Gedikli, E., Tuğcu, E. (2024). Meta-Sezgisel Yöntemlere Dayalı Kör Kaynak Sinyal Ayırma. *Karadeniz Fen Bilimleri Dergisi*, 14(3), 1456-1470.

Karadeniz Fen Bilimleri Dergisi, 14(3), 1456-1470, 2024. DOI: 10.31466/kfbd.1474735

KFBD

Karadeniz Fen Bilimleri Dergisi
The Black Sea Journal of Sciences
ISSN (Online): 2564-7377 https://dergipark.org.tr/tr/pub/kfbd

Araştırma Makalesi / Research Article

Meta-Sezgisel Yöntemlere Dayalı Kör Kaynak Sinyal Ayırma

Eyüp GEDİKLİ[1*], Emin TUĞCU[2]

Öz

Kör kaynak ayırma problemi, en az iki karışmış sinyalin bilinmeyen kaynak sinyallerini belirleme işlemidir. Kaynak sinyaller, tıbbi alanda doğru teşhisin yapılmasında, kablosuz haberleşmede, radar, görüntü, ses verilerinin analizi için önemlidir. Kör kaynak ayırma probleminde yaygın olarak bağımsız bileşen analizi kullanılır. Bağımsız bileşen analizinde, ileri istatistiksel ve cebirsel yöntemler kullanılarak entropi ve korelasyon uyumluluğuna bakılır. Sinyalleri ayırmak için en yaygın kullanılan bağımsız bileşen analizi (Independent Component Analysis, ICA) algoritmalarından FastICA, Gauss dağılımı olmama ve negentropinin maksimum uygunluk kriterlerini iterasyon tabanlı olarak araştırır. Bu çalışmada, benzer şekilde iterasyon tabanlı yöntemler olan meta-sezgisel algoritmalar (MSA), uygunluk fonksiyonunu optimize etmek için kullanılmıştır. Uygunluk fonksiyonu, karışık sinyal ayırma matris üretimi ve yakınsamayı kontrol etmek için kullanılır. Bu çalışmada, vektörleri ortogonalleştiren Gram Schmidt sürecine dayalı ayırma matris üretimi önerilmiştir. Deneyler, FastICA ile meta-sezgisel (MS) algoritmalardan ateş böceği algoritması ve parçacık sürü optimizasyonu algoritmasıyla yapılmıştır. Üç kaynaktan üretilen sinyallerin karıştırılıp gürültü eklenmesi ile karışık sinyaller oluşturulmuştur. Sinyallerin farklı frekanslarda üretilerek gerçekleştirilen deneylerde, önerilen yöntem ile geleneksel FastICA algoritmasından daha başarılı korelasyon katsayısı ve kök ortalama kare hata sonuçları elde edilmiştir.

Anahtar Kelimeler: Ateş böceği algoritması, Bağımsız bileşen analizi, FastICA, , Gram Schmidt süreci, Kör kaynak ayırma, Parçacık sürü optimizasyonu.

Blind Source Signal Separation Based on Meta-heuristics Methods

Abstract

The blind source separation problem is the process of identifying unknown source signals from at least two mixed signals. Source signals are important for accurate diagnosis in the medical field, wireless communication, and analysis of radar, image and sound data. Independent component analysis (ICA) is often used for the problem of blind source separation. In independent component analysis, entropy and correlation compatibility are checked using advanced statistical and algebraic methods. FastICA, one of the most widely used independent component analysis algorithms for signal separation, iteration-based searches for non-Gaussianity and negentropy maximum fitness criteria. In this study, meta-heuristic algorithms (MHA), which are also iteration-based methods, were used to optimize the fitness function. The fitness function is used to generate the separation matrix for mixed signal to control convergence. In this study, the generation of the separation matrix is proposed based on the Gram-Schmidt process, which orthogonalizes the vectors. Experiments were performed using FastICA and meta-heuristic algorithms such as the firefly algorithm and the particle swarm optimization algorithm. Mixed signals are generated by mixing the signals from three sources and adding noise. In the experiments carried out by generating signals with different frequencies, more successful correlation coefficients and root mean square error results were obtained with the proposed method than the traditional FastICA algorithm.

Keywords: Firefly algorithm, Independent component analysis, FastICA, Gram Schmidt Process, Blind source separation, Particle swarm optimization.

[1]Karadeniz Teknik Üniversitesi, Yazılım Mühendisliği Bölümü, Trabzon, Türkiye, gediklie@ktu.edu.tr
[2]Karadeniz Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği Bölümü, Trabzon, Türkiye, emintugcu@ktu.edu.tr

[*]Sorumlu Yazar/Corresponding Author
Geliş/Received: 28.04.2024 **Kabul/Accepted:** 12.09.2024 **Yayın/Published:** 15.09.2024

1. Giriş

Kör kaynak ayrıştırma (Blind Source Separation, BSS) problemi, kaynak sinyallerinin ve iletim modellerinin bilinmediği durumlarda, karışmış sinyallerden kaynak sinyallerini tahmin etme işlemidir (Zi ve ark., 2022; Comon ve Jutten, 2020). Kokteyl parti problemi olarak araştırma konusu olan bu problem, sonar ve radar sinyal işleme (Ghahramani ve ark., 2014; Jiang ve ark., 2015), kablosuz iletişim (Jin-Wang ve ark., 2014), biyomedikal sinyal işleme (Metsomaa ve ark., 2014), konuşma ve görüntü işlemede (Prakash ve Hepzibha, 2015; Barnie ve Oppenheimer, 2015) yaygın olarak araştırılan bir konudur.

BSS problemi, yaygın olarak bağımsız bileşen analizi (independet component analysis-ICA) yöntemi ile çözülür (Hyvarinen ve Oja, 2000). Bu yöntemlerde ileri istatistik gerektiren bağımsız sinyal ayrıştırması için lineer cebir yaklaşımları kullanılır. En yaygın kullanılan kurtosis ve negentropy yöntemleridir. FastICA bu yaklaşımlardan hızlı ve doğruluk olarak öne çıkan, iteratif yakınsayamaya dayalı bir yöntemdir. FastICA' nın yakınsama yaklaşımından hareketle son zamanlarda meta-sezgisel algoritmalara (MSA) dayalı yaklaşımlarla BSS problemine çözüm araştırılmaktadır (Abbas ve Kabudian, 2017; Kumar ve Jayanti, 2020). MS algoritmalar, lokal optimumlardan kurtulma mekanizmalarına sahip, düşük maliyetli, hızlı ve robust yapay zeka algoritmalarıdır (Zi ve ark., 2022).

Bu çalışmada, MS algoritmalardan sürü tabanlı parçacık sürü optimizasyonu (PSO) algoritması ve ateş böceği algoritması (FA), FastICA algoritması ile karşılaştırılmıştır. FastICA yönteminde de olduğu gibi genelde rastgele üretilen ayırma matrisi, bizim yaklaşımımızda Gram-Schmidt yöntemi ile orthonormal hale getirilerek başarım artırılmıştır. Çalışmanın ilerleyen kısımlarında, materyal ve yöntemlerden bahsedilerek elde edilen bulgular verilmiştir.

2. Materyal ve Metot

Bu kısımda kör kaynak problemi hakkında genel bilgi, bağımsız bileşen analizi yaklaşımı ve kullanılan MSA hakkında bilgi sunulacaktır. Daha sonra önerilen yöntem ve işlem adımları verilecektir.

2.1. Kör Kaynak Ayırma Problemi

BSS problemi bilinmeyen kaynaklardan $S(t) = \{s_1(t), s_2(t), s_3(t), \cdots, s_r(t)\}$ gelen sinyalleri, karışmış halinden $X(t) = \{x_1(t), x_2(t), x_3(t), \cdots, x_r(t)\}$ tahmin etme işlemidir. Burada $S(t)$ sinyallerin, $A_{r\times r}$ matrisi ile karıştığı varsayılmaktadır. A matrisi ve $S(t)$ kaynak sinyalleri bilinmediği

için, karışmış $X(t)$ sinyallerinden $Y(t)$ sinyalleri $W_{r\times r}$ matrisi ile denklem 1'de olduğu gibi tahmin edilmeye çalışılır (Baysal ve Efe, 2023; Feng ve Kowalski, 2018; Jin-Wang ve ark., 2014; Kumar ve Jayanthi, 2020; Li ve ak., 2016; Prakash ve Hepzibha, 2015; Wang, 2021; Zi ve arka., 2020). $W = A^{-1}$ olması durumunda kaynak sinyaller doğru tahmin edilebilir. Ancak, burada A matrisi bilinmemektedir ve W' nin farklı çözümleri her zaman mevcuttur.

$$Y_{rxd} = W_{rxr}X_{rxd} = W_{rxr}A_{rxr}S_{rxd} \quad (1)$$

Burada A ve W, rxr boyutlarında sırasıyla doğrusal karıştırma matrisi ve ayrıştırma matrisidir. r, kaynak sinyallerin sayısı ve genelde bağımsız bileşen analizinde bağımsız bileşen sayısı olur. d, sinyalin boyut sayısı (örneklem sayısı) olarak verilmiştir. A ve $S(t)$ bilinmediğinden W ayırma matrisi, korelasyon azaltma ve Gaussian olmama kriterlerine göre çözülür.

2.2. Bağımsız Bileşen Analizi

Bağımsız bileşen analizi (Independent Component Analysis, ICA), BSS için en yaygın kullanılan yöntemlerden biridir (Pati ve ark., 2021). Bağımsız bileşen analizinde, kaynakların bağımsız ve Gaussian olmayan dağılım değerlerine sahip olduğu kabul edilir. ICA tabanlı geliştirilen çoğu algoritma, ayırma matrisini optimizasyon problemi olarak ele alır. Bu yaklaşımda maliyet fonksiyonu, kaynak sinyallerle ayrıştırılmış sinyaller arasındaki benzerliği, ayrıştırma matrisinin ayrıştırma başarısını oldukça etkilemektedir. Geleneksel ICA yönteminde maliyet fonksiyonu gradyant tabanlı yaklaşımlarla çözülür (Li ve ark., 2016). Bu yaklaşımın yerel optimumlara takılma dezavantajı oldukça fazla olup, Gaussian olmayan fonksiyonlarda yerel minimumlara daha fazla takılma olduğu belirtilmektedir (Kumar ve Jayanti, 2020). ICA, JADE, SOBI gibi yaklaşımlarla çözülmekte olup en yaygın kullanılan algoritma FastICA'dır (Zi ve ark, 2022; Luo ve ark., 2012).

FastICA: FastICA algoritması, kurtosis ve negentropy kullanarak ayrıştırılmış sinyallerin bağımsızlığına bakar. Negentropi maksimum olduğunda, ayrılmış sinyaller arasındaki Gauss olmayan özellik maksimum olur (Luo ve ark., 2012). Bundan hareketle FastICA algoritmasında negentropi veya kurtosis, ayrılmış sinyallerin tutarlılığını ölçmek için kullanılabilir. Sinyalin kurtosisi (4. moment) denklemi 2'deki gibi tanımlanır (Hyvarinen ve Oja, 2000).

$$kurt(y) = k_4(y_i) = E(y_i^4) - 3E(y_i^2)^2 \quad (2)$$

Negentropinin tahmin edilmesi zordur ve bu nedenle bu kontrast fonksiyonu esasen teorik bir fonksiyon olarak kalmaktadır. Pratikte, bazı yaklaşımların kullanılması gerekmektedir. Burada, ICA

için verimli bir yöntem olarak kullanılan yaklaşım verilmiştir (Li ve ark., 2016). Li ve ark., (2016)' na göre, negentropiye yaklaşmanın klasik yöntemi, yüksek dereceli momentler kullanmaktır. Çok değişkenli negentropi yaklaşık olarak denklem 3 ile tanımlanabilir (Zi ve ark., 2022).

$$J_{(y_i)} = \frac{1}{12}k_3^2(y_i) + \frac{1}{48}k_4^2(y_i) + \frac{7}{48}k_3^4(y_i) - \frac{1}{8}k_3^2(y_i)k_4(y_i) \quad (3)$$

Burada $k_3(y_i)$, sinyalin üçüncü dereceden ve $k_4(y_i)$ ise sinyalin dördüncü dereceden kümülantıdır. FastICA için farklı negentropi yöntemleri de kullanılmakta olup, (Hyvarinen ve Oja, 2000; Li ve ark., 2016) çalışmalarında maksimum entropi prensibine dayalı yaklaşımlar kullanılmıştır. FastICA algoritması, kurtosis veya negentropiyi, Newton yaklaşımı ile maksimize etmeye çalışarak ayırma matrisini bulmaya çalışır.

2.3. Meta-sezgisel Algoritmalar

MS algoritmalar, iteratif çözümleme ile optimizasyon yapıp, evrimsel, doğadan etkilenen, biyolojik, insan tabanlı gibi gerçek hayattan seçilen genelde avlanma/beslenme/yaşam süreçlerin modellenmesi ile gerçekleştirilir. Tüm problemler için genel çözüme sahip bir algoritma olmayıp, analitik yöntemler ile kıyaslandığında kabul edilebilir şartlarda kullanılabilen yakınsama sağlayan popüler yaklaşımlardır. Bu kısımda, BSS için tüm MSA algoritmalarına temel oluşturabilecek genel işlem adımları sunulmuştur.

MSA tabanlı ICA genel algoritma adımları:

Adım 1. Merkezileştirme (denklem 4) ve beyazlatma (denklem 5) işlemi yapılır.

$$X(t) = X(t) - E[X(t)] \quad (4)$$

$$X(t) = \Lambda D^{-\frac{1}{2}}\Lambda^T X(t) \quad (5)$$

Beyazlatma denkleminde Λ ve D sırasıyla $X(t)$' nin özvektör ve özdeğerlerin köşegen matrisidir.

Adım 2. Başlangıç parametrelerinin değerleri belirlenir. (İterasyon sayısı, durdurma kriteri, popülasyon sayısı, başlangıç konum, hız, parlaklık, kromozom gibi MS algoritmasına özgü değerler)

Adım 3. Ayrıştırma matrisinin rastgele belirlenmesi (karışmış sinyal sayısı x bağımsız bileşen sayısı boyutlarında)

Adım 4. Uygunluk değerine göre en iyi ayrıştırma matrisinin belirlenmesi (amaç fonksiyonunun optimum noktasının bulunması)

Adım 5. Bireylerin güncellenmesi (ayrıştırma matrisi, konum, hız, parlaklık, çaprazlama, mutasyon gibi MS algoritmalara özgü işlemler)

Adım 6. Durdurma kriter/leri (iterasyon sayısı, eşik değeri) sağlanana kadar Adım 4 ve Adım 5'in tekrar edilmesi

Adım 7. Ayrıştırma matrisi ile sinyallerin ayrıştırılması, $Y(t) = W_{eniyi}X(t)$

2.3.1. Parçacık Sürü Optimizasyonu Algoritması

Sürü tabanlı olan parçacık sürü optimizasyonu algoritması (PSO), pek çok problem için oldukça başarılı sonuçlar üreten yaygın kullanılan MS algoritmalardandır (Bonyadi ve Michalewicz, 2017). PSO algoritmasının adımları şöyledir:

Adım 1. Başlangıç değerleri (parametreler, başlangıç konumu, iterasyon sayısı, yakınsama koşulu ilk hız değerleri) atanır. (c1, c2 : sabit değerler, rand1, rand2 : rastgele üretilen değerler)

Adım 2. Her parçacık için uygunluk değeri, uygunluk fonksiyonundan hesaplanır. Denklem 6'da temsili gösterilen uygunluk fonksiyonu probleme göre değişen amaç fonksiyonudur.

$$fs_i = fitnesFunc(x_i) \tag{6}$$

Adım 3. Hesaplanan uygunluk değerlerinden parçacık için o ana kadar en iyi olan (*pBest*) parçacık konumu ve sürü içindeki o ana kadar en iyi (*gBest*) parçacıkların konumu belirlenir ve güncellenir.

Adım 4. Parçacıkların hızları denklem 7 ile hesaplanır.

$$v_{i+1} = v_i + c_1 * rand_1 * (pBest - x_i) + c_2 * rand_2 * (gBest - x_i) \tag{7}$$

Adım 5. Yeni hız değerlerine göre her parçacığın konumu denklem 8 ile güncellenir.

$$x_{i+1} = x_i + v_{i+1} \tag{8}$$

Adım 6. Durdurma kriteri sağlanana kadar adım 2'ye dönülür.

2.3.2. Ateşböceği Algoritması

Ateşböceği algoritması (FA), doğadan ilham alan veya evrimsel algoritmalar olarak da bilinen biyo-esinlenmiş algoritmalardandır. FA, ateşböceklerinin yanıp sönme davranışlarına göre

hareketlerini taklit eder. Bu algoritma ilk olarak 2007 yılının sonlarında Xin-She Yang tarafından geliştirilmiş ve 2008 yılında yayınlanmıştır (Yang, 2008).

FA' nın uygun tasarımında iki önemli tanım vardır: ışık yoğunluğunun *(I)* değişimi ve çekiciliğin formülasyonu *(β).* Bir ateşböceğinin çekiciliği ışık yoğunluğu veya parlaklığı ile belirlenir. Amaç fonksiyonu, parlaklık ile ilişkilendirilir. Işık yoğunluğu *I(r), r* mesafesi ile monotonik ve üstel olarak azalır ve denklem 9'daki gibi ifade edilebilir.

$$I(r) = I_0 e^{-\gamma r} \tag{9}$$

Denklem 9'da, I_0 orijinal ışık yoğunluğu ve γ ışık soğurma katsayısıdır. Bir ateşböceğinin çekiciliği, komşu ateşböcekleri tarafından görülen ışık yoğunluğu ile orantılı olduğundan, bir ateşböceğinin çekiciliği β, denklem 10 ile ifade edilebilir.

$$\beta = \beta_0 e^{-\gamma r^2} \tag{10}$$

Denklem 10' da, $\beta_0, r = 0$' daki çekiciliktir. Herhangi iki ateşböceği s_i ve s_j arasındaki mesafe öklid mesafesi olarak denklem 11 ile hesaplanabilir.

$$r_{ij} = \|s_i - s_j\| = \sqrt{\sum_{k=1}^{n}(s_{ik} - s_{jk})^2} \tag{11}$$

Burada *n*, problemin boyutunu ifade eder. Birinci ateşböceği, daha çekici başka bir ateşböceği *j*' ye hareket eder. Ateşböceklerinin hareketleri üç şekilde oluşabilir: birinci ateşböceğinin mevcut konumu, daha çekici başka bir ateşböceğine çekilme ve bir rastgeleleştirme parametresi α ve [0, 1] aralığında rastgele üretilen ε_i sayısından oluşan rastgele bir yürüyüş. Böylelikle ateşböceği hareketi denklem 12 ile ifade edilebilir.

$$s_i = s_i + \beta_0 e^{-\gamma r_{ij}^2}\left(s_i - s_j\right) + \alpha\varepsilon_i \tag{12}$$

Ateşböceği algoritmasının işlem adımları :

Adım 1. Ateş böceği popülasyonunun konumlarını arama uzayında rastgele dağıt.

Adım 2. Amaç fonksiyonundan her bir ateş böceğinin uygunluk değerini hesapla.

Adım 3. Her bir ateş böceğinin pozisyonunu, diğer ateş böceklerinin çekicilik ve parlaklığına bağlı olarak güncelle.

Adım 4. Uygunluk değerine bağlı olarak her bir ateş böceğinin parlaklık değerini güncelle.

Adım 5. Durdurma kriteri sağlanana dek adım 2' den devam et.

2.4. Gram-Schmidt Süreci

Doğrusal cebir ve sayısal analizde, Gram–Schmidt süreci bir dizi vektörleri bir iç çarpım uzayı içinde orthonormal etmek için kullanılan bir yöntemdir. Algoritma, bu işlem sonucunda vektörleri orthonormal hale getirir. Bu çalışmada, ayrıştırma matrisinin vektörleri orthonormal hale getirilerek maksimum entropi ve maksimum korelasyon araştırması yapılır. Bu işlem, MS algoritması ile hesaplanan ayrıştırma matrisi, uygunluk fonksiyonunda Gram-Schmidt sürecine tabi tutularak bağımsız vektörler haline getirme şeklinde gerçekleşir.

Gram-Schmidt süreci, $k \leq n$ için sonlu, doğrusal olarak bağımsız bir $S = \{v_1, ..., v_k\}$ vektör kümesini alıp, R_n'nin S ile aynı boyutlu k alt uzayını kapsayan bir $S' = \{u_1, ..., u_k\}$ dik kümesini üretir. Sıfır olmayan bir u vektörü üzerinde v vektörünün izdüşümü denklem 13'deki gibi tanımlanır.

$$projeksiyon_u(v) = \frac{\langle v,u \rangle}{\langle u,u \rangle} u \tag{13}$$

Bilinen k vektör için v_i, u_i vektörleri denklem 14 ile hesaplanır.

$$\begin{aligned} u_1 &= v_1, \\ u_2 &= v_2 - projeksiyon_{u1}(v_2), \\ u_i &= v_i - \sum_{j=1}^{i} projeksiyon_{u_j}(v_i) \end{aligned} \tag{14}$$

Elde edilen u_i dizisi ortogonal vektörler sistemidir ve denklem 15 ile normalleştirilerek bir orthonormal küme oluşturulur. Bu işlemler sarasıyla Gram-Schmidt ortogonalizasyonu ve orthonormalizasyonu olarak adlandırılır. Elde edilen vektör dizisi yeni ayrıştırma matrisi olarak değerlendirilir.

$$e_i = \frac{u_1}{\|u_1\|} \tag{15}$$

2.5. Önerilen Yöntem

Jiali ve ark. (2022) çalışmalarında, 2 karmaşık sinyal için MS algoritmalarını kullanmış, ayırma için korelasyondan faydalanmıştır. Burada karmaşıklık matrisi 2x2 boyutlarında rastgele üretilmiştir. Çalışmalarında sundukları karıştırma matrisi, korelasyon $V_{1\times d}$ boyutlarında vektörün $(V_{1\times d})^r$ hali ile

birleştirilmesi ile oluşturulmuştur. Bu durumda 2'den büyük bağımsız bileşenlerde orta sütunlar aynı olmaktadır.

Çalışmamızda algoritma akışı MS algoritmaları genel adımlarına benzerlik göstermektedir. Ancak bizim çalışmamızın farkı, ayırma matrisinin Gram-Schmidt süreci ile oluşturularak bağımsızlık sağlanmasıdır. Böylece vektörlerin ortalarında oluşan tekrarlılık ta önlenmiş olmaktadır. Daha önemlisi, her bir ayırma vektörü birbirine dik olarak orthonormal formda üretilmektedir. Çalışmada, MS algoritmaları ile ilk ayırma vektörü oluşturulduktan sonra, diğer vektörler rastgele vektörlerin ortogonalleştirilmesi ile elde edilir. Böylece ilk belirlenen ayrıştırma vektörü referans alınarak diğer ayrıştırma vektörleri ortogonal bir şekilde üretilmiş olmaktadır. Sonuç olarak, ilgili MS algoritmasının parlaklık, konum gibi parametrelerine göre belirlenen ilk sinyal ayırma vektöründen diğer ayırma vektörleri, birbirinden bağımsız olacak şekilde üretilmiş olmaktadır.

Çalışmada ana yapı olarak Brian Moore (URL-1, 2024) kodu takip edilmiştir. Brian Moore (URL-1, 2024) paketinin ICA demosunda 3 sinyal üretilip gauss gürültüsü eklenmiştir. Sonra random üretilen bir *A* karıştırma matrisi ile karışmış sinyaller elde edilmiştir. Çalışmalarımızda, her bir sinyal için T- peryot parametresi değiştirilerek FastICA ile MS algoritmaları kıyaslanmıştır.

FastICA çalışmamızda, 150 iterasyonla sınırlandırılmıştır. Sinyal uzunlukları görsel olarak daha iyi sunum için *n=300* alınmıştır. Sunulan demodan farklı olarak çalışmamızda, FA ve PSO algoritmaları, Gram Schmidt süreci tabanlı ayırma matrisi oluşturan amaç fonksiyonu ile geliştirilerek uygulamaya entegre edilmiştir. Çalışmamızda sinyallerin periyotları değiştirilerek FastICA, FA ve PSO algoritmalarının performansı karşılaştırılmıştır. Kullanılan sinyaller Şekil 1'deki gibi üretilip, T-peryot değişkenine göre farklı sinyaller üretilmiştir.

```
% Sinyal Parametreleri
n   = 300;          % Örnekler
T   = [t1,t2,t3];   % Her bir sinyal için periyot
SNR = 50;           % Sinyal SNR

% Sinyal üretme
t = @(n,T) linspace(0,1,n) * 2 * pi * T;
S1 = sin(t(n,T(1)));              % Sinus
S2 = sign(sin(t(n,T(2))));        % Karedalga
S3 = sawtooth(t(n,T(3)));         % Testere dişi

% Gürültü ekleme
sigma = @(SNR,X) exp(-SNR/10)*(norm(X(:))/sqrt(numel(X)));
Si = Si + sigma(SNR,Si) * randn(size(Si));
```

Şekil 1. Kaynak sinyal üretimi

Gerçekleştirilen uygulama Matlab2023 Akademik versiyonda, 8 GB belleğe sahip Intel(R) Xeon(R) CPU E5-1650 v3 @ 3.50GHz donanımlı x64 tabanlı işletim sisteminde geliştirilmiştir. MS

algoritmaların deneysel ayarlanmış parametreleri Tablo 1'de verilmiştir. MS algoritmalarının iterasyon sayısı FastICA ile aynı sayıda uygunluk fonksiyonu çağrısı yapılsın diye 50 tutulmuştur.

Tablo 1. MSA parametreleri

FA Algoritması		**PSO Algoritması**	
Popülasyon sayısı	30	Popülasyon sayısı	30
alpha	0,2	Pozisyon alt sınır	-30
beta	1,8	Pozisyon üst sınır	30
gamma	0,1	Hız alt sınır	-10
theta	0,98	Hız üst sınır	10
alpha_damp	0,98	wdamp	0,99
Alt sınır	-10	c1	1,7
Üst sınır	10	c2	2,3
İterasyon sayısı	50	İterasyon sayısı	50

3. Bulgular ve Tartışma

Tablo 2. Ayrıştırılmış sinyallerin kaynak sinyallerle olan korelasyon katsayıları

Kaynak	**T**	**FastICA**			**FA**			**PSO**		
S1	3	-0.00295	0.006762	**-0.99997**	-0.0508	-0.07501	**0.995889**	-0.01256	0.00209	**0.999919**
S2	4	0.002362	**-0.99996**	-0.00817	**-0.99794**	-0.03708	-0.05228	**0.999831**	-0.01196	0.013992
S3	5	**-0.99991**	0.013638	-1.8E-05	-0.02503	**0.996728**	0.076866	-0.00406	**0.999992**	0.000916
S1	4	-0.00395	-0.0078	**0.999962**	-0.00828	0.063061	**0.997976**	0.001935	-0.00442	**-0.99999**
S2	3	-0.00908	**0.999801**	0.017815	0.077735	**0.995607**	-0.0522	-0.00658	**0.999874**	-0.01448
S3	5	**0.998624**	-0.04436	0.027972	**-0.99957**	0.025651	0.014513	**0.998663**	-0.04668	-0.02224
S1	6	-0.62228	0.508341	0.595282	-0.46666	0.569655	**-0.67656**	0.454526	-0.61238	0.646842
S2	2	0.166478	**0.971691**	-0.16764	**-0.98117**	-0.15178	0.119491	**0.98397**	0.142278	-0.10752
S3	4	**0.691456**	0.023111	**0.72205**	-0.05367	**0.753743**	0.654975	-0.00991	**0.713847**	**0.700232**
S1	6	**0.854584**	0.116767	**0.506017**	**-0.97527**	-0.21714	0.041375	-0.19589	0.024922	**0.98031**
S2	4	-0.09302	**-0.92385**	0.371274	0.04599	-0.01856	**0.99877**	-0.00159	**0.999681**	-0.02522
S3	2	-0.61829	0.66994	0.410975	0.027052	**0.918925**	-0.39351	**0.910947**	-0.40889	-0.05463
S1	4	0.010755	-0.02934	**-0.99951**	0.106173	0.803214	**0.586154**	-0.70109	**-0.66128**	0.266806
S2	4	**0.319191**	-0.35482	-0.87876	0.076419	**0.984229**	0.159542	-0.54528	-0.49944	**0.673229**
S3	4	0.143787	**0.654775**	0.742022	**0.474653**	-0.85471	-0.21019	**0.844962**	0.037168	-0.53353
S1	20	0.701478	-0.00726	**-0.71266**	-0.05564	**0.715641**	-0.69625	**0.711192**	0.005093	-0.70298
S2	30	-0.0012	**-0.99998**	0.006504	**-0.99847**	-0.04717	0.028751	-0.00223	**0.999995**	0.002452
S3	40	**-0.72038**	-0.01483	-0.69342	-0.02284	0.688248	**0.725117**	0.695002	0.010971	**0.718925**
S1	50	**0.679984**	-0.00767	0.733188	**-0.54791**	0.091058	0.831569	0.815922	-0.00258	**0.578157**
S2	50	0.411609	-0.35591	**0.838994**	-0.12384	0.086483	**0.988526**	**0.987105**	0.009676	0.159787
S3	50	-0.66354	**0.611629**	-0.43085	0.141978	**0.446507**	-0.88345	-0.84601	**0.51482**	-0.13867
S1	7	-0.0097	0.008631	**-0.99992**	0.002687	-0.00493	**0.999985**	0.017572	-0.01202	**0.999774**
S2	12	-0.00604	**-0.99996**	-0.00665	0.001873	**0.999994**	0.002995	0.017348	**0.999802**	0.009787
S3	13	**-0.99973**	-0.02118	0.009528	**-0.99958**	0.029022	0.002819	**0.999798**	0.009988	-0.01746
S1	3	0.022135	-0.00603	**0.999737**	0.00447	-0.00412	**0.999982**	-0.00938	-0.00646	**0.999936**
S2	5	0.02411	**0.999703**	-0.00359	-0.01318	**0.999902**	-0.00491	-0.01034	**0.999943**	-0.00273
S3	8	**-0.99911**	0.026567	0.032875	**-0.99983**	-0.01075	0.015017	**0.999718**	0.012742	0.020056

Tablo 2'de satırlarda sinyaller periyot katsayısı (T) ile birlikte verilmiştir. Sütunlarda her bir algoritmanın ayrıştırma sinyali ile kaynak sinyallerinin korelasyon katsayısı verilmiştir. Tabloda mutlak 1'e yakın olan en çok benzeşen sinyallerdir. Yani sütunlarda maksimum olan değerli hücre, eşleşen kaynak sinyalini ifade eder. Tablo 2'de T={6,2,4} periyotlarında sadece FA'nın, T={6,4,2} periyotlarında ise FA ve PSO algoritmalarının tüm sinyalleri ayrıştırdığı görülmektedir. Diğer durumlarda başarı aynı olup neredeyse %99 korelasyon ile sinyaller ayrıştırılmaktadır.

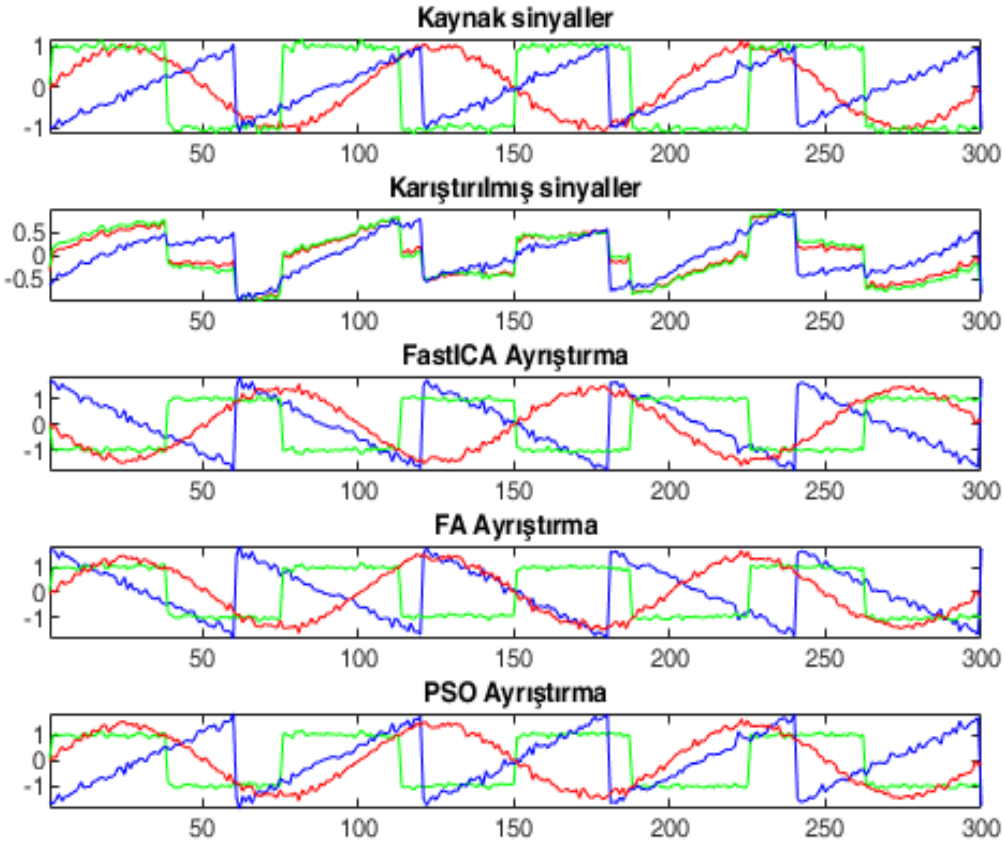
Tablo 3'de, kök ortalama kare hata (RMSE) değerleri verilmiştir. Tablonun 2-10 satırlarında, yöntemlerin 3 kaynak sinyalinin, 3-12 sütunlarında farklı periyotlar için RMSE değerleri verilmiştir. Son sütun, kaynak sinyallerinin farklı periyotlardaki RMSE değerinin ortalamasıdır. Son 3 satır her bir yöntemin deney bazlı farklı 3 kaynak sinyalin RMSE değerlerinin ortalamasıdır.

Buna göre son üç satır incelendiğinde, PSO {(6,4,2), (50,50,50), (7,12,13), (3,5,8)} periyotlu işaretleri daha iyi ayrıştırırken diğer iki algoritma 2'şer deneyde daha başarılı olabilmiştir. Son 3 satırın son sütunlarında tüm deneylerin genel ortalaması verilmiştir. Buna göre kullanılan yöntemlerin yapılan deneylerdeki başarı sırası PSO > FA > FastICA şeklindedir.

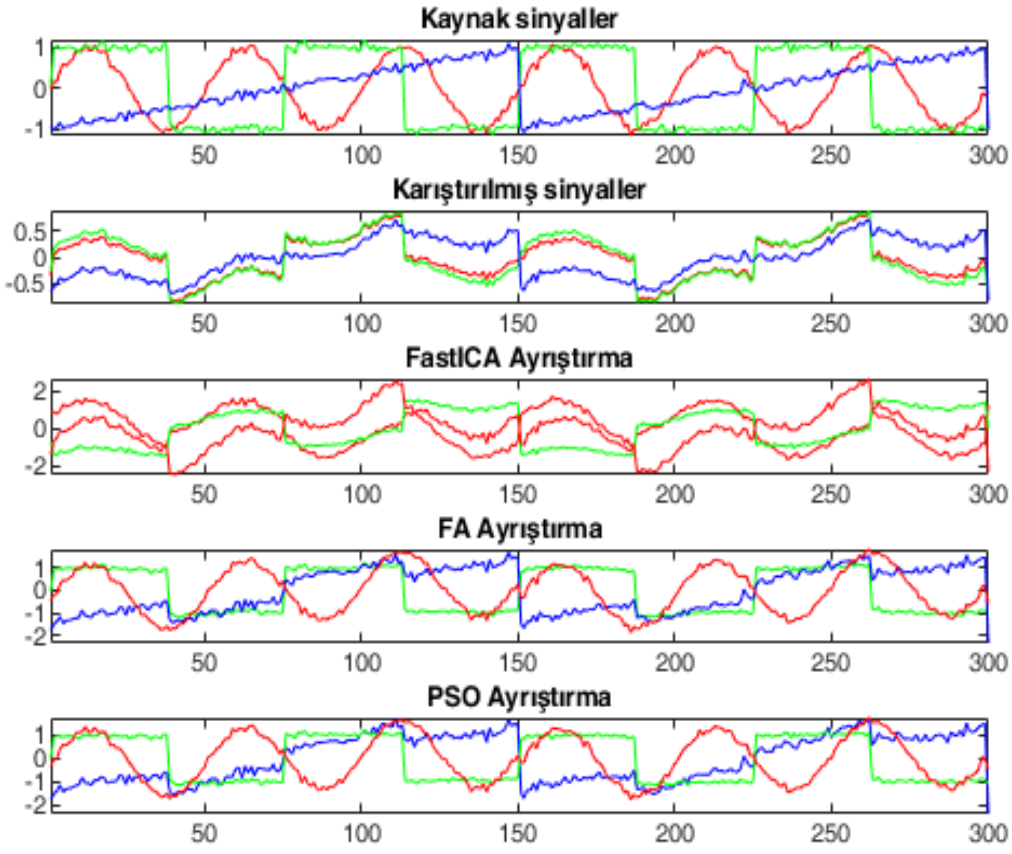
Tabloda kaynak sinyal bazlı değerlendirmelerde en iyi sonuç, ilgili deney sütunda kalın olarak sunulmuştur. Örneğin, (3,4,5) periyotlu deney için S1 ve S2 kaynaklarına en yakın sinyalleri FA ayırırken, S3 kaynağına en yakın sinyali PSO üretmiştir. Buna göre 9 deneyin hepsinin değerlendirilmesinde, toplam 27 sinyalin 6'sını FastICA, 9'nu FA ve geriye kalan 12'sini PSO daha bağımsız bulmuştur. PSO'nun genel ortalamasında da RMSE en düşük çıkmaktadır.

Tablo 3. Sinyallerin RMSE değerleri

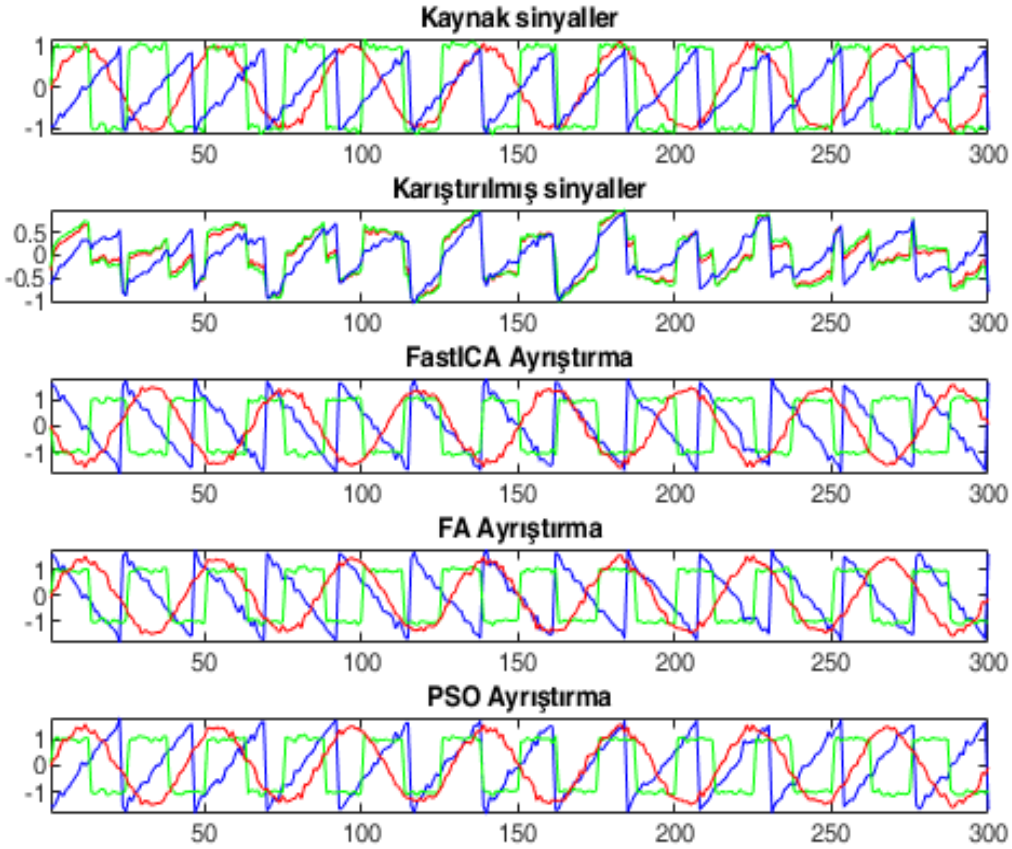
	T	(3,4,5)	(4,3,5)	(6,2,4)	(6,4,2)	(4,4,4)	(20,30,40)	(50,50,50)	(7,12,13)	(3,5,8)	G.Ort.
FastICA	**S1**	0.0764	0.0178	1.1745	0.9145	2.0869	1.1888	**1.6750**	0.0658	0.0475	0.8052
	S2	0.1122	**0.0092**	**0.0216**	0.7764	**0.0322**	0.1098	2.3972	0.1255	0.0343	0.4020
	S3	2.5458	0.6703	0.2716	1.8174	**0.9909**	2.0425	**1.7073**	2.4607	2.5270	1.6704
FA	**S1**	**0.0485**	0.0966	**0.9865**	1.7165	2.7938	1.2096	2.9354	0.0159	**0.0192**	1.0913
	S2	**0.0168**	0.0380	0.0464	**0.5243**	0.3464	1.0272	0.3859	**0.0021**	0.0250	0.2680
	S3	2.5448	**0.6637**	**0.0836**	0.9137	1.3151	**0.8825**	1.1994	2.4607	2.5263	1.3989
PSO	**S1**	1.5736	**0.0126**	1.1861	**0.3019**	**0.4855**	**1.1208**	3.0119	**0.0157**	0.0271	0.8595
	S2	0.0940	0.0159	1.0967	0.0755	2.8651	**0.0035**	**0.0714**	0.0314	**0.0123**	0.4740
	S3	**1.0033**	0.6701	0.1037	**0.8437**	1.2976	0.1666	1.2800	**0.5852**	**0.6526**	0.7337
S. Ort.	**FastICA**	0.9115	**0.2324**	0.4892	1.1694	**1.0367**	1.1137	1.9265	0.8840	0.8696	0.9592
	FA	**0.8700**	0.2661	**0.3721**	1.0515	1.4851	**1.0398**	1.5069	0.8262	0.8568	0.9194
	PSO	0.8903	0.2329	0.7955	**0.4071**	1.5494	0.4303	**1.4545**	**0.2108**	**0.2307**	**0.6890**



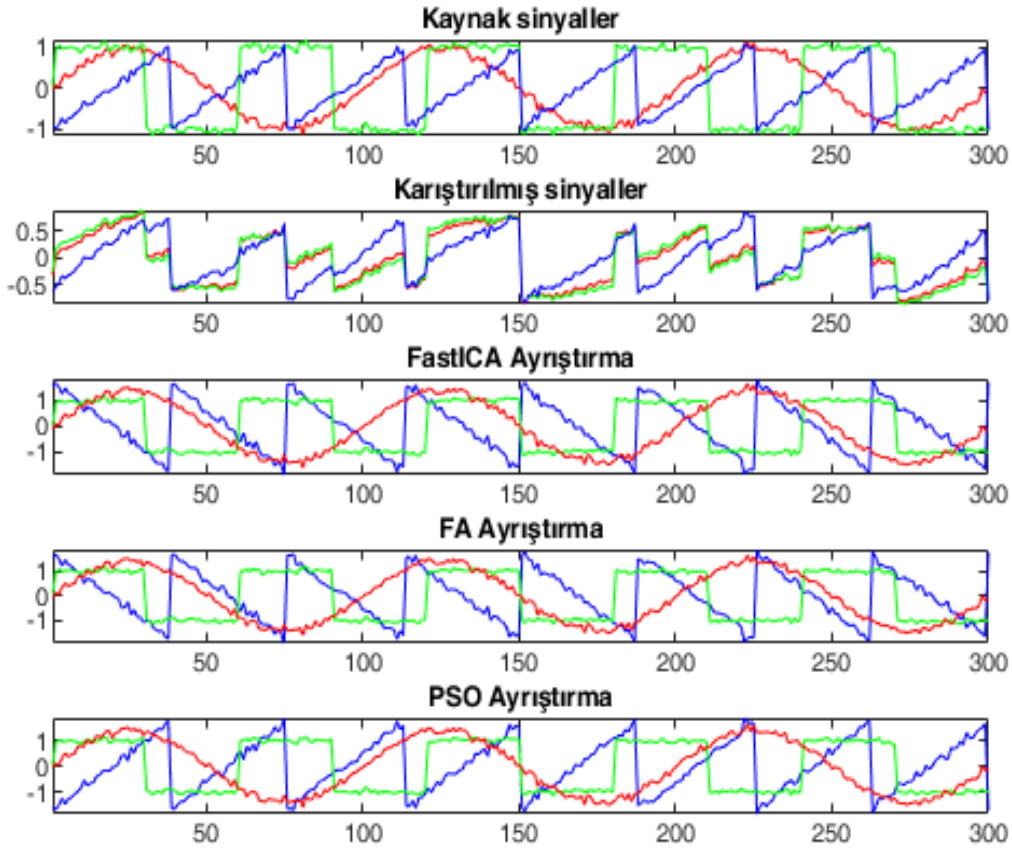
Şekil 2. (3,4,5) periyotlarındaki sinyallerin ayrıştırılması



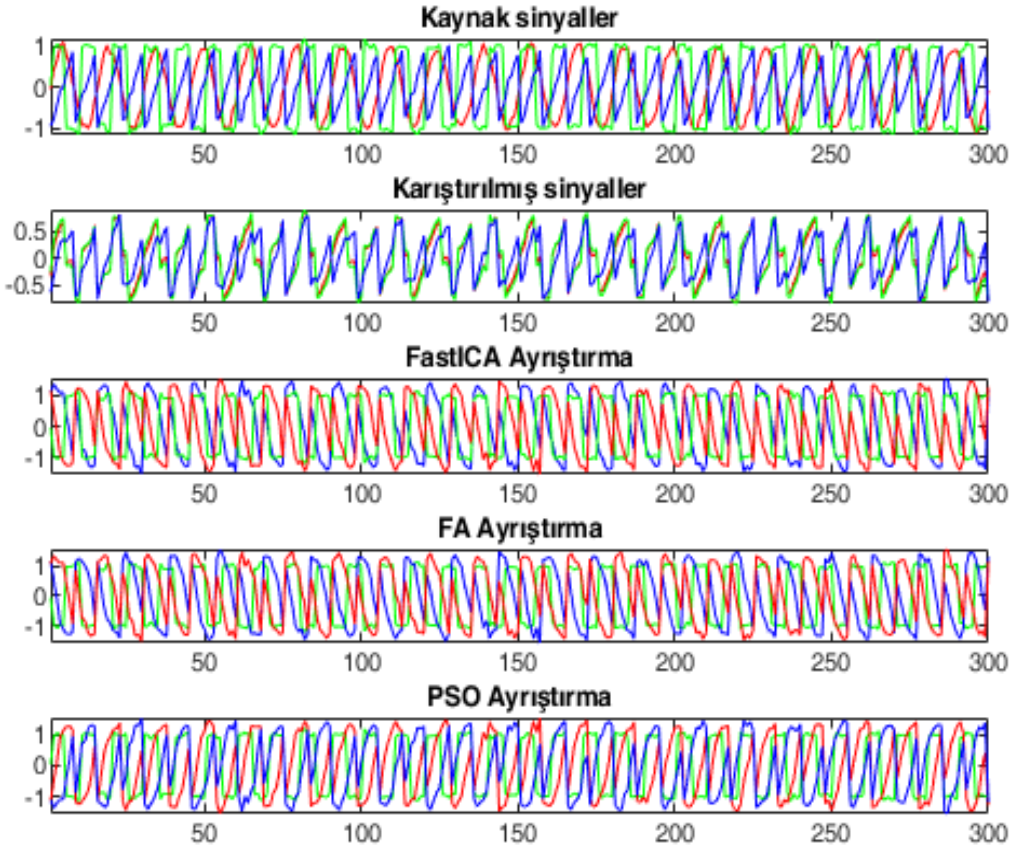
Şekil 3. (6,4,2) periyotlarındaki sinyallerin ayrıştırılması



Şekil 4. (7,12,13) periyotlarındaki sinyallerin ayrıştırılması



Şekil 5. (3,5,8) periyotlarındaki sinyallerin ayrıştırılması



Şekil 6. (20,30,40) periyotlarındaki sinyallerin ayrıştırılması

Şekil 2-6'da, yapılan deneylerdeki, karıştırma ve ayırma sonuç grafiklerinin bir kısmı verilmiştir. Şekil 2 ,4 ve 5'te tüm algoritmalar başarılı bir şekilde ayrıştırma yapabildiği gözükmektedir. Bunlardan bir kısmı negatif korelasyona sahiptir ancak bunlar üstesinden gelinebilen problemlerdir. Şekil 6'da ayrıştırma başarılı gözükmektedir. Tablo 2'ye göre bu kombinasyonda her algoritma sinyalleri farklı kaynaklarla eşleştirmiştir.

Şekil 3'de FastICA algoritmasının bulduğu farklı üç sinyalden ikisinin, kaynak sinyallerle karşılaştırıldığında tek bir sinyale daha çok benzediği görülmektedir. Gerçekleştirilen uygulamada, en çok benzeyenler aynı renge boyandığı için Şekil 3'ün FastICA satırında aynı renkli iki sinyal mevcuttur. Burada, korelasyon tabanlı eşleştirme yapıldığı için, esas kaynağa çok uzak bir belirleme yapıldığı sonucuna varılabilir.

4. Sonuçlar ve Öneriler

Bu çalışmada kör kaynak ayırma problemi için MS algoritmaların kullanılabilirliği görülmüştür. Mevcut FastICA algoritmasında yakınsama sürekli artmayıp dalgalanma yaparken MS algoritmalar her iterasyonda ya daha çok yakınsama yapmaktadır ya da en uygun çözümden

vazgeçmemektedir. Önerilen ayırma matris üretim yaklaşımı ile daima orthonormal uzayda ayırma yapılmakta ve bunun başarıyı artırdığı görülmüştür. Çalışmada orthonormal uzay, Gram-Schmidt süreci ile sağlanmıştır. Genel olarak bakıldığında PSO algoritmasının RMSE ortalama değeri diğer yaklaşımlardan daha iyi çıkmaktadır.

Sonraki çalışmalarda, orthonormal ayırma matrisi üretildikten sonra yerel arama yapıp daha iyi ayırma yapan orthonormal şartı olmayan ayırma matrisi türetilebilir mi diye araştırılabilir. Biyomedikal sinyal işlemede, belirli bir örüntünün karmaşık sinyallerde olup olmadığı MSA tabanlı BSS algoritmaları ile araştırılarak otomatik hastalık teşhisinde kullanılabilir. Bu çalışmada yaygın kullanılan MS algoritmaları denenmiştir. Problem için daha başarılı MS algoritmalarının olup olmadığı üzerine araştırma yapılabilir.

Yazarların Katkısı

Tüm yazarlar çalışmaya eşit katkıda bulunmuştur.

Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

Kaynaklar

Abbas, N., and Kabudian, J., (2017). Speech Scrambling based on Independent Component Analysis and Particle Swarm Optimization, The International Arab Journal of Information Technology (IAJIT) , 14(4), 109–115.

Baysal, B., and Efe, M.Ö., (2023). A comparative study of blind source separation methods. Turkish Journal of Electrical Engineering and Computer Sciences, 31(7). https://doi.org/10.55730/1300-0632.4047

Bonyadi, M.R., and Michalewicz, Z., (2017). Particle swarm optimization for single objective continuous space problems: a review. Evolutionary Computation. 25(1),1–54. doi:10.1162/EVCO_r_00180

Comon, P., and Jutten, C. (2010). Handbook of Blind Source Separation: Independent Component Analysis and Applications. Oxford, UK: Academic Press.

Feng, F., and Kowalski, M., (2018). Revisiting sparse ICA from a synthesis point of view: Blind Source Separation for over and underdetermined mixtures. Signal Processing, 152, 165–177. https://doi.org/10.1016/j.sigpro.2018.05.017

Ghahramani, H., Barari, M., and Bastani, M.H. (2014). Maritime radar target detection in presence of strong sea clutter based on blind source separation. IETE Journal of Research, 60(5), 331–344. https://doi.org/10.1080/03772063.2014.961573

Hyvärinen, A., and Oja, E., (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13 4–5, 411-430. https://doi.org/10.1016/S0893-6080(00)00026-5

Jiang, L., Li, L., and Zhao, G.Q. (2015, January). Pulse-compression radar signal sorting using the blind source separation algrithms. International Conference on Estimation, Detection and Information Fusion (ICEDIF) (pp.268-271). Harbin. doi: 10.1109/ICEDIF.2015.7280204

Jin-Wang, H., Jiu-Chao, F., and Shan-Xiang, L. (2014). Blind source separation of chaotic signals in wireless sensor networks. Acta Physica Sinica, 63(5): 050502. https://doi.org/10.7498/aps.63.050502

Kumar, M., and Jayanthi, V.E., (2020). Blind source separation using kurtosis, negentropy and maximum likelihood functions. International Journal of Speech Technology, 23, 13–21. https://doi.org/10.1007/s10772-019-09664-z

Li, H., Li, Z., and Li, H., (2016). A Blind Source Separation Algorithm Based on Dynamic Niching Particle Swarm Optimization. MATEC Web of Conferences 61, 03008. DOI: 10.1051/matecconf/20166103008

Luo, D., Sun, H., and Wen, X., (2012). Research and Application of Blind Signal Separation Algorithm to the Aircraft Engine Vibration Signal and Fault Diagnosis Based on Fast ICA. J. Converg. Inf. Technol. 7(10), 248–254. http://dx.doi.org/10.4156/jcit.vol7.issue10.29

Metsomaa, J., Sarvas, J., and Ilmoniemi, R.J. (2014). Multi-trial evoked EEG and independent component analysis. Journal of Neuroscience Methods, 228, 15–26. https://doi.org/10.1016/j.jneumeth.2014.02.019

Pati, R., Pujari, A.K., Gahan, P., and Kumar, V., (2021). Independent Component Analysis: A Review, with Emphasis on Commonly used Algorithms and Contrast Function. Computación y Sistemas, 25(1). doi: 10.13053/CyS-25-1-3449

Prakash, K., and Hepzibha Rani, D., (2015, January). Blind Source Separation for Speech Music and Speech Mixtures. International Journal of Computer Applications. 110, 40-43. DOI=10.5120/19372-1087

URL-1: Moore, B., PCA and ICA Package, MATLAB Central File Exchange. (https://www.mathworks.com/matlabcentral/fileexchange/38300-pca-and-ica-package), (Erişim tarihi 23 Nisan 2024).

Wang, R., (2021). Blind Source Separation Based on Adaptive Artificial Bee Colony Optimization and Kurtosis. Circuits System and Signal Processing, 40, 3338–3354. https://doi.org/10.1007/s00034-020-01621-5

Yang, X. S. (2008). Nature-Inspired Metaheuristic Algorithms. United Kingdom: University of Cambridge, Luniver Press. ISBN 978-1-905986-10-1.

Zi J., Lv D., Liu J., Huang X., Yao W., Gao M., Xi R., and Zhang Y. (2022). Improved Swarm Intelligent Blind Source Separation Based on Signal Cross-Correlation, Sensors, 22(1),118. https://doi.org/10.3390/s22010118