Edellä olevassa esimerkissä parametrien arvojen alkuarvaus oli melko huono. Hyvällä alkuarvauksella saadaan kaikilla suotimilla suhteellisen tarkat estimaatit. Esimerkiksi hajustamatonta- ja bootstrap-suodinta verrattaessa estimaatit edelliselle ongelmalle alkuarvauksella $\hat{\theta} = (0.9, 1.9, -0.9, 1.9)$ olivat $(0.9880, 1.9772, -0.9870, 1.9735)$ ja $(1.005, 1.9915, -1.0245, 2.0210)$. Vastaavat virheet näille suotimille olivat nyt $(0.0120, 0.0228, 0.0130, 0.0265)$ ja $(0.0050, 0.0085, 0.0245, 0.0210)$. LM-menetelmällä oikeat parametrien arvot

saavutettiin neljännellä kierroksella.

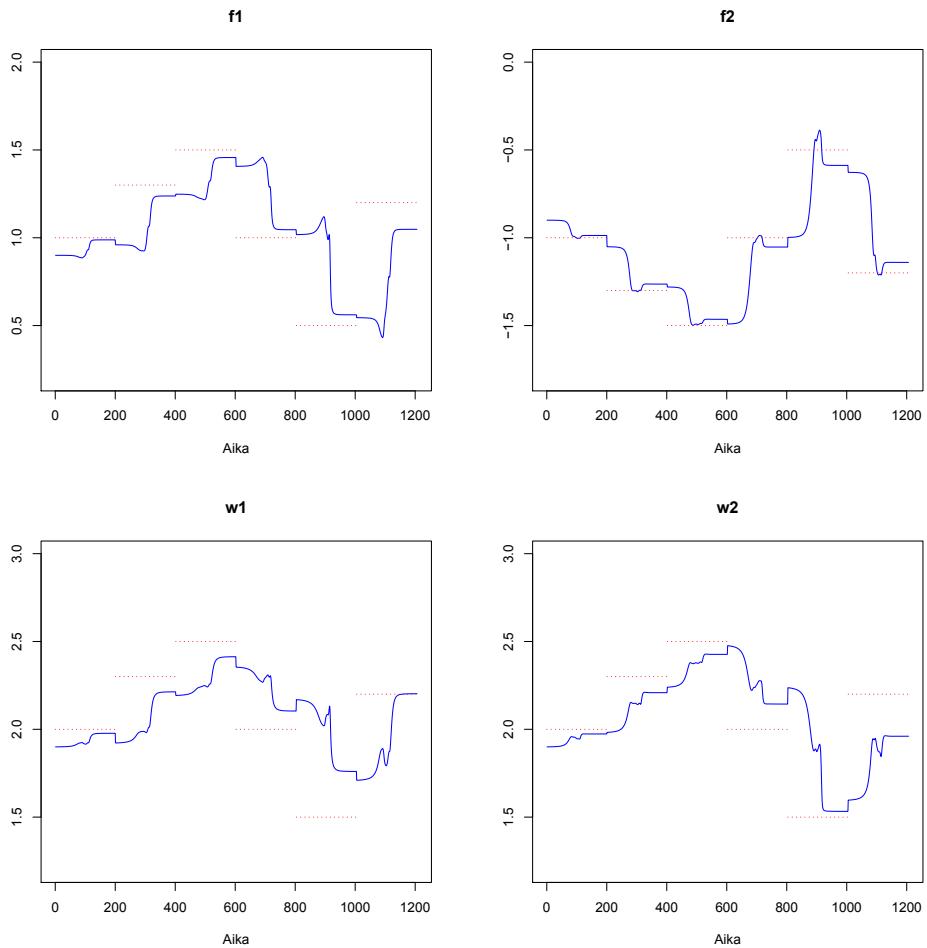
Tarkastellaan seuraavaksi parametrien estimointia usealla korkeudella. Käytetään satunnaishyppyä

$$\widehat{\boldsymbol{\theta}}_{h-1} + \frac{1}{10} |\boldsymbol{\theta}_h - \widehat{\boldsymbol{\theta}}_{h-1}| \times \mathcal{N}(\mathbf{0}, \mathbf{I})$$

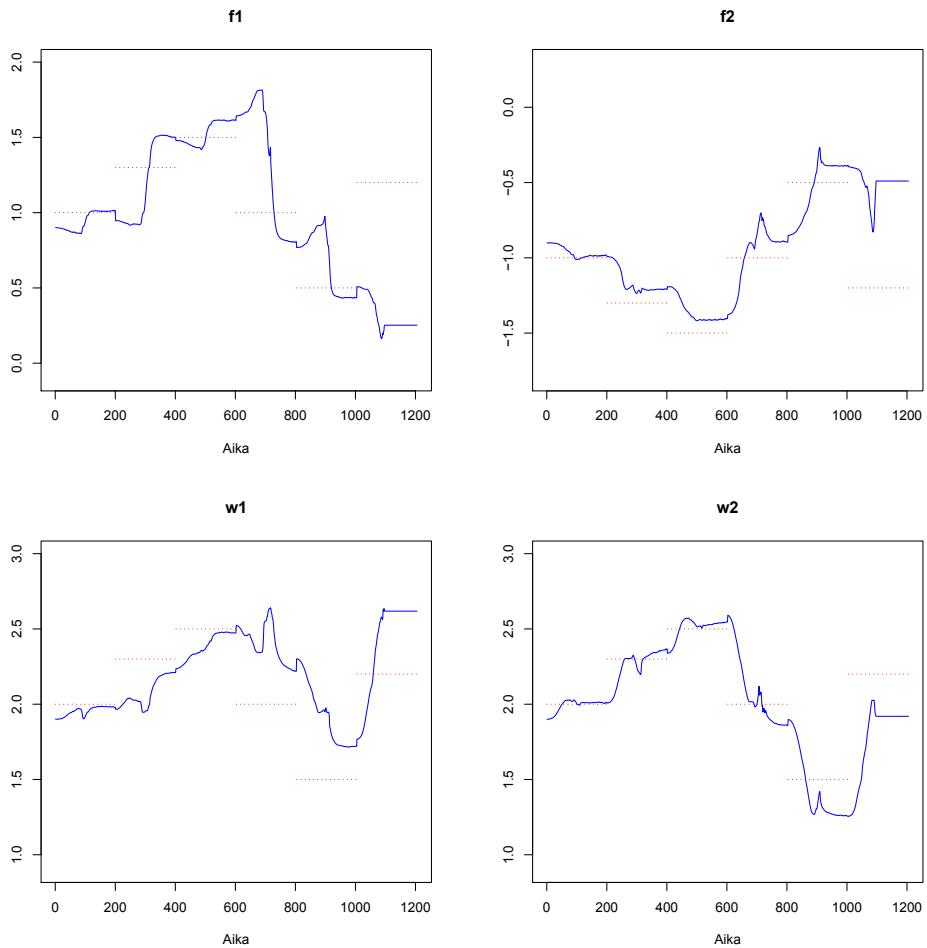
priorina korkeuden h parametreille, missä normi on määritelty, kuten aikaisemmin parametrien etäisyysien tarkastelussa. Alustetaan kohinoiden Kovarianssimatriisit alkuperäisiksi korkeuden vaihtuessa. Lasketaan estimaatit kuudella korkeudella priorin kanssa. Käytetään laskentaan toisen asteen laajennettua Kalman-suodinta, bootstrap-suodinta ja LM-menetelmää.

Parhaiten estimoinnissa onnistuttiin LM-menetelmällä. Huonoimmat estimaatit saatiin bootstrap-menetelmällä. Estimoinnin kuvat löytyvät seuraavalta sivulta.

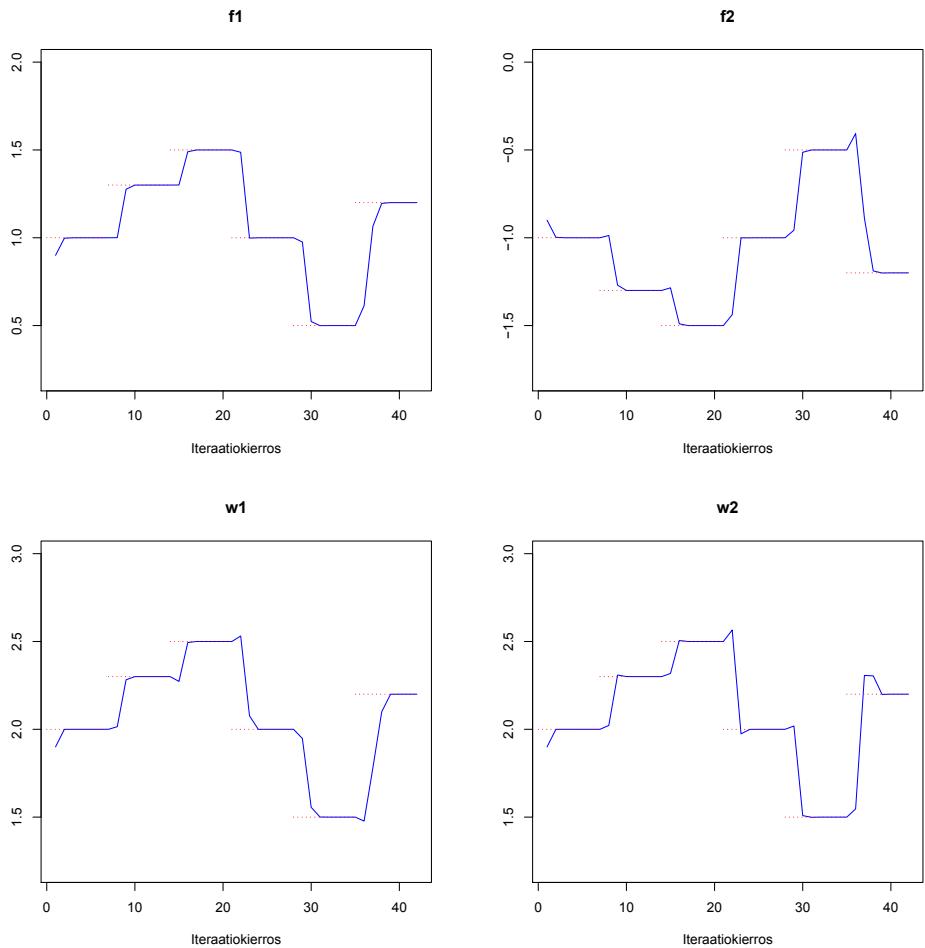
Yhteenvedona saatujen tulosten perusteella LM-menetelmä vaikuttaa olevan tarkempi tarkasteltuihin Kalman-suotimiin verrattuna. Se vaikuttaa olevan myös algoritmiltaan kevyempi kuin hajustamaton- ja bootstrap-suodin. Haitapuolena LM-menetelmässä on Jacobin matriisin laskenta, mikä voi olla työlästä, mikäli spektritiheysfunktion malli on monimutkainen. Tällöin hajustamaton Kalman-suodin tai hiukkassuodin saattaa olla parempi ratkaisu, jos tarkkuudesta voidaan tinkiä.



Kuva 8: Estimaatit toisen asteen laajennetulla Kalman-suotimella kuudelta korkeudelta. Oikeat parametrien arvot punaisella ja estimaatit sinisellä viivalla.



Kuva 9: Parametrien estimaatit bootstrap-suotimella kuudelta korkeudelta.
Oikeat parametrien arvot punaisella ja estimaatit sinisellä viivalla.



Kuva 10: Parametrien estimaatit LM-menetelmällä kuudelta korkeudelta. Oikeat parametrien arvot punaisella ja estimaatit sinisellä viivalla.

Lähdeluettelo

- [1] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter estimation and inverse problems*. Academic Press, 2013.
- [2] David R Brillinger. *Time series: data analysis and theory*, volume 36. Siam, 2001.
- [3] Baylie Damtie. *New Incoherent Scatter Radar Measurement Techniques and Data Analysis Methods*. Citeseer, 2004.
- [4] W. E. Gordon. Incoherent scattering of radio waves by free electrons with applications to space exploration by radar. *Proceedings of the IRE*, 46(11):1824–1829, Nov 1958.
- [5] Emanuel Parzen. *Stochastic processes*, volume 24. Siam, 1999.
- [6] Simo Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

Liitteet

A EKF

```
1 #####  
2  
3 # Function: EKF (First order Extended Kalman filter)  
4  
5 # Input:  
6  
7 # L: Measurement vector  
8 # m: Number of Lorentzian spectra  
9 # f: Frequency interval  
10 # hdistdiff: Difference between heights  
11 # theta0: Initial value for parameters of form (f1 ,f2 ,w1 ,w2)  
12 # P0: Coefficient of covariance matrix Pk  
13 # Q0: Initial error  
14 # R0: Initial measurement model error variance  
15  
16 # Output:  
17  
18 # thetak: Parameter estimate vector  
19  
20 #####  
21  
22 EKF <- function(L,m,f , hdistdiff ,theta0 ,P0,Q0,R0){  
23  
24 # Initializing  
25  
26  
27 # Define the Jacobian Hf:  
28  
29 Hf <- function(f1 ,theta0){  
30  
31 J <- mat.or.vec(1 ,4)  
32
```

```

33     W1 <- ((theta0[1]-f1)/(theta0[3]/2))^2
34     W2 <- ((theta0[2]-f1)/(theta0[4]/2))^2
35
36     J[1] <- -W1/((theta0[1]-f1)*(1+W1)^2)
37     J[2] <- -W2/((theta0[2]-f1)*(1+W2)^2)
38
39     J[3] <- W1/(theta0[3]*(1+W1)^2)
40     J[4] <- W2/(theta0[4]*(1+W2)^2)
41
42     J <- 2 * J
43
44     return(J)
45
46 }
47
48 # Define the Lorentzian spectrum:
49
50 spec <- function(f1,theta0){
51
52     x1 <- (theta0[1]-f1)/(theta0[3]/2)
53     x2 <- (theta0[2]-f1)/(theta0[4]/2)
54
55     return((1/(1+(x1^2))) + (1/(1+(x2^2))))
56
57 }
58
59 i <- 1
60
61 Qk <- Q0 * diag(4)
62
63 Pk <- P0 * diag(4)
64
65
66 thetak <- mat.or.vec(4,length(L))
67
68 Kk <- mat.or.vec(4,1)
69
70

```

```

71   f1 <- rep(f, length.out = length(L))
72
73   altit <- seq(length(L)/m, length(L), by = length(L)/m)
74
75
76   # Main loop
77
78
79   for(l in 1:length(L)){
80
81     # Prediction:
82
83     if ((l %in% altit) && (l != altit[length(altit)])){
84
85       i <- i + 1
86
87       theta0 <- theta0 + 1/10* abs(hdistdiff[i]) *
88         rnorm(4,0,1)
89
90       Qk <- Q0 * diag(4) #initializing
91
92       Pk <- P0 * diag(4)
93
94       Kk <- mat.or.vec(4,1)
95
96     }
97
98
99     # Update:
100
101
102     Hftemp <- Hf(f1[1],theta0)
103
104     vk <- L[1] - spec(f1[1],theta0)
105
106
107     Sk <- Hftemp %*% Pk %*% t(Hftemp) + R0
108     Kk <- (Pk %*% t(Hftemp)) %*% (1/Sk)

```

```

109     theta0 <- theta0 + (Kk * vk)
110     Pk <- Pk - (Kk %*% Sk %*% t(Kk))
111
112     thetak[,1] <- theta0
113
114 }
115
116
117
118 return(thetak)
119
120 }
```

B EKF2

```

1 #####
2
3 # Function: EKF2 (Second order Extended Kalman filter)
4
5 # Input:
6
7 # L: Measurement vector
8 # m: Number of Lorentzian spectra
9 # f: Frequency interval
10 # hdistdiff: Difference between heights
11 # theta0: Initial value for parameters of form (f1,f2,w1,w2)
12 # P0: Coefficient of covariance matrix Pk
13 # Q0: Initial error coefficient
14 # R0: Initial measurement model error variance
15
16 # Output:
17
18 # thetak: Parameter estimate vector
19
20 #####
21
22
```

```

23 EKF2 <- function(L,m,f , hdistdiff , theta0 , P0 , Q0 , R0) {
24
25   # Initializing
26
27   # Define the Jacobian Hf:
28
29   Hf <- function(f1 , theta0){
30
31     J <- mat.or.vec(1 , 4)
32
33     W1 <- ((theta0[1] - f1) / (theta0[3] / 2))^2
34     W2 <- ((theta0[2] - f1) / (theta0[4] / 2))^2
35
36     J[1] <- -W1 / ((theta0[1] - f1) * (1 + W1)^2)
37     J[2] <- -W2 / ((theta0[2] - f1) * (1 + W2)^2)
38
39     J[3] <- W1 / (theta0[3] * (1 + W1)^2)
40     J[4] <- W2 / (theta0[4] * (1 + W2)^2)
41
42     J <- 2 * J
43
44     return(J)
45
46   }
47
48
49   # Define the Hessian matrix Hff:
50
51   Hff <- function(f1 , theta0){
52
53     J <- mat.or.vec(4 , 4)
54
55     F1 <- theta0[1] - f1
56     F2 <- theta0[2] - f1
57
58     W1 <- 4 * ((F1 / theta0[3])^2)
59     W2 <- 4 * ((F2 / theta0[4])^2)
60

```

```

61      wW1 <- theta0[3] * (1+W1)
62      wW2 <- theta0[4] * (1+W2)
63
64      J[1,1] <- ((8*W1-8)/((wW1^2) * (1+W1))) +
65          (W1/((F1 * (1+W1))^2))
66      J[2,2] <- ((8*W2-8)/((wW2^2) * (1+W2))) +
67          (W2/((F2 * (1+W2))^2))
68
69      J[3,3] <- ((8*(F1^2)*(W1^2 - 1))/(wW1^4)) -
70          ((W1/(wW1^2)))
71      J[4,4] <- ((8*(F2^2)*(W2^2 - 1))/(wW2^4)) -
72          ((W2/(wW2^2)))
73
74      J[1,3] <- -(F1 * (8*W1 - 8))/(wW1^3)
75      J[3,1] <- J[1,3]
76
77      J[2,4] <- -(F2 * (8*W2 - 8))/(wW2^3)
78      J[4,2] <- J[2,4]
79
80      J <- 2 * J
81
82      return(J)
83
84  }
85
86
87 # Define the Lorentzian spectrum:
88
89 spec <- function(f1,theta0){
90
91     x1 <- (theta0[1]-f1)/(theta0[3]/2)
92     x2 <- (theta0[2]-f1)/(theta0[4]/2)
93
94     return((1/(1+(x1^2))) + (1/(1+(x2^2))))
95
96  }
97
98 i <- 1

```

```

99
100 Qk <- Q0 * diag(4)
101
102 Pk <- P0 * diag(4)
103
104
105 thetak <- mat.or.vec(4, length(L))
106
107 Kk <- mat.or.vec(4, 1)
108
109
110 f1 <- rep(f, length.out = length(L))
111
112 altit <- seq(length(L)/m, length(L), by = length(L)/m)
113
114
115
116 # Main loop
117
118 for(l in 1:length(L)){
119
120   # Prediction:
121
122   if((l %in% altit) && (l != altit[length(altit)])){
123
124     i <- i + 1
125
126     theta0 <- theta0 + 1/10* abs(hdistdiff[i]) *
127       rnorm(4, 0, 1)
128
129     Qk <- Q0 * diag(4) #initializing
130
131     Pk <- P0 * diag(4)
132
133     Kk <- mat.or.vec(4, 1)
134
135   }
136

```

```

137      # Update:
138
139
140      Hftemp <- Hf( f1 [ 1 ] , theta0 )
141
142      Hfftemp <- Hff( f1 [ 1 ] , theta0 )
143
144      vk <- L[ 1 ] - spec ( f1 [ 1 ] , theta0 ) -
145          0.5 * (sum( diag( Hfftemp %*% Pk )) )
146
147
148      Sk <- (Hftemp %*% Pk %*% t (Hftemp)) +
149          0.5 * sum( diag( Hfftemp %*% Pk %*% Hfftemp %*% Pk )) + R0
150      Kk <- (Pk %*% t (Hftemp)) %*% (1 / Sk)
151      theta0 <- theta0 + (Kk * vk)
152      Pk <- Pk - (Kk %*% Sk %*% t (Kk))
153
154      thetak [ , 1 ] <- theta0
155
156  }
157
158  return (thetak)
159
160 }
```

C UKF

```

1 #####
2
3 # Function: UKF (Unscented Kalman filter)
4
5 # Input:
6
7 #   L: Measurement vector
8 #   m: Number of Lorentzian spectra
9 #   f: Frequency interval
10 #  hdistdiff: Difference between heights
```

```

11 #     theta0: Initial value for parameters of form (f1 ,f2 ,w1,w2)
12 #     P0: Coefficient of covariance matrix Pk
13 #     Q0: Initial error coefficient
14 #     R0: Initial measurement model error variance
15 #     Parameters for the unscented transform: alpha , kappa , beta
16
17 # Output:
18
19 #     thetak: Parameter estimate vector
20
21 #####
22
23
24 UKF <- function (L,m,f , hdistdiff ,theta0 ,P0,Q0,R0,
25           alpha ,kappa ,beta){
26
27   # Initializing
28
29   j <- 1
30
31   f1 <- rep(f , length.out = length(L))
32
33   thetak <- mat.or.vec (4,length(L))
34
35   Pk <- P0 * diag(4)
36   Qk <- Q0 * diag(4)
37
38
39   n <- dim(theta0 )[1]
40   CHI <- mat.or.vec(n , (2*n) +1)
41   Y <- mat.or.vec(1 , (2*n) +1)
42
43   altit <- seq(length(L)/m, length(L) , by = length(L)/m)
44
45   # Coefficients
46
47   lambda <- (alpha ^2)*(n+kappa) - n # Scaling parameter
48

```

```

49 Wm <- mat.or.vec(1 , (2*n) +1)
50
51 Wm[1 ,] <- 1 / (2*(n + lambda))
52 Wc <- Wm
53
54 Wm[ ,1] <- lambda/(n + lambda)
55 Wc[ ,1] <- Wm[ ,1] + (1-(alpha^2) + beta)
56
57 sqcoef <- sqrt(n+lambda)
58
59 # Define the Lorentzian spectrum
60
61
62 spec <- function(f1 ,f1 ,f2 ,w1,w2){
63
64   x1 <- (f1-f1)/(w1/2)
65   x2 <- (f2-f1)/(w2/2)
66
67   return(1/(1+x1^2) + 1/(1+x2^2))
68
69 }
70
71
72 cat(c("\n","Number of altitudes computed:","\n","\n"))
73
74
75 # Main loop
76
77 for(l in 1:length(L)){
78
79   # Prediction:
80
81   CHI[ ,1] <- theta0
82
83   if ((l %in% altit) && (l != altit [length(altit)])){
84
85     j <- j + 1
86

```

```

87      CHI[ ,1] <- theta0 + 1/10* abs( hdistdiff [j]) *
88      rnorm (4 ,0 ,1)
89
90      Qk <- Q0 * diag (4) #initializing
91
92      Pk <- P0 * diag (4)
93
94      Kk <- mat.or.vec (4 ,1)
95
96      }
97
98
99      sqP <- try ( matrix (as.numeric (sqrtm (Pk)) ,4 ,4) , silent = TRUE)
100
101
102      for (i in 2:(n+1)){
103
104          CHI[ ,i] <- CHI[ ,1] + (sqcoef * sqP[ ,i-1])
105          CHI[ ,i+n] <- CHI[ ,1] - (sqcoef * sqP[ ,i-1])
106
107      }
108
109
110      theta0 <- matrix (c (0 ,0 ,0 ,0) ,4 ,1)
111
112
113      for (i in 1:(2*n + 1)){
114
115          theta0 <- theta0 + (Wm[ ,i] * CHI[ ,i])
116
117      }
118
119      # Pkline:
120
121
122      Pk <- mat.or.vec (4 ,4)
123
124

```

```

125      for(i in 1:(2*n + 1)){
126
127          Pk <- Pk + (Wc[,i] * (CHI[,i] - theta0) %*%
128                      t(CHI[,i] - theta0))
129
130      }
131
132
133      # Update:
134
135
136      # Predicted matrices:
137
138      # mkline:
139
140      CHI[,1] <- theta0
141
142      sqPline <- try(matrix(as.numeric(sqrtm(Pk)), 4, 4),
143                           silent = TRUE)
144
145
146      for(i in 2:(n+1)){
147
148          CHI[,i] <- CHI[,1] + (sqcoef * sqPline[,i-1])
149          CHI[,i+n] <- CHI[,1] - (sqcoef * sqPline[,i-1])
150
151      }
152
153      # Propagate the sigma points:
154
155
156
157      for(i in 1:(2*n + 1)){
158
159          Y[,i] <- spec(f1[1], CHI[1,i], CHI[2,i], CHI[3,i], CHI[4,i])
160
161      }
162

```

```

163
164      # The needed submatrices:
165
166      # muk:
167
168      muk <- 0
169
170      for (i in 1:(2*n + 1)){
171          muk <- muk + (Wm[, i] * Y[, i])
172
173      }
174
175
176      # Sk:
177
178      Sk <- 0
179
180      for (i in 1:(2*n + 1)){
181          Sk <- Sk + (Wc[, i] * (Y[, i] - muk)^2)
182
183      }
184
185
186      Sk <- Sk + R0
187
188      # Ck:
189
190      Ck <- mat.or.vec(4, 1)
191
192
193      for (i in 1:(2*n + 1)){
194          Ck <- Ck + (Wc[, i] * (CHI[, i] - theta0) * (Y[, i] - muk))
195
196      }
197
198
199      # Calculation of needed matrices:
200

```

```

201     Kk <- Ck %*% (1 / Sk)
202     theta0 <- theta0 + Kk %*% (L[1] - muk)
203     Pk <- Pk - Kk %*% Sk %*% t(Kk)
204
205     thetak[, 1] <- theta0
206
207     # Printing:
208
209     if (l %in% altit) cat(c(round((m*l)/length(L)), " "))
210
211 }
212
213 return(thetak)
214
215 }
```

D BF

```

1 #####
2
3 # Function: BF (Bootstrap filter)
4
5 # Input:
6
7 # L: Measurement vector
8 # f: Frequency interval
9 # N: Number of particles
10 # theta0: Initial value for parameters of form (f1,f2,w1,w2)
11 # P0: Coefficient of covariance matrix Pk
12 # Q0: Initial error coefficient
13 # R0: Initial measurement model error variance
14
15 # Output:
16
17 # thetak: Parameter estimate vector
18
19 #####
```

```

20
21
22 BF <- function(L,f,N,theta0,P0,Q0,R0){
23
24   # Initialize:
25
26   thetak <- mat.or.vec(4,N)
27
28   solution <- mat.or.vec(4,length(L))
29
30   wk <- mat.or.vec(N,1)
31
32
33   spec <- function(f,thetak){
34
35     x1 <- (thetak[1]-f)/(thetak[3]/2)
36     x2 <- (thetak[2]-f)/(thetak[4]/2)
37
38     return(1/(1+x1^2) + 1/(1+x2^2))
39
40   }
41
42
43   # Draw samples from the evolution model
44   # and calculate the weights
45
46   # Prior distribution:
47
48   for(i in 1:N){
49
50     thetak[,i] <- rnorm(4, mean = theta0, sd = P0)
51
52   }
53
54   for(k in 1:length(L)){
55
56     for(i in 1:N){
57

```

```

58     thetak[ , i ] <- rnorm( 4 , mean = thetak[ , i ] ,
59                               sd = Q0)
60
61   }
62
63   for (i in 1:N){
64
65     wk[ i ] <- dnorm( L[ k ] , mean = spec(f[ k ] , thetak[ , i ]) ,
66                         sd = R0)
67
68   }
69
70   if (sum(wk) == 0){
71
72     for (j in k:length(L)){
73
74       solution[ , j ] <- solution[ , k-1]
75
76     }
77
78   return(solution)
79
80 # Normalize
81
82 wk <- wk/sum(wk)
83
84
85 # The solution
86
87 sol <- mat.or.vec(4,1)
88
89 for (i in 1:N){
90
91   sol <- sol + wk[ i ] *thetak[ , i ]
92
93 }
94
95 solution[ , k ] <- sol

```

```

96
97
98     # Resampling
99
100
101    theta_column <- sample(1:N, replace = TRUE ,
102                           prob = wk )
103
104    thetak <- thetak[,theta_column]
105
106
107 }
108
109
110 return(solution)
111
112 }
```

E LM

```

1 #####
2
3 # Function: LM (Levenberg–Marquardt–method)
4
5 # Input:
6
7 #   f: Frequency interval
8 #   solution: Parameter expectation
9 #   theta0: Initial value for parameters of form (f1,f2,w1,w2)
10 #  iter: Number of iterations
11 #  R0: Initial measurement model error variance
12 #  lambda: Parameter for LM
13
14 # Output:
15
16 #   Parameter estimation vector: thetak
17
```

```

18 #####
19
20 LM <- function(f, solution, theta0, iter, R0, lambda){
21
22
23 # Initialize:
24
25 chi2 <- 0; chi2old <- 1
26
27 thetak <- mat.or.vec(4, iter+1)
28
29 thetak[,1] <- theta0
30
31
32 L <- function(f, solution){
33
34   L <- mat.or.vec(length(f), 1)
35
36   x1 <- (solution[1]-f)/(solution[3]/2)
37   x2 <- (solution[2]-f)/(solution[4]/2)
38
39   L <- 1/(1+x1^2) + 1/(1+x2^2)
40
41   return(as.vector(L))
42
43 }
44
45
46 data <- L(f, solution)
47
48
49 G <- function(i, mr){
50
51   L <- 0
52
53   x1 <- (mr[1]-i)/(mr[3]/2)
54   x2 <- (mr[2]-i)/(mr[4]/2)
55

```

```

56     L <- (1 / (1+x1^2) + 1/(1+x2^2))
57
58     return (L)
59
60 }
61
62
63 Ff <- function(mr,f,R0,data){
64
65     Fr <- mat.or.vec(length(f),1)
66
67     for(i in 1:length(f)){
68
69         Fr[i] <- G(f[i],mr) - data[i]
70
71     }
72
73     Fr <- Fr * (1/R0)
74
75
76     return(Fr)
77 }
78
79
80 Hf <- function(i,mr){
81
82     J <- mat.or.vec(1,4)
83
84     W1 <- ((mr[1]-i)/(mr[3]/2))^2
85     W2 <- ((mr[2]-i)/(mr[4]/2))^2
86
87     J[1] <- -W1/((mr[1]-i)*(1+W1)^2)
88     J[2] <- -W2/((mr[2]-i)*(1+W2)^2)
89
90     J[3] <- W1/(mr[3]*(1+W1)^2)
91     J[4] <- W2/(mr[4]*(1+W2)^2)
92
93     J <- 2 * J

```

```

94
95      return (J)
96
97  }
98
99
100 J <- function(f ,mr ,R0){
101
102   M <- mat.or.vec(length(f ),4)
103
104   for(j in 1:length(f )){
105
106     for(k in 1:4){
107
108       M[j ,k ] <- Hf(f [j ],mr )[ ,k ]
109
110     }
111
112   }
113
114   M <- M*(1/R0)
115
116
117   return(M)
118
119 }
120
121
122 # Main loop
123
124
125 for(k in 1:iter ){
126
127   Japu <- J(f ,thetak[,k ],R0)
128
129   chi2 <- sum((Ff(thetak[,k ],f ,R0,data)^2))
130
131

```

```

132     if (is.na(chi2)) return(theta.k)
133
134     if (chi2 >= chi2.old){
135
136         lambda = lambda * 10
137
138     } else{
139
140         lambda = lambda/10
141
142     }
143
144     chi2.old <- chi2
145
146     dtheta <- -1* (solve(t(Japu) %*% Japu
147                           +(lambda*diag(4))) %*%
148                           (t(Japu) %*% Ff(theta.k[,f],R0,data)))
149
150     theta.k[,k+1] <- theta.k[,k] + dtheta
151
152 }
153
154 return(theta.k)
155
156 }
```