

Spatiaalisen ikkunoinnin ja sokean
signaalinerottelun menetelmien hyödyntäminen
MEG-aineiston analysoinnissa

Karita Hakala

Tilastotieteen pro gradu -tutkielma

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
29. tammikuuta 2018

JYVÄSKYLÄN YLIOPISTO

Matematiikan ja tilastotieteen laitos

Hakala, Karita: *Spatiaalisen ikkunoinnin ja sokean signaalinerottelun menetelmien hyödyntäminen MEG-aineiston analysoinnissa*

Tilastotieteen pro gradu -tutkielma (45 sivua) + liite (5 sivua)

Tiivistelmä

Sokean signaalinerottelun menetelmät ovat käyttökelpoisia magnetoenkefalografilla mitattujen aivomagneettikäyrien analysoinnissa. Sokean signaalinerottelun menetelmien avulla voidaan selvittää, mistä aivoaktivaatio on peräisin ja millaisiin toimintoihin aktivaatio voidaan yhdistää. MEG-aineiston ajallinen tarkkuus on hyvä, joten lyhytkin mittausaika tuottaa hyvin suuren aineiston.

Suuren aineiston tuottamat laskennalliset ongelmat voidaan kiertää ikkunoinnilla MEG-aineisto aika- tai tilasuunnassa. Aineiston ikkunointia on sovellettu aiemmissa tutkimuksissa toiminnallisen magneettikuvauksen aineistoon, jolloin ikkunointi on tehty aikasuunnassa. Tässä tutkielmassa ikkunointia sovellettiin MEG-aineistoon tilasuunnassa, jolloin kaikkien MEG-sensorien yhtäaikaisen analysoinnin sijaan käytetään vain osaa sensoreista kerrallaan. Tällöin moniulotteiseen MEG-aineistoon voidaan käyttää sokean signaalinerottelun menetelmiä pienemmissä osioissa ja MEG-aineistosta voidaan löytää vaikeasti erotettavissa olevia, pieniamplitudisia lähdesignaaleja.

Spatiaalisen ikkunoinnin hyödyllisyyttä tutkittiin simulointiaselman ja MEG-aineiston avulla. Simulointiaselmassa pyrittiin selvittämään, onko ikkunointimenetelmästä hyötyä sokean signaalinerottelun menetelmiä sovellettaessa. Simulointi toteutettiin MEG-mittausasetelmaa mukaillen. MEG-aineisto koostui yhden koehenkilön noin neljän minuutin MEG-mittauksesta, jossa tarkoituksena oli kontrolloida koehenkilön näköjärjestelmän toimintaa.

Simulointiaselman tulosten perusteella spatiaalisesta ikkunoinnista on hyötyä sokean signaalinerottelun menetelmistä erityisesti riippumattomien komponenttien analyysin (ICA) käytön yhteydessä, eikä ikkunoinnista ole ainakaan haittaa toisen asteen lähde-erottelumallin (SOS) käytön yhteydessä. MEG-aineiston tapauksessa ikkunointi ei herätevasteiden tasolla pystynyt erottelemaan haluttua lähdesignaalia tarkasti, mutta lähdesignaaleiden topografiat vastasivat oletetunlaisia lähdesignaalien jakaumia.

Avainsanat: Sokea signaalinerottelu (BSS), riippumattomien komponenttien analyysi (ICA), toisen asteen lähde-erottelumallit (SOS), ikkunointi, magnetoenkefalografia (MEG), aivotutkimus, simulointi

Sisältö

1 Johdanto	1
2 Sokea signaalinerottele	4
2.1 Valkaisu	5
2.2 Yhteisdiagonalisointi	7
3 Toisen asteen lähde-erottele	10
3.1 Toisen asteen sokea tunnistus (SOBI)	10
4 Riippumattomien komponenttien analyysi	12
4.1 Keskinäinen tilastollinen riippumattomuus	12
4.2 Neljännen asteen tunnusluvut	13
4.3 Kumulanttimatriisien approksimatiivinen yhteisdiagonalisointi (JA- DE & k-JADE)	14
4.4 Epänormaalisuusmittojen maksimointi (FastICA)	15
5 Spatiaalinen ikkunointi sokeassa signaalinerottelussa	17
6 Aivojen sähköisen toiminnan mittaaminen ja aineiston ikkunointi	19
6.1 Simulointi	20
6.2 Sovellus	28
7 Yhteenveto	40
Viitteet	41
Liite: Simulointiasetelman R-koodi	45

1 Johdanto

Ihmisaivoissa on noin 86 miljardia hermosolua, joiden väliseen sähkökemialliseen viestintään aivojen toiminta perustuu (Azevedo ym., 2009). Aivojen on reagoitava jatkuvasti ulkoisiin ja sisäisiin ärsykkeisiin, eivätkä ne ole levossaakaan toimettomat. Hermosolujen aktivoitumisen aiheuttamia sähkövirtoja voidaan tutkia pään ulkopuolelta, ja niistä voidaan tehdä päätelmiä aivotoiminnan synnystä ja mekanismeista.

Aivotoimintojen tutkimiseen on kehitetty erilaisia kuvantamismenetelmiä, jotka perustuvat aivojen biologisiin ominaisuuksiin. Esimerkiksi funktionaalinen magneettikuvaus (functional magnetic resonance imaging, fMRI) käyttää hyväksi aivojen verenkiertoa aivoaktivaatioalueiden selvittämiseksi. Elektroencefalografia (electroencephalography, EEG) mittaa aktivoituneiden hermosolujen aiheuttamia sähköisiä potentiaaleja ja magnetoencefalografia (magnetoencephalography, MEG) mittaa näiden sähköisten potentiaalien aiheuttamien magneettikenttien voimakkuuksia. Kaikkien edellä mainittujen menetelmien mittaukset tehdään pään ulkopuolelta, joten ne eivät vaadi kirurgisia toimenpiteitä. Tällaisten menetelmien ongelmana on usein joko alueellinen tai ajallinen epätarkkuus: Funktionaalinen magneettikuvaus antaa tarkasti tietoa siitä, missä aivoaktivaatio tapahtuu, mutta ei kykene erottelamaan aivoaktivaatiota täsmällisemmin kuin sekuntien tarkkuudella. MEG kykenee millisekunnin erottelutarkkuuteen, mutta sen alueellinen erottelukyky rajoittuu millimetreihin. Aivokuvantamismenetelmän valinta riippuu siitä, millaista tietoa aivoista tai niiden toiminnasta halutaan selvittää. Toiminnallisella magneettikuvantamisella ei pystytä selvittämään aivoaktivaation tarkkaa alkamisaikaa eikä luotettavasti sen kokonaiskesto, ja toisaalta EEG ja MEG eivät sovellu esimerkiksi aivojen rakenteen tutkimiseen. Äskettäin julkaistu koonti edellä mainituista menetelmistä löytyy esimerkiksi teoksesta Papanicolaou (2017).

MEG-tutkimuksessa kiinnostuksen kohteena voivat olla syntyvän aivovasteen ominaisuudet: MEG-signaalin muoto, kesto ja voimakkuus. Tällaista tutkimusta kutsutaan herätevastetutkimukseksi. Herätevastepotentiaali (event related potential, ERP) on hermosolujen synnyttämä sähkövaraus, jonka havaitsemiseksi kymmenientuhansien hermosolujen on reagoitava ärsykkeeseen yhtäaikaisesti. Herätevastekenttä (event related field, ERF) on kyseisen sähkövarauksen aiheuttama magneettikenttä, jonka voimakkuutta MEG:llä toteutetuissa tutkimuksissa mitataan. Herätevastetutkimukset vaativat koeasetelman, jossa ärsykettä toistetaan koehenkilölle kymmeniä kertoja. Yleensä aivosignaaleista erotellaan ärsykkeen vaatiman reaktioajan mittaiset osiot, joiden yli signaalit keskiarvoistetaan. Keskiarvosignaalin ajatellaan vastaavan ärsykkeen synnyttämää aivovastetta (Walter ym., 1964). Uudempi, mittava katsaus ERP-tutkimuksiin löytyy esimerkiksi lähteestä Luck (2014).

Kiinnostavan signaalin löytäminen aivotoiminnasta ei ole yksinkertaista. MEG-mittauksessa koehenkilön pään pinnalle asetetaan satoja sensoreita, joista jokainen mittaa millisekunnin välein magneettikentän voimakkuutta. Aineiston

määrä voi olla valtava jo lyhyessäkin MEG-tutkimuksessa. Kun aivoissa syntyviä magneettikenttiä mitataan pään ulkopuolelta, sensoreihin päätyvät arvot ovat todennäköisesti monen eri aivosignaalin sekoitus. Kuinka ärsykettä vastava aiovaste voidaan erottaa muusta aivotoiminnasta ja miten aivotoiminnan lähde voidaan paikantaa? Tilannetta hankaloittaa se, että kiinnostavat aiovasteet ovat usein muita, joko aivoperäisiä tai aivojen ulkopuolisia, signaaleja useita kertaluokkia heikompia. Esimerkiksi silmänliikkeiden ja räpyttelyn mahdollistavat lihasjännitykset aiheuttavat amplitudiltaan huomattavasti suurempia MEG-signaaleja kuin aiovasteisiin liittyvät signaalit.

Sokean signaalinerottelun (blind source separation, BSS) menetelmät on kehitetty signaalien erottelun työkaluiksi tilanteisiin, joissa tiedossa on vain sekoittuneiden signaalien aineisto. Alkuperäisistä, kiinnostuksen kohteena olevista signaaleista ei tiedetä välttämättä mitään, joskin joitain oletuksia on tehtävä ongelman ratkaisemiseksi. Havaintosignaalit oletetaan lineaarikombinaatioiksi sekoittuneista alkuperäisistä signaaleista tuntemattomilla sekoituskertoimilla. Sokean signaalinerottelun menetelmien avulla voidaan saada selville signaalien väliset sekoitussuhteet, joiden avulla alkuperäiset signaalit voidaan palauttaa. Menetelmiä on käytetty neurotieteen tutkimuksissa laajalti erityyppisissä aseteissa: häiriösignaalien erottamisessa aiovasteista (esimerkiksi Vigário ym., 1998; Fatima ym., 2013), aivojen lepotilatutkimuksissa erilaisten aivorytmien erotteluun (Hyvärinen ym., 2010), reaaliaikaisessa aiovasteiden estimoinnissa (Esposito ym., 2003; Hsu ym., 2016) ja herätevastetutkimuksissa kiinnostavien aivosignaalien löytämiseksi (esimerkiksi Müller ym., 2004; Onton ym., 2006; Tang ym., 2006; Metsomaa ym., 2016). Sokean signaalinerottelun menetelmät ovat MEG-tutkimusten kontekstissa erityisen käyttökelpoisia, koska sekoitussuhteiden avulla voidaan selvittää alkuperäisten aiovasteiden magneettikenttien jakauma jokaiselle löydetylle vasteelle erikseen. Sekoitussuhteiden avulla voidaan päätellä, mistä aivojen osista kukin lähdesignaali on peräisin (Cichocki ja Amari, 2002).

Sokean signaalinerottelun menetelmät ovat aineistolähtöisiä menetelmiä: mallista ei tiedetä muuta kuin havaitut sekoittuneet signaalit. Lähdesignaalit etsitään usein iteratiivisilla algoritmeilla, joiden estimointi on moniulotteisilla aineistoilla työlästä ja hidasta. Lähdesignaalien estimointi moniulotteisella aineistolla voi johtaa heikkojen aiovasteiden hukkumiseen: todellisuudessa fysiologisesti erilaiset, muita estimoituja lähdesignaaleja heikommat signaalit saatetaan virheellisesti estimoida samaksi lähdesignaaliksi, jolloin BSS-ratkaisussa on edelleen sekoittuneita signaaleja eroteltujen lähdesignaalien sijaan. Makeig, Enghoff, Jung, ja Sejnowski (2000) lähestyivät ongelmaa herätevastetyyppisen asetelman kautta: tutkimuksessa jaettiin EEG-aineisto koehenkilölle esitettyjen ärsykkeiden mukaisesti aikaikkunoihin, joista lähdesignaalit estimoitiin. Tällaista ikkunointitapaa on sovellettu myös fMRI-tutkimuksiin (Esposito ym., 2003; Karvanen ja Theis, 2004; Kiviniemi ym., 2011). Näissä tutkimuksissa ikkunointi on tehty liukuvalla aikaikkunalla: koko aineisto on jaettu aikaikkunoihin, jotka menevät osin päällekkäin. Lähdesignaalit on estimoitu näistä ikkunoista. Kaikissa edellämainituissa tutkimuksissa huomattiin, että signaaliaineiston ikkunointi

auttoi löytämään heikot lähdesignaalit, joita koko aineistosta estimoidussa BSS-ratkaisussa ei oltu kyetty erottelemaan.

Tämän tutkielman tavoitteena on selvittää, onko ikkunointimenetelmällä mahdollista löytää esitetyn ärsykkeen synnyttämä aiovaste MEG-aineistosta sokean signaalinerottelun menetelmien avulla. Aikasuunnan sijaan ikkunointia sovelletaan tilasuunnassa: ikkuna muodostuu valitusta määrästä spatiaalisesti toistensa lähellä olevia havaintosignaaleja koko mitatulta ajanjaksolta. Simuloidun asetelman avulla pyritään selvittämään, millainen merkitys ikkunan koon valinnalla ja spatiaalisella ikkunoinnilla on ja kuinka se toimii eri BSS-menetelmiä sovellettaessa. Lopuksi spatiaalista ikkunointia sovelletaan mitattuun MEG-aineistoon. Luvussa 2 esitellään sokean signaalinerottelun malli ja BSS-menetelmiin olennaisesti liittyviä käsitteitä. Luvuissa 3 ja 4 käsitellään kahda eri BSS-mallia, toisen asteen lähde-erottelumallia ja riippumattomien komponenttien analyysiä, joita luvussa 6 sovelletaan. Luvussa 5 esitellään spatiaalisten ikkunoiden muodostaminen yksityiskohtaisemmin. Sekä simulointi että MEG-sovellus toteutetaan R-ohjelmistolla (R Core Team, 2017) käyttäen paketteja `BSSasymp`, `JADE` (Miettinen ym., 2017) ja `fICA` (Miettinen ym., 2015).

2 Sokea signaalinerottele

Sokea signaalinerottele on alunperin signaalinkäsittelyn alalla käytetty termi. Termin alle kuuluvilla menetelmillä pyritään selvittämään, millaisista piilevistä ominaisuuksista havaittu aineisto muodostuu (Cichocki ja Amari, 2002). Piilevien komponenttien tulee olla jonkin ominaisuuden perusteella toisistaan eroavia, jotta niiden erotteleminen on mahdollista. Signaalinkäsittelyn alan kirjallisuudessa ajasta t , $t = 1, \dots, T$, riippuvaa vektoria $\mathbf{y}(t) = [y_1(t) \dots y_p(t)]^\top$ kutsutaan usein havaintosignaaliiksi. Piilevää komponenttia $\mathbf{s}(t) = [s_1(t) \dots s_q(t)]^\top$ kutsutaan lähdesignaaliiksi. Sokean signaalinerottelun ongelma voidaan muotoilla seuraavasti: millaisilla kertoimilla lähdesignaalit $\mathbf{s}(1), \dots, \mathbf{s}(T)$ on sekoitettu, että on saatu aikaan havaitut signaalit $\mathbf{y}(1), \dots, \mathbf{y}(T)$? Matemaattisesti ongelma voidaan esittää signaaleille $\mathbf{y}(t)$ yksinkertaisimmillaan muodossa

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t), \quad (1)$$

missä matriisia $A_{(p \times q)}$ kutsutaan sekoitusmatriisiksi. Havaitut signaalit $\mathbf{y}(t)$ ovat lineaarikombinaatio tuntemattomista sekoitusmatriisin kertoimista ja latenteista signaaleista $\mathbf{s}(t)$. Ongelma voidaan määrittellä yleisemmässä muodossa, jossa malliin lisätään satunnainen, normaalijakautunut kohinakomponentti $\epsilon(t) = [\epsilon_1(t) \dots \epsilon_p(t)]^\top$. Merkitään tällöin havaintosignaalin vektoria $\mathbf{x}(t) = [x_1(t) \dots x_p(t)]^\top$, ja kirjoitetaan

$$\mathbf{x}(t) = \mathbf{y}(t) + \epsilon(t), \quad \epsilon(t) \sim \mathcal{N}_p(\mathbf{0}, \sigma^2 \mathbf{I}_p). \quad (2)$$

Kohinakomponentin oletetaan olevan valkoisen kohinan prosessi, jolloin vektorit $\epsilon(1), \dots, \epsilon(T)$ ovat toisistaan riippumattomia.

Määritelmien (1) ja (2) perusteella sokean signaalinerottelun ongelma on vaivallisesti määritelty: sekä sekoitusmatriisi \mathbf{A} että lähdesignaalit $\mathbf{s}(t)$ ovat tuntemattomia. Ongelman ratkaisemiseksi lähdesignaaleista sekä lähdesignaalin ja kohinakomponenttien suhteesta on tehtävä lisäoletuksia:

- 2.1. Lähdesignaaleille pätee $\mathbf{E}[\mathbf{s}(t)] = \mathbf{0}$ ja $\mathbf{Cov}[\mathbf{s}(t)] = \mathbf{\Sigma}_s = \mathbf{I}_q$
- 2.2. Kohinakomponentit $\epsilon(t)$ ja lähdesignaalit ovat toisistaan riippumattomia.
- 2.3. Sekoitusmatriisi \mathbf{A} on sarakeasteeltaan täysiasteinen.

Oletuksesta (2.3) seuraa, että lähdesignaaleja saa olla korkeintaan yhtä monta kuin havaintosignaaleja, siis $p \geq q$. (Hyvärinen ym., 2001). Näiden lisäksi tarvitaan menetelmäkohtaisia oletuksia, joita esitellään tarkemmin kappaleissa 3 ja 4.

Yllämainitut lisäoletuksetkaan eivät takaa ongelmalle yksikäsitteistä ratkaisua. Määrittelemättömiksi jäävät lähdesignaalien etumerkki, keskinäinen järjestys ja suuruusluokka (Hyvärinen ym., 2001). Signaalien muoto pystytään kuitenkin estimoimaan, mikä on useimpiin sovelluksiin riittävä ratkaisu. Tyydytään toteamaan, että sekoitusmatriisiin \mathbf{A} estimoinnissa hyväksytään myös ratkaisut \mathbf{AP} , missä $\mathbf{P}_{(q \times q)}$ on sellainen matriisi, jossa jokaisella rivillä ja sarakkeella on täsmälleen yksi nollasta eroava alkio.

Seuraavissa alaluvuissa esitellään hyödyllisiä esiaskeleita sokean signaalinerottelun ongelman ratkaisemiseksi. Lisäksi esitellään lähde-erottelumalli, joka hyödyntää havaintosignaalien aikarakennetta ja toisia momentteja ongelman ratkaisussa.

2.1 Valkaisu

Lähdesignaalien oletetaan olevan keskenään korreloimattomia (oletus 2.1). Tätä ominaisuutta voidaan käyttää hyödyksi estimoitaessa lähdesignaaleja sekoittuneista signaaleista. Sekoittuneille signaaleille voidaan tehdä lineaarimuunnos, jonka jälkeen sekoitussignaalit ovat keskenään korreloimattomia ja niiden varianssit skaalattu ykkösiksi. Sekoitussignaalit siis projisoidaan uuteen koordinaatistoon lineaarimuunnoksella. Tätä esiprosessointivaihetta kutsutaan valkaistuksi. BSS-menetelmien ratkaistavaksi jää valkaistun jälkeen enää yksi vaihe: valkaistujen sekoitussignaalien palauttaminen takaisin alkuperäiseen koordinaatistoon.

Keskistetään signaalit

$$\mathbf{x}_\mu(t) = \mathbf{x}(t) - \mu_x, \quad (3)$$

jolloin $\mathbf{E}[\mathbf{x}_\mu(t)] = \mathbf{0}$. Merkitään jatkossa $\mathbf{x}_\mu := \mathbf{x}(t)$ ja oletetaan, että signaalit ovat keskistettyjä yhtälön (3) osoittamalla tavalla.

Valkaisuksi kutsutaan sellaista lineaarimuunnosta

$$\tilde{\mathbf{x}}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{y}(t) + \mathbf{W}\epsilon(t), \quad (4)$$

jossa $\mathbf{W}\mathbf{y}(t) := \tilde{\mathbf{y}}(t)$ ja jolle $\mathbf{Cov}[\tilde{\mathbf{y}}(t)] = \boldsymbol{\Sigma}_{\tilde{\mathbf{y}}} = \mathbf{I}_q$. Matriisia \mathbf{W} kutsutaan valkaismusmatriisiksi ja se halutaan löytää siten, että sekoitussignaalit $\tilde{\mathbf{y}}(t)$ ovat keskenään korreloimattomia. Valkaismusmatriisi ei ole yksikäsitteinen ja se voidaan löytää monella erilaisella menetelmällä (Cichocki ja Amari, 2002).

Johdetaan aluksi valkaismusmatriisi sekoitussignaaleille $\mathbf{y}(t)$ tapauksessa, jossa havaituissa signaaleissa ei ole kohinaa. Lähdesignaalit valkaiseva matriisi voidaan löytää esimerkiksi kovarianssimatriisiin $\mathbf{Cov}[\mathbf{y}(t)] = \boldsymbol{\Sigma}_y$ ominaisarvohajotelman avulla. Kovarianssimatriisi on määritelmänsä mukaisesti symmetrinen ja positiivisesti definiitti. Sen ominaisarvohajotelma on muotoa

$$\boldsymbol{\Sigma}_y = \mathbf{U}_y \boldsymbol{\Lambda}_y \mathbf{U}_y^\top, \quad (5)$$

missä matriisi \mathbf{U}_y on hajontamatriisin Σ_y ominaisvektoreista muodostuva ja siten ortogonaalinen matriisi ja $\Lambda_y = \text{diag}(\lambda_1, \dots, \lambda_p)$, jonka diagonaali-alkiot λ_i ovat matriisin Σ_y ominaisarvot. Valitsemalla valkaisumatriisiksi $\mathbf{W}_y = \Lambda_y^{-\frac{1}{2}} \mathbf{U}_y^\top$ voidaan kirjoittaa

$$\begin{aligned}\Sigma_{\bar{y}} &= \text{Cov}[\mathbf{W}_y \mathbf{y}(t)] \\ &= \mathbf{W}_y \Sigma_y \mathbf{W}_y^\top \\ &= \Lambda_y^{-\frac{1}{2}} \mathbf{U}_y^\top \mathbf{U}_y \Lambda_y \mathbf{U}_y^\top \mathbf{U}_y \Lambda_y^{-\frac{1}{2}} \\ &= \mathbf{I}_q,\end{aligned}\tag{6}$$

joten \mathbf{W}_y on eräs valkaisumatriisi signaaleille $\mathbf{y}(t)$.

Kohinatermin sisältävien signaalien $\mathbf{x}(t)$ tapauksessa kovarianssimatriisi Σ_x on muotoa

$$\Sigma_x = \mathbf{A} \mathbf{A}^\top + \sigma^2 \mathbf{I}_p\tag{7}$$

lähdesignaalien oletusten ja kohinatermien $\epsilon(t)$ normaalisuus- ja riippumattomuusoletusten perusteella (oletus 2.2). Kun kohinatermi lisätään malliin, ominaisarvohajotelma on monimutkaisempi. Kovarianssimatriisin Σ_x ominaisarvohajotelma on kohinamallissa muotoa

$$\begin{aligned}\Sigma_x &= \Sigma_s + \Sigma_\epsilon \\ &= [\mathbf{U}_s \ \mathbf{U}_\epsilon] \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \Lambda_\epsilon \end{bmatrix} \begin{bmatrix} \mathbf{U}_s^\top \\ \mathbf{U}_\epsilon^\top \end{bmatrix} \\ &= \mathbf{U}_s \Lambda_s \mathbf{U}_s^\top + \mathbf{U}_\epsilon \Lambda_\epsilon \mathbf{U}_\epsilon^\top,\end{aligned}\tag{8}$$

missä matriisi \mathbf{U}_s on signaaleihin ja kohinavektoreihin $[\epsilon_1(t), \dots, \epsilon_q(t)]^\top$ liittyvien ominaisvektoreiden muodostama $(p \times q)$ -kokoinen matriisi (Cichocki ja Amari, 2002, s. 140). Matriisi $\Lambda_s = \text{diag}(\lambda_1, \dots, \lambda_q)$ on edellämainittuihin ominaisvektoreihin liittyvä ominaisarvojen matriisi, missä arvot λ_j merkitsevät ominaisvektoreita vastaavia ominaisarvoja. Ominaisarvojen oletetaan olevan järjestyksessä pienimmästä ominaisarvosta suurimpaan. Matriisi \mathbf{U}_ϵ on kohinavektoreihin $[\epsilon_{q+1}(t), \dots, \epsilon_p(t)]^\top$ liittyvien ominaisvektoreiden muodostama $(p \times (p - q))$ -kokoinen matriisi. Kaikkien kohinakomponenttien $\epsilon(t)$ ominaisarvojen muodostama diagonaalimatriisi on

$$\Lambda_\epsilon = \text{diag}(\lambda_{q+1}, \dots, \lambda_p) = \sigma^2 \mathbf{I}_{p-q}.$$

Kohinatermien varianssi σ^2 voidaan siis estimoida kovarianssimatriisin Σ_x pienimpinä ominaisarvoina.

Yhtälön (7) perusteella voidaan kirjoittaa

$$\begin{aligned}\mathbf{A} \mathbf{A}^\top &= \Sigma_x - \sigma^2 \mathbf{I}_p \\ &= \mathbf{U}_s \text{diag}(\lambda_1 - \sigma^2, \dots, \lambda_q - \sigma^2) \mathbf{U}_s^\top.\end{aligned}\tag{9}$$

Merkitsemällä $\mathbf{Cov}[\tilde{\mathbf{x}}(t)] = \Sigma_{\tilde{x}}$ saadaan

$$\Sigma_{\tilde{x}} = \mathbf{Cov}[\mathbf{W}_{\tilde{x}}\mathbf{x}(t)] = \mathbf{W}_{\tilde{x}}\mathbf{A}\mathbf{A}^{\top}\mathbf{W}_{\tilde{x}}^{\top} + \mathbf{W}_{\tilde{x}}\sigma^2\mathbf{I}_p\mathbf{W}_{\tilde{x}}^{\top},$$

josta nähdään, että valitsemalla $\mathbf{W}_{\tilde{x}} = (\Lambda_s - \sigma^2\mathbf{I}_q)^{-\frac{1}{2}}$ sekoitussignaaliolosan varianssi on \mathbf{I}_q .

Lähdesignaalien valkaisu jakaa sokean signaalinerottelun ongelman kahteen osaan. Merkitään valkaisu yhtälön (4) perusteella

$$\mathbf{V} = \mathbf{W}_{\tilde{x}}\mathbf{A}, \quad (10)$$

jolloin ongelma yksinkertaistuu valkaisu matriisiin $\mathbf{W}_{\tilde{x}}$ ja rotaatiomatriisiin \mathbf{V} estimointiin. Tässä tutkielmassa esiteltävät ja sovelluksessa käytettävät menetelmät eroavat toisistaan matriisiin \mathbf{V} estimointimenetelmien suhteen. Seuraavassa kappaleessa esitellään eräs menetelmä, yhteisdiagonalisointi, jonka avulla matriisi \mathbf{V} voidaan estimoida.

Edellä esitelty kovarianssimatriisiin ominaisarvoon perustuva valkaisu menetelmä voidaan korvata millä tahansa menetelmällä, joka löytää sekoitussignaali korreloimattomiksi tekevän lineaarimuunnoksen. Kirjallisuudessa on käytetty esimerkiksi pääkomponenttianalyysiä, faktorianalyysiä ja singulaariarvo hajotelmaa esitetyn ominaisarvo hajotelman sijaan (Hyvärinen ym., 2001; Cichocki ja Amari, 2002).

2.2 Yhteisdiagonalisointi

Yhteisdiagonalisointi (joint diagonalisation, Cardoso ja Souloumiac (1996)) on eräs tapa löytää sokean signaalinerottelun ongelman ratkaiseva rotaatiomatriisi \mathbf{V} . Matriisien joukon yhteisdiagonalisointi tarkoittaa sellaisen rotaatiomatriisin etsimistä, joka diagonalisoi kaikki joukon matriisit. Jotta matriisien täydellinen diagonalisointi olisi mahdollista, matriisijoukon matriisien tulee olla keskenään kommutoivia.

Matriisijoukon valinta riippuu käytetystä sokean signaalinerottelun menetelmästä. Merkitään nyt valittujen matriisien joukkoa

$$\mathcal{M} = \{\mathbf{M}_1, \dots, \mathbf{M}_R\} \quad (11)$$

ja palataan joukon \mathcal{M} valintaan tarkemmin luvuissa 3, 4.3 ja 4.4. Joukon \mathcal{M} matriisien tulee olla symmetrisiä ja niillä tulee olla riippumattomuusominaisuus. Matriisilla $\mathbf{M}_r(\mathbf{x})$ on riippumattomuusominaisuus, jos se on diagonaalinen, kun satunnaismuuttujan \mathbf{x} komponentit ovat riippumattomia. Esimerkiksi kovarianssimatriisilla Σ_x on riippumattomuusominaisuus. (Oja ym., 2006).

Matriisiin \mathbf{V} tulee diagonalisoida kaikki joukon \mathcal{M} matriisit. Tällainen matriisi \mathbf{V} löydetään, kun minimoidaan matriisien $\mathbf{M}_1, \dots, \mathbf{M}_R$ diagonaalien ulkopuolisten arvojen neliösumma

$$\sum_{r=1}^R \|\text{off}(\mathbf{V}\mathbf{M}_r\mathbf{V}^\top)\|^2.$$

Vaihtoehtoisesti kriteerinä voidaan käyttää matriisien $\mathbf{M}_1, \dots, \mathbf{M}_R$ diagonaalialkioiden neliösumman

$$f(\mathbf{V}) = \sum_{r=1}^R \|\text{diag}(\mathbf{V}\mathbf{M}_r\mathbf{V}^\top)\|^2 \quad (12)$$

maksimointia.

Matriisien \mathcal{M} yhteisdiagonalisoijaksi kutsutaan sitä rotaatiomatriisia \mathbf{V} , joka diagonalisoi kaikki matriisijoukon \mathcal{M} matriisit täydellisesti. Jos kaikki matriisit joukossa \mathcal{M} eivät kommutoi keskenään, kyseistä yhteisdiagonalisoijaa ei löydy. Tällöin puhutaan approksimatiivisesta yhteisdiagonalisoinnista.

Diagonalisoiva matriisi \mathbf{V} voidaan löytää esimerkiksi iteratiivisella algoritmilla, joka käyttää hyväkseen Givensin rotaatiomatriisia

$$\mathbf{G}(i, j, \theta) = \begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots & & \vdots & & \vdots \\ \vdots & & 1 & 0 & & 0 & & \\ 0 & \cdots & 0 & \cos(\theta) & 0 & \cdots & 0 & -\sin(\theta) & 0 & \cdots & 0 \\ & & & 0 & 1 & & & 0 & & & \\ \vdots & & & \vdots & & \ddots & & \vdots & & & \vdots \\ & & & 0 & & 1 & 0 & & & & \\ 0 & \cdots & 0 & \sin(\theta) & 0 & \cdots & 0 & \cos(\theta) & 0 & \cdots & 0 \\ & & & 0 & & & & 0 & 1 & & \vdots \\ \vdots & & & \vdots & & & & \vdots & & \ddots & 0 \\ 0 & \cdots & & 0 & \cdots & & 0 & \cdots & 0 & 1 \end{pmatrix},$$

$i, j = 1, \dots, r$, missä indeksit i, j merkitsevät funktioiden $\cos(\theta)$ ja $\sin(\theta)$ rivi- ja sarakepaikkoja matriisissa \mathbf{G} : toisin sanoen funktio $\cos(\theta)$ paikassa (i, i) ja (j, j) , $-\sin(\theta)$ paikassa (i, j) ja $\sin(\theta)$ paikassa (j, i) . Jos kulma $\theta = 0$, Givensin matriisi on identiteettimatriisi \mathbf{I}_r . Diagonalisoiva matriisi \mathbf{V} löydetään, kun matriisi \mathbf{G} on mahdollisimman lähellä identiteettimatriisia, toisin sanoen kun kulma θ on lähellä arvoa 0. (Miettinen ym., 2017).

Kulma θ lasketaan iteratiivisesti kaikille $i < j$ matriisien joukon \mathcal{M} matriisien alkioiden $m_{1,ii}, m_{1,ij}, m_{1,jj}, \dots, m_{r,ii}, m_{r,ij}, m_{r,jj}$ avulla. Menettelytapa on esitetty tarkasti Bunse-Gerstnerin ym. artikkelissa (1993). Diagonalisoiva matriisi \mathbf{V} ja matriisit joukossa \mathcal{M} päivitetään jokaiselle kolmikolle (i, j, θ) seuraavasti:

1. $\mathbf{V} \leftarrow \mathbf{V}\mathbf{G}(i, j, \theta)$
2. $\mathbf{M}_r \leftarrow \mathbf{G}(i, j, \theta)\mathbf{M}_r\mathbf{G}(i, j, \theta)$ kaikille matriiseille joukossa \mathcal{M} .

Kun kulma θ on likimain 0 kaikille $i < j$, voidaan sekoitusmatriisi \mathbf{A} estimoida matriisin \mathbf{V} ja valkaisumatriisin $\mathbf{W}_{\bar{x}}$ avulla.

Esitetty Givensin matriisia hyödyntävä tapa diagonalisoivan matriisin löytämiseksi ei ole ainoa keino estimoida matriisi \mathbf{V} . Edellä esitetty tapa on eräs esimerkki symmetrisestä lähestymistavasta. Symmetristen lähestymistapojen lisäksi on olemassa deflaatiotyyppisiä lähestymistapoja, joissa matriisi \mathbf{V} estimoidaan rivi kerrallaan (Miettinen ym., 2016). On kuitenkin osoitettu, että symmetrinen menetelmä toimii deflaatiopohjaista menetelmää paremmin useimmissa tapauksissa.

3 Toisen asteen lähde-erottelu

Toisen asteen lähde-erottelumallit (second order source separation, SOS) ovat sokean signaalinerottelun malleja, joissa käytetään hyväksi signaalien aikariippuvuutta ja kovarianssimatriiseja. Lähdesignaalien $\mathbf{s}(t)$ ja kohinan $\epsilon(t)$ oletetaan olevan heikosti stationaarisia ajasta riippuvia prosesseja. Heikko stationaarisuus tarkoittaa, että aikasarjaprosessilla on äärellinen varianssi, sen odotusarvo on ajan suhteen vakio ja että prosessin autokovarianssi

$$\gamma(t, t + \tau) = \text{Cov}[s_i(t), s_i(t + \tau)]$$

riippuu ainoastaan aikapisteiden välisestä viiveestä τ . Siten aiemmin tehtyjen lähdesignaalien oletusten (2.1-2.3) lisäksi pätee

$$3.1. \mathbf{Cov}[\mathbf{s}(t+\tau), \mathbf{s}(t)] = \mathbf{Cov}[\mathbf{s}(t), \mathbf{s}(t+\tau)] = \text{diag}(\gamma_1(\tau), \dots, \gamma_m(\tau)) = \mathbf{\Gamma}_\tau$$

kaikilla τ .

$$3.2. \text{Kaikille pareille } i \neq j \text{ on olemassa } \tau > 0 \text{ siten, että } (\mathbf{\Gamma}_\tau)_{ii} \neq (\mathbf{\Gamma}_\tau)_{jj}.$$

SOS-mallissa lähdesignaalien erottaminen perustuu oletukseen niiden erilaisista aikariippuvuusrakenteista. Seuraavassa aluvuossa esitellään eräs SOS-menetelmä, toisen asteen sokean tunnistuksen menetelmä.

3.1 Toisen asteen sokea tunnistus (SOBI)

Toisen asteen sokean tunnistuksen menetelmä (second order blind identification, SOBI) pyrkii erottelemaan lähdesignaalit sekoitussignaalien kovarianssi- ja autokovarianssimatriisien avulla. Havaintosignaalien valkaisuun jälkeen rotaatiomatriisi \mathbf{V} etsitään diagonalisoimalla joukko autokovarianssimatriiseja eri viiveillä τ .

Valkaisu tehdään SOBIssa signaalien $\mathbf{x}(t)$ kovarianssimatriisiin $\mathbf{\Sigma}_x$ avulla. Kovarianssimatriisille lasketaan ominaisarvohajotelma, jonka avulla saadaan laskettua valkaisu-matriisi $\mathbf{W}_{\tilde{x}}$ luvun 2.1 mukaisesti. Sekoitusmatriisi \mathbf{A} löydetään yhtälön (10) mukaisesti rotaatiomatriisin \mathbf{V} avulla. Rotaatiomatriisi saadaan yhteisdiagonalisoimalla valkaistujen havaintosignaalien autokovarianssimatriisien

$$\mathbf{\Sigma}_{\tilde{x}, \tau} = \mathbf{E}[(\tilde{\mathbf{x}}(t + \tau) - \mathbf{E}[\tilde{\mathbf{x}}(t)])(\tilde{\mathbf{x}}(t) - \mathbf{E}[\tilde{\mathbf{x}}(t)])^\top]$$

määräämä joukko $\mathcal{M} = \{\mathbf{\Sigma}_{x, \tau_1}, \dots, \mathbf{\Sigma}_{x, \tau_K}\}$ valituilla viiveillä τ_k .

Sopivien viiveiden, ja siten sopivien diagonalisoitavien autokovarianssimatriisien, valinta ei ole yksiselitteinen. Mitä useampi autokovarianssimatriisi vali-

taan, sitä hitaampaa sekoitusmatriisin estimointi on. Toisaalta tulokset ovat luotettavampia kuin silloin, kun yhteisdiagonalisointi tehdään vain muutamalla autokovarianssimatriisilla. Kirjallisuudessa on esitetty peräkkäisten viiveiden valintaa, useampien viiveiden valintaa eri etäisyyksillä (Tang ym., 2005; Miettinen ym., 2016) ja satunnaistettua viiveiden valintaa (Brewick ja Smyth, 2017). Myös hienostuneempia, sekoitusmatriisin SOBI-estimaattorin asymptoottisiin ominaisuuksiin perustuvia lähestymistapoja on esitetty hiljattain (Taskinen ym., 2016).

4 Riippumattomien komponenttien analyysi

Riippumattomien komponenttien analyysi (independent component analysis, ICA) on lähdesignaalien välisen tilastollisen riippumattomuuden oletukseen nojaava menetelmä sokean signaalinerottelun ongelman ratkaisemiseksi. Edellä esitellystä SOS-mallista poiketen ICA-mallissa ei tehdä oletuksia lähde- tai kohinasignaalien aikarakenteesta. Keskeisenä lähtökohtana on oletus lähdesignaalien välisestä keskinäisestä tilastollisesta riippumattomuudesta. Luvun 2 oletusten lisäksi ICA-mallissa tulee päteä seuraavat oletukset:

4.1. Lähdesignaalit ovat keskenään tilastollisesti riippumattomia.

4.2. Korkeintaan yksi lähdesignaaleista on normaalijakautunut.

Yllämainitut oletukset asettavat perustan riippumattomien komponenttien analyysille: lähdesignaalit voidaan löytää riippumattomuutta tai normaalijakautuneisuutta indikoivien mittojen avulla.

Seuraavissa alaluvuissa määritellään aluksi riippumattomien komponenttien mallille olennaiset termit: keskinäinen tilastollinen riippumattomuus ja satunnaismuuttujan neljännen asteen tunnusluvut. Määritelmien jälkeen esitellään kaksi sekoitusmatriisin estimointitavalta eroavaa menetelmää, kumulanttimatriisien approksimatiivinen yhteisdiagonalisointi ja uudelleenladattu Fast-ICA. Jatkossa signaalien aikarakenne on jätetty merkitsemättä, jolloin $\mathbf{x}(t) := \mathbf{x}$ ja $\mathbf{s}(t) := \mathbf{s}$.

4.1 Keskinäinen tilastollinen riippumattomuus

Satunnaismuuttujat s_1, \dots, s_q ovat toisistaan tilastollisesti riippumattomia, jos niiden yhteisjakauma $p_{s_1, s_2, \dots, s_q}(s_1, s_2, \dots, s_q)$ voidaan ilmaista marginaalijakaumien tulona:

$$p_{s_1, s_2, \dots, s_q}(s_1, s_2, \dots, s_q) = p_{s_1}(s_1) \cdots p_{s_q}(s_q), \quad (13)$$

missä kaikkien satunnaismuuttujien s_i , $i = 1, \dots, q$, muodostamien parien, kolmikkojen, nelikköjen ja niin edelleen tulee olla toisistaan riippumattomia.

Riippumattomuuden mitaksi ei riitä lähdesignaalien välinen korrelaatio, joka mittaa vain lähdesignaalien välistä lineaarista yhteyttä. Keskinäisen riippumattomuuden takaamiseksi lähdesignaalien välillä ei saa olla myöskään epälineaarisia korrelaatioita. Tämän vuoksi lähdesignaalien joukossa saa olla vain yksi normaalijakautunut lähdekomponentti: normaalijakautuneelle satunnaismuuttujalle korreloimattomuudesta seuraa riippumattomuus, jolloin normaali-

jakautuneita lähdesignaaleja ei kyetä erottamaan toisistaan.

4.2 Neljännen asteen tunnusluvut

Riippumattomien komponenttien estimoinnissa hyödynnetään korkeamman asteen tunnuslukuja. Kolmatta ja neljättä keskusmomenttia, vinoutta ja huipukkuutta, voidaan käyttää satunnaismuuttujien jakaumien määrittämiseen ja mitana poikkeamalle normaalijakaumasta. Lisäksi huipukkuuden mittaa tarvitaan määrittettäessä kumulanttimatriiseja (16), joiden avulla sokean signaalinerotteen ongelma voidaan ratkaista.

Huipukkuus määritellään satunnaisvektorin neljäntenä standardoituna keskusmomenttina, josta vähennetään normaalijakauman huipukkuus. Standardoidulle satunnaismuuttujalle \mathbf{x} odotusarvo $\mathbf{E}[\mathbf{x}] = 0$ ja varianssi $\mathbf{E}[\mathbf{x}]^2 = 1$, huipukkuus κ saa muodon

$$\kappa = \frac{\mathbf{E}[(\mathbf{x} - \mathbf{E}[\mathbf{x}])^4]}{(\mathbf{E}[(\mathbf{x} - \mathbf{E}[\mathbf{x}])^2])^2} - 3 = \mathbf{E}[\mathbf{x}^4] - 3. \quad (14)$$

Moniulotteisiin havaintosignaaleihin liittyvä neljännen asteen informaatio voidaan kerätä matriisia laajempaan kokonaisuuteen, kumulanttiiin. Standardoitujen lähdesignaalien \mathbf{x} tapauksessa muuttujan standardoitu neljäs keskusmomentti on itseasiassa myös sen neljäs kumulantti. Kumulantit määritellään kumulantit generoivan funktion avulla seuraavasti:

$$G(k) = \log(\mathbf{E}[e^{\mathbf{k}^\top \mathbf{x}}]),$$

missä $\mathbf{E}[e^{\mathbf{k}^\top \mathbf{x}}]$ on standardoidun satunnaisvektorin \mathbf{x} momentit generoiva funktio. Tällöin neljännen asteen kumulantti $\mathbf{K}_{(p^2 \times p^2)}$ vektorille \mathbf{x} on

$$\begin{aligned} \mathbf{K} &= \mathbf{E}[\mathbf{xx}^\top \otimes \mathbf{xx}^\top] \\ &= \mathbf{E}(\mathbf{x}_i \mathbf{x}_j \mathbf{x}_k \mathbf{x}_l) - \mathbf{E}(\mathbf{x}_i \mathbf{x}_j) \mathbf{E}(\mathbf{x}_k \mathbf{x}_l) - \mathbf{E}(\mathbf{x}_i \mathbf{x}_k) \mathbf{E}(\mathbf{x}_j \mathbf{x}_l) \\ &\quad - \mathbf{E}(\mathbf{x}_i \mathbf{x}_l) \mathbf{E}(\mathbf{x}_j \mathbf{x}_k), \end{aligned} \quad (15)$$

missä $i, j, k, l = 1, \dots, p$ (Hyvärinen ym., 2001). Symbolilla \otimes merkitään tensorituloa (Horn ja Johnson, 1994, luku 4.2). Vektorin \mathbf{x} neljännen asteen kumulanttimatriisi määritellään $(p \times p)$ -matriisille \mathbf{M}_r seuraavasti:

$$\mathbf{C}_x(\mathbf{M}_r) = \mathbf{E}[(\mathbf{x}^\top \mathbf{M}_r \mathbf{x}) \mathbf{xx}^\top] - \mathbf{M}_r - \mathbf{M}_r^\top - \text{tr}(\mathbf{M}_r) \mathbf{I}_n. \quad (16)$$

4.3 Kumulanttimatriisien approksimatiivinen yhteisdiagonalisointi (JADE & k-JADE)

Kumulanttimatriisien approksimatiivinen yhteisdiagonalisointi (joint approximate diagonalisation of eigenmatrices, JADE) ja k-JADE-menetelmät perustuvat neljännen asteen kumulanttimatriiseihin ja niiden yhteisdiagonalisointiin (Cardoso, 1993). k-JADE-menetelmä on JADEsta muunneltu, laskennallisesti nopeampi menetelmä (Miettinen ym., 2013). ICA-mallin oletusten mukaisesti JADE- ja k-JADE-menetelmissä korkeintaan yksi lähdesignaali saa olla normaalijakautunut, joten vain yhden lähdesignaalin huipukkuuden arvo saa olla 0.

Standardoidut lähdesignaalit \mathbf{x} valkaistaan aluksi luvun 2.1 osoittamalla tavalla kovarianssmatriisiin avulla. Valkaistujen lähdesignaalien neljännet kumulantit lasketaan kumulanttimatriisien $\mathbf{C}_{\bar{x}}(\mathbf{E}^{(i,j)})$ avulla, missä $\mathbf{E}^{(i,j)} = \mathbf{e}_i \mathbf{e}_j^\top$ ja \mathbf{e}_i (vastaavasti \mathbf{e}_j) on sellainen vektori, jossa alkio 1 on paikassa i (paikassa j) ja muut alkioit ovat nollia.

Kumulanttimatriisien joukko \mathcal{M} on tällöin yhtälön (16) mukaisesti sellainen matriisien joukko, jossa yksittäinen kumulanttimatriisi on muotoa

$$\mathbf{C}_{\bar{x}}(\mathbf{E}^{(i,j)}) = \mathbf{E}[(\mathbf{x}^\top \mathbf{E}^{(i,j)} \mathbf{x}) \mathbf{x} \mathbf{x}^\top] - \mathbf{E}^{(i,j)} - (\mathbf{E}^{(i,j)})^\top \text{tr}(\mathbf{E}^{(i,j)}) \mathbf{I}_p. \quad (17)$$

Kun matriisien joukko \mathcal{M} yhteisdiagonalisoidaan, löydetään rotaatiomatriisi \mathbf{V} . Maksimoitava diagonalisointikriteeri on tällöin

$$f(\mathbf{V}) = \sum_{i=1}^p \sum_{j=1}^p \|\text{diag}(\mathbf{V} \mathbf{C}(\mathbf{E}^{(i,j)}) \mathbf{V}^\top)\|^2.$$

JADE-menetelmässä yhteisdiagonalisoidaan p^2 kumulanttimatriisia, joten suurilla aineistoilla yhteisdiagonalisointi on työlästä. On osoitettu, että pienempi määrä yhteisdiagonalisoitavia kumulanttimatriiseja riittää joissakin tapauksissa sekoitusmatriisin estimointiin (Miettinen ym., 2013). k-JADE-menetelmässä havaittujen signaalien diagonalisointi tehdään kahdessa vaiheessa. Merkitään $\mathbf{Cov}_4[\tilde{\mathbf{x}}] = \mathbf{E}[(\tilde{\mathbf{x}}^\top \tilde{\mathbf{x}}) \tilde{\mathbf{x}} \tilde{\mathbf{x}}^\top]$ ja maksimoidaan valkaistuille signaaleille kriteeri

$$f(\mathbf{V}^*) = \|\text{diag}(\mathbf{V}^* \mathbf{Cov}_4[\tilde{\mathbf{x}}] \mathbf{V}^{*\top})\|^2,$$

missä diagonalisoiva matriisi \mathbf{V}^* on erään tämän tutkielman ulkopuolelle jäävän BSS-menetelmän FOBI (fourth order blind identification) ratkaisumatriisi (Cardoso, 1989). Signaalit $\tilde{\mathbf{x}}$ valkaistaan uudelleen FOBI-ratkaisulla, merkitään $\tilde{\mathbf{x}}^* = \mathbf{V}^* \tilde{\mathbf{x}}$. Yhteisdiagonalisointi tehdään niille matriiseille $\mathbf{C}_{\tilde{x}^*}(\mathbf{E}^{(i,j)})$, joille $|i - j| < k$. Lukumäärä k vastaa lähdesignaalien s yhtäsuurten huipukkuuden arvojen lukumäärää. Yhteisdiagonalisointikriteeri on tällöin

$$f(\mathbf{V}) = \sum_{|i-j|<k} \|\text{diag}(\mathbf{V} \mathbf{C}_{\tilde{x}^*}(\mathbf{E}^{(i,j)}) \mathbf{V}^\top)\|^2, \quad i, j = 1, \dots, p.$$

Mitä vähemmän lähdesignaaleissa oletetaan olevan huipukkuuksiltaan yhtäsuuria lähteitä, sitä pienempi k voidaan valita ja siten yhteisdiagonalisoitavien matriisien määrä vähenee huomattavasti verrattuna JADE-menetelmään.

4.4 Epänormaalisuusmittojen maksimointi (FastICA)

Suosittu menetelmä riippumattomien komponenttien analyysin ongelmaan on FastICA-algoritmi (Hyvärinen ja Oja, 1997). FastICAssa riippumattomat komponentit löydetään maksimoimalla niiden epänormaalisuus jonkin normaalisuutta mittaavan funktion avulla. Esimerkiksi kappaleessa 4.2 esitetty huipukkuus sopii epänormaalisuuden mitaksi, jolloin huipukkuuden poikkeama nolasta on epänormaalisuuden mitta. Edellä esitellyistä menetelmistä poiketen FastICA ei käytä hyväkseen yhteisdiagonalisointia sekoitusmatriisiin löytämiseksi. Ratkaisumatriisi voidaan etsiä joko rivi kerrallaan, jolloin puhutaan deflaatiopohjaisesta FastICA-menetelmästä, tai rivit voidaan etsiä samanaikaisesti, jolloin puhutaan symmetrisestä FastICAssa.

Deflaatiopohjaisessa FastICA-algoritmissa ratkaisumatriisi etsitään rivi kerrallaan maksimoimalla epänormaalisuusmitta

$$\left| \mathbf{E}[G(\mathbf{v}_k^\top \tilde{\mathbf{x}})] \right|, \quad k = 1, \dots, p, \quad (18)$$

jossa funktio G on tässä tutkielmassa käytettyjen menetelmien mukaisesti $G(\mathbf{v}_k^\top \tilde{\mathbf{x}}) = (\mathbf{v}_k^\top \tilde{\mathbf{x}})^4 - 3$. Vektori \mathbf{v}_k on ratkaisumatriisin \mathbf{V} rivivektori rivillä k . Vektorille \mathbf{v}_k täytyy päteä seuraavat ehdot: $\mathbf{v}_k^\top \mathbf{v}_k = 1$ ja $\mathbf{v}_j^\top \mathbf{v}_k = 0$, kun $j = 1, \dots, k-1$. Epänormaalisuusmitta voidaan tämän rajoituksen avulla maksimoida esimerkiksi gradienttimenelmällä tai Lagrangen menetelmällä.

Lagrangen menetelmää käytettäessä valitaan aluksi ratkaisumatriisille \mathbf{V} alkuarvo \mathbf{V}_0 , jonka rivejä merkitään $\mathbf{v}_{k,0}$. Ratkaisumatriisin \mathbf{V} rivi k lasketaan lausekkeesta

$$\mathbf{v}_k = \mathbf{E}[g(\mathbf{v}_{k,0}^\top \tilde{\mathbf{x}}) \tilde{\mathbf{x}}] - \mathbf{E}[g'(\mathbf{v}_{k,0}^\top \tilde{\mathbf{x}})] \mathbf{v}_{k,0}, \quad (19)$$

missä $g(\mathbf{v}_{k,0}^\top \tilde{\mathbf{x}})$ on funktion $G(\mathbf{v}_{k,0}^\top \tilde{\mathbf{x}})$ ensimmäinen ja $g'(\mathbf{v}_{k,0}^\top \tilde{\mathbf{x}})$ toinen derivaatta. Gram-Schmidt-ortogonalisointimenetelmällä voidaan skaalata vektorit ortonormaaleiksi. Tehdään aluksi vektorista \mathbf{v}_k ortogonaalinen muihin vektoreihin \mathbf{v}_j nähden:

$$\mathbf{v}_k := \mathbf{v}_k - \sum_{j=1}^{k-1} (\mathbf{v}_k^\top \mathbf{v}_j) \mathbf{v}_j$$

ja normalisoidaan vektori jakamalla se pituudellaan:

$$\mathbf{v}_k := \frac{\mathbf{v}_k}{\|\mathbf{v}_k\|}.$$

Asetetaan \mathbf{v}_k uudeksi alkuarvoksi: $\mathbf{v}_{k,0} \leftarrow \mathbf{v}_k$. Proseduuri toistetaan yhtälöstä (19) alkaen, kunnes on löydetty sellainen \mathbf{v}_k , jolle iteraatiokierroksen estimaatti on lähempänä kyseisen iteraatiokierroksen alkuarvoa kuin ennalta asetettu iteraatiotarkkuus.

Alkuarvomatriisin \mathbf{V}_0 valinta vaikuttaa FastICA-menetelmän lähdesignaalien erottelukykyyyn. Uudelleenladattu FastICA on Nordhausen ym. (2011) kehittämä menetelmä, jossa lähdekomponentit erotellaan optimaalisessa järjestyksessä täsmällisen alkuarvomatriisin valinnan avulla. Apuna käytetään jollakin toisella BSS-menetelmällä (esimerkiksi k-JADE, kappale 4.3) laskettuja lähdesignaaleja, joiden optimaalinen järjestys voidaan estimoida. Optimaaliseen järjestykseen permutoituja lähdesignaaleja käytetään deflaatiopohjaisessa Fast-ICAssa valkaistujen signaalien sijaan ja ratkaisumatriisin alkuarvoksi asetetaan identiteettimatriisi. Lähdesignaalien optimaalista järjestystä kuvaavan tunnusluvun tarkka esitys ja teoria löytyvät artikkelista Nordhausen ym. (2011).

5 Spatiaalinen ikkunointi sokeassa signaalinerottelussa

Havaintoaineiston koko voi olla rajoittava tekijä sovellettaessa aiemmin esiteltyjä sokean signaalinerottelun menetelmiä käytäntöön. Esimerkiksi JADE-menetelmä on laskennallisesti hyvin raskas, kun havaintosignaaleja on paljon. Tällöin joudutaan yhteisdiagonalisoimaan suuri määrä neljännen asteen kumulanttimatriiseja. Karvanen ja Theis (2004) ja myöhemmin Kiviniemi ym. (2011) esittivät erityisesti toiminnallisen magneettikuvauksen tuottamaan aineistoon sopivan, aikasuunnassa aineistoa pienempiin paloihin jakavan ikkunointimenetelmän, jonka avulla lähdesignaalien laskemista voidaan helpottaa. Samankaltaista lähestymistapaa on sovellettu myös reaaliaikaiseen aivosignaalien erotte luun fMRI-tutkimuksissa (Esposito ym., 2003). Laskemisen nopeutumisen lisäksi lähdesignaalien estimointi on tarkempaa, jos lähdesignaaleja voidaan estimoida vähän kerrallaan (Karvanen ja Koivunen, 2002; Kiviniemi ym., 2011).

Aikasuunnan lisäksi havaittujen signaalien moniulotteisuutta voidaan pienentää spatiaalisessa suunnassa: tällöin vain osaa havaituista signaaleista analysoidaan kerrallaan signaalien havaintopaikan perusteella. Havaittujen signaalien aineisto jaetaan pienempiin osiin käyttämällä ikkunointikriteerinä havaintopisteiden koordinaattien välisiä etäisyyksiä. Tällainen lähestymistapa on hyödyllinen, jos lähdesignaalien voidaan olettaa sekoittuneen paikallisesti.

Pisteiden $p_k = (p_{k,x}, p_{k,y}, p_{k,z})$ ja $p_i = (p_{i,x}, p_{i,y}, p_{i,z})$ välinen euklidinen etäisyys määritellään kolmiulotteisessa karteesisessa koordinaatistossa

$$d(p_k, p_i) = \sqrt{(p_{k,x} - p_{i,x})^2 + (p_{k,y} - p_{i,y})^2 + (p_{k,z} - p_{i,z})^2}. \quad (20)$$

Ikkuna \mathcal{H}_k on h kappaletta havaintosignaaleja sisältävä joukko, jossa signaalien $x_i(t)$ mittauspisteet p_i ovat mitan $d(p_k, p_i)$ perusteella lähinnä signaalin $x_k(t)$ mittauspistettä p_k . Toisin sanoen

$$\mathcal{H}_k = \{x_i(t) : \arg \min_i d(p_k, p_i), \#x_i(t) = h\}, \quad (21)$$

missä merkintä $\#x_i(t)$, $i = 1, \dots, N$ tarkoittaa tässä havaintosignaalien lukumäärää. Ikkunan koko h valitaan sovelluskohteen kannalta järkevällä tavalla: esimerkiksi myöhemmin tässä tutkielmassa ikkunan koko valitaan MEG-sensorien sijaintien perusteella (luku 6).

Ikkunointimenetelmällä voidaan valita havaintosignaalien joukosta paikallisesti yksi tai useampi kiinnostava havaintosignaalien ikkuna tai ikkunoida koko havaintosignaalien aineisto. Jälkimmäisessä tapauksessa eri ikkunoista estimoidut lähdesignaalit ovat osittain estimoitu samoista havaintosignaaleista. Samaa lähdesignaalia vastaavat estimoidut signaalit voidaan löytää laskemalla estimoitujen signaalien väliset korrelaatiokertoimet eri ikkunoiden välillä.

Aivokuvantamismenetelmien tapauksessa usein kiinnostuksen kohteena on löytää esitetyn ärsykkeen aikaansaama signaali. Koeasetelman perusteella tie-

detään, milloin kyseinen ärsyke on esitetty. Tämän tiedon perusteella voidaan määritellä funktio, johon estimoituja lähdesignaaleja voidaan verrata. Referenssifunktio voi olla tällöin esimerkiksi muotoa

$$r(t) = \begin{cases} 1 & \text{ajanhetkillä } t, \text{ joilla ärsykettä on esitetty} \\ 0 & \text{muulloin,} \end{cases} \quad (22)$$

jolloin ärsykettä vastaavan lähdesignaalin tulisi olla se estimoitu signaali, joka korreloi itseisarvoisesti vahvimmin referenssifunktion kanssa.

6 Aivojen sähköisen toiminnan mittaaminen ja aineiston ikkunointi

Aivojen sähköisen toiminnan synnyttämiä heikkoja magneettikenttiä voidaan mitata magnetoenkefalografilla. MEG-laitteistoon kuuluu MEG-kypärä, jonka kautta varsinainen magneettikenttien mittausta tapahtuu pään pinnalta suprajohdettavilla SQUID-antureilla (superconducting quantum interference devices, kuva 1). Uusimmissa laitteissa SQUID-kypärässä on 102 mittaavaa anturia. Magneettikenttien voimakkuus viittaa aivoalueen aktivaation tasoon kyseisellä aivokuoren alueella.

Aivotoiminnan aiheuttamien magneettikenttien voimakkuudet ja niiden muutokset mitataan SQUID-antureissa olevien sensorien, magnetometrien ja gradiometrien, avulla. Yksi SQUID-anturi sisältää yhden magnetometrin ja kaksi gradiometriä. Magnetometrit mittaavat magneettikentän voimakkuutta pään pintaan nähden kohtisuoraan. Gradiometrit mittaavat magneettikentän muutoksen voimakkuutta pään pinnan suunnassa: kaksi gradiometriä asetetaan kohtisuoraan toisiaan vastaan, jolloin niiden mittaustuloksesta lasketaan magneettikentän gradientti. Gradiometrit poistavat magnetometriä tehokkaammin häiriösignaaleja, mutta näin ollen havaitsevat vain lokaalit aivosignaalit. Magnetometrejä käytettäessä päästään käsiksi myös aivojen syvemmistä osista tuleviin aivosignaaleihin (Clarke ja Braginski, 2006).



Kuva 1: MEG-kypärä ja 102 SQUID-anturia. (Lähde: <http://www.supraconductivite.fr/en/index.php?p=applications-medical-meg>)

MEG-mittausten heikkoutena voidaan pitää mittauksiin liittyvää epävarmuutta aivotoiminnan syntypaikasta. Pään pinnalta mitattujen sekoittuneiden havaintosignaalien, eli aivoaktivaation eri puolilla päätä synnyttämien magneettikenttien voimakkuuksien, perusteella halutaan tehdä päätelmiä siitä, mistä lähdesignaalit eli alkuperäiset magneettikentät ovat peräisin. Sokean signaalinerottelun menetelmät tarjoavat mahdollisia ratkaisutapoja ongelmaan: BSS-

menetelmien avulla voidaan erotella eri lähteistä peräisin olevat magneettikentät toisistaan. Tällöin tutkittavaan ärsykkeeseen liittyvien aivovasteiden oletetaan olevan ominaisuuksiltaan erilaista verrattuna mittauksen kannalta häiriöllisiin signaaleihin (Vigario ja Oja, 2008).

Edellä kuvatun MEG-laitteiston tuottamassa aineistossa on yhteensä 306 aivojen magneettikenttien voimakkuutta mittaavaa sensoria. MEG on ajallisesti tarkka aivokuvantamismenetelmä: sensoreiden lähettämää informaatiota voidaan mitata jopa 1000 kertaa sekunnissa. Näin ollen lyhytkin mittausaika tuottaa paljon dataa.

Seuraavassa alaluvussa esitellään simulointiasetelma ikkunointimenetelmän toimivuuden selvittämiseksi. Tämän jälkeen ikkunointia sovelletaan yhden koehenkilön MEG-aineistoon. Molemmissa tapauksissa kiinnostava lähdesignaali pyritään löytämään sellaisen funktion avulla, joka kuvaa simulointitapauksessa kiinnostavan aivoaktivaation signaalin muotoa ja sovellustapauksessa sitä, milloin kiinnostavassa signaalissa tapahtuu muutoksia. Tätä funktiota kutsutaan molemmissa tapauksissa referenssifunktioksi.

6.1 Simulointi

Ikkunoinnin toimivuutta selvitettiin aluksi R-ohjelmalla (R Core Team, 2017) toteutettujen simulointikokeiden avulla. Simulointiasetelmalla haluttiin selvittää, auttaako ikkunointimenetelmän käyttö lähdesignaalien erottelussa MEG-mittausdatan tyyppiselle simuloitulle aineistolle. Lisäksi haluttiin selvittää, olisiko mahdollista löytää tietty, kiinnostava komponentti sekoitussignaaleista ja vaikuttaako käytetyn ikkunan koko komponentin löytymiseen. Ikkunamenetelmää verrattiin myös pääkomponenttianalyysiin, jota käytetään toisinaan esiprosessointivaiheena havaintoaineiston moniulotteisuuden rajoittamiseksi ennen sokean signaalin erottelun menetelmien soveltamista (esimerkiksi Hyvärinen ym., 2001).

Vertailtaviksi sokean signaalin erottelun menetelmiksi valittiin toisen asteen lähde-erottelumalleista SOBI, jolle asetettiin viiveet $\tau = 2, 4, \dots, 10, 15, 20, \dots, 50, 60, 70, \dots, 100$ ja ICA-menetelmistä JADE, 1-JADE ja uudelleenladattu FastICA. 1-JADElle valittiin toiseksi valkaisumatriisiksi FOBI-menetelmällä estimoitu ratkaisumatriisi. Uudelleenladatulle FastICAlle, josta jatkossa käytetään lyhennettä fICA, sekoitusmatriisin alkuarvoksi valittiin 1-JADE-ratkaisu ja epänormaalisuusmitaksi $g(x) = x^3$.

Simulointiasetelma pyrittiin tekemään magnetoencefalografian tyyppisen käytännön sovelluksen kanssa mahdollisimman yhtäläiseksi ja malleista ja havaintosignaaleista tehtyjen oletusten mukaisiksi (oletukset 2.1.-2.3., 3.1. ja 4.1.-4.2.). Aineisto pyrittiin simuloimaan siten, ettei se suosisi - tai vaihtoehtoisesti haittaisi - minkään valitun BSS-mallin käyttöä, mutta että aineisto olisi MEG-aineiston tapaan aikasarja. Signaalien mittausgeometria oletettiin aiemmin esitellynlaiseksi SQUID-kypäräksi (kuva 1), jossa on 102 signaaleja mittaavaa magnetometriä. Kiinnostavaksi lähdesignaaliiksi valittiin signaali $\mathbf{s}_1(t)$, joka pidettiin

samana kaikilla simulointikiirroksilla.

Lähdesignaaleiksi simuloitiin mielenkiintoista vastetta kuvaavan referenssifunktion

$$r(t) = \mathbf{s}_1(t) = \begin{cases} 1 & \text{kun } 130 \leq t \leq 200 \\ 0 & \text{muulloin} \end{cases}, \quad \text{missä } t = 1, \dots, T \quad (23)$$

lisäksi taulukon 1 mukaisia aikasarjaprosesseja riippumattomilla $\text{Exp}(1)$ -jakautuneilla innovaatioilla. Aivotoimintoja kuvaavia lähdesignaaleja simuloitiin yhteensä 20 kappaletta ja aikapisteiden lukumääräksi valittiin $T = 1000$.

Taulukko 1: Simuloitujen lähdesignaalien MA-, AR- ja ARMA-prosessien ϕ - ja θ -kertoimet. Kaikille simuloituille lähdesignaaleille innovaatiot olivat riippumattomia, eksponenttijakautuneita muutujia parametrillä $\lambda = 1$.

$s_i(t)$	AR-kertoimet			MA-kertoimet		
	ϕ_1	ϕ_2	ϕ_3	θ_1	θ_2	θ_3
2	0.8	-	-	-	-	-
3	0	-0.3	-	-	-	-
4	-	-	-	0.2	-	-
5	-	-	-	-0.1	-0.2	-
6	0.1	-0.2	0.3	-	-	-
7	-	-	-	-0.1	0.2	-0.3
8	-0.1	-0.2	-	0.1	0.2	-
9	0.7	-0.4	-	-0.7	0.4	-
10	-0.5	0.4	-	0.7	-0.4	-
11	0.2	-0.2	-	-	-	-
12	-	-	-	-0.2	0.2	-
13	0	-0.4	-	0	0.4	-
14	-0.2	0	0.5	-	-	-
15	-	-	-	-0.5	0.2	0.1
16	0	-0.2	-	-0.7	0	0.2
17	0.1	-0.2	-	0.5	0.6	-
18	0.7	-	-	0	0.1	-
19	0.2	-0.4	-	-0.5	-	-
20	0.7	-	-	-0.7	0.2	-

Sekoitusmatriisin \mathbf{A} alkiot arvottiin tasajakaumasta väliltä $[-1, 1]$ referenssifunktiota vastaavaa saraketta lukuunottamatta. Referenssifunktion sarakkeessa ne rivit arvottiin tasajakaumasta, jotka vastasivat simuloituja lähdesignaaleja, mutta muut sarakkeen alkiot asetettiin nolaksi.

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,20} \\ \vdots & \vdots & & \vdots \\ a_{20,1} & a_{20,2} & \cdots & a_{20,20} \\ 0 & a_{21,2} & \cdots & a_{21,20} \\ \vdots & \vdots & & \vdots \\ 0 & a_{102,2} & \cdots & a_{102,20} \end{pmatrix}, \quad a_{i,j} \sim \text{Tas}(-1, 1). \quad (24)$$

Sekoitusmatriisi pidettiin samana kaikille simulointikiirroksille ($n_{sim} = 1000$) ja -asetelmille.

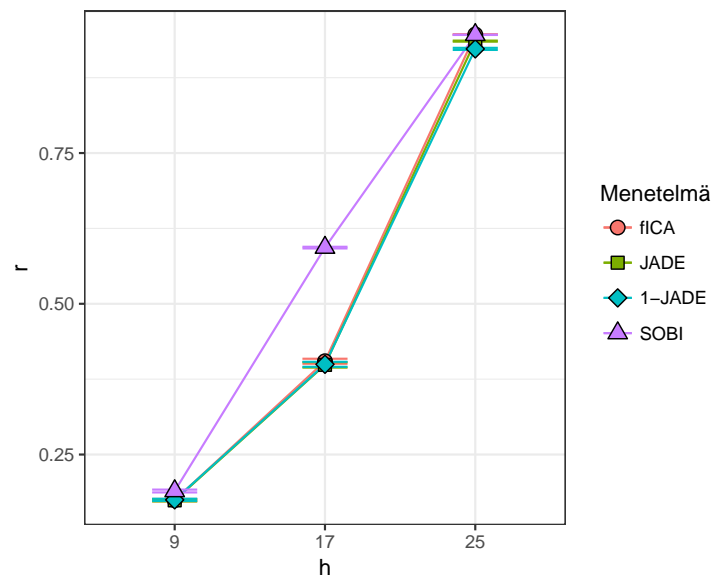
Havaintosignaalit laskettiin simuloituista lähdesignaaleista ja sekoitusmatriisista mallin (1) osoittamalla tavalla, jonka jälkeen niihin lisättiin riippumattomia, normaalijakautunutta kohinaa mallin (2) mukaisesti: $\epsilon(t) \sim \mathcal{N}_{102}(\mathbf{0}, \sigma^2 \mathbf{I}_{102})$. Varianssin σ^2 arvoiksi valittiin 0.1 ja 1. Lähdesignaalit laskettiin asetelmille $h = 9$, $h = 17$ ja $h = 25$ sekä samankokoisille pääkomponenttimatriiseille. Ikkunoiden koot valittiin sensorigeometriaan sopiviksi (kuva 1). Suurimmalla ikkunakoolla, $h = 25$, ikkunan koko on suurempi kuin simulointiasetelman lähdesignaalien lukumäärä.

6.1.1 Tulokset

Simulointiasetelmien vertailusuurena käytettiin lähdesignaalien ja referenssifunktion välistä itseisarvoista korrelaatiokerrointa. Simulointiaineisto jaettiin kullakin simulointikiirroksella ikkunoihin, joihin sovellettiin aiemmin mainittuja BSS-menetelmiä. Menetelmien estimoimien lähdesignaalien ja referenssifunktion väliset korrelaation itseisarvot laskettiin kullekin ikkunoidulle simulointiaineistolle. Vertailusuureksi valittiin korrelaation itseisarvon maksimiarvo.

Kuvassa 2 on esitetty vertailusuureen keskiarvo BSS-menetelmille ikkunoinnain 95 %:n luottamusväleinen, kun kohinavarianssiksi asetettiin $\sigma^2 = 0.1$. Referenssifunktiota vastaavan lähdesignaalin korrelaatio on jokaiselle käytetylle menetelmälle sitä suurempi, mitä suurempi ikkunan koko on. ICA-menetelmät toimivat hyvin samankaltaisesti keskenään: erot ikkunakoiden $h = 9$ ja $h = 17$ tulosten välillä ovat pienemmät kuin ikkunoiden $h = 17$ ja $h = 25$ välillä. Kahden pienemmän ikkunakoon tapauksessa ICA-menetelmien simulointikierron itseisen korrelaation keskiarvo on sama, mutta suurimman ikkunakoon tapauksessa keskiarvoissa on jo pieniä eroja: fICA näyttäisi toimivan ICA-menetelmistä parhaiten suurimmassa ikkunassa ja 1-JADE muita huonommin (taulukko 2). SOBI toimii ICA-menetelmiä paremmin kaikissa ikkunakoissa ja erot ICA-menetelmiin kasvavat sitä mukaa, kun ikkunakoko kasvaa. SOBI- ja ICA-menetelmien erot ovat erityisen suuret ikkunakoissa $h = 17$, jossa myös ICA-menetelmien tulosten vaihteluväli on suuri.

Kun kohinavarianssiksi asetettiin $\sigma^2 = 1$, BSS-menetelmien lähdesignaalien erottelukyky kärsi huomattavasti verrattuna pienemmän kohinavarianssin tapaukseen (taulukko 3). ICA-menetelmät toimivat edelleen keskenään hyvin



Kuva 2: Eri BSS-menetelmien referenssifunktion erottelukyky ikkunoittain mitattuna referenssifunktion ja parhaiten sitä vastaavan estimoidun lähdesignaalin korrelaation itseisarvona. Kuvassa tuhanen simulointikierroksen korrelaatioiden keskiarvot ja niiden 95 %:n luottamusvälit, kun $\sigma^2 = 0.1$.

Taulukko 2: Referenssifunktion kanssa itseisesti parhaiten korreloivien estimoitujen lähdesignaalien minimi, maksimit ja kaikkien simulointikierrosten keskiarvot \bar{r} , kun $\sigma^2 = 0.1$.

Menetelmä	$h = 9$			$h = 17$			$h = 25$		
	\bar{r}	min	max	\bar{r}	min	max	\bar{r}	min	max
fICA	0.17	0.11	0.34	0.40	0.28	0.61	0.89	0.89	0.96
JADE	0.17	0.11	0.31	0.40	0.24	0.62	0.85	0.85	0.95
1-JADE	0.17	0.11	0.33	0.40	0.27	0.59	0.78	0.78	0.95
SOBI	0.19	0.10	0.37	0.59	0.46	0.66	0.93	0.93	0.96

samankaltaisesti, ja SOBI löytää referenssisignaalin hieman ICA-menetelmiä tarkemmin. Ikkunan koko vaikuttaa huomattavasti vähemmän kaikkien BSS-menetelmien tuloksiin: erityisesti ICA-menetelmille ikkunan koolla ei juurikaan näytä olevan merkitystä kiinnostavan signaalin löytymisen suhteen. SOBille ikkunan koon vaikutus näkyy ICA-menetelmiä selvemmin (kuva 3).

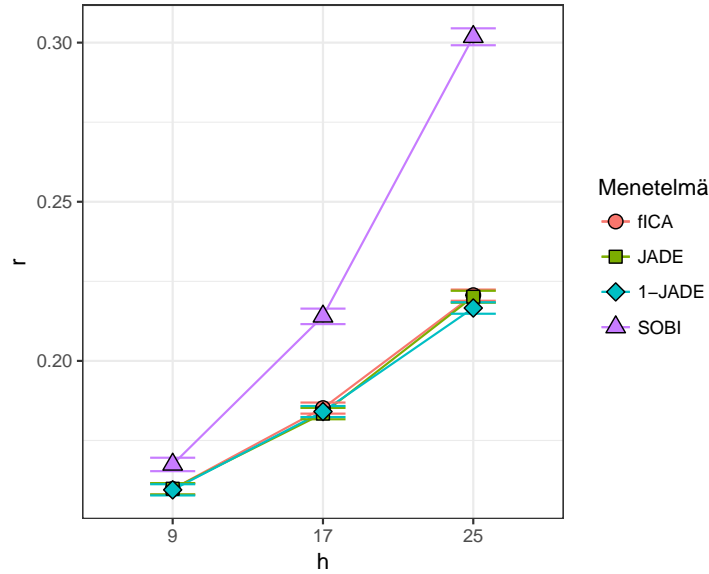
Ikkunointimenetelmää verrattiin pääkomponenttianalyysillä toteutettuun esiprosessointiin, jossa simulointiaineistojen ulottuvuutta pienennettiin ikkunoiden kokoja vastaavaan määrään pääkomponentteja. Yhdeksän ensimmäistä pääkomponenttia selittivät noin 70 % aineiston vaihtelusta ja 17 ensimmäistä pääkomponenttia yli 90 %. Kohinavarianssin $\sigma^2 = 0.1$ tapauksessa ikkunointiratkaisut toimivat pienimpien ikkunoiden osalta paremmin kuin pääkomponenttiratkaisut: erityisesti ikkunakoossa $h = 17$ ikkunoinnin ja pääkomponenttiratkaisun ero vertailusuureessa on suuri. Suurimmalla ikkunakoolla pääkomponenttianalyysiratkaisu on ikkunarakaisua parempi (kuva 4).

Kohinavarianssin $\sigma^2 = 1$ tapauksessa ikkunointiratkaisu toimii jokaisessa ikkunakoossa ja jokaisella pääkomponenttien lukumäärällä paremmin kuin pääkomponenttiratkaisu. Lisäksi pääkomponenttiratkaisuissa simulointikierrosten väliset vaihtelut ovat keskimäärin suurempia ja simuloinnin keskiarvon luottamusväli on keskimäärin leveämpi kuin ikkunointiratkaisujen (kuva 5). Pääkomponenttianalyysiratkaisun ja ikkunointiratkaisun väliset erot ovat pienimmät ikkunakoossa ja pääkomponenttien määrällä $h = 25$.

Vertailun vuoksi laskettiin myös koko simulointiaineiston BSS-ratkaisu il-

Taulukko 3: Referenssifunktion kanssa itseisesti parhaiten korreloivien estimoitujen lähdesignaalien minimi, maksimit ja kaikkien simulointikierrosten keskiarvot \bar{r} , kun $\sigma^2 = 1$.

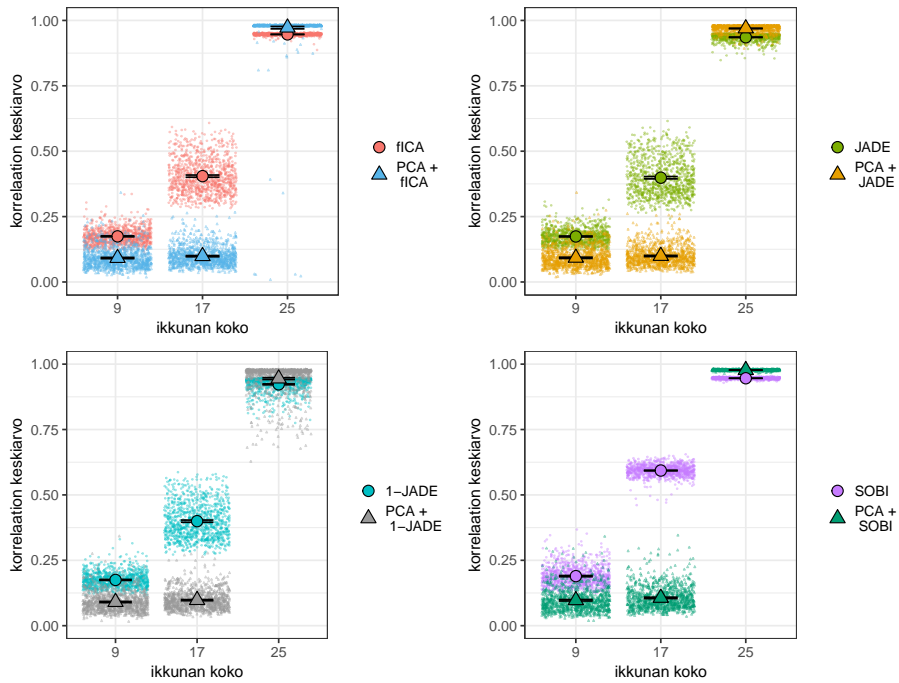
Menetelmä	$h = 9$			$h = 17$			$h = 25$		
	\bar{r}	min	max	\bar{r}	min	max	\bar{r}	min	max
fICA	0.16	0.09	0.32	0.19	0.13	0.35	0.22	0.15	0.35
JADE	0.16	0.09	0.30	0.18	0.12	0.36	0.22	0.15	0.37
1-JADE	0.16	0.10	0.31	0.18	0.12	0.35	0.22	0.14	0.37
SOBI	0.17	0.10	0.33	0.21	0.12	0.37	0.30	0.17	0.42



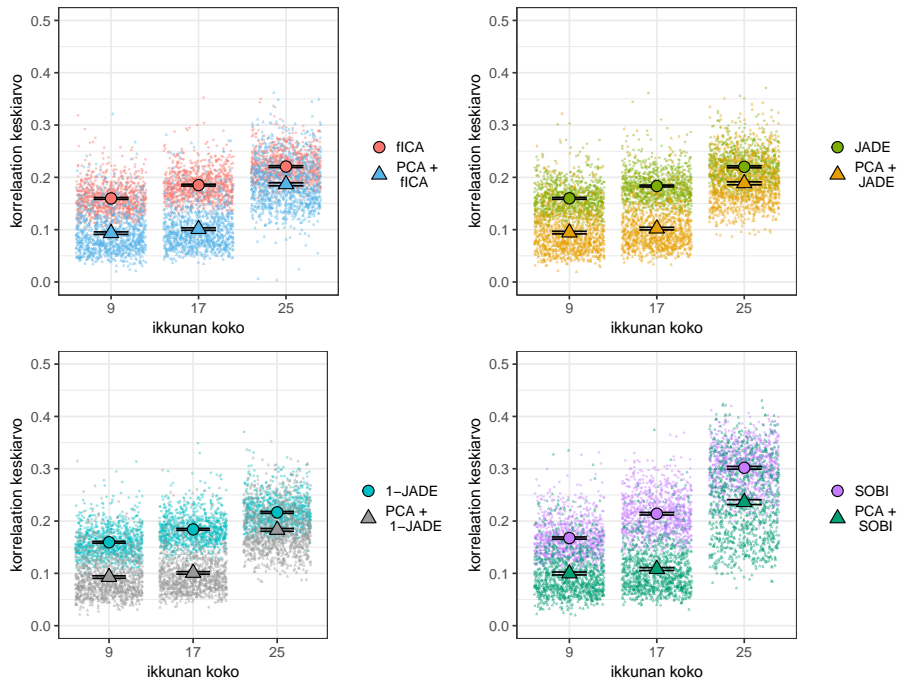
Kuva 3: Eri BSS-menetelmien referenssifunktion erottelukyky ikkunoittain mitattuna referenssifunktion ja parhaiten sitä vastaavan estimoidun lähdesignaalin korrelaation itseisarvona. Kuvassa tuhat simuloitakerroksen vertailusuureiden keskiarvot ja niiden 95 %:n luottamusvälit, kun $\sigma^2 = 1$.

man ikkunointia muutamilla nopeimmilla ja kiinnostavimmilla BSS-menetelmillä (taulukko 4). Kohinavarianssin arvolla $\sigma^2 = 0.1$ koko simulointiaineiston SOBI-ratkaisun vertailusuureen arvo ($\bar{r} = 0.97$) on hiukan parempi kuin ikkunan $h = 25$ SOBI-ratkaisu ($\bar{r} = 0.93$), mutta ero on hyvin pieni. Kohinavarianssin arvolla $\sigma^2 = 1$ koko simulointiaineiston SOBI-ratkaisu ($\bar{r} = 0.42$) on melko paljon parempi kuin paras ikkunointiratkaisu ($\bar{r} = 0.30$). 1-JADElle sen sijaan koko simulointiaineiston ratkaisu on selvästi huonompi kuin ikkunointimenetelmän ratkaisut: vertailusuureen arvo on koko simulointiaineiston ratkaisulle kohinavarianssin $\sigma^2 = 0.1$ tapauksessa $\bar{r} = 0.42$, kun ikkunaratkaisu on suurimmalle ikkunalle $\bar{r} = 0.78$. Kohinavarianssin $\sigma^2 = 1$ tapauksessa vertailusuureen arvo koko simulointiaineiston ratkaisulle on $\bar{r} = 0.16$, mikä vastaa 1-JADEn simulointikeskiarvoa ikkunakoolla $h = 9$. BSS-ratkaisun laskeminen koko aineistolle on huomattavasti työläämpää kuin ikkuna-BSS-ratkaisun laskeminen.

Simulointikokeiden perusteella voidaan todeta, että pienen kohinavarianssin tapauksessa ikkunoinnista ei ole juurikaan hyötyä: etsitty lähdefunktio löytyy sitä paremmin, mitä suurempi ikkuna on kyseessä, ja erot referenssifunktion ja löydetyn lähdefunktion välisten korrelaatioiden keskiarvoissa ovat suuret ikkunoiden välillä. Koko aineiston tuloksiin verrattuna ikkunointi toimi huonommin pienemmissä ikkunoissa ($h = 9$ ja $h = 17$), mutta ikkunakoolla $h = 25$ joko



Kuva 4: Eri BSS-menetelmien referenssifunktion erottelukyky ikkunoinnain verrattuna pääkomponenttianalyysiin ikkunoiden kokoa vastaavilla komponenttien lukumäärillä, kun $\sigma^2 = 0.1$. Vertailusuureen simulointikierrosten keskiarvojen lisäksi kuvaan on piirretty keskiarvon luottamusvälit (mustat viivat) ja kunkin menetelmän simulointikierrosten hajontakuviot ikkunointi- ja pääkomponenttiratkaisuille (pienet pisteet). Vertailusuureen simulointikierröskohtaiset arvot on hajautettu keinotekoisesti x-akselin suunnassa, jotta arvot erottuvat toisistaan paremmin.



Kuva 5: Eri BSS-menetelmien referenssifunktion erottelukyky ikkunoinnain verrattuna pääkomponenttianalyysiin ikkunoiden kokoa vastaavilla komponenttien lukumäärillä, kun $\sigma^2 = 1$. Vertailusuureen simulointikierrosten keskiarvojen lisäksi kuvaan on piirretty keskiarvon luottamusvälit (mustat viivat) ja kunkin menetelmän simulointikierrosten hajontakuviot ikkunointi- ja pääkomponenttiratkaisuille (pienet pisteet). Vertailusuureen simulointikierriskohtaiset arvot on hajautettu keinotekoisesti x-akselin suunnassa, jotta arvot erottuvat toisistaan paremmin.

Taulukko 4: Referenssifunktion kanssa itseisesti parhaiten korreloivien estimoitujen lähdesignaalien vaihteluvälit ja simulointikierrosten keskiarvot \bar{r} kaikilla 102:lla sensorilla lasketuille simulointiaineistoille.

Menetelmä	σ^2	\bar{r}	min	max
1-JADE	0.1	0.42	0.28	0.92
SOBI	0.1	0.97	0.95	0.98
1-JADE	1	0.16	0.11	0.27
SOBI	1	0.42	0.25	0.58

lähes yhtä hyvin (SOBI) tai hieman paremmin (1-JADE).

Suuremman kohinavarianssin tapauksessa ikkunointi tuottaa samankaltaisia tuloksia jokaiselle ikkunakoolle, mutta referenssifunktio löytyy silti parhaiten suurimmasta ikkunasta. Koko aineistosta laskettuihin tuloksiin verrattuna 1-JADE toimii paremmin ikkunoinnin kanssa, mutta SOBille ikkunointi antaa huonomman tuloksen kuin koko aineiston tulos.

Pääkomponenttianalyysi esiprosessointivaiheena on pienellä komponenttien määrällä tehottomampi kuin ikkunointimenetelmä. Se toimii kuitenkin hyvin tapauksessa, jossa lähdesignaalien lukumäärä on pienempi kuin pääkomponenttien lukumäärä (25 pääkomponenttia): pienemmän kohinavarianssin tapauksessa jopa paremmin kuin ikkunointi. Suuremman kohinavarianssin tapauksessa ikkunointi on kaikissa ikkunakoissa pääkomponenttiratkaisua parempi vaihtoehto.

6.2 Sovellus

Tutkielmassa käytettävä MEG-aineisto on Jarmo Hämäläisen (Jyväskylän yliopisto, Psykologian laitos), Minna Torpan (Jyväskylän yliopisto, Kasvatustieteen laitos) ja Tiina Parviaisen (Jyväskylän yliopisto, Monitieteinen aivotutkimuskeskus) tutkimusaineistoa projektista, jossa tutkittiin dysleksian neuraalista perustaa kaksostutkimusten avulla. Osa-aineisto on yhden koehenkilön aineisto noin neljän minuutin ajalta. Koasetelman tarkoituksena oli kontrolloida aivojen näköjärjestelmän toimintaa koehenkilökohtaisesti.

Koasetelma kontrolloitutumuksessa oli yksinkertainen: koehenkilölle näytettiin näytöltä kasvokuvia, joista hänen piti tunnistaa, olivatko kyseessä miehen vai naisen kasvot. Koehenkilölle oli annettu vastauslaite, jonka nappeja hänen tuli painaa riippuen siitä, kumman sukupuolen kasvokuvia hänelle näytettiin. Kasvokuvia näytettiin yhteensä 96 kappaletta ja ne välähtivät näytöllä nopeasti, vain 100 millisekunnin ajan. Koehenkilölle oli annettu ohjeeksi vastata mahdollisimman nopeasti vastauslaitteen avulla. MEG-mittaukset tehtiin Jyväskylän yliopiston Monitieteisen aivotutkimuskeskuksen MEG-laboratoriossa 306-kanavaisella (102 magnetometriä, 204 gradiometriä) Elekta Neuromag TRIUX

-laitteella. Näytteenottotaajuus oli noin neljän minuutin kokeen aikana 1000 hertsiä. Tässä tutkielmassa analysoitiin ainoastaan MAG-kanavien mittaama aineisto.

Kasvoihin liittyvän näköhavainnon informaation tehokas käsittely on normaalin sosiaalisen kanssakäymisen edellytys. Ihmisaivoissa on kasvojen välittämän informaation prosessointiin erikoistunut alue, FFA (fusiform face area), joka sijaitsee ohimolohkon alueella fusiform-poimussa. Kasvojen tunnistukseen liittyvät prosessit ovat keskittyneet merkittävän tarkasti FFA-alueelle, useimilla ihmisillä oikean aivopuoliskon dominanssilla. (Halgren ym., 2000). Kasvojen näkemisen on huomattu EEG- ja MEG-tutkimuksissa saavan aikaan niin kutsutun N170-vasteen, joka alkaa 130 millisekuntia kasvojen näkemisen jälkeen ja jonka huippu on noin 170 millisekunnin kohdalla (Halgren ym., 1991; Prieto ym., 2011). N170-vaste on selkeästi nähtävissä oleva piikki elektrofysiologisessa mittauksessa (Kanwisher ja Yovel, 2006).

Magnetometriä mittausta aineistosta estimoitujen lähdesignaalien ominaisuuksia tarkasteltiin magneettikenttien jakaumakuvien ja lähdesignaaleista lasketun ERF:ien avulla. Magnettikenttien jakaumakuvista eli topografoista saadaan selville magneettikenttien jakaumat pään pinnalla. BSS-menetelmien yksikäsitteisyysongelman (kappale 2) vuoksi magneettikenttien absoluuttisen suuruuden tai etumerkin tulkitseminen ei ole mielekäästä, joten ainoastaan niiden muoto ja suhteelliset muutokset ovat kiinnostavia suureita. N170-vaste pyrittiin löytämään estimoiduista lähdesignaaleista seuraavan referenssifunktion avulla:

$$r(t) = \begin{cases} 1, & \text{kun } t = [t_i + 130 \text{ ms}, t_i + 200 \text{ ms}], \quad i = 1, \dots, 96 \\ 0 & \text{muulloin.} \end{cases} \quad (25)$$

Kuten simulointikokeen tapauksessa, sovelluksessakin kiinnostavan signaalin löytäminen perustui suurimpaan itseiseen korrelaatioon referenssifunktion ja lähdesignaalin välillä. Referenssifunktio korreloi sellaisen signaalin kanssa, joka ei ole nolla aikavälillä 130-200 ms kasvokuvan esittämisen jälkeen, mutta on muutoin nollaa. ERF:t laskettiin segmentoimalla parhaiten referenssifunktion kanssa korreloinut lähdesignaali aikaväleillä $[t_i - 70 \text{ ms}, t_i + 500 \text{ ms}]$ ja laskemalla segmenttien keskiarvot jokaisessa aikapisteessä.

6.2.1 Tulokset

Sovellusaineiston tulokset esitetään kahdessa vaiheessa. Aluksi esitellään jokaisen käytetyn BSS-menetelmän estimointien ja parhaiten referenssifunktion kanssa korreloineiden lähdesignaalien topografiakuvat sekä näiden lähdesignaalien ERF:t. Lisäksi esitellään referenssifunktion kanssa parhaiten korreloineiden lähdesignaalien ERF:t kaikilla menetelmillä kaikille lasketuille ikkunoille. Parhaiten toimineelle BSS-menetelmälle laskettiin myös pääkomponenttianalyysiratkaisu, josta laskettua ERF:ää verrattiin ikkunointimenetelmän ratkaisuun. Kaikille BSS-menetelmille käytettiin simulointiasetelman mukaisesti ikkunakokoja $h = 9$ ja $h = 17$ ja lisäksi kokeiltiin ikkunakokoa $h = 13$. Parhaiten referens-

sifunktion kanssa korreloineille menetelmille (JADE ja SOBI) kokeiltiin myös simulointiasetelman suurinta ikkunakokoa ($h = 25$).

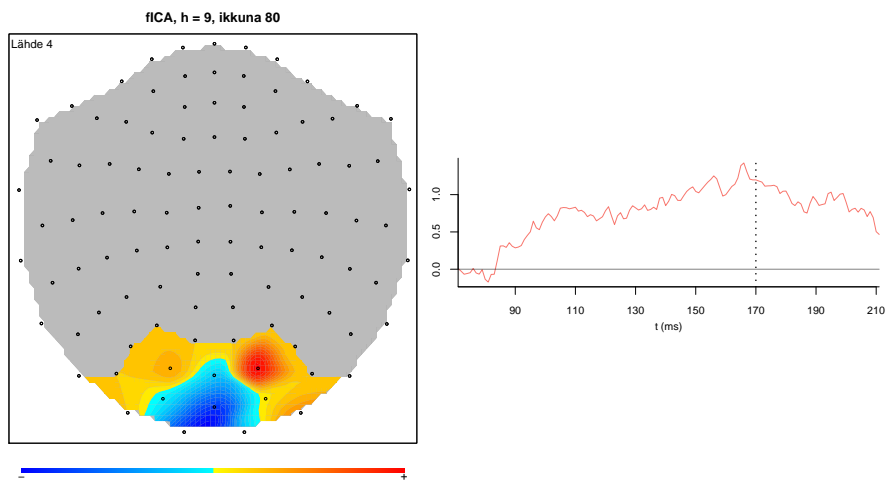
Taulukossa 5 on esitetty korkeimmat korrelaatioiden itseisarvot lähdesignaalien ja referenssifunktion välillä menetelmittäin ja ikkunoittain. Kaiken kaikkiaan itseiset korrelaatiot referenssifunktion kanssa ovat pieniä. Suurimmat korrelaatiot erottuivat kuitenkin muiden lähdesignaalien korrelaatioista selvästi. Suurimmat korrelaatiot referenssifunktion kanssa olivat fICA- ja JADE-menetelmillä estimoiduilla lähdesignaaleilla ($|\text{Cor}(\mathbf{s}_k(t), r(t))| = 0.16$), ikkunas-
sa $h = 9$. MEG-aineistossa on nähtävissä, että ICA-menetelmille referenssi-
funktioon sopiva lähde löytyy paremmin pienimmästä ikkunasta kuin muista
ikkunoista. SOBI-menetelmää käytettäessä ikkunan koolla ei juurikaan ole mer-
kitystä. ICA-menetelmille paras ikkunakoko on $h = 9$, SOBIlle $h = 17$. Erot ik-
kunoiden välillä ovat kuitenkin hyvin pienet kaikille menetelmille, joten ikkunan
koolla ei juurikaan näytä olevan vaikutusta kiinnostavan signaalin löytymiseen.

Taulukko 5: MEG-aineistosta estimoitujen lähdesignaalien itseisar-
voltaan suurin korrelaatio referenssifunktion kanssa ikkunoittain.

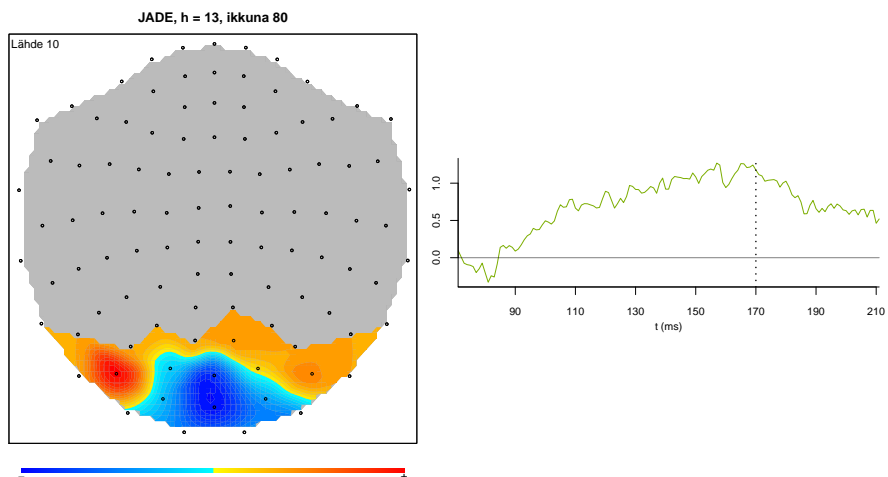
Menetelmä	$ \max [\text{Cor}(\mathbf{s}_k(t), r(t))] $			
	$h = 9$	$h = 13$	$h = 17$	$h = 25$
fICA	0.16	0.14	0.15	-
JADE	0.16	0.15	0.15	0.12
1-JADE	0.15	0.14	0.14	-
SOBI	0.15	0.14	0.15	0.14

Parhaiten referenssifunktioon liittyvien lähdesignaalien topografiat ja sig-
naaleista lasketut ERF:t on esitetty kuvissa 6-9. ICA-menetelmien osalta topo-
grafiakuvien perusteella löydettyt lähdesignaalit ovat jakaumiltaan melko lähellä
toisiaan eri ikkunakoista huolimatta (kuvat 6 ja 7) ja magneettikenttien lokaa-
tio on oletetusti ohimolohkon ja takaraivolohkon alueella, joskin fICAlle pään
oikealla puolella (kuva 6) ja JADElle vasemmalla (kuva 7). ICA-menetelmien
ERF:t eivät ole N170-vasteen kaltaisia: selvää piikkiä ei näy 170 millisekun-
nin kohdalla. SOBIlle ikkunas-
sa $h = 17$ magneettikentän jakauma on hiukan
erilainen verrattuna ICA-menetelmien topografioihin (kuva 8). Parhaiten refe-
rens-
sifunktion kanssa korreloivassa lähdesignaalis-
sa näkyy kasvokuvan aiheu-
tama molempien aivopuolisko-
jen aktivaatio oikean aivo-
puoliskon dominanssil-
la. Ikkunas-
sa $h = 25$ SOBI-ratkaisun
topografia eroaa merkittävästi
muiden ratkaisujen topo-
grafiasta. Molempien SOBI-
ratkaisujen ERF:t ovat lähem-
pänä N170-vastetta kuin ICA-
menetelmien: molemmista
ikkunoista laskettujen läh-
designaalien ERF:t ovat 130
millisekunnin kohdalla nol-
lassa tai lähellä nollaa ja
saavuttavat huippunsa 170
millisekunnin kohdalla. Näi-
denkään lähdesignaalien
ERF:t eivät kuitenkaan ole
kovin selkeitä, erityisesti
170 millisekunnin jälkeen
ERF-käyrän lasku on loivaa.

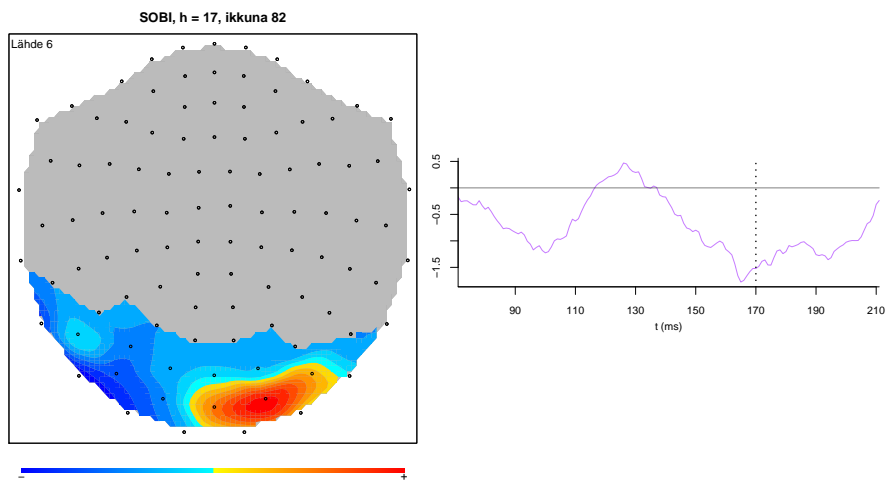
Kaikista BSS-menetelmistä parhaiten referenssifunktion kanssa korreloi-
neista lähdesignaaleista lasketut ERF:t eivät ole selkeästi N170-vasteen mukai-
sia (kuva 10). N170-vasteessa 130 millisekunnin kohdalla pitäisi näkyä muutosta



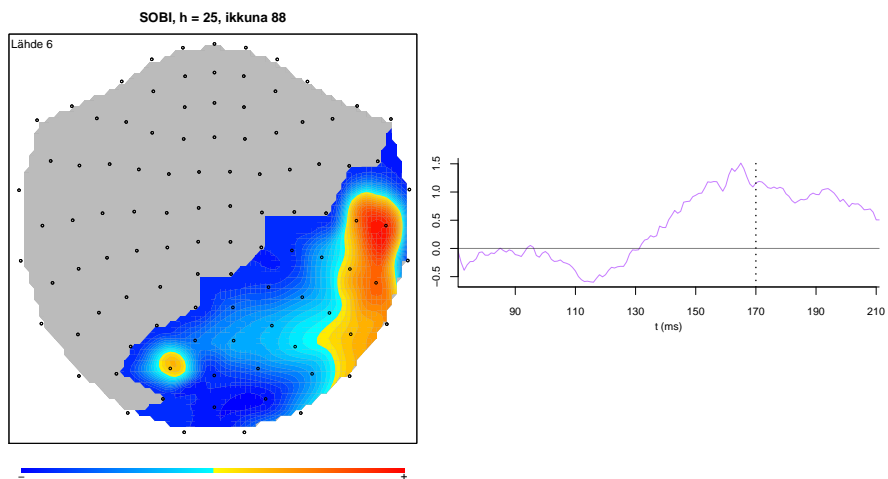
Kuva 6: fICA-menetelmällä estimoidun parhaiten referenssifunktion kanssa korreloivan komponentin topografia ikkunassa $k = 80$ ja komponentista laskettu ERF. ERF-kuvaan on merkitty nollassa mustalla viivalla ja 170:n millisekunnin kohta katkoviivalla.



Kuva 7: JADE-menetelmällä estimoidun parhaiten referenssifunktion kanssa korreloivan komponentin topografia ikkunassa $k = 80$ ja komponentista laskettu ERF. ERF-kuvaan on merkitty nollassa mustalla viivalla ja 170:n millisekunnin kohta katkoviivalla.



Kuva 8: SOBI-menetelmällä estimoidun parhaiten referenssifunktion kanssa korreloivan komponentin topografia ikkunassa $k = 82$ ja komponentista laskettu ERF. ERF-kuvaan on merkitty nollataso mustalla viivalla ja 170:n millisekunnin kohta katkoviivalla.

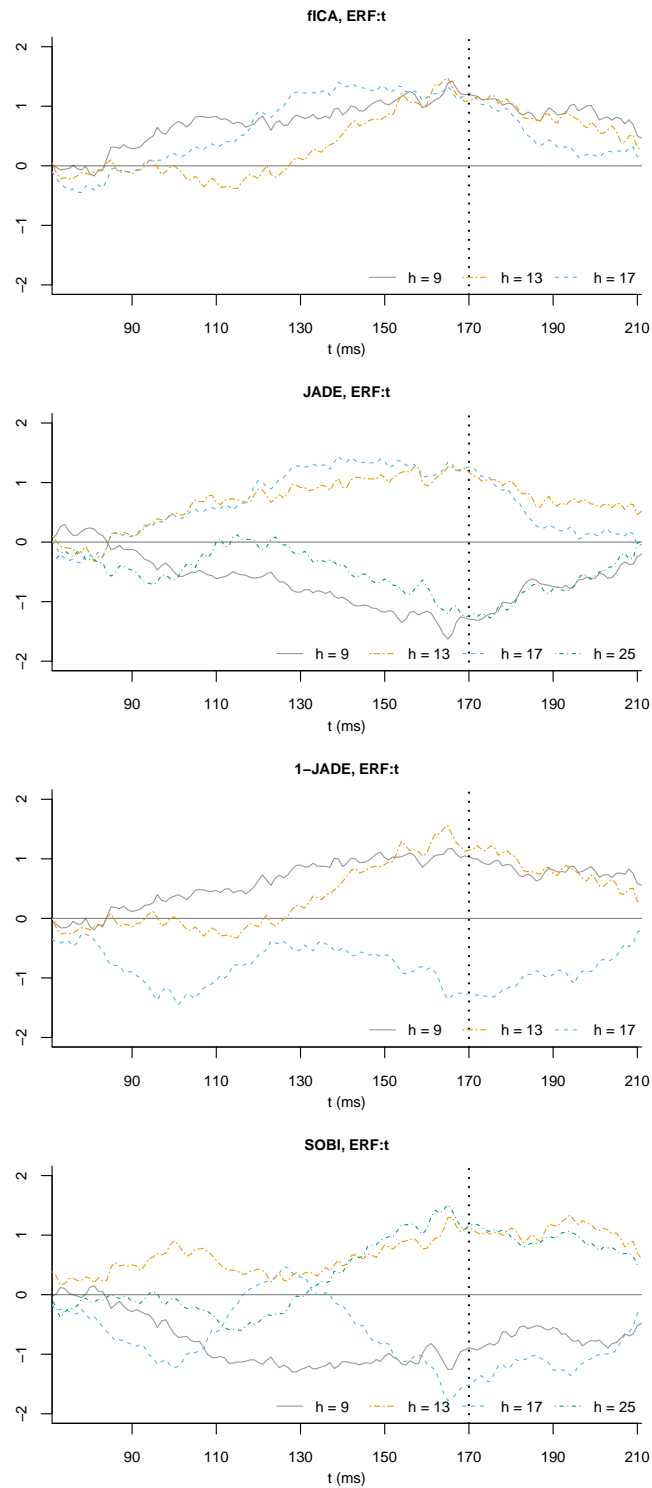


Kuva 9: SOBI-menetelmällä estimoidun parhaiten referenssifunktion kanssa korreloivan komponentin topografia ikkunassa $k = 88$ ja komponentista laskettu ERF. ERF-kuvaan on merkitty nollataso mustalla viivalla ja 170:n millisekunnin kohta katkoviivalla.

ERF:ssä ja 170 millisekunnin kohdalla selvä piikki. Lähimpänä odotetunkaltaista muotoa ovat fICAn ikkunasta $h = 13$, JADEn ikkunasta $h = 25$, 1-JADEn ikkunasta $h = 13$ ja SOBIn ikkunoista $h = 13$ ja $h = 25$ ERF:t, tosin piikki näkyy 150-170 millisekunnin välillä. N170-vaste näkyisi luultavasti selvemmin, mikäli ERF laskettaisiin usean koehenkilön aineistosta keskiarvoistamalla. Toisaalta ERF-käyrissä on myös näkyvissä suuriamplitudisia, hitaita aaltoja, mikä viittaa siihen, että MEG-aineisto pitäisi ylipäästäsuodattaa ennen BSS-menetelmien käyttöä.

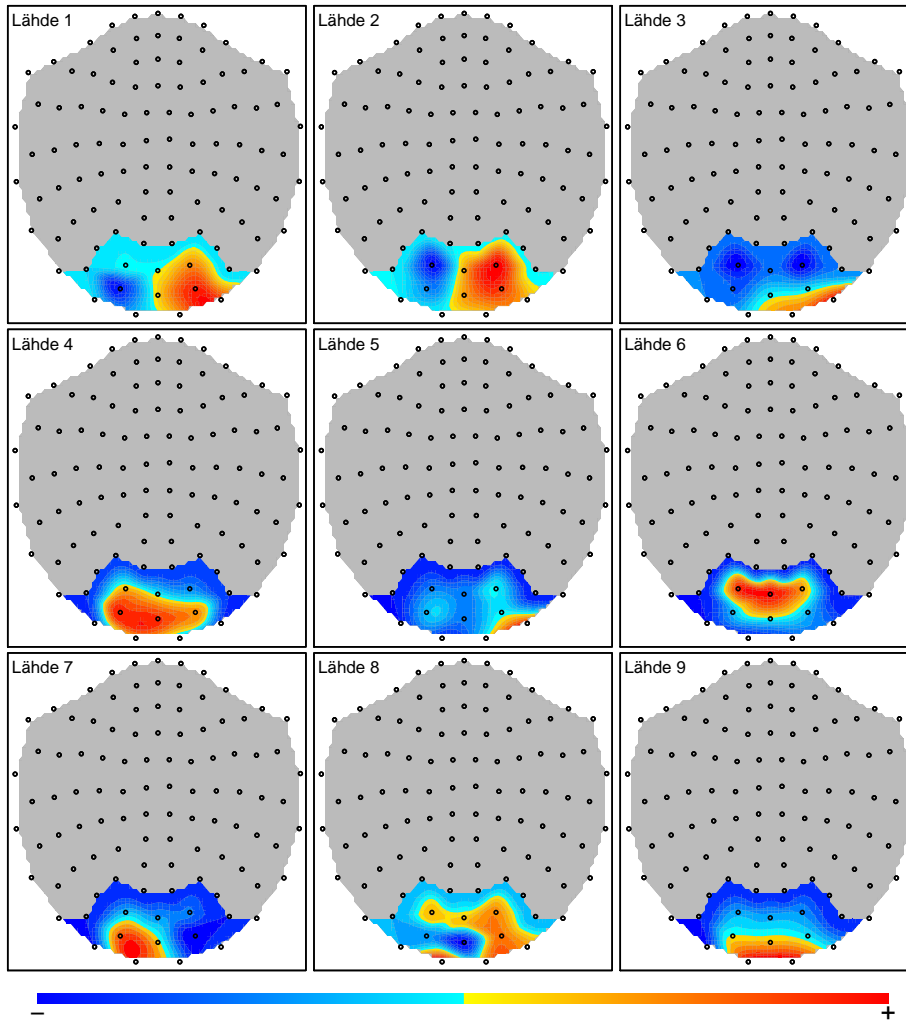
Tarkastellaan SOBI-menetelmän tuloksia tarkemmin ikkunoittain, jotta voidaan vertailla ikkunan koon aiheuttamia muutoksia komponenttien topografioissa. Pienimmässä ikkunassa ($h = 9$) topografiakuissa värien vaihtelu on selkeää ja estimoidut lähdesignaalit ovat selvästi toisistaan eroavia (kuva 11). Referenssifunktion kanssa parhaiten korreloiva lähde (lähde 3) ei kuitenkaan topografiakuvan perusteella vaikuta jakaumaltaan etsityn vasteen kaltaiselta: pelkkää referenssifunktiota ei siis voida pitää luotettavana perusteena lähdesignaalin löytymiselle. Lisäksi simulointitulosten perusteella SOBI on tehokkaampi suurilla ikkunoilla, joten referenssifunktion osoittama lähdesignaali ei välttämättä vastaa etsittyä lähdesignaalia. Ikkunakoolla $h = 13$ ikkunasta $k = 81$ laskettu parhaiten referenssifunktion kanssa korreloiva lähdesignaali eroaa topografialtaan pienemmän ikkunan lähdesignaalista huomattavasti (kuva 12). Ikkunakoolla $h = 17$ parhaiten referenssifunktion kanssa korreloiva lähdesignaali näyttää vastaavan paremmin oletetunlaista topografiaa (kuva 13) oikean puolen dominanssilla. Isoimmassa ikkunassa (kuva 14) ongelmana näyttää olevan se, että nollassa eroavaa aivoaktivaatiota näkyy olevan kaikkien sensorien alueella melko tasaisesti. Tämä vaikeuttaa kuvien tulkintaa: kiinnostavan aktivaation erottaminen on tällaisista topografiakuvista hankalaa.

Pääkomponenttianalysiratkaisuun valittiin 65 ensimmäistä pääkomponenttia, jotka selittivät yli 99 % kaikkien komponenttien vaihtelusta. Parhaiten referenssifunktion kanssa korreloi SOBilla laskettu lähdesignaali numero 7 ($|\text{Cor}(\mathbf{s}_7(t), r(t))| = 0.15$), josta laskettu ERF on kuvassa 15. Pääkomponenttiratkaisusta laskettu ERF on selvästi monotonisempi kuin BSS-menetelmillä lasketut ERF:t, eikä N170-vastetta pysty erottamaan ERF:stä lainkaan.



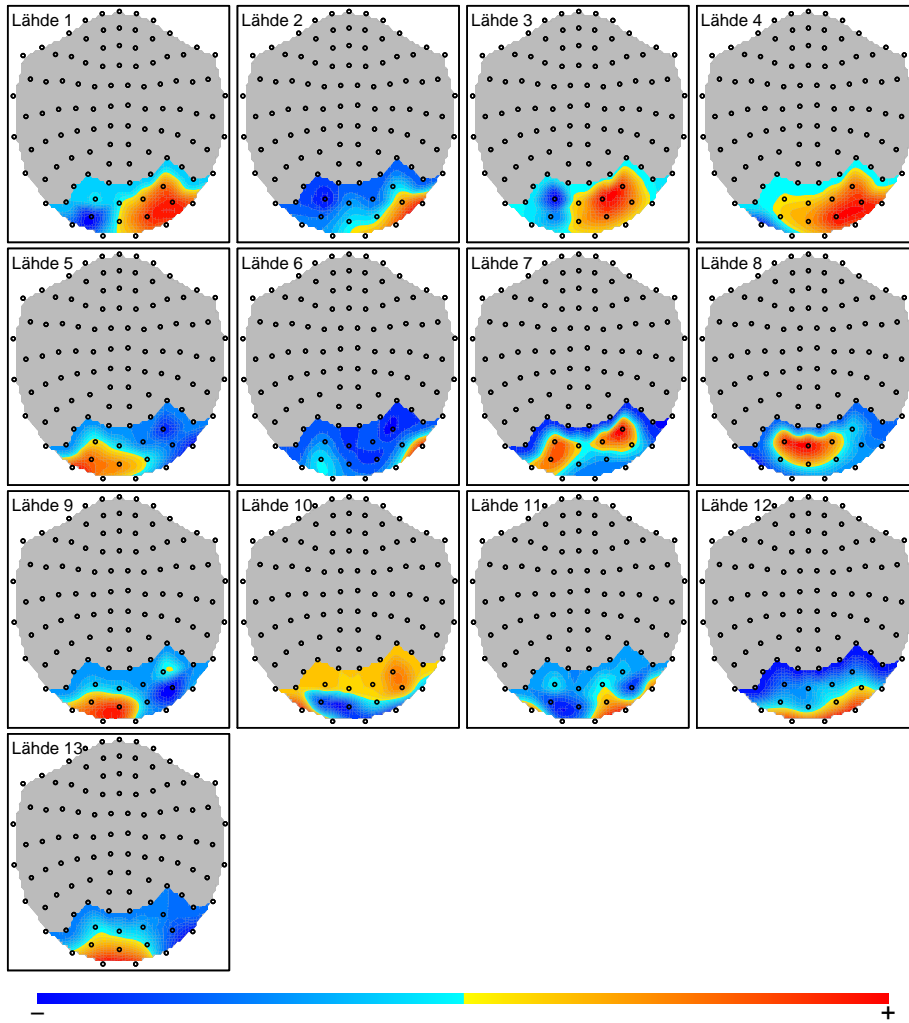
Kuva 10: Kaikista BSS-menetelmistä lasketut ERF:t ikkunoittain. ERF-kuviin on merkitty nolлатaso mustalla viivalla ja 170:n millisekunnin kohta katkoviivalla.

SOBI, $h = 9$, ikkuna 80



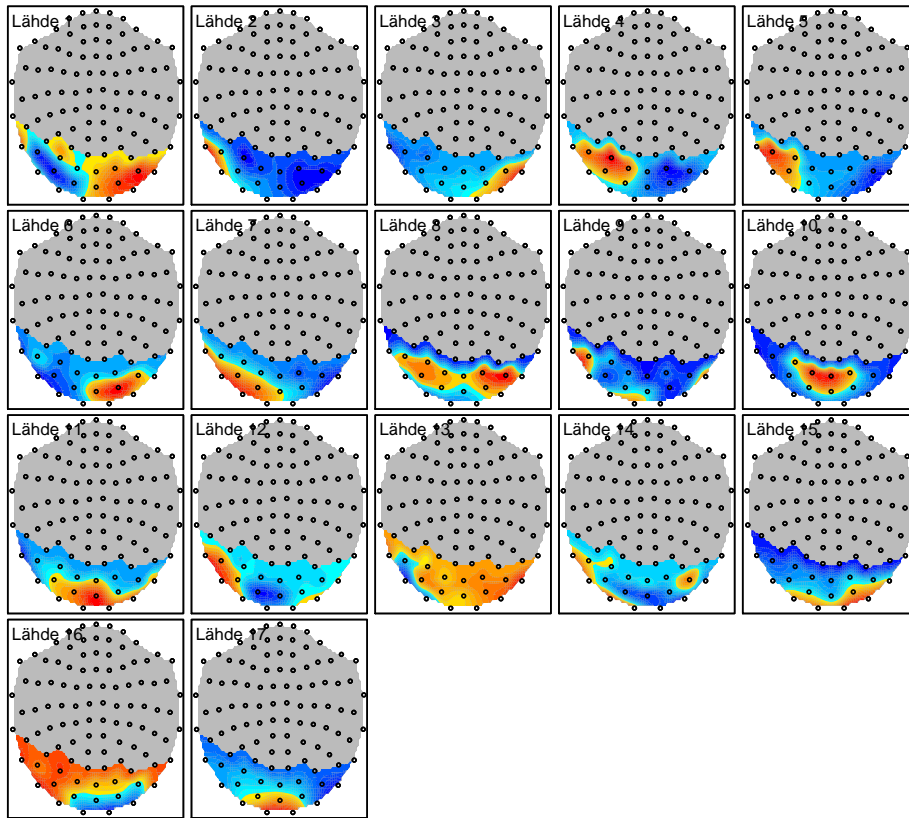
Kuva 11: SOBI-menetelmällä estimoitujen lähdesignaalien jakaumat ikkunassa 80 ikkunakoolla $h = 9$. Referenssifunktion kanssa parhaiten korreloi kolmas lähdesignaali (kuvassa Lähde 3).

SOBI, $h = 13$, ikkuna 81



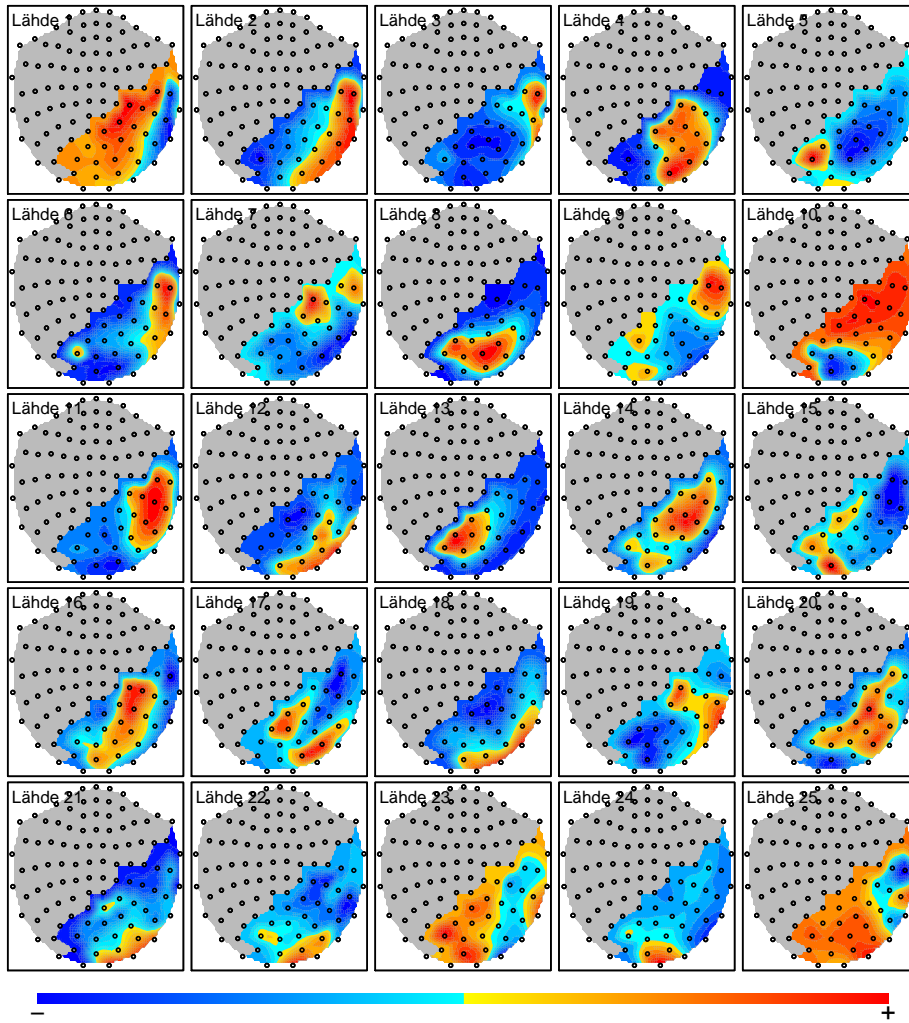
Kuva 12: SOBI-menetelmällä estimoitujen lähdesignaalien jakaumat ikkunassa 81 ikkunakoolla $h = 13$. Referenssifunktion kanssa parhaiten korreloi seitsemäs lähdesignaali (kuvassa Lähde 7).

SOBI, $h = 17$, ikkuna 82

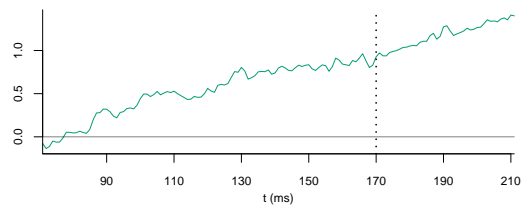


Kuva 13: SOBI-menetelmällä estimoitujen lähdesignaalien jakaumat ikkunassa 82 ikkunakoolla $h = 17$. Referenssifunktion kanssa parhaiten korreloi kuudes lähdesignaali (kuvassa Lähde 6).

SOBI, $h = 25$, ikkuna 88



Kuva 14: SOBI-menetelmällä estimoitujen lähdesignaalien jakaumat ikkunassa 88 ikkunakoolla $h = 25$. Referenssifunktion kanssa parhaiten korreloi kuudes lähdesignaali (kuvassa Lähde 6).



Kuva 15: Pääkomponenttiratkaisusta SOBI-menetelmällä estimoidun parhaiten referenssifunktion kanssa korreloivan lähdesignaalin ERF.

7 Yhteenveto

Tilasuuntaisen ikkunointimenetelmän toimivuutta MEG-aineistolle eri sokean signaalinerottelun menetelmillä tutkittiin simulointikokeiden ja MEG-tutkimusaineiston avulla. Ikkunoinnin hyödyllisyyttä tutkittiin neljällä eri BSS-menetelmällä: uudelleenladatulla FastICAlla, JADElla, 1-JADElla ja SOBilla. Ikkunan koon vaikutuksia tutkittiin soveltamalla ikkunoinnissa eri kokoisia ikkunoita sekä simulointiasetelmaan että tutkimusaineistoon.

Simulointikokeiden perusteella lähdesignaalien erottelu on sitä hankalampaa, mitä vaihtelevampaa kohinaa simulointiaineistoon lisätään. Referenssifunktiota vastaava lähdesignaali löytyi poikkeuksetta parhaiten SOBI-menetelmällä, joskin BSS-ikkunoinnista oli eniten hyötyä ICA-menetelmiä käytettäessä. Simuloidut lähdesignaalit olivat heikosti stationaarisia AR-, MA- ja ARMA-prosesseja, mikä saattaa selittää SOBIn selvää paremmuutta. Ikkunointimenetelmä yhdessä BSS-menetelmien kanssa löytää referenssifunktion kaikkien BSS-menetelmien kanssa paremmin kuin pääkomponenttianalyysin ja BSS-menetelmän yhdistelmä yhtä poikkeusta lukuunottamatta: pienemmän kohinavarianssin tapauksessa pääkomponenttianalyysin ja BSS-menetelmän yhdistelmä toimi paremmin 25:llä pääkomponentilla kuin ikkunointi vastaavassa tapauksessa. Pääkomponenttiratkaisu toimi parhaiten silloin, kun komponentteja valittiin enemmän kuin mitä alkuperäisiä lähdesignaaleja oli. Koko aineiston BSS-ratkaisu oli SOBIn tapauksessa hieman parempi kuin suurimman ikkunan ($h = 25$) antama ratkaisu, mutta 1-JADElle ikkunointi tuotti hieman paremman tuloksen kuin koko aineistosta laskettu BSS-ratkaisu.

MEG-aineiston jonkinlainen esiprosessointi ennen BSS-ikkunoinnin toteutusta olisi hyödyllistä. ERF-kuvien perusteella sopivan ylipäästösuotimen käyttö saattaisi tuoda N170-komponentin paremmin näkyviin myös yhden koehenkilön aineistolle. BSS-menetelmien estimoimien lähdesignaalien topografiakuvien perusteella referenssifunktio auttoi löytämään oletetunlaiset komponentit tutkimuksessa käytetylle koasetelmalle, vaikka korrelaatiot referenssifunktion kanssa ovatkin pieniä. BSS-ikkunointia tulisi kokeilla useamman koehenkilön aineistoon, jotta voitaisiin varmistua, ettei kasvojen prosessointia vastaavien lähdesignaalien löytyminen ollut koehenkilökohtaista. Lisäksi vaadittavien esiprosessointisuodatusten arviointi olisi helpompaa, jos tuloksia voitaisiin verrata useamman koehenkilön välillä.

Referenssifunktion valinta ei ole yksinkertaista eikä sen käyttäminen ongelmatonta: näin karkealla funktion valinnalla ei voida saavuttaa kovin korkeita korrelaatioita aivovasteiden kanssa. Arvoja nolla ja yksi saava laatikkofunktio ei koskaan kuvaa täysin sovellusaineistosta etsittävää vastetta, joten korrelaatiot aivovasteiden ja referenssifunktion välillä ovat pieniä. Simulointitapauksessa tilanne on eri, kun referenssifunktio kuvaa täysin etsittävää vastetta. Tämä selittää erityisesti pienen kohinavarianssin simulointiasetelmasta laskettujen lähdesignaalien moninkertaisesti korkeammat korrelaatiot referenssifunktion kanssa verrattuna aineistosta laskettuihin korrelaatioihin. Lisäksi lähteiden ja referens-

sifunktion väliseen itseiseen korrelaatioon ei voi luottaa sokeasti: löydetyn lähteen vastaavuus haluttuun ilmiöön tulee tarkistaa topografiakuvista tai ERF-käyristä.

Valitut BSS-menetyelmät eivät välttämättä ole optimaalisia tutkitunkaltaisten aiovasteiden etsimiseen. Metsomaa ym. (2016) huomauttavat, että aiovasteet ovat usein epästationaarisia, eikä niitä tulisi tutkia BSS-menetyelmillä, jotka olettavat lähdesignaalien olevan stationaarisia. Vertailun vuoksi simuloinnin ja sovelluksen voisi toteuttaa esimerkiksi jollakin Nordhausenin artikkelissaan (2014) esittelemällä epästationaarisen lähde-erottelun mallilla (non-stationary source separation, NSS). NSS-mallit käyttävät SOS-mallien tapaan hyödyksi havaintosignaalien aikarakennetta.

BSS-ikkunoinnin sijaan voitaisiin kokeilla myös niin kutsuttua harvan sekoitusmatriisin ICA-menetyelmää, jossa sekoitusmatriisin estimoinnissa käytetään hyväksi prioritietoa sekoitusmatriisin rakenteesta (Hyvärinen ja Raju, 2002). Menetyelmässä oletetaan, että sekoitusmatriisi sisältää vain vähän nollasta eroavia alkioita. Tällainen sekoitusmatriisin rakenne sopisi hyvin MEG-aineiston sovelluksiin ja erityisesti aivojen näköjärjestelmän toiminnan tutkimiseen, koska näköjärjestelmän hermoverkoston tiedetään käyttävän vain tiettyjä hermoverkoston reittejä koko verkoston sijaan (Bell ja Sejnowski, 1997).

Aineiston ikkunointi on erityisen hyödyllistä silloin, kun halutaan tutkia hyvin lokaalisti syntyviä aivosignaaleja ja ollaan kiinnostuneita pienistä muutoksista estimoitujen lähdesignaalien välillä. MEG-tutkimuksessa ikkunoinnin hyöty saattaa olla juuri tässä tutkielmassa käytetyn tutkimusaineiston tyyppisen kontrollointiasetelman analysoinnissa: ikkunoinnin avulla voidaan tutkia ainoastaan ne alueet, joissa terveissä ihmisaivoissa näkyy tietynlaista aktivaatiota. Tulosten perusteella voidaan todeta, vastaako kyseinen koehenkilö sitä kohderyhmää, jonka edustajan tutkimukseen halutaan osallistuvan. Paikantamalla ikkuna tarkasti ja säätämällä ikkunakokoa kontrollointiaktivaation fysiologiaan sopivaksi voidaan säästää aikaa koehenkilöiden sopivuuden analysoinnissa: tutkijan ei tarvitse käydä läpi kaikkia 102 estimoitua lähdesignaalia.

Viitteet

- Azevedo, F.A., Carvalho, L.R., Grinberg, L.T., Farfel, J.M., Ferretti, R.E., Leite, R.E., Lent, R., Herculano-Houzel, S. ym. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, 513(5):532–541, 2009.
- Bell, A.J. ja Sejnowski, T.J. The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.
- Brewick, P. ja Smyth, A. Increasing the efficiency and efficacy of second-order blind identification SOBI methods. *Structural Control and Health Monitoring*, 24(6), 2017.
- Bunse-Gerstner, A., Byers, R., ja Mehrmann, V. Numerical methods for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 14(4):927–949, 1993.
- Cardoso, J-F. & Souloumiac, A. Blind beamforming for non-Gaussian signals. Teoksessa *IEE proceedings F (radar and signal processing)*, osa 140, sivut 362–370. IET, 1993.
- Cardoso, J.F. Source separation using higher order moments. Teoksessa *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, sivut 2109–2112. IEEE, 1989.
- Cardoso, J.F. ja Souloumiac, A. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1996.
- Cichocki, A. ja Amari, S. *Adaptive blind signal and image processing: learning algorithms and applications*. John Wiley & Sons, 2002.
- Clarke, J. ja Braginski, A.I. *The SQUID Handbook: Applications of SQUIDs and SQUID systems*. John Wiley & Sons, 2006.
- Esposito, F., Seifritz, E., Formisano, E., Morrone, R., Scarabino, T., Tedeschi, G., Cirillo, S., Goebel, R., ja Di Salle, F. Real-time independent component analysis of fMRI time-series. *Neuroimage*, 20(4):2209–2224, 2003.
- Fatima, Z., Quraan, M.A., Kovacevic, N., ja McIntosh, A.R. ICA-based artifact correction improves spatial localization of adaptive spatial filters in MEG. *Neuroimage*, 78:284–294, 2013.
- Halgren, E., Marinkovic, K., Baudena, P., Devaux, B., Broglin, D., Heit, G., ja Chauvel, P. Human intracranial potentials evoked by faces. *Society for Neuroscience Abstracts*, 17:656, 1991.
- Halgren, E., Raij, T., Marinkovic, K., Jousmäki, V., ja Hari, R. Cognitive response profile of the human fusiform face area as determined by MEG. *Cerebral Cortex*, 10(1):69–81, 2000.

- Horn, R. ja Johnson, C. *Topics in Matrix Analysis*. Cambridge University Press, 1994.
- Hsu, S.H., Mullen, T.R., Jung, T.P., ja Cauwenberghs, G. Real-time adaptive EEG source separation using online recursive independent component analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(3):309–319, 2016.
- Hyvärinen, A., Karhunen, J., ja Oja, E. *Independent component analysis*. John Wiley & Sons, 2001.
- Hyvärinen, A. ja Oja, E. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483, 1997.
- Hyvärinen, A. ja Raju, K. Imposing sparsity on the mixing matrix in independent component analysis. *Neurocomputing*, 49(1):151–162, 2002.
- Hyvärinen, A., Ramkumar, P., Parkkonen, L., ja Hari, R. Independent component analysis of short-time Fourier transforms for spontaneous EEG/MEG analysis. *NeuroImage*, 49(1):257–271, 2010.
- Kanwisher, N. ja Yovel, G. The fusiform face area: a cortical region specialized for the perception of faces. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 361(1476):2109–2128, 2006.
- Karvanen, J. ja Koivunen, V. Blind separation methods based on Pearson system and its extensions. *Signal Processing*, 82(4):663–673, 2002.
- Karvanen, J. ja Theis, F.J. Spatial ICA of fMRI data in time windows. Teoksessa *AIP Conference Proceedings*, osa 735, sivut 312–319. AIP, 2004.
- Kiviniemi, V., Vire, T., Remes, J., Elseoud, A.A., Starck, T., Tervonen, O., ja Nikkinen, J. A sliding time-window ICA reveals spatial variability of the default mode network in time. *Brain connectivity*, 1(4):339–347, 2011.
- Luck, S.J. *An introduction to the event-related potential technique*. MIT press, 2014.
- Makeig, S., Enghoff, S., Jung, T.P., ja Sejnowski, T.J. Moving-window ICA decomposition of EEG data reveals event-related changes in oscillatory brain activity. Teoksessa *2nd Int. Workshop on Independent Component Analysis and Signal Separation*, sivut 627–632. 2000.
- Metsomaa, J., Sarvas, J., ja Ilmoniemi, R.J. Blind source separation of event-related EEG/MEG. *IEEE Transactions on Biomedical Engineering*, 2016.
- Miettinen, J., Illner, K., Nordhausen, K., Oja, H., Taskinen, S., ja Theis, F.J. Separation of uncorrelated stationary time series using autocovariance matrices. *Journal of Time Series Analysis*, 37(3):337–354, 2016.

- Miettinen, J., Nordhausen, K., Oja, H., ja Taskinen, S. Fast equivariant JADE. Teoksessa *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, sivut 6153–6157. IEEE, 2013.
- Miettinen, J., Nordhausen, K., Oja, H., ja Taskinen, S. *fICA: Classical, reloaded and adaptive FastICA algorithms*, 2015. R package version 1.0-3.
- Miettinen, J., Nordhausen, K., ja Taskinen, S. Blind source separation based on joint diagonalization in R: The packages JADE and BSSasyp. *Journal of Statistical Software*, 76(2):1–31, 2017.
- Müller, K.R., Vigario, R., Meinecke, F., ja Ziehe, A. Blind source separation techniques for decomposing event-related brain signals. *International Journal of Bifurcation and Chaos*, 14(02):773–791, 2004.
- Nordhausen, K. On robustifying some second order blind source separation methods for nonstationary time series. *Statistical Papers*, 55(1):141–156, 2014.
- Nordhausen, K., Ilmonen, P., Mandal, A., Oja, H., ja Ollila, E. Deflation-based FastICA reloaded. Teoksessa *2011 19th European Signal Processing Conference*, sivut 1854–1858. IEEE, 2011.
- Oja, H., Sirkiä, S., ja Eriksson, J. Scatter matrices and independent component analysis. *Austrian Journal of Statistics*, 35(2&3):175–189, 2006.
- Onton, J., Westerfield, M., Townsend, J., ja Makeig, S. Imaging human EEG dynamics using independent component analysis. *Neuroscience & Biobehavioral Reviews*, 30(6):808–822, 2006.
- Papanicolaou, A.C. *The Oxford handbook of functional brain imaging in neuropsychology and cognitive neurosciences*. Oxford University Press, 2017.
- Prieto, E.A., Caharel, S., Henson, R., ja Rossion, B. Early (N170/M170) face-sensitivity despite right lateral occipital brain damage in acquired prosopagnosia. *Frontiers in Human Neuroscience*, 5, 2011.
- R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.
- Tang, A., Sutherland, M., ja Wang, Y. Contrasting single-trial ERPs between experimental manipulations: Improving differentiability by blind source separation. *NeuroImage*, 29(1):335–346, 2006.
- Tang, A.C., Liu, J.Y., ja Sutherland, M.T. Recovery of correlated neuronal sources from EEG: the good and bad ways of using SOBI. *Neuroimage*, 28(2):507–519, 2005.
- Taskinen, S., Miettinen, J., ja Nordhausen, K. A more efficient second order blind identification method for separation of uncorrelated stationary time series. *Statistics & Probability Letters*, 116:21–26, 2016.

- Vigário, R., Jousmäki, V., Hämäläinen, M., Hari, R., ja Oja, E. Independent component analysis for identification of artifacts in magnetoencephalographic recordings. Teoksessa *Advances in neural information processing systems*, sivut 229–235. 1998.
- Vigario, R. ja Oja, E. BSS and ICA in neuroinformatics: from current practices to open challenges. *IEEE reviews in biomedical engineering*, 1:50–61, 2008.
- Walter, W.G., Cooper, R., Aldridge, V., McCallum, W., ja Winter, A. Contingent negative variation: an electric sign of sensori-motor association and expectancy in the human brain. *Nature*, 203(4943):380–384, 1964.

Liite: Simulointiasetelman R-koodi

```
library(dplyr)
set.seed(071117)

# Sensoreiden spatiaalinen ikkunointi -----

# Funktio, joka hakee koordinaatit aineistosta.
### coordinates = koordinaattirakenne
### h = ikkunan koko (kuinka monta lähintä sensoria valitaan)
# Palauttaa sensorit.
spatial.windows <- function(coordinates, h){
  # koordinaatit array-rakenteessa
  dim <- dim(coordinates)
  # koordinaattien keskipisteet
  cp <- lapply(1:dim[3], function(i) apply(coordinates[, ,i], 1, mean))
  cp2 <- cp
  dist_mat <- sapply(lapply(cp, function(cp) sapply(cp2, function(cp2)
    sum({cp-cp2}^2))), rbind)
  # sarakkeilla valitun sensorin h:n lähimmän sensorin indeksit
  sensors <- apply(dist_mat, 1, function(d) which(d %in% sort(d)[1:{h+1}]))

  return(sensors)
}

# Sensoreiden koordinaattien lukeminen aineistosta.
## locations on (3 x 4 x 306)-dimensioinen array, jossa kussakin
## (3 x 4)-matriisissa on neliönmuotoisten sensoreiden nurkkapisteiden
## koordinaatit xyz-koordinaatistossa MAG-, GRAD1- ja GRAD2-sensoreille.
## tässä tutkielmassa käytettiin aineiston mukaista MEG-sensorien geometriaa.
dim <- dim(locations)
cp <- lapply(1:dim[3], function(i) apply(locations[, ,i], 1, mean)) %>%
  unlist %>%
  matrix(ncol = 3, byrow = TRUE)

i <- seq(3, dim(locations)[3], by=3) #MAG-sensorien indeksit
#i <- seq(1, dim(locations)[3], by=3) #GRAD1
#i <- seq(2, dim(locations)[3], by=3) #GRAD2
coordinates <- locations[, ,i] #MAG-sensorien koordinaatit
cpm <- cp[i,] # sensorien keskipisteet

h <- 9 # ikkunan koko
# alla h-1, koska myös valittu sensori sisältyy ikkunaan.
sensors <- spatial.windows(coordinates, h-1)
```

```

# esim. h kpl sensoria 1 lähimmän sensorin (coordinates-arrayn mukaiset)
# indeksit:
sensors[,1]

# Simulointiin tarvittavat funktiot -----

# Referenssifunktion tekevä funktio.
### T = aikapisteiden lkm
### t1 = ensimmäinen mahd. aika kun vaste näkyy
### t2 = viimeinen mahd. aika kun vaste näkyy
### et = t:n arvo, jolloin ärsyke on aineistossa näkynyt
# Palauttaa referenssifunktion.
resf.gen <- function(T0, t1, t2, et) {
  resf <- rep(0,T0)
  for(i in 1:length(et)){
    resf[(et[i]+t1):(et[i]+t2)] <- 1
  }
  return(resf)
}

# Komponenttien korrelaatiot referenssifunktion kanssa laskeva funktio
# ikkuna-ICAlla.
### X = generoitu data
### method = algoritmin nimen merkkijono, yksi seuraavista: sobi, jade,
###          kjade, fica
### h = ikkunan koko
### resf = referenssifunktio
# Palauttaa referenssifunktion ja laskettujen lähteiden väliset suurimmat
# korrelaatiot.
resf.cor <- function(X, method=c("sobi", "jade", "kjade", "fica"),
                    resf, sensors, print=T) {
  method <- match.arg(method, c("sobi", "jade", "kjade", "fica"))
  if (print==T) print(paste("Processing dataset:", X$ID))
  max.cor <- vector(length=ncol(sensors))

  if (method=="sobi") {
    lags <- unique(c(seq(2, 10, by=2), seq(10, 50, by=5),
                    seq(50, 100, by=10)))
    for (sens in 1:ncol(sensors)) {
      sobi <- JADE::SOBI(X$dat[,sensors[,sens]], k=lags,
                        method="frjd", eps=1e-03, maxiter=5000,
                        na.action=na.fail)
      max.cor[sens] <- sobi$S %>%
        apply(2, function(S, resf) cor(S, resf) %>% abs, resf) %>%
        max
    }
  }
}

```

```

    }

    return(max.cor)
}

if (method=="jade") {
  for (sens in 1:ncol(sensors)) {
    jade <- JADE::JADE(X$dat[,sensors[,sens]], eps=1e-03, maxiter=5000,
                      na.action=na.fail)
    max.cor[sens] <- jade$$ %>%
      apply(2, function(S, resf) cor(S, resf) %>% abs, resf) %>%
      max
  }

  return(max.cor)
}

if (method=="kjade") {
  for (sens in 1:ncol(sensors)) {
    kjade <- JADE::k_JADE(X$dat[,sensors[,sens]], k=1, eps=1e-03,
                          maxiter=5000, na.action=na.fail)
    max.cor[sens] <- kjade$$ %>%
      apply(2, function(S, resf) cor(S, resf) %>% abs, resf) %>%
      max
  }

  return(max.cor)
}

if (method=="fica"){
  for (sens in 1:ncol(sensors)) {
    fica <- fICA::reloaded_fICA(X$dat[,sensors[,sens]], g="pow3",
                                dg=NULL, G=NULL, kj=1, method="alpha",
                                inR=FALSE, eps=1e-03, maxiter=5000)
    max.cor[sens] <- fica$$ %>%
      apply(2, function(S, resf) cor(S, resf) %>% abs, resf) %>%
      na.omit %>%
      max
  }

  return(max.cor)
}

}

# Simulointi -----

```

```

ch <- 102 # kanavien lkm
n.sources <- 20 # lähteiden lkm
T0 <- 1000 # aikapisteiden lkm
nsim <- 1000 # simulointikierrosten lkm
et <- 420 # eventtien aikapiste(et), aineiston mukainen
var <- "1" # kohinavarianssin arvo

# referenssifunktio
resf <- resf.gen(T0, 130, 200, et)

# sekoitusmatriisi
A <- matrix(runif(n.sources*ch, -1, 1), nrow = ch, ncol = n.sources)
A[-(1:n.sources), 1] <- 0

# lähdesignaalit ja havaitut signaalit
X <- vector(mode = "list", length = nsim)
for (i in 1:nsim) {
  S <- vector(mode = "list", length = n.sources)
  S[[1]] <- resf
  S[[2]] <- arima.sim(model = list(ar = c(0.8)), n = T0,
    innov = rexp(T0, 1))
  S[[3]] <- arima.sim(model = list(ar = c(0, -0.3)), n = T0,
    innov = rexp(T0, 1))
  S[[4]] <- arima.sim(model = list(ma = c(0.2)), n = T0,
    innov = rexp(T0, 1))
  S[[5]] <- arima.sim(model = list(ma = c(-0.1, -0.2)), n = T0,
    innov = rexp(T0, 1))
  S[[6]] <- arima.sim(model = list(ar = c(0.1, -0.2, 0.3)), n = T0,
    innov = rexp(T0, 1))
  S[[7]] <- arima.sim(model = list(ma = c(-0.1, 0.2, -0.3)), n = T0,
    innov = rexp(T0, 1))
  S[[8]] <- arima.sim(model = list(ar = c(-0.1, -0.2), ma = c(0.1, 0.2)),
    n = T0, innov = rexp(T0, 1))
  S[[9]] <- arima.sim(model = list(ar = c(0.7, -0.4), ma = c(-0.7, 0.4)),
    n = T0, innov = rexp(T0, 1))
  S[[10]] <- arima.sim(model = list(ar = c(-0.5, 0.4), ma = c(0.7, -0.4)),
    n = T0, innov = rexp(T0, 1))
  S[[11]] <- arima.sim(model = list(ar = c(0.2, -0.2)), n = T0,
    innov = rexp(T0, 1))
  S[[12]] <- arima.sim(model = list(ma = c(-0.2, 0.2)), n = T0,
    innov = rexp(T0, 1))
  S[[13]] <- arima.sim(model = list(ar = c(0, -0.4), ma = c(0, 0.4)),
    n = T0, innov = rexp(T0, 1))
  S[[14]] <- arima.sim(model = list(ar = c(-0.2, 0, 0.5)),
    n = T0, innov = rexp(T0, 1))
}

```

```

S[[15]] <- arima.sim(model = list(ma = c(-0.5, 0.2, 0.1)),
                    n = T0, innov = rexp(T0, 1))
S[[16]] <- arima.sim(model = list(ar = c(0, -0.2), ma = c(-0.7, 0, 0.2)),
                    n = T0, innov = rexp(T0, 1))
S[[17]] <- arima.sim(model = list(ar = c(0.1, -0.2), ma = c(0.5, 0.6)),
                    n = T0, innov = rexp(T0, 1))
S[[18]] <- arima.sim(model = list(ar = c(0.7), ma = c(0, 0.1)),
                    n = T0, innov = rexp(T0, 1))
S[[19]] <- arima.sim(model = list(ar = c(0.2, -0.4), ma = c(-0.5)),
                    n = T0, innov = rexp(T0, 1))
S[[20]] <- arima.sim(model = list(ar = c(0.7), ma = c(-0.7, 0.2)),
                    n = T0, innov = rexp(T0, 1))
S <- mapply(cbind,S)

X[[i]] <- list(dat = S %*% t(A) + matrix(rnorm(T0*ch, 0, 0.1), nrow = T0,
                                         ncol = ch), ID = i)
}

# ikkunointi
res.sobi <- lapply(X, resf.cor, method = "sobi", resf = resf,
                  sensors = sensors, print = T)
# kootaan kunkin simuloinnin korkeimmat korrelaatiot yhteen
sapply(res.sobi, max)

res.jade <- lapply(X, resf.cor, method = "jade", resf = resf,
                  sensors = sensors, print = T)
sapply(res.jade, max)

res.kjade <- lapply(X, resf.cor, method = "kjade", resf = resf,
                  sensors = sensors, print = T)
sapply(res.kjade, max)

res.fica <- lapply(X, resf.cor, method = "fica", resf = resf,
                  sensors = sensors, print = T)
sapply(res.fica, max)

```