

Sokean signaalinkäsittelyn menetelmiä: sovelluksena
EEG-aineiston analysointi

Suvi Mänttari

Tilastotieteen pro gradu -tutkielma

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
Joulukuu 2011

Jyväskylän yliopisto

Matematiikan ja tilastotieteen laitos

Mänttari, Suvi: Sokean signaalinkäsittelyn menetelmiä: sovelluksena EEG-aineiston analysointi

Tilastotieteen pro gradu -tutkielma, 54 s.

Joulukuu 2011

Tiivistelmä

Sokeita signaalinkäsittelyn menetelmiä (Blind Source Separation, BSS) käytetään erottelemaan keskenään sekoittuneista signaaleista yksittäiset lähdesignaalit. Nämä tuntemattomat lähdesignaalit tulee estimoida pelkästään havaittujen signaalien avulla, sillä sekoitusmenetelmä on tuntematon.

Tässä tutkielmassa käsitellään hajonta- ja autokovarianssimatriisien diagonalisointiin perustuvia BSS-menetelmiä. Perinteiset AMUSE-, SOBI- ja FOBI-menetelmät esitellään. Näille algoritmeille esitellään myös robustit vaihtoehdot, jotka saadaan käyttämällä robusteja kovarianssiestimaattoreita. Lisäksi perinteiset AMUSE- ja FOBI-menetelmät laajennetaan tilanteeseen, jossa havaitut signaalit sisältävät kohinaa.

Menetelmien toimivuutta vertaillaan sekä simulointikokeiden että käytännön sovellusten avulla. Simulointikokeet osoittavat, että jos lähdesignaaleissa on aikariippuvuutta, AMUSE ja SOBI toimivat paremmin kuin FOBI. Jos taas aikariippuvuutta ei ole, niin FOBI toimii parhaiten. Simuloinnit myös osoittavat, että perinteiset menetelmät ovat hyvin herkkiä poikkeaville havainnoille. Robustit menetelmät toimivat sen sijaan hyvin, vaikka signaaleissa on poikkeavia havaintoja. Robustien menetelmien heikkoutena on se, että algoritmit ovat suurten aineistojen tapauksessa laskennallisesti hitaita. Lopuksi osoitetaan, että jos havaitut signaalit sisältävät kohinaa, lähdesignaalien erottelu ei onnistu perinteisillä AMUSE- ja FOBI-menetelmillä. Menetelmien laajennukset kohinamalleille toimivat huomattavasti paremmin.

Sovellusesimerkeissä BSS-menetelmiä käytetään äänisignaalien ja aivosähkökäyrien analysointiin. Alkuperäisten aikariippuvien äänisignaalien erottelu onnistuu esimerkissä parhaiten AMUSE-menetelmällä. Esimerkissä myös osoitetaan, että perinteinen AMUSE on herkkä poikkeaville havainnoille. Aivosähkökäyrien analysointi osoittaa, että BSS-menetelmien avulla aineistosta voidaan erotella sekä häiriötekijöitä että todellisia aivoperäisiä komponentteja, joiden tarkka mittaaminen pään pinnalta ei ole mahdollista.

Avainsanoja: AMUSE, FOBI, riippumattomien komponenttien analyysi, robustisuus, SOBI, sokea signaalinkäsittely

Sisältö

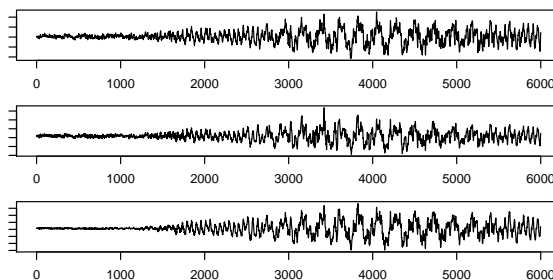
1	Johdanto	5
2	Sokea signaalinkäsittely (BSS)	7
2.1	BSS-malli	7
2.2	Yksikäsitteisyydestä	8
2.3	Hajonta- ja autokovarianssimatriisit	8
2.4	AMUSE-menetelmä	9
2.4.1	Valkaisu	9
2.4.2	Rotatointi	10
2.4.3	AMUSE-algoritmi	11
2.4.4	Affini invarianttisuus	12
2.5	SOBI-menetelmä	13
2.5.1	SOBI-algoritmi	13
2.6	Robusti SOBI-menetelmä	14
3	Riippumattomien komponenttien analyysi (ICA)	16
3.1	ICA-malli	16
3.2	Hajontamatriisit	16
3.3	FOBI-menetelmä	17
3.3.1	Valkaisu	17
3.3.2	Rotatointi	17
3.3.3	FOBI-algoritmi	18
3.4	Robusti versio FOBI-menetelmästä	19
3.5	Usean hajontamatriisin yhteisdiagonalisointi	20
4	Kohinaa sisältävä BSS-malli	21
4.1	AMUSE-menetelmä kohinamallille	21
4.1.1	Valkaisu ja rotatointi	21
4.1.2	AMUSE-algoritmi kohinamallille	23
4.2	EFOBI-menetelmä	24
4.2.1	Valkaisu ja rotatointi	25
4.2.2	EFOBI-algoritmi	25
5	Menetelmien vertailu	27
5.1	Vertailuindeksejä	27
5.2	Simulointiasetelmat	28
5.3	Tulokset	30

6	Sovelluksia	35
6.1	Äänisignaalien erottelu	35
6.2	Aivosähkökäyrien analysointi	38
6.2.1	Aineisto	38
6.2.2	Tulokset	39
7	Yhteenveto	43
	Viitteet	45
A	Todistuksia	47
B	R-koodeja	50

1 Johdanto

Sokeat signaalinkäsittelyn menetelmät (Blind Source Separation, BSS) on kasvava ja laajalle levinnyt tutkimusalue, jolla on paljon sovelluksia useilla eri tieteenaloilla. BSS-ongelma syntyy esimerkiksi silloin, kun useat vastaanottimet havaitsevat yhtä aikaa joukon signaaleja. Tällöin jokainen havaittu signaali on painotettu summa alkuperäisistä lähdesignaaleista. BSS-menetelmien ideana onkin löytää nämä tuntemattomat lähdesignaalit, kun havaitsemme ainoastaan jonkinlaisia lineaarisia sekoituksia niistä (Cichocki & Amari, 2002). Termiä 'sokea' käytetään, koska sekoitusmenetelmä on tuntematon. BSS-menetelmiä sovelletaan esimerkiksi aivotutkimuksissa, kuvankäsittelyssä, telekommunikaatiossa ja ekonometriassa.

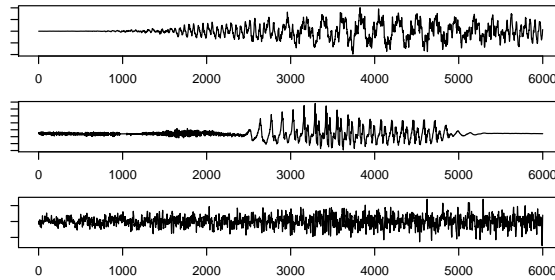
Yksinkertainen BSS-malli on muotoa $\mathbf{x} = \mathbf{A}\mathbf{s}$, missä \mathbf{x} sisältää havaitut signaalit ja \mathbf{s} tuntemattomat lähdesignaalit. Matriisi \mathbf{A} on tuntematon sekoitusmatriisi. Tuntemme ainoastaan havainnot, joiden perusteella tulisi estimoida sekä sekoitusmatriisi että lähdesignaalit. Havainnot x_1, x_2, \dots, x_p voivat olla esimerkiksi eripuolilla huonetta sijaitsevien mikrofoniin äänittämiä puhesignaaleja, kun useampi henkilö puhuu huoneessa samanaikaisesti toisistaan riippumatta (kuva 1.1). Jokaisen mikrofoniin äänittämiä signaali on tällöin sekoitus yksittäisten henkilöiden puhesignaaleista s_1, s_2, \dots, s_p (kuva 1.2). BSS-menetelmien avulla voidaan erotella tästä puheensorinasta yksittäisten henkilöiden äänet.



Kuva 1.1: Havaitut signaalit

BSS-ongelman ratkaisemiseksi on kehitetty lukuisia algoritmeja. Toiset algoritmit olettavat lähdesignaalit korreloimattomiksi ja käyttävät ongelman ratkaisemiseksi tietoa lähdesignaalien aikarakenteesta. Toiset algoritmit sen sijaan olettavat lähdesignaalien olevan riippumattomia. Tätä BSS:n erikoistapausta kutsutaan riippumattomien komponenttien analyysiksi eli ICAksi (Independent Component Analysis) (Hyvärinen et al., 2001).

Tässä pro gradu -tutkielmassa käsitellään hajonta- ja autokovarianssimatriisien käyttöön perustuvia ratkaisualgoritmeja. Näiden algoritmien avulla



Kuva 1.2: Alkuperäiset puhesignaalit

lähdesignaalit löydetään kahdessa vaiheessa. Ensimmäisessä vaiheessa havaitut signaalit valkaistaan käyttäen tavallista otoskovarianssimatriisia. Valkaisun jälkeen riittää, että signaalit rotatoidaan. Rotatointiin tarvittava ortogonaalinen matriisi löydetään joko diagonalisoimalla yksi tai useampi autokovarianssimatriisi (AMUSE, SOBI) (Belouchrani et al., 1997) tai diagonalisoimalla neljänsiin momentteihin perustuva hajontamatriisi (FOBI) (Cardoso, 1989). Nämä perinteiset menetelmät ovat herkkiä poikkeaville havainnoille eli lähdesignaalien erottelu ei onnistu, jos aineistossa on epätavallisen suuria tai pieniä havaintoja. Algoritmeista saadaan robustimmat käyttämällä robusteja kovarianssiestimaattoreita (Oja et al., 2006; Nordhausen et al., 2011b).

Tutkielman tarkoituksena on vertailla erilaisten ratkaisumenetelmien toimivuutta sekä simulointien että käytännön sovellusten avulla. Menetelmien teoreettinen vertailu suoritetaan simuloimalla keinotekoiset lähdesignaalit, joista sekoituksen jälkeen estimoidaan lähdesignaalit kullakin menetelmällä. Estimaatteja voidaan nyt verrata todellisiin arvoihin ja ratkaisujen hyvyttä mitata erilaisilla indekseillä. Tutkielmassa käsitellään lisäksi kahta sovellusesimerkkiä. Ensimmäisessä esimerkissä vertaillaan eri ratkaisumenetelmien kykyä erotella äänisignaaleja. Jälkimmäisessä esimerkissä sokeaa signaalinkäsittelyä sovelletaan aivosähkökäyrien eli EEG:n analysointiin. Sokeiden signaalinkäsittelyn menetelmien avulla pyritään estimoimaan todelliset, aivojen sisältä lähtöisin olevat EEG-signaalit, joiden tarkka mittaaminen pään pinnalta on mahdotonta.

Tutkielman rakenne on seuraava. Luvussa 2 määritellään BSS-malli ja sen oletukset. Lisäksi esitellään AMUSE- ja SOBI-algoritmit sekä robusti SOBI-algoritmi. Luvussa 3 käydään läpi ICA-malli sekä FOBI-algoritmi ja sen robusti versio. Luvussa 4 mallia laajennetaan tilanteeseen, jossa havaitut signaalit sisältävät kohinaa. Menetelmien teoreettinen vertailu suoritetaan luvussa 5. Luvussa 6 sokeaa signaalinkäsittelyä sovelletaan äänisignaalien erotteluun sekä EEG-aineiston analysointiin. Luku 7 sisältää tutkielman yhteenvedon.

2 Sokea signaalinkäsittely (BSS)

2.1 BSS-malli

Tarkastellaan aluksi yksinkertaista sokeaa signaalinkäsittelyn ongelmaa, jossa malli ei sisällä kohinaa, ja havaittuja signaaleja on yhtä monta kuin lähdesignaaleja. Yksinkertainen BSS-malli (Cichocki & Amari, 2002) on muotoa

$$\mathbf{x}_t = A\mathbf{s}_t,$$

missä $\mathbf{x}_t = [x_{1t}, x_{2t}, \dots, x_{pt}]^T$ on p -ulotteinen sekoitetut signaalit sisältävä vektori ja t kuvaa ajanhetkeä. Vektori $\mathbf{s}_t = [s_{1t}, s_{2t}, \dots, s_{pt}]^T$ on p -ulotteinen lähdesignaalien vektori. Matriisi A on $p \times p$ -sekoitusmatriisi.

Tavoitteena on selvittää tuntemattomat lähdesignaalit \mathbf{s}_t havaittujen signaalien \mathbf{x}_t avulla ilman minkäänlaista ennakkotietoa sekoitusmatriisista A . Tarkoituksena on siis estimoida $p \times p$ -matriisi W siten, että vektori

$$\mathbf{y}_t = W\mathbf{x}_t$$

sisältää korreloimattomat lähdesignaalit skaalausta ja permutaatiota vaille (ks. luku 2.2). Jotta tämä olisi mahdollista, tehdään seuraavat oletukset:

- (A1) Sekoitusmatriisi A on täysiasteinen.
- (A2) Lähdesignaalien vektori \mathbf{s}_t on toisen asteen stationaarinen prosessi, toisin sanoen odotusarvo ja kovarianssi ovat olemassa eivätkä ne riipu ajanhetkestä t . Tällöin siis $E[\mathbf{s}_t] = \mu$ on vakio ja $\text{Cov}(\mathbf{s}_t, \mathbf{s}_{t+\tau})$ riippuu vain viiveestä τ .
- (A3) Lähdesignaalien \mathbf{s}_t kaikki autokovarianssimatriisit ovat diagonaalisia, toisin sanoen kaikilla $\tau = 0, 1, 2, \dots$

$$\text{Cov}(\mathbf{s}_t, \mathbf{s}_{t+\tau}) = D_\tau,$$

missä D_τ on diagonaalimatriisi.

Merkintöjen yksinkertaistamisen vuoksi tehdään lisäksi oletus, että lähdesignaalit ovat keskistettyjä eli niiden odotusarvo on nolla. Lähdesignaalit saadaan keskistettyä, kun keskistetään sekoitetut signaalit, sillä jos sekoitettujen signaalien odotusarvo $E[\mathbf{x}_t] = \mathbf{0}$, niin silloin myös $E[\mathbf{s}_t] = A^{-1}E[\mathbf{x}_t] = \mathbf{0}$. Oletetaan siis tästä eteenpäin, että odotusarvot ovat nollia.

2.2 Yksikäsitteisyydestä

Lähdesignaalien \mathbf{s}_t yksikäsitteinen ratkaiseminen ei ole mahdollista, sillä sekoitusmatriisi A on täysin tuntematon. Signaalit saadaan ratkaistua vain skaalausta ja permutaatiota vaille yksikäsitteisesti. Tämä nähdään seuraavasta. Jos $D = \text{diag}(d_1, d_2, \dots, d_p)$ on jokin diagonaalimatriisi, niin tällöin pätee

$$\mathbf{x}_t = A\mathbf{s}_t = (AD^{-1})(D\mathbf{s}_t).$$

Sekoitusmatriisi AD^{-1} ja uusi skaalattu lähdesignaalien vektori $D\mathbf{s}_t$ toteuttavat nyt BSS-mallin, sillä jos vektorin \mathbf{s}_t komponentit ovat korreloimattomia myös vektorin $D\mathbf{s}_t$ komponentit ovat korreloimattomia. Koska skaalaa ei voida määrittää yksikäsitteisesti, voidaan tästä eteenpäin olettaa, että lähdesignaalien varianssit ovat ykkösiä eli $\text{Cov}(\mathbf{s}_t) = I_p$. Valitsemalla diagonaalimatriisin D diagonaalelementit siten, että ne ovat ± 1 , huomataan että myöskään signaalien merkkejä ei voida ratkaista yksikäsitteisesti.

Olkoon nyt P $p \times p$ -permutaatiomatriisi. Permutaatiomatriisi saadaan muuttamalla identtisen matriisin I_p rivien järjestystä. Kun jokin matriisi kerrotaan vasemmalta permutaatiomatriisilla voidaan sen rivien järjestystä vaihtaa. Nyt pätee

$$\mathbf{x}_t = A\mathbf{s}_t = (AP^{-1})(P\mathbf{s}_t).$$

Koska lähdesignaalien permutointi ei vaikuta niiden korreloimattomuuteen, myös AP^{-1} ja $P\mathbf{s}_t$ toteuttavat BSS-mallin.

BSS-ongelman yksikäsitteinen ratkaiseminen on siis mahdotonta. Käytännön sovelluksissa on kuitenkin usein tärkeintä tietää lähdesignaalien muoto ei niinkään niiden suuruus, merkki tai järjestys.

2.3 Hajonta- ja autokovarianssimatriisit

Tässä tutkielmassa käsiteltävät BSS-ongelman ratkaisumenetelmät perustuvat hajonta- ja autokovarianssimatriisien käyttöön. Merkitään käytettävää hajontamatriisia $\Sigma(\mathbf{x}_t)$ ja autokovarianssimatriisia $\Sigma_\tau(\mathbf{x}_t)$. Ratkaisu löydetään kahdessa vaiheessa. Ensimmäisessä vaiheessa sekoitetut signaalit valkaistaan hyödyntäen hajontamatriisia $\Sigma(\mathbf{x}_t)$. Toisessa vaiheessa valkaistut signaalit rotatoidaan autokovarianssimatriisin $\Sigma_\tau(\mathbf{x}_t)$ avulla. Algoritmeja voidaan muokata vaihtamalla käytettäviä matriiseja. Matriisien on kuitenkin toteutettava seuraavat oletukset:

(B1) Hajontamatriisi $\Sigma(\mathbf{x}_t)$ on affiinisti ekvivariantti, toisin sanoen

$$\Sigma(B\mathbf{x}_t) = B\Sigma(\mathbf{x}_t)B^T,$$

kun B on mikä tahansa ei-singulaarinen $p \times p$ -matriisi.

(B2) Autokovarianssimatriisi $\Sigma_\tau(\mathbf{x}_t)$ on ortogonaalisesti ekvivariantti, toisin sanoen

$$\Sigma_\tau(Q\mathbf{x}_t) = Q\Sigma_\tau(\mathbf{x}_t)Q^T,$$

kun Q on mikä tahansa ortogonaalinen $p \times p$ -matriisi.

(B3) Jos vektorin \mathbf{x}_t komponenttien ristikorrelaatiot ovat nollia eli

$$E[x_{it}x_{j(t+\tau)}] = 0, \quad \text{kun } i \neq j,$$

niin $\Sigma_\tau(\mathbf{x}_t)$ on diagonaalinen.

Seuraavissa kappaleissa kerrotaan tarkemmin, millaisia matriiseja eri menetelmät hyödyntävät ja kuinka lähdesignaalit estimoidaan.

2.4 AMUSE-menetelmä

AMUSE-menetelmä (Algorithm for Multiple Unknown Signals Extraction) on laskennallisesti helppo ratkaisualgoritmi, joka perustuu yhden autokovarianssimatriisin diagonalisointiin (Tong et al., 1990). Ratkaisu löydetään kahdessa vaiheessa. Ensimmäisessä vaiheessa sekoitetut signaalit valkaistaan käyttämällä kovarianssimatriisia

$$\Sigma(\mathbf{x}_t) = E[\mathbf{x}_t\mathbf{x}_t^T].$$

Toisessa vaiheessa valkaistut signaalit rotatoidaan käyttämällä autokovarianssimatriisia viiveellä τ

$$\Sigma_\tau(\mathbf{x}_t) = E[\mathbf{x}_t\mathbf{x}_{t+\tau}^T].$$

Huomaa, että jos \mathbf{x}_t noudattaa BSS-mallia, niin autokovarianssimatriisi $\Sigma_\tau(\mathbf{x}_t)$ on symmetrinen, sillä

$$\Sigma_\tau(\mathbf{x}_t) = \Sigma_\tau(A\mathbf{s}_t) = A\Sigma_\tau(\mathbf{s}_t)A^T,$$

ja oletuksen (A3) nojalla matriisi $\Sigma_\tau(\mathbf{s}_t)$ on diagonaalinen. Tarkastellaan seuraavissa kappaleissa tarkemmin, miten valkaisu ja rotointi tehdään.

2.4.1 Valkaisu

Sokeiden signaalinkäsittelyn menetelmien yhteydessä termillä valkaisu tarkoitetaan signaalien välisen korrelaation poistamista. Valkaisussa havaintovektorille tehdään sellainen lineaarimuunnos, jonka jälkeen sen komponentit ovat korreloimattomia ja komponenttien varianssit ovat ykkösiä. Tarkoituksena on siis etsiä $p \times p$ -matriisi V siten, että

$$\mathbf{z}_t = V\mathbf{x}_t \quad \text{ja} \quad \Sigma(\mathbf{z}_t) = I_p,$$

missä I_p on $p \times p$ -identtinen matriisi.

Valkaisu voidaan tehdä monella tapaa. Tässä tutkielmassa käytettävä valkaisumenetelmä perustuu havaittujen signaalien \mathbf{x}_t kovarianssimatriisin ominaisarvohajotelmaan, sillä se on helposti yleistettävissä myös kohinaa sisältäville malleille. Merkitään nyt kyseistä ominaisarvohajotelmaa seuraavasti

$$\Sigma(\mathbf{x}_t) = \Gamma \Lambda \Gamma^T,$$

missä Γ on ortogonaalinen matriisi, jonka sarakkeina ovat matriisin $\Sigma(\mathbf{x}_t)$ normeeratut ominaisvektorit ja $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ sisältää vastaavat ominaisarvot. Haluttu valkaisumatriisi saadaan määrittelemällä (Cichocki & Amari, 2002)

$$V = \Lambda^{-1/2} \Gamma^T = \text{diag} \left(\frac{1}{\sqrt{\lambda_1}}, \frac{1}{\sqrt{\lambda_2}}, \dots, \frac{1}{\sqrt{\lambda_p}} \right) \Gamma^T.$$

Nyt kovarianssimatriisin affiinista ekvivanttisuudesta seuraa, että valkaistuille signaaleille $\mathbf{z}_t = V \mathbf{x}_t$ pätee

$$\Sigma(\mathbf{z}_t) = \Sigma(V \mathbf{x}_t) = V \Sigma(\mathbf{x}_t) V^T = \Lambda^{-1/2} \Gamma^T (\Gamma \Lambda \Gamma^T) \Gamma \Lambda^{-1/2} = I_p.$$

Huomaa, että valkaistut signaalit noudattavat BSS-mallia, sillä

$$\mathbf{z}_t = V \mathbf{x}_t = V A \mathbf{s}_t = \tilde{A} \mathbf{s}_t,$$

missä $\tilde{A} = V A = \Lambda^{-1/2} \Gamma^T A$. Uusi sekoitusmatriisi \tilde{A} on ortogonaalinen, sillä

$$I_p = \Sigma(\mathbf{z}_t) = \tilde{A} \Sigma(\mathbf{s}_t) \tilde{A}^T = \tilde{A} \tilde{A}^T. \quad (2.1)$$

Valkaisu siis helpottaa BSS-ongelman ratkaisua, sillä ratkaisumatriisin etsintä voidaan rajoittaa ortogonaalisiin matriiseihin, toisin sanoen lähdesignaalit löydetään rotatoimalla valkaistut signaalit.

2.4.2 Rotatointi

Rotatointimatriisi löydetään diagonalisoimalla valkaistujen signaalien autokovarianssimatriisi. Koska autokovarianssimatriisi on ortogonaalisesti ekvivantti, niin

$$\Sigma_\tau(\mathbf{z}_t) = \Sigma_\tau(\tilde{A} \mathbf{s}_t) = \tilde{A} \Sigma_\tau(\mathbf{s}_t) \tilde{A}^T,$$

missä $\tilde{A} = V A$ on ortogonaalinen matriisi kuten yllä. Nyt $\Sigma_\tau(\mathbf{s}_t)$ on oletuksen (A3) nojalla diagonaalinen, joten BSS-mallin sekoitusmatriisi ja lähdesignaalit saadaan estimoitua ominaisarvohajotelman

$$\Sigma_\tau(\mathbf{z}_t) = \Gamma_z \Lambda_z \Gamma_z^T \quad (2.2)$$

avulla. Tällöin ratkaisumatriisi on

$$W = \Gamma_z^T V,$$

ja

$$\mathbf{y}_t = W \mathbf{x}_t$$

sisältää lähdesignaaliin merkkiä ja permutaatiota vaille (todistus, ks. liite A).

Jotta lähdesignaaliin saadaan eroteltua niiden autokovarianssien viiveellä τ eli matriisin $E[\mathbf{s}_t \mathbf{s}_{t+\tau}^T]$ diagonaalelementtien on oltava erisuuria. Tällöin ominaisarvohajotelman (2.2) matriisi Γ_z sisältää p merkkiä ja permutaatiota vaille yksikäsitteistä ominaisvektoria ja menetelmällä saadaan eroteltua p korreloimatonta lähdesignaalia. Käytännön sovelluksissa AMUSEn erottelemien signaalien lukumäärää voidaan tutkia tarkastelemalla valkaistujen havaintojen otosautokovarianssimatriisin ominaisarvoja. Käytettävä viive τ pyritään valitsemaan siten, että ominaisarvot ovat mahdollisimman kaukana toisistaan.

Ratkaisumatriisi W voidaan myös löytää käyttäen yleistettyä ominaisarvohajotelmaa, jolloin ei tarvitse käydä läpi kaksivaiheista prosessia. Tällöin W saadaan ratkaistua yleistetystä ominaisarvohajotelmasta

$$\Sigma_\tau(\mathbf{x}_t) W^T = \Sigma(\mathbf{x}_t) W^T D,$$

jos matriisilla $\Sigma(\mathbf{x}_t)^{-1} \Sigma_\tau(\mathbf{x}_t)$ on erisuuret ominaisarvot (ks. Cichocki & Amari, 2002, luku 4.2.2).

2.4.3 AMUSE-algoritmi

Oletetaan nyt, että olemme havainneet signaalit \mathbf{x}_t , missä $t = 1, 2, \dots, n$. Tuntemattomat lähdesignaaliin voidaan estimoida korvaamalla edellä mainitut kovarianssimatriisit vastaavilla otoskovarianssimatriiseilla, toisin sanoen

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T \quad \text{ja} \quad \hat{\Sigma}_\tau(\mathbf{x}_t) = \frac{1}{n - \tau} \sum_{t=1}^{n-\tau} (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_{t+\tau} - \bar{\mathbf{x}})^T.$$

Tämä johtaa seuraavaan algoritmiin:

1. Lasketaan havaintojen \mathbf{x}_t otoskovarianssimatriisin ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T = \hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T.$$

2. Valkaistaan havaintoaineisto

$$\mathbf{z}_t = \hat{V}(\mathbf{x}_t - \bar{\mathbf{x}}), \text{ missä } \hat{V} = \hat{\Lambda}^{-1/2}\hat{\Gamma}^T.$$

3. Lasketaan valkaistuille havainnoille symmetrisoitu otosautokovarianssimatriisi viiveellä τ (yleensä $\tau = 1$)

$$\tilde{\Sigma}_\tau(\mathbf{z}_t) = \frac{1}{2}[\hat{\Sigma}_\tau(\mathbf{z}_t) + \hat{\Sigma}_\tau(\mathbf{z}_t)^T].$$

4. Lasketaan ominaisarvohajotelma

$$\tilde{\Sigma}_\tau(\mathbf{z}_t) = \hat{\Gamma}_z \hat{\Lambda}_z \hat{\Gamma}_z^T.$$

5. Lasketaan ratkaisumatriisi

$$\hat{W} = \hat{\Gamma}_z^T \hat{V}$$

ja lähdesignaalit

$$\hat{\mathbf{s}}_t = \hat{W}(\mathbf{x}_t - \bar{\mathbf{x}}).$$

Huomaa, että askeleessa 3 otoksesta laskettu autokovarianssimatriisi ei ole välttämättä symmetrinen, joten se täytyy symmetrisoida, jotta ominaisarvohajotelma on hyvin määritelty ja helppo laskea.

2.4.4 Affini invarianttisuus

AMUSE-algoritmin avulla ratkaistut lähdesignaalit ovat invariantteja affiinille muunnokselle. Toisin sanoen, jos $\mathbf{x}_t^* = B\mathbf{x}_t$, missä B on mikä tahansa ei-singulaarinen $p \times p$ -matriisi, ja W^* on signaaleista \mathbf{x}_t^* laskettu ratkaisumatriisi, niin tällöin

$$W^* \mathbf{x}_t^* = W \mathbf{x}_t$$

merkkiä ja permutaatiota vaille. Tulos pätee itse asiassa myös silloin, kun algoritmissa käytetään mitä tahansa hajonta- ja autokovarianssimatriisia, jotka toteuttavat luvun 2.3 oletukset. Todistus löytyy liitteestä A.

2.5 SOBI-menetelmä

AMUSE-algoritmin ongelmana on se, että lähdesignaalien autokovarianssit eivät välttämättä ole erisuuria valitulla viiveellä. Tämän ongelman välttämiseksi on kehitetty SOBI-menetelmä (Belouchrani et al., 1997). SOBI-algoritmi (Second-Order Blind Identification) on AMUSEn laajennus, joka perustuu yhden autokovarianssimatriisin sijaan useamman eri viiveillä lasketun autokovarianssimatriisin yhteisdiagonalisointiin. Usean eri viiveillä olevan autokovarianssimatriisin yhteisdiagonalisointi on hyödyllistä, koska on todennäköistä, että jollakin käytettävistä viiveistä lähdesignaalien autokovarianssit ovat erisuuria ja signaalit saadaan näin ollen eroteltua.

Merkitään nyt $\hat{\Sigma}_{\tau_k}(\mathbf{z}_t)$ on valkaistujen havaintojen otosautokovarianssimatriisi viiveellä τ_k , $k = 1, 2, \dots, K$. Otosautokovarianssimatriisien yhteisdiagonalisoinnissa etsitään ortogonaalinen matriisi $\hat{\Gamma}$, joka minimoi

$$\sum_{k=1}^K \text{off}(\Gamma^T \hat{\Sigma}_{\tau_k}(\mathbf{z}_t) \Gamma).$$

Yllä merkintä $\text{off}(M)$ tarkoittaa matriisin M ei-diagonaalialkioiden neliöiden summaa. Minimi löydetään esimerkiksi yleistetyllä Jacobin menetelmällä. Yksityiskohtainen algoritmi löytyy julkaisusta Belouchrani et al. (1997, s. 441).

2.5.1 SOBI-algoritmi

Oletetaan jälleen, että havaitsemme signaalit \mathbf{x}_t , missä $t = 1, 2, \dots, n$. Lähdesignaalit voidaan estimoida SOBI-menetelmällä seuraavasti:

1. Lasketaan havaintojen \mathbf{x}_t otoskovarianssimatriisin ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T = \hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T.$$

2. Valkaistaan havaintoaineisto

$$\mathbf{z}_t = \hat{V}(\mathbf{x}_t - \bar{\mathbf{x}}), \text{ missä } \hat{V} = \hat{\Lambda}^{-1/2} \hat{\Gamma}^T.$$

3. Lasketaan valkaistuille havainnoille symmetrisoidut otosautokovarianssimatriisit viiveillä $\tau_1, \tau_2, \dots, \tau_K$

$$\tilde{\Sigma}_{\tau_k}(\mathbf{z}_t) = \frac{1}{2} [\hat{\Sigma}_{\tau_k}(\mathbf{z}_t) + \hat{\Sigma}_{\tau_k}(\mathbf{z}_t)^T].$$

4. Suoritetaan yhteisdiagonalisointi eli etsitään ortogonaalinen matriisi $\hat{\Gamma}_z$, joka minimoi

$$\sum_{k=1}^K \text{off}(\Gamma_z^T \hat{\Sigma}_{\tau_k}(\mathbf{z}_t) \Gamma_z).$$

5. Lasketaan ratkaisumatriisi

$$\hat{W} = \hat{\Gamma}_z^T \hat{V}$$

ja lähdesignaalit

$$\hat{\mathbf{s}}_t = \hat{W}(\mathbf{x}_t - \bar{\mathbf{x}}).$$

2.6 Robusti SOBI-menetelmä

AMUSE ja SOBI perustuvat klassisten otoskovarianssi- ja autokovarianssimatriisien käyttöön. Nämä estimaattorit eivät kuitenkaan ole robusteja eli ne reagoivat herkästi poikkeaviin havaintoihin. Tämän vuoksi lähdesignaalien erottelu ei onnistu näillä menetelmillä, jos havaitut signaalit sisältävät poikkeavia havaintoja.

Robusti SOBI-algoritmi (Nordhausen et al., 2011b) saadaan yksinkertaisesti korvaamalla algoritmissa käytetyt kovarianssimatriisit vastaavilla robusteilla estimaattoreilla. Robusteilta kovarianssiestimaattoreilta vaaditaan, että ne toteuttavat luvun 2.3 oletukset. Valkaisuun käytettävän estimaattorin täytyy siis olla affiinisti ekvivariantti ja rotatointiin käytettävän estimaattorin ortogonaalisesti ekvivariantti.

Keskistyksessä ja valkaisussa voidaan käyttää esimerkiksi Hettmansperger-Randles-estimaattoreita (Hettmansperger & Randles, 2002) $\bar{\mathbf{x}}_{HR}$ ja $\hat{\Sigma}_{HR}(\mathbf{x}_t)$. Merkitään nyt $\mathbf{r}_t = \hat{\Sigma}_{HR}^{-1/2}(\mathbf{x}_t - \bar{\mathbf{x}}_{HR})$, missä $\hat{\Sigma}_{HR}^{-1/2}$ on symmetrinen matriisi. Tällöin HR-estimaattorit toteuttavat seuraavat M-estimointiyhtälöt

$$\frac{1}{n} \sum_{t=1}^n \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} = \mathbf{0} \quad \text{ja} \quad \frac{1}{n} \sum_{t=1}^n \frac{\mathbf{r}_t \mathbf{r}_t^T}{\|\mathbf{r}_t\|^2} = \frac{1}{p} I_p.$$

Rotatoinnissa käytetty otosautokovarianssimatriisi voidaan korvata vastaavalla robustilla matriisilla

$$\hat{\Sigma}_{\tau}^R(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n \left[\frac{\mathbf{x}_t}{\|\mathbf{x}_t\|} \frac{\mathbf{x}_{t+\tau}^T}{\|\mathbf{x}_{t+\tau}\|} \right].$$

Huomaa, että hajontamatriisi Σ_{HR} toteuttaa ehdon (B1) eli se on affiinisti ekvivalentti (Hettmansperger & Randles, 2002). Lisäksi helposti nähdään, että matriisi Σ_{τ}^R on ortogonaalisesti ekvivalentti eli oletus (B2) on voimassa. Ehto (B3) toteutuu, jos signaalit ovat symmetrisiä nollan suhteen. Edellä mainittujen estimaattoreiden robustisuus perustuu siihen, että havainnot jaetaan normillaan, jolloin poikkeavan suuret havainnot eivät pääse vaikuttamaan estimaatteihin. Estimaattoreiden heikkoutena on se, etteivät ne huomioi havaintojen etäisyyttä havaintojoukon keskipisteestä. Tämä voi joissain tapauksissa johtaa tehottomiin estimaattoreihin (Tyler, 1987).

3 Riippumattomien komponenttien analyysi (ICA)

3.1 ICA-malli

Riippumattomien komponenttien analyysiä (Independent Component Analysis) (Hyvärinen et al., 2001) voidaan pitää BSS-menetelmien erikoistapauksena. ICA-menetelmät olettavat, että lähdesignaalit ovat riippumattomia satunnaismuuttujia, joka on korreloimattomuutta vahvempi oletus. Malli on edelleen muotoa

$$\mathbf{x}_t = A\mathbf{s}_t,$$

missä $\mathbf{x}_t = [x_{1t}, x_{2t}, \dots, x_{pt}]^T$ on p -ulotteinen sekoitetut signaalit sisältävä vektori ja $\mathbf{s}_t = [s_{1t}, s_{2t}, \dots, s_{pt}]^T$ on p -ulotteinen lähdesignaalien vektori. Matriisi A on $p \times p$ -sekoitusmatriisi ja t on havainnon indeksi. Tehdään nyt seuraavat ICA-mallia koskevat oletukset:

- (C1) Sekoitusmatriisi A on täysiasteinen.
- (C2) Lähdesignaalit \mathbf{s}_t ovat riippumattomia.

Kuten edellä lähdesignaalit saadaan estimoitua skaalaa ja permutaatiota vaille.

3.2 Hajontamatriisit

ICA-ongelma voidaan ratkaista käyttämällä kahta hajontamatriisiä $\Sigma_1(\mathbf{x}_t)$ ja $\Sigma_2(\mathbf{x}_t)$, joista ensimmäistä käytetään signaalien valkaisuun ja jälkimmäistä signaalien rotointiin. Käytettävien hajontamatriisien täytyy toteuttaa seuraavat ehdot:

- (D1) Hajontamatriisi $\Sigma_1(\mathbf{x}_t)$ on affiinisti ekvivalentti.
- (D2) Hajontamatriisi $\Sigma_2(\mathbf{x}_t)$ on ortogonaalisesti ekvivalentti.
- (D3) Jos vektorin \mathbf{x}_t komponentit ovat riippumattomia, niin $\Sigma_2(\mathbf{x}_t)$ on diagonaalinen.

Tarkastellaan seuraavassa luvussa, millaisia hajontamatriiseja FOBI-menetelmässä käytetään ja kuinka lähdesignaalit voidaan estimoida niiden avulla. Luvussa 3.4 kerrotaan, kuinka kyseinen menetelmä saadaan robustifioitua käyttämällä robusteja hajontamatriiseja.

3.3 FOBI-menetelmä

Koska lähdesignaalien ei oleteta sisältävän aikariippuvuutta, autokovarianssimatriisien käyttäminen ICA-ongelman ratkaisemiseksi ei ole mielekäästä. FOBI-menetelmä (Fourth-Order Blind Identification) käyttääkin hyödyksi havaintojen neljänsiin momentteihin perustuvaa informaatiota BSS-ongelman ratkaisemiseksi (Cardoso, 1989). Lähdesignaalit löydetään jälleen kaksivaiheisen algoritmin avulla. Ensimmäisessä vaiheessa signaalit valkaistaan käyttäen kovarianssimatriisia ja toisessa vaiheessa ne rotatoidaan käyttäen neljänsien momenttien matriisia.

3.3.1 Valkaisu

Valkaisu tapahtuu kuten luvussa 2.4.1 eli lasketaan ominaisarvohajotelma

$$\Sigma(\mathbf{x}_t) = E[\mathbf{x}_t \mathbf{x}_t^T] = \Gamma \Lambda \Gamma^T$$

ja määritellään

$$\mathbf{z}_t = V \mathbf{x}_t, \text{ missä } V = \Lambda^{-1/2} \Gamma^T.$$

Valkaisun jälkeen riittää, että signaalit rotatoidaan.

3.3.2 Rotatointi

Rotaatiomatriisi löydetään diagonalisoimalla neljänsien momenttien matriisi. Neljänsien momenttien matriisi määritellään seuraavasti

$$\Sigma(\mathbf{z}_t) = E[|\mathbf{z}_t|^2 \mathbf{z}_t \mathbf{z}_t^T],$$

missä $|\mathbf{z}_t|^2 = \mathbf{z}_t^T \mathbf{z}_t$.

Nyt koska yhtälön (2.1) nojalla $\tilde{A} = VA$ on ortogonaalinen, niin $|\mathbf{z}_t|^2 = |\tilde{A} \mathbf{s}_t|^2 = |\mathbf{s}_t|^2$. Lisäksi pätee

$$E[|\mathbf{s}_t|^2 \mathbf{s}_t \mathbf{s}_t^T] = \text{diag}(E[s_{it}^4] + p - 1). \quad (3.1)$$

Yllä oleva tulos seuraa siitä, että lähdesignaalit ovat riippumattomia, jolloin matriisin $E[|\mathbf{s}_t|^2 \mathbf{s}_t \mathbf{s}_t^T]$ ei-diagonaali-alkiot ovat nollia, ja i 's diagonaali-alkio on

$$E[s_{it}^2 (s_{it}^2 + \dots + s_{pt}^2)] = E[s_{it}^4] + (p - 1) E[s_{it}^2] E[s_{jt}^2] = E[s_{it}^4] + p - 1,$$

sillä lähdesignaalien varianssit oletettiin ykkösiksi. Tällöin

$$\Sigma(\mathbf{z}_t) = E[|\mathbf{z}_t|^2 \mathbf{z}_t \mathbf{z}_t^T] = E[|\tilde{A} \mathbf{s}_t|^2 \tilde{A} \mathbf{s}_t (\tilde{A} \mathbf{s}_t)^T] = \tilde{A} E[|\mathbf{s}_t|^2 \mathbf{s}_t \mathbf{s}_t^T] \tilde{A}^T.$$

Koska yhtälön (3.1) nojalla $E[||\mathbf{s}_t||^2 \mathbf{s}_t \mathbf{s}_t^T]$ on diagonaalinen, sekoitusmatriisi ja lähdesignaalit saadaan estimoitua ominaisarvohajotelman

$$\Sigma(\mathbf{z}_t) = \Gamma_z \Lambda_z \Gamma_z^T \quad (3.2)$$

avulla. Tällöin ratkaisumatriisi on

$$W = \Gamma_z^T V,$$

ja

$$\mathbf{y}_t = W \mathbf{x}_t$$

sisältää lähdesignaalit skaalausta ja permutaatiota vaille.

Huomaa, että matriisin (3.1) diagonaalielementit ovat lähdesignaalien huipukkuuksia vastaavia tunnuslukuja. FOBI-menetelmän löytämät riippumattomat komponentit ovat siis järjestyksessä huipukkuuden mukaan siten, että ensimmäisellä komponentilla on suurin huipukkuuden arvo. Vastaavaa ideaa käytetään myös FastICA-menetelmässä, jossa komponentit löydetään maksimoimalla niiden epänormaalisuus käyttämällä mittarina esimerkiksi huipukkuutta (Hyvärinen et al., 2001).

Komponentit saadaan eroteltua vain, jos matriisin (3.1) diagonaalielementit ovat erisuuria. Tällöin ominaisarvohajotelman (3.2) ominaisvektorit ovat merkkiä ja järjestystä vaille yksikäsitteiset ja p riippumatonta komponenttia pystytään erottelemaan. Käytännössä tämä tarkoittaa sitä, että lähdesignaalien neljännet momentit eivät saa olla yhtäsuuria. Erityisesti samoinjakautuneita lähdesignaaleja ei saada eroteltua. Sovelluksissa menetelmän toimivuutta pystytään jälleen tutkimaan tarkastelemalla neljänsien momenttien matriisin ominaisarvoja. Huomaa, että myös FOBI-menetelmä on invariantti affiinille muunnokselle (todistus, ks. liite A).

3.3.3 FOBI-algoritmi

Oletetaan jälleen, että havaitsemme signaalit \mathbf{x}_t , missä $t = 1, 2, \dots, n$. Lähdesignaalit löydetään FOBI-menetelmällä seuraavasti:

1. Lasketaan havaintojen \mathbf{x}_t otoskovarianssimatriisin ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T = \hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T.$$

2. Valkaistaan havaintoaineisto

$$\mathbf{z}_t = \hat{V}(\mathbf{x}_t - \bar{\mathbf{x}}), \text{ missä } \hat{V} = \hat{\Lambda}^{-1/2} \hat{\Gamma}^T.$$

3. Lasketaan valkaistuille havainnoille neljänsien momenttien matriisi

$$\hat{\Sigma}(\mathbf{z}_t) = \frac{1}{n} \sum_{t=1}^n [||\mathbf{z}_t||^2 \mathbf{z}_t \mathbf{z}_t^T],$$

$$\text{missä } ||\mathbf{z}_t||^2 = \mathbf{z}_t^T \mathbf{z}_t.$$

4. Lasketaan ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{z}_t) = \hat{\Gamma}_z \hat{\Lambda}_z \hat{\Gamma}_z^T.$$

5. Lasketaan ratkaisumatriisi

$$\hat{W} = \hat{\Gamma}_z^T \hat{V}$$

ja lähdesignaalit

$$\hat{\mathbf{s}}_t = \hat{W}(\mathbf{x}_t - \bar{\mathbf{x}}).$$

3.4 Robusti versio FOBI-menetelmästä

Jos aineisto sisältää poikkeavia havaintoja, FOBI-algoritmi ei toimi, sillä käytetyt hajontamatriisit eivät ole robusteja. Erityisen herkkä poikkeaville havainnoille on rotatointiin käytetty neljänsiin momentteihin perustuva matriisi, sillä se painottaa kaukana olevia havaintoja.

Robusti versio FOBIsta (Oja et al., 2006) saadaan korvaamalla valkaisu- ja rotatoinnissa käytettävät estimaattorit vastaavilla robusteilla estimaattoreilla. Robusteilta hajontamatriiseilta vaaditaan, että ne toteuttavat luvun 3.2 ominaisuudet eli valkaisu- ja rotatoinnissa käytettävän hajontamatriisin tulee olla affinisti ekvivariantti ja rotatoinnissa käytettävän hajontamatriisin ortogonaalisesti ekvivariantti. Lisäksi vaaditaan, että rotatointiin käytettävällä hajontamatriisilla on riippumattomuusominaisuus eli se toteuttaa oletuksen (D3).

Tässä tutkielmassa FOBI-menetelmä robustifioidaan käyttämällä valkaisuun affinisti ekvivarianttia Hettmansperger-Randles-estimaattoria kuten luvussa 2.6. Neljänsien momenttien matriisi korvataan havaintojen parittaisista

erotuksista muodostetulla Kendallin tau -matriisilla (Visuri et al., 2000)

$$\hat{\Sigma}_K(\mathbf{x}_i) = \binom{n}{2}^{-1} \sum_{i=1}^n \sum_{i < j}^n \left[\frac{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T}{\|\mathbf{x}_i - \mathbf{x}_j\|^2} \right].$$

Yllä oleva Kendallin tau -matriisi on robusti, koska poikkeavan suurten havaintojen vaikutus estimaatteihin eliminoiduu, kun havainnot normeerataan. Matriisi lasketaan käyttäen parittaisia erotuksia, jolloin sillä on vaadittu riippumattomuusominaisuus (todistus, ks. liite A). Lisäksi on helppo nähdä, että Σ_K on ortogonaalisesti ekvivariantti eli se toteuttaa myös ehdon (D2).

3.5 Usean hajontamatriisin yhteisdiagonalisointi

Kuten luvussa 3.3 todettiin, FOBI-menetelmällä ei saada lähdesignaaleja eroteltua, jos niiden neljännet momentit ovat yhtäsuuria. Tämä ongelma voidaan välttää, kun käytetään vastaavaa ideaa kuin SOBI-menetelmässä, eli yhteisdiagonalisoidaan useita eri hajontamatriiseja (Nordhausen et al., 2011a). Käytettävien hajontamatriisien täytyy toteuttaa luvun 3.2 oletukset (D2) ja (D3), eli niiden tulee olla ortogonaalisesti ekvivariantteja ja niillä on oltava riippumattomuusominaisuus. Käytännön ongelmiin tätä menetelmää on kuitenkin vielä tällä hetkellä vaikea soveltaa, sillä toimiakseen se vaatii useita laskennallisesti nopeita hajontamatriiseja, joilla on riippumattomuusominaisuus.

4 Kohinaa sisältävä BSS-malli

Usein on realistista olettaa, että emme havaitse suoraan lähdesignaalien lineaarikombinaatioita, vaan havaitut signaalit sisältävät myös kohinaa. Kohina voi olla peräisin esimerkiksi mittauslaitteista. Tällöin luvussa 3 esitetyt algoritmit eivät pysty erottelamaan lähdesignaaleja. Laajennetaan nyt BSS-mallia tilanteeseen, jossa sekoitetut signaalit sisältävät kohinaa. Tällöin BSS-malli on muotoa

$$\mathbf{x}_t = A\mathbf{s}_t + \boldsymbol{\nu}_t,$$

missä $\mathbf{x}_t = [x_{1t}, x_{2t}, \dots, x_{qt}]^T$ on q -ulotteinen sekoitetut signaalit sisältävä vektori, $\mathbf{s}_t = [s_{1t}, s_{2t}, \dots, s_{pt}]^T$ on p -ulotteinen ($p < q$) lähdesignaalien vektori ja $\boldsymbol{\nu}_t = [\nu_{1t}, \nu_{2t}, \dots, \nu_{qt}]^T$ on q -ulotteinen kohinan sisältävä vektori. Indeksillä t kuvaa ajanhetkeä ja matriisi A on $q \times p$ -sekoitusmatriisi. (Cichocki & Amari, 2002.)

Kuten edellä, tavoitteena on selvittää tuntemattomat lähdesignaalit \mathbf{s}_t havaittujen signaalien \mathbf{x}_t avulla, kun sekoitusmatriisi A on tuntematon. Mallin sisältäessä kohinaa lähdesignaalien ratkaiseminen monimutkaistuu, sillä lähdesignaalien ja kohinan erottelu toisistaan on vaikeaa. Luvun 2.1 yksinkertaisen BSS-mallin oletuksien (A1)-(A3) lisäksi tehdään seuraavat kohinaa koskevat lisäoletukset:

- (A4) Kohina $\boldsymbol{\nu}_t$ ja lähdesignaalit \mathbf{s}_t ovat keskenään riippumattomia.
- (A5) Kohina vektorin komponentit ovat normaalijakautuneita odotusarvolla nolla ja varianssilla σ^2 . Lisäksi komponentit ovat korreloimattomia. Tällöin siis $\boldsymbol{\nu}_t \sim N_q(\mathbf{0}, \sigma^2 I_q)$.
- (A6) Kohina ei sisällä aikariippuvuutta, toisin sanoen $\text{Cov}(\boldsymbol{\nu}_t, \boldsymbol{\nu}_{t+\tau}) = \mathbf{0}$ kaikilla viiveillä $\tau = 1, 2, 3, \dots$

4.1 AMUSE-menetelmä kohinamallille

4.1.1 Valkaisu ja rotatointi

Kohinamallin tapauksessa valkaisu-algoritmit täytyy muokata, jotta kohina saadaan eroteltua ja sen varianssi estimoitua. Sekoitettujen signaalien kovarianssimatriisi on nyt

$$\Sigma(\mathbf{x}_t) = E[\mathbf{x}_t \mathbf{x}_t^T] = A \Sigma(\mathbf{s}_t) A^T + \sigma^2 I_q = A A^T + \sigma^2 I_q.$$

Oletuksen (A1) perusteella matriisin A aste on p , jolloin myös matriisin $A A^T$ aste on p . Tämä tarkoittaa, että matriisilla $A A^T$ on p nollaa suurempaa ominaisarvoa. Lisäksi huomataan, että matriisin $A A^T + \sigma^2 I_q$ ominaisarvot

saadaan lisäämällä matriisiin AA^T ominaisarvoihin σ^2 . Tästä seuraa, että matriisin $\Sigma(\mathbf{x}_t)$ ominaisarvohajotelma on muotoa

$$\Sigma(\mathbf{x}_t) = [\Gamma_s \quad \Gamma_\nu] \begin{bmatrix} \Lambda_s & \mathbf{0} \\ \mathbf{0} & \Lambda_\nu \end{bmatrix} [\Gamma_s \quad \Gamma_\nu]^T,$$

missä $\Lambda_s = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ ja $q \times p$ -matriisi Γ_s sisältävät lähdesignaaleihin ja kohinaan liittyvät ominaisarvot ja ominaisvektorit. Vastaavasti $\Lambda_\nu = \text{diag}(\lambda_{p+1}, \dots, \lambda_q) = \sigma^2 I_{q-p}$ ja $q \times (q-p)$ -matriisi Γ_ν sisältävät pelkästään kohinaan liittyvät ominaisarvot ja ominaisvektorit. Teoreettisesti σ^2 on siis yhtä suuri kuin matriisin $\Sigma(\mathbf{x}_t)$ $(q-p)$ pienintä ominaisarvoa, mikäli oletetaan, että kohinan varianssi on suhteellisen pientä. Tällöin $p \times q$ -valkaisumatriisi V on muotoa

$$V = (\Lambda_s - \sigma^2 I_p)^{-1/2} \Gamma_s^T = \text{diag} \left(\frac{1}{\sqrt{\lambda_1 - \sigma^2}}, \frac{1}{\sqrt{\lambda_2 - \sigma^2}}, \dots, \frac{1}{\sqrt{\lambda_p - \sigma^2}} \right) \Gamma_s^T$$

ja valkaistut signaalit ovat $\mathbf{z}_t = V\mathbf{x}_t$. Huomaa, että valkaistuja signaaleja on p kappaletta ($p < q$), eli valkaaisu pienentää havaintoaineiston dimensiota. Nyt

$$\begin{aligned} AA^T &= \Sigma(\mathbf{x}_t) - \sigma^2 I_q \\ &= [\Gamma_s \quad \Gamma_\nu] \text{diag}(\lambda_1 - \sigma^2, \lambda_2 - \sigma^2, \dots, \lambda_p - \sigma^2, 0, \dots, 0) [\Gamma_s \quad \Gamma_\nu]^T \\ &= \Gamma_s \text{diag}(\lambda_1 - \sigma^2, \lambda_2 - \sigma^2, \dots, \lambda_p - \sigma^2) \Gamma_s^T \end{aligned}$$

ja siten

$$VAA^T V^T = I,$$

eli VA on ortogonaalinen. Merkitään nyt $\tilde{A} = VA$ ja $\boldsymbol{\eta}_t = V\boldsymbol{\nu}_t$. Tällöin valkaistuille havainnoille pätee

$$\mathbf{z}_t = V\mathbf{x}_t = V(A\mathbf{s}_t + \boldsymbol{\nu}_t) = \tilde{A}\mathbf{s}_t + \boldsymbol{\eta}_t,$$

missä \tilde{A} on ortogonaalinen. Oletuksen (A4) nojalla lähdesignaalit ja kohina ovat korreloimattomia ja oletuksen (A6) nojalla kohina ei sisällä aikariippuvuutta, joten

$$\Sigma_\tau(\mathbf{z}_t) = \tilde{A} \Sigma_\tau(\mathbf{s}_t) \tilde{A}^T.$$

Nyt $\Sigma_\tau(\mathbf{s}_t)$ on oletuksen (A3) nojalla diagonaalinen, joten sekoitusmatriisi saadaan estimoitua ominaisarvohajotelmasta

$$\Sigma_\tau(\mathbf{z}_t) = \Gamma_z \Lambda_z \Gamma_z^T.$$

Tällöin ratkaisumatriisi on

$$W = \Gamma_z^T V$$

ja

$$\mathbf{y}_t = W \mathbf{x}_t = W(A \mathbf{s}_t + \boldsymbol{\nu}_t)$$

sisältää lähdesignaalit merkkiä ja permutaatiota vaille. Estimoidut lähdesignaalit sisältävät vielä kohinaa. Tämän pienentämiseksi kehitettyjä menetelmiä on esitelty kirjassa Hyvärinen et al. (2001, luku 15.5).

4.1.2 AMUSE-algoritmi kohinamallille

Oletetaan, että havaitsemme signaalit $\mathbf{x}_t, t = 1, 2, \dots, n$, jotka sisältävät kohinaa, ja merkitään havaittujen signaalien otoskovarianssimatriisin $\hat{\Sigma}(\mathbf{x}_t)$ ominaisarvohajotelmaa $\hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T$. Tällöin kohinan varianssi voidaan estimoida $(q - p)$ pienimmän ominaisarvon keskiarvona. Estimoitavien lähdesignaalien määrä p pyritään valitsemaan siten, että

$$\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_p > \hat{\lambda}_{p+1} \approx \dots \approx \hat{\lambda}_q.$$

Valinta tehdään joko heuristisesti tai siihen voidaan käyttää esimerkiksi AIC- (Akaike's Information Criterion) tai MDL-kriteeriä (Minimum Description Length). Jos oletetaan, että $\mathbf{x}_t, t = 1, 2, \dots, n$, ovat riippumattomia ja normaalijakautuneita satunnaisvektoreita, niin voidaan osoittaa, että estimaatti lähdesignaalien määrälle p löydetään minimoimalla

$$AIC(p) = -2n(q - p) \log r(p) + 2p(2q - p)$$

tai

$$MDL(p) = -n(q - p) \log r(p) + 0.5p(2q - p) \log n,$$

missä

$$r(p) = \frac{(\hat{\lambda}_{p+1} \hat{\lambda}_{p+2} \cdots \hat{\lambda}_q)^{\frac{1}{q-p}}}{\frac{1}{q-p} (\hat{\lambda}_{p+1} + \hat{\lambda}_{p+2} + \dots + \hat{\lambda}_q)}.$$

Lähdesignaalien lukumäärän estimaatti \hat{p} valitaan siten, että se minimoi joko AIC- tai MDL-kriteerin (ks. Cichocki & Amari, 2002, luku 3.2.3).

Tiivistettynä AMUSE-algoritmi kohinamallille on seuraava:

1. Lasketaan havaintojen \mathbf{x}_t otoskovarianssimatriisin ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T = \hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T.$$

2. Jos lähdesignaalien määrä p ei ole tunnettu, estimoidaan se merkitsevien ominaisarvojen määrästä kuten yllä.
3. Kohinan varianssi $\hat{\sigma}^2$ estimoidaan $(q-p)$ pienimmän ominaisarvon keskiarvona.
4. Valkaistaan havaintoaineisto $\mathbf{z}_t = \hat{V}(\mathbf{x}_t - \bar{\mathbf{x}})$, missä

$$\hat{V} = \text{diag} \left(\frac{1}{\sqrt{\hat{\lambda}_1 - \hat{\sigma}^2}}, \frac{1}{\sqrt{\hat{\lambda}_2 - \hat{\sigma}^2}}, \dots, \frac{1}{\sqrt{\hat{\lambda}_p - \hat{\sigma}^2}} \right) \hat{\Gamma}_s^T,$$

ja $\hat{\lambda}_1, \dots, \hat{\lambda}_p$ ovat p suurinta ominaisarvoa ja $\hat{\Gamma}_s^T$ sisältää vastaavat ominaisvektorit.

5. Lasketaan valkaistuille havainnoille symmetrisoitu otosautokovarianssimatriisi viiveellä τ

$$\tilde{\Sigma}_\tau(\mathbf{z}_t) = \frac{1}{2}[\hat{\Sigma}_\tau(\mathbf{z}_t) + \hat{\Sigma}_\tau(\mathbf{z}_t)^T].$$

6. Lasketaan ominaisarvohajotelma

$$\tilde{\Sigma}_\tau(\mathbf{z}_t) = \hat{\Gamma}_z \hat{\Lambda}_z \hat{\Gamma}_z^T.$$

7. Lasketaan ratkaisumatriisi

$$\hat{W} = \hat{\Gamma}_z^T \hat{V}$$

ja lähdesignaalit

$$\hat{\mathbf{s}}_t = \hat{W}(\mathbf{x}_t - \bar{\mathbf{x}}).$$

4.2 EFOBI-menetelmä

Myös FOBI-menetelmä voidaan yleistää tilanteeseen, jossa sekoitetut signaalit sisältävät kohinaa. Tästä menetelmästä käytetään kirjallisuudessa nimitystä EFOBI (Extended Fourth-Order Blind Identification) (Tong et al., 1991). Koska FOBI lukeutuu ICA-menetelmiin, lähdesignaalit \mathbf{s}_t oletetaan nyt riippumattomiksi. Kohinalle $\boldsymbol{\nu}_t$ pätee edelleen luvun 4 oletukset (A4) ja (A5), eli $\boldsymbol{\nu}_t$ ja \mathbf{s}_t ovat riippumattomia ja $\boldsymbol{\nu}_t \sim N_q(\mathbf{0}, \sigma^2 I_q)$.

4.2.1 Valkaisu ja rotatointi

Valkaisu suoritetaan kuten AMUSE-metelmän kohdalla luvussa 4.1.1. Rotatointiin käytetään autokovarianssimatriisin sijaan neljänsiin momentteihin perustuvaa matriisia. Merkitään nyt

$$\Sigma(\mathbf{x}_t) = E[|\mathbf{x}_t|^2 \mathbf{x}_t \mathbf{x}_t^T]$$

ja

$$\Delta\Sigma(\mathbf{x}_t) = \text{diag}(\delta_1, \delta_2, \dots, \delta_p),$$

missä

$$\delta_i = \frac{(p+4)\sigma^2}{\lambda_i - \sigma^2} + \frac{\sigma^2}{\lambda_i - \sigma^2} \left(\sum_{k=1}^p \frac{\lambda_k}{\lambda_k - \sigma^2} + \frac{2\sigma^2}{\lambda_i - \sigma^2} \right).$$

Tällöin voidaan osoittaa (ks. Tong et al., 1991, luku 5.1), että valkaistuille havainnoille $\mathbf{z}_t = V\mathbf{x}_t$ pätee

$$\Sigma(\mathbf{z}_t) - \Delta\Sigma\mathbf{z}_t = \tilde{A}\text{diag}(\kappa_1, \kappa_2, \dots, \kappa_p)\tilde{A}^T,$$

missä $\tilde{A} = VA$ on ortogonaalinen matriisi, ja $\kappa_i = E[s_{it}^4]$. Sekoitusmatriisi saadaan siten estimoitua ominaisarvohajotelmasta

$$\Sigma(\mathbf{z}_t) - \Delta\Sigma\mathbf{z}_t = \Gamma_z \Lambda_z \Gamma_z^T,$$

jolloin ratkaisumatriisi

$$W = \Gamma_z^T V.$$

4.2.2 EFOBI-algoritmi

Oletetaan, että havaitsemme signaalit $\mathbf{x}_t, t = 1, 2, \dots, n$, jotka sisältävät kohinaa. Tällöin lähdesignaali voidaan estimoida EFOBI-menetelmällä seuraavasti:

1. Lasketaan havaintojen \mathbf{x}_t otoskovarianssimatriisin ominaisarvohajotelmasta

$$\hat{\Sigma}(\mathbf{x}_t) = \frac{1}{n} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T = \hat{\Gamma} \hat{\Lambda} \hat{\Gamma}^T.$$

2. Jos lähdesignaalien määrä p ei ole tunnettu, estimoidaan se merkitsevien ominaisarvojen määrästä, kuten luvussa 4.1.2.

3. Kohinan varianssi $\hat{\sigma}^2$ estimoidaan $(q-p)$ pienimmän ominaisarvon keskiarvona.

4. Valkaistaan havaintoaineisto $\mathbf{z}_t = \hat{V}(\mathbf{x}_t - \bar{\mathbf{x}})$, missä

$$\hat{V} = \text{diag} \left(\frac{1}{\sqrt{\hat{\lambda}_1 - \hat{\sigma}^2}}, \frac{1}{\sqrt{\hat{\lambda}_2 - \hat{\sigma}^2}}, \dots, \frac{1}{\sqrt{\hat{\lambda}_p - \hat{\sigma}^2}} \right) \hat{\Gamma}_s^T,$$

ja $\hat{\lambda}_1, \dots, \hat{\lambda}_p$ ovat p suurinta ominaisarvoa ja $\hat{\Gamma}_s^T$ sisältää vastaavat ominaisvektorit.

5. Lasketaan valkaistuille havainnoille neljänsien momenttien matriisi

$$\hat{\Sigma}(\mathbf{z}_t) = \frac{1}{n} \sum_{t=1}^n [||\mathbf{z}_t||^2 \mathbf{z}_t \mathbf{z}_t^T]$$

ja

$$\Delta \hat{\Sigma}(\mathbf{z}_t) = \text{diag}(\hat{\delta}_1, \hat{\delta}_2, \dots, \hat{\delta}_p),$$

missä

$$\hat{\delta}_i = \frac{(p+4)\hat{\sigma}^2}{\hat{\lambda}_i - \hat{\sigma}^2} + \frac{\hat{\sigma}^2}{\hat{\lambda}_i - \hat{\sigma}^2} \left(\sum_{k=1}^p \frac{\hat{\lambda}_i}{\hat{\lambda}_k - \hat{\sigma}^2} + \frac{2\hat{\sigma}^2}{\hat{\lambda}_i - \hat{\sigma}^2} \right).$$

6. Lasketaan ominaisarvohajotelma

$$\hat{\Sigma}(\mathbf{z}_t) - \Delta \hat{\Sigma} \mathbf{z}_t = \hat{\Gamma}_z \hat{\Lambda}_z \hat{\Gamma}_z^T.$$

7. Lasketaan ratkaisumatriisi

$$\hat{W} = \hat{\Gamma}_z^T \hat{V}$$

ja lähdesignaalit

$$\hat{\mathbf{s}}_t = \hat{W}(\mathbf{x}_t - \bar{\mathbf{x}}).$$

5 Menetelmien vertailu

BSS-algoritmien toimivuutta voidaan vertailla simulointikokeiden avulla, jolloin todelliset lähdesignaali sekä sekoitusmatriisi tunnetaan. Erilaisten indeksien avulla voidaan tutkia, kuinka lähdesignaalien estimointi on onnistunut. Menetelmien vertailua hankaloittaa se, että BSS-ongelmaa ei voida ratkaista yksikäsitteisesti. Tarkastellaan seuraavaksi kahta yleisesti käytössä olevaa indeksä, jotka vertaavat estimoitua ratkaisumatriisia W todelliseen sekoitusmatriisiin A .

5.1 Vertailuindeksejä

Olkoon nyt W jollakin algoritmilla laskettu BSS-ongelman ratkaisumatriisi ja oletetaan, että todellinen sekoitusmatriisi A tunnetaan. Määritellään

$$G = WA,$$

ja merkitään matriisin G alkioita g_{ij} . Jos lähdesignaali saadaan eroteltua täydellisesti, niin

$$G = PD,$$

missä P on permutaatiomatriisi ja D diagonaalimatriisi. G on siis permutaatiota ja merkkejä vaille identtinen matriisi, sillä lähdesignaalien merkkejä ja järjestystä ei voida määrittää yksikäsitteisesti. Koska lähdesignaalien estimointi havainnoista ei koskaan onnistu täydellisesti, niin $G \approx PD$. Tätä poikkeamaa voidaan mitata esimerkiksi ISR-indeksillä (Interference-to-Signal Ratio), joka määritellään (Douglas, 2007) seuraavasti

$$\text{ISR}(G) = \sqrt{\frac{1}{p} \left(\sum_{i=1}^p \sum_{j=1}^p \delta_{ij}^2 \right)} - 1,$$

missä

$$\delta_{ij} = \frac{|g_{ij}|}{\max_j |g_{ij}|}.$$

Jos algoritmi erottelee lähdesignaali täydellisesti, niin matriisin G jokaisella rivillä on yksi nollasta eroava alkio. Tällöin $\sum_{i=1}^p \sum_{j=1}^p \delta_{ij}^2 = p$ ja $\text{ISR}(G) = 0$. Mitä huonommin algoritmi toimii, sitä suuremmaksi ISR-indeksin arvo kasvaa.

Toinen paljon käytetty vertailuindeksi on ns. Amari error (Amari et al., 1996), joka määritellään

$$\text{AE}(G) = \sum_{i=1}^p \left(\sum_{j=1}^p \delta_{ij} - 1 \right) + \sum_{j=1}^p \left(\sum_{i=1}^p \rho_{ij} - 1 \right),$$

missä

$$\rho_{ij} = \frac{|g_{ij}|}{\max_i |g_{ij}|} \quad \text{ja } \delta_{ij} \text{ kuten edellä.}$$

Kun lähdesignaalit saadaan eroteltua täydellisesti $\text{AE}(G) = 0$. Erottelun huonontuessa indeksin arvo kasvaa.

5.2 Simulointiasetelmat

Edellisissä luvuissa esiteltyjä ratkaisumenetelmiä vertailtiin simulointikokeilla. Simulointeihin käytettiin R-ohjelmistoa (R Development Core Team, 2011). Lähdesignaaleja generoitiin kolme kappaletta s_{1t}, s_{2t}, s_{3t} . Otoskooksi valittiin $n = 2000$. Lähdesignaalit sekoitettiin satunnaisella 3×3 -matriisilla A , jonka alkioit generoitiin $Tas(0, 1)$ -jakaumasta. Sekoitetuista havainnoista estimoitiin lähdesignaalit ja ratkaisumatriisi kullakin menetelmällä. Tämän jälkeen kullekin ratkaisumatriisille laskettiin ISR-indeksi. AMUSE-algoritmin autokovarianssimatriisi laskettiin viiveellä yksi. SOBI-algoritmeissa käytettiin autokovarianssimatriiseja viiveillä $1, 2, \dots, 10$. Simuloinneissa käytettiin 1000 toistoa. Huomaa, että jokaisessa toistossa voitiin käyttää samaa sekoitusmatriisia, sillä ratkaisumenetelmät ovat affiinisti invariantteja (ks. liite A). Käytetyllä sekoitusmatriisilla ei siis ole vaikutusta tuloksiin. Lähdesignaalit generoitiin seuraavasti:

- (I) Lähdesignaalit olivat riippumattomia satunnaismuuttujia siten, että $s_{1t} \sim N(0, 1)$, $s_{2t} \sim \text{Exp}(1)$ ja $s_{3t} \sim Tas(-\sqrt{3}, \sqrt{3})$.
- (II) Lähdesignaalit olivat AR(1)-prosesseja parametrien arvoilla $\phi_1 = 0.8$, $\phi_2 = 0.5$ ja $\phi_3 = 0.3$, ja virhetermit olivat standardinormaalijakautuneita, toisin sanoen

$$\begin{aligned} s_{1t} &= 0.8s_{1(t-1)} + \epsilon_{1t} \\ s_{2t} &= 0.5s_{2(t-1)} + \epsilon_{2t} \\ s_{3t} &= 0.3s_{3(t-1)} + \epsilon_{3t}, \end{aligned}$$

missä $\epsilon_{it} \sim N(0, 1)$, $i = 1, 2, 3$.

- (III) Suoritettiin kuten simulointi (II), mutta nyt AR(1)-prosessien virhetermit generoitiin paksuhäntäisestä $t(3)$ -jakaumasta normaalijakouman sijaan.

Menetelmien robustisuutta tutkittiin lisäämällä sekoitettuihin signaaleihin poikkeavia havaintoja seuraavasti:

- (IV) Lähdesignaalit generoitiin normaali-, eksponentti- ja tasajakaumasta kuten simuloinnissa (I), mutta nyt jokaisen sekoitetun signaalin yksi arvo korvattiin poikkeavalla havainnolla. Poikkeava havainto muodostettiin kertomalla signaalin suurin arvo kymmenellä ja arpomalla tälle satunnainen merkki.
- (V) Lähdesignaalit generoitiin kuten asetelmassa (II), mutta jälleen jokaisen sekoitetun signaalin yksi arvo korvattiin poikkeavalla havainnolla kuten simuloinnissa (IV).

Otoskoon vaikutusta menetelmien toimivuuteen ja laskenta-aikoihin tutkittiin seuraavasti:

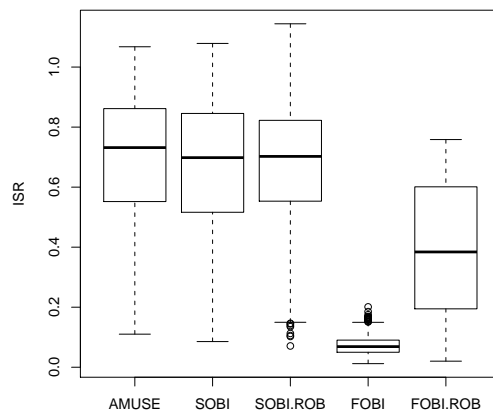
- (VI) Lähdesignaalit olivat riippumattomia satunnaismuuttujia kuten asetelmassa (I). Simuloinneissa käytettiin 100 toistoa, joista laskettiin ISR-indeksin mediaani, kun otoskoko n oli 50, 100, 200, 500, 1000 ja 1500.
- (VII) Suoritettiin kuten edellinen mutta nyt lähdesignaalit olivat AR(1)-prosesseja simuloinnin (II) tapaan.
- (VIII) Simuloinneissa käytettiin 10 toistoa, joista laskettiin kunkin algoritmin käyttämän laskenta-ajan mediaani, kun otoskoko n oli 500, 1000, 2000, 5000 ja 10000.

Lopuksi perinteisiä AMUSE ja FOBI menetelmiä vertailtiin vastaaviin luvun 4 kohinamenetelmiin. Kuten edellä, lähdesignaaleja generoitiin kolme kappaletta s_{1t} , s_{2t} , s_{3t} ja otoskooksi valittiin 2000. Lähdesignaalit sekoitettiin nyt satunnaisella 5×3 -matriisilla A , jonka alkiot generoitiin $Tas(0, 1)$ -jakaumasta. Sekoitettuihin signaaleihin lisättiin kohinaa siten, että $\boldsymbol{\nu}_t \sim N_5(\mathbf{0}, \sigma^2 I_5)$. Kohinan varianssin suuruuden vaikutusta algoritmien toimintaan tutkittiin seuraavilla simulointiasetelmilla:

- (IX) Lähdesignaalit olivat riippumattomia satunnaismuuttujia kuten asetelmassa (I). Simuloinneissa käytettiin 1000 toistoa, joista laskettiin ISR-indeksin mediaani, kun kohinan varianssi oli 0, 0.02, 0.04, \dots , 0.3.
- (X) Suoritettiin kuten edellinen mutta nyt lähdesignaalit olivat AR(1)-prosesseja simuloinnin (II) tapaan.

5.3 Tulokset

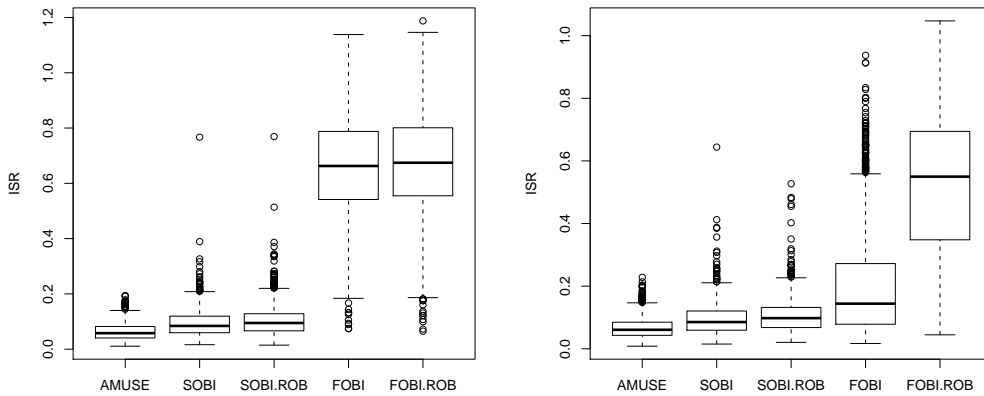
Kuvaan 5.1 on piirretty boxplot-kuviot simuloinnin (I) tuottamista ISR-indeksien arvoista menetelmittäin. Nähdään, että tavallinen FOBI-menetelmä erottelee riippumattomat lähdesignaalit parhaiten, sillä sen ISR-indeksin mediaani on lähimpänä nollaa. Vaihtelu toistojen välillä on myös pientä. AMUSE, SOBI ja robusti SOBI eivät toimi, sillä menetelmät olettavat, että lähdesignaalit sisältävät aikariippuvuutta. Kun lähdesignaalit eivät sisällä poikkeavia havaintoja, robusti versio FOBI-algoritmista ei toimi yhtä hyvin kuin tavallinen FOBI. Tämä voi johtua siitä, että valkaisuissa käytetty Hettmansperger-Randles-estimaattori on otoskovarianssimatriisia tehottomampi, kun aineisto ei sisällä poikkeavia havaintoja.



Kuva 5.1: ISR-indeksit menetelmittäin, kun lähdesignaalit ovat normaali-, tasa- ja eksponenttijakaumista.

Simulointien (II) ja (III) tulokset on esitelty kuvassa 5.2. Lähdesignaalien ollessa aikasarjoja AMUSE, SOBI ja robusti SOBI toimivat hyvin. SOBI-menetelmän useamman autokovarianssimatriisin käytöstä ei näyttäisi olevan hyötyä AR(1)-prosessien tapauksessa, vaan parhaimmat tulokset saadaan AMUSE-menetelmällä, joka käyttää vain yhtä autokovarianssimatriisia viiveellä yksi. Tämä voi johtua siitä, että AR(1)-prosessien tapauksessa autokorrelaatio vähenee eksponentiaalisesti viiveen kasvaessa, joten viiveellä yksi laskettu autokovarianssimatriisi sisältää kaiken oleellisen informaation lähdesignaaleista eikä useamman viiveen käyttäminen tuo uutta tietoa. FOBI-menetelmällä ei saada AR(1)-prosesseja eroteltua, jos prosessien virhetermit ovat normaalijakautuneita. ICA-ongelman ratkaisu ei ole mahdollista,

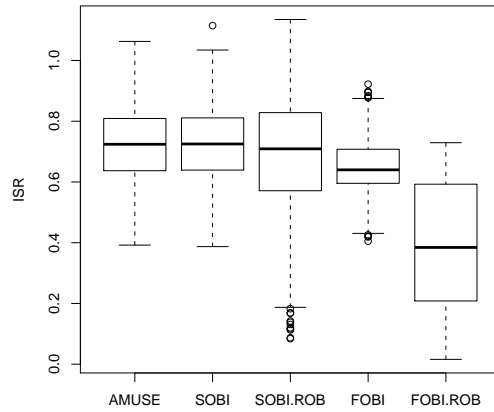
jos kaikki signaalit ovat normaalijakautuneita. Tällöin valkaistut signaalit ovat jakautuneet täsmälleen samoin kuin niiden mikä tahansa ortogonaalinen muunnos eikä rotaatiomatriisia saada näin ollen ratkaistua. Simuloinnissa (III) AR(1)-prosessien virhetermit generoitiin $t(3)$ -jakaumasta normaali-jakauman sijaan. Kuvasta 5.2 nähdään, että FOBI-menetelmä toimii tällöin huomattavasti paremmin kuin normaalijakautuneiden virheiden tapauksessa, vaikkakin vaihtelu ISR-indeksissä on suurta. Muiden menetelmien toimintaan virheiden jakauman muutos ei vaikuttanut.



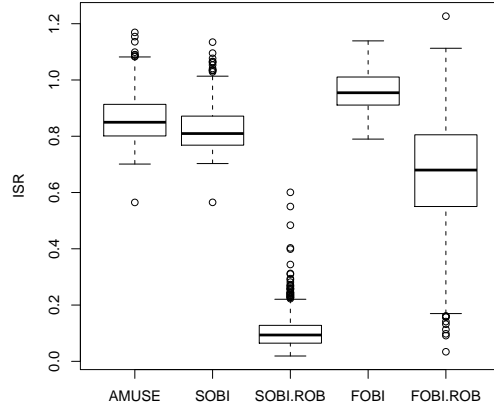
Kuva 5.2: ISR-indeksit menetelmittain, kun lähdesignaalit ovat AR(1)-prosesseja. Vasemmanpuoleisessa kuvassa prosessien virhetermit ovat normaalijakaumasta ja oikealla $t(3)$ -jakaumasta.

Simuloinneissa (IV) ja (V) tutkittiin menetelmien robustisuutta. Kuvassa 5.3 esitellään tulokset simulointikokeesta (IV), jossa riippumattomiin lähdesignaaleihin lisättiin sekoituksen jälkeen poikkeavia havaintoja. Kun tätä kuvaa verrataan simuloinnin (I) tuottamaan kuvaan, huomataan että perinteinen FOBI-menetelmä on herkkä poikkeaville havainnoille. Robustiin FOBI-menetelmään poikkeavat havainnot eivät sen sijaan vaikuta. Sama ilmiö nähdään myös aikariippuvien lähdesignaalien tilanteessa. Kun verrataan kuvia 5.2 ja 5.4, huomataan että AMUSE ja SOBI eivät enää toimi, kun aineistoon on lisätty poikkeavia havaintoja. Robustiin SOBI-menetelmään poikkeavat havainnot eivät vaikuta, ja signaalien erottelu onnistuu hyvin myös tässä tilanteessa.

Kuvassa 5.5 on esitetty keskimääräinen ISR-indeksin arvo otoskoon funktiona simulointiasetelmien (VI) ja (VII) osalta. Riippumattomien komponenttien tapauksessa FOBI:n ja robustin FOBI:n toiminta paranee, kun otos-

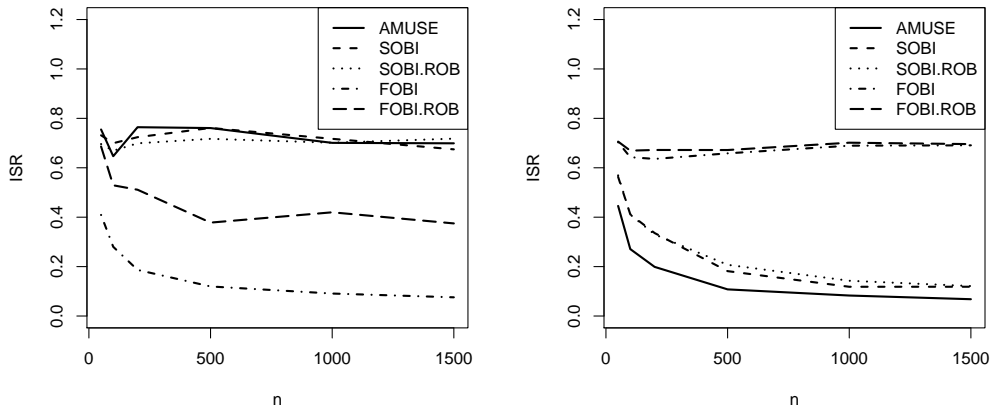


Kuva 5.3: ISR-indeksit menetelmittäin, kun lähdesignaalit ovat normaali-, tasa- ja eksponenttijakaumista, ja sekoitettuihin signaaleihin on lisätty poikkeavia havaintoja.



Kuva 5.4: ISR-indeksit menetelmittäin, kun lähdesignaalit ovat AR(1)-prosesseja, ja sekoitettuihin signaaleihin on lisätty poikkeavia havaintoja.

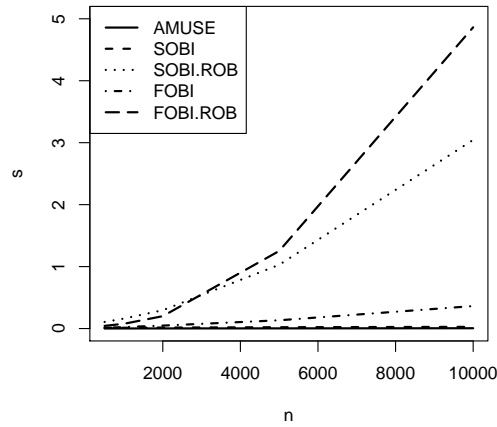
koko kasvaa. AMUSE ja SOBI eivät toimi vaikka otoskoko kasvatetaan. Lähdesignaalien ollessa aikasarjoja AMUSE ja SOBI toimivat sitä paremmin, mitä suurempi aineisto on. FOBI-menetelmä ei toimi aikariippuvalle aineistolle, vaikka otoskoko olisi kuinka suuri tahansa.



Kuva 5.5: Otoksoon vaikutus algoritmien toimivuuteen, kun lähdesignaalit ovat riippumattomia satunnaismuuttujia (vasemmalla) ja AR(1)-prosesseja (oikealla).

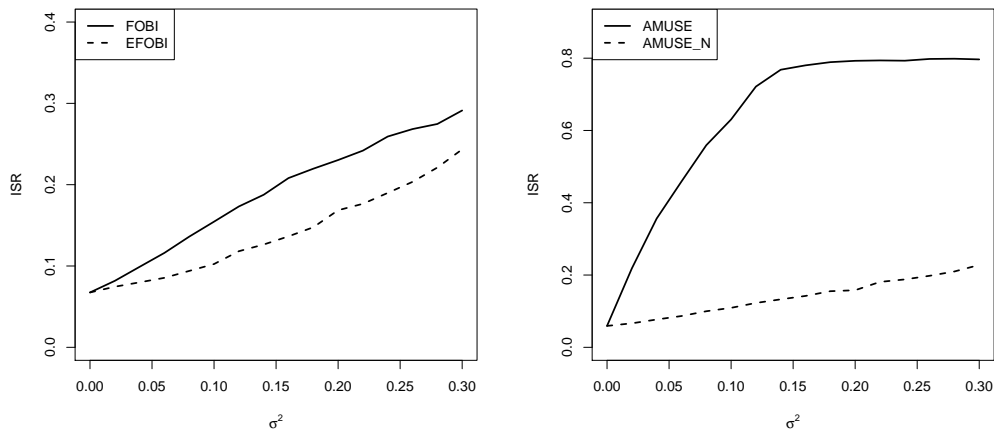
Simulointikokeessa (VII) tutkittiin, kuinka otoskoon kasvu vaikuttaa algoritmien laskenta-aikoihin. Kuvassa 5.6 on esitelty menetelmien keskimääräiset laskenta-ajat otoskoon funktiona. Huomataan, että AMUSEn ja SOBIN laskenta-aika ei juurikaan kasva, vaikka otoskoko kasvaa. FOBI-menetelmänkin laskenta-aika kasvaa vain hieman otoskoon kasvaessa. Sen sijaan robustit algoritmit hidastuvat merkittävästi, kun otoskoko kasvatetaan. Erityisen hidas on robusti-FOBI, sillä rotatointiin käytetty Kendallin tau -matriisi käyttää havaintojen parittaisia erotuksia, jolloin otoskoon kasvaessa parien lukumäärä kasvaa todella nopeasti ja hidastaa siten laskentaa.

Lopuksi simuloinneissa (IX) ja (X) tutkittiin kohinan vaikutusta vertailemalla perinteistä FOBI-menetelmää EFOBIin sekä AMUSE-menetelmää sen laajennukseen kohinamallille (merk. AMUSE_N). Simuloinneissa lähdesignaalien lukumäärä oletettiin tunnetuksi. Kuvasta 5.7 nähdään, että havaittujen signaalien sisältäessä kohinaa, kohinan huomioonottavat menetelmät toimivat paremmin kuin vastaavat perinteiset menetelmät. Mitä suurempi kohinan varianssi on, sitä vaikeampaa lähdesignaalien erottelu kuitenkin on. Kun verrataan vierekkäisiä kuvia, huomataan että aikariippuville lähdesignaaleille tarkoitettujen AMUSE ja AMUSE_N menetelmien välinen ero on selvästi suurempi kuin FOBI-menetelmien. Nähdään, että kohinan huomioonottavan AMUSE-menetelmän keskimääräinen ISR-indeksi on selvästi pienempi kuin perinteisen AMUSE-menetelmän. Kohinan varianssin kasvaessa perinteisen AMUSE-menetelmän toimivuus heikkenee merkittävästi. Mutta jos algorit-



Kuva 5.6: Otoksoon vaikutus algoritmien laskenta-aikoihin

mi huomioi kohinan, toimivuus heikkenee vain hieman. FOBI-menetelmien välinen ero ei ole yhtä suuri, mutta kohinamalleille tarkoitettu EFOBI toimii silti paremmin kuin tavallinen FOBI.

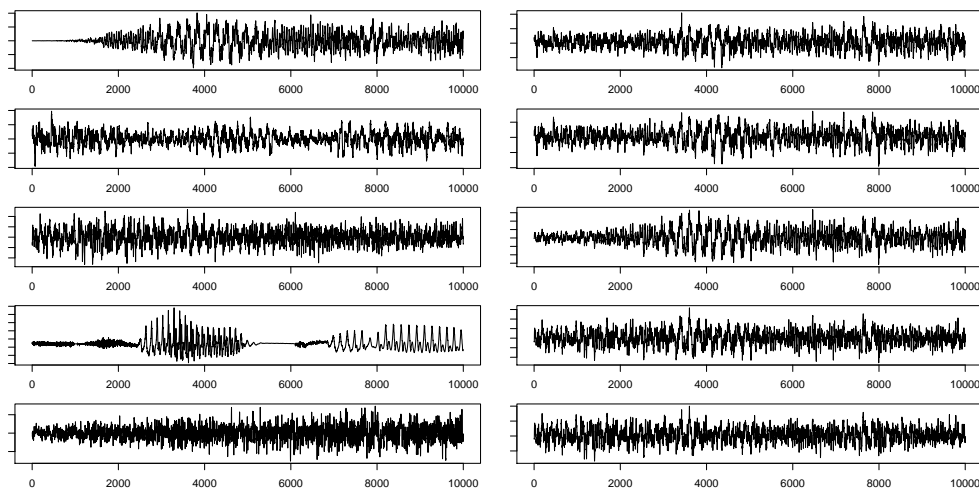


Kuva 5.7: Kohinan varianssin vaikutus algoritmien toimivuuteen, kun lähesignaalit ovat riippumattomia satunnaismuuttujia (vasemmalla) ja AR(1)-prosesseja (oikealla).

6 Sovelluksia

6.1 Äänisignaalien erottelu

Sokeita signaalinkäsittelyn menetelmiä voidaan käyttää erottelemaan keskenään sekoittuneista äänisignaaleista yksittäiset äänisignaalit. Tässä esimerkiksi tarkastellaan viittä todellista äänisignaalia, jotka on esitetty kuvassa 6.1. Signaalit on poimittu Internet-sivuilta ICA '99 Synthetic Benchmarks, <http://sound.media.mit.edu/ica-bench>. Jokaisessa signaalissa on 10 000 havaintoa. Taulukossa 1 on signaalien autokorrelaatiomatriisi viiveellä yksi. Taulukosta nähdään, että viiveellä yksi signaaleissa on selvästi aikariippuvuutta, eli peräkkäiset havainnot riippuvat toisistaan. Signaalien väliset ristikorrelaatiot ovat sen sijaan likimain nollia.



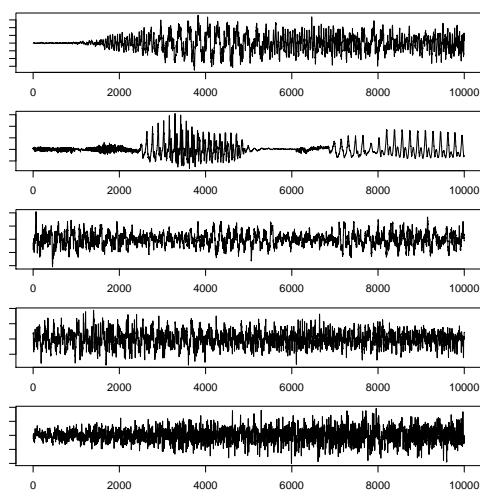
Kuva 6.1: Alkuperäiset äänisignaalit vasemmalla ja sekoitetut signaalit oikealla

	s_1	s_2	s_3	s_4	s_5
s_1	0.983	0.057	0.019	-0.003	-0.044
s_2	0.062	0.973	-0.046	-0.021	0.006
s_3	0.022	-0.057	0.962	0.012	0.014
s_4	-0.009	-0.013	0.008	0.976	0.037
s_5	-0.045	-0.003	0.016	0.038	0.935

Taulukko 1: Alkuperäisten äänisignaalien autokorrelaatiomatriisi viiveellä yksi

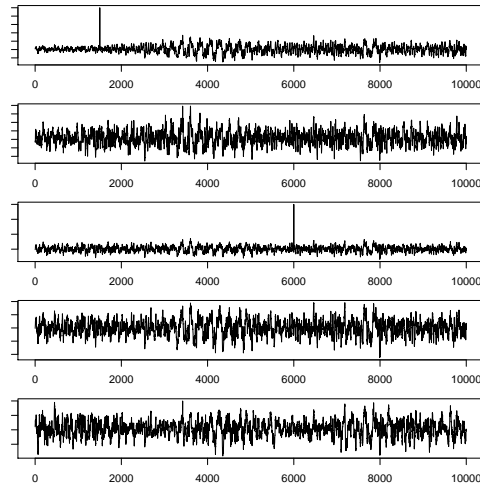
Menetelmien vertailemiseksi alkuperäiset äänisignaalit sekoitettiin keino-
tekoisesti satunnaisella 5×5 -matriisilla, jonka alkiot generoitiin $Tas(0, 1)$ -
jakaumasta. Sekoitetuista signaaleista (kuva 6.1) pyrittiin erottelemaan al-
kuperäiset äänisignaalit. Menetelmien onnistumista tutkittiin vertailemalla
estimoitujen äänisignaalien kuvia alkuperäisten äänisignaalien kuviin sekä
laskemalla ISR-indeksin arvot.

AMUSE-menetelmä onnistui erottelemaan alkuperäiset äänisignaalit par-
haiten ($ISR = 0.14$). AMUSEn erottelemat signaalit löytyvät kuvasta 6.2.
Kun näitä verrataan alkuperäisiin äänisignaaleihin, huomataan että kaikki
alkuperäiset äänisignaalit saatiin estimoitua. Signaalien järjestys on vaihtu-
nut, kuten myös joidenkin signaalien merkki. Tämä johtuu siitä, että sig-
naalien järjestystä ja merkkiä ei saada ratkaistua yksikäsitteisesti (ks. luku
2.2). Myös SOBI-menetelmää kokeiltiin, mutta useamman autokovarianssi-
matriisin käytöstä ei ollut hyötyä. ISR-indeksin arvo kasvoi sitä suuremmak-
si mitä enemmän viiveitä otettiin käyttöön. Myöskään FOBI-menetelmä ei
toiminut hyvin, sillä vain kolme alkuperäistä äänisignaalia pystyttiin erotte-
lemaan selvästi ($ISR = 0.38$). Tämä johtunee siitä, että äänisignaaleissa on
aikariippuvuutta ja osa signaaleista on lisäksi likimain normaalijakautuneita.



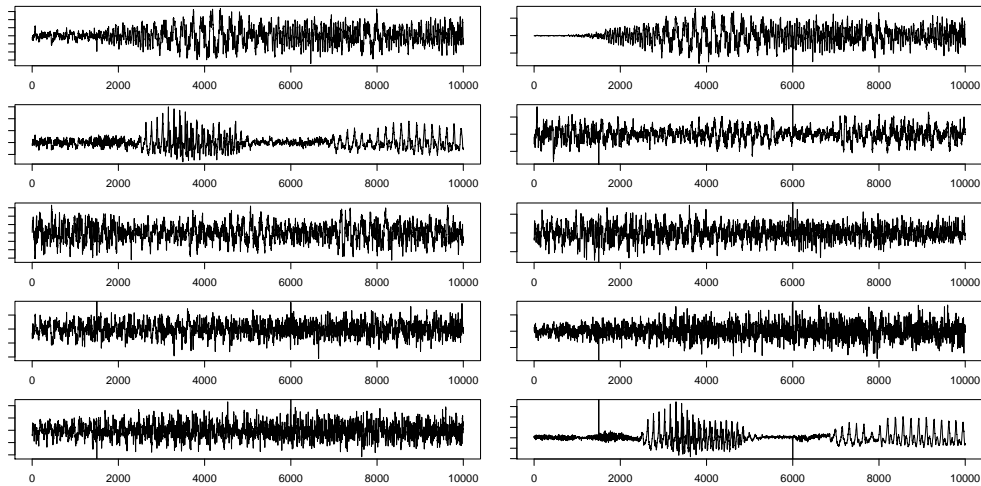
Kuva 6.2: AMUSE-menetelmän erottelemat äänisignaalit

Menetelmien robustisuutta tutkittiin lisäämällä kahteen sekoitettuun sig-
naaliin poikkeavan suuret havainnot, jotka näkyvät yksittäisinä piikkeinä ku-
vassa 6.3. Kuvassa 6.4 on sekä perinteisen AMUSE-menetelmän että robustin
AMUSE-menetelmän erottelemat äänisignaalit. Kuvista huomataan, että pe-
rinteinen AMUSE-menetelmä ei pysty erottelemaan äänisignaaleja aineiston
manipuloinnin jälkeen. Myös ISR-indeksin arvo 0.62 kertoo, ettei erottelu



Kuva 6.3: Kaksi poikkeavan suurta havaintoa lisättyä sekoitettuihin äänisignaaleihin

ole onnistunut. Sen sijaan robusti AMUSE-menetelmä ($ISR = 0.16$) kykenee erottelamaan kaikki alkuperäiset äänisignaalit lukuunottamatta piikkejä poikkeavien havaintojen lisäämiskohdissa. Myös robusteja SOBI- ja FOBI-menetelmiä kokeiltiin, mutta ne eivät toimineet yhtä hyvin kuin robusti AMUSE.



Kuva 6.4: AMUSE-menetelmän erottelamat äänisignaalit vasemmalla ja robustin AMUSEn erottelamat äänisignaalit oikealla, kun aineistoon on lisätty poikkeavia havaintoja.

6.2 Aivosähkökäyrien analysointi

Aivosähkökäyrä eli EEG (elektroenkefalografia) kuvaa aivojen sähköistä aktiivisuutta, joka syntyy signaalien kulkiessa hermosolujen eli neuronien välillä. Aivosähkökäyrää mitataan usemmiten päänahan pinnalle asetettavilla elektrodeilla. Jotta syntyvä sähkökenttä pystytään havaitsemaan, tuhansien hermosolujen täytyy aktivoitua samanaikaisesti. Aktiivisuus on siis peräisin aivojen sisältä, mutta mittaukset tehdään kallon pinnalta. Tämän vuoksi signaaleja ei saada mitattua tarkasti vaan jokainen havaittu signaali on summa tuntemattomista lähdesignaaleista. Sokeiden signaalinkäsittelyn menetelmien avulla voidaan saada selville nämä tuntemattomat lähdesignaalit ja paikantaa ne. (Cichocki & Amari, 2002; Lang et al., 1994.)

Usein aivojen toimintaa selvitetään tapahtumasidonnaisten herätevasteiden (Event Related Potential, ERP) avulla. ERP:llä tarkoitetaan jonkin ulkoisen ärsyksen aiheuttamaa hetkellistä muutosta aivojen aktiivisuudessa. Ongelmana on, että ärsyksen aiheuttamat jännitemuutokset eivät yleensä ole erotettavissa EEG-käyrästä, sillä ne ovat muuhun aivotoimintaan nähden heikkoja. Signaalinkäsittelyn menetelmien avulla saadaan herätevasteet paremmin eroteltua EEG:stä. (Lang et al., 1994.)

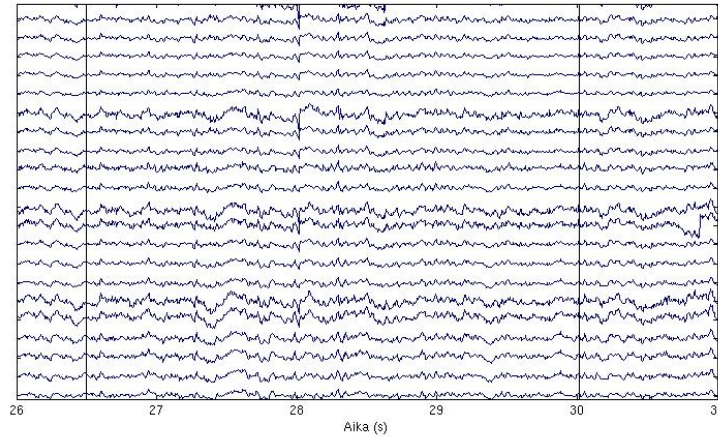
Aivosähkökäyrän tulkintaa haittaavat usein erilaiset häiriöt eli artefaktit. Artefaktit voivat olla peräisin esimerkiksi silmänliikkeistä, lihasjännityksestä tai sydämen sykkeestä (Lang et al., 1994). Sokeita signaalinkäsittelyn menetelmiä voidaan käyttää erottelemaan ja poistamaan artefakteja.

6.2.1 Aineisto

Käytettävä aineisto on osa laajempaa tutkimusta, jossa on pyritty selvittämään lukihäiriön eli dysleksian mahdollisia aivoperäisiä syitä. Koehenkilöitä tutkittiin kaikkiaan 26, joista osalla oli diagnosoitu dysleksia ja osalla ei. EEG nauhoituksiin käytettiin 128 elektrodia. Nauhoituksen aikana koehenkilöille soitettiin satunnaisessa järjestyksessä viittä erilaista ääntä siten, että jokaista ääntä toistettiin 150 kertaa. Nauhoitus kesti noin puoli tuntia. Tutkimuksen yhtenä mielenkinnon kohteena on selvittää, eroaako kuuloärsykkeiden käsittely henkilöillä, joilla on lukihäiriö verrattuna niihin, joilla ei ole lukihäiriötä.

Tässä tutkielmassa BSS-menetelmiä kokeillaan yhden koehenkilön aivosähkökäyrä mittauksiin. Tarkoituksena on tutkia, kuinka BSS-menetelmät toimivat todellisessa havaintoaineistossa ja saadaanko BSS:n avulla aineistosta eroteltua yksittäisiä mielenkiintoisia komponentteja. Analysoitavan aineiston näytteenottotaajuus oli 200 Hz, mikä tarkoittaa 200 havaintoa sekunnissa. Kaikkiaan havaintoja oli siis noin puoli miljoonaa jokaista 128 kanavaa

kohti. Kuvassa 6.5 on pieni pätkä käytetystä aivosähkökäyrä aineistosta. Kuva sisältää viiden sekunnin pätkän 20 EEG-signaalista. Pystysuorat viivat kertovat soitettujen äänien ajankohdat.

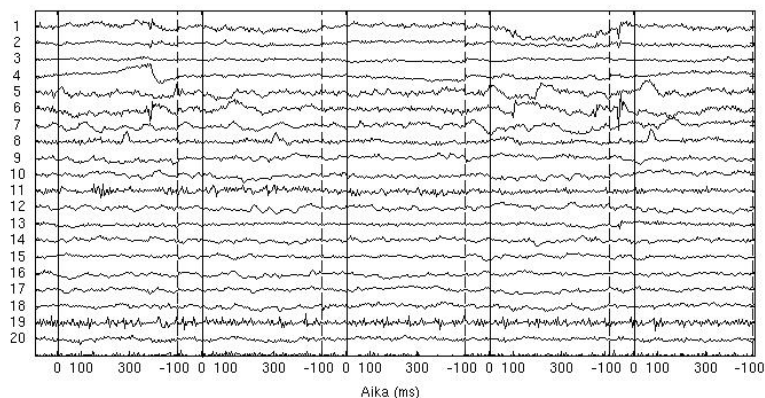


Kuva 6.5: Pieni osa alkuperäisestä EEG-aineistosta

6.2.2 Tulokset

Aivosähkökäyrien analysointiin käytettiin Matlab-ohjelmistoon pohjautuvaa EEGLAB-ilmaisohjelmaa (Delorme & Makeig, 2004). Aluksi aineistoon tehtiin ylipäästösuodatus. Ylipäästösuodin vaimentaa signaalien hitaita taajuuksia eli poistaa hitaasti muuttuvia trendejä. Tässä aineistossa käytettiin 1 Hz ylipäästösuodinta, jotta esimerkiksi koehenkilön hikoiluun liittyvät artefaktit saatiin poistettua. Muita aineiston sisältämiä artefakteja poistettiin hylkäämällä häiriötä sisältäviä EEG-pätkiä. BSS-analysointiin käytettiin SOBI-algoritmia.

Kuvassa 6.6 on 20 ensimmäistä SOBI-algoritmin erottelemaa lähdesignaalia. Lähdesignaaleja tutkittiin vain soitettujen ääniärsykkeiden läheisyydessä. Jokaisen ääniärsyksen kohdalta eristettiin aikaikkuna, jonka pituus oli 0.6 sekuntia alkaen 0.1 sekuntia ennen ärsykettä ja loppuen 0.5 sekuntia ärsyksen jälkeen. Ajatuksena on, että kaikki ärsykeeseen liittyvä aivojen toiminnan muutos ilmenee tämän ajanjakson aikana. Kuvassa yhtenäinen pystysuora viiva osoittaa äänen soittoaikajankohdan ja katkoviiva erottaa aikaikkunat toisistaan.

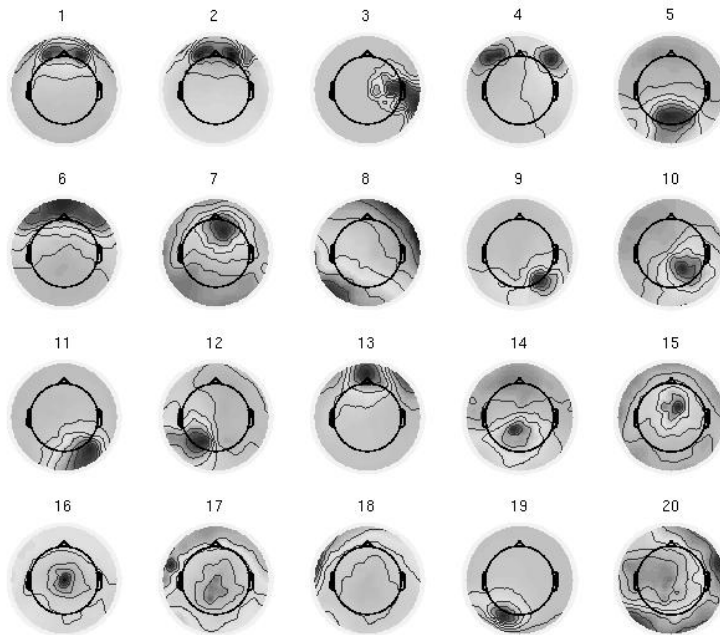


Kuva 6.6: 20 ensimmäistä lähdesignaalia

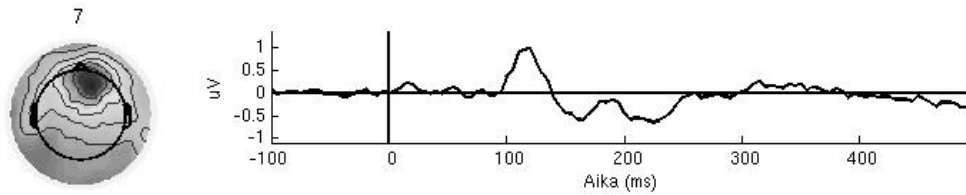
Kuvan 6.6 perusteella lähdesignaalien tulkinta on vaikeaa. Tarkempaa tietoa signaalien lähteistä saadaan topografiakuvista (ks. kuva 6.7). Topografiakuvien värityksen perusteella voidaan tutkia, mistä kohtaa aivoja lähdesignaali on peräisin. Väritys kertoo, kuinka alkuperäiset mittauskanavat painottuvat kyseisessä lähdesignaalissa. Se on siis muodostettu ratkaisumatriisin W rivien perusteella. Tummat harmaansävyt tarkoittavat, että sen alueen kanavat saavat suuren painon ja haaleamman sävyn alueet pienemmän painon. Kuvan avulla voidaan karkeasti päätellä, onko lähdesignaali artefaktia vai aivoperäistä toimintaa. Kuvasta nähdään, että lähdesignaalit 1, 2 ja 4 ovat silmänliikkeistä johtuvaa artefaktia. Kahdesta ensimmäisestä lähdesignaalista toinen on todennäköisesti silmänräpäytyksistä ja toinen silmän pystysuuntaisista liikkeistä peräisin olevaa artefaktia. Neljäs lähdesignaali on puolestaan silmän sivuttaissuuntaisista liikkeistä johtuvaa häiriötä.

Ärsykkeen käsittelyyn liittyvät lähdesignaalit saatiin paremmin esiin lasquemalla ärsykkeen toistojen yli keskiarvoistetut herätevasteet. Paljastui, että aivotoiminnasta lähtöisin olevia lähdesignaaleja, jotka liittyvät ärsykkeen käsittelyyn, ovat ainakin signaalit numero 7 ja 16. Kuvissa 6.8 ja 6.9 on esitetty kyseisten signaalien topografiat sekä keskiarvoistetut ERP-kuvat. Molemmissa lähdesignaaleissa on selvästi tapahtunut muutos noin sata millisekuntia ärsykkeen jälkeen.

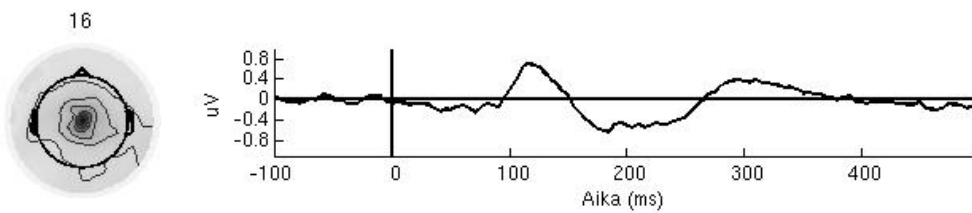
BSS-menetelmä osoittautui siis toimivaksi. Aineistosta pystyttiin erottelamaan kiinnostavia aivoperäisiä komponentteja. Myös artefaktien erottelu onnistui, mikä sekin on tärkeää, sillä niiden avulla aineistosta voidaan poistaa häiriötä. BSS-ajot toistettiin myös käyttäen FOBI-algoritmia ja saadut tulokset olivat hyvin samankaltaisia. Tässä esimerkissä FOBI-menetelmä siis toimi, vaikka signaaleissa oli aikariippuvuutta. Luvun 5.3 simulointitulokset



Kuva 6.7: Topografia kuvat 20 ensimmäisestä lähdesignaalista



Kuva 6.8: Lähdesignaalin nro 7 topografia (vasemmalla) ja heräteväste (oikealla)



Kuva 6.9: Lähdesignaalin nro 16 topografia (vasemmalla) ja heräteväste (oikealla)

antoivatkin viitteitä siitä, että myös FOBI-menetelmä voi toimia aikariippuville lähdesignaaleille, jos ne eivät ole normaalijakautuneita. Robusteja menetelmiä ei käytetty, sillä tällä hetkellä robustien kovarianssimatriisien laskenta ei onnistu näin suuren aineiston tapauksessa. Tutkimusongelman tarkempi selvittely edellyttäisi aivoperäisten komponenttien lisätarkasteluja ja muiden koehenkilöiden EEG-mittausten analysointia.

7 Yhteenveto

Sokeaa signaalinkäsittelyä sovelletaan tilanteissa, joissa päämääränä on selvittää tuntemattomat lähdesignaalit, kun havaintomme koostuu ainoastaan niiden sekoituksista. Tässä tutkielmassa esiteltiin teoriaa siitä, kuinka nämä tuntemattomat lähdesignaalit saadaan ratkaistua. Tutkielmassa rajoituttiin ratkaisualgoritmeihin, jotka perustuvat kahden hajontamatriisin diagonalisointiin. Perinteisistä menetelmistä käsiteltiin AMUSE-, SOBI- ja FOBI-menetelmät. AMUSE- ja SOBI-menetelmät olettavat, että lähdesignaalit ovat keskenään korreloimattomia aikasarjoja. FOBI-menetelmä puolestaan olettaa, että lähdesignaalit ovat riippumattomia. Perinteisille algoritmeille esitettiin myös robustit vaihtoehdot, jotka saatiin käyttämällä robusteja hajontamatriiseja. Robusteja BSS- ja ICA-menetelmiä on käsitelty kirjallisuudessa todella vähän ja alan tutkimus on vasta alkuvaiheessa.

Tutkielman tarkoituksena oli vertailla eri ratkaisumenetelmien toimivuutta. Menetelmien teoreettiseen vertailuun käytettiin simulointikokeita. Simuloinneissa luotiin aluksi keinotekoiset lähdesignaalit, joista sekoituksen jälkeä estimoitiin lähdesignaalit kullakin menetelmällä. Estimoinnin onnistumista mitattiin ISR-indeksillä.

Simulointitulokset olivat odotetun kaltaisia. Kun lähdesignaalit olivat riippumattomia satunnaisuuttuja ilman minkäänlaista aikarakennetta, FOBI-menetelmä estimoiti lähdesignaalit parhaiten. Jos lähdesignaalit sen sijaan sisälsivät aikariippuvuutta, AMUSE- ja SOBI-menetelmät toimivat paremmin kuin FOBI. Tulokset kuitenkin antoivat viitteitä siitä, että myös FOBI-menetelmä voi toimia aikariippuville lähdesignaaleille, jos signaalit eivät ole normaalijakautuneita.

Lisäämällä aineistoon poikkeavia havaintoja osoitettiin, että perinteiset menetelmät eivät ole robusteja, eli ne eivät pysty erottelemaan lähdesignaaleja, jos aineisto sisältää epätavallisen suuria tai pieniä arvoja. Robustit menetelmät sen sijaan toimivat hyvin, kun signaalit sisälsivät poikkeavia havaintoja. Robustien algoritmien heikkoutena on niiden hitaus suurten aineistojen tapauksessa. Käytännön sovelluksissa otoskoko on usein vähintään satojen tuhansien suuruinen. Tällöin robustien hajontamatriisien laskenta on liian työlästä. Uusia estimaattoreita tuleekin kehittää.

Havaitut signaalit sisältävät usein kohinaa. Tutkielmassa esitettiin perinteisille AMUSE- ja FOBI-menetelmille yleistyksyet, joita voidaan käyttää tällaisissa tilanteissa. Simuloinnit osoittivat, että kohinan huomioonottaminen on tärkeää. Perinteisten AMUSE- ja FOBI-algoritmien kyky erotella lähdesignaaleja heikkeni merkittävästi, kun signaalit sisälsivät kohinaa. Kohinan huomioonottavat menetelmät toimivat selvästi paremmin kuin nämä perinteiset menetelmät.

Tutkielman sovellusesimerkeissä sokeita signaalinkäsittelyn menetelmiä käytettiin äänisignaalien erotteluun sekä aivosähkökäyrien analysointiin. Äänisignaalien erottelua kokeiltiin eri menetelmillä. Parhaiten aikariippuvien äänisignaalien erottelu onnistui AMUSE-menetelmällä. Kun aineistoon lisättiin poikkeavia havaintoja, huomattiin että perinteinen AMUSE-menetelmä ei enää toiminut. Sen sijaan robustilla AMUSE-menetelmällä äänisignaalit saatiin eroteltua. Aivosähkökäyrien analysointi osoitti, että sokeiden signaalinkäsittelyn menetelmien avulla EEG-mittauksista voidaan erotella artefakteja sekä paljastaa aivoperäisestä toiminnasta lähtöisin olevia signaaleja, joiden tarkka määrittäminen pään pinnalta tehtävin mittauksin ei olisi muutoin mahdollista.

Ongelman abstraktin luonteen takia on mahdotonta määrittää yhtä oikeaa ratkaisualgoritmia kuhunkin käytännön sovellukseen. Jos on realistista olettaa, että lähdesignaalit sisältävät aikariippuvuutta, luultavasti AMUSE- tai SOBI-menetelmä estimoit lähdesignaalit paremmin kuin FOBI. Jos aikariippuvat signaalit eivät kuitenkaan ole normaalijakautuneita, myös FOBI-menetelmä voi toimia hyvin. Kun lähdesignaaleilla ei ole aikarakennetta, FOBI-menetelmä toimii paremmin kuin AMUSE tai SOBI. Robustien menetelmien käyttäminen on suotavaa mahdollisten poikkeavien havaintojen vuoksi. Myös mahdollinen kohina tulee ottaa huomioon algoritmia valittaessa. Käytännössä aineistoon voidaan soveltaa eri algoritmeja ja vertailla niiden estimoimia lähdesignaaleja toisiinsa. Lähdesignaalien tulkintaan tarvitaan usein sovellusalan asiantuntijuutta.

Viitteet

- Amari, S., Cichocki, A., & Yang, H. (1996). A New Learning Algorithm for Blind Signal Separation. *Advances in Neural Information Processing Systems 8*, 757–763.
- Belouchrani, A., Abed-Meraim, K., Cardoso, J., & Moulines, E. (1997). A Blind Source Separation Technique Using Second-Order Statistics. *IEEE Trans. on Signal Processing*, 45(2):434–444.
- Cardoso, J. (1989). Source separation using higher order moments. *IEEE ICASSP*, 4:2109–2112.
- Cichocki, A. & Amari, S. (2002). *Adaptive Blind Signal and Image Processing*. Wiley. Chichester.
- Delorme, A. & Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134:9–21.
- Douglas, S. C. (2007). Fixed-Point Algorithms for the Blind Separation of Arbitrary Complex-Valued Non-Gaussian Signal Mixtures. *EURASIP J. Advances in Signal Processing*, 2007(1).
- Hettmansperger, T. & Randles, R. (2002). A practical affine equivariant multivariate median. *Biometrika*, 89(4):851–860.
- Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. Wiley. New York.
- Lang, H., Häkkinen, V., Larsen, T., Partanen, J., & Tolonen, U. (1994). *Sähköiset aivomme*. Suomen kliinisen neurofysiologian yhdistys ry.
- Nordhausen, K., Gutch, H., Oja, H., & Theis, F. (2011a). Joint Diagonalization of Several Scatter Matrices for ICA. Submitted.
- Nordhausen, K., Ilmonen, P., Oja, H., & Theis, F. (2011b). On Robust Blind Source Separation. Manuscript.
- Oja, H., Sirkiä, S., & Eriksson, J. (2006). Scatter Matrices and Independent Component Analysis. *Austrian Journal of Statistics*, 35:175–189.
- R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, <http://www.R-project.org>.

- Tong, L., Liu, R., Soon, V., & Huang, Y. (1991). Indeterminacy and Identifiability of Blind Identification. *IEEE Trans. on Circuits and Systems*, 38(5):499–509.
- Tong, L., Soon, V., Huang, Y., & Liu, R. (1990). AMUSE: A New Blind Identification Algorithm. *IEEE ISCAS*, 3:1784–1787.
- Tyler, D. (1987). A Distribution-Free M-Estimator of Multivariate Scatter. *Annals of Statistics*, 15:234–251.
- Visuri, S., Koivunen, V., & Oja, H. (2000). Sign and rank covariance matrices. *Journal of Statistical Planning and Inference*, 91:557–575.

A Todistuksia

Osoitetaan, että AMUSE-menetelmällä estimoidut lähdesignaalit ovat merkkiä ja permutaatiota vaille todelliset lähdesignaalit.

Lause 1. *Oletetaan, että $\mathbf{x}_t = A\mathbf{s}_t$ noudattaa BSS-mallia ja $\Sigma(\mathbf{s}_t) = I_p$ ja $\Sigma_\tau(\mathbf{s}_t) = D$, missä D on diagonaalimatriisi. Olkoon*

$$W = \Gamma_z^T V,$$

missä $V = \Lambda^{-1/2}\Gamma^T$ on valkaisu-matriisi ja Γ_z :n sarakkeina ovat matriisin $\Sigma_\tau(\mathbf{z}_t)$ ominaisvektorit. Silloin

$$W\mathbf{x}_t = PJ\mathbf{s}_t,$$

missä P on jokin permutaatiomatriisi ja J jokin diagonaalimatriisi, jonka diagonaalielementit ovat ± 1 .

Todistus. Merkitään matriisin A singulaariarvohajotelmaa $A = U_1 L U_2^T$, missä U_1 ja U_2 ovat ortogonaalisia matriiseja ja L on diagonaalimatriisi. Silloin $\mathbf{x}_t = U_1 L U_2^T \mathbf{s}_t$ ja $\Sigma(\mathbf{x}_t) = U_1 L U_2^T \Sigma(\mathbf{s}_t) (U_1 L U_2^T)^T = U_1 L^2 U_1^T$. Edelleen

$$\mathbf{z}_t = L^{-1} U_1^T \mathbf{x}_t = L^{-1} U_1^T U_1 L U_2^T \mathbf{s}_t = U_2^T \mathbf{s}_t,$$

ja

$$\Sigma_\tau(\mathbf{z}_t) = U_2^T \Sigma_\tau(\mathbf{s}_t) U_2.$$

Koska matriisin $\Sigma_\tau(\mathbf{z}_t)$ ominaisvektorit Γ_z ovat merkkiä ja permutaatiota vaille yksikäsitteiset, niin

$$W\mathbf{x}_t = \Gamma_z^T V \mathbf{x}_t = \Gamma_z^T \mathbf{z}_t = \Gamma_z^T U_2^T \mathbf{s}_t = PJ\mathbf{s}_t.$$

□

Osoitetaan seuraavaksi, että kahden hajontamatriisin käyttöön perustuvat BSS- ja ICA-menetelmät ovat affiinisti invariantteja. Merkitään valkaisuun käytettävää hajontamatriisia $\Sigma_1(\mathbf{x}_t)$ ja rotatointiin käytettävää hajontamatriisia $\Sigma_2(\mathbf{x}_t)$. Todistusta varten tarvitaan seuraava Lemma.

Lemma 1. *Olkoon $\Sigma_1(\mathbf{x}_t)$ mikä tahansa affiinisti ekvivariantti hajontamatriisi, B jokin ei-singulaarinen $p \times p$ -matriisi ja $\mathbf{x}_t^* = B\mathbf{x}_t$. Merkitään lisäksi signaalien \mathbf{x}_t valkaisu-matriisia V :llä ja vastaavasti signaalien \mathbf{x}_t^* valkaisu-matriisia V^* :llä. Tällöin*

$$V^* = QVB^{-1},$$

missä Q on ortogonaalinen $p \times p$ -matriisi.

Todistus. Merkitään matriisin $\Sigma_1(\mathbf{x}_t)$ ominaisarvohajotelmaa $\Gamma\Lambda\Gamma^T$, ja vastaavasti matriisin $\Sigma_1(\mathbf{x}_t^*)$ ominaisarvohajotelmaa $\Gamma^*\Lambda^*\Gamma^{*T}$. Tällöin luvun 2.4.1 perusteella signaalien \mathbf{x}_t valkaisu-matriisi $V = \Lambda^{-1/2}\Gamma^T$, ja vastaavasti signaalien \mathbf{x}_t^* valkaisu-matriisi $V^* = \Lambda^{*-1/2}\Gamma^{*T}$. Koska $I = B\Gamma\Lambda^{1/2}\Lambda^{-1/2}\Gamma^TB^{-1}$ saadaan

$$\begin{aligned} V^* &= \Lambda^{*-1/2}\Gamma^{*T}B\Gamma\Lambda^{1/2}\Lambda^{-1/2}\Gamma^TB^{-1} \\ &= QVB^{-1}, \end{aligned}$$

missä $Q = \Lambda^{*-1/2}\Gamma^{*T}B\Gamma\Lambda^{1/2}$. Nyt

$$\begin{aligned} QQ^T &= \Lambda^{*-1/2}\Gamma^{*T}B\Gamma\Lambda^{1/2}\Lambda^{1/2}\Gamma^TB^T\Gamma^*\Lambda^{*-1/2} \\ &= \Lambda^{*-1/2}\Gamma^{*T}B\Sigma_1(\mathbf{x}_t)B^T\Gamma^*\Lambda^{*-1/2} \\ &= \Lambda^{*-1/2}\Gamma^{*T}\Sigma_1(\mathbf{x}_t^*)\Gamma^*\Lambda^{*-1/2} \\ &= \Lambda^{*-1/2}\Gamma^{*T}\Gamma^*\Lambda^*\Gamma^{*T}\Gamma^*\Lambda^{*-1/2} = I. \end{aligned}$$

□

Lause 2. *Olkoon $\Sigma_1(\mathbf{x}_t)$ affiniesti ekvivariantti ja $\Sigma_2(\mathbf{x}_t)$ ortogonaalisesti ekvivariantti hajontamatriisi. Tällöin*

$$W^*\mathbf{x}_t^* = W\mathbf{x}_t.$$

Todistus. Lemma 1:stä seuraa, että valkaistuille havainnoille \mathbf{z}_t^* pätee

$$\mathbf{z}_t^* = V^*\mathbf{x}_t^* = QVB^{-1}B\mathbf{x}_t = QV\mathbf{x}_t = Q\mathbf{z}_t.$$

Koska hajontamatriisi $\Sigma_2(\mathbf{x}_t)$ on ortogonaalisesti ekvivariantti, niin

$$\Sigma_2(\mathbf{z}_t^*) = Q\Sigma_2(\mathbf{z}_t)Q^T = Q\Gamma_z\Lambda_z\Gamma_z^TQ^T = (Q\Gamma_z)\Lambda_z(Q\Gamma_z)^T,$$

eli kyseessä on hajontamatriisin $\Sigma_2(\mathbf{z}_t^*)$ ominaisarvohajotelma. Tällöin luvun 2.4.2 perusteella

$$W^*\mathbf{x}_t^* = (Q\Gamma_z)^T\mathbf{z}_t^* = (Q\Gamma_z)^TQ\mathbf{z}_t = \Gamma_z^T\mathbf{z}_t = W\mathbf{x}_t.$$

□

Todistetaan vielä, että Kendallin tau -matriisilla Σ_K on riippumattomuusominaisuus.

Lause 3. *Olkoon vektorin \mathbf{s}_t komponentit riippumattomia. Tällöin $\Sigma_K(\mathbf{s}_t)$ on diagonaalinen.*

Todistus. Olkoot \mathbf{s}_i ja \mathbf{s}_j kaksi riippumatonta kopiota vektorista \mathbf{s}_t . Koska vektorin \mathbf{s}_t komponentit ovat riippumattomia, niin myös vektorin $\mathbf{s}_i - \mathbf{s}_j$ komponentit ovat riippumattomia. Lisäksi $\mathbf{s}_i - \mathbf{s}_j$ on symmetrinen nollan suhteen, sillä $\mathbf{s}_i - \mathbf{s}_j$ on samoinjakautunut kuin $\mathbf{s}_j - \mathbf{s}_i = -(\mathbf{s}_i - \mathbf{s}_j)$. Jos nyt merkitään I_k^- on $p \times p$ -diagonaalimatriisi, jonka k 's diagonaalelementti on -1 muiden ollessa ykkösiä, niin tällöin $\mathbf{s}_i - \mathbf{s}_j$ ja $I_k^-(\mathbf{s}_i - \mathbf{s}_j)$ ovat samoinjakautuneita kaikilla $k = 1, 2, \dots, p$.

Merkitään nyt

$$\Sigma_K(\mathbf{s}_i - \mathbf{s}_j) := \Sigma_K(\mathbf{s}_t) = E \left[\frac{(\mathbf{s}_i - \mathbf{s}_j)(\mathbf{s}_i - \mathbf{s}_j)^T}{\|\mathbf{s}_i - \mathbf{s}_j\|^2} \right].$$

Tällöin

$$\Sigma_K(\mathbf{s}_i - \mathbf{s}_j) = \Sigma_K(I_k^-(\mathbf{s}_i - \mathbf{s}_j)) = I_k^- \Sigma_K(\mathbf{s}_i - \mathbf{s}_j) I_k^- \quad \text{kaikilla } k = 1, 2, \dots, p.$$

Tästä seuraa, että

$$[\Sigma_K(\mathbf{s}_i - \mathbf{s}_j)]_{km} = -[\Sigma_K(\mathbf{s}_i - \mathbf{s}_j)]_{km} \quad \text{kaikilla } k \neq m$$

eli ei-diagonaalelementtien on oltava nollia. □

B R-koodeja

Seuraavassa esimerkkejä tutkielmassa käytetyistä R-koodeista. Robustit algoritmit saadaan muokkaamalla perinteisiä algoritmeja siten, että tavallisten kovarianssiestimaattoreiden tilalla käytetään vastaavia robusteja estimaattoreita. Tarvittavat R-funktiot löytyvät paketista ICSNP.

```
## AMUSE ##

autocov <- function(x,tau){
# laskee symmetrisoidun autokovarianssimatriisin
# x = datamatriisi
# tau = viive
  n <- dim(x)[1]
  S <- (t(x[1:(n-tau),])%*%x[(tau+1):n,])/(n-tau)
  (S + t(S))/2
}

amuse <- function(x,tau){
# estimoi BSS-ongelman ratkaisumatriisin ja
# lähdesignaalit AMUSE-menetelmällä
# x = datamatriisi
# tau = viive, jolla rotatointiin käytettävä
# autokovarianssimatriisi lasketaan

#keskistys
ka <- apply(x,2,mean)
x <- sweep(x,2,ka,"-")

#valkaisu
Sx <- cov(x)
eig <- eigen(Sx)
V <- eig$vectors
L <- eig$values
T <- diag(1/sqrt(L))%*%t(V)
z <- x%*%t(T)

#autokov.matriisi viiveellä tau ja sen ominaisarvovektorit
Sz <- autocov(z,tau)
E <- eigen(Sz)$vectors

#ratkaisumatriisi W ja estimoidut signaalit S
W <- t(E)%*%T
```

```

S <- x%*%t(W)
list(S=S, W=W)
}

## SOBI ##

library(JADE) # ladataan paketti JADE, josta löytyy
              # yhteisdiagonalisointifunktio rjd

sobi <- function(x,k){
# estimoi BSS-ongelman ratkaisumatriisin ja
# lähdesignaalit SOBI-menetelmällä
# x = datamatriisi
# k = rotatointiin käytettävien autokovarianssimatriisien lkm

#keskistys
ka <- apply(x,2,mean)
x <- sweep(x,2,ka,"-")

#valkaisu
Sx <- cov(x)
eig <- eigen(Sx)
V <- eig$vectors
L <- eig$values
T <- diag(1/sqrt(L))%*%t(V)
z <- x%*%t(T)

#autokov.matriisit viiveillä 1:k
Sz <- autocov(z,1)
if( k > 1 )
  for(i in 2:k){Sz <- rbind(Sz,autocov(z,i))}

#yhteisdiagonalisointi ja om.arvovektorit
result <- rjd(Sz)
E <- result$V

#ratkaisumatriisi W ja estimoidut signaalit S
W <- t(E)%*%T
S <- x%*%t(W)
list(S=S, W=W)
}

```

```

## FOBI ##

fobi <- function(x){
# estimoi ICA-ongelman ratkaisumatriisin ja
# lähdesignaalit FOBI-menetelmällä
# x = datamatriisi

n <- dim(x)[1]

#keskistys
ka <- apply(x,2,mean)
x <- sweep(x,2,ka,"-")

#valkaisu
Sx <- cov(x)
eig <- eigen(Sx)
V <- eig$vectors
L <- eig$values
T <- diag(1/sqrt(L))%*%t(V)
z <- x%*%t(T)

#neljänsien momenttien matriisi ja sen ominaisarvovektorit
r <- NULL
for(i in 1:n){ r[i] <- t(z[i,])%*%z[i,] }
rz <- sweep(z,1,sqrt(r),"*")
Sz <- (t(rz)%*%rz)/n
E <- eigen(Sz)$vectors

#ratkaisumatriisi W ja estimoidut signaalit S
W <- t(E)%*%T
S <- x%*%t(W)
list(S=S, W=W)
}

## ISR-indeksi ##

ISR <- function(W,A){
# laskee ISR-indeksin
# W = estimoitu ratkaisumatriisi
# A = todellinen sekoitusmatriisi
G <- W %*% A

```

```

p <- dim(G)[2]
rivimax <- apply(abs(G),1,max)
delta <- sweep(abs(G),1,rivimax,"/")
sqrt((1/p)*sum(delta^2)-1)
}

## Esimerkki simulointifunktiosta ##

sim <- function(n,A,rep){
# generoi lähdesignaali normaali-, tasa- ja eksponenttijakaumista
# sekoittaa signaalit
# estimoi ratkaisumatriisit eri menetelmillä
# palauttaa ISR-indeksit menetelmittain
# n = signaalien otoskoko
# A = sekoitusmatriisi
# rep = toistojen lkm

ISR1 <- NULL
ISR2 <- NULL
ISR3 <- NULL
ISR4 <- NULL
ISR5 <- NULL

for(i in 1:rep){

s <- cbind(rnorm(n),rexp(n),runif(n,-sqrt(3),sqrt(3)))
x <- s %*% t(A)

W1 <- amuse(x,1)$W
ISRa <- ISR(W1,A)
ISR1 <- c(ISR1,ISRa)

W2 <- sobi(x,k=10)$W
ISRb <- ISR(W2,A)
ISR2 <- c(ISR2,ISRb)

W3 <- sobi.rob(x,k=10)$W
ISRc <- ISR(W3,A)
ISR3 <- c(ISR3,ISRc)

W4 <- fobi(x)$W
ISRd <- ISR(W4,A)

```

```
ISR4 <- c(ISR4,ISRf)

W5 <- fobi.rob(x)$W
ISRfr <- ISR(W5,A)
ISR5 <- c(ISR5,ISRfr)

}
list(AMUSE=ISR1,SOBI=ISR2,SOBI.ROB=ISR3,FOBI=ISR4,FOBI.ROB=ISR5)
}
```