

**Karışık İşitsel ve Görsel Sinyallerin K r Kaynak
Ayrıştırma Yöntemiyle Birbirlerinden Ayrılması**

Pelin G ven
Y ksek Lisans Tezi

Fen Bilimleri Enstit s 
Elektrik-Elektronik M hendisliđi Anabilim Dalı
Nisan – 2005

ÖZET

Yüksek Lisans Tezi

KARIŞIK İŞİTSEL VE GÖRSEL SİNYALLERİN KÖR KAYNAK AYRIŞTIRMA YÖNTEMİYLE BİRBİRLERİNDEN AYRILMASI

PELİN GÜVEN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr.Emin GERMEN

2005, 62 Sayfa

Bu çalışmada, karışık sinyallerin ayrıştırılması ve ters evrişimlenmesi üzerinde durulmuştur. Karışımlar doğrusal veya evrişimlenmiş olabilir. Doğrusal karışımların ayrıştırılmasında Kör Kaynak Ayrıştırma algoritmaları kullanılmıştır. Ancak evrişimlenmiş karışımların ayrıştırılmasında ve ters evrişiminde, bu algoritmalar yetersiz kalır. Evrişimlenmiş karışımlarda gecikme, yansıma ve çoklu-yol etkileri de gözönüne alınmalıdır. Bu evrişimlenmiş karışımların ters evrişiminde Kör Ters Evrişim yöntemi kullanılır. Bu çalışmada, evrişimlenmiş karışımları ayırmak ve ters evrişimlerini sağlamak için bir frekans uzayı yöntemi sunulmuştur. Bu çalışmada, evrişimlenmiş karışımlar olarak yansımaların olduğu ortamda kaydedilen gerçek-dünya karışımları kullanılmıştır. Kör Kaynak Ayrıştırma ve Kör Ters Evrişim algoritmaları MATLAB ortamında gerçekleştirilmiştir. Ayrıca, frekans uzayı yönteminin yakınsama oranının geliştirilebileceği de, bu çalışmada sunulmuştur.

Anahtar Kelimeler: Kör Kaynak Ayrıştırma, Kör Ters Evrişim,
Frekans Uzayı Ters Evrişim, Doğrusal Karışımlar,
Evrişimlenmiş Karışımlar

ABSTRACT

Graduate School of Science Thesis

BLIND SOURCE SEPERATION OF MIXED VISUAL AND AUDIO SIGNALS

PELİN GÜVEN

Anadolu University

Graduate School of Sciences

Electrical and Electronics Engineering Program

Supervisor: Assist. Prof. Dr. Emin GERMEN

2005, 62 pages

In this study, separation and deconvolution of blind source mixtures is considered. These mixtures can be linear or convolutive. Blind Source Separation algorithms are applied to linear mixtures of sources but they are unable to separate and deconvolve convolutive mixtures. Delays, reverbation and multi-path effects must be considered in convolutive mixtures. Blind Deconvolution algorithms can be used to deconvolve these mixtures. In this study, to deconvolve and unmix convolutive mixtures, a frequency domain Blind Deconvolution method is presented. The mixtures used in this work, are real-world mixtures recorded in reverberating environment. Blind Source Separation and Blind Deconvolution algorithms are implemented in MATLAB. In this study, it is also presented that convergence rate of frequency domain method can be improved.

**Keywords: Blind Source Separation, Blind Deconvolution,
Frequency Domain Deconvolution, Linear Mixtures,
Convolved Mixtures**

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİLLER DİZİNİ	v
1. GİRİŞ.....	1
2. KÖR KAYNAK AYRIŞTIRMA (Blind Source Separation).....	4
2.1. Bağımsız Bileşen Analizi (Independent Component Analysis).....	6
2.2. Kör Kaynak Algoritmalarının Geliştirilmesi.....	8
2.2.1. Kullback-Liebnir Iraksaklığı ve Karşılıklı Bilgi Minimizasyonu...10	
2.2.2. Kurtosis Maksimizasyonu-Minimizasyonu ve Negentropi Maksimizasyonu.....	11
2.3. Kör Kaynak Ayrıştırma Algoritmaları.....	14
2.3.1. Bell ve Sejnowsky Algoritması.....	15
2.3.2. Amari Algoritması.....	17
3. KÖR TERS EVRİŞİM (Blind Deconvolution).....	20
3.1.Frekans Uzayı Algoritmaları.....	24
4. UYGULAMA.....	32
4.1. Kör Kaynak Ayrıştırma Algoritmalarının Gerçeklenmesi.....	32
4.2. Kör Ters Evrişim Algoritmasının Gerçeklenmesi.....	40
5. SONUÇLAR.....	49

KAYNAKLAR.....51

EKLER.....54

ŞEKİLLER DİZİNİ

2.1. 2 Kaynak-2 Algılayıcı durumunda kaynakların doğrusal karışması.....	4
2.2. Kör Kaynak Ayırıştırma Modeli	5
2.3. Permütasyon ve Ölçekleme Problemi	9
2.4. Kör Kaynak Ayırıştırma Modeli	10
2.5. Gauss Dağılımı ve Süper Gaussian Dağılım.....	12
2.6. Gauss dağılımına sahip rassal bir sinyal ve konuşma sinyali Histogramları	13
2.7. Hiperbolik Tanjant Fonksiyonu	16
3.1. Evrişimli Karışımlar Durumu	20
3.2. 2x2 Evrişimli Karışımlar Modeli	21
3.3. İki Kaynak İki Algılayıcı için Bell ve Sejnowski Ağ Yapısı.....	23
3.4. Karmaşık Yapay Sinir Ağları Nöronu	26
3.5. Karmaşık Uzayda Hiperbolik Tanjant Fonksiyonu	27
3.6. Karmaşık Aktivasyon Fonksiyonu.....	28
3.7. Frekans Uzayı Kör Ters Evrişim Algoritması Çizeneği.....	29
3.8. Hamming Penceresi	30
4.1. Çello ve keman çalgılarından kaydedilen kaynak sinyalleri.....	32
4.2. Kaynakların doğrusal üstdüşümlerinden oluşan algılayıcı sinyalleri	33
4.3. Amari Algoritması kaynak kestirimleri	35
4.4. Bell algoritması kaynak kestirimleri	35
4.5. Viola çalgısından kaydedilen iki farklı kaynak sinyali.....	36
4.6. Kaynakların doğrusal üstdüşümlerinden oluşan algılayıcı sinyalleri	37
4.7. Bell algoritması kaynak kestirimleri	37
4.8. Amari algoritması kaynak kestirimleri	38
4.9. Kaynak görüntü sinyalleri.....	39
4.10. Algılayıcı sinyalleri.....	39
4.11. Amari algoritması kaynak kestirimleri	40
4.12. Bu çalışmada kullanılan güncelleme şeması temel gösterimi.....	42

4.13. Konuşma - Konuşma karışımı mikrofon 1.....	43
4.14. Konuşma - Konuşma karışımı mikrofon 2.....	43
4.15. Konuşma sinyali 1 kestirimi	44
4.16. Konuşma sinyali 2 kestirimi	44
4.17. Konuşma- Müzik karışımı mikrofon 1.....	45
4.18. Konuşma- Müzik karışımı mikrofon 2.....	45
4.19. Müzik sinyali kestirimi	46
4.20. Konuşma sinyali kestirimi	47
4.21. Farklı ters evrişim filtre dereceleriyle konuşma sinyali kestirimleri	47

1. GİRİŞ

Günümüzde, karışık sinyallerden kaynak sinyallerine ulaşmak, pek çok uygulamada gereklidir. Farklı kaynaklardan gelen sinyaller, çeşitli algılayıcılarla karışmış bir şekilde algılandığında hangi sinyalin, hangi kaynaktan geldiğini bulmak ve orijinal kaynak sinyallerini kestirmek için bu güne kadar çeşitli yöntemler geliştirilmiştir. Karışık verileri ayırtmak ve kaynak sinyallerini mümkün olduğunca doğru hesaplamak gerekir. Ancak; çoğu zaman orijinal sinyaller ve sinyalleri karıştıran aktarım fonksiyonu hakkında bir bilgi bulunmaz. Özellikle gerçek dünya uygulamalarında kaynak ayırtmaya ihtiyaç duyulsa da, kaynak sinyallerinin kendisi, karıştırma ortamı, kaynaklarının yerleri vb. önveriler bilinmediğinden kaynak sinyallerinin kendilerine ulaşmak oldukça zordur.

Kokteyl parti problemi, bu ayırtma probleminin açıklandığı en temel örnektir. Oldukça kalabalık ve gürültülü bir ortamda insanlar, herhangi bir sese odaklanabilirler. Bu bir konuşma, bir müzik melodisi veya başka bir ses olabilir. İnsan beyni, karışık ses sinyallerini ayırtma yeteneğine sahiptir. Ancak dijital dünya, kaynakların karışmış verilerinden, başka hiçbir bilgi olmaksızın, orijinallerine ulaşma yeteneğinden yoksundur. Sadece algılayıcılardan alınan karışık kaynak sinyallerinden oluşan verileri kullanarak, kaynak sinyallerinin kestirimini amaçlayan çözümler geliştirilmiştir[1-30]. Bu kaynak ayırtma yöntemleri, Kör Kaynak Ayırtma yöntemleridir. Kör Kaynak Ayırtma, ortamın karıştırma aktarım fonksiyonunu bilmeden, sadece bir algılayıcı dizisinden alınan karışık verileri kullanarak bilinmeyen kaynaklardan gelen sinyalleri kestirmeye çalışır. Burada, sinyallerle ilgili belirli varsayımlar belirlenir. Sinyallerin İstatistiksel olarak bağımsız olduğu kabul edilir. Kaynakları farklı olan sinyaller zaten bağımsız olduğundan, bağımsızlık ölçütü Kör Kaynak Ayırtma açısından rahatlıkla kullanılabilir. Kaynak sinyallerinin evrişimlenmiş olduğu durumlarda, Kör Kaynak Ayırtma yaklaşımının dışında Kör Ters Evrişim

yöntemleri kullanılmaktadır. Evrişimlenmiş karışımlar zaman gecikmeleri, ortam yansımaları gibi etkileri de içeren karışımlardır.

Kör Kaynak Ayırıştırma ve Kör Ters Evrişim son zamanlarda geliştirilmeye başlanmış olduğundan, üzerinde hala pek çok araştırma yapılmaktadır ve çok çeşitli uygulama alanları vardır. Konuşma geliştirme ve konuşma tanıma, akustik, biyomedikal sinyal analizleri, sonar ve radar sistemleri, görüntü işleme ve telekomünikasyon Kör Kaynak Ayırıştırma ve Kör Ters Evrişim'in uygulandığı alanların başlıcalarıdır.

Kör Kaynak Ayırıştırma yöntemi üzerinde ilk çalışmalar, Jutten ve Herault tarafından[28] geliştirilmiştir. Çalışmalarında, biyolojik beyin dalgalarını ayrıştırılması amaçlanmıştır. Biyolojik algılayıcılar, merkezi sinir sisteminden pek çok farklı kaynaklardan gelen sinyallerin karışmış bir şeklini algılar. Bu karışık algılayıcı sinyallerinin her bir bileşeni, beynin farklı kısmına ait bir sinyaldir. Burada her bir bileşeni ayrıştırıp, kaynak sinyallerine ulaşabilmek için algılayıcılardan gelen sinyallerin ayrıştırılması gereklidir. Böylece, ilk Kör Kaynak Ayırıştırma yaklaşımı geliştirilmiştir. Zamanla Kör Kaynak Ayırıştırma yaklaşımı giderek daha popüler hale gelmiştir. Bell ve Sejnowsky[29] tarafından ilk olarak bilgi teorisi tabanlı bir algoritma ortaya koyulmuştur. Bu algoritma, oldukça başarılı olmasına rağmen kaynak sayısı çoğaldıkça sonuçlarda istenilen ayrıştırma kalitesine ulaşamadığı gözlenmiştir. Daha sonra, Amari[10,20] tarafından doğal gradyan yöntemi ile yine bilgi teorisi tabanlı ve uygulanabilirliği daha yüksek başka bir algoritma daha geliştirilmiştir.

Önceleri, doğrusal karışık sinyallerin ayrıştırılmasına çalışılırken, gerçek dünya uygulamalarına geçilmesiyle birlikte, evrişimlenmiş karışımların ayrıştırılmasında Kör Kaynak Ayırıştırma algoritmaları yetersiz kalmıştır. Böylece Kör Ters Evrişim yöntemleri geliştirilmiştir. Farklı uzaylarda gerçekleştirilebilir Kör Ters Evrişim algoritmaları bulunmaktadır[15,16,3,6,17,18]. Bu çalışmada, Amari'nin Kör Kaynak Ayırıştırma algoritmasının Smagradis[2,3,4] tarafından geliştirilen frekans uzayına adaptasyonu kullanılmıştır.

Yapay Sinir Ağları, Kör Kaynak Ayırıştırma ve Kör Ters Evrişim algoritmalarında oldukça sıklıkla kullanılmaktadır. Bu çalışmada da, yapay sinir ağları yaklaşımlarıyla frekans uzayında Amari algoritmasının gerçekleştirilmesi

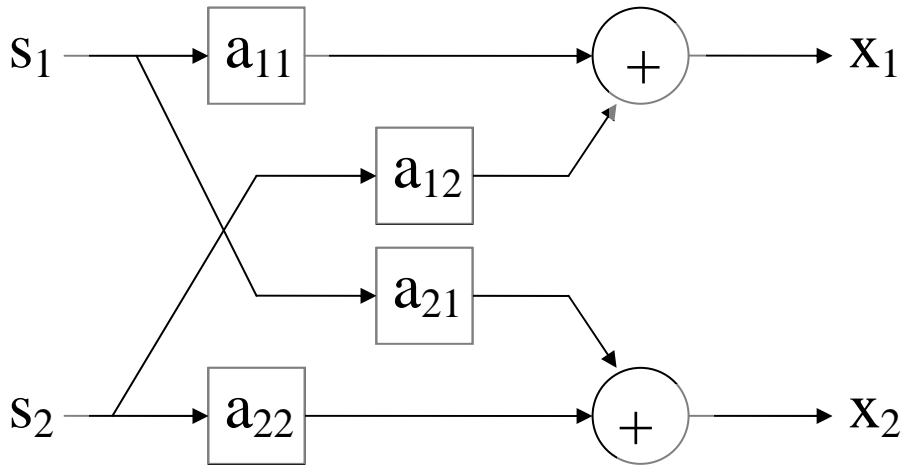
yapılmış ve mzik-mzik karışımının ayrıştırılması hedeflenmiştir. Öncelikle Kr Kaynak Ayrıştırma ve Kr Ters Evrişim prensipleri incelenmiş, ardından bu yöntemler için geliştirilen algoritmalar araştırılmıştır. Kr Kaynak Ayrıştırma ve Kr Ters Evrişim algoritmalarının gereklenmeleri yapılmıştır. Kr Kaynak Ayrıştırma algoritmalarının gereklenmesinde, doğrusal olarak karıştırılmış mzik sinyalleri hem Amari, hem de Bell algoritmalarıyla ayrıştırılmıştır. Ardından Kr Ters Evrişim algoritmasıyla evrişimlenmiş konuşma-konuşma ve konuşma-mzik karışımının ayrıştırılması sağlanmıştır.

2. KÖR KAYNAK AYRIŞTIRMA (Blind Source Separation)

Kör Kaynak Ayırıştırma yöntemi, belli sayıdaki kaynak sinyalinin gerçel sayılarla çarpılıp, üstdüşümleri alındığını öngörerek, bu sinyallerden oluşmuş gözlemlenen sinyallerin doğrusal biçimde karıştığını varsayar ve bunların yardımıyla kaynak sinyallerinin yeniden ayrıştırılmasını sağlar[1-30]. İki kaynak ve iki algılayıcı için Kör Kaynak Ayırıştırma yöntemini modelleyecek olursak bu model:

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) \end{aligned} \quad (2.1)$$

şeklinde ifade edilebilir. $x_1(t)$ ve $x_2(t)$ gözlemlenen sinyallerdir. Şekil 2.1’de iki kaynak-iki algılayıcı için Kör Kaynak Ayırıştırma karıştırma modeli verilmiştir. Kör Kaynak Ayırıştırma’nın amacı $x_1(t)$ ve $x_2(t)$ sinyallerini kullanarak $s_1(t)$ ve $s_2(t)$ sinyallerini elde etmektir.



Şekil 2.1. 2 Kaynak-2 Algılayıcı durumunda kaynakların doğrusal karışması

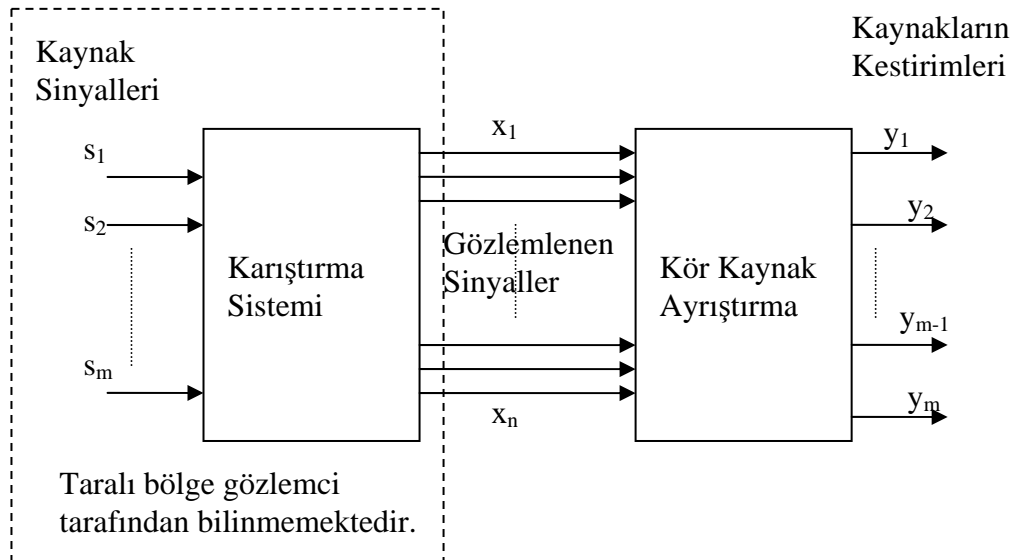
Şekil 2.1’de de görüldüğü gibi, s_1 ve s_2 kaynak sinyallerinin, bilinmeyen a_{11} , a_{21} , a_{12} , a_{22} gerçel katsayılarıyla doğrusal üstdüşümleri alınmaktadır. a_{11} ve a_{22} kaynakların karışımlara doğrudan etkilerini, a_{12} ve a_{21} kaynakların karışımlara olan çapraz etkilerini belirtir. s_1 ve s_2 kaynak sinyalleri ve a_{11} , a_{12} , a_{21} , a_{22} , karıştırma katsayıları gözlemci tarafından bilinmemektedir. Gözlemci sadece algılayıcılardan

alınan ve kaynak sinyallerinin doğrusal üstdüşümlerinden oluşan, x_1 ve x_2 anlık karışmış sinyalleri bilmektedir. Bu, en temel Kör Kaynak Ayrıştırma modelidir.

Yöntemin genel tanımını yapacak olursak: Gözlemci n adet gözlemlenen sinyali $(x_1(t), x_2(t), \dots, x_n(t))$ kullanarak m adet kaynak sinyalini $(s_1(t), \dots, s_m(t))$ ayrıştırmayı hedeflemektedir. Ne var ki, ne kaynakların sayısı m , ne de karışım hakkında gözlemci hiçbir bilgiye sahip değildir. Gözlemci için yegane varsayım, bu karışık sinyallerin sıfır ortalamalı bir istatistikten geldiğidir. Şekil 2.2’de görüldüğü gibi, gözlemci, kaynak sinyallerinin n kanaldan alınan doğrusal karışmış durumlarını bilir. Karıştırma katsayıları da bilinmeyen değerlerdir. Bu durumda Kör Kaynak Ayrıştırma karıştırma modelinin matematiksel ifadesi aşağıda verilmiştir:

$$\mathbf{x}(t) = \mathbf{A}s(t) = \sum_{j=1}^m s_j(t)\mathbf{a}_{ji} \quad (2.2)$$

Burada $\mathbf{x}(t)=[x_1(t), x_2(t), \dots, x_n(t)]^T$ elemanları t anındaki doğrusal karışımlardan oluşan, n boyutlu gözlemlenen veri vektörü, $\mathbf{s}(t)=[s_1(t), s_2(t), \dots, s_m(t)]^T$ elemanları t anındaki kaynak sinyallerinden oluşan, m boyutlu kaynak vektörü, $\mathbf{A}=[\mathbf{a}_1 \dots \dots \mathbf{a}_m]$ olmak üzere, elemanları karıştırma katsayılarından oluşan $n \times m$ matristir.



Şekil 2.2. Kör Kaynak Ayrıştırma Modeli

Kör Kaynak Ayırıştırma'da bilinmeyen kaynak sinyallerini bulmak için çeşitli yaklaşımlar geliştirilmiştir. Bağımsız Bileşen Analizi de bu yaklaşımlardan biridir ve Kör Kaynak Ayırıştırma'nın temelini oluşturur.

2.1. Bağımsız Bileşen Analizi (Independent Component Analysis)

İstatistik, veri analizi, sinyal işleme, yapay sinir ağları gibi disiplinlerde genel problemlerden biri, verinin uygun bir gösterim biçimini bulmaktır, bu uygun bir dönüşüm demektir[1]. Bu dönüşüm, m boyutlu rassal değişken x 'i, istenilen özelliklere sahip n boyutlu s değişkenine dönüştüren ve $s=f(x)$ ile tanımlanan f fonksiyonu ile belirtilir. Doğrusal dönüşümler, x değişkeninin doğrusal bir W matrisiyle $s=Wx$ şeklinde çarpımıyla tanımlanır. Pek çok metot, uygun bir doğrusal dönüşüm bulmak için uygulanmaktadır. Bağımsız Bileşen Analizi bu metotlardan en çok kullanılanlarındandır[1,19]. Bu metotlar genelde bu dönüşümü optimum kılmak için gerekli durumu tanımlarlar. Bu optimallik, boyut azaltımı, istatistiksel bağımsızlık, W dönüşümünün basitliği ya da başka ölçütler olabilir.

Kör Kaynak Ayırıştırma'nın da temeli olan Bağımsız Bileşen Analizi, rassal $x=[x_1, x_2, \dots, x_n]^T$ vektörü için rassal $y=[y_1, y_2, \dots, y_m]^T$ vektörünün y_j bileşenlerini istatistiksel olarak bağımsız hale getirecek doğrusal $y=Wx$ dönüşümünü, yani W matrisini bulmaya çalışır. Kör Kaynak Ayırıştırma için optimum dönüşüm prensibi, y_j bileşenlerinin istatistiksel olarak bağımsız olmasıdır; çünkü s_j bileşenleri farklı kaynak sinyalleridir. İstatistiksel bağımsızlık şu şekilde tanımlanır:

y_1, y_2, \dots, y_m sıfır ortalamalı rassal değişkenler için ortak yoğunluk fonksiyonu $f(y_1, \dots, y_m)$, $f_i(y_i)$, y_i değişkeninin bileşen olasılık yoğunluk fonksiyonu olmak üzere; eğer

$$f(y_1, \dots, y_m) = f_1(y_1) f_2(y_2) \dots f_m(y_m) \quad (2.4)$$

şeklinde tanımlanırsa y_i değişkenleri bağımsızdır. Başka bir ifadeyle; bileşenlerin ortak yoğunluk fonksiyonları, her bir bileşenin, bileşen olasılık yoğunluk fonksiyonlarının çarpımına eşitse, bu bileşenler istatistiksel olarak bağımsızdır.

Rassal $x=[x_1, x_2, \dots, x_n]^T$ vektörü için Bağımsız Bileşen Analizi, bağımsızlık ölçüsü herhangi bir $F(y_1, \dots, y_m)$ maliyet fonksiyonunu optimize ederek, y_i bileşenlerini olabildiğince bağımsız hale getiren $y=Wx$ doğrusal dönüşümünü

bulmaktır. Bu tanım, veri üzerinde herhangi bir varsayım yapmadan Bağımsız Bileşen Analizi için en genel tanımdır. Burada amaçlanan, herhangi bir maliyet fonksiyonunu optimize ederek bileşenlerin bağımsızlığını sağlamaktır. Tanımda geçen maliyet fonksiyonu için çeşitli yaklaşımlar belirlenebilir. Kör Kaynak Ayırıştırma için, y_j bileşenleri, kaynakların kestirimleridir ve kestirimlerin bağımsızlığını sağlayan ve böylece kaynak sinyallerine ulaşan doğrusal dönüşüm aslında $W=A^{-1}$ dönüşümüdür. Kör Kaynak Ayırıştırma'da amaç; A doğrusal dönüşümü ve kaynak sinyalleri olan s_j bileşenlerini bilmeden, kaynakların bağımsız olması bilgisini kullanarak $W=A^{-1}$ doğrusal dönüşümünü ve dolayısıyla

$$Y=Wx=A^{-1}x=A^{-1}As=s \quad (2.5)$$

durumunu sağlayarak kaynakları hesaplamaktır.

Bağımsız Bileşen Analizi modelinin geçerli olabilmesi için kaynakların bağımsızlığı yanında aşağıdaki gereklilikler de sağlanmalıdır:

1. En fazla bir s_j bileşeni Gauss dağılımında olmalıdır.
2. Gözlemlenen doğrusal karışımların sayısı m bağımsız bileşenlerin sayısı n 'den büyük veya eşit olmalıdır.
3. A matrisi tam sütun ranklı olmalıdır.

Yukarıda bahsedilen üç durum da Bağımsız Bileşen Analizi'ni uygulamak için gereklidir[13]. Son iki gereklilik gözlemlenen verinin kaynak verileri bilgilerini yeterince içermesiyle ilgilidir. Herhangi bir doğrusal sisteminin tek çözümü olması için eldeki denklemlerin, bilinmeyen değişken sayısından büyük olması gerekir. A matrisinin tam sütun ranklı olmaması durumunda n bağımsız bileşenin doğrusal üstdüşümleriyle elde edilen m tane gözlemlenen sinyalde bağımsız bileşenlerin hepsinin bilgisi bulunmayacaktır. Ayrıca bu A karıştırma matrisiyle ifade edilen karıştırma sisteminin tersinin alınmadığını gösterir. Başka bir deyişle; A 'nın tam sütun ranklı olmaması, W ayırıştırma matrisinin de hesaplanamayacağını gösterir.

Birden fazla Gauss dağılımına sahip bileşeni ayırıştırmak imkansızdır; çünkü Gauss dağılımına sahip bileşenlerin doğrusal üstdüşümleri yine Gauss dağılımına sahiptir. Bu yüzden s_j bileşenlerinin en fazla bir tanesinin Gauss dağılımında olması gereklidir[21,22].

2.2. Kör Kaynak Algoritmalarının Geliştirilmesi

Kör Kaynak Ayırıştırma algoritmaları, Bağımsız Bileşen Analizi temelini kullanarak kaynakların istatistiksel bağımsızlığını temel alan, belirli maliyet fonksiyonlarının optimizasyonunu sağlar. Kör Kaynak Ayırıştırma algoritmalarının geliştirilmesinde önemli olan iki konu: doğru maliyet fonksiyonunun tanımlanması ve bu fonksiyonu en iyi optimize edecek yaklaşımın belirlenmesidir[7,9]. Kör Kaynak Ayırıştırma algoritmalarıyla, doğrusal bir W dönüşümü bularak kaynakların kestirimlerine yakınsamaya çalışılır.

Kör Kaynak Ayırıştırma karıştırma modeli

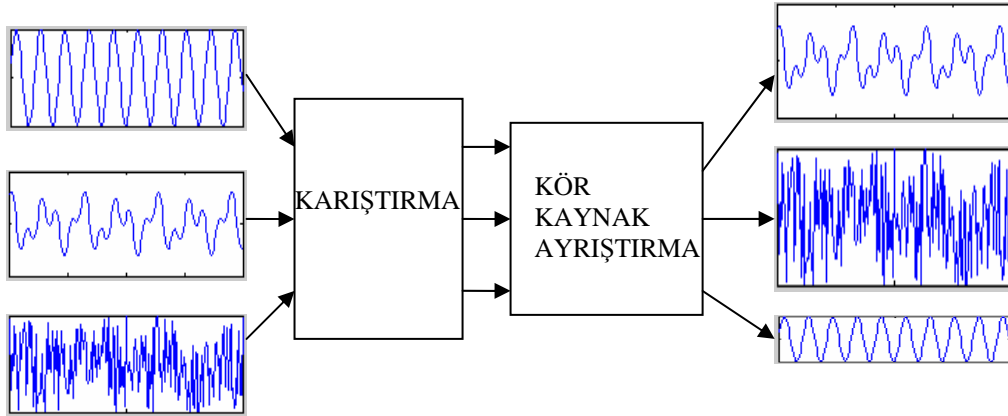
$$\mathbf{x}(t) = A\mathbf{s}(t) = \sum_{j=1}^m s_j(t)\mathbf{a}_j \quad (2.6)$$

şeklindedir. $A=[\mathbf{a}_1 \dots \mathbf{a}_m]$ matrisi $n \times m$ boyutlu doğrusal bir matristir. Amaç

$$\mathbf{y}(t) = W\mathbf{x}(t) \quad (2.7)$$

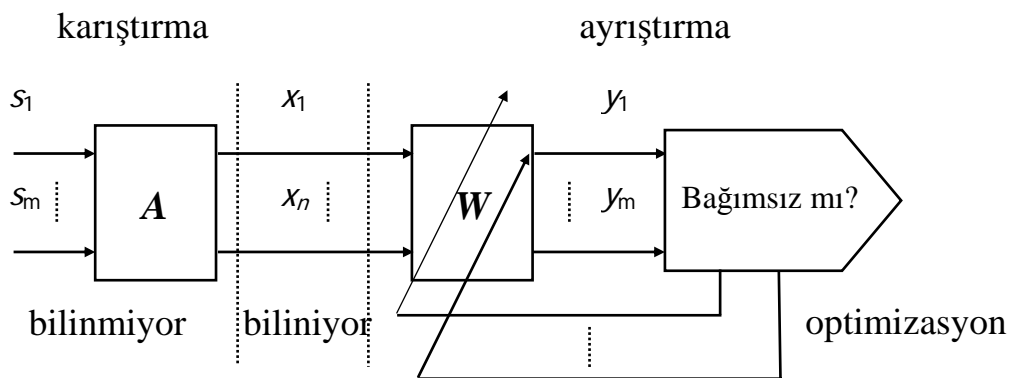
$$\mathbf{y}(t) = \hat{\mathbf{s}}(t) = WAs(t) = PDs(t) \quad (2.8)$$

dönüşümünü bulmaktır. Burada P permütasyon matrisi ve D ölçekleme matrisidir. Kör Kaynak Ayırıştırma'da kaynakların bağımsızlığı dışında başka bilgi bulunmadığından kestirimlerde iki belirsizlik vardır. Bunlar Permütasyon ve Ölçekleme belirsizlikleridir[1,11,13,19,20]. Kaynakların kestirimi P ve D matrisleriyle çarpılmış şekildedir. i 'inci kaynak, çıkışta P matrisine bağlı olarak $\mathbf{y}(t)$ 'nin başka herhangi bir bileşeni olarak bulunabilir. Bu durum, permütasyon belirsizliğinden kaynaklanmaktadır. Kaçınıcı kaynağın kaçınıcı çıkış bileşeni olarak bulunacağı bilinmemektedir. Ölçeklenme belirsizliğine bağlı olarak; kaynakların kestirimleri, kaynak sinyallerinin D köşegen matrisinin katsayılarıyla ölçeklenmiş durumlarıdır. Şekil 2.3'de permütasyon ve ölçekleme belirsizliklerine bağlı olarak kaynakların kestirimlerinin çıkışta nasıl bulunabileceği gösterilmektedir.



Şekil 2.3. Permutasyon ve Ölçekleme Problemi: Farklı kaynaklar, çıkışta farklı bileşenler olarak bulunabilir. Çıkış bileşenleri D matrisiyle ölçeklenirler.

Kör Kaynak Ayrıştırma, farklı kaynaklardan gelen sinyallerin bağımsız olduğu varsayımını kullanarak karışık sinyalleri ayırmaya çalışır. Kör Kaynak Ayrıştırma algoritmalarında kullanılacak maliyet fonksiyonunun belirlenmesinde daha önce bahsedilen istatistiksel bağımsızlık tanımı doğrudan doğruya kullanılamaz. Gerçeklenebilir bir maliyet fonksiyonu bulabilmek için literatürde çeşitli yaklaşımlar önerilmiştir[1,19,20]. İstatistiksel bağımsızlık prensipleri, bu maliyet fonksiyonlarını oluşturmak için kullanılır.



Şekil 2.4. Kör Kaynak Ayrıştırma Modeli: Bileşenlerin bağımsızlığı sağlanana kadar maliyet fonksiyonunun optimizasyonu sağlanır.

Şekil 2.4'de Kör Kaynak Ayırıştırma modeli verilmiştir. Bilinen karışık sinyalleri kullanarak, bileşenlerin bağımsızlığı sağlanana kadar maliyet fonksiyonunun optimizasyonu sağlanır. Kör Kaynak Ayırıştırma algoritmalarının başarısı, doğru optimizasyonun sağlanmasıyla ilintilidir. Bu başarı, hem en uygun maliyet fonksiyonu tanımıyla, hem de bu maliyet fonksiyonunun optimizasyonu için en uygun algoritmanın kullanılmasıyla sağlanır[7]. Maliyet fonksiyonlarının belirlenmesinde çeşitli yaklaşımlar kullanılmıştır. Kullback-Liebner Iraksaklığı çıkış bileşenleri arasındaki bağımlılığı minimize eden bir yaklaşımdır[3,4,10,19,20]. Aynı şekilde, Karşılıklı Bilgi Minimizasyonu yaklaşımı, giriş ve çıkış bileşenleri arasındaki karşılıklı bilginin minimize edilmesi esasına göre çalışır. Bunların yanında, karışımları Gauss dağılımından uzaklaştırarak ayırıştırma sağlayan Kurtosis Maksimizasyonu-Minimizasyonu ve negatif entropiyi kullanan maksimizasyon yaklaşımları da literatürde kullanılmaktadır[6,13,19,20,23,24].

2.2.1. Kullback-Liebner Iraksaklığı ve Karşılıklı Bilgi Minimizasyonu

Kullback-Liebner Iraksaklığı ve Karşılıklı Bilgi Minimizasyonu, bilgi teorisi tabanlı bir yaklaşımdır. Bunun için öncelikle bilgi teorisi açısından temel oluşturacak entropi tanımını yapmak konunun anlaşılmasında yardımcı olacaktır. Entropi, bilgi teorisi yaklaşımlarının temel ölçüm birimidir ve rassal değişkenlerin belirsizliğinin ölçüsüdür. $f(\mathbf{y})$ rassal \mathbf{y} vektörünün olasılık yoğunluk fonksiyonu olmak üzere, \mathbf{y} vektörünün entropisi aşağıdaki şekilde tanımlanır:

$$H(\mathbf{y}) = \int f(\mathbf{y}) \log f(\mathbf{y}) d\nu \quad (2.9)$$

Karşılıklı bilgi, iki bağımsız rassal değişken için bir bağımlılık tanımıdır. Karşılıklı bilgi tanımı aşağıdaki şekildedir:

$$I(x, y) = \sum_i H(x_i) - H(y) \quad (2.10)$$

Karşılıklı bilgi rassal değişken x 'in rassal değişken y hakkında ne kadar bilgi içerdiğinin bir göstergesidir. Karşılıklı bilgi ancak ve ancak rassal değişkenler istatistiksel olarak bağımsız olduğunda sıfır olur.

Kör Kaynak Ayırıştırma'da bileşenlerin karşılıklı bilgileri şu şekilde tanımlanır:

$$I(y_1, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{y}) \quad (2.11)$$

Dolayısıyla, Kör Kaynak Ayırıştırma'da bileşenlerin karşılıklı bilgilerini minimize edecek bir algoritma, bileşenlerin bağımsızlığını sağlayacaktır[10,19,20]. Sonunda da, bağımsız kaynak kestirimlerine ulaşılabilecektir.

Aynı şekilde $f_i(\mathbf{y})$ \mathbf{y} rassal vektörünün olasılık yoğunluk fonksiyonu olmak üzere Kullback- Leibner İraksaklığı

$$\delta(f_1, f_2) = \int f_1(\mathbf{y}) \log \frac{f_1(\mathbf{y})}{f_2(\mathbf{y})} d\mathbf{y} \quad (2.12)$$

şeklinde tanımlanır. Bu tanım iki rassal dağılım arasındaki entropik uzaklık olarak düşünülür. Kör Kaynak Ayırıştırma algoritmalarının temeli, çıkış bileşenleri arasındaki bağımlılığı minimize etmek olduğundan, bağımlılığı gösteren Kullback- Leibner İraksaklığını minimize etmek, ayırıştırma algoritması olarak benimsenebilir[2,3,4,10,19,20]. Bu iraksaklığın sıfır olması durumunda, çıkışlar bağımsız bileşenlerden oluşur. Karşılıklı bilgi tanımındaki y_i bileşenlerinin bağımsız olduğu düşünülürse, bunların bileşen olasılık yoğunluk fonksiyonlarının çarpımı, istatistiksel bağımsızlık tanımını sağlar. Bağımsızlık durumunda, bileşen olasılık yoğunluk fonksiyonlarının çarpımı, ortak olasılık dağılımına eşittir. Sonuç olarak, y_i bileşenlerinin bağımsızlığını sağlamak, $f(\mathbf{y})$ gerçek yoğunluk fonksiyonlarıyla, bağımsız olmaları durumundaki ortak dağılımları arasındaki Kullback-Leibner uzaklığını minimize etmektir. Bu aynı zamanda karşılıklı bilginin maksimizasyonudur.

Algılayıcılardan alınan verinin karşılıklı bilgilerinin minimize edilmesi, çıkış bileşenlerinin bilgilerinin maksimum olmasıdır, böylece çıkış bileşenlerinin bağımsızlığı sağlanır. Çıkış bileşenlerinde maksimum bilgi olmalıdır. Yani kaynakların kestirimlerindeki bilgiyi maksimize etmek kaynakları doğru kestirmeye yarar.

2.2.2. Kurtosis Maksimizasyonu-Minimizasyonu ve Negentropi

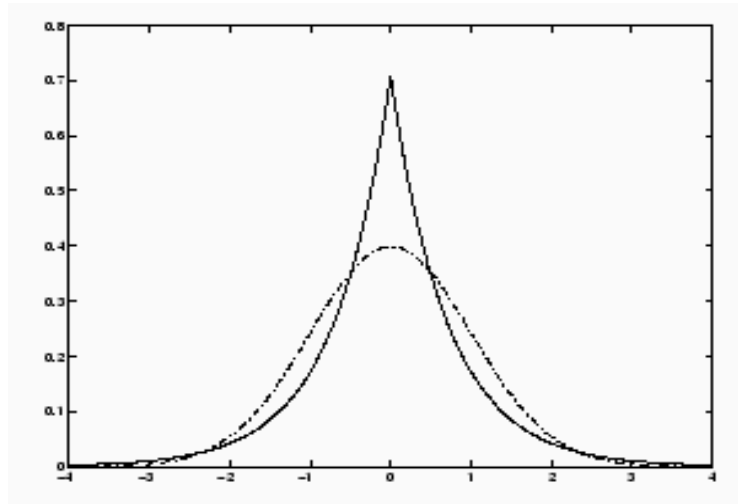
Maksimizasyonu

Maliyet fonksiyonlarının belirlenmesinde bileşenlerin yüksek-derece istatistikleri de kullanılabilir. Kurtosis, sinyalin ilgili dördüncü derece momentini

kullanarak sinyalin Gaussian derecesi hakkında bilgi veren önemli bir istatistiktir[13,19,20,23]. Rassal $s(t)$ sinyali için kurtosis tanımı:

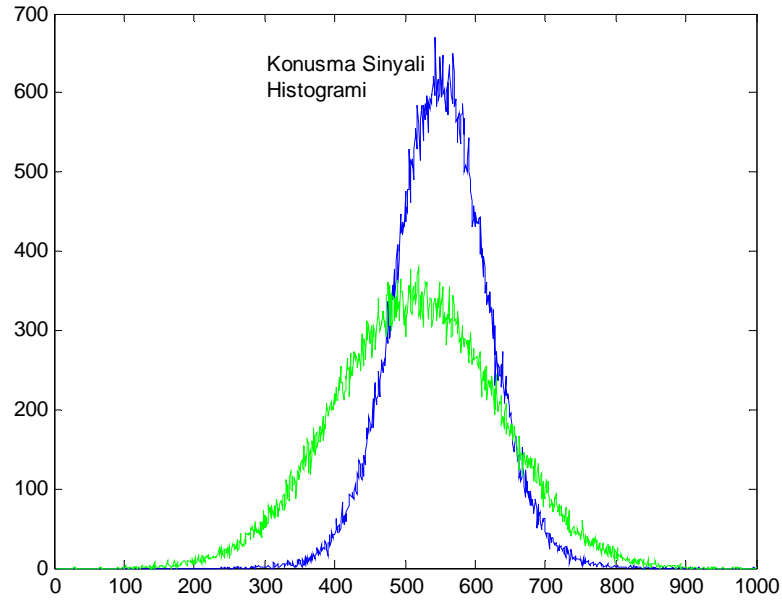
$$\kappa_4 [s(t)] = \frac{E\{s(t)^4\}}{E^2\{s(t)^2\}} - 3 \quad (2.13)$$

şeklindedir. $S(t)$ sinyalinin Gaussian olması durumunda kurtosis 0'a eşittir. Negatif kurtosise sahip sinyal sub-Gaussian, pozitif kurtosise sahip sinyal süper-Gaussian sinyal olarak adlandırılır. Sub-Gaussian sinyallerin olasılık dağılım fonksiyonları Gaussian sinyallere göre daha yassıdır. Düzgün dağılıma sahip herhangi bir rassal sinyal sub-Gaussian sinyaldir. Laplace olasılık dağılım fonksiyonu süper-Gaussian dağılıma örnek verilebilir. Şekil 2.5'de Gauss dağılımı ve Laplace dağılımı verilmiştir. Süper-Gaussian sinyallerin de dağılımları uzun kuyruğa sahiptir ve daha keskin tepeleri vardır.



Şekil 2.5. Gauss Dağılımı ve Süper Gaussian Dağılımı: Kesikli çizgili dağılım Gauss dağılımıdır. Süper-Gaussian Laplace dağılımının daha uzun kuyruğu ve daha keskin tepesi vardır.

Konuşma ve müzik sinyalleri süper-Gaussian sinyallerdir. Şekil 2.6.'da konuşma sinyali ile Gauss dağılımına sahip rassal bir sinyalin histogramları görülmektedir.



Şekil 2.6. Gauss dağılımına sahip rassal bir sinyal ve konuşma sinyali histogramları

Aynı işaretli kurtosisine sahip rassal sinyallerin karışımları, Gaussian dağılımına sahip sinyallere yaklaşır. Farklı bir ifadeyle, süper-Gaussian kaynak sinyallerin karışımlarının kurtosisleri, kaynak sinyallerinin kendi kurtosislerinden daha küçüktür. Aynı şekilde, sub-Gaussian kaynak sinyallerin karışımlarının kurtosisleri, kaynak sinyallerinin kendi kurtosislerinden daha büyüktür. Kısaca, kaynak sinyallerinin karışımlarının kurtosisleri 0'a daha yakındır. Bu karışımların Gaussian sinyale daha yakın olduğunu gösterir.

Bağımsız süper-Gaussian ve ya sub-Gaussian sinyallerin karışımlarını ayrıştırmak için çıkış bileşenlerinin dağılımlarını mümkün olduğunca Gaussian dağılımdan uzak olmasını sağlamak gerekir. Bunun için de süper-Gaussian sinyaller için kurtosis maksimizasyonu, sub-Gaussian sinyaller için de kurtosis minimizasyonu yapılır. Böylece, çıkış bileşenlerinin mümkün olduğunca kaynak sinyallerine yakınsaması yani bağımsız olması sağlanır.

Maliyet fonksiyonlarının oluşturulmasında kullanılan diğer bir yaklaşım ise negentropidir[6,19,24]. $H(s)$ rassal s vektörünün entropisi olmak üzere; negentropi şu şekilde tanımlanır.

$$J(s) = H(s_{Gauss}) - H(s) \quad (2.14)$$

Burada s_{Gauss} Gauss dağılımına sahip ve s vektörüyle aynı ilintiye sahip rassal vektördür[19,24]. Negentropi ancak ve ancak s rassal değişkeni Gauss dağılımına sahip olduğunda sıfır olur. Kaynakların karışımları Gauss dağılımına yakınsayacağından, karışımların negentropisi 0'a yakın olur. Negentropi maksimizasyonu, çıkış bileşenlerini, Gauss dağılımından mümkün olduğunca uzaklaştırır. Böylece, negentropi maksimizasyonu ile çıkış bileşenleri, kaynak sinyallerine mümkün olduğunca yaklaşır.

2.3. Kör Kaynak Ayırıştırma Algoritmaları

Kör Kaynak Ayırıştırma algoritmalarının amacı, kaynakların bağımsızlığının ifade edildiği maliyet fonksiyonlarını optimize etmek, böylece bağımsız bileşenleri, dolayısıyla kaynakların doğru kestirimlerini bulmaktır. Kör Kaynak Ayırıştırma üzerine pek çok algoritma geliştirilmiştir[1-30]. Bu algoritmalar çeşitli teknikleri kullanırlar.

Kör Kaynak Ayırıştırma algoritmalarında en yaygın kullanılan tekniklerden birisi Yapay Sinir Ağları yaklaşımlarıdır. Ağa giriş sinyalleri olarak, algılayıcılardan alınan karışık sinyaller kullanılır ve bileşenlerin bağımsızlığı sağlanana kadar ayırıştırma matrisi W ağırlık katsayılarının güncellenmesi sağlanır. İleri beslemeli ve geri beslemeli yapay sinir ağları yaklaşımları ağırlık katsayılarının güncellenmesinde kullanılmıştır. Matematiksel açıdan en uygun Bağımsız Bileşen Analizi, Pierre Comon [27] tarafından geliştirilmiştir. Bu yöntemde yüksek dereceli istatistik kullanarak optimizasyon sağlanmıştır. Bağımsız kaynak sinyalleri, gözlemlenen sinyalleri oluşturan temel kümeler olarak varsayıp, tanım kümelerinin ayrışımını yapılmıştır. Bunu sağlamak için optimizasyon yönteminde üçüncü ve dördüncü logaritmik momentleri temel alınmıştır. Kör Kaynak Ayırıştırma yaklaşımlarına bir başka yaklaşım da Herault ve Jutten [28] tarafından getirilmiştir. Bu yaklaşım, 'neuromimetic' olarak adlandırılmıştır. Herault&Jutten ağında w_{ii} ağırlık katsayıları bire eşitlenmiştir. Sadece çapraz etkileri sağlayan w_{ij} ağırlık katsayılarının güncellenmesi yapılmaktadır. Burada ayırıştırılan çıkış bileşenleri

$$y_i(t) = x_i(t) - \sum_{j=1, j \neq i}^N w_{ij} s_j(t) \quad (2.15)$$

şeklinde tanımlanır. Doğrusal cebir formunda da

$$\mathbf{y}(t) = \mathbf{x}(t) - W\mathbf{s}(t) \quad (2.16)$$

olarak gösterilir. Ancak burada önemli ve göz önünde bulundurulması gereken bir kısıtlama: N kaynak için N algılayıcı olması gerekliliğidir.

Comon ve Herault&Jutten tarafından ortaya koyulan algoritmalar, sonraları Kör Kaynak Ayırıştırma'da daha değişik algoritmaların geliştirilmesinde yardımcı olmuştur. Bu algoritmalar bilgi teorisi yaklaşımını kullanmış ve gerçek-dünya uygulamalarında da oldukça verimli sonuçlar elde etmişlerdir. Bunlardan en önemlileri Bell ve Sejnowsky ile Amari algoritmalarıdır.

2.3.1. Bell ve Sejnowsky Algoritması

Bell ve Sejnowsky Algoritması, bilgi teorisine dayalı bir öğrenme algoritmasıdır. Bu yaklaşıma göre, ayırıştırma sistemindeki bilgi akışını maksimize etmenin çıkış bileşenlerinin bağımsızlığını sağlayacağını ortaya konulmuştur[2,29]. Bunun için öncelikle giriş ve çıkış bileşenleri arasındaki karşılıklı bilgiyi tanımlamak gerekir:

$$I(\mathbf{G}, \mathbf{Ç}) = H(\mathbf{Ç}) - H(\mathbf{G}|\mathbf{Ç}) \quad (2.17)$$

Bu algoritmada, giriş ve çıkış bileşenleri arasındaki entropi, ayırıştırma sistemine bağlı olmadığından, çıkış bileşenlerinin entropisini maksimize etmenin aslında giriş ve çıkış arasındaki bilgi maksimizasyonu olduğu açıklanmıştır.

1-giriş 1-çıkış durumunda ayırıştırma olmamasına rağmen Bell'in kullandığı çıkış entropi maksimizasyonunu açıklamak için 1-giriş 1-çıkış durumunu kullanabilir. Entropinin tanımı aşağıda tekrar verilmiştir:

$$H(y) = -\int f(y) \log f(y) dv = \langle \log f(y) \rangle \quad (2.18)$$

Burada, $f(y)$ olasılık yoğunluk fonksiyonu, $\langle \cdot \rangle$ beklenti göstergesidir. $y = g(w \cdot x)$ ve $g(\cdot)$ doğrusal olmayan tekdüze fonksiyondur. Amaç, y 'nin entropisini maksimize edecek bir w bulmaktır. $F(y)$ olasılık yoğunluk fonksiyonu x 'in olasılık yoğunluk fonksiyonu cinsinden şöyle ifade edilir.

$$f(y) = \frac{f(x)}{\left| \frac{\partial y}{\partial x} \right|} \quad (2.19)$$

Her iki denklemi kullanarak

$$H(y) = \left\langle \log\left(\frac{\partial y}{\partial x}\right) \right\rangle - \langle \log f(x) \rangle \quad (2.20)$$

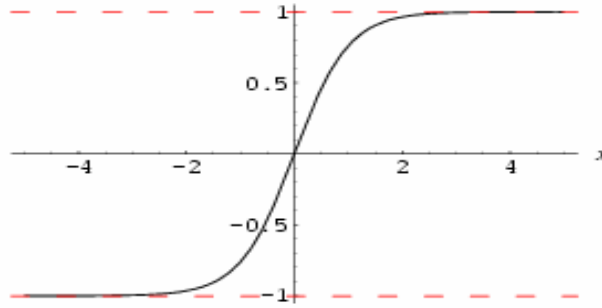
sonucunu elde edilir. Burada x 'in olasılık yoğunluk fonksiyonuna bir etki söz konusu olamayacağından optimizasyon sadece birinci terime göre yapılacaktır.

$\log\left(\frac{\partial y}{\partial x}\right)$ terimini maksimize etmek için bir güncelleme kuralı geliştirilmiştir. Bu

da

$$\Delta w \propto \frac{\partial H(y)}{\partial x} = \frac{\partial}{\partial x} \left(\log \left| \frac{\partial y}{\partial x} \right| \right) = \left(\frac{\partial y}{\partial x} \right)^{-1} \frac{\partial}{\partial w} \left(\frac{\partial y}{\partial x} \right) \quad (2.21)$$

şeklindedir. $g(\cdot)$ fonksiyonu olarak hiperbolik tanjant kullanılabilir. Şekil 2.7.'de görüldüğü gibi, hiperbolik tanjant doğrusal olmayan bir fonksiyondur ve yapay sinir ağlarında sıklıkla aktivasyon fonksiyonu olarak kullanılır.



Şekil 2.7. Hiperbolik Tanjant Fonksiyonu

$g(\cdot)$ fonksiyonu olarak hiperbolik tanjant kullanıldığında

$$\Delta w \propto \frac{1}{w} - 2yx \quad (2.22)$$

kuralı elde edilir.

Bu kuralı, N kaynak N karışım durumunda kullanabilmek için aynı şekilde, $\mathbf{y} = g(\mathbf{W} \cdot \mathbf{x})$ ve $g(\cdot)$ doğrusal olmayan tekdüze fonksiyon belirlemeleri yapılır. Ancak burada \mathbf{x} ve \mathbf{y} rassal değişken değil vektör ve \mathbf{W} ağırlık katsayısı

değil matrisidir. N kaynak N karışım durumunda $f(y) = \frac{f(x)}{|J|}$ şeklinde

Jacobian kullanılır.

$$J = \det \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \dots & \frac{\partial y_n}{\partial x_n} \end{pmatrix} \quad (2.23)$$

1-giriş 1-çıkış durumuyla aynı şekilde J'ye göre maksimizasyon yapılırsa

$$\Delta W \propto [W]^{-1} - 2y.x \quad (2.24)$$

kuralı elde edilir. Bu kural, Bell'in yaklaşımıyla elde edilen maliyet fonksiyonunu optimize etmek için ayrıştırma matrisi W'nın güncellenmesinde kullanılır. Bell algoritması 10 kaynak 10 karışım durumuna kadar tatmin edici sonuçlar vermektedir.

2.3.2. Amari Algoritması

Amari, Kullback-Liebner İraksaklığını başlangıç noktası olarak kullanarak benzer şekilde bir güncelleme kuralı geliştirmiştir. Bileşenlerin bağımsız olması için bileşik entropi, bileşenlerin ayrı ayrı entropileri çarpımına eşit olmalıdır. Dolayısıyla, bileşenlerin ayrı ayrı entropileri çarpımıyla çıkış bileşenlerinden oluşan vektörün bileşik entropisinin iraksaklığını minimize etmek, bileşenlerin bağımsızlığını sağlamış olur[2, 10]. Bu iraksaklık için önce bileşik entropiyle, bileşenlerin entropilerinin çarpımı arasındaki iraksaklığı tanımlamak gerekir.

$$K(W) = \int P(y) \log \frac{P(y)}{\prod_{i=1}^N P(y_i)} dy \quad (2.25)$$

Burada y çıkış bileşenlerinden oluşan vektördür. Entropinin tanımını kullanarak bu iraksaklık

$$K(W) = \sum_{i=1}^N H(y_i) - H(y) \quad (2.26)$$

şeklinde de yazılır. Kırpılmış Gram-Charlier açılımını kullanarak $H(y_i)$ bileşenlerin tekli entropileri şu şekilde ifade edilir.

$$H(y_i) \approx \frac{1}{2} \log(2\pi e) - \frac{\kappa_3(i)^2}{2.3!} - \frac{\kappa_4(i)^2}{2.4!} + \frac{5}{8} \kappa_3(i)^2 \kappa_4(i) + \frac{1}{16} \kappa_4(i)^3 \quad (2.27)$$

Burada $\kappa_a(i)$, i 'inci çıkış bileşeninin a 'ıncı momentidir.

$H(y) = -H(x) + \log|\det(W)|$ denklemini ve yukarıdaki denklemi kullanarak

$$K(W) \approx -H(x) - \log|\det(W)| + \frac{N}{2} \log(2\pi e) - \sum_{i=1}^N \left[\frac{\kappa_3(i)^2}{2.3!} - \frac{\kappa_4(i)^2}{2.4!} + \frac{5}{8} \kappa_3(i)^2 \kappa_4(i) + \frac{1}{16} \kappa_4(i)^3 \right] \quad (2.28)$$

elde edilir. Bu minimize edilmeye çalışılan maliyet fonksiyonudur.

Bu fonksiyonun gradyanını bulmak için $\frac{\partial K(W)}{\partial w_{rc}}$, 'yi bulmamız gerekir. Burada w_{rc}

W ayrıştırma matrisinin r 'inci satırın c 'inci elemanıdır. Bu yaklaşımla

$$\frac{\partial K(W)}{\partial w_{rc}} = -q_{rc} + f(\kappa_3((r), \kappa_4(r))) \cdot \langle y_r^2 \cdot x_c \rangle + g(\kappa_3((r), \kappa_4(r))) \cdot \langle y_r^3 \cdot x_c \rangle \quad (2.29)$$

sonucuna varılır. Burada q_{rc} $Q = [W^T J]^{-1}$ matrisinin elemanlarıdır ve $f(\cdot)$ ve $g(\cdot)$ fonksiyonları şu şekilde tanımlanır:

$$f(x, y) = -\frac{1}{2}x + \frac{15}{4}xy, g(x, y) = -\frac{1}{6}y + \frac{5}{2}x^2 + \frac{3}{2}y^2 \quad (2.30)$$

Bu algoritma beklenen değer ve momentlerin yerine anlık değerlerini koyulduğunda:

$$\nabla K(W) = [W^T J]^{-1} - f(y) \cdot y^T \quad (2.31)$$

elde edilir. Burada $f(\cdot)$ aktivasyon fonksiyonudur. Amari bu fonksiyonu

$$f(x) = \frac{3}{4}x^{11} + \frac{25}{4}x^9 - \frac{14}{3}x^7 - \frac{47}{4}x^5 + \frac{29}{4}x^3 \quad (2.32)$$

şeklinde tanımlamıştır. Ancak sonraki çalışmalarda, Amari'nin aktivasyon fonksiyonu yerine hiperbolik tanjant kullanmanın daha kararlı sonuçlar verdiği görülmüştür. Çünkü Amari'nin aktivasyon fonksiyonu sınırlı değildir ve yüksek öğrenme oranları nümerik taşmalara yol açar.

Maliyet fonksiyonunun gradyanı Öklid uzayında en dik yönü verir. Amari'nin çalışmalarında[10] belirttiği ayrı bir nokta da, bu maliyet fonksiyonun gradyanının en dik yönü verdiği uzayın Reimannian uzayı olduğudur. Buna göre gradyan azalması yöntemi genelde Öklid gradyanı ve bulunulan uzayın ölçevidin çarpımı olarak tanımlanır.

$$\nabla_R f = G^{-1} \nabla_E f \quad (2.33)$$

Burada ∇_R Riermann gradyanı, ∇_E Öklid gradyanı, G Riermann uzayının ölçavidir. Amari bu problemde Riermann uzay ölçevi yerine WW^T sağ çarpımının koyulabileceğini gösterir.

$$\begin{aligned}\nabla_R K(W) &= G^{-1} \cdot \nabla_E K(W) = \nabla_E K(W) \cdot WW^T \\ &= [W^T]^{-1} - f(y) \cdot y^T \cdot WW^T = [I - f(u)u^T] \cdot W\end{aligned}\tag{2.34}$$

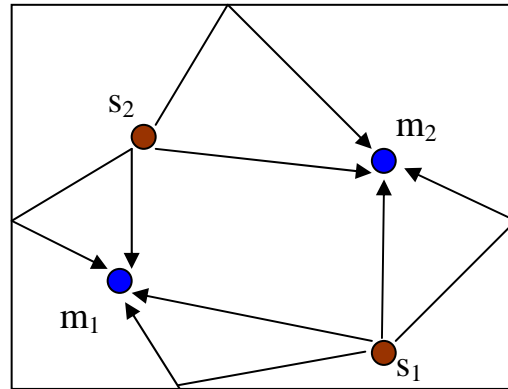
Amari'nin bu öğrenme kuralı yüksek yakınsama oranına sahip ve oldukça verimli bir kuraldır. Riermann uzayı kullanılarak elde edilen kuralla matris ters evirmesi elenmiş olur.

Yukarıda açıklanan her iki algoritma da bilgi teorisi tabanlıdır ve çıkış bileşenlerinin bilgilerini maksimize edilmesi esasına dayanır. Böylece bileşenlerin bağımsızlığı sağlanır. Ancak Amari'nin algoritması çok daha kararludur ve daha yüksek yakınsama oranına sahiptir. Bu algoritma kullanıldığında gerçek-dünya uygulamalarında da oldukça verimli sonuçlar elde edilmektedir.

3. KÖR TERS EVRİŞİM (Blind Deconvolution)

Kör Kaynak Ayırıştırma'nın amacı doğrusal karışmış sinyalleri kullanarak kaynak sinyallerini bulmaktır. Kaynak sinyallerinin, gerçel sayılarla çarpılıp üstdüşümleri alınarak gözlemlenen sinyallerin oluşturulduğu varsayılır. Ancak gözlemlenen sinyallerdeki gecikme, yansıma ve çoklu-yol etkisi göz önüne alındığında Kör Kaynak Ayırıştırma yöntemi yeterli olmamaktadır. Gözlemlenen sinyallerde, gecikme, yansıma ve çoklu-yol etkilerinin olduğu durumlarda uygulanan ayırıştırma yöntemi, Kör Ters Evrişim'dir.

Şekil 3.1.'de gösterildiği gibi, herhangi bir odada iki kaynaktan gelen ses sinyallerinin iki mikrofon tarafından kaydedildiği düşünülürken, mikrofonlar kaynakların doğrusal karışımlarını değil, aynı zamanda duvardan sonsuz yansımalarını da kaydedeceği açıktır. Mikrofonlara kadar olan zaman gecikmeleri ve duvardan yansımaların oluşturduğu çoklu-yol etkileri de göz önüne alınmalıdır. Böylece mikrofonlardan alınan karışımlar farklı frekans karakterlerine sahip olacaktır. Bu yüzden kaydedilen karışımları doğrusal üstdüşümler olarak modellemek iyi bir çözüm değildir. Kör Ters Evrişim yaklaşımında, karışımların evrişimli karışımlar olduğu düşünülür. Karışımların evrişimlenmiş olmasının sebebi, gözlemlenen karışık sinyallerin, kaynak sinyallerinin gözlemciye ulaşmaya kadar olan gecikmesi, duvarlardan olan yansımalar ve farklı yansımaların çoklu-yol etkilerini de içermesidir. Bu yüzden, evrişimlenmiş karışımları modellemek için ortamın dürtü yanıtını da göz önünde bulundurmak gerekir.



Şekil 3.1. Evrişimli Karışımlar Durumu

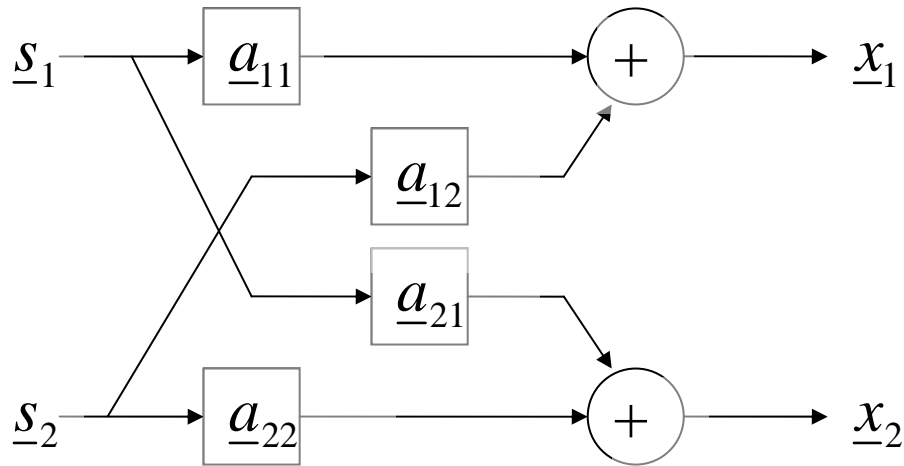
Mikrofonlardan gözlemlenen sinyallerin frekans karakteristikleri, ortamın etkilerinden dolayı filtreden geçmiş gibi olacaktır[5]. Aslında herhangi bir oda, Sonsuz Dürtü Yanıtlı bir sistemdir. Odanın yanıtı sonsuz yansımaları içermesine rağmen, orijinal sinyallerin 60dB altındaki sinyaller ihmal edilebilir[2]. Bu nedenlerle, problemin karmaşıklığını azaltmak için oda, Sonlu Dürtü Yanıtlı bir sistem olarak düşünülebilir. Buna göre evrişimlenmiş karışımlar

$$x_i(t) = \sum_{j=1}^N \sum_{k=0}^M s_j(t-k)a_{ij}(k) \quad (3.1)$$

şeklinde modellenilir. Burada s_j N adet orijinal kaynak, a_{ij} M dereceli karıştırma matrisi katsayıları ve x_i mikrofonlardan gözlemlenen evrişimlenmiş karışımlardır. Buradaki işlem, evrişim işlemi olduğundan, s_j kaynaklarının Kör Kaynak Ayırıştırma modelindeki gibi ters matris çarpımıyla geri elde edilemeyeceği açıktır.

Bu model, Lambert'in [7] ortaya koyduğu, Sonlu Dürtü Yanıtlı matris cebiri ile açıklanabilir. Sonlu Dürtü Yanıtlı matris cebirinde, matrislerin elemanları zaman serileridir. Bu durumda elemanların çarpımları, elemanların evrişimleri olmaktadır. 2x2 durum için evrişimlenmiş karışım modeli Sonlu Dürtü Yanıtlı matris cebiriyle aşağıdaki gibi ifade edilir.

$$\begin{bmatrix} \underline{a}_{11} & \underline{a}_{12} \\ \underline{a}_{21} & \underline{a}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{s}_1 \\ \underline{s}_2 \end{bmatrix} = \begin{bmatrix} \underline{a}_{11} * \underline{s}_1 + \underline{a}_{12} * \underline{s}_2 \\ \underline{a}_{21} * \underline{s}_1 + \underline{a}_{22} * \underline{s}_2 \end{bmatrix} \quad (3.2)$$



Şekil 3.2. 2x2 Evrişimli Karışımlar Modeli

Şekil 3.2.'de, \underline{a}_{11} , \underline{a}_{22} filtreleri direk filtreler ve \underline{a}_{12} , \underline{a}_{21} filtreleri de çapraz filtrelerdir. Bu modelde Sonlu Dürtü Yanıtlı filtre matrislerinin zaman serileri, altı çizgili olarak gösterilmekte, Sonlu Dürtü Yanıtlı filtre matrisleri de alt çizgili olarak ifade edilmektedir. Sonlu Dürtü Yanıtlı matris cebirini kullanarak karıştırma işlemini $\underline{X} = \underline{A} \cdot \underline{S}$ şeklinde tanımlandığında \underline{A} , Sonlu Dürtü Yanıtlı filtrelerden oluşan karıştırma matrisini belirtir. Sonlu Dürtü Yanıtlı matris cebiriyle basit bir çarpım şeklinde ifade edilir. \underline{A} 'nın elemanları M 'inci dereceden karıştırma filtrelerinin katsayılarıdır.

$$\underline{A} = \begin{bmatrix} \underline{a}_{11}^T & \cdots & \underline{a}_{1n}^T \\ \vdots & \ddots & \vdots \\ \underline{a}_{n1}^T & \cdots & \underline{a}_{nm}^T \end{bmatrix} \quad (3.3)$$

Karıştırma filtreleri, aynı dereceden olmayabilir. Ancak göreceli olarak az dereceli olanlara sıfır eklenerek tüm filtrelerin aynı dereceden olması sağlanır. Evrişim işlemi, Sonlu Dürtü Yanıtlı matris cebiriyle basit bir matris çarpımı haline dönüşür. Aynı şekilde, ters evrişim işlemini de yine matris çarpımıyla aşağıdaki gibi açıklanır:

$$\underline{S} = \underline{A}^{-1} \cdot \underline{X} \quad (3.4)$$

Aşağıda \underline{A} matrisinin evriği \underline{A}^{-1} tanımlanmıştır.

$$\underline{A}^{-1} = \frac{1}{\det(\underline{A})} \underline{G} \quad (3.5)$$

Buradaki $\det(\underline{A})$ ve \underline{G} aşağıdaki şekilde tanımlanır:

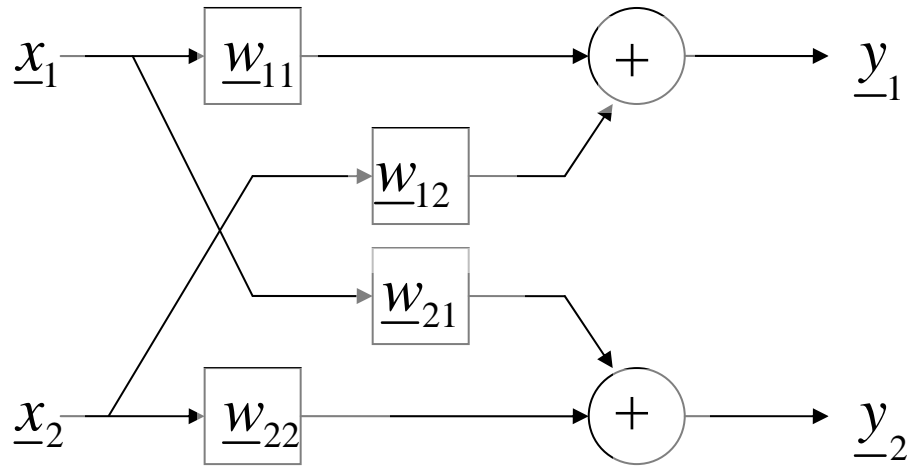
$$\det(\underline{A}_{n \times n}) = \sum_{j=1}^n \underline{a}_{1j} \cdot \underline{A}_{1j} \quad (3.6)$$

$$\underline{G} = \begin{bmatrix} \det(\underline{A}_{11}) & \cdots & \det(\underline{A}_{1n}) \\ \vdots & \ddots & \vdots \\ \det(\underline{A}_{n1}) & \cdots & \det(\underline{A}_{nm}) \end{bmatrix} \quad (3.7)$$

Tüm bu tanımlar frekans uzayı içinde elde edilebilir. Burada önemli olan, zaman uzayında evrişim işleminin, frekans uzayında çarpıma dönüşmesidir.

Kör Ters Evrişim algoritmaları bu ters evrişim işlemini gerçeklemek için kullanılır. Kör Kaynak Ayırıştırma algoritmalarında kullanılan Yapay Sinir Ağları

yaklaşımları, Kör Ters Evrişim algoritmalarına da uygulanmaktadır. Literatürde pek çok Kör Ters Evrişim algoritmaları sunulmuştur. Bunlar, zaman uzayı algoritmaları[12,15,16] ve frekans uzayı algoritmaları[3,4,6,17,18] olmak üzere temelde ikiye ayrılmaktadırlar. Frekans uzayı algoritmaları zaman uzayı algoritmalarına göre daha az karmaşıktır ve daha iyi yakınsama zamanı sağlar. Zaman uzayında evrişimlenmiş sinyallerin ters evrişimi için kullanılan ilk algoritma, Bell ve Sejnowski [29] tarafından ortaya konulmuştur. Doğrusal karışımlar için geliştirdikleri, daha önce bahsedilen algoritma, evrişimlenmiş sinyallerin ters evrişimi için de kullanılabilir. İki-kaynak iki-algılayıcı için ortaya koydukları ağ yapısı Şekil 3.3.'de verilmiştir .



Şekil 3.3. İki kaynak iki algılayıcı için Bell ve Sejnowski ağ yapısı

Şekil 3.3.'de verilmiş olan ağ yapısı ileri beslemeli bir ağ yapısıdır. w_{ij} ayrıştırma filtrelerinin katsayılarıdır. Y_i çıkışları, ayrıştırılmış ve ters evrişimlenmiş sinyalleri, x_i girişleri, kaynakların evrişimleri olan gözlemlenen sinyalleri ifade etmektedir. Yukarıdaki ağın gözlemlenen sinyaller üzerinde yaptığı işlem temelde yine evrişim işlemidir. Bir başka deyişle, ayrıştırma filtreleriyle gözlemlenen sinyaller evriştirilmiş sinyallerdir.

$$\begin{aligned} y_1 &= w_{11} * x_1 + w_{12} * x_2 \\ y_2 &= w_{21} * x_1 + w_{22} * x_2 \end{aligned} \quad (3.8)$$

Amaç, kör kaynak ayrıştırma uygulamalarındaki amaçla aynıdır. Çıkış bileşenlerinin (y_i) bağımsızlığını sağlamak, Kör Ters Evrişim ağlarının da

sağlamaya çalıştığı durumdur. Bell'in ağ yapısı, burada da, giriş ve çıkış bileşenleri arasındaki bilgi akışını maksimize etmeye çalışır. Burada optimizasyon $\langle \ln|J| \rangle$ 'nin maksimize edilmesidir. J , ağın çıkışlarının sigmoid aktivasyon fonksiyonu uygulanmış durumlarının Jacobian'ıdır. Kör Kaynak Ayırıştırma konusunda, Bell ve Sejnowski öğrenme kuralı için yapılan türetme kullanılarak Kör Ters Evrişim algoritmasının öğrenme kuralı bulunur. Bu türetmeleri açıklamak için Burada ilk ağırlık katsayılarının güncellenmesinde, diğer ağırlıkların güncellenmesinden farklı bir öğrenme kuralı uygulanır.

$$\Delta \underline{W}_0 \propto [\underline{W}_0]^{-1} - 2 \tanh(\underline{y}_0) \underline{x}_0^T \quad (3.9)$$

Bu öğrenme kuralı ilk ağırlık katsayıları içindir. Diğer ağırlık katsayıları için

$$\Delta \underline{W}_k \propto -2 \tanh(\underline{y}_k) \underline{x}_k^T, k > 0 \quad (3.10)$$

kuralı uygulanır.

3.1.Frekans Uzayı Algoritmaları

Frekans uzayı algoritmaları zaman uzayı algoritmalarına göre, yakınsama oranında, öğrenme hızında, uygulanabilirlik ve verimlilik konusunda belirgin bir gelişme sağlamıştır. Zaman uzayı algoritmaları, dereceleri az olan karıştırma filtreleri için verimli çalışırken, filtrelerin boyutu arttıkça bazı yakınsama problemleriyle karşılaşmıştır. Gerçek-dünya uygulamaları yüksek boyutlu filtrelerle gerçekleştirilebileceğinden, zaman uzayı algoritmaları bu konuda yetersiz kalmaktadır. Bunun yanında, zaman uzayı algoritmaları, frekans uzayı algoritmalarından daha karmaşık bir yapıya sahiptir. Özellikle filtrelerin boyutları arttıkça zaman uzayı algoritmaları daha da karmaşıklaşır.

Evrişimlenmiş karışımlar, frekans uzayında da, Sonlu Dürtü Yanıtlı matris cebirini kullanarak ifade edilir. Ancak frekans uzayında evrişim işlemi çarpma ve ters evrişim işlemi de bölme işlemine dönüşecektir. Buna göre:

$$\hat{\underline{X}} = \hat{\underline{A}} \cdot \hat{\underline{S}} \quad (3.11)$$

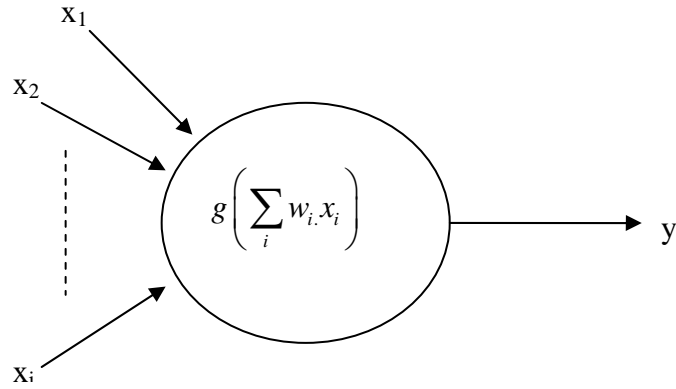
şeklinde yazabiliriz. Kaynakların frekans dönüşümleri ile Sonlu Dürtü Yanıtlı filtre matrisinin frekans dönüşümünün çarpımı, karışımların frekans dönüşümlerini verecektir. Buna göre, 2×2 durumu için:

$$\begin{bmatrix} \hat{a}_{11} & \hat{a}_{12} \\ \hat{a}_{21} & \hat{a}_{22} \end{bmatrix} \cdot \begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \end{bmatrix} = \begin{bmatrix} \hat{a}_{11} \cdot \hat{s}_1 + \hat{a}_{12} \cdot \hat{s}_2 \\ \hat{a}_{21} \cdot \hat{s}_1 + \hat{a}_{22} \cdot \hat{s}_2 \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \quad (3.12)$$

Bu sonuç, evrişimlenmiş sinyal karışımlarını oldukça basit bir duruma getirir ve evrişimlenmiş sinyal karışımlarının frekans uzayında doğrusal karışım olarak matris çarpımıyla ifade edilebileceğini gösterir. Bu yüzden doğrusal Kör Kaynak Ayırıştırma algoritmaları, frekans uzayında, evrişimlenmiş karışımların frekans bölmelerini ayırıştırmada kullanılır[2,3,4]. Bunu sağlamak için frekans uzayı Sonlu Dürtü Yanıtlı filtre ters evrişim matrislerinin ağırlıklarına öğrenme işlemini uygulanır. Burada göz önünde bulundurulması gereken bir nokta, evrişim işleminin çarpma işlemine, ters evrişim işlemi de bölme işlemine dönüştüğüdür. Frekans uzayında anlık ayırıştırma işlemini çarpma işlemiyle aşağıdaki gibi açıklarız:

$$e_f(t) = W_f x_f(t) \quad \forall f, t \quad (3.13)$$

Burada $e_f(t)$, $x_f(t)$ kaynak kestirimlerinin ve gözlemlenen sinyallerin frekans dönüşümleri, W_f frekans uzayında Sonlu Dürtü Yanıtlı filtre ters evrişim matrisidir. Böylece doğrusal Kör Kaynak Ayırıştırma algoritmaları ile frekans uzayında, evrişimlenmiş sinyal karışımlarının frekans bölmeleri ayırıştırmış olur. Frekans uzayında sinyaller, reel sayılar yerine karmaşık sayılarla belirtilir. Bu yüzden, bu uzayda uygulanan yapay sinir ağları, karmaşık nöronlardan oluşmaktadır. Böylece, frekans uzayındaki yapay sinir ağlarına Karmaşık Yapay Sinir Ağları denir. Karmaşık Yapay Sinir Ağları temel birimleri Yapay Sinir Ağlarındaki birimlerle aynıdır. Matematiksel gösterim açısından Karmaşık Yapay Sinir Ağları ile Yapay Sinir Ağları arasında fark yoktur. Ancak Karmaşık Yapay Sinir Ağları birimlerinin ağırlıkları, giriş ve çıkışları reel sayılar yerine, karmaşık sayılardan oluşur. Çıkışların hesaplanması, ağırlıkların atanması ve güncellenmesi karmaşık sayılarla yapılır. Temel bir Karmaşık Yapay Sinir Ağı nöronu Şekil 3.4.'de gösterildiği gibidir.



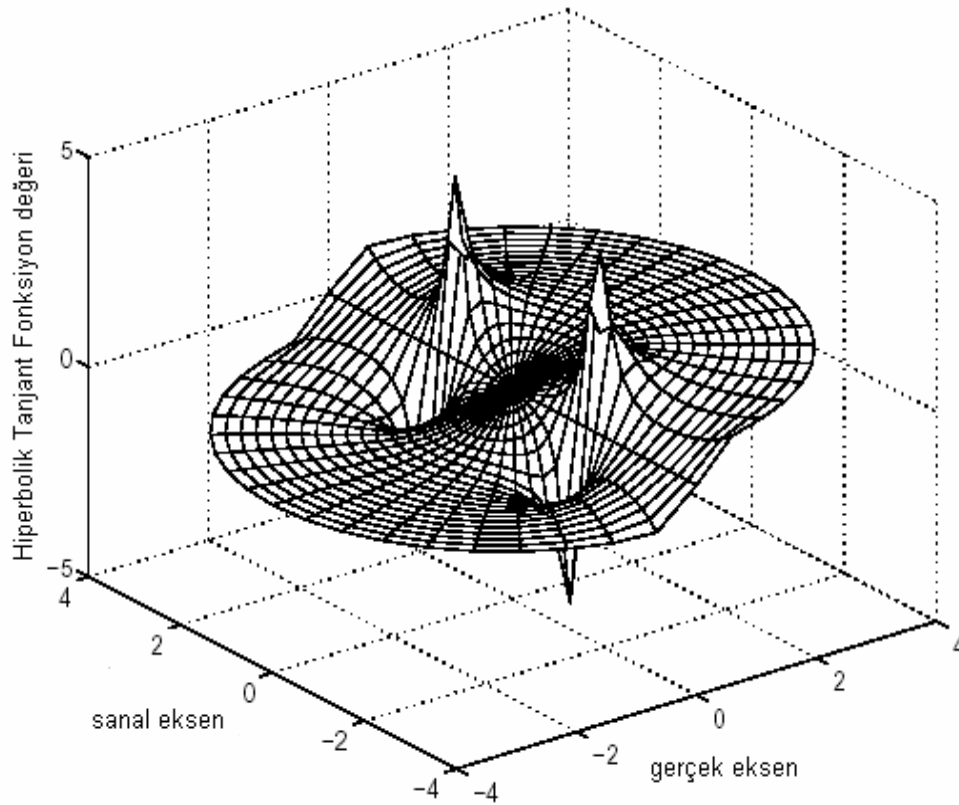
Şekil 3.4. Karmaşık Yapay Sinir Ağları Nöronu

Buradaki $g(\cdot)$ doğrusal olmayan, karmaşık aktivasyon fonksiyonudur. Karmaşık Yapay Sinir Ağları, bu karmaşık nöronların birleştirilmesiyle oluşur. Frekans uzayında yapay sinir ağları ile öğrenmeyi sağlamak için bazı gereklilikler yerine getirilmelidir. Yapay sinir ağlarında aktivasyon fonksiyonu olarak çoğunlukla hiperbolik tanjant fonksiyonu kullanılır. Fakat Karmaşık Yapay Sinir Ağları için bu fonksiyon aktivasyon fonksiyonu olarak uygun değildir. Karmaşık sayılar için hiperbolik tanjant şu şekilde tanımlanır.

$$\tanh(z) = \frac{e^z + e^{-z}}{e^z - e^{-z}} \quad (3.14)$$

Şekil 3.5.'de de görüleceği gibi hiperbolik tanjant fonksiyonu karmaşık uzayda

$z = \left(k + \frac{1}{2}\right)\pi$ noktalarında tanımlı değildir.

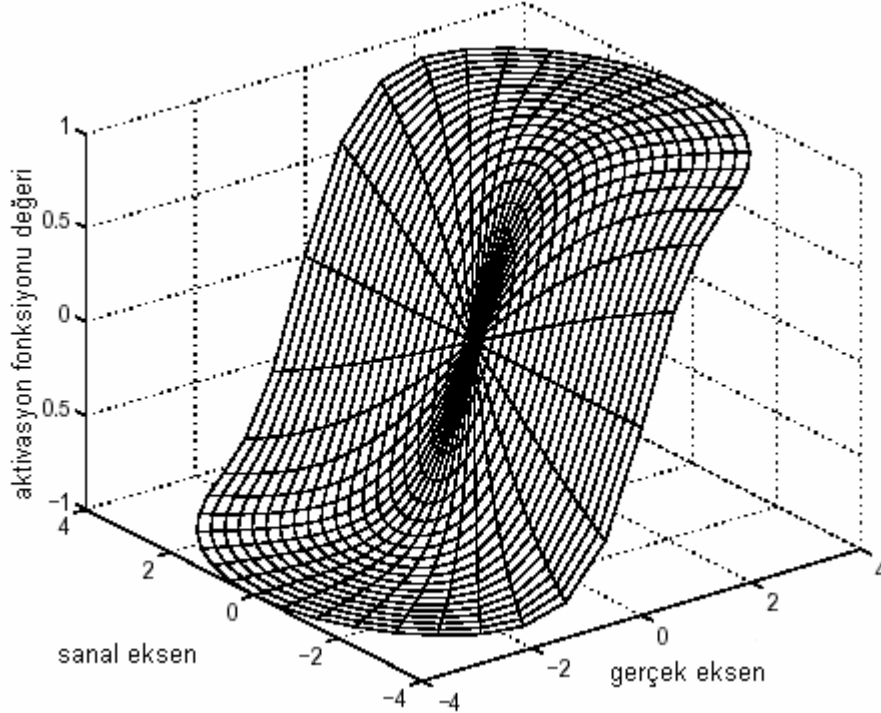


Şekil 3.5. Karmaşık Uzayda Hiperbolik Tanjant Fonksiyonu

Bu noktadaki tekillikler, bazı kararsızlık problemlerine sebep olur ve öğrenme işlemi engeller. Bu istenmeyen durumları yok etmek için Karmaşık Yapay Sinir Ağları'nda kullanılacak $g(\cdot)$ aktivasyon fonksiyonu için bazı koşullar sağlanmalıdır. Öncelikle aktivasyon fonksiyonu $g(\cdot)$ hem gerçek hem de sanal ekseninde doğrusal olmayan bir fonksiyon olmalıdır. Nümerik taşmaları önlemek için $g(\cdot)$ fonksiyonu sınırlı ve kısmi türevlenebilir olmalıdır. $g(\cdot)$ “entire” fonksiyon olmamalıdır. “Entire” fonksiyon karmaşık uzayda her bölgede analitik olan karmaşık fonksiyondur ve karmaşık uzayda sınırlı “entire” fonksiyonlar sabit fonksiyonlardır. Son olarak maliyet fonksiyonunun ağırlıklara göre kısmi türevlerinin toplamı, sıfırdan farklı olmalıdır. Eğer bu koşul sağlanmazsa, hata fonksiyonunun gradyanı sıfır olamayacağından öğrenme işlemi gerçekleşmez. Yukarıda aktivasyon fonksiyonu için belirtilen gereklilikleri sağlayan bazı önermeler sunulmuştur[25,26]. Bunların arasında

$$g(z) = \tanh(\operatorname{Re}\{z\}) + i \cdot \tanh(\operatorname{Im}\{z\}) \quad (3.15)$$

bu tezde kullanılmıştır. Şekil 3.6.'da de görüldüğü gibi, bu aktivasyon fonksiyonu yukarıda bahsedilen gereklilikleri sağlar. Burada karmaşık sayının $\text{Re}\{z\}$ gerçek kısmı, $\text{Im}\{z\}$ sanal kısmıdır.



Şekil 3.6. Karmaşık Aktivasyon Fonksiyonu

Karmaşık Yapay Sinir Ağları'nda kullanılacak aktivasyon fonksiyonunu belirledikten sonra, öğrenme kurallarını da karmaşık uzayda uygulayabilecek şekilde değiştirmek gerekir. Doğrusal Kör Kaynak Ayırıştırma algoritmaları karmaşık uzayda kullanıldığı için matris devirmelerini değiştirmek gerekir. Bunun için Hermitian devirmesi uygundur. Buna göre Amari'nin doğal gradyan kuralı:

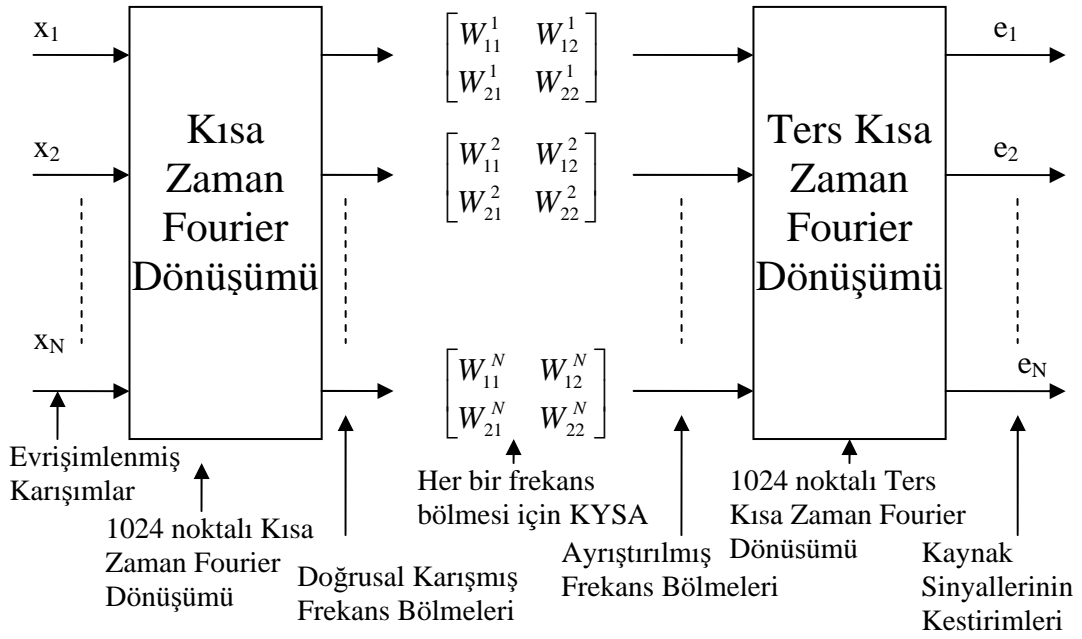
$$\Delta W_f \propto [I - g(e_f)e_f^H] W_f \quad (3.16)$$

şeklinde yazılır.

Burada amaç, doğrusal Kör Kaynak Ayırıştırma algoritmalarını kullanarak karmaşık uzayda her biri anlık karışmış, frekans bölmelerini ayırıştırmak olduğundan, Karmaşık Yapay Sinir Ağları ile her bir frekans bölmesinin Sonlu Dürtü Yanıtlı filtre ağırlıklarının güncellenmesi yapılacaktır. Öncelikle evrişimlenmiş karışımlar Fourier dönüşümü ile frekans uzayına dönüştürülür.

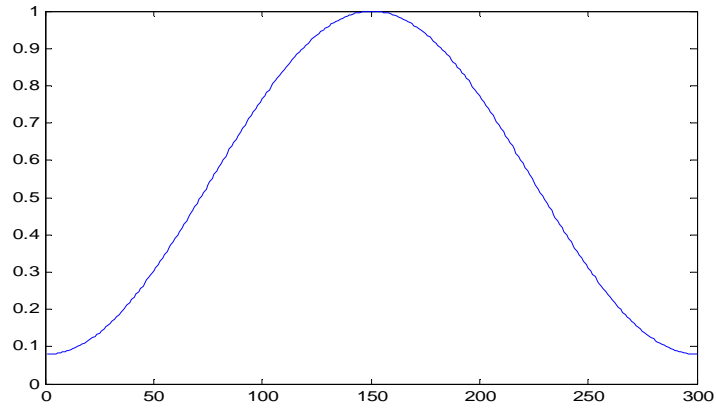
Frekans uzayında, Fourier katsayılarını ayırtmak için her bir frekans bölmesine Karmaşık Yapay Sinir Ağı uygulanır. Bunun için, çok sayıda ağ kullanılması gerekir. Örneğin 2x2 durumu için (2 Kaynak 2 Algılayıcı) frekans uzayında, 2x2 karmaşık ağırlık matrisine sahip Karmaşık Yapay Sinir Ağı, her bir frekans bölmesi için kullanılacaktır. Bu matrislerden oluşan dizi, zaman uzayındaki direk ve çapraz filtrelerin, frekans cevabı matrislerinin karşılığı olacaktır. Temel olarak her bir frekans bölmesine gelen Karmaşık Yapay Sinir Ağı'nın ağırlık matrislerinin elemanları, Sonlu Dürtü Yanıtlı filtre ters evrişim matrislerinin frekans cevabı değerlerine karşılık gelir. Frekans bölmelerinin ayrıştırılmasından sonra, ayrıştırılmış frekans katsayıları, Ters Fourier dönüşüm kullanılarak zaman uzayına dönüştürülür ve zaman uzayında ters evrişimlenmiş kaynak kestirimleri elde edilir.

Şekil 3.7.'de Frekans Uzayı Kör Ters Evrişim Algoritması temel olarak verilmiştir. Bu algoritmanın gerçekleştirilmesinde sinyallerin frekans uzayına dönüşümünde ve zaman uzayına ters dönüşümünde Kısa Zaman Fourier Dönüşümü ve Kısa Zaman Ters Fourier Dönüşümü kullanılmıştır. Kısa Zaman Fourier Dönüşümü'nde frekans bölmelerinin sayısı Sonlu Dürtü Yanıtlı filtre ters evrişim matrisinin derecesine eşit alınmaktadır. Bunun nedeni; anlık Kör Kaynak Ayrıştırma bir matris ile her frekans bölmesine uygulanacaktır.



Şekil 3.7. Frekans Uzayı Kör Ters Evrişim Algoritması Çizeneği

Yakınsama zamanı açısından frekans uzayı algoritmaları, zaman uzayı algoritmalarından daha verimli olmalarına rağmen dönüşümlerde bazı problemlerle karşılaşmaktadır. Bunlardan biri “circularity” problemidir[14]. “Circularity” problemi frekans uzayında ayrıştırılmış sinyallerin zaman uzayına dönüşümünden sonra kestirimlerin bazı noktalarında tekilliklere yol açmaktadır. Bu problemin çözümünde frekans uzayında Kör Kaynak Ayrıştırma algoritmasında kullanılan karışmış sinyallerin frekans spektrumunu düzlemek kullanılabilir. Spektral zarfı düzlemek için Hamming penceresi gibi uygun bir pencerenin Kısa Zaman Fourier Dönüşümü’nden önce uygulanması gerekir. Şekil 3.8.’de Hamming penceresi verilmiştir.



Şekil 3.8. Hamming Penceresi

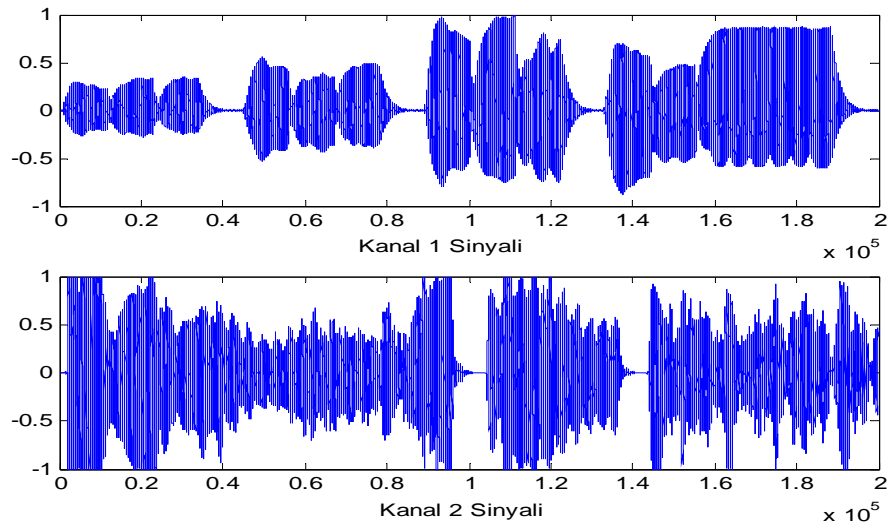
Diğer bir metot da, sinyallerin frekans uzayına dönüşümünden önce zaman uzayında başına ve sonuna sıfır ekleme metodudur. Ayırıştırma frekans uzayında doğrusal Kör Kaynak Ayırıştırma algoritmasının her bir frekans bölmesine uygulanmasıyla ve sonuçların Ters Kısa Zaman Fourier Dönüşümü ile tekrar zaman uzayına dönüşümüyle sağlanır. Ölçekleme ve permütasyon problemleri frekans uzayında da Kör Kaynak Ayırıştırma algoritmaları gibi iki temel problem olarak tekrar ortaya çıkmaktadır. Ölçekleme problemi, her bir frekans bölmesini D'nin köşegen elemanlarıyla farklı ölçekleyerek spektral bir bozuma yol açar. Bu sorun, ayırıştırma matrisinin determinantının her zaman "1"e eşit olmasını sağlayarak aşılabilir[4]. Permütasyon problemi ise ölçekleme probleminden daha karmaşık bir problemdir[11]. Bu problemin çözümü, öğrenme parametrelerinin dikkatlice seçimiyle, azalan öğrenme oranı ve momentum kullanımıyla sağlanabilir[3]. Frekans uzayında permütasyon problemini çözmek için ayırıştırma filtrelerini çevrim-dışı hesaplayıp, bir filtreyi bir frekans bölmesine uyguladıktan sonra diğer frekans bölmesi için hesaplamalara geçerken bir önceki bölme için güncellenen ağırlıkları bu bölmenin başlangıç değerleri yapılır. Öğrenme işleminin sonunda ayırıştırılmış frekans bölmeleri Ters Kısa Zaman Fourier Dönüşümü ile tekrar zaman uzayına dönüşümü sağlanır.

4. UYGULAMA

Bu çalışmada, müzik-müzik karışımlarının Kör Ters Evrişim yöntemiyle ayrıştırılması hedeflenmiştir. Bunun için, Kör Kaynak Ayrıştırma algoritmalarının frekans uzayına adaptasyonu ile gerçekleştirilen Frekans Uzayı Kör Ters Evrişim Algoritması kullanılmıştır. Bu yüzden, öncelikle, Kör Kaynak Ayrıştırma algoritmaları ile doğrusal karışmış sinyallerin ayrışımı gerçekleştirilmiştir.

4.1. Kör Kaynak Ayrıştırma Algoritmalarının Gerçeklenmesi

Kör Kaynak Ayrıştırma algoritmalarının uygulanmasında, kaynak sinyalleri olarak öncelikle, iki farklı çalgıdan kaydedilen sinyaller kullanılmıştır. Bu çalgılar çello ve kemandır. Bu kaynak sinyalleri Şekil 4.1.'de verilmiştir. Kaynak sinyallerinin sayısı daha fazla olabilir ancak bu çalışmadaki algoritmaların gerçekleştirilmesinde, iki kaynak sinyali karıştırılmıştır.

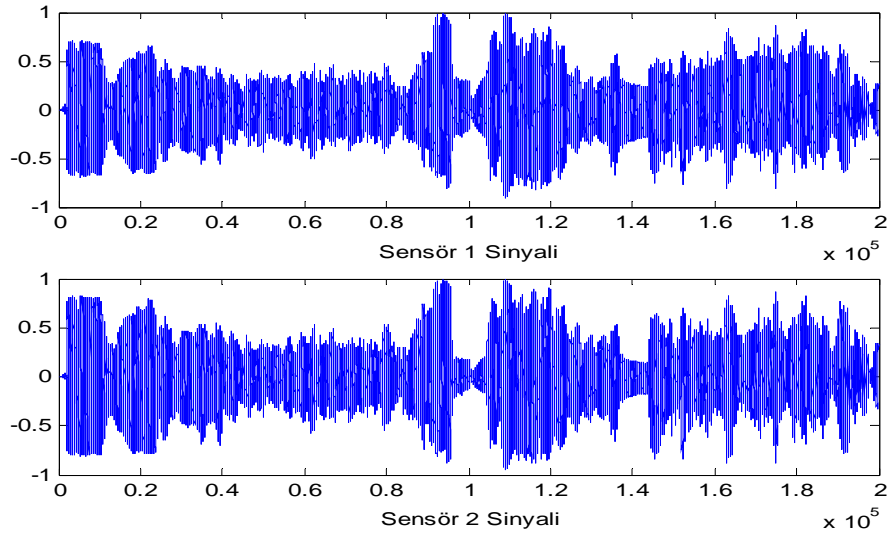


Şekil 4.1.: Çello ve keman çalgılarından kaydedilen kaynak 1 ve kaynak 2 sinyalleri

Kör Kaynak Ayrıştırma algoritmaları, doğrusal karışmış kaynak sinyallerini ayrıştıracağından, bu iki kaynak sinyalinin doğrusal üstdüşümleri oluşturulmuştur ve bunlar, algılayıcı sinyalleri olarak kullanılmıştır. Bunun için bir karıştırma matrisi A belirlenerek, bu iki kaynak sinyalinin karışması sağlanmıştır. Rasgele sayılardan oluşan 2×2 A matrisi

$$A = \begin{bmatrix} 0.1389 & 0.1987 \\ 0.2028 & 0.6038 \end{bmatrix}$$

olarak atanmıştır. A karıştırma matrisi ile kaynak sinyallerinden oluşan s vektörü çarpılarak x gözlemlenen algılayıcı sinyalleri bulunmuştur. Şekil 4.2.'de bu iki algılayıcı sinyali görülebilir.



Şekil 4.2.: Kaynakların doğrusal üstdüşümlerinden oluşan algılayıcı sinyalleri

A 'nın determinantının 0'dan farklı olması gereklidir. Bu, Bağımsız Bileşen Analizi konusunda belirtilen gereklilikleri sağlamak içindir. Diğer bir gereklilik de kaynak sinyallerinin Gaussian sinyaller olmamasıdır. Kaynak sinyallerinin kurtosisleri hesaplanarak, sinyallerin Gaussian dereceleri bulunabilmektedir. Daha önce belirtildiği gibi, müzik ve konuşma sinyalleri super-Gaussian sinyalleridir ve bu sinyallerin kurtosisleri 0'dan büyüktür. Bu örnekte kaynakların kurtosis değerleri Kanal 1 sinyali için 1.8596, Kanal 2 sinyali için 0.8312 olarak hesaplanmıştır. Böylece Bağımsız Bileşen Analizi ile kaynakları ayrıştırabilmek için gerekli koşullar sağlanmıştır ve deney sonucunda bu sinyallerin ayrıştırılması beklenmektedir.

Kör Kaynak Ayrıştırma yaklaşımının amacı, sadece bu iki algılayıcı sinyalini kullanarak kaynak sinyallerini geri elde etmektir. Karıştırma matrisi ve kaynaklar ile ilgili herhangi bir önveri kullanılmamaktadır. Kaynak sinyallerinin

Gaussian derecesi, karışmış algılayıcı sinyallerinin kurtosislerine bakılarak bulunabilir. Algılayıcı sinyalleri yine super-Gaussian sinyallerdir ancak kaynak sinyallerine göre Gaussian sinyallere daha yakınlardır. Başka bir deyişle; algılayıcı sinyallerinin kurtosisleri, kaynak sinyallerinin kurtosislerinden daha küçük olması beklenmektedir.. Algılayıcı 1 sinyalinin kurtosisi 0.6732, algılayıcı 2 sinyalinin kurtosisi 0.7753 olarak hesaplanmıştır. Algılayıcı sinyallerinin kurtosis değerleri, A karıştırma matrisinin elemanlarına bağlı olarak değişebilmektedir.

İlk ayrıştırma deneylerinde, farklı A karıştırma matrisleri kullanılarak hem Amari hem de Bell'in ayrıştırma algoritmaları kullanılmıştır. Amari'nin algoritmasında ayrıştırma matrisi W 'nin ağırlıklarının güncellenmesinde

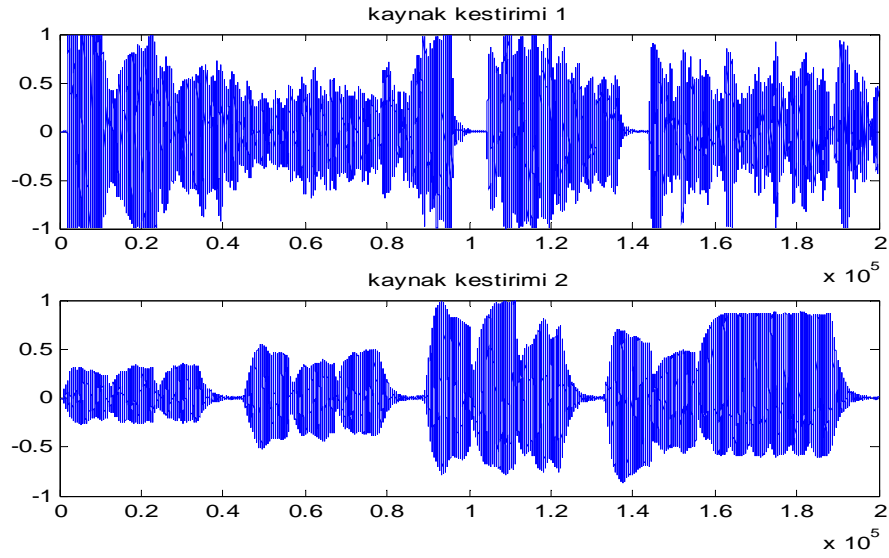
$$W = W + \eta [I - f(\mathbf{y})\mathbf{y}^T] W \quad (4.1)$$

kuralı uygulanmıştır. Burada, \mathbf{y} çıkış bileşenlerinin(kaynak kestirimleri) oluşturduğu vektör ve $f(\mathbf{y})$ aktivasyon fonksiyonudur. Aktivasyon fonksiyonu olarak $\tanh(\cdot)$ kullanılmıştır. Denklem 4.1'de η öğrenme oranıdır ve sabit bir değerdir. Daha önceki deneysel çalışmalarla, öğrenme oranı, konuşma ve müzik sinyalleri için 0.01 olarak belirlenmiştir[6]. Öğrenme oranının daha önce açıklandığı üzere algoritmanın doğru ayrıştırmaları verebilmesi açısından oldukça önemi vardır. Öğrenme oranının küçük olması durumunda yakınsama zamanında ciddi bir artış olacak, büyük olması durumunda da öğrenme kuralı optimum W ağırlıklarını yakalayacak ve öğrenme gerçekleşmeyecektir.

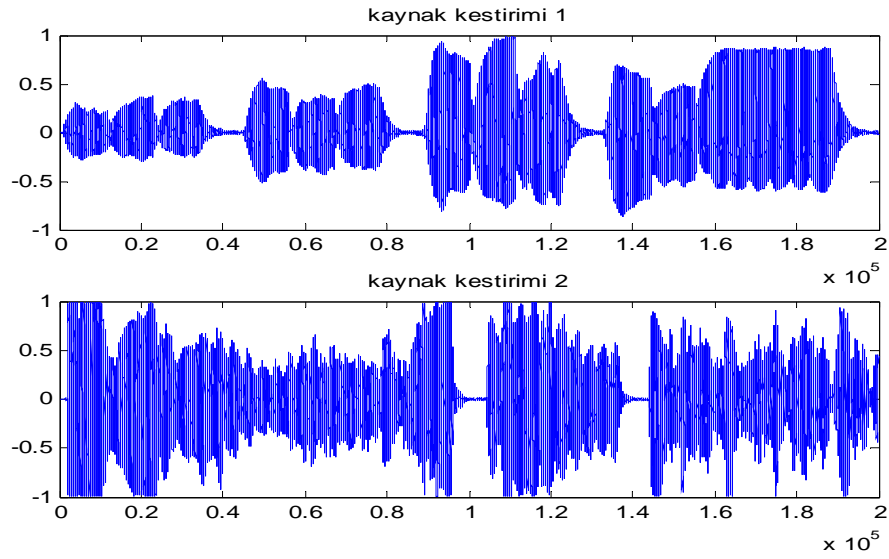
Bell algoritmasının öğrenme kuralı ise aşağıda verilmiştir:

$$W = W + \eta ([W]^{-1} - 2f(\mathbf{y})\cdot\mathbf{x}) \quad (4.2)$$

Amari'nin algoritmasıyla aynı şekilde burada da η öğrenme oranı, $f(\mathbf{y})$ aktivasyon fonksiyonu, \mathbf{y} çıkış bileşenlerinden(kaynak kestirimleri) oluşan vektördür. \mathbf{x} ise giriş bileşenlerinin(algılayıcı sinyalleri) oluşturduğu vektördür. Bell'in algoritmasında da öğrenme oranı, η değerine, 0.01 atanmıştır. Aktivasyon fonksiyonu olarak Amari'nin algoritmasında olduğu gibi $\tanh(\cdot)$ kullanılmıştır.



Şekil 4.3.: Amari algoritması kaynak kestirimleri

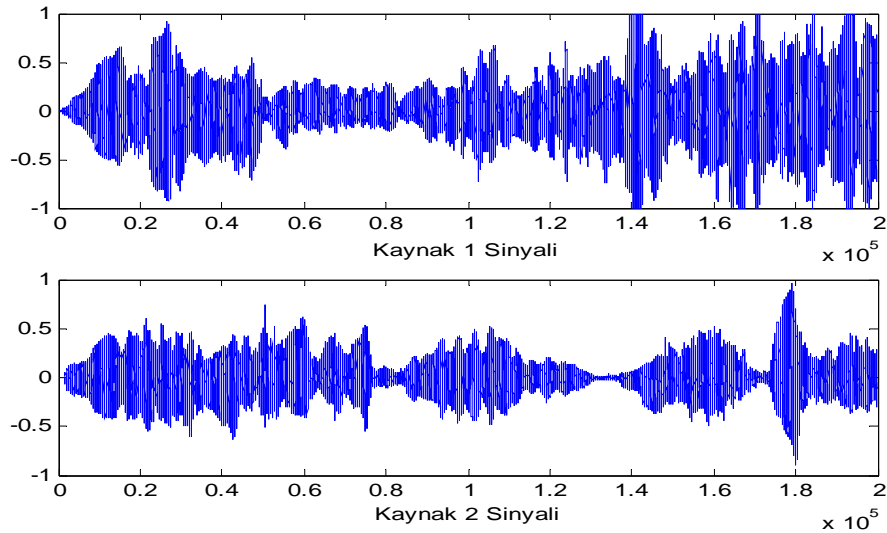


Şekil 4.4.: Bell algoritması kaynak kestirimleri

Şekil 4.3. ve Şekil 4.4.'de, Amari'nin ve Bell'in algoritmalarının kaynak kestirimleri verilmiştir. Daha önceki konularda belirtildiği gibi, permütasyon belirsizliğine bağlı olarak, farklı kaynaklar farklı çıkış bileşeni olarak kestirilebilecektir. Aynı zamanda çıkış bileşenleri ölçeklenmiş olarak bulunabilmektedir. Bu deneylerde farklı çalgılardan kaydedilen kaynak sinyallerinin, doğrusal karışımlarını ayrıştırmada, her iki algoritma da başarılı sonuçlar vermiştir. Ancak sonuçlar karşılaştırıldığında Amari'nin algoritması,

yakınsama zamanı açısından daha başarılı olduğu gözlenmiştir. Elbette karıştırma matrisi ve ayrıştırma matrisine atanan başlangıç değerlerine göre yakınsama zamanları her iki algoritma için de değişmektedir. Ancak, bu durum için Bell'in algoritması karıştırılmış sinyal verilerinin 70%'inin kullanılmasıyla ayrıştırılırken, Amari'nin algoritmasında tam ayrıştırma, verinin 15%'inin kullanılmasıyla sağlanmıştır.

Diğer bir deneyde, bu algoritmalar farklı kaynak sinyalleri için denenmiştir. Burada, aynı çalgıdan kaydedilen kaynak sinyalleri, rasgele A karıştırma matrisi kullanılarak karıştırılmıştır. Şekil 4.5.'de bu deneyde kullanılan Viola çalgısından kaydedilen iki farklı kaynak sinyali görülmektedir. Kaynak sinyallerinin kurtosis değerleri Kaynak 1 için 1.9341, Kaynak 2 için 2.0014 olarak hesaplanmıştır.



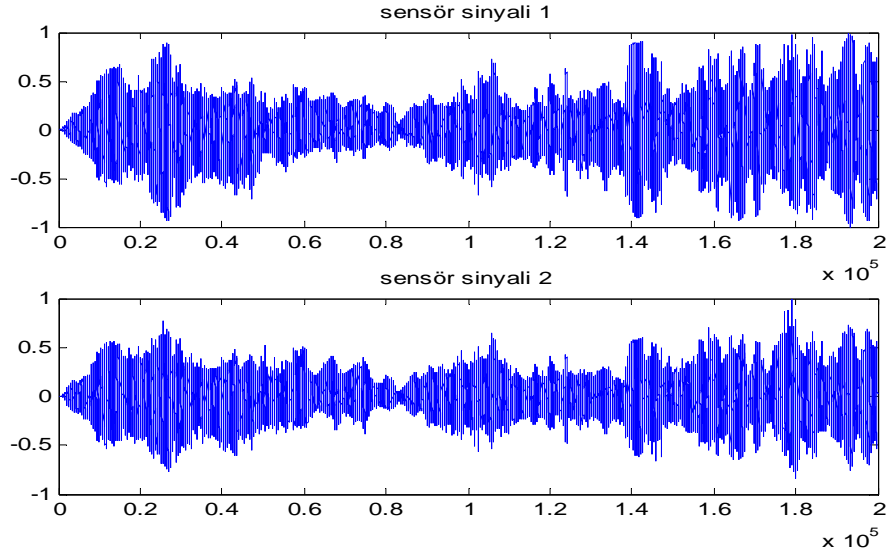
Şekil 4.5.: Viola çalgısından kaydedilen iki farklı melodi kaynağı 1 ve kaynağı 2 sinyalleri olarak kullanılmıştır.

A karıştırma matrisi

$$A = \begin{bmatrix} 0.8318 & 0.7095 \\ 0.5028 & 0.4289 \end{bmatrix}$$

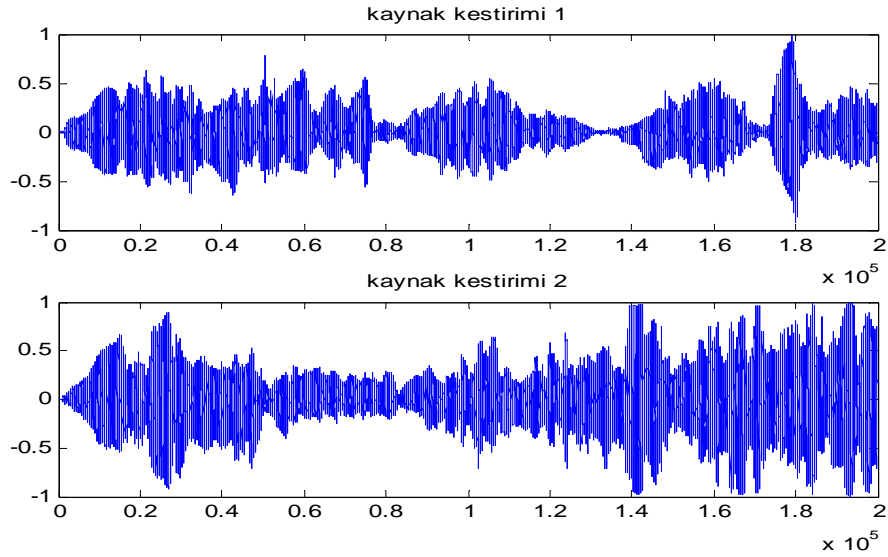
olarak belirlenmiştir. Bu karıştırma matrisiyle oluşturulan algılayıcı sinyalleri, Şekil 4.6.'da verilmiştir. Kaynakların doğrusal karışık durumları olan algılayıcı sinyallerinin kurtosis değerleri, 1. algılayıcı sinyali için 1.5212, ikinci algılayıcı sinyali için 1.5464 olarak hesaplanmıştır. Daha öncede belirtildiği gibi, algılayıcı

sinyalleri yine super-Gaussian sinyallerdir ancak Gaussian dereceleri kaynak sinyallerinden küçüktür.

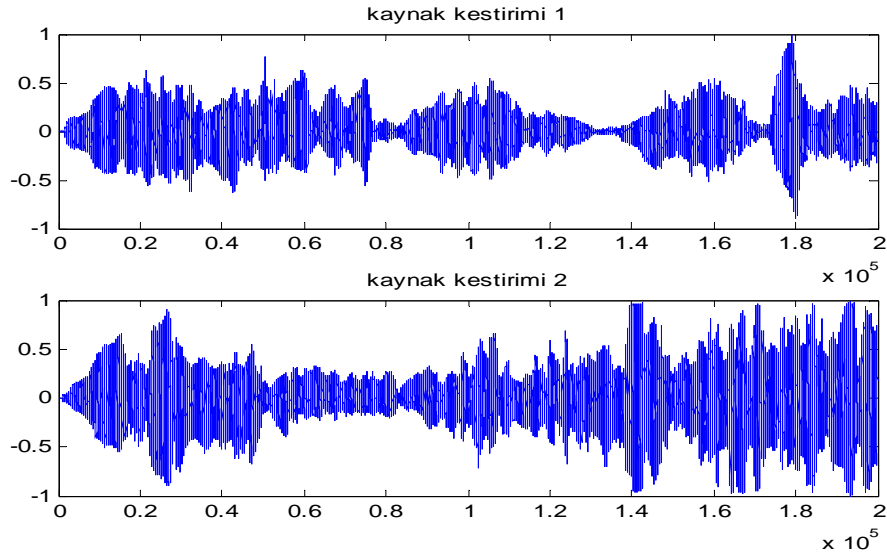


Şekil 4.6.: Kaynakların doğrusal üstdüşümlerinden oluşan algılayıcı sinyalleri

Bell ve Amari algoritmaları, bu karışık sinyaller içinde gerçekleşmiş ve her iki algoritmanın kaynak kestirimleri, Şekil 4.7. ve Şekil 4.8.'de verilmiştir.



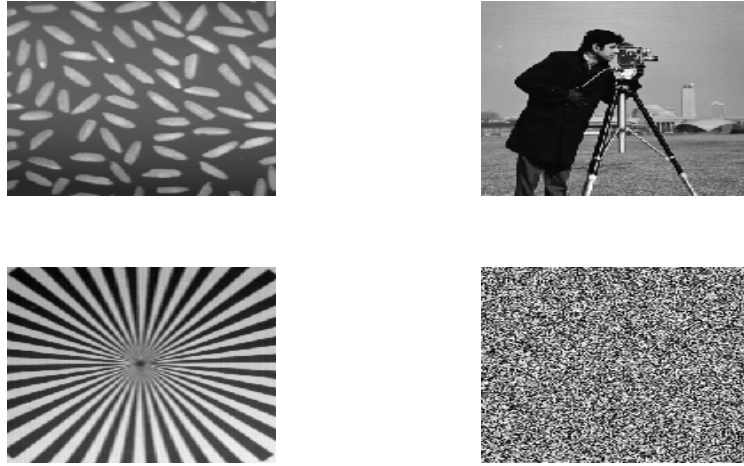
Şekil 4.7.: Bell algoritması kaynak kestirimleri



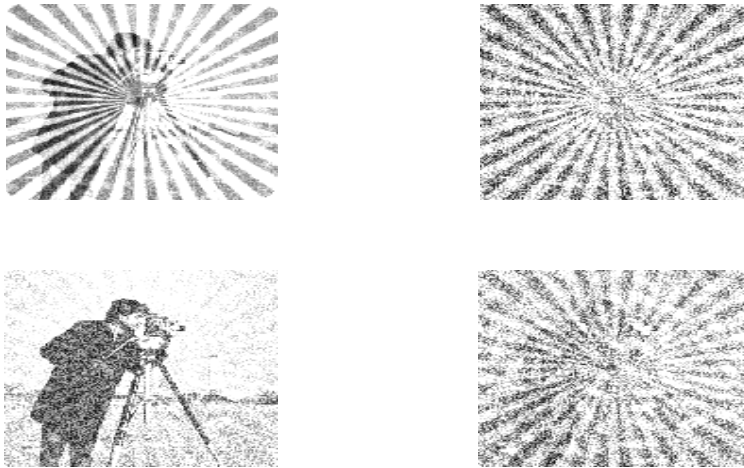
Şekil 4.8.: Amari algoritması kaynak kestirimleri

Aynı çalgıdan kaydedilen sinyallerin ayrıştırılmasında, Amari algoritmasının daha başarılı olduğu görülmüştür. Bell'in algoritması da kaynak sinyallerinin ayrıştırmasını sağlamıştır ancak Amari algoritmasının yakınsama zamanı Bell'in algoritmasına göre çok daha iyidir. Bell'in algoritması verinin tamamının 6 defa kullanılmasıyla ayrıştırma sağlarken, Amari algoritması verinin tamamının 1 defa kullanılmasıyla karışımları ayrıştırmıştır.

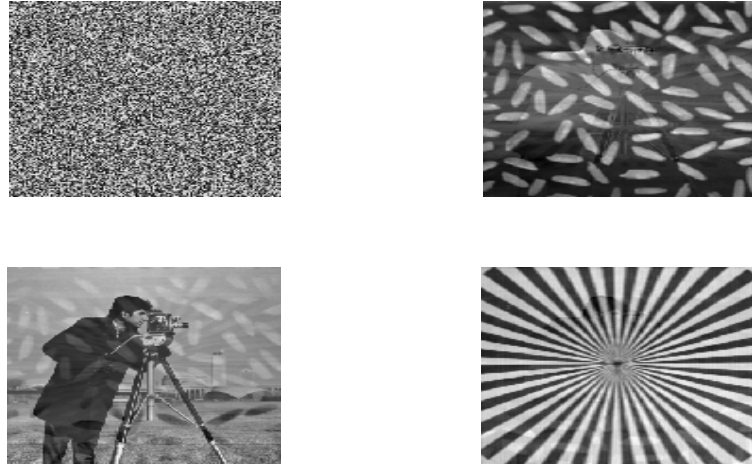
Sinyal İşleme alanında Kör Kaynak Ayrıştırma algoritmaları, görüntü işleme uygulamalarında da kullanılmaktadır[13,31,32]. Amari algoritması, birbirine karışmış pek çok görüntü sinyalini ayrıştırmak için de kullanılmakta ve doğrusal karışmış görüntü sinyallerini ayrıştırmada da başarılı sonuçlar vermektedir. Şekil 4.9.'da kaynak görüntü sinyalleri, Şekil 4.10.'da doğrusal karışmış kaynak sinyallerinden oluşan algılayıcı sinyalleri, Şekil 4.11.'de Amari algoritmasının kaynak kestirimleri verilmiştir.



Şekil 4.9.: Kaynak görüntü sinyalleri



Şekil 4.10.: Algılayıcı sinyalleri



Şekil 4.11.: Amari algoritması kaynak kestirimleri

4.2. Kör Ters Evrişim Algoritmasının Gerçeklenmesi

Kör Kaynak Ayırıştırma algoritmalarının gerçekleşmesiyle, doğrusal karışık kaynak sinyallerinin ayrıştırılması sağlandığı görüldükten sonra, evrişimlenmiş kaynak sinyallerinin ayrışımını için Kör Kaynak Ayırıştırma algoritmalarının frekans uzayına adaptasyonu yapılmış ve frekans uzayında ayrıştırılmış frekans bölmeleri, zaman uzayına çevrilerek zaman uzayında ters evrişimlenmiş kaynak sinyal kestirimleri elde edilmiştir. Bu çalışmada, evrişimlenmiş müzik-müzik karışımlarının ters evrişimlenmesi hedeflenmiştir. Bunun için konuşma-konuşma ve konuşma-müzik karışımlarının ters evrişimlenmesi yapılmıştır.

Frekans Uzayı Kör Ters Evrişim algoritmasının amacı, evrişimlenmiş karışımların frekans bölmelerinin ayrışımını sağlamaktır. Kör Kaynak Ayırıştırma algoritması öğrenme kuralı ile frekans uzayında her bir frekans bölmesi Karmaşık Yapay Sinir Ağı ağırlıkların güncellenmesi sağlanmıştır. Böylece, doğrusal karışık frekans bölmeleri ayrıştırılmıştır. Ayrıştırılan frekans bölmeleri, tekrar zaman uzayına dönüştürülerek orijinal kaynak sinyallerinin kestirimleri geri kazanılmaya çalışılmıştır.

Kör Ters Evrişim Algoritmasını gerçeklemek için, öncelikle zaman uzayında evrişimlenmiş karışımların frekans uzayına dönüştürülmesi gerekmektedir. Frekans uzayına dönüşüm için Kısa Zaman Fourier Dönüşümü uygulanmıştır. Her bir frekans bölmesi için kullanılan Karmaşık Yapay Sinir Ağı'nın ağırlık matrislerinin elemanları, Sonlu Dürtü Yanıtlı filtre ters evrişim matrislerinin frekans cevabı değerlerine karşılık geldiğinden, Kısa Zaman Fourier Dönüşümü'nün boyutu, frekans uzayında Sonlu Dürtü Yanıtlı filtrelerin derecesini verecektir. Başka bir deyişle, Kısa Zaman Fourier Dönüşümü boyutu, ters evrişim işleminin Sonlu Dürtü Yanıtlı filtrelerinin dereceleriyle eşit olacaktır. Uygulamada her bir frekans bölmesine bir Karmaşık Yapay Sinir Ağı kullanılacağından, Kısa Zaman Fourier Dönüşümü'nün boyutu aynı zamanda Karmaşık Yapay Sinir Ağı sayısına eşit olacaktır. Giriş sinyallerini frekans uzayına dönüştürmeden önce Hamming penceresinden geçirilmiştir. Hamming penceresi frekans spektrumunu düzleyerek daha önce bahsedilen "circularity" problemini çözmek için kullanılmıştır. Frekans uzayında her bir frekans bölmesi için atanan Karmaşık Yapay Sinir Ağı ağırlıklarının güncellenmesinde Amari'nin öğrenme kuralı uygulanmıştır.

$$\Delta W_f \propto [I - g(e_f)e_f^H]W_f \quad (4.3)$$

Frekans uzayı Kör Ters Evrişim algoritmasının gerçeklenmesinde, öğrenme kuralı olarak Bell algoritması kullanılmamıştır. Bell algoritmasında bulunan ters evriği işlemi bazı tekilliklere yol açmaktadır. Bu, frekans uzayında istenmeyen bir durumdur. Ayrıca Amari algoritmasının daha kararlı olmasından dolayı, bu çalışmada, ters evrişim filtre matrislerinin güncellenmesinde Amari algortması kullanılmıştır.

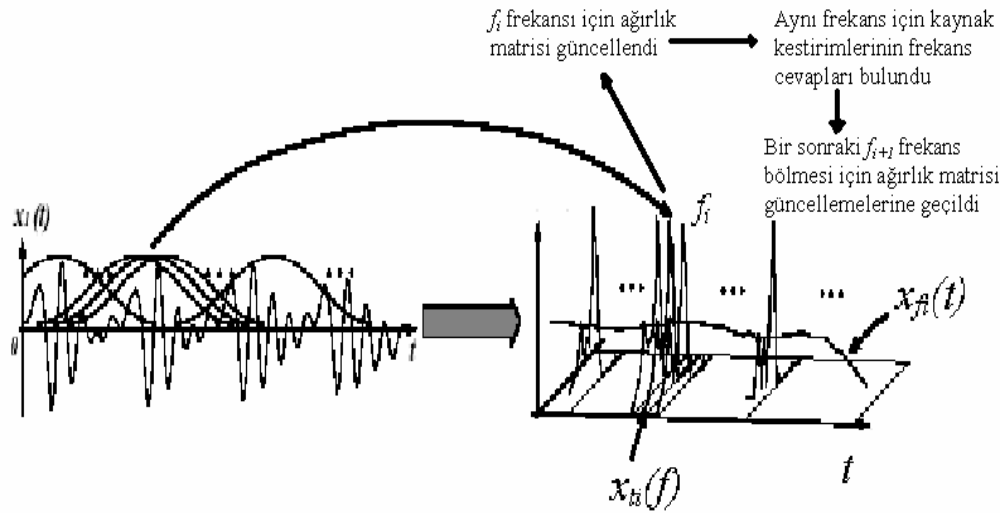
Önceki bölümlerde belirtildiği üzere, her bir frekans bölmesine uygulanan öğrenme kuralındaki aktivasyon fonksiyonu karmaşık aktivasyon fonksiyonudur. Bu fonksiyon:

$$g(e_f) = \tanh(\operatorname{Re}\{e_f\}) + i \cdot \tanh(\operatorname{Im}\{e_f\}) \quad (4.4)$$

Her bir frekans bölmesi için çıkış frekans bölmelerinin değerleri, ters evrişim filtre matrisi, frekans bölmesinin $e_f = W_f \cdot x_f$ matris çarpımıyla hesaplanmaktadır. f_i bölmesinin ters evrişim filtre matris ağırlıkları ise, f_i frekansı

için çıkış frekans bölmelerinin hesaplanmasının ardından, bu çıkış frekans bileşenleriyle Amari kuralı uygulanarak hesaplanmakta ve ardından yeni f_{i+1} frekans bölmesi için çıkış frekans bölmelerinin hesaplanmasına geçilmektedir.

Bu çalışmada, farklı bir güncellenme şeması kullanılmıştır. Bu şemaya göre, belli bir f_i frekans bölmesi için ters evrişim filtre matris ağırlıklarının güncellenmesinin ardından, bu yeni ağırlıklar, aynı f_i frekans bölmesinin çıkış frekans bileşenlerinin bulunmasında tekrar kullanılmıştır. Başka bir deyişle, f_i frekans bölmesinin içerdiği bilgi, ters evrişim filtre matrislerinin güncellenmesinde kullanılırken, aynı zamanda o frekans bölmesindeki kaynak kestirimlerine ulaşmak için de kullanılmış olur. Bu şema ile yakınsama zamanı oldukça kısalmaktadır. Kullanılan bu yaklaşım Şekil 4.12.'de şematik bir biçimde gösterilmektedir.

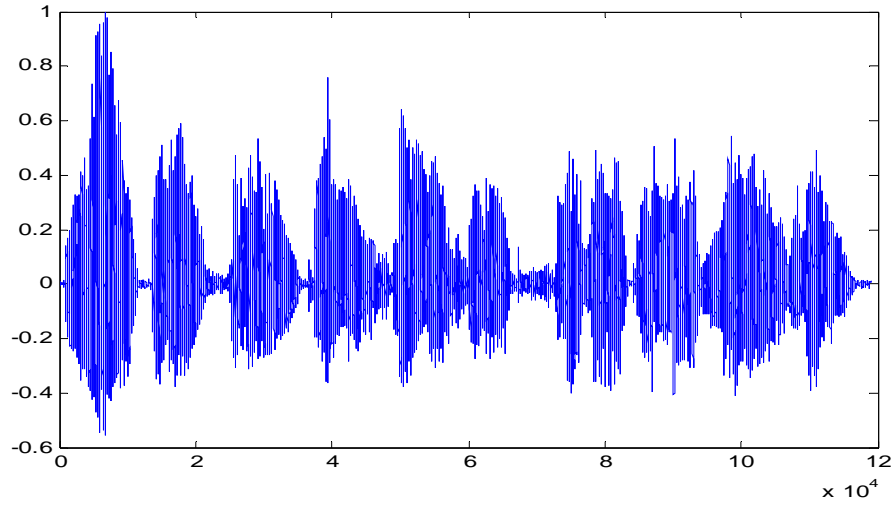


Şekil 4.12.: Bu çalışmada kullanılan güncelleme şeması temel gösterimi

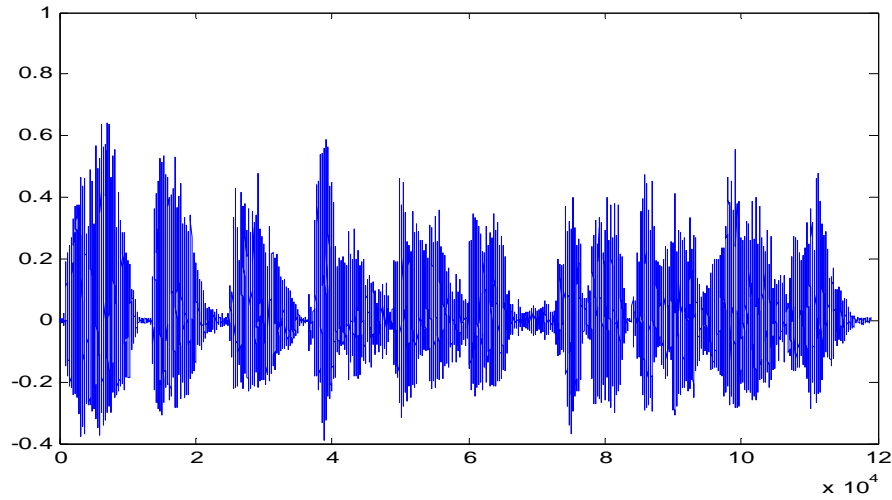
Bu çalışmada, gerçek dünya konuşma-konuşma, konuşma-müzik ve müzik-müzik karışımlarının ayrılması üzerinde çalışılmıştır. Öncelikle konuşma-konuşma ve konuşma-müzik karışımlarının ayrıştırılması hedeflenmiştir. Bu algoritmanın gerçekleştirilmesinde kullanılan ses dosyaları http://inc2.ucsd.edu/~tewon/ica_cnl.html adresinden alınan Te-Won Lee'nin kaydettiği ses dosyalarıdır. Lee, Bir odada farklı yerlere yerleştirilen iki mikrofon,

farklı iki kaynaktan gelen konuşma ve müzik sinyallerini 16kHz ile kaydetmiştir ve bu iki mikrofon sinyali, kaynak ayrışımında kullanılmıştır. [6]'da belirtildiği gibi Te-Won Lee müzik-konuşma ayrışımını 30 epokta tamamlarken, bu yaklaşımla konuşma sinyalinin temiz bir kestirimi yaklaşık 3 epokta sağlanmıştır. Bu da gerçek kaynakların ayrıştırmasında belirgin bir ilerleme kaydedildiğinin göstergesi olmaktadır.

Şekil 4.13. ve Şekil 4.14.'de her iki mikrofon ile kaydedilen ve Kör Ters Evrişim algoritmasında kullanılan konuşma-konuşma karışımları görülmektedir. Daha önce de belirtildiği gibi, bunlar, evrişimlenmiş gerçek dünya karışımlarıdır.

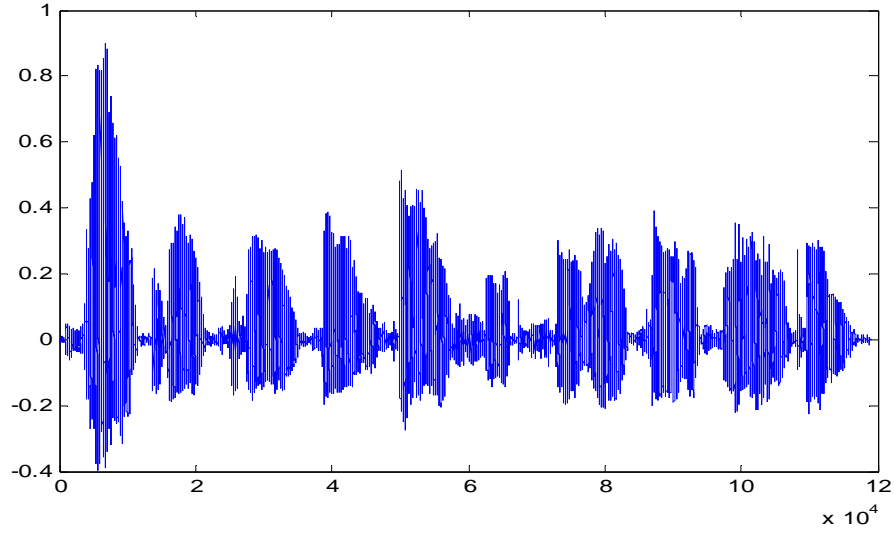


Şekil 4.13.: Konuşma- Konuşma karışımı mikrofon 1

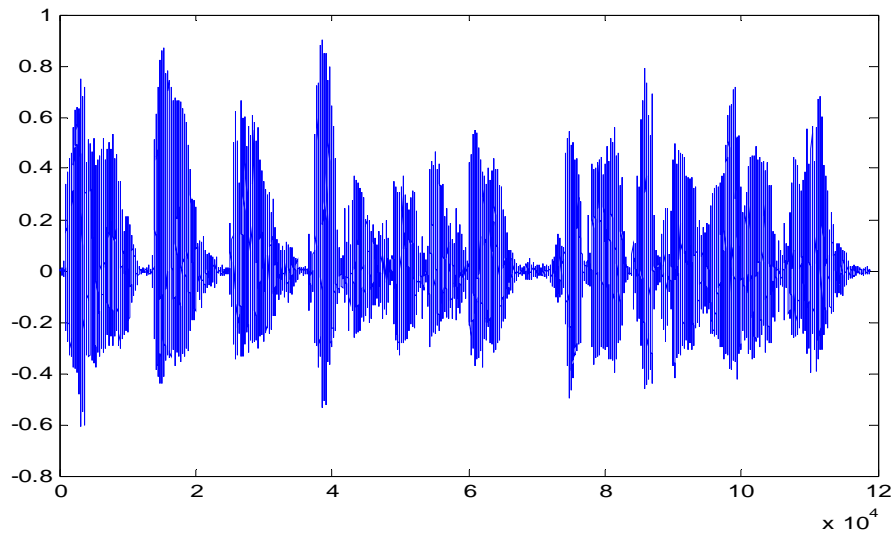


Şekil 4.14.: Konuşma- Konuşma karışımı mikrofon 2

Bu karışımların ayrıştırılmasında kullanılan ters evrişim filtre matrislerinin dereceleri 1024 olarak belirlenmiştir. Bu aynı zamanda, Kısa zaman Fourier Dönüşümü'nün boyutu ve frekans uzayında her bir frekans bölmesine karşılık gelen Karmaşık Yapay sinir Ağı'nın sayısıdır. Şekil 4.15. ve Şekil 4.16.'da da gerçekleştirilen Kör Ters Evrişim algoritması sonucunda kaynak kestirimleri görülmektedir.

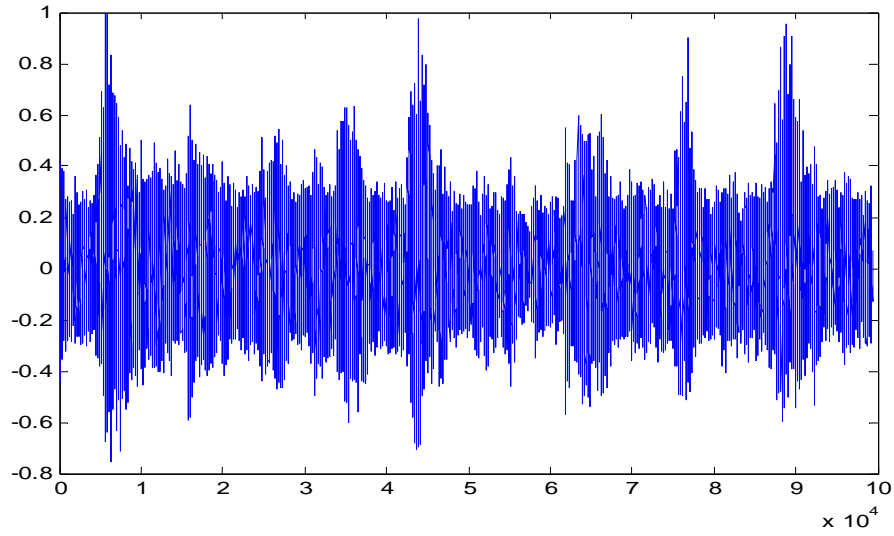


Şekil 4.15.: Konuşma sinyali 1 kestirimi

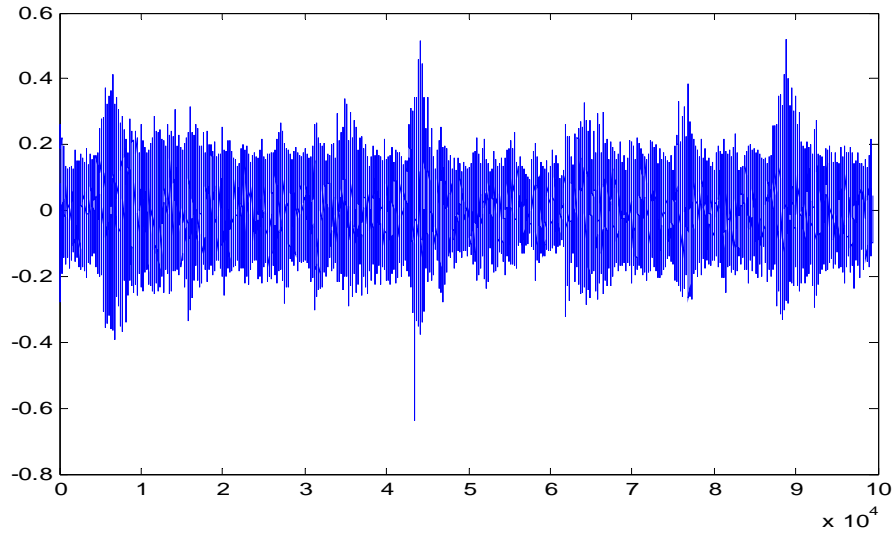


Şekil 4.16.: Konuşma sinyali 2 kestirimi

Şekil 4.17. ve Şekil 4.18.'de her iki mikrofon ile kaydedilen konuşma-müzik karışımları görülmektedir.



Şekil 4.17.: Konuşma- Müzik karışımı mikrofon 1

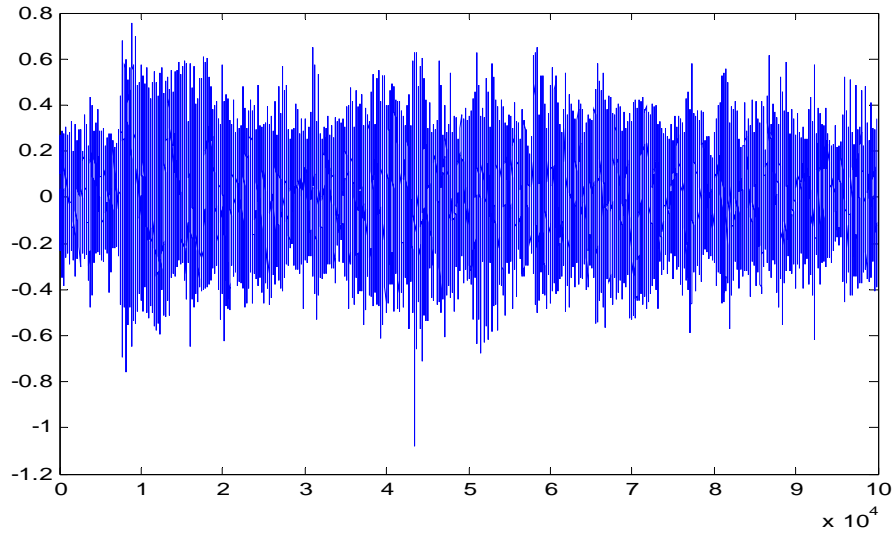


Şekil 4.18.: Konuşma- Müzik karışımı mikrofon 2

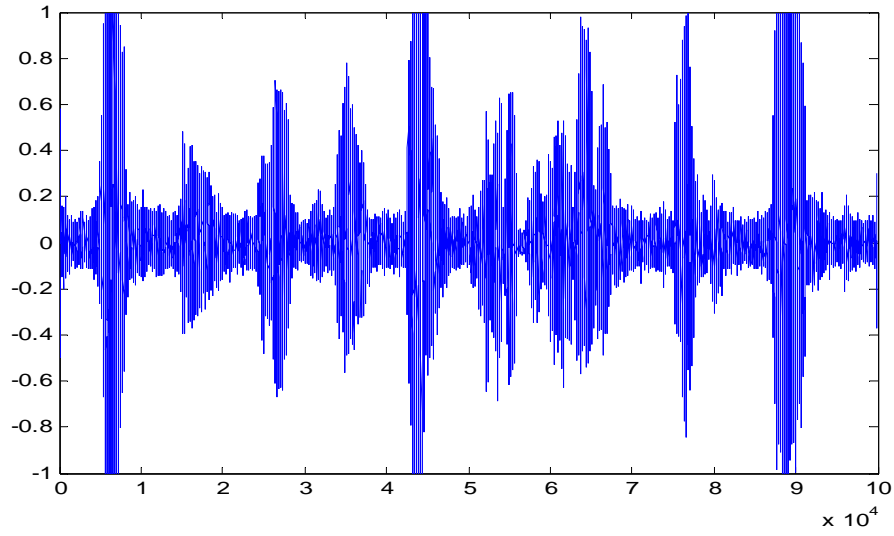
Kör Ters Evrişim algoritmasının gerçekleştirilmesinde de, permütasyon ve ölçekleme belirsizlikleri ile karşılaşmıştır. Her bir frekans bölmesinin için farklı bir Karmaşık Yapay Sinir Ağı kullanıldığından, her frekans bölmesi için kaynak kestirimleri farklı katsayılarla ölçeklenecek ve farklı bir matrisle permüte edilecektir. Permütasyon belirsizliğini aşmak için, ayrıştırma filtreleri çevrim-dışı

hesaplanmış ve bir filtreyi bir frekans bölgesine uyguladıktan sonra diğer frekans bölgesi için hesaplamalara geçerken, bir önceki bölme için güncellenen ağırlıklar bu yeni bölmenin başlangıç değerleri yapılmıştır. Ölçekleme problemi de, ters evrişim filtre matrislerinin determinantının her zaman “1”e eşit olmasını sağlayarak çözülmüştür.

Frekans uzayında gerçekleşen algoritmayla, konuşma-müzik karışımlarının da ters evrişimleri sağlanmış ve kaynak kestirimleri hesaplanmıştır. Ters evrişim filtre matrisi derecesi 1024 olarak belirlenmiştir. Konuşma sinyalinin temiz bir kestirime 3 epokta ulaşılırken, müzik sinyalinin kestirime yakınsama için daha fazla zaman gerekmektedir. Bunun sebebi; müzik sinyali, Gaussian sinyale konuşma sinyalinden daha yakındır. Başka bir deyişle, müzik sinyalinin kurtosisi konuşma sinyalinin kurtosisinden daha küçüktür. Bu kestirimler, Şekil 4.19. ve Şekil 4.20.’de görülmektedir.

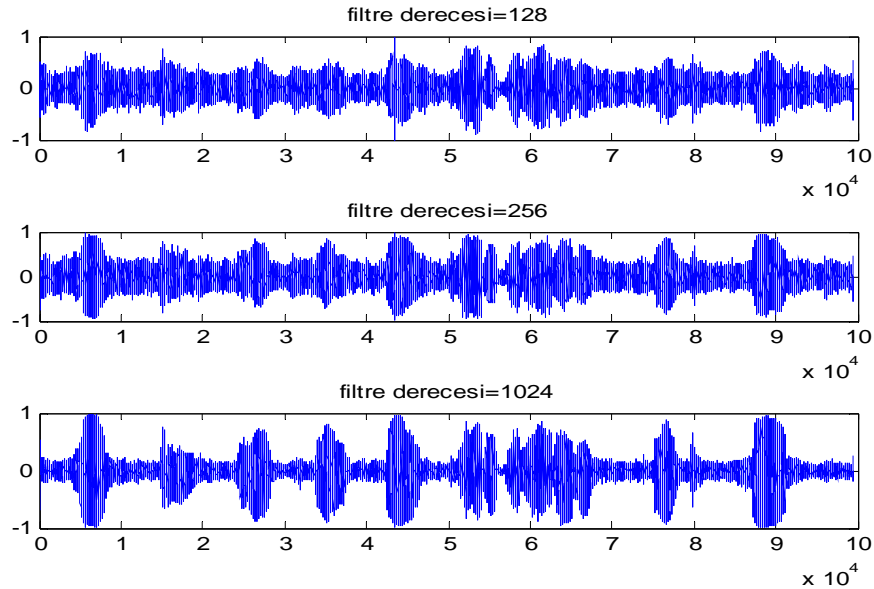


Şekil 4.19.: Müzik sinyali kestirimi



Şekil 4.20.: Konuşma sinyali kestirimi

Farklı ters evrişim filtre matris dereceleri belirlenerek algoritma denenmiştir. Buna göre yüksek filtre matris dereceleri daha verimli sonuçlar vermiştir. Şekil 4.21.'da, filtre matris dereceleri 128, 256 ve 1024 olarak belirlendikten sonra aynı sayıdaki epokta kaynak kestirimleri görülmektedir. 1024 matris dereceli algoritma konuşma sinyalinin daha doğru kestirimine ulaşmıştır.



Şekil 4.21.: Farklı ters evrişim filtre dereceleriyle konuşma sinyali kestirimleri

Algılayıcı sinyallerinin sayısı, en az kaynak sinyalleri kadar olması gerekliliğinden daha önce bahsedilmiştir. Evrişimlenmiş karışımların kayıtları sırasında mikrofonlar, en az kaynak sinyallerinin sayısı kadar ve bir dizi oluşturacak şekilde kullanılmalıdır. Mikrofonlar, tek yönlü sesleri kaydeden mikrofonlar olmamalıdır. Tek yönü kaydeden mikrofon kullanıldığında, her mikrofon bir kaynak sinyalinin sesini kaydedecektir, diğer kaynak sinyalleri o mikrofonda duyulmayacaktır. Bu yüzden, çok yönlü mikrofonlar kullanılmalıdır. Çok yönlü mikrofonlar, hem yakın olduğu kaynak sinyalinin sesini hem de diğer kaynakların kendi seslerini ve duvardan yansımalarını kaydedecektir.

Evrişimlenmiş enstrümantal müzik-müzik karışımlarının ayrıştırılması için öncelikle flüt-klarnet karışımları üzerinde çalışılmıştır. Bu iki çalgının tam olarak ayrıştırılması oldukça zordur. Bu iki çalgının frekans karakteristiklerinin temel olarak birbirine yakın olmasından kaynaklandığı düşünülmektedir. Ayrıca her iki çalgının da temel frekansları baskın değildir[30]. Müzik-müzik karışımlarında hem ton, hem gürlük her ikisi de frekans görüncesini değiştirmektedir. Aynı notalar aynı oktavda çalınmasına rağmen spektrumları farklı olabilmektedir. Bununla birlikte, gürlükte spektrumu değiştirmektedir. Öyle ki; gür çalınan bir notanın doğuşkanları, kısık çalınan aynı notanın doğuşkanlarından daha yüksektir[30]. Tüm bunlar çalgıların ayrıştırılmasını zorlaştırmaktadır. Müzik-müzik karışımlarının tam olarak ayrıştırılması sağlanamamıştır.

5. SONUÇLAR

Bu çalışmada, doğrusal ve evrişimlenmiş olarak karışmış sinyallerin ayrıştırılması sağlanmıştır. Her iki Kör Kaynak Ayrıştırma algoritması da (Bell algoritması ve Amari algoritması) doğrusal karışık kaynak sinyallerinin ayrıştırılmasında kullanılmıştır. Böylece, her iki algoritmanın performansları denenmiştir. Amari algoritması, Bell'in algoritmasına göre yakınsama zamanı açısından daha verimli olmuştur. Bell'in algoritmasında matris ters evriği işlemi bulunmaktadır. Amari algoritmasında maliyet fonksiyonunun Riemann uzayında olması, bu algorithmada matris ters evriğinin alınması işlemini yok etmiştir. Böylece algoritma daha kararlı hale gelmiştir ve yakınsama zamanı açısından daha hızlı olmuştur. Doğrusal karışık sinyallerin ayrıştırılmasında, hem farklı çalgılardan kaydedilen, hem de aynı çalgılardan kaydedilen sinyaller, kaynak sinyalleri olarak kullanılmış ve her iki algorithmada bu kaynak sinyallerinin ayrıştırılmasında denenmiştir.

Bu çalışmada, Bell ve Amari Kör Kaynak Ayrıştırma algoritmaları MATLAB ortamında gerçekleştirilmiştir. EK-1'de Bell algoritması, EK-2'de Amari algoritması verilmiştir.

Kör Ters Evrişim uygulaması frekans uzayında gerçekleştirilmiştir. Zaman uzayında evrişimlenmiş olarak karışmış kaynak sinyallerinin, frekans uzayındaki frekans bölmelerinin doğrusal karışmış olduğunu göz önüne alınarak, Smagradis tarafından [2,3,4] gerçekleştirilen Amari algoritmasının frekans uzayı adaptasyonu ile evrişimlenmiş karışımların ayrıştırılması sağlanmıştır. Kör Kaynak Ayrıştırma uygulamalarında Amari algoritması daha verimli olduğundan, Kör Ters Evrişim yaklaşımının gerçekleştirilmesinde de Amari algoritması kullanılmıştır. Herhangi bir ofiste kaydedilen konuşma-konuşma ve konuşma-müzik gerçek-dünya karışımlarının ters evrişimlenmesi sağlanmıştır. Amari'nin algoritmasına ek olarak her frekans bölmesine ait filtre matrisi ağırlık değerleri güncellendikten sonra o güncellenmiş ağırlıklara göre, aynı frekans bölmesine ait kestirimler yeniden hesaplanmaktadır. Böylece yakınsama zamanı azaltılmıştır.

Kör Kaynak Ayrıştırma ve Kör Ters Evrişim algoritmalarının performanslarını arttırmak için ölçekleme ve permütasyon belirsizliklerine daha

verimli çözümler getirilmesi gerekir. Özellikle frekans uzayında permütasyon belirsizliği oldukça zor durumlara yol açabilmektedir. Ayrıca yapay sinir ağlarının parametreleri dikkatlice seçilmelidir. Yanlış parametrelerin kullanımı yerel bir minimum bulunarak, hatalı sonuçların elde edilmesine sebep olmaktadır.

Frekans uzayı kör ters evrişim algoritması da MATLAB ortamında gerçekleştirilmiştir ve EK-3’de verilmiştir. Ters evrişim matrislerinin dereceleri 128, 256, 512 ve 1024 olarak farklı seçeneklerle denenmiştir. En iyi sonuçlar 512 ve 1024 ters evrişim matris derecesiyle alınmıştır.

Bu çalışmada, gerçek-dünya karışımları için verimli bir ters evrişim sağlandıysada, gelecek çalışmalarda permütasyon ve “circularity” problemlerine daha iyi çözümler getirmek gerekecektir.

Frekans uzayı kör ters evrişim algoritmasında, her frekans bölmesi için aynı permütasyonun sağlanması oldukça zordur. Permütasyon probleminin üstesinden gelmek için, her frekans bölmesi için elde edilen ters evrişim filtre matrisi, bir sonraki frekans bölmesinin başlangıç değerleri olarak atanır. Ancak bu yaklaşım filtre parametreleri arasında bağımlılık oluşturacağından oldukça güçlü yerel minimumlara sebep olabilir ve sistemin çözümü sağlanamayabilir. Permütasyon probleminin daha kararlı bir çözüme ihtiyacı vardır. Günümüzde “circularity” probleminin çözümü için de çok değişik yaklaşımlar geliştirilmektedir. Bu değişik yaklaşımlarla frekans uzayı ters evrişim algoritmasının verimliliği artırılabilir.

KAYNAKLAR

- [1] HYVARIEN A., *Survey on Independent Component Analysis*, Neural Computing Surveys, 94-128 (1999)
- [2] SMAGRADIS P.J., *Information Theoretic Approach to Source Separation*, Master Thesis, MAS Department, Massachusetts Institute of Technology, (1997)
- [3] SMAGRADIS P.J., *Efficient Blind Separation of Convolved Sound Mixtures*, IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz NY, (1997)
- [4] SMAGRADIS P.J., *Blind Separation of Convolved Mixtures in the Frequency Domain*, International Workshop on Independence & Artificial Neural Networks University of La Laguna, Tenerife, Spain, (1997)
- [5] STONE JV., *Blind Deconvolution Using Temporal Predictability*, Neurocomputing, **49**, 79-86, (2002)
- [6] LEE T-W, BELL A.J. ve ORGLMEISTER R, *Blind Separation of Real-World Signals*, Proceedings of International Conference on Neural Networks, Houston, ABD, (1997)
- [7] LAMBERT R.H., *Multichannel Blind Deconvolution: Fir Matrix Algebra And Separation Of Multipath Mixtures*, Ph.D. Thesis, (1996)
- [8] WESTNER A. ve BOVE V.M. JR., *Applying Blind Source Separation and Deconvolution to Real-World Acoustic Environments*, Proc. 106th Audio Engineering Society(AES) Convention, (1999)
- [9] CHOI S., *Acoustic source separation: Fundamental issues*, Proc. Int. Conf. Speech Processing, 505-510, (1999)
- [10] AMARI S., CICHOCKI A. ve YANG H.H., *A New Learning Algorithm for Blind Signal Separation*, Advances in Neural Information Processing Systems, (1996)
- [11] PHAM D.-T., SEVIERE C. ve BOUMARAF H., *Blind Separation of Audio Mixtures Using Nonstationary*, International Symposium on Independent Component Analysis and Blind Signal Separation, (2003)

- [12] PLATT J.C. ve FAGGIN F., *Networks for the Separation of Sources that Are Superimposed and Delayed*, Advances in Neural Information Processing Systems, 730-737, (1991)
- [13] CICHOCKI A., KARHUNEN J., KASPRZAK W. ve VIGARIO R., *Neural Networks for Blind Separation With Unknown Number Of Sources*, Neurocomputing, **24**, 55-93, (1999)
- [14] SAWADA H., MUKAI R., RYHOVE K.S., ARAKI S. ve MAKINO S., *Spectral Smoothing for Frequency Domain Based Blind Source Separation*, International Workshop on Acoustic Echo & Noise Control, (2003)
- [15] CHOI S., LYU Y., BERTHOMMIER F., GLOTIN H. ve CICHOCKI A., *Blind separation of delayed and superimposed acoustic sources: Learning algorithm and experimental study*, Proc. Int. Conf. Speech Processing 109-114, (1999)
- [16] CHOI S. ve CICHOCKI A., *Adaptive Blind Separation of Speech Signals: Cocktail Party Problem*, International Conference on Signal Processing, 617-620, (1997)
- [17] LAMBERT R.H. ve NIKIAS C.L., *Bussgang Methods for Separation of Multipath Mixtures*, Neural Information Processing Systems Workshop on Blind Signal Processing Applications, (1997)
- [18] LEE T-W., BELL A. ve LAMBERT R.H., *Blind Separation of Delayed and Convolved Sources*, Advances in Neural Information Processing Systems, (1997)
- [19] HYVARIEN A., KARHUNEN J. ve OJA E., *Independent Component Analysis*, Wiley –Interscience, New York, ABD, (2001)
- [20] CICHOCKI A. ve AMARI S., *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, West Sussex, İngiltere, (2003)
- [21] CAO X.R. ve LIU, *A General Approach to Blind Source separation*, IEEE Transactions on Signal Processing, **44**, 562-571, (1996)
- [22] TONG L., LIU R. VE SOON V.C., *Indeterminacy and Identifiability of Blind Identification*, IEEE Transactions on Circuit Systems, **38**, 499-509, (1991)

- [23] CARDOSO J.F., *Source Separation Using Higher Order Moments*, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2109-2112, (1989)
- [24] GIROLAMI M. ve FYFE C., *Negentropy and Kurtosis as Projection Pursuit Indices Provide Generalised ICA Algorithms*, Neural Information Processing Systems Semineri, Denver, ABD, (1996)
- [25] PIERANI A, PIAZZA F, SOLLAZZI M ve UCINI A, *Low Complexity Adaptive Non-Linear Function For Blind Signal Separation*, International Joint Conference on Neural Networks, 333-338, (2000)
- [26] SAWADA H, MUKAI R, ARAKI S ve MOKINA S, *A Polar-Coordinate Based Activation Function for Frequency Domain Blind Source Separation*, Independent Component Analysis, Wiley –Interscience, New York, ABD, (2001)
- [27] COMON P., *Independent Component Analysis- a new concept?*, Signal Processing, **36**, 284-314, (1989)
- [28] HERAULT J. ve JUTTEN C., *Blind Separation of Sources, part I, An Adaptive Algorithm Based on Neuromimetic Architecture*, Signal Processing, **24**, 1-10, (1991)
- [29] BELL A.J. ve SEJNOWSKY.J., *An Information Maximization Approach to Blind Separation and Deconvolution*, Neural Computation, **7**, 1129-1159,(1995)
- [30] KLINGSEISEN J. ve PLUMBLEY M. D., *Towards Musical Instrument Separation Using Multiple-Cause Neural Networks*, Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation, 447-452, (2000)
- [31] KUNDUR D. ve HATZIKANOS D., *Blind Image Deconvolution*, IEEE Signal Processing Magazine, **13**, 43-64, (1996)
- [32] CHEM J.L. ve CHUNAG Y.T., *Blind images separation and restoration*, OSA (Optical Society of American) annual meeting, 2002

EK – 1 Kör Kaynak Ayırıştırma Bell Algoritması

```

%Linear Blind Source Separation Bell's Algorithm
clear;
close all;

%two zero mean source signals
signal_1=wavread('C:\MATLAB6p5\work\eff_data\Viola11');
signal_2=wavread('C:\MATLAB6p5\work\eff_data\Viola21');
signal_1=signal_1';
signal_2=signal_2';
signal_1=signal_1(:,1:200000);
signal_2=signal_2(:,1:200000);
signal_1=signal_1-mean(signal_1);
signal_2=signal_2-mean(signal_2);
signal=[signal_1;signal_2];

entry = input('folder adı\n','s');
mkdir('C:\Documents and Settings\Pelin\My Documents\tez',entry);%creates a
directory

%for gaussian distribution gaussian=0
kurtosis_signal1=kurtosis(signal_1)%kurtosis of source1
kurtosis_signal2=kurtosis(signal_2)%kurtosis of source2
%both signals are super-gaussian because they are music signals

A=rand(2,2)%mixing Matrix

x=A*signal;
x(1,:)=x(1,:)/max(abs(x(1,:)));

```

```

x(2,:)=x(2,+)/max(abs(x(2,:)));
kurtosis_data1=kurtosis(x(1,:))%kurtosis of observed signal 1
kurtosis_data2=kurtosis(x(2,:))%kurtosis of observed signal 2

sound(x(1,:),22050);
pause(10)
d1=strcat('C:\Documents and Settings\Pelin\My Documents\tez\',entry,'\Data_1');
wavwrite(x(1,:),22050,d1);
sound(x(2,:),22050);
pause(10)
d2=strcat('C:\Documents and Settings\Pelin\My Documents\tez\',entry,'\Data_2');
wavwrite(x(2,:),22050,d2);

W=rand(2,2)%initial values of unmixing matrix
y=W*x;
I=[1 0;0 1];

for k=1:600000

    m=mod(k,200000);

    if m==0
        m=200000;
    end

    y(:,m)=W*x(:,m);
    W=W+0.001*(inv(W')-2*tanh(y(:,m))*x(:,m)');
    if(mod(k,50000)==0)
        s=num2str(k);
        e1=strcat('C:\Documents          and          Settings\Pelin\My
Documents\tez\',entry,'\estimation1_',s);

```

```

e2=strcat('C:\Documents          and          Settings\Pelin\My
Documents\tez\entry,'estimation2_',s);
y=W*x;
y(1,:)=y(1,+)/max(abs(y(1,:)));
y(2,:)=y(2,+)/max(abs(y(2,:)));
figure
subplot(2,2,1),plot(signal_1);
title('source signal 1');
subplot(2,2,2),plot(signal_2);
title('source signal 2');
subplot(2,2,3),plot(y(1,:));
title('estimation of source 1');
subplot(2,2,4),plot(y(2,:));
title('estimation of source 2');
sound(y(1,:),22050);
pause(10)
wavwrite(y(1,:),22050,e1);
sound(y(2,:),22050);
pause(10)
wavwrite(y(2,:),22050,e2);
k
W
end
end

```

EK – 2 Kör Kaynak Ayırıştırma Amari Algoritması

```

%Linear Blind Source Separation Amari's Algorithm
clear;
close all;

signal_1=wavread('Viola11');

```

```

signal_2=wavread('Viola21');
signal_1=signal_1';
signal_2=signal_2';
signal_1=signal_1(:,1:200000);
signal_2=signal_2(:,1:200000);
signal_1=signal_1-mean(signal_1);
signal_2=signal_2-mean(signal_2);
signal=[signal_1;signal_2];

entry = input('folder adý\n','s');
mkdir('C:\Documents and Settings\Pelin\My Documents\tez',entry);%creates a
directory

%for gaussian distribution gaussian=0
kurtosis_signal1=kurtosis(signal_1)%kurtosis of cello
kurtosis_signal2=kurtosis(signal_2)%kurtosis of sarasati
%both signals are super-gaussian because they are music signals

A=round(rand(2,2)*10)+1%mixing Matrix

x=A*signal;
x(1,:)=x(1,+)/max(abs(x(1,:)));
x(2,:)=x(2,+)/max(abs(x(2,:)));
kurtosis_data1=kurtosis(x(1,:))%kurtosis of observed signal 1
kurtosis_data2=kurtosis(x(2,:))%kurtosis of observed signal 2

sound(x(1,:),22050);
pause(10)
d1=strcat('C:\Documents and Settings\Pelin\My Documents\tez\',entry,'\Data_1');
wavwrite(x(1,:),22050,d1);
sound(x(2,:),22050);
pause(10)

```



```
d2=strcat('C:\Documents and Settings\Pelin\My Documents\tez\',entry,'\Data_2');
wavwrite(x(2,:),22050,d2);
```

```
W=rand(2,2)%initial values of unmixing matrix
```

```
y=W*x;
```

```
I=[1 0;0 1];
```

```
for k=1:200000
```

```
    m=mod(k,200000);
```

```
    if m==0
```

```
        m=200000;
```

```
    end
```

```
    y(:,m)=W*x(:,m);
```

```
    W=W+0.001*(I-tanh(0.01*y(:,m))*y(:,m)')*W;
```

```
    if(mod(k,50000)==0)
```

```
        s=num2str(k);
```

```
        e1=strcat('C:\Documents and Settings\Pelin\My
Documents\tez\',entry,'\estimation1_',s);
```

```
        e2=strcat('C:\Documents and Settings\Pelin\My
Documents\tez\',entry,'\estimation2_',s);
```

```
        y=W*x;
```

```
        y(1,:)=y(1,+)/max(abs(y(1,:)));
```

```
        y(2,:)=y(2,+)/max(abs(y(2,:)));
```

```
        figure
```

```
        subplot(2,2,1),plot(signal_1);
```

```
        title('source signal 1');
```

```
        subplot(2,2,2),plot(signal_2);
```

```
        title('source signal 2');
```

```
        subplot(2,2,3),plot(y(1,:));
```

```

title('estimation of source 1');
subplot(2,2,4),plot(y(2,:));
title('estimation of source 2');
sound(y(1,:),22050);
pause(10)
wavwrite(y(1,:),22050,e1);
sound(y(2,:),22050);
pause(10)
wavwrite(y(2,:),22050,e2);
k
W
end
end

```

EK – 3 Frekans Uzayı Kör Ters Evrişim Algoritması

```

%implementation of frequency domain multichannel blind deconvolution
clear all
close all

```

```

signal_1=wavread('C:\MATLAB6p5\work\eff_data\mix1');
signal_2=wavread('C:\MATLAB6p5\work\eff_data\mix2');

```

```

signal_1=signal_1';
signal_2=signal_2';

```

```

x1=signal_1;
x2=signal_2;
x=[x1;x2];

```

```

W=rand(2,2,M)+j*rand(2,2,M);%initial values of unmixing filter matrix

```

```

[R,C]=size(x);
M=1024;% size of stft=unmixing filter size=number of cnn
par=M/2;

i=1;
Window_cut=(hamming(M))';% size of hamming window is equal to size of stft
I=eye(2,2);

while (par*(i-1))+M<=C
    p=1;

    for k=(par*(i-1))+1:(par*(i-1))+M

        x_st(:,p)=x(:,k);
        p=p+1;
    end

    x_fft_1=fft((Window_cut.*x_st(1,:)),M);
    x_fft_2=fft((Window_cut.*x_st(2,:)),M);
    x_fft=[x_fft_1;x_fft_2];

    if i==1

        fft_=x_fft;
    else

        fft_=[fft_ x_fft];
    end

    i=i+1;
end%end of stft

```

```

[R1,C1]=size(fft_);
    al=C1/M;

    for ss=1:3

        for tt=1:M
            for pp=1:al

                fft_y(:,((pp-1)*M+tt))=W(:,:,tt)*fft_(:,((pp-1)*M+tt));

                W(:,:,tt)=W(:,:,tt)+0.001*(I-(tanh(real(fft_y(:,(pp-
1)*M+tt))))+j*tanh(imag(fft_y(:,(pp-1)*M+tt))))*fft_y(:,((pp-1)*M+tt))'*W(:,:,tt);

                fft_y(:,((pp-1)*M+tt))=W(:,:,tt)*fft_(:,((pp-1)*M+tt));%every
frequency bin is updated by the updated weigt matrix of itself
                W(:,:,tt+1)=W(:,:,tt);%to overcome pernutation problem
weights of frequency bin fi will be the initial values of fi+1

            end

        end

    end%end of frequency domain seperation

for ee=1:M

    W(:,:,ee)=W(:,:,ee)/sqrt(abs(det(W(:,:,ee))));
end%scaling of filter matrix weights that determinat of every matris is equal to 1

for t=1:i-1

    p=1;
    for k=(1+M*(t-1)):M*t

```

```

        fft_1(:,p)=fft_y(:,k);
        p=p+1;
    end
    for oo=1:M

        fft_out(:,oo)=W(:,:,oo)*fft_1(:,oo);
    end

    signal_get1=(ifft(fft_out(1,:),M))./Window_cut;
    signal_get2=(ifft(fft_out(2,:),M))./Window_cut;
    signal_get=[signal_get1;signal_get2];

    if t==1

        signal=signal_get(:,1:((M/2)+(M/4)));

    else

        signal=[signal signal_get(:,(M/4)+1:((M/2)+(M/4)))];
    end

    if t==i-1

        signal=[signal signal_get(:,(M/4)+1:M)];
    end
end%end of inverse stft

wavwrite(real(signal(1,:)),16000,'C:\matlab6p5\work\blind\son_fft\son1');
wavwrite(real(signal(2,:)),16000,'C:\matlab6p5\work\blind\son_fft\son2');

```