

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

UYARLAMALI HUZME BİÇİMLENDİRME

**YÜKSEK LİSANS TEZİ
Harun ÜNALDI**

Anabilim Dalı : Elektronik ve Haberleşme Mühendisliği

Programı : Telekomünikasyon Mühendisliği

HAZİRAN 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

UYARLAMALI HUZME BİÇİMLENDİRME

**YÜKSEK LİSANS TEZİ
Harun ÜNALDI
504071314**

**Tezin Enstitüye Verildiği Tarih : 28 Nisan 2009
Tezin Savunulduğu Tarih : 1 Haziran 2009**

**Tez Danışmanı : Doç. Dr. Işın ERER
Diğer Jüri Üyeleri : Prof. Dr. Ahmet Hamdi KAYRAN
Prof. Dr. Serhat ŞEKER**

HAZİRAN 2009

ÖNSÖZ

Tez çalışması boyunca tecrübe, fikir ve bilgilerinden yararlandığım Sayın Doç. Dr. Işın Erer'e teşekkür ederim. Fikir ve tecrübelerini benimle paylaşan aynı zamanda yazılım desteği sağlayan Sayın Araş. Gör. Özgür Gültekin'e teşekkürlerimi sunarım.

Uluslararası kaynaklara erişim kolaylığı sağlayan İstanbul Teknik Üniversitesi Kütüphanesi ve ULAKBİM'e ayrıca teşekkür ederim.

Yüksek Lisans eğitimim süresince maddi destek sağlayan T.C Kredi ve Yurtlar Kurumu'na teşekkür ederim.

Benden hiç bir zaman desteğini esirgemeyen, beni anlayışla karşılayan anneme...

Nisan 2009

Harun Ünaldı

İÇİNDEKİLER

Sayfa

ÖZET.....	xi
SUMMARY	xiii
1. GİRİŞ	xiv
1.1 Uyarlamalı Huzme Biçimlendirme Tarihçesi	4
1.2 Tezin İçeriği	5
2. ANTEN DİZİLERİYLE İŞARET İŞLEME.....	7
2.1 Dizi Teorisi.....	7
2.1.1 Bazı anten parametre tanımları	8
2.1.1.1 Işıma güç yoğunluğu	9
2.1.1.2 Işıma şiddeti	9
2.1.1.3 Yönlendirilebilirlik.....	9
2.1.1.4 Işıma örüntüsü.....	10
2.1.2 Düzgün doğrusal anten dizileri	10
2.1.2.1 Dizi faktörünün sıfır ve maksimumları	14
2.1.2.2 Faz taramalı diziler.....	16
2.2 Akıllı Antenler	18
2.2.1 Akıllı anten sistem çeşitleri	19
2.2.1.1 Anahtarlama huzme sistemleri.....	20
2.2.1.2 Uyarlamalı huzme sistemleri.....	21
2.2.1.3 Akıllı anten sistem çeşitlerinin karşılaştırılması	23
2.2.1.4 Akıllı anten sisteminin yararları.....	24
2.3 Uyarlamalı Huzme Biçimlendirme	25
2.3.1 Uyarlamalı huzme biçimlendirme problem kurulumu	26
2.3.2 Uyarlamalı huzme biçimlendirme yöntemleri	29
2.3.2.1 Geliş yönü bulma algoritmaları.....	29
Kaynak sayısının kestirilmesi	29
MUSIC yöntemi.....	31
ESPRIT yöntemi	32
2.3.2.2 Yan lob engelleyici	33
2.3.2.3 LCMV	34
2.3.2.4 Sıfır yönlendirme huzme biçimlendirme	35
2.3.2.5 Sabit genlik algoritması	35
2.3.2.6 LMS algoritması	36
2.3.2.7 RLS algoritması	37
2.3.2.8 SMI algoritması	38
3. HUZME BİÇİMLENDİRME	39
3.1 LMS Algoritması	39
3.1.1 LMS algoritmasıyla huzme biçimlendirme.....	40
3.1.2 LMS algoritması simülasyon sonuçları.....	43
3.2 RLS Algoritması	50
3.2.1 RLS algoritmasıyla huzme biçimlendirme.....	50

3.2.2 RLS algoritması simülasyon sonuçları.....	53
3.3 LMS ve RLS Algoritmalarının Karşılaştırılması	57
3.4 Genelleştirilmiş Yan Lob Engelleyici	58
3.4.1 LCMV huzme biçimlendirici	58
3.4.1.1 Frost'un uyarlamalı LCMV algoritması	59
3.4.2 Genelleştirilmiş Yan Lob Engelleyici	61
3.4.3 Simülasyon sonuçları	64
4. KAFES KESTİRİCİLERLE HUZME BİÇİMLENDİRME	69
4.1 İleri ve Geri Kestirici Hataları.....	69
4.2 Kafes Kestiriciler	75
4.3 Ortak Süreç Kestirimi	77
4.4 Kafes Yapısının Anten Dizilerine Uygulanması.....	79
4.4.1 Kafes algoritmasıyla huzme biçimlendirme.....	79
4.4.2 Kafes algoritması simülasyon sonuçları.....	84
4.5 Uyarlamalı Kafes Algoritmasıyla Genelleştirilmiş Yan Lob Engelleyici.....	88
4.5.1 Kafes-GSLC simülasyon sonuçları.....	88
5. SONUÇLAR	91
5.1 İlerideki Çalışmalar	92
KAYNAKLAR	93

KISALTMALAR

ADC	: Analog to Digital Converter
AF	: Array Factor
AIC	: Akaike Information Criteria
BER	: Bit Error Rate
CDMA	: Code Division Multiple Access
CMA	: Constant Modulus Algorithm
DOA	: Direction of Arrival
DSP	: Digital Signal Processing
ESPRIT	: Estimation of Signal Parameters via Rotational Invariance Technique
FDMA	: Frequency Division Multiple Access
GAL	: Gradient Adaptive Lattice
GSLC	: Generalized Sidelobe Canceller
GSM	: Global System for Mobile Communication
HPBW	: Half Power Beamwidth
LCMV	: Linearly Constrained Minimum Variance
LMS	: Least Mean Squares
LS	: Least Squares
MDL	: Minimum Descriptive Length
MIMO	: Multiple Input Multiple Output
MMSE	: Minimum Mean Square Error
MSE	: Mean Square Error
MUSIC	: Multiple Signal Classification
MVDR	: Minimum Variance Distortionless Response
NLMS	: Normalized Least Mean Squares
Radar	: Radio Detection and Ranging
RF	: Radio Frequency
RLS	: Recursive Least Squares
SDMA	: Space Division Multiple Access
SINR	: Signal to Interference and Noise Ratio
SIR	: Signal to Interference Ratio
SMI	: Sample Matrix Inversion
SNR	: Signal to Noise Ratio
Sonar	: Sound Navigation and Ranging
TDMA	: Time Division Multiple Access
ULA	: Uniform Linear Array

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1	: Uyarlamalı anten dizisi.....	2
Şekil 1.2	: İki antenli uyarlamalı yan lob engelleyici.....	5
Şekil 2.1	: Anten ışıma örüntüsü.....	11
Şekil 2.2	: İki antenli dizi yapısı.....	12
Şekil 2.3	: ULA yapısı.....	14
Şekil 2.4	: AF' nin d ile değişimi.....	15
Şekil 2.5	: AF'nin N ile değişimi.....	16
Şekil 2.6	: AF'nin N ve d ile değişimi.....	17
Şekil 2.7	: $\theta_0 = 60^\circ$ için faz taramalı dizi AF biçimi.....	18
Şekil 2.8	: Anahtarlamalı sistemlerde çoklu huzme yapısı.....	20
Şekil 2.9	: Anahtarlamalı sistem birimleri.....	21
Şekil 2.10	: Uyarlamalı sistem yapısı.....	22
Şekil 2.11	: Uyarlamalı huzme biçimlendirme şebekesi.....	27
Şekil 2.12	: DOA algoritmasının huzme biçimlendirme şebekesinde kullanımı....	30
Şekil 3.1	: LMS algoritmasıyla huzme biçimlendirme.....	40
Şekil 3.2	: LMS algoritması blok diyagramı.....	42
Şekil 3.3	: Hatanın zamanla değişimi.....	44
Şekil 3.4	: Ağırlık genliklerinin zamanla değişimi.....	44
Şekil 3.5	: Huzme biçimi.....	45
Şekil 3.6	: Hatanın zamanla değişimi ($\alpha = 0.0001$).....	45
Şekil 3.7	: Hatanın zamanla değişimi ($\alpha = 0.01$).....	46
Şekil 3.8	: Huzme biçimi ($\alpha = 0.01$).....	46
Şekil 3.9	: Hatanın zamanla değişimi.....	47
Şekil 3.10	: Huzme biçimi.....	48
Şekil 3.11	: Huzme biçiminin zamanla değişimi.....	48
Şekil 3.12	: Huzme biçimi ($M = 7$).....	49
Şekil 3.13	: Huzme biçimi ($M = 10$).....	50
Şekil 3.14	: RLS algoritması akış diyagramı.....	53
Şekil 3.15	: Hatanın zamanla değişimi.....	54
Şekil 3.16	: Ağırlık genliklerinin zamanla değişimi.....	54
Şekil 3.17	: Huzme biçimi.....	55
Şekil 3.18	: Hatanın zamanla değişimi.....	56
Şekil 3.19	: Ağırlık genliklerinin zamanla değişimi.....	56
Şekil 3.20	: Huzme biçimi.....	57
Şekil 3.21	: Ağırlık genliklerinin zamanla değişimi.....	60
Şekil 3.22	: Huzme biçimi.....	61
Şekil 3.23	: GSLC yapısı.....	63
Şekil 3.24	: Uyarlamalı GSLC yapısı.....	64
Şekil 3.25	: LMS-GSLC huzme biçimi.....	65

Şekil 3.26	: RLS-GSLC huzme biçimi (a).....	65
Şekil 3.27	: RLS-GSLC huzme biçimi (b).	66
Şekil 3.28	: LMS-GSLC huzme biçimi	67
Şekil 3.29	: RLS-GSLC huzme biçimi	67
Şekil 4.1	: İleri kestirici hata filtresi.	71
Şekil 4.2	: Geri kestirici hata filtresi.	73
Şekil 4.3	: İleri ve geri kestirim hataları işaret akış diyagramı	77
Şekil 4.4	: Ortak süreç yapısı işaret akış diyagramı.....	78
Şekil 4.5	: Kafes algoritması ileri ve geri hata işaret akış diyagramı.	82
Şekil 4.6	: Kafes algoritması ortak süreç kestirimi	82
Şekil 4.7	: Hatanın zamanla değişimi.	84
Şekil 4.8	: Ağırlık genliklerinin zamanla değişimi.	85
Şekil 4.9	: Huzme biçiminin zamanla değişimi.	85
Şekil 4.10	: Hatanın zamanla değişimi.	86
Şekil 4.11	: Ağırlık genliklerinin zamanla değişimi.	87
Şekil 4.12	: Huzme biçiminin zamanla değişimi.	87
Şekil 4.13	: Kafes algoritmasıyla GSLC yapısı.	88
Şekil 4.14	: w_a ağırlık katsayılarının değişimi.	89
Şekil 4.15	: Kafes-GSLC huzme biçimi.	89
Şekil 4.16	: w_a ağırlık katsayılarının değişimi.	90
Şekil 4.17	: Kafes-GSLC huzme biçimi.	90

UYARLAMALI HUZME BİÇİMLENDİRME

ÖZET

İlk çalışmaları 1950'li yıllarda başlayan uyarlamalı filtreler bir çok alanda uygulama bulmuştur. Uyarlamalı huzme biçimlendirme teknikleri de uyarlamalı filtrelere paralel olarak gelişim göstermiştir. Uyarlamalı anten dizilerini kullanan akıllı antenler son yıllarda sayısal teknolojinin hızlı gelişimiyle birlikte kablosuz haberleşmede anahtar teknoloji olmuştur.

Kablosuz haberleşmede çok hızlı artan kullanıcı sayısına hizmet verebilmek için, kapasiteyi artıran önemli teknolojilerden SDMA teknolojisi sabit antenler yerine akıllı antenler ya da uyarlamalı anten dizilerini kullanmaktadır. Uyarlamalı anten dizileri, sahip oldukları uyarlamalı işaret işleyici birim sayesinde anten dizisi huzme biçimini istenen kullanıcı yönünde maksimum ışımaya sahip olacak şekilde yönlendirirken aynı zamanda değişik yönlerden gelen diğer kullanıcı işaretlerini mümkün olduğu kadar zayıflatmaya çalışır. Bu sistem yazılım kontrolüyle çalışır, anten dizisinin herhangi bir yöne, mekanik olarak yönlendirilmesi söz konusu değildir.

Uyarlamalı anten dizilerinde, işaret işleyici birimde kullanılan değişik kriterlere göre çalışan bir çok uyarlamalı algoritma mevcuttur. Aynı zamanda değişik tekniklere göre çalışan uyarlamalı huzme biçimlendirme yöntemleri de yaygın olarak kullanılmaktadır. Bu tez çalışmasında huzme biçimlendirme yöntemleri ve kullanılan uyarlamalı algoritmalar üzerinde durulmuş, bu algoritma ve yöntemler, performansları bilgisayar simülasyonlarıyla değerlendirilerek birbirleriyle karşılaştırılmıştır. Anten dizisi olarak Düzgün Doğrusal Dizi (ULA) yapısı kullanılmıştır. Referans işareti kullanan LMS ve RLS algoritmalarıyla elde edilen huzme biçimleri simülasyonlarla sunulmuştur. Aynı zamanda bir çok avantajı beraberinde getiren, Kafes kestiricilerle huzme biçimlendirme yöntemi anlatılmış ve performansı simülasyonlarla değerlendirilmiştir.

Ayrıca referans işaretine ihtiyaç duymayan LCMV huzme biçimlendirme yöntemlerinden Genelleştirilmiş Yan Lob Engelleyici huzme biçimlendirici, LMS ve RLS algoritmalarıyla gerçekleştirilerek simülasyon sonuçları değerlendirilmiştir. Alternatif bir yöntem olarak ise Kafes kestirici algoritması Genelleştirilmiş Yan Lob Engelleyici yapısının uyarlamalı kısmında kullanılmış, simülasyon sonuçlarından interferans işaretleri geliş açılarını yakalayan düzgün huzme biçimleri verdiği görülmüştür.

ADAPTIVE BEAMFORMING

SUMMARY

Adaptive filters, emerged in late 1950s, have been applied to many different fields. Adaptive beamforming techniques have evolved subsequently to the improvements in adaptive filters. With rapid development of digital technology, smart antennas which use adaptive antenna arrays have been a key technology in wireless communications.

SDMA, one of the emerging capacity improving technologies, uses smart antennas or adaptive arrays instead of fixed antennas to serve the increasing demand in wireless communications. Adaptive arrays are used to maximize the radiation in the desired direction by using their adaptive signal processor unit while weakening the signals of interferers which arrive from different angles. This cancelling system is based on software control, antenna arrays are not mechanically steered to any direction.

At the signal processor unit of adaptive arrays, based on different performance criteria, an extensive amount of adaptive algorithms are used. There are many adaptive beamforming methods based on different techniques. In this thesis, adaptive algorithms for adaptive beamforming are investigated. These methods are also compared by evaluating their performance via computer simulations. In the simulations, ULA is chosen as the antenna array geometry. The beamforms obtained by the LMS and RLS algorithms using a reference signal are illustrated. Moreover, the lattice estimator, which has some desirable properties, is adapted to beamforming and its performance is evaluated by simulations.

The Generalized Sidelobe Canceller, which is a different way of realizing of blind adaptive LCMV beamforming, is implemented with LMS and RLS algorithms and the simulation results are evaluated. As an alternative method, the lattice estimator is used at the adaptive side of the Generalized Sidelobe Canceller. It is shown by the simulation results that the lattice estimator as Generalized Sidelobe Canceller forms beams which properly intercept the arrival angles of interference signals.

1. GİRİŞ

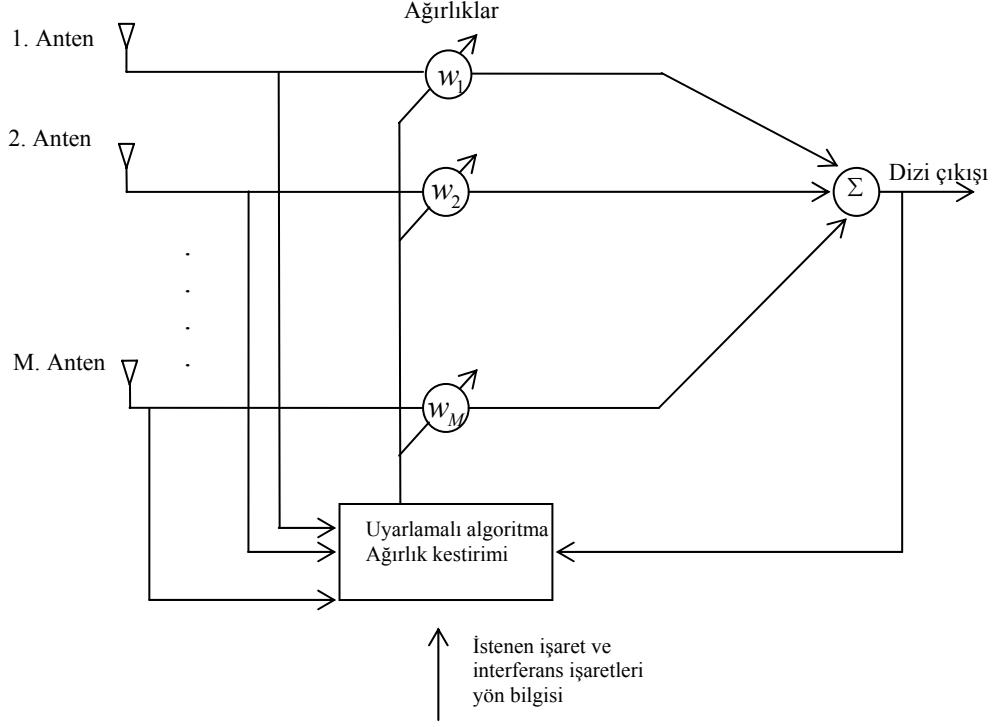
2. nesil kablosuz hücreli sistemler TDMA (Time Division Multiple Access), CDMA (Code Division Multiple Access) ve GSM (Global System for Mobile Communication) gibi çeşitli çoklu erişim tekniklerini kullanmaktadırlar. Anten dizileri kullanımı bu sistemlerde ve 3. nesil sistemlerde önemli miktarda performans iyileşmesi sağlar.

Uyarlamalı anten terimi her bir anten elemanına uygulanan ağırlıkların dinamik olarak değiştiği faz taramalı diziler için kullanılır. İşaretler alınmaya başladıktan itibaren ağırlıklar uyarlamalı olarak güncellenir; uyarlama işlemi sistem tarafından kontrol edilir ve dizi örüntüsü değişen durumlara uyarlanabilir. Dizi çıkışı SNR oranı, interferans işaretleri yönü için alınan işaretler engellenerek maksimize edilir. Bu durumda dizi örüntüsü istenen işaret yönünde ana huzmeye sahiptir ve interferans işaretleri yönünde sıfırlara sahiptir. İnterferans işaretleri geliş yönlerinin değişmesi durumunda da uyarlamalı algoritma her bir anten dizi çıkışına uygulanan ağırlıkları güncelleyerek değişen işaret durumları için dizi örüntüsünü değişen duruma uydurabilir.

Akıllı anten terimi dizi örüntüsünü, dinamik olarak değişen çevreye uyarlayan anten dizilerini ifade eder. Akıllı anten içeren bir sistem anten dizisi tarafından alınan işaretleri işler. Darband işaret durumu için böyle bir sistem Şekil 1.1'de görülmektedir. Bu sistem aynı zamanda istenen işaretin geliş açısı ve interferans işaretleri geliş açıları ek bilgilerini de kullanır. Bu bilgiler sisteme DOA (Direction of Arrival) algoritması tarafından sağlanır. Anten dizisi tarafından alınan işaretler uyarlanabilir karmaşık ağırlıklar tarafından çarpılarak anten dizisi çıkış işareti elde edilir.

Genellikle bir dizi, uzamsal olarak yayılan işaretleri alır ve belirli bir yönden gelen işareti elde etmek için bu işaretleri işler; uzamsal ayırıcı bir filtre gibi çalışır. Bu uzamsal filtreleme işlemi huzme biçimlendirme olarak isimlendirilir. Dizi işleyicisi, ağırlıklandırılmış anten dizisi çıkış işaretlerini işleyerek istenen yöne huzme

yönlendirir. Huzme biçimlendirici istenen yönden gelen işaretin alınımını maksimum yaparken istenmeyen yönden gelen işaretleri mümkün olduğu kadar zayıflatmayı amaçlar.



Şekil 1.1 Uyarlamalı anten dizisi.

Yüksek hızlı ADC (Analog to Digital Converter) teknolojisinin gelişimi ve bütün kablosuz haberleşme biçimlerindeki hızlı gelişmeler akıllı anten sistemlerine olan ihtiyacı artırmıştır. Akıllı anten konusu 1950'lerin sonlarında [1-3] araştırmaya başlanmasına rağmen, uygulamaya konulabilmesi için gerekli teknoloji son zamanlarda elde edilmiştir. İlk zamanlar uyarlamalı algoritmalar analog donanımsal yapıya sahip olduklarından, ilk akıllı antenler ya da uyarlamalı anten dizileri sınırlı kapasiteye sahiptiler. ADC ve DSP (Digital Signal Processing) teknolojilerindeki gelişmeler artık yazılımsal olarak çok hızlı çözümler sunmaktadır [4]. Akıllı antenler değişik uygulamalarda kullanılmaktadır [1,5,6-10].

Kablosuz haberleşme ve algılamada çok hızlı değişim ve gelişmelerle birlikte akıllı antenler oldukça popüler olmuştur. Kablosuz haberleşmedeki bir çok uygulamada akıllı antenler, kapasite ve bandgenişliğini artırmak, SIR (Signal to Interferans Ratio) oranını artırmak, sönümlemeyi azaltmak, MIMO (Multiple Input Multiple Output) iletişimi geliştirmeyi amaçlamaktadır. Diğer uygulamalarda ise akıllı antenlerden

güvenli iletişimi sağlamada, yön bulmada, dalgabıçımı çeşitleme tekniklerinde ve MIMO radar uygulamalarında yararlanılmaktadır. Her iki grup da aynı nedenler için akıllı anten kullanımını ön plana çıkarmakta fakat konuya farklı gereksinim ve farklı bakış açılarıyla yaklaşmaktadır.

Son yıllarda özellikle hareketli haberleşme alanında kablosuz kullanıcı sayısında çok hızlı bir artış yaşandı. Geleceğin kablosuz hareketli haberleşme sistemleri çok daha karmaşık ve çok daha yaygın olacaktır. Bu büyüme hem kapsama alanının hem kapasitenin hem de daha iyi hizmet kalitesine olan ihtiyacın artmasına neden olmuştur. Sınırlı kaynakların etkin kullanımını sağlamak için bir çok yeni teknoloji araştırılmış ve uygulamaya konulmuştur. Hareketli hücreli teknolojide kapasiteyi artırmanın bir yolu geniş bir kapsama alanını küçük altıgen (hexagonal) hücrelere bölmektir. Her bir hücreye komşu hücrelerden farklı frekans band genişlikleri tahsis edilir. Ancak aynı frekans birbirine interferans yaratmayacak kadar uzakta yer alan hücreler tarafından yeniden kullanılabilir. Bu hücreler aynı frekansı tekrar kullandıkları için önemli bir kapasite artışı sağlanabilir. Ancak artan abone sayısını karşılayabilmek için hücre sayısının artırımı etkin ve ekonomik bir çözüm değildir. İlk nesil hücreli sistemler analog frekans bölme yöntemini kullanırken (FDMA), sayısal modülasyon tekniklerini kullanan 2. nesil hücreli sistemler zaman bölmeli çoklu erişim yöntemini kullanmaktadır; 3. nesil haberleşme sistemleri ise kod bölmeli çoklu erişim tekniğini (CDMA) kullanır.

Yakın bir zamana kadar kablosuz hizmetler için artan ihtiyacı karşılamakta bu teknolojiler çok önemli katkı sağladılar. Ancak kişisel kablosuz haberleşme üstel bir biçimde büyümeye devam etmektedir ve hareketli haberleşmede gelecekteki kapasite ihtiyacını karşılayabilmek için yeni teknolojilere ihtiyaç duyulmaktadır. Uzay bölmeli çoklu erişim (SDMA) mevcut haberleşme sisteminde şebeke kapasitesini artırmak için hareketli radio kanalının uzamsal alanını (space domain) kullanır. Sabit antenler kullanan kablolu sistemlerin tersine SDMA sistemler değişen trafik durumlarına kendisini uydurabilen akıllı antenleri ya da uyarlamalı dizileri kullanır. Akıllı antenler genellikle, farklı kullanıcılara hizmet etmek için dar huzmeler ışımlayan baz istasyonlarında çalışırlar. Kullanıcılar uzamsal olarak yeterli derecede ayrık oldukları zaman, aynı hücre içinde olsalar dahi , aynı frekans yeniden kullanılabilir. Uzamsal ayırmaya yönelik bu hücre içi kanal yeniden kullanımı, sistem kapasitesi artırımı başarımında anahtar bir yöntemdir.

Uyarlamalı huzme biçimlendirme olarak bilinen akıllı anten tekniği üzerine kurulmuş en gelişmiş yaklaşım sadece istenen hareketli kullanıcıya maksimum ışımayı yönlendirmez, aynı zamanda kullanıcıyı takip ederken interferans işaretleri yönünde sıfırlar yerleştirir. Uyarlama, gelen işaretlerin karmaşık ağırlıklarla çarpılıp birleştirilerek istenen ışımaya örüntüsünün elde edilmesiyle gerçekleştirilir. Bu karmaşık ağırlıklar işaret çevresindeki değişime uyum sağlamak için uyarlamalı olarak hesaplanır. Farklı kriterlere göre karmaşık ağırlıkların hesabı yazılımsal olarak işaret işleyici birimde gerçekleştirilir; değişik kriterlere göre çalışan bir çok uyarlamalı algoritma vardır [11-19].

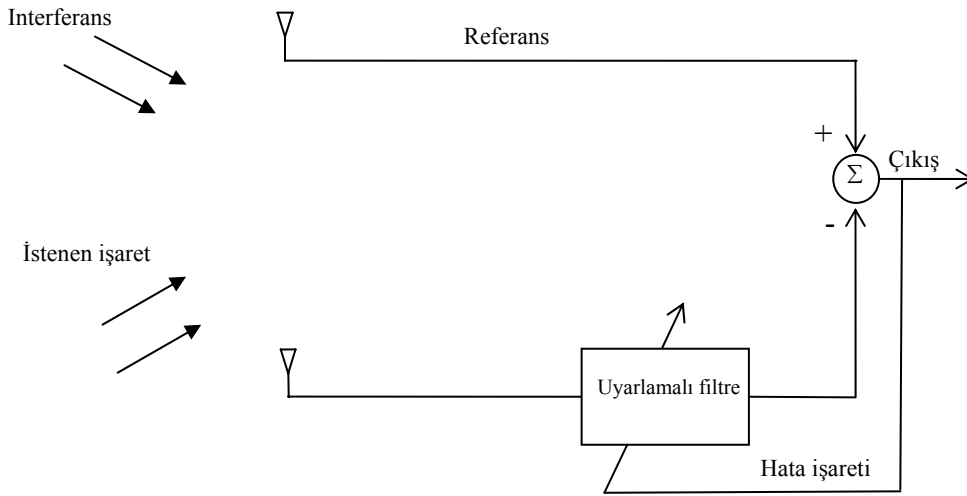
1.1 Uyarlamalı Huzme Biçimlendirme Tarihçesi

Anten dizileri çoğu haberleşme sisteminde kullanılmaktadır. İlk kullanımlardan birisi 1930'lu yıllardaki atlas okyanusu üzerinden gerçekleştirilen kırsadalga haberleşmesidir [20].

İlk basit uyarlamalı anten 1950'lerin sonunda Howells [2] ve Applebaum [21] tarafından geliştirilen uyarlamalı yan lob engelleyicidir, (Adaptive Sidelobe Canceller). Böyle bir sistemde uyarlamalı gürültü gidermedeki mikrofonlara benzer şekilde, iki tane her yöne bakan anten kullanılmaktadır. Bu antenlerden bir tanesi referans işaretini alırken diğere anten uyarlamalı filtrenin giriş işaretini almaktadır. Gerçekte her iki anten de interferans işaretlerini ve istenen işareti alır, fakat antenler uzamsal olarak ayrılabilir olduklarından bu iki işaret uygun filtre kullanılarak birbirinden ayrılabilir. Bu yapı SIR düşük olduğu durumlarda çok iyi çalışır. Ancak SIR yüksek olduğu durumlarda istenen işaretin zayıflamasına neden olur. Bu etkiyi azaltmak için ilave gürültü eklenebilir ya da özel uyarlamalı algoritmalar kullanılabilir.

Diğere bir uyarlamalı huzme biçimlendirme yöntemi referans işareti kullanır. Bu yöntem Widrow, Mantey, Griffiths ve Goode [22] tarafından geliştirilmiştir. İstenen işaret yönünde bir huzme biçimlendirirken aynı zamanda interferans işaretleri yönünde sıfırlar yerleştirir. Uyarlamalı huzme biçimlendirme problemini çözen farklı bir metot 1969 yılında Capon tarafından önerilmiştir. Capon çıkış gücünü minimum yapan ağırlık katsayı vektörünün sağlanması gereken bir şart önerdi, bu çeşit bir

sınırlandırılmış uyarlamalı huzme biçimlendirme yapısı MVDR olarak bilinir (Minimum Variance Distortionless Response).



Şekil 1.2 İki antenli uyarlamalı yan lob engelleyici.

Griffiths [23] ve Frost [24] daha basit ve bazı durumlarda daha iyi performans gösteren benzer yapılar geliştirmişlerdir. 1976'da Swarner, Ksienski, Compton ve Huff [25] radar öncelikli uygulamalarda kullanılan uyarlamalı huzme biçimlendiriciyi kablosuz haberleşme sistemlerine uygulamışlardır.

1.2 Tezin İçeriği

Bu tez çalışmasında ağırlıklı olarak mevcut huzme biçimlendirme yöntemleri üzerinde durulmuş, çok kullanılan bazı huzme biçimlendirme yöntem ve algoritma performansları bilgisayar simülasyonlarıyla değerlendirilmiştir. Ayrıca alternatif huzme biçimlendirme yöntemlerinden Kafes kestirici algoritmasıyla huzme biçimlendirme yöntemi üzerinde durulmuş, bu yapının da en az diğer yöntemler kadar iyi çalıştığı tespit edilmiştir. Bu tez çalışmasında önerilen alternatif bir yöntem olarak ise Kafes kestirici algoritması Genelleştirilmiş Yan Lob Engelleyici (GSLC: Generalized Sidelobe Canceller) huzme biçimlendirme yapısına uygulanmış, yine simülasyon sonuçlarıyla performansı değerlendirilmiştir.

2. bölümde genel olarak anten dizileriyle işaret işleme teknikleri anlatılmaktadır. Tez boyunca kullanılan dizi geometrisi olan ULA yapısı, uyarlamalı akıllı antenlerin çalışma şekli, genel huzme biçimlendirme yöntem ve algoritmaları, bazı önemli geliş açısı bulma teknikleri bu bölümde anlatılmaktadır.

3. bölümde ise çok sık kullanılan huzme biçimlendirme algoritmalarından LMS (Least Mean Squares) ve RLS (Recursive Least Squares) algoritmaları detaylı olarak anlatılmakta, bu iki algoritmayla elde edilmiş bilgisayar simülasyon huzme biçimleri sunulmaktadır. Bu iki algoritmanın performansları birbirleriyle karşılaştırılmış olumlu ve olumsuz yönleri tartışılmıştır. Ayrıca, referans işaretine gerek duymadan çalışan huzme biçimlendirme yöntemlerinden LCMV (Linearly Constrained Minimum Variance) yapısı yine bu bölümde anlatılmaktadır. LCMV'nin değişik bir gerçekleştirme yöntemi olan GSLC huzme biçimlendirme yöntemi LMS ve RLS algoritmalarıyla gerçekleştirilerek performansları simülasyonlarla değerlendirilip tartışılmıştır.

4. bölüm Kafes kestiriciler konusuna ayrılmıştır. Kafes kestirici yapısının uyarlamalı anten dizilerine uygulanış biçimi anlatılmadan önce bu yapının anlaşılabilmesi için genel Kafes kestiriciler ve Kafes kestirici algoritması detaylarıyla anlatılmıştır. Anten dizilerine uygulanmış bir Kafes algoritması üzerinde durulmuş ve bu algoritma kullanılarak elde edilen huzme biçimleri bilgisayar simülasyonlarıyla sunularak performansı değerlendirilmiştir. Bu bölümde ayrıca önerilen farklı bir yöntem olarak Kafes algoritmasının, Genelleştirilmiş Yan Lob Engelleyici huzme biçimlendiricinin uyarlamalı kısmında kullanımı anlatılmış bu yapıyla elde edilen huzme biçimlendirme simülasyon sonuçları sunulup değerlendirilmiştir.

Son bölümde ise tez boyunca kullanılan algoritma ve huzme biçimlendirme yöntemleri birbirleriyle kıyaslanarak sonuçlar değerlendirilmiştir.

2. ANTEN DİZİLERİYLE İŞARET İŞLEME

Anten dizileriyle işaret işleme bir çok alanda önemli uygulamalara sahiptir. Modern radar ve sonar sistemlerinin çoğu anten dizilerini kullanmaktadır. Haberleşme sistemlerinde kapasite artırmak için faz taramalı anten dizilerinin kullanımı gün geçtikçe artmaktadır.

Bu bölümde ilk olarak doğrusal anten dizileri anlatıldıktan sonra akıllı anten sistemleri tartışılmakta ve mevcut sistemler karşılaştırılmaktadır. Uyarlamalı huzme biçimlendirme şebekesinin ihtiyaç duyduğu bazı geliş açısı bulma algoritmalarından bahsedildikten sonra çok sık kullanılan uyarlamalı huzme biçimlendirme algoritmaları anlatılmaktadır.

2.1 Dizi Teorisi

Tek bir anten geniş bir ışına örüntüsüne sahiptir, yönlendirilebilirliği azdır, dolayısıyla uzak mesafe iletişimi için uygun değildir. Yine de antenin fiziksel boyutu artırılarak yüksek yönlendirilebilirlik elde edilebilir. Birbirleriyle bağlantılı ışına elemanlarının uzamsal olarak sıralanmasından oluşan anten dizileri, yönlü ışına örüntüsü üretmek için kullanılabilir. Anten dizileri çeşitli geometrik yapıda olabilir [26-29], en genel yapı ise doğrusal dizidir. Diziler genellikle özdeş anten elemanlarından oluşur. Bu alandaki bir çok çalışma, hareketli iletişim sistemlerinde anten dizilerinin kullanımının kanal kapasitesi ve spektrum etkinliğini, kapsama alanını artırdığını göstermektedir. Anten dizilerinin kullanımı aynı zamanda çoklu yol sönmemesini (multipath fading), ortak kanal interferansını (co channel interference) ve bit hata oranını (BER) azaltır [30-31].

Anten dizisi, istenen işaret ve istenmeyen ortak kanal interferans işaretlerinin farklı yönlerden geldiği bilgisini kullanır. Anten dizisi huzme biçimi, farklı anten çıkışlarının uygun şekilde ağırlıklandırılıp birleştirilmesiyle ayarlanır. Faz kaydırmalı alıcı anten dizilerinde huzme biçimlendirmek için farklı antenler tarafından alınan işaretler genellikle RF (Radio Frekans) biriminde toplanır. Uyarlamalı anten

dizilerinde ise uyarlamalı şebeke her bir anten çıkış işaretinin faz ve genliğini kontrol eder, bu durum dizideki antenler tarafından alınan işaret ağırlıklarının ayarlanmasını iteratif olarak sağlayan, uyarlamalı algoritmaların kullanımıyla sağlanır. İşaretler, ana huzme yönünün dışındaki yönlerden gelen interferans işaretlerini engellemek için, istenen işaret gücünün toplam interferans ve gürültü işaret gücüne oranını (SINR) maksimum yapacak şekilde birleştirilir.

Aynı karakteristikli antenlerden oluşan bir anten dizisinde, dizinin huzme biçimini etkileyen faktörleri şu şekilde sıralayabiliriz:

- Dizinin geometrik yapısı
- Dizi elemanları arasındaki mesafe
- Dizideki her bir antenin uyarılma genliği
- Dizideki her bir antenin uyarılma fazı
- Dizideki her bir antenin göreceli huzme örüntüsü

Uyarlamalı anten dizileri tek bir çıkış oluşturmak için, genlik kontrolü ve faz kaydırması yapan anten şebekesinden oluşur. Hücresel uygulamalarda anten dizileri baz istasyonlarında kullanıldıklarında, istenen kullanıcı yönünde yüksek kazanç sağlarken interferans işaret kaynakları yönünde ise alıcı işaret gücünü sıfır (null) yapabilme yeteneğine sahiptirler.

Genlik ve faz kontrolü her bir anten çıkışı için kompleks ağırlık katsayılarının uygulanmasıyla gerçekleştirilir. Her bir anten çıkışındaki kompleks ağırlık katsayıları uygun değerlerde seçilerek anten dizisinin huzmesi θ_d istenen işaret yönüne yönlendirilebilir, bu durumda θ_i interferans işaret yönlerinden gelen işaretlerin etkisi minimum yapılır. Çok kullanılan en basit anten dizisi, doğrusal şekilde antenlerin sıralanması ile oluşturulur. Bu şekildeki diziye ULA dizi (Uniform Linear Array) denir.

2.1.1 Bazı anten parametre tanımları

Anten dizilerini ayrıntılı olarak açıklamadan önce bazı anten parametrelerini anlamak faydalı olacaktır.

2.1.1.1 Işıma güç yoğunluğu (Radiation power density)

Bütün ışıklanan anten alanları uzaktaki antenlerin alabileceği bir güç taşır; haberleşme sistemlerinde bu güç kullanılır. Işıma güç yoğunluğu W_r anten tarafından herhangi bir yönde yapılan ortalama ışımaya gücünün bir ölçüsüdür ve Poynting vektörünün, \bar{P} , zaman ortalaması alınarak hesaplanır.

$$\bar{P} = \bar{E} \times \bar{H} \quad (W / m^2) \quad (2.1)$$

$$\bar{W}_r(r, \theta, \phi) = \frac{1}{2} \text{Re}[\bar{E} \times \bar{H}^*] = \frac{1}{2\eta} |\bar{E}(r, \theta, \phi)|^2 \quad (Watts / m^2) \quad (2.2)$$

E , elektrik alan kuvvetini, H manyetik alan kuvvetini ve η ise ortamın geçirgenliğini belirtir.

2.1.1.2 Işıma şiddeti (Radiation intensity)

Işıma şiddeti U , mesafe ile normalize edilmiş güç yoğunluğu olarak görülebilir, ışımaya gücündeki mesafe bağımlılığını ortadan kaldırarak uzak alan örüntü çizimlerini mesafe bağımsız yapar. Işıma şiddeti, belirli bir yönde birim açı başına antenin yaptığı ışımaya gücünü belirtir. Işıma yoğunluğu ile r mesafesinin karesinin çarpımıyla hesaplanır.

$$U = r^2 W_r \quad (Watts / radyan) \quad (2.3)$$

Toplam ışımaya gücü ise antenin tüm yönlerde yaptığı toplam ışımaya gücünü belirtir.

$$P_{top} = \int_0^{2\pi} \int_0^{\pi} W_r(r, \theta, \phi) r^2 \sin(\theta) d\theta d\phi = \int_0^{2\pi} \int_0^{\pi} U(\theta, \phi) \sin(\theta) d\theta d\phi \quad (Watts) \quad (2.4)$$

2.1.1.3 Yönlendirilebilirlik (Directivity)

Yönlendirilebilirlik, antenin aynı toplam güç yayan yönbağımsız (isotropic) bir antene göre ne kadar yönlendirilebilir olduğunun bir ölçüsüdür. Diğer bir ifadeyle, yönlendirilebilirlik, yönbağımlı bir antenin toplam güç yoğunluğunun, aynı güce sahip yönbağımsız bir antenin güç yoğunluğuna oranıdır.

$$D = \frac{4\pi U(\theta, \phi)}{P_{top}} = \frac{4\pi r^2 W_r(r, \theta, \phi)}{\int_0^{2\pi} \int_0^{\pi} W_r(r, \theta, \phi) r^2 \sin(\theta) d\theta d\phi} = \frac{4\pi U(\theta, \phi)}{\int_0^{2\pi} \int_0^{\pi} U(\theta, \phi) \sin(\theta) d\theta d\phi} \quad (2.5)$$

Maksimum yönlendirilebilirlik ise (2.5) ifadesinin maksimum değeri hesaplanarak bulunur.

$$D_0 = \frac{4\pi U_{\max}(\theta, \phi)}{P_{top}} = \frac{4\pi U_{\max}(\theta, \phi)}{\int_0^{2\pi} \int_0^{\pi} U(\theta, \phi) \sin(\theta) d\theta d\phi} \quad (2.6)$$

$U_{\max}(\theta, \phi)$, maksimum ışımaya şiddetini belirtir. Yönbağımsız antenler tüm yönlerde eşit olarak ışımaya yaptıkları için, yönlendirilebilirlik daima 1'e eşittir; dolayısıyla bu antenler yönlendirilebilir değildir.

2.1.1.4 Işıma örüntüsü (Radiation pattern)

Işıma örüntüsü antenin alan şiddetinin, konum ya da açının fonksiyonu olarak değişimi olarak tanımlanabilir. Yöne bağımlı olan (anisotropic) bir anten bazı yönlerde diğer yönlerden daha güçlü ışımaya yapar. Böyle bir antenin ışımaya örüntüsü Şekil 2.1'de görüldüğü gibi bir çok lobdan oluşmaktadır. Loblardan biri diğerlerine göre en güçlü ışımaya şiddetine sahiptir ve ana lob olarak isimlendirilir, daha zayıf alan şiddetine sahip diğer bütün loblar küçük lob olarak isimlendirilir. Ana huzme genişliği, yarım güç huzme genişliği (HPBW: Half Power Beamwidth) esas alınarak hesaplanır ve huzmenin yarı güç noktaları arasındaki açıdır.

2.1.2 Düzgün doğrusal anten dizileri (ULA: Uniform Linear Array)

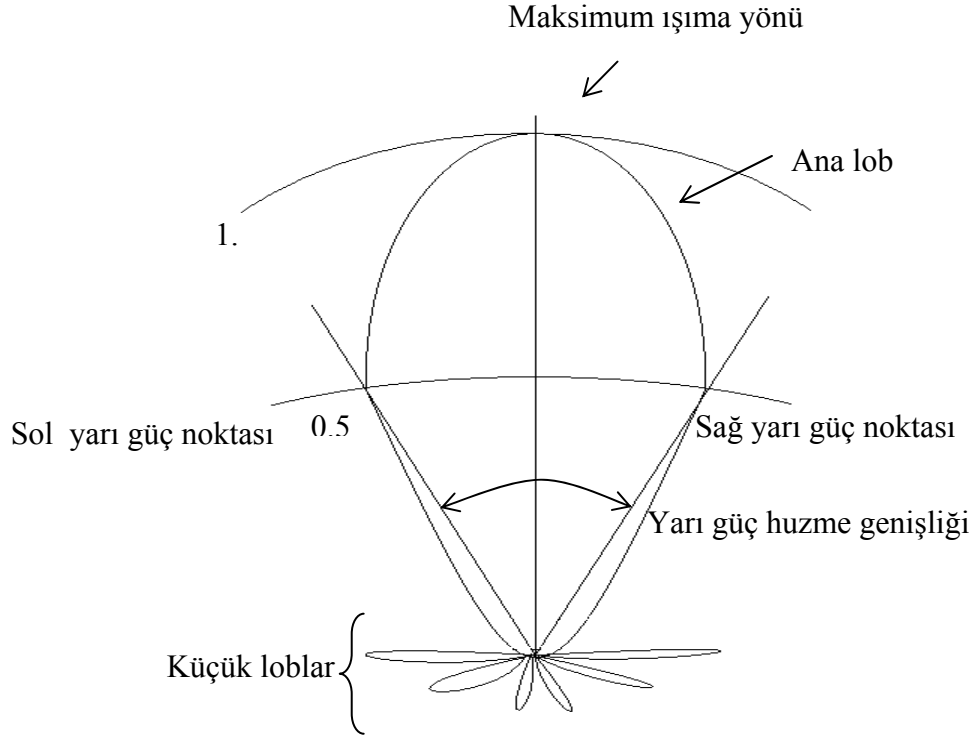
İki antenli bir yapıda uzaktaki bir P noktasında oluşturulan elektrik alan şiddetleri aşağıdaki ifadeyle verilir;

$$\bar{E}_1 = w_1 f_1(\theta_1, \phi_1) \frac{e^{-j\left(kr_1 - \frac{\beta}{2}\right)}}{r_1} \quad (2.7)$$

İkinci antenden kaynaklanan elektrik alan şiddeti ise;

$$\bar{E}_2 = w_2 f_2(\theta_2, \phi_2) \frac{e^{-j\left(kr_2 - \frac{\beta}{2}\right)}}{r_2} \quad (2.8)$$

w_1 ve w_2 ağırlıkları, f_1 ve f_2 herbir anten için normalize alan örüntüsünü, r_1 ve r_2 sırasıyla birinci ve ikinci antenin P gözlem noktasına olan uzaklıklarını, β ise iki antenin beslemeleri arasındaki faz farkını belirtir.



Şekil 2.1 Anten ışıma örüntüsü.

Eğer P noktası çok uzakta ise şu yaklaşıklıklar kabul edilebilir;

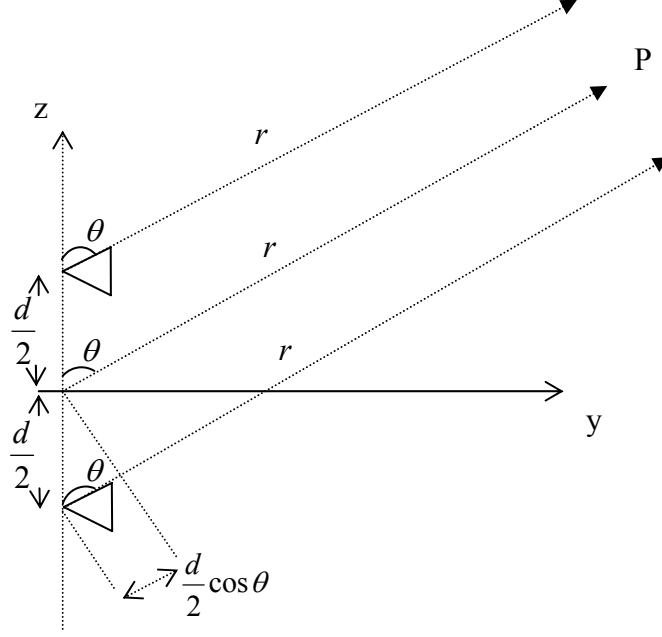
$$\theta_1 \cong \theta_2 \cong \theta \quad (2.9)$$

$$r_1 \cong r_2 \cong r ; \text{genlik de\u011fi\u015fimleri i\u00e7in} \quad (2.10)$$

$$\left. \begin{array}{l} r_1 \cong r - \frac{d}{2} \cos \theta \\ r_2 \cong r + \frac{d}{2} \cos \theta \end{array} \right\} \text{Faz de\u011fi\u015fimleri i\u00e7in} \quad (2.11)$$

Dizi elemanları özde\u015f oldukları i\u00e7in ışıma örüntü \u015fiddetleri e\u015fit olacaktır;

$$F_1(\theta_1, \phi_1) = F_2(\theta_2, \phi_2) = F(\theta, \phi) \quad (2.12)$$



Şekil 2.2 İki antenli dizi yapısı.

P noktasındaki toplam elektrik alan şiddeti her bir antenden kaynaklanan elektrik alan şiddet vektörlerinin toplamına eşit olacaktır;

$$\bar{E} = \bar{E}_1 + \bar{E}_2 = w_1 f(\theta, \phi) \frac{e^{-j\left(k\left(r - \frac{d}{2}\cos\theta\right) - \frac{\beta}{2}\right)}}{r} + w_2 f(\theta, \phi) \frac{e^{-j\left(k\left(r + \frac{d}{2}\cos\theta\right) + \frac{\beta}{2}\right)}}{r} \quad (2.13)$$

$$\bar{E} = \frac{e^{-jkr}}{r} f(\theta, \phi) \underbrace{\left[w_1 e^{j\left(k\frac{d}{2}\cos\theta + \frac{\beta}{2}\right)} + w_2 e^{-j\left(k\frac{d}{2}\cos\theta + \frac{\beta}{2}\right)} \right]}_{\text{Dizi Faktörü (AF: Array Factor)}} \quad (2.14)$$

Eşit ağırlıklandırma için;

$$\bar{E} = w \frac{e^{-jkr}}{r} f(\theta, \phi) \underbrace{2 \cos\left(\frac{kd \cos\theta + \beta}{2}\right)}_{AF} \quad (2.15)$$

Bu eşitlik, örüntü çarpımı olarak bilinir, uzak bir noktada dizinin oluşturduğu toplam elektrik alan şiddeti dizi merkezindeki tek bir antenin elektrik alan şiddetiyle dizi faktörünün (AF) çarpımına eşittir. Bu kural sadece özdeş antenlerden oluşmuş diziler için geçerlidir.

$$\bar{E}_{top} = \bar{E}_{merkezde\ tek\ bir\ anten} \times AF \quad (2.16)$$

İki antenli dizi için normalize edilmiş dizi faöktörü ise;

$$AF_n = \cos\left(\frac{kd \cos \theta + \beta}{2}\right) \quad (2.17)$$

Bu eşitliklere göre AF şu etkenlere bağlıdır;

- Dizi eleman sayısına
- Dizinin geometrik yapısına
- Dizi elemanlarının uyarılma genliklerine
- Dizi elemanları arasındaki faz farkına

Düzgün doğrusal bir dizi eşit mesafeyle sıralanmış elemanlardan oluşur; her bir anten eşit akımla beslenir ve antenler arasında sabit bir faz farkı vardır. İki anten için yapılan gösterim N elemanlı düzgün doğrusal dizi için genişletilirse;

Yönbağımsız N tane antenden oluşmuş düzgün doğrusal bir dizi için AF aşağıdaki eşitlikle verilir.

$$AF = 1 + e^{j(kd \cos \theta + \beta)} + e^{j2(kd \cos \theta + \beta)} + \dots + e^{j(N-1)(kd \cos \theta + \beta)} \quad (2.18)$$

(2.18) ifadesi düzenlenirse;

$$AF = e^{j\left(\frac{N-1}{2}\right)\psi} \frac{\sin\left(\frac{N}{2}\psi\right)}{\sin\left(\frac{\psi}{2}\right)} \quad (2.19)$$

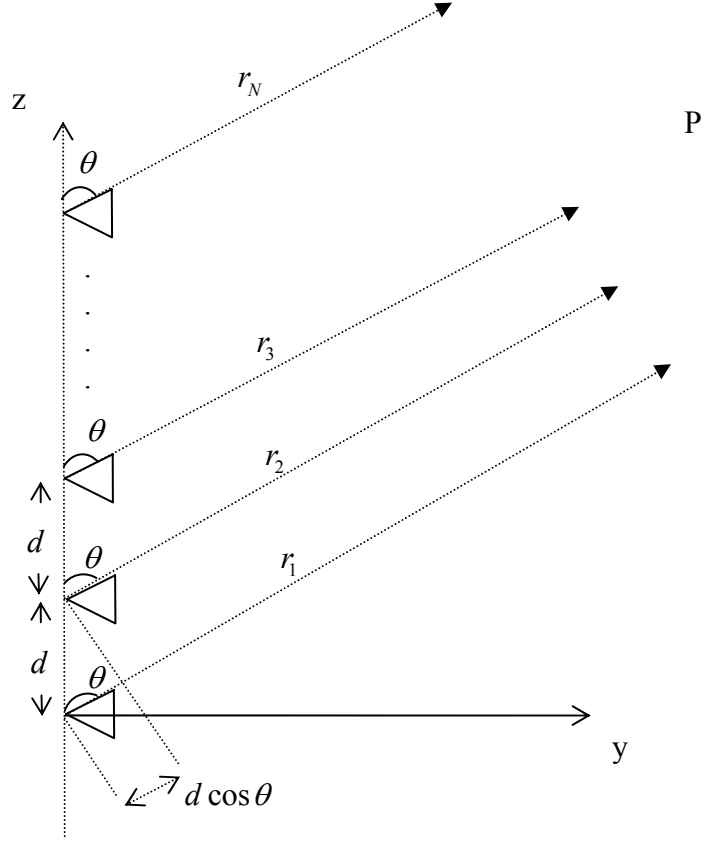
(2.19) ifadesindeki $e^{j\left(\frac{N-1}{2}\right)\psi}$ terimi, eğer orijin dizinin merkezinde olursa 1'e eşit olur.

Bu durumda AF (2.20) eşitliği ile verilebilir.

$$AF = \frac{\sin\left(\frac{N}{2}\psi\right)}{\sin\left(\frac{\psi}{2}\right)} \quad (2.20)$$

AF'nin maksimum değeri N 'dir; dolayısıyla normalize edilmiş AF ifadesi;

$$AF_n = \frac{1}{N} \frac{\sin\left(\frac{N}{2}\psi\right)}{\sin\left(\frac{\psi}{2}\right)} \quad (2.21)$$



Şekil 2.3 ULA yapısı.

2.1.2.1 Dizi faktörünün sıfır ve maksimumları

AF' nin sıfırlarını bulmak için (2.21) ifadesi sıfıra eşitlenerek θ_n değeri hesaplanırsa;

$$\sin\left(\frac{N}{2}\psi\right) = 0 \Rightarrow \frac{N}{2}\psi = \pm n\pi \Rightarrow \frac{N}{2}(kd \cos \theta_n + \beta) = \pm n\pi \quad (2.22)$$

$$\theta_n = \cos^{-1}\left[\frac{\lambda}{2\pi d}\left(-\beta \pm \frac{2n}{N}\right)\right] \quad n = 1, 2, 3, \dots (n \neq N, 2N, 3N \dots) \quad (2.23)$$

$n \neq N, 2N, 3N \dots$ olduğunda sıfırlar mevcut değildir çünkü bu değerler için kosinüs değeri 1'den büyük olur.

AF'nin maksimum olduğu θ_m açı değeri ise aşağıdaki eşitlikten bulunabilir;

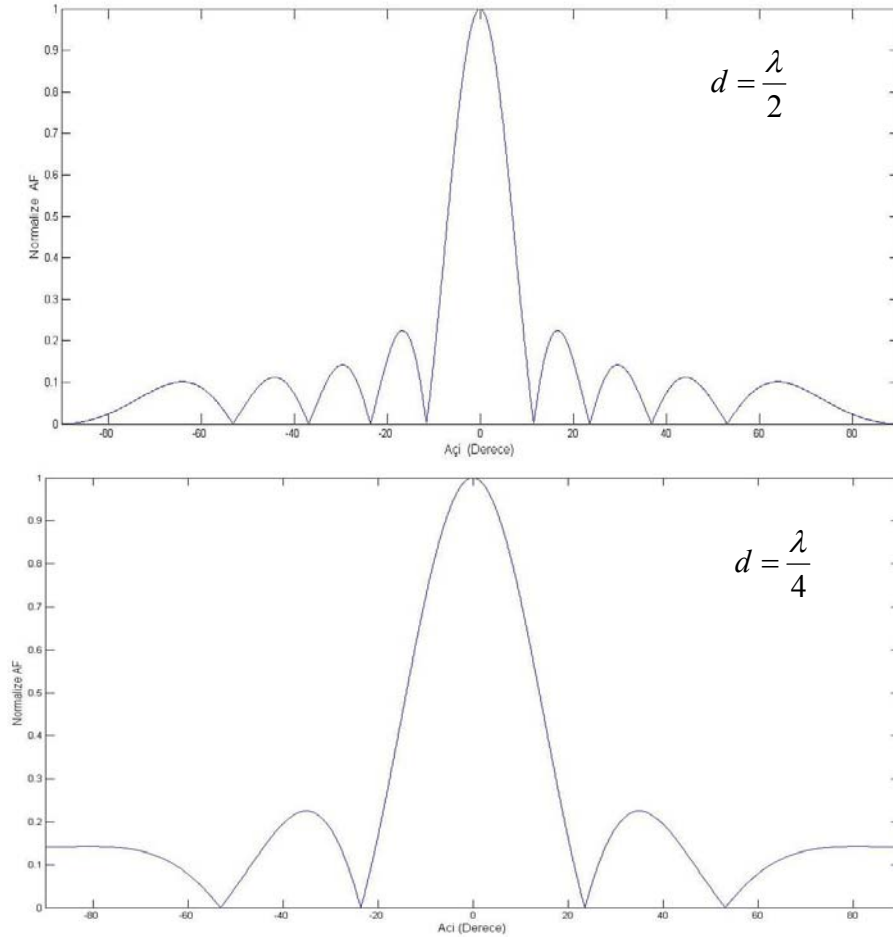
$$\theta_m = \cos^{-1}\left[\frac{\lambda}{2\pi d}(-\beta \pm 2m\pi)\right] \quad (2.24)$$

Eğer $\frac{d}{\lambda}$ değeri çok küçük seçilirse (2.24) ifadesinde sadece $m=0$ için tek bir maksimum değeri olur.

$$\theta_m = \cos^{-1} \left[\frac{\lambda \beta}{2\pi d} \right] \quad (2.25)$$

$m = 1, 2, \dots$ değerleri için (2.24) ifadesinde kosinüslü terimin değeri 1'den büyük olur.

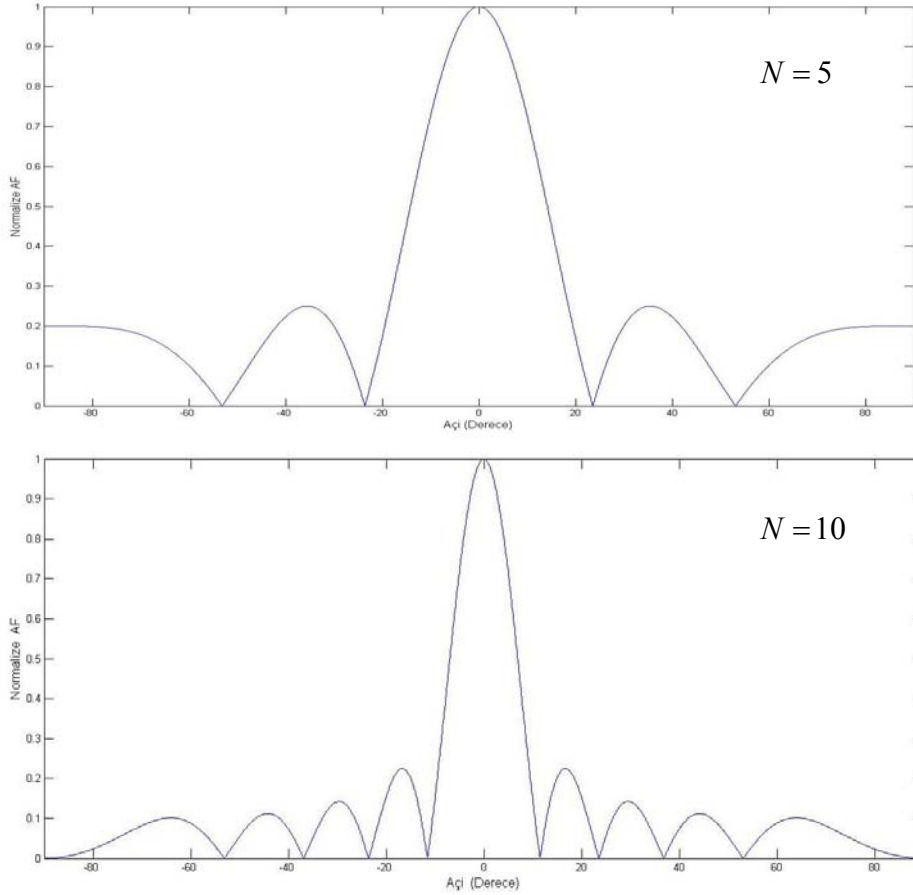
AF'nin dizi anten sayısı N ve dizi elemanları arasındaki λ dalgaboyu cinsinden d mesafesi arasındaki ilişkiyi inceleyecek olursak, N parametresi sabit tutulup d mesafesi değiştirildiğinde AF'nin d parametresiyle ters orantılı olduğu görülür; yani d arttığında ana huzme genişliği azalmakta, azaldığında ise ana huzme genişliği artmaktadır. Doğal olarak interferans işaretlerinin etkisini en aza indirmek için ana huzme genişliğinin mümkün olduğu kadar dar olması istenir. Şekil 2.4'deki çizimlerde anten sayısı $N = 10$ sabit tutulmuş, $d = \lambda/2$ ve $d = \lambda/4$ için AF çizdirilmiştir.



Şekil 2.4 AF' nin d ile değişimi.

AF'nin, anten sayısı N 'ye bağımlılığını görmek için ise anten elemanları arasındaki mesafe $d = \lambda/2$ olarak sabit tutularak $N = 5$ ve $N = 10$ için AF çizdirilmiştir; bu

çizimlerden AF'nin N ile doğru orantılı olduğu, N arttığında ana huzme genişliğinin azaldığı N azaldığında ise ana huzme genişliğinin arttığı görülmektedir. Şekil 2.4 ve Şekil 2.5'den N ve d 'nin bazı değerleri için AF'nin aynı olduğu görülmektedir; N ve d sırasıyla aynı oranda artırılıp aynı oranda azaltılırsa AF değişmez. Bunu görebilmek için $N = 10$ ve $d = \lambda / 4$, $N = 5$ ve $d = \lambda / 2$ için AF'yi çizdirmek yeterlidir.



Şekil 2.5 AF'nin N ile değişimi.

2.1.2.2 Faz taramalı diziler (Phased scanning arrays)

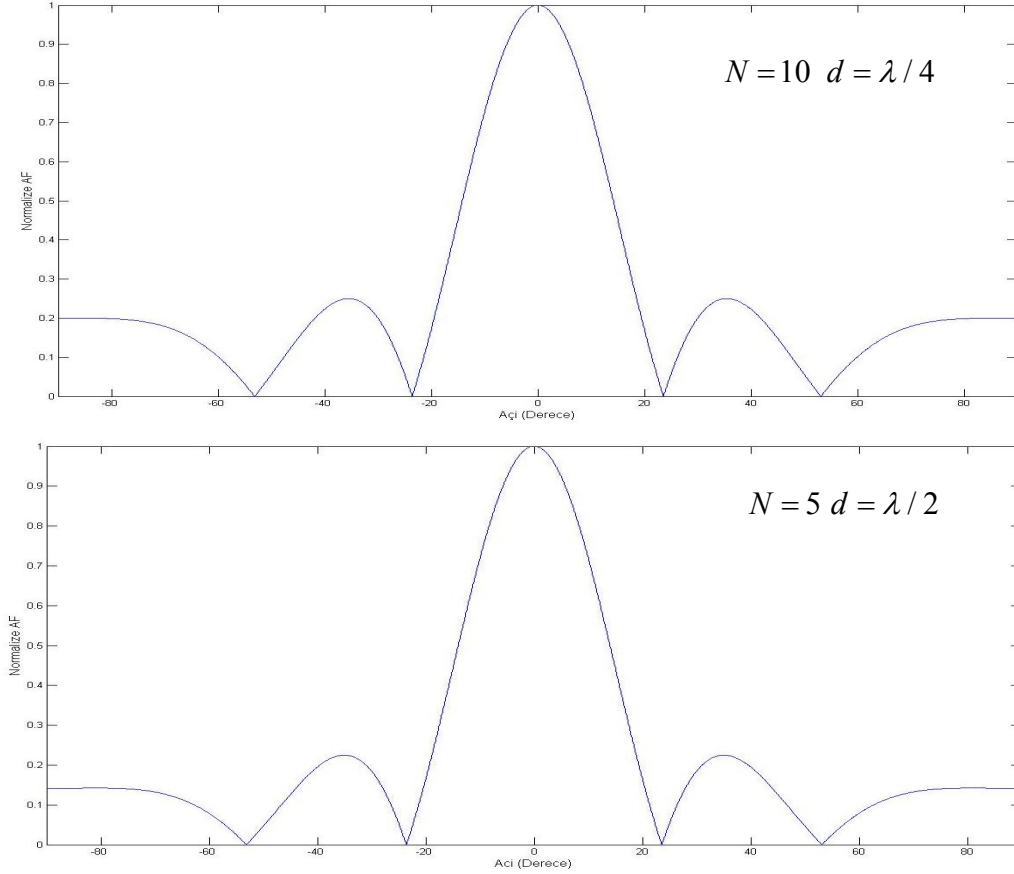
İncelemelerimizdeki matematiksel ifadelerden de görüldüğü gibi dizi ışınma örüntüsünde ana huzmenin yönü dizi elemanları arasındaki β faz farkına bağlıdır. Dolayısıyla dizi elemanları arasındaki β faz farkını sürekli değiştirerek ana huzmeyi sürekli olarak herhangi bir yöne çevirmek mümkündür. Bu şekilde ana huzmenin istenen yönlere doğru yöneltildiği diziler faz taramalı diziler olarak bilinir.

N elemanlı bir dizi için AF ifadesi (2.26) eşitliği ile verilir.

$$AF(\theta) = \sum_{n=0}^{N-1} w_n e^{jknd \cos(\theta)} \quad (2.26)$$

β faz farkı her dizi elemanı ağırlık katsayısında $e^{-jknd \cos(\theta)}$ ifadesiyle ayarlanır. Eğer yönlendirme açısının θ_0 olması isteniyorsa bu durumda β şu şekilde ayarlanır;

$$\psi = kd \cos \theta + \beta = kd \cos \theta_0 + \beta = 0 \Rightarrow \beta = -kd \cos \theta_0 \quad (2.27)$$

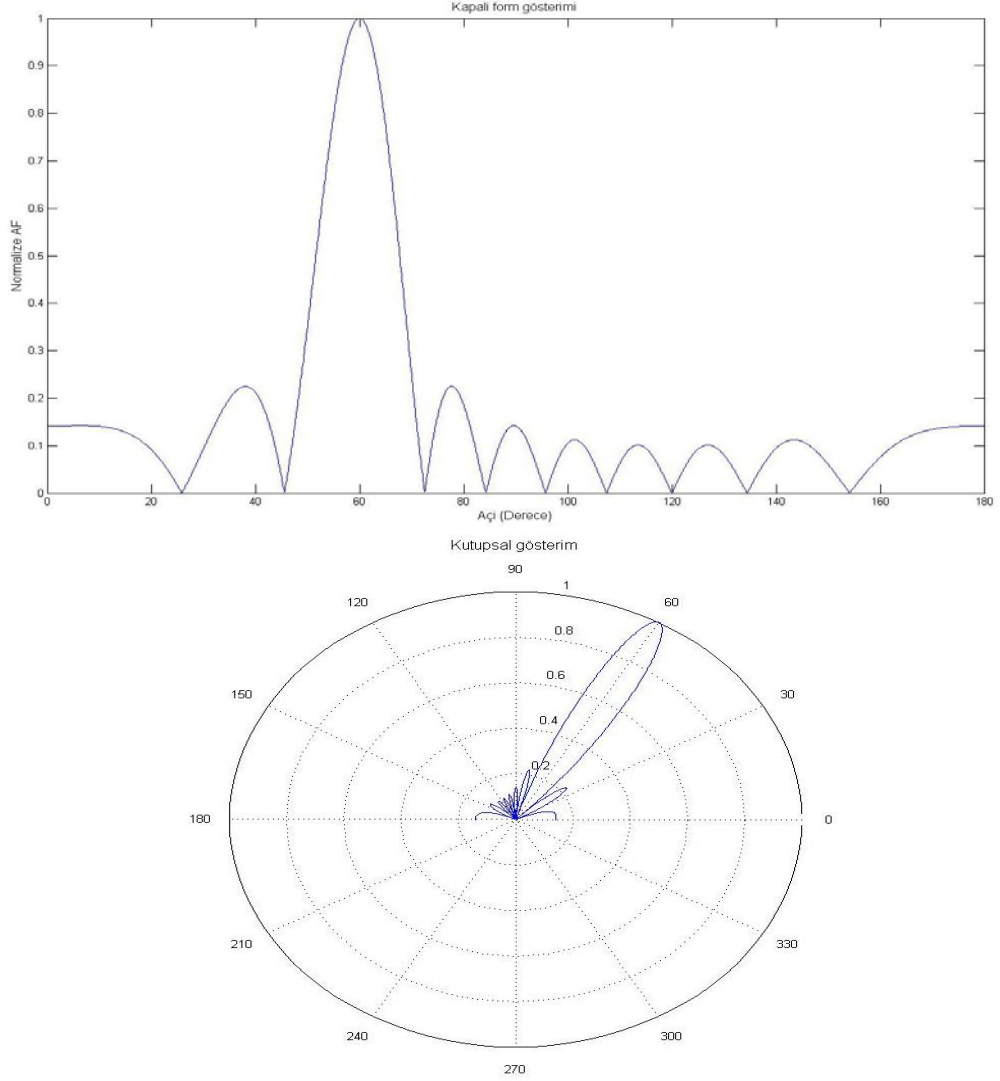


Şekil 2.6 AF'nin N ve d ile değişimi.

Çok küçük ψ değerleri için, yönlendirilmiş bir dizinin AF ifadesi ise aşağıdaki gibidir;

$$AF_n = \frac{\sin\left(N\pi \frac{d}{\lambda} (\cos \theta - \cos \theta_0)\right)}{N\pi \frac{d}{\lambda} (\cos \theta - \cos \theta_0)} \quad (2.28)$$

Şekil 2.7'de anten sayısı $N = 10$ ve antenler arasındaki mesafe $d = \lambda / 2$ olan bir faz taramalı dizinin, istenilen yönlendirme açısı $\theta_0 = 60^\circ$ için AF çizimi görülmektedir.



Şekil 2.7 $\theta_0 = 60^\circ$ için faz taramalı dizi AF biçimi.

2.2 Akıllı (Smart) Antenler

Akıllı anten sistemlerinde ışınma sadece istenen yönde yapılır ve sistem, çevresinde değişen işaret ortamına uyum sağlayabilir. Akıllı anten sistemleri ışınma yapan elemanların belirli biçimde sıralanmasından oluşur; bu dizi elemanlarından alınan işaretler istenen kullanıcıya doğru yöneltilecek ya da anahtarlanacak huzme örüntüsü oluşturmak için birleştirilir. Akıllı anten sistemlerinde akıllı olan antenlerin kendileri değildir; onları akıllı yapan beraber çalıştıkları sayısal işaret işleme birimidir. Alınan işaretleri belirli bir yönde ışınma yapacak şekilde birleştirmek, sayısal huzme biçimlendirme olarak bilinir.

Çoklu yol işareti, gönderilen işaretin iletim ortamındaki nesnelere çarpıp yansımından dolayı alıcıya işaretin bir çok yönden ulaşmasıdır; bu durum alıcıda

sanki etrafta bir çok kaynak varmış etkisi yaratır. Çoklu yol işaretleri farklı yollar izleyerek alıcıya ulaştıkları için farklı fazlara sahiptirler ve alıcı tarafta birleştirilirken faz uyumsuzluğu nedeniyle işaret kalitesinin bozulmasına neden olur. Ortak kanal interferansı ise aynı frekansa sahip iki işaret arasındaki interferanstır. Hareketli hücreli haberleşmede interferans, genellikle farklı hücrelerdeki aynı frekans bandını paylaşan işaretler arasında meydana gelir.

Akıllı antenler, çoklu yol ve ortak kanal interferansını azaltarak kablosuz şebekelerde daha yüksek kapasite sağlaması planlanan teknolojilerdendir. Eski akıllı anten sistemleri, askeri uygulamalarda interferans işaretlerini ya da casus işaretleri bastırma için tasarlanmıştır. Bu sistemlerde interferans bastırma özelliği olduğu için, interferansın şebeke kapasitesini sınırladığı kişisel kablosuz haberleşme sistemlerine uygulanmışlardır. Kişisel haberleşme sistemlerinde, karmaşık hesaplamalar yapmak için zaman oldukça sınırlıdır. Ancak, düşük maliyetli ve güçlü sayısal işaret işleme birimlerinin icadı ve yazılım temelli tekniklerin gelişimiyle, akıllı anten sistemleri hücreli haberleşme için uygulanabilir olmuştur.

2.2.1 Akıllı anten sistem çeşitleri

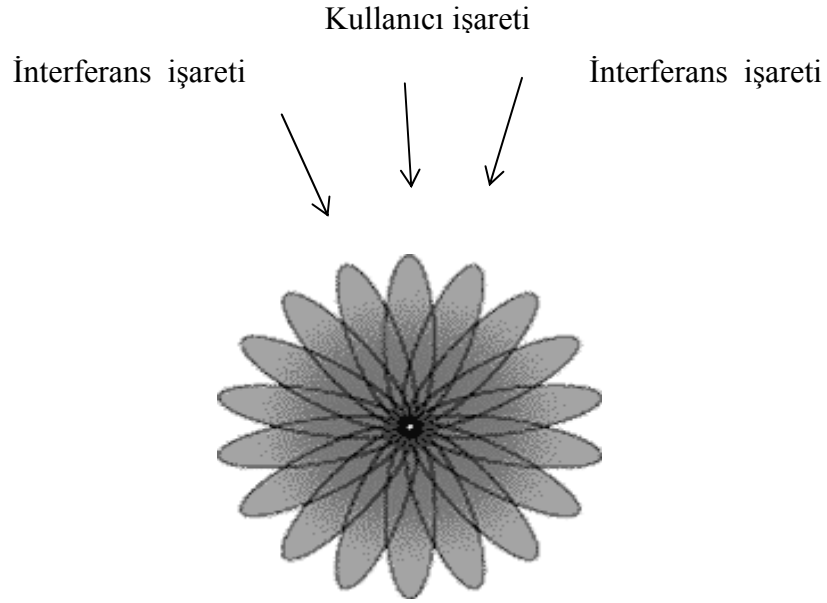
Dinamik olarak anten örüntüsünü değiştirerek interferans ve çoklu yol işaretlerinin etkisini engelleyen, sistem kapsama alanını artıran, temel olarak iki yaklaşım vardır.

- Anahtarlama huzmeleme
- Uyarlamalı diziler

Anahtarlama huzmeleme yaklaşımında, bir anten dizisi birbiriyle örtüşen huzmeler üreterek çevresini kapsamaktadır. Gelen işaret algılandığı zaman baz istasyonu ilgilenilen işaret yönünde en uygun huzmeye sahip antene karar verir ve kullanıcıya hizmet vermek için bu huzmeye anahtarlama yapar. Anahtarlama huzmeleme yaklaşımı, uyarlamalı yaklaşımdan daha basittir. Geleneksel her yöne bakan ya da sektör tabanlı anten sistemleriyle kıyaslandığında şebeke kapasitesinde önemli bir artış sağlamaktadır.

Anahtarlama sistemler gibi, uyarlamalı sistemler de anten dizileri kullanırlar. Uyarlamalı dizi sistemleri daha akıllı sistemlerdir. Bu sistemler ana huzmeyi kullanıcıya doğru yönelterek, hareketli kullanıcıyı sürekli olarak izlerler; aynı zamanda huzme örüntüsünde interferans işaretleri yönünde sıfırlar koyarlar. Her bir

anten elemanı tarafından alınan işaret kompleks bir ağırlıkla çarpılarak faz ve genlik ayarlaması yapılır. Her bir dizi elemanı tarafından alınan bu işaretler, dizinin çıkış işaretini oluşturmak için birleştirilir. Dizi anten elemanı çıkışlarına uygulanan kompleks ağırlıklar, baz istasyonu tarafından ışılanan işareti yöneten, önceden programlanmış sayısal işaret işleme biriminde çalışan, karmaşık uyarlamalı algoritmalar tarafından hesaplanır.



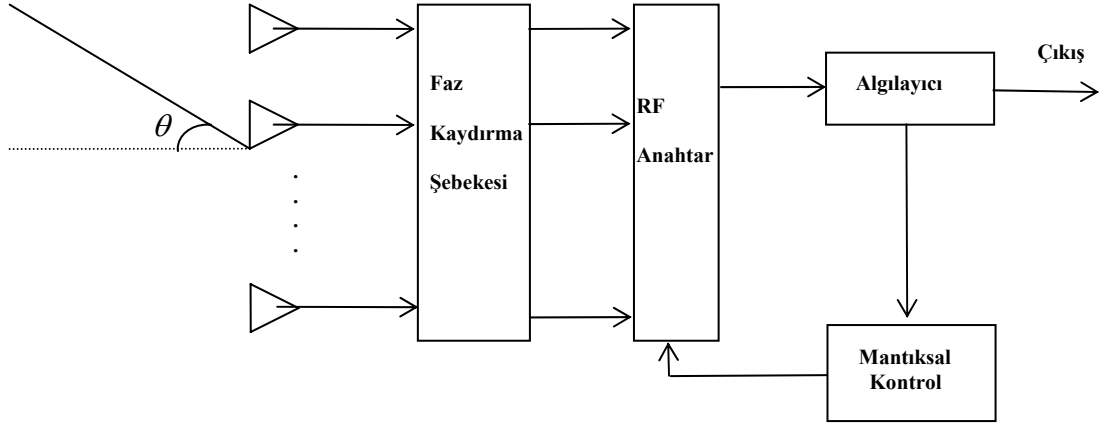
Şekil 2.8 Anahtarlamalı sistemlerde çoklu huzme yapısı [32].

2.2.1.1 Anahtarlamalı huzme sistemleri

Anahtarlamalı huzme sistemlerinde her bir sektör birimi bir çok dar huzmeye bölünür; her bir sektör kendi özel kullanıcılarına ya da kullanıcı gruplarına hizmet verir. Bu uyarlamalı teknik gerçekte istenen işaret yönünde huzmeyi yönlendirme ya da tarama yapmaz. Anahtarlamalı huzme sistemleri, belirli bir alanı kapsayan çoklu birbirleriyle örtüşen huzmeler ışımlayan anten dizisinden oluşurlar.

Anahtarlamalı huzme sistemleri, istenilen hareketli kullanıcı için en iyi alımı yapabilen huzmeyi seçen anahtarlama mekanizmasına sahiptirler; bu seçim işlemi o kullanıcı için alınan maksimum güce göre yapılır. Doğal olarak aynı huzme hem kullanıcıdan baz istasyonuna hem de baz istasyonundan kullanıcıya işaret gönderimi için kullanılır. Uzamsal olarak ayrılmış yönlendirmeli huzmeler interferansı azaltarak ve kapsama alanını artırarak, frekans kanalının yeniden kullanılabilme ihtimalini

artırırlar. Bu antenler, bütün yönlerde eşit kazanca sahip değillerdir fakat geleneksel anten sistemleriyle kıyaslandıklarında belirli bir yön için yüksek kazanca sahiptirler.



Şekil 2.9 Anahtarlama sistemi birimleri.

Baz istasyonları için tipik bir anahtarlama sistemi her biri belirli bir sektörü kapsayan çoklu anten dizilerinden oluşur. Şekil 2.9'daki sistem belirli yönlere bakan çoklu huzme oluşturan faz kaydırma şebekesi içerir. RF anahtar, istenen yöndeki doğru huzmeyi aktif hale getirir. Doğru huzmenin seçimi ise mantıksal kontrol birimi tarafından gerçekleştirilir. Mantıksal kontrol birimi üzerinde çalışan algoritma, bütün huzmeleri tarayarak algılayıcı tarafından alınan en güçlü işaretin geldiği huzmeyi seçer.

Anahtarlama sistemleri tümüyle uyarlamalı olan sistemlere göre daha az karmaşıktır ve önemli bir alan genişlemesi sağlayarak kapasiteyi artırır; istenen kullanıcı huzme merkezinde ise, interferans etkisini engelleme başarısı yüksektir. Aynı zamanda bu sistemler daha ucuzdur, eski sistemlere göre daha kolay inşa edilebilirler. Anahtarlama sistemlerinde sabit huzmeleri oluşturmak için değişik yaklaşımlar mevcuttur [33-36].

2.2.1.2 Uyarlamalı huzme sistemleri

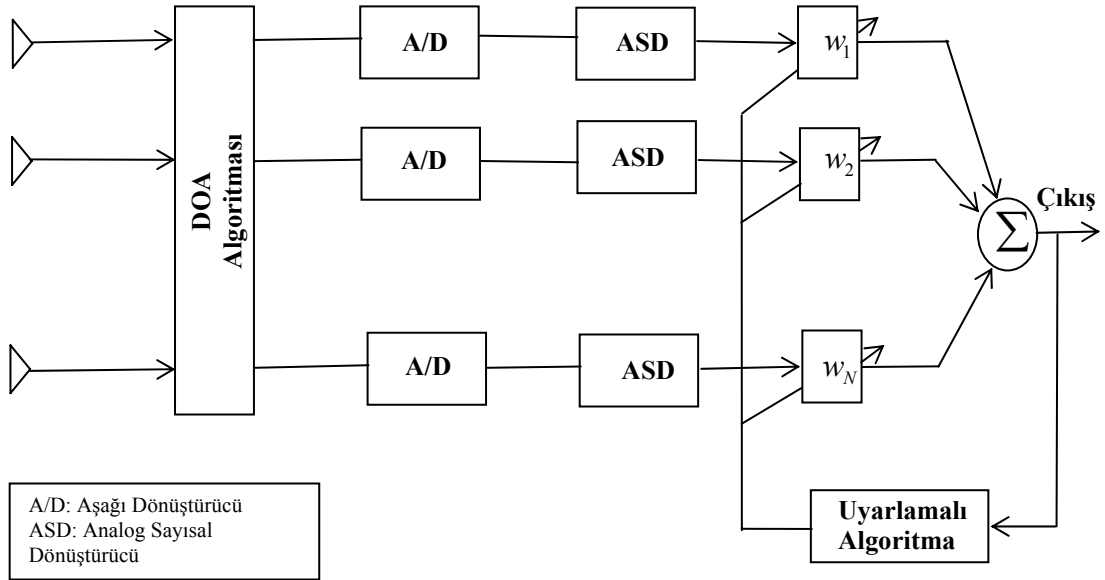
Anahtarlama sistemlerinin tersine uyarlamalı sistemler oldukça akıllıdır, değişen RF çevresine uyum sağlayabilirler. Uyarlamalı diziler, anahtarlama sistemleri gibi anten dizilerini kullanırlar fakat bu dizi, bir işaret işleme birimi tarafından kontrol edilir. İşaret işleme birimi ışına huzmesini istenilen hareketli kullanıcıya doğru

yönlendirir, kullanıcı hareket ettikçe onu takip eder ve interferans işaretleri yönünde sıfırlar koyarak diğer kullanıcılardan kaynaklanan interferansı minimize eder.

Uyarlamalı anten dizileri gerçekten akıllıdır ve akıllı anten sistemleri olarak da isimlendirilirler, bu sistemleri akıllı yapan akıllı sayısal işaret işleyicidir. Bu işaret işleme, işlem yoğunluğu fazla olan karmaşık algoritmalar tarafından yapılır.

Akıllı anten sistemlerinde DOA algoritması tarafından interferans işaretleri ve çoklu yol işaretleri de dahil olmak üzere tüm işaretlerin geliş açısı belirlenir. Daha sonra kullanıcı işareti belirlenir ve diğer istenmeyen işaretlerden ayrılır. Son olarak istenen kullanıcı yönüne huzme yöneltilir ve kullanıcı hareket ettiği sürece izlenir; sürekli olarak güncellenen karmaşık ağırlıklarla, interferans işaretleri yönüne sıfırlar yerleştirilir.

Bir anten dizisindeki ana huzmenin ışın yönü, dizi elemanları arasındaki β faz farkına bağlıdır. Dolayısıyla dizi elemanları arasındaki β faz farkını ayarlayarak ana huzmeyi sürekli olarak herhangi bir yöne yönlendirmek mümkündür. Uyarlamalı sistemler de aynı fikri kullanırlar ve istenen yönde maksimum ışın elde etmek için dizi elemanları arasındaki faz farkını ayarlarlar.



Şekil 2.10 Uyarlamalı sistem yapısı.

Şekil 2.10'daki huzme biçimlendirme şebekesinde, dizi elemanları tarafından alınan işaretler istenen çıkış huzme biçimini oluşturmak için akıllı bir şekilde birleştirilirler.

Alınan işaretler ağırlıklandırılmadan önce temel band işaretine dönüştürülürler. Her bir elemanın çıkışındaki alıcı (receiver), bu frekans aşağı indirme işlemini gerçekleştirir. Uyarlamalı anten dizileri gelen işaretleri ağırlıklandırmak için sayısal işaret işleyici (DSP: Digital Signal Processor) kullanırlar. Dolayısıyla gelen işaret DSP tarafından kullanılmadan önce sayısal forma dönüştürülmelidir. ASD (Analog Sayısal Dönüştürücü) birimleri bu amaçla kullanılır. DSP bu sistemin en önemli birimidir; sayısal biçime dönüştürülmüş işareti kabul ederek üzerindeki yazılım sayesinde sayısal veriyi işler. DSP, gelen veri bilgisini yorumlar her bir çıkış için karmaşık ağırlıkları hesaplar ve dizi örüntüsünü optimize etmek için her bir eleman çıkışını bu ağırlıklarla çarpar. Optimizasyon özel bir kritere göre yapılır, istenen yönde maksimum huzme kazancı sağlarken interferans ve gürültü etkisini minimize eder. Değişik kriterleri temel alarak optimum ağırlıkları hesaplayan ve güncelleyen bir çok algoritma vardır [11-19].

2.2.1.3 Akıllı anten sistem çeşitlerinin karşılaştırılması

Anahtarlama huzme sistemleri;

- Dar huzme genişliğine sahip bir çok sabit, yönlendirilmiş huzmeler kullanır.
- Basit, sabit faz kaydırmalı bir şebekeyle faz kaydırması yapılır.
- Karmaşık algoritmalara ihtiyaç duymaz, huzme seçimi için basit algoritmalar kullanılır.
- Gelişmiş bir teknoloji kullanmadığı için karmaşık değildir ve düşük maliyetlidir.
- Mevcut hücrel sistemle entegrasyonu kolay ve ucuzdur.
- Geleneksel anten temelli sistemlere göre kapasite ve kapsamada önemli derecede artış sağlar.
- Bir çok dar huzme kullanıldığı için, hücrel kullanıcı bir huzme kapsamından diğer huzmeye geçtiğinde sık sık hücre içi aktarma (hand offs) yapılmalıdır.
- İstenen işaret, çoklu yol işareti ya da interferans işareti ayırımı yapamaz, bu nedenle yanlış bir şekilde istenen işaret yerine interferans işaretlerinde iyileştirme yapabilir.

- Huzme örüntüsünde istenmeyen işaretler yönünde sıfır koyulmadığı için anahtarlama sistemler uyarlamalı dizi sistemlerine göre daha sınırlı bir ortak kanal interferans bastırması sağlayabilirler.

Uyarlamalı dizi sistemleri;

- Tamamıyla uyarlamalı bir sistem, huzmeyi istenilen işarete yönlendirir ve interferans işaretleri yönünde sıfırlar yerleştirir.
- DSP teknolojisine ihtiyaç duyar.
- Huzmeyi istenen yöne yönlendirebilmek için karmaşık, uyarlamalı algoritmalara ihtiyaç duyar.
- Anahtarlama sistemlere göre daha iyi interferans engelleme yeteneğine sahiptir.
- Mevcut sistemlerle entegrasyonu kolay değildir ve pahalıdır.
- Huzme, kullanıcıyı sürekli olarak takip ettiği için hücre içi aktarmalar daha azdır.
- İleri interferans engelleme yeteneği sayesinde anahtarlama sistemlere göre daha fazla kapasite ve kapsama sunar.
- Çoklu yol bileşenlerini ayırıp engelleyebilir ya da işaret kalitesini artırmak için gecikmeleri düzelterek işarete ekleyebilir.

2.2.1.4 Akıllı anten sisteminin yararları

Akıllı antenler, dar huzme biçimli ışılanan enerjiyi istenilen hareketli kullanıcı yönünde odaklamak için uzamsal filtreleme özelliğine sahiptir. Buna ek olarak aynı zamanda huzme örüntüsünde, çevresindeki diğer kullanıcılar yönünde sıfırlar koyarlar. Bu nedenle bu sistemlerde ortak kanal interferansı yok denecek kadar azdır.

Akıllı antenler, anten dizisinden oluştukları için aynı gücü kullanan geleneksel antenlere göre daha yüksek kazançlı dar huzmeye sahiptirler. Kazancın artması sınırın artmasını, dolayısıyla kapsama alanının artmasını sağlar ve belirli bir bölgeyi kapsamak için daha az sayıda baz istasyonu gerekir.

Akıllı antenler, ortak kanal interferansını azaltarak frekans tekrar kullanılma oranını artırır. Bu sayede akıllı antenler, aynı frekans spektrumunun daha fazla kullanıcı tarafından kullanılmasını sağlarlar ve kapasiteyi oldukça artırır.

Sıradan antenler enerjiyi tüm yönlerde yayarak güç israfına neden olurlar. Akıllı antenler ise sadece belirli yönde ışınım yaparlar. Bu nedenle baz istasyonu, ışınım için daha az güce ihtiyaç duyar. Gönderilen gücün azalması aynı zamanda diğer kullanıcılara olan interferansın azalması anlamına gelir.

Yoğun hücreli şebekelerde kapasiteyi artırmak için, aşırı yoğun hücreler frekans yeniden kullanım faktörünü artırmak için mikro hücrelere bölünürler. Bu durum hücre boyutları daha küçük olduğu için hücre içi aktarım sayısını artırır. Baz istasyonlarında akıllı antenlerin kullanılması durumunda ise hücreleri bölmeye gerek yoktur çünkü bağımsız, belirli bir istenilen yöne bakan huzmeler sayesinde kapasite artırılır. Sadece aynı frekans kullanan iki huzme arasında etkileşim olabilir ve hücre içi aktarma nadiren olur.

Akıllı antenler çoklu yol bileşenlerini interferans olarak engelleyebilirler. Dolayısıyla çoklu yol etkisini zayıflatırlar ya da sistem performansını iyileştirmek için işarete ekleyerek, işaret kalitesini artırabilirler.

Akıllı anten teknolojisi TDMA, FDMA, CDMA gibi bir çok çoklu erişim teknolojisine uygulanabilir. Tüm frekans bantlarında tüm modülasyon yöntemleriyle birlikte çalışabilir.

2.3 Uyarlamalı Huzme Biçimlendirme

Uyarlamalı huzme biçimlendirme şu bilgiyi kullanır; farklı göndericilerden yayılan işaretler aynı frekans kanalını kullansalar bile farklı yön bilgisini içermektedirler. Huzme biçimlendirme daha çok, enerji ışınımıyla ilgili görünse de, hem ışınım hem de işaret alımı için uygulanabilir. Uzamsal ayırım, istenilen işareti interferans işaretlerinden ayırmak için kullanılır. Uzamsal bir filtre kullanımı, uzamsal bir yüzeyden toplanan verinin işlenmesine ihtiyaç duyar. Uzamsal örnekleme ayrık zamanlı olduğundan, uzamsal filtrelemeyi gerçekleyen işlemci, huzme biçimlendirici adını alır. Huzme biçimlendirici, her bir algılayıcının, uzamsal olarak örneklemediği veriyi tek bir çıkış elde etmek için doğrusal olarak birleştirir.

Uyarlamalı anten dizileri olarak da isimlendirilen uyarlamalı huzme biçimlendiriciler, belirli bir aboneyi dinleyebilir ve daha etkin bir şekilde bu aboneye enerji gönderebilirler. Uyarlamalı olarak güncellenen her bir anten çıkışına bağlı kompleks ağırlıklar sayesinde, interferans işaretleri yönünde işaret alımını en aza indirgeyebilirler. Bu sayede alınan işaret kalitesini artırıp ortak kanal interferansını engellerler. Bu sistemler, çıkış işaretinde SNR (Signal to Noise Ratio: İşaret gücünün gürültü gücüne oranı) oranını artırmak için alıcı ya da ışınma örüntüsünü, değişen kanal gürültüsü ve interferansa dinamik olarak ayarlayan bir dizi antenden oluşurlar.

Uyarlamalı huzme biçimlendirme, anten dizisinin kullanıldığı bir teknolojidir; gelen işaretin kestirimi yapılarak istenen yön için maksimum alım sağlanır, diğer yönlerden gelen aynı frekanslı kullanan işaretler ise engellenir. Bu, dizideki her bir alıcı çıkışına uygulanan ağırlıkların değiştirilmesiyle başarılır. Uyarlamalı huzme biçimlendirmede optimum ağırlıklar, farklı kriterlere göre çalışan karmaşık algoritmalar tarafından iteratif olarak hesaplanır.

Huzme biçimlendirme genellikle dizideki her bir antenin beslemesi aynı faza getirilerek yapılır; bu sayede bütün antenlerden gönderilen ya da alınan işaretler belirli bir yön için aynı fazda olacaktır. Faz ve genlikler alınan işareti optimize etmek için ayarlanır. Uyarlamalı huzme biçimlendirici kullanımında temel amaç, her bir anten çıkışına uygulanan ağırlıkları dinamik olarak güncelleyerek ortak kanal interferans işaretleri ve gürültü varlığında, dizi çıkış huzmesini istenen işaret yönünde optimum alım yapacak şekilde biçimlendirmektir.

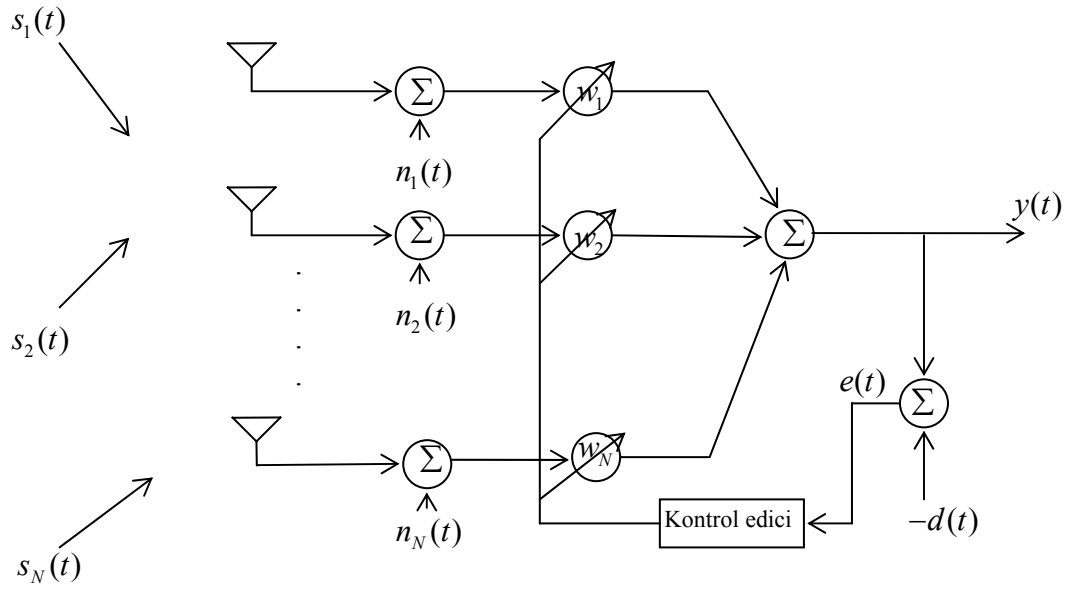
2.3.1 Uyarlamalı huzme biçimlendirme problem kurulumu

Şekil 2.11’de uyarlamalı huzme biçimlendirme yapısı görülmektedir. Dizi çıkışı $y(t)$, her bir dizi elemanı tarafından alınan kaynak işareti $s_i(t)$ ve gürültü işareti $n(t)$ ’nin ağırlıklandırılmış toplamına eşittir. w_m ağırlıkları dizi çıkışı $y(t)$, istenilen işaretle oldukça korelasyonlu referans işareti $d(t)$ ve daha önceki ağırlıklar esas alınarak iteratif olarak hesaplanır. Referans işareti alıcı tarafta bilinen bir alıştırma dizisi ya da yayılım kodu (spreading code) kullanılarak istenilen işarete yaklaştırılır. Referans işaretleri uyarlamalı huzme biçimlendirme sistemlerinin kullanıldığı yapılara göre değişiklik gösterir. Referans işareti genellikle istenilen işaretle yüksek derecede ilişkilidir; bu ilişkinin mertebesi algoritmanın doğruluğunu ve yakınsamasını etkiler.

Dizi çıkış ifadesi (2.29) ifadesi ile verilir;

$$y(t) = \mathbf{w}^H \mathbf{x}(t) \quad (2.29)$$

w^H ifadesi w ağırlık vektörünün kompleks eşlenik transpozisini belirtir. Optimum ağırlıkları hesaplayabilmek için, dizi cevap vektörü, örneklenmiş dizi çıkışı verisi bilinmelidir. Dizi cevap vektörü frekansın olduğu kadar, geliş açısının da bir fonksiyonudur. N . antende alınan temel band işareti orijinal $s_i(t)$ işaretinin faz kaymış ve zayıflamış biçimlerinin toplamıdır.



Şekil 2.11 Uyarlamalı huzme biçimlendirme şebekesi.

$$x_N(t) \cong \sum_{i=1}^N a_N(\theta_i) s_i(t) e^{-j2\pi f_c \tau_N(\theta_i)} \quad (2.30)$$

$s_i(t)$ ifadesi hem istenen işareti hem de interferans işaretlerini belirtir, f_c taşıyıcı frekansı, $\tau_N(\theta_i)$ ise gecikmedir.

$$\mathbf{a}(\theta_i) = [a_1(\theta_i) e^{-j2\pi f_c \tau_1(\theta_i)}, a_2(\theta_i) e^{-j2\pi f_c \tau_2(\theta_i)}, \dots, a_N(\theta_i) s_i(t) e^{-j2\pi f_c \tau_N(\theta_i)}]^T \quad (2.31)$$

$\mathbf{a}(\theta)$ ifadesi belirli bir θ değeri için yönlendirme vektörünü belirtir.

$$\mathbf{A}(\theta) = [\mathbf{a}(\theta_1) \mathbf{a}(\theta_2) \dots \mathbf{a}(\theta_d)] \quad (2.32)$$

$$\mathbf{S}(t) = [s_1(t) s_2(t) \dots s_d(t)]^T \quad (2.33)$$

$$\mathbf{x}(t) = \mathbf{A}(\theta)\mathbf{S}(t) \quad (2.34)$$

(2.34) ifadesine gürültü de eklenirse;

$$\mathbf{x}(t) = \mathbf{A}(\theta)\mathbf{S}(t) + \mathbf{n}(t) \quad (2.35)$$

(2.35) ifadesi $s_0(t)$ istenen işaret interferans işaretlerinden ayrılıp yeniden düzenlenirse;

$$\mathbf{x}(t) = s_0(t)\mathbf{a}(\theta_0) + \sum_{i=1}^{N_u} u_i(t)\mathbf{a}(\theta_i) + \mathbf{n}(t) \quad (2.36)$$

$\mathbf{a}(\theta_i)$, i . interferans işareti için, $\mathbf{a}(\theta_0)$ ise istenen işaret için yönlendirme vektörüdür. $u_i(t)$, N_u tane θ_i açılardan gelen istenmeyen interferans işaretlerini belirtir. Bu durumda, geliş açılarının önceden DOA algoritması tarafından bilindiği varsayılmaktadır. Bu bilgilerin bilinmesi durumunda uyarlamalı algoritmalar istenen işaret $s_0(t)$ ile kestirilen işaret $\hat{s}_0(t)$ arasındaki hatayı minimize ederek $x(t)$ 'den $s_0(t)$ 'yi elde edebilirler.

$d(t)$ işareti istenen $s_0(t)$ işareti ile çok yakın ilişkili referans işaretini belirtsin. Huzme biçimlendirici çıkışı ve referans işareti arasındaki MSE (Mean Square Error), $E\{e^2(t)\}$, şu şekilde hesaplanır;

$$E\{e^2(t)\} = E\left\{\left[d(t) - \mathbf{w}^H \mathbf{x}(t)\right]^2\right\} \quad (2.37)$$

$$E\{e^2(t)\} = E\{d^2(t)\} - 2\mathbf{w}^H \mathbf{r} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (2.38)$$

$\mathbf{r} = E\{\mathbf{x}(t)d^*(t)\}$ ifadesi istenen işaret ile alınan işaret arasındaki çapraz ilişki vektörüdür; $\mathbf{R} = E\{\mathbf{x}(t)\mathbf{x}^H(t)\}$ ifadesi ise alınan işaretin özilişki matrisidir. Minimum MSE (2.38) ifadesinin \mathbf{w} 'ye göre gradyan vektörünün sıfıra eşitlenmesiyle bulunur.

$$\nabla_{\mathbf{w}}(E\{e^2(t)\}) = -2\mathbf{r} + 2\mathbf{R}\mathbf{w} = 0 \quad (2.39)$$

Ağırlık vektörü için optimum çözümü veren ifade ise;

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{r} \quad (2.40)$$

(2.40) ifadesi optimum Wiener çözüm ifadesidir.

2.3.2 Uyarlamalı huzme biçimlendirme yöntemleri

Bu bölümde bir çok huzme biçimlendirme yaklaşımı ve uyarlamalı algoritmalar ayrıntılı olarak ele alınmaktadır. Ama öncesinde sık kullanılan bazı geliş yönü bulma algoritmaları anlatılacaktır.

2.3.2.1 Geliş yönü bulma algoritmaları

Huzme biçimlendirmede, ışımayı istenilen yöne yönlendirmek ve interferans işaretleri yönünde sıfırlar koyabilmek için işaretlerin geliş açıları önceden bilinmelidir. DOA algoritmaları, alınan dizi çıkış işareti üzerinde çalışırlar ve işaretlerin geliş açılarını hesaplarlar. Geliş açıları bilindikten sonra bu bilgi huzme biçimlendirme şebekesine, huzme yönlendirme için gerekli olan kompleks ağırlıkların hesaplanması için dağıtılır. Bu bölümde ilk önce ortamdaki kaynak sayısını bulmaya yönelik kriterler ele alınacak, daha sonra da sık kullanılan bazı DOA algoritmaları anlatılacaktır.

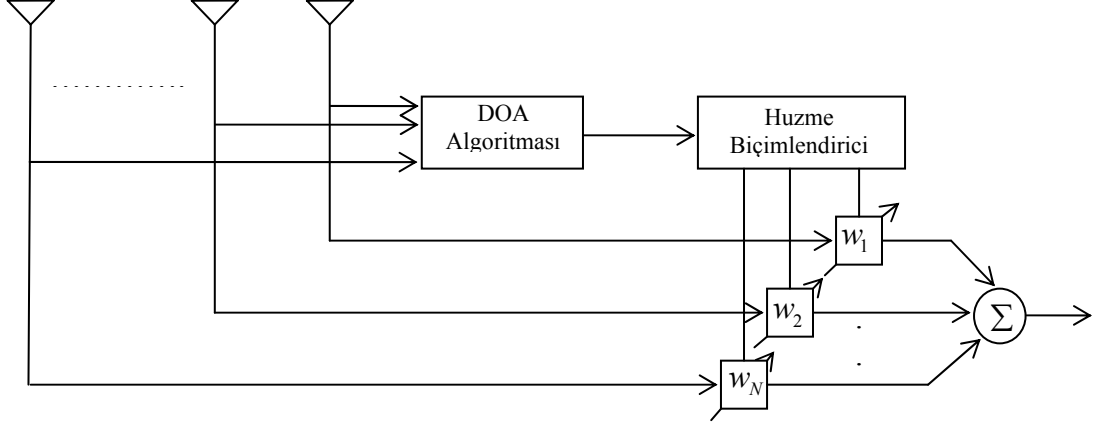
Kaynak sayısının kestirilmesi

Çoğu yüksek çözünürlüklü yön bulma metodları, yönlü kaynak sayısına ihtiyaç duyar ve bu metodların başarımı bu kaynak sayısının kesinlikle bilinmesine bağlıdır. Dizi korelasyon matrisinin özdeğerlerini kısımlara ayıran tüm yöntemlerin kullandığı temel fikir şudur: M tane yönlü kaynağa karşı gelen M tane özdeğer, geri kalan ortam gürültüsüne karşı gelen $L - M$ tane özdeğerden daha büyüktür.

Basit bir algılama, R_u kovaryans matrisinin en küçük özdeğerinin, λ_{\min} , K çarpma faktörünün ve buna uyan gürültü alt uzayının kestirimiyle yapılabilir. Bu K tane özdeğer gürültü varyansı σ_n^2 'ye eşit olmalıdır. Bu durumda gelen işaret sayısı kestirimi $M = L - K$ olur. Ancak R_u , sonlu bir veri örneğinden kestirildiği için gürültü gücüne karşı gelen tüm özdeğerler özdeş değillerdir. AIC (Akaike Information Criteria) ve MDL (Minimum Descriptive Length) kriterleriyle işaret kaynak sayısı şu şekilde belirlenir;

N adet bağımsız ve özdeş dağılımlı örnekten dizi korelasyon matrisi kestirilir. Korelasyon matrisinin $\lambda_i, i=1,2,\dots,L$; L tane özdeğeri $\lambda_1 > \lambda_2 > \dots > \lambda_L$ olacak şekilde hesaplanır. M kaynak sayısı aşağıdaki ifadeyi minimum yapan M değeri olarak hesaplanır:

$$N(L-M) \log \left(\frac{f_1(M)}{f_2(M)} \right) + f_3(M, N) \quad (2.41)$$



Şekil 2.12 DOA algoritmasının huzme biçimlendirme şebekesinde kullanımı.

$f_1(M)$ ve $f_2(M)$ ifadeleri ise;

$$f_1(M) = \frac{1}{L-M} \sum_{i=M+1}^L \lambda_i \quad (2.42)$$

$$f_2(M) = \left(\prod_{i=M+1}^L \lambda_i \right)^{\frac{1}{L-M}} \quad (2.43)$$

$f_3(M, N)$ ise maliyet fonksiyonudur;

$$f_3(M, N) = \begin{cases} M(2L-M) & \text{AIC için} \\ \frac{1}{2} M(2L-M) \log N & \text{MDL için} \end{cases} \quad (2.44)$$

L , dizideki anten sayısını belirtir.

Kaynak sayılarının hesaplanmasında alternatif bir yöntem Lee tarafından tartışılmıştır [37]; bu yöntem dizi korelasyon matrisi özdeğerlerini kullanan diğer yöntemlerin tersine dizi korelasyon matrisi özvektörlerini kullanır. Özvektör algılaması yapan bu teknik, yönleri bilinen kaynaklara uygulanabilir ve AIC ve MDL'den daha düşük SNR değerlerinde kaynak sayılarını kestirme yeteneğine sahiptir. Kaynak sayılarının belirlenmesi aynı zamanda dizi elemanları arasındaki mesafeye, örnekleme sayısına ve dizi geometrisine de bağlıdır [36].

MUSIC (Multiple Signal Classification) yöntemi

N antenden oluşan bir diziye M tane işaretin geldiği durumu göz önüne alalım. Bu durumda dizi çıkış ifadesi;

$$\mathbf{x}(t) = \mathbf{A}(\theta)\mathbf{S}(t) + \mathbf{n}(t) \quad (2.45)$$

Ya da matris gösterimiyle;

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \mathbf{n} \quad (2.46)$$

S , işaret vektörü, A yönlendirme vektörü, n ise sıfır ortalamalı ve σ_n^2 varyanslı gürültü vektörüdür.

$N \times N$ boyutlu kovaryans matrisi ifadesi ;

$$\mathbf{R}_x = E\{\mathbf{X}\mathbf{X}^H\} = \mathbf{A}E\{\mathbf{S}\mathbf{S}^H\}\mathbf{A}^H + E\{\mathbf{n}\mathbf{n}^H\} = \mathbf{A}\mathbf{P}\mathbf{A}^H + \sigma_n^2\mathbf{I} \quad (2.47)$$

$$\mathbf{P} = E\{\mathbf{S}\mathbf{S}^H\} = \text{diag}(p_1, p_2, \dots, p_M) \quad (2.48)$$

$$p_k = s_k^2, \quad k = 1, 2, \dots, M \quad (2.49)$$

(2.49) ifadesinde p_k anten dizisine değişik yönlerden gelen işaretlerin güçlerini belirtir. $\mathbf{a}_k = [a_1(\theta_k) \ a_2(\theta_k) \ \dots \ a_N(\theta_k)]$ olmak üzere;

$$\mathbf{A}\mathbf{P}\mathbf{A}^H = \sum_{k=1}^M p_k \mathbf{a}_k \mathbf{a}_k^H \quad (2.50)$$

(2.50) ifadesi $\mathbf{A}\mathbf{P}\mathbf{A}^H$ matrisinin özvektörleri kullanılarak (2.51) ile verilebilir;

$$\mathbf{A}\mathbf{P}\mathbf{A}^H = \sum_{k=1}^N \lambda_k \mathbf{v}_k \mathbf{v}_k^H = \sum_{k=1}^M \lambda_k \mathbf{v}_k \mathbf{v}_k^H \quad (2.51)$$

(2.51) ifadesinde N boyutlu uzay M işaret sayısına düşürülmüştür çünkü $\mathbf{A}\mathbf{P}\mathbf{A}^H$ matrisinin λ_k , $k = M + 1, M + 2, \dots, N$, özdeğerleri sıfıra eşit olacaktır.

$$\lambda_k = 0, \quad k = M + 1, M + 2, \dots, N \quad (2.52)$$

R_x matrisinin özdeğerleri ise $\mathbf{A}\mathbf{P}\mathbf{A}^H$ matrisinin özdeğerlerine gürültü işaret gücü (varyansı) eklenmiş hali olacaktır.

$$\lambda_{xk} = \lambda_k + \sigma_n^2 \quad k = 1, 2, \dots, N \quad (2.53)$$

Gürültü ve işaret özvektörleri birbirine dik olduklarından işaret ve gürültü alt uzayı da birbirine diktir;

$$\mathbf{a}^H(\theta_k)\mathbf{v}_i = 0 \quad k = 1, 2, \dots, M \quad i = M + 1, M + 2, \dots, N \quad (2.54)$$

$$\mathbf{v}_i = \sigma_n^2 \quad i = M + 1, M + 2, \dots, N \quad (2.55)$$

(2.54) ifadesinde \mathbf{v}_i, R_x matrisinin özvektörlerini belirtmektedir, $i = M + 1, M + 2, \dots, N$ için bu özvektörler gürültü varyansına eşit olacaktır. Bu ifade tüm θ değerleri üzerinden yazılıp güç spektrumu elde edilirse M tane işaret için geliş yönlerinde keskin tepcikler elde edilir.

$$\sum_{i=M+1}^N \mathbf{a}^H(\theta)\mathbf{v}_i = 0 \quad (2.56)$$

$$P(\theta) = \sum_{i=M+1}^N \left| \mathbf{a}^H(\theta)\mathbf{v}_i \right|^2 \quad (2.57)$$

$$P_{MUSIC} = \frac{1}{P(\theta)} = \frac{1}{\sum_{i=M+1}^N \left| \mathbf{a}^H(\theta)\mathbf{v}_i \right|^2} = \frac{1}{\mathbf{a}^H(\theta)\mathbf{v}\mathbf{v}^H\mathbf{a}(\theta)} \quad (2.58)$$

Gelen işaretler birbiriyle az ilişkili ya da ilişkisiz ise P işaret kovaryans matrisi tekil değildir, aksi halde P kötü koşulludur ve MUSIC algoritmanın performansını olumsuz yönde etkiler [38].

ESPRIT (Estimation of Signal Parameters via Rotational Invariance Technique) yöntemi

ESPRIT, DOA kestirimi için en etkili yöntemlerden birisidir. İki ayrı dizi kullanır ve her bir dizideki elemanlar ilk elemanla aynı yönde eşit mesafe uzaklıktadırlar.

İki ayrı dizi tarafından alınan işaretler sırasıyla $\mathbf{x}(t)$ ve $\mathbf{y}(t)$ olsun:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}_x(t) \quad (2.59)$$

$$\mathbf{y}(t) = \mathbf{A}\phi\mathbf{s}(t) + \mathbf{n}_y(t) \quad (2.60)$$

A , $K \times M$ boyutlu bir matristir; K dizilerdeki anten sayısını belirtir, M ise M tane yönlendirilmiş kaynağa karşı gelen M tane yönlendirme vektörünü belirtir. $n_x(t)$ ve $n_y(t)$ her iki dizi elemanlarının aldığı gürültüyü ifade etmektedir. M kaynak sayısı,

AIC (Akaike Information Criteria) ve MDL (Minimum Descriptive Length) kriterleri temel alınarak kestirilir. U_x ve U_y , her iki dizi korelasyon matrisleri olan R_{xx} ve R_{yy} 'nin M tane en büyük özvektörlerini, V ise aşağıdaki şekilde tanımlanan matrisin özvektörlerini gösterebilir;

$$\mathbf{V} = \begin{bmatrix} \mathbf{U}_x^H \\ \mathbf{U}_y^H \end{bmatrix} \begin{bmatrix} \mathbf{U}_x & \mathbf{U}_y \end{bmatrix} \quad (2.61)$$

V özvektörleri elde edildikten sonra bu özvektörlere karşı düşen $\lambda_m, m = 1, 2, \dots, M$ özdeğerleri hesaplanır.

DOA ifadesi ise (2.62) ifadesi ile verilir;

$$\theta_m = \cos^{-1} \left\{ \frac{\text{Arg}(\lambda_m)}{2\pi d} \right\}, m = 1, 2, \dots, M \quad (2.62)$$

d , dalga boyu cinsinden dizi elemanları arasındaki mesafedir. ESPRIT algoritmasının değişik durumlar için tasarlanmış değişik biçimleri vardır [33-36].

ESPRIT yöntemiyle DOA kestirimi MUSIC yöntemiyle kıyaslandığında daha hızlı ve daha gürbüzdür; hesaplama yoğunluğu ise daha az karmaşıktır. Ancak ESPRIT birbirleriyle ilişkili işaret gönderen kaynakları ayırt edemez.

2.3.2.2 Yan lob engelleyici

Bu huzme biçimlendirme sistemi, bir tane ana anten ve bir çok yardımcı antenden oluşur. Ana anten oldukça yönlendirilebilir bir antendir ve istenen işaret yönüne yönlendirilir. Ana antenin hem istenen işareti hem de interferans işaretlerini yan lobları aracılığıyla aldığı varsayılmaktadır. Yardımcı antenler ise istenen işaret yönünde düşük kazançta sahip olduklarından öncelikli olarak interferans işaretlerini alırlar. Yardımcı anten dizisi ağırlıkları, ana anten dizisi cevabının yan loblarındaki interferans işaretlerinin etkisini yok edecek şekilde ayarlanır.

Eğer interferans işaretlerine hem ana anten dizi cevabı hem de yardımcı anten dizi cevabı eşitse bu durumda toplam cevap sifıra eşit olacaktır ve beyaz gürültüyü oluşturur. Dolayısıyla ağırlıklar, beyaz gürültü için interferans bastırımına karşın toplam çıkış gücünün beklenen değerini minimize edecek şekilde seçilir. Bu ifade matematiksel olarak aşağıdaki gibidir;

$$\min_{w_a} E \left\{ \left| y_m - \mathbf{w}_a^H \mathbf{x}_a \right|^2 \right\} \quad (2.63)$$

Yan lob engelleyicinin uyarlamalı bileşenlerine karşılık gelen optimum w_a ağırlıkları ise (2.64) eşitliği ile verilir;

$$\mathbf{w}_a = \mathbf{R}_a^{-1} \mathbf{r}_{ma} \quad (2.64)$$

$\mathbf{R}_a = E \left\{ \mathbf{x}_a \mathbf{x}_a^H \right\}$ ifadesi yardımcı dizi korelasyon matrisi ve r_{ma} ifadesi ise yardımcı dizi ve ana dizi arasındaki çapraz ilişki vektörüdür. Bu yöntem basit olmasına rağmen istenen işaret gücünün interferans işaretleri gücünden daha zayıf olduğu durumlarda oldukça etkilidir. İstenecek işaret gücü arttıkça toplam çıkış gücüne etkisi de artar ve bu durum engelleme oranının artmasına, istenen işaretin de engellenmesine neden olur.

2.3.2.3 LCMV (Linearly Constrained Minimum Variance)

Çoğu huzme biçimlendirme tekniği istenen işaret ve referans işareti bilgisine ihtiyaç duyar. Bu sınırlamaların üstesinden, ağırlık vektörüne doğrusal sınırlama uygulanmasıyla gelinebilir. LCMV uzamsal filtreler, filtrenin çıkış varyansını ya da gücünü bazı sınırlamalara göre minimize ederek ağırlıkları seçen huzme biçimlendiricilerdir. Diğer sınırlamalar ile beraber bu sınırlamalar, diğer kaynaklardan gelen işaretlerin varyans etkilerini minimize ederken istenen kaynaktan gelen işareti korumayı garanti altına alır.

$$\min E \{ |y|^2 \} = \mathbf{w}^H \mathbf{R}_x \mathbf{w} \quad , \quad \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (2.65)$$

\mathbf{R}_x , $x(t)$ gözlem işaretinin kovaryans matrisidir. \mathbf{C} , K tane sütun vektöründen oluşan kısıtlama matrisidir ve \mathbf{f} ise K tane skaler kısıtlama değeri içeren cevap vektörüdür. Bu ifade, Lagrange faktörizasyonu kullanılarak çözümlerse optimum ağırlıklar için çözüm aşağıdaki gibi elde edilir;

$$\mathbf{w}_{opt} = \mathbf{R}_x^{-1} \mathbf{C} (\mathbf{C}^H \mathbf{R}_x^{-1} \mathbf{C})^{-1} \mathbf{f} \quad (2.66)$$

Bu huzme biçimlendirme yöntemi esnek, optimum ağırlıkları hesaplamak için referans işarete ihtiyaç duymaz fakat kısıtlayıcı ağırlık vektörü \mathbf{C} 'nin hesaplanmasına ihtiyaç duyar.

LCMV yapısı ve LCMV'nin değişik bir uygulaması olan Genelleştirilmiş Yan Lob Engelleme (GSLC: Generalized Sidelobe Canceller) 3. bölümde ayrıntılı olarak anlatılıp performansı simülasyonlarla değerlendirilecektir.

2.3.2.4 Sıfır yönlendirme huzme biçimlendirme (Null steering beamforming)

Diğer algoritmaların aksine sıfır yönlendirme huzme biçimlendirme gelen işaretle ve onu iyileştirmekle ilgilenmez; bunun yerine istenen işaretin gelmediği yönleri bulur ve bu yönler için çıkış işaret gücünü minimize eder. Bu yaklaşımı temel alan bir teknik, ağırlık vektörünün normunu 1'e çekerken dizi çıkış ifadesi karesinin beklenen değerini minimize etmektedir.

$$\min_w \mathbf{w}^H \mathbf{R} \mathbf{w}, \quad \mathbf{w}^H \mathbf{A} \mathbf{w} = 1 \quad (2.67)$$

Bu ifadede \mathbf{A} , pozitif tanımlı simetrik bir matristir ve ağırlık vektörü kısmi parçalarının diğerleri üzerindeki önemini ayarlamak için kullanılır. Optimum ağırlık vektörü (2.68) eşitliğini sağlamalıdır;

$$\mathbf{R} \mathbf{w} = -\lambda \mathbf{A} \mathbf{w} \quad (2.68)$$

2.3.2.5 Sabit genlik algoritması (CMA: Constant Modulus Algorithm)

CMA uyarlamalı huzme biçimlendirme yapısı referans işareti dışında SMI (Sample Matrix Inversion: Örnekleme Matrisi Tersisi) yöntemiyle hemen hemen aynıdır. CMA uyarlamalı huzme biçimlendirme referans işareti ihtiyacı duymaz. Kullandığı gradyan temelli algoritmanın teorisi: İnterferans işaretlerinin varlığı gönderilen işaretin genliğini değiştirir; aksi takdirde sabit bir zarfa sahip olmalıdır. Ağırlık güncellemesi maliyet fonksiyonu beklenen değerini minimize ederek yapılır.

$$J_n = \frac{1}{2} E \{ (|y(n)|^2 - y_0^2)^2 \} \quad (2.69)$$

Ağırlıklar ise (2.70) ifadesiyle güncellenir;

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu g(\mathbf{w}(n)) \quad (2.70)$$

$y(n)$, n . iterasyondaki dizi çıkışıdır, y_0 interferansın olmadığı durumda istenen işaret zarfının genliğini belirtir; $g(\mathbf{w}(n))$ ise maliyet fonksiyonunun bir kestirimidir. CMA algoritması bazı durumlarda yakınsamayabilir; bu durumun üstesinden istenen işaretle ilgili daha çok bilgiye sahip olunarak gelinebilir.

2.3.2.6 LMS (Least Mean Squares) algoritması

Sürekli uyarılama için en genel algoritma LMS algoritmasıdır. LMS algoritması üzerinde çok fazla çalışma yapılmış çalışma şekli çok iyi anlaşılmıştır. Çok iyi bilinen hızlı iniş (steepest descent) yöntemi üzerine inşa edilmiştir, özinelemeli olarak w ağırlık vektörünü hesaplar ve günceller. Gradyan vektörünün tersi yönünde tekrarlı olarak ağırlık vektörüne düzeltmeler yaparak sonunda en küçük beklenen karesel hataya (MMSE) ulaşılabacağı fikri oldukça mantıklıdır; bu durumda ağırlık vektörü optimum değerini alacaktır. Hızlı iniş metoduna göre $n+1$ anındaki ağırlık vektörünün güncellenen değeri aşağıdaki gibi bulunur;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2} \alpha \left[-\nabla_w (E\{e^2(n)\}) \right] \quad (2.71)$$

$$\nabla_w (E\{e^2(n)\}) = -2\mathbf{r} + 2\mathbf{R}\mathbf{w} \Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) + \alpha [\mathbf{r} - \mathbf{R}\mathbf{w}(n)] \quad (2.72)$$

Alınan işaretin özilişki matrisi R ve alınan işaretle istenen işaret arasındaki karşıt ilişki vektörü r hakkında önceden bilgi sahibi olunması gerektiğinden, gerçekte gradyan vektörünün tam olarak hesaplanması mümkün değildir. En genel strateji R ve r nin anlık kestirimlerini kullanmaktır ve aşağıdaki eşitliklerle verilir;

$$\hat{\mathbf{R}} = \mathbf{x}(n)\mathbf{x}^H(n) \quad (2.73)$$

$$\hat{\mathbf{r}} = \mathbf{x}(n)d^*(n) \quad (2.74)$$

(2.73) ve (2.74) , (2.72)'de yerine konulursa;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha \mathbf{x}(n)[d^*(n) - \mathbf{x}^H(n)\mathbf{w}(n)] \quad (2.75)$$

(2.75) ifadesinde $e(n) = d(n) - \mathbf{w}^H(n)\mathbf{x}(n)$ bilgisi kullanılırsa LMS algoritması formülleri aşağıdaki gibi elde edilir;

$$\text{Çıkış,} \quad y(n) = \mathbf{w}^H \mathbf{x}(n) \quad (2.76)$$

$$\text{Hata,} \quad e(n) = d(n) - y(n) \quad (2.77)$$

$$\text{Ağırlıklar,} \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \alpha \mathbf{x}(n)e^*(n) \quad (2.78)$$

3. bölümde LMS algoritması ayrıntılı olarak anlatılıp performansı simülasyonlarla değerlendirilecektir.

2.3.2.7 RLS (Recursive Least Squares) algoritması

R matrisini ve r vektörünü ağırlıklandırılmış toplam hesabıyla kestirebiliriz.

$$\hat{\mathbf{R}}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) \quad (2.79)$$

$$\hat{\mathbf{r}}(n) = \sum_{i=1}^n \gamma^{n-i} \mathbf{x}(i) d^*(i) \quad (2.80)$$

γ , unutulma faktörü olarak isimlendirilir ve $0 < \gamma \leq 1$ eşitsizliğini sağlamalıdır; bu sayede uyarlamalı işlemci geçmişteki gözlemleri unutarak, istatistiksel değişim gösteren durumlara algoritma uyum sağlar. (2.79) ve (2.80) ifadeleri $i = n$ için olan terimler dışarı alınıp tekrar düzenlenirse;

$$\hat{\mathbf{R}}(n) = \gamma \hat{\mathbf{R}}(n-1) + \mathbf{x}(n) \mathbf{x}^H(n) \quad (2.81)$$

$$\hat{\mathbf{r}}(n) = \gamma \hat{\mathbf{r}}(n-1) + \mathbf{x}(n) d^*(n) \quad (2.82)$$

\hat{R} özilişki matrisinin tersi ise (2.83) ifadesiyle verilir;

$$\hat{\mathbf{R}}^{-1}(n) = \gamma^{-1} [\hat{\mathbf{R}}^{-1}(n-1) - \mathbf{k}(n) \mathbf{x}(n) \hat{\mathbf{R}}^{-1}(n-1)] \quad (2.83)$$

(2.83) ifadesindeki $k(n)$ kazanç vektörü olarak isimlendirilir ve (2.84) eşitliği ile verilir;

$$\mathbf{k}(n) = \frac{\gamma^{-1} \hat{\mathbf{R}}^{-1}(n-1) \mathbf{x}(n)}{1 + \gamma^{-1} \mathbf{x}^H(n) \hat{\mathbf{R}}^{-1}(n-1) \mathbf{x}(n)} \quad (2.84)$$

Optimum Wiener çözüm ifadesinde (2.81) ve (2.82) ifadeleri yerine konursa;

$$\mathbf{w}(n) = \hat{\mathbf{R}}^{-1}(n) \hat{\mathbf{r}}(n) = \gamma^{-1} [\hat{\mathbf{R}}^{-1}(n-1) - \mathbf{k}(n) \mathbf{x}(n) \hat{\mathbf{R}}^{-1}(n-1)] x [\gamma \hat{\mathbf{r}}(n-1) + d(n) \mathbf{x}(n)] \quad (2.85)$$

(2.85) ifadesi tekrar düzenlenerek istenilen iteratif ağırlık vektörü güncelleme ifadesi aşağıdaki gibi bulunur;

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{k}(n) [d(n) - \mathbf{w}^H(n-1) \mathbf{x}(n)] = \mathbf{w}(n-1) + \mathbf{k}(n) e^*(n) \quad (2.86)$$

$e(n)$ bir önceki an için hesaplanan kestirim hatasıdır ve (2.87) eşitliği ile verilir;

$$e(n) = d(n) - \mathbf{w}^H(n-1) \mathbf{x}(n) \quad (2.87)$$

Vektör iç çarpımı $w^H(n-1)x(n)$, $n-1$ anındaki ağırlık vektörü kestirimi kullanılarak elde edilen $\hat{d}(n)$ istenen işaret kestirimini belirtir. RLS algoritması 3. bölümde ayrıntılı olarak anlatılıp performansı simülasyonlarla değerlendirilicektir.

2.3.2.8 SMI (Sample Matrix Inversion) algoritması

Bu algoritmada, huzme biçimlendirici çıkışı ile referans işareti arasındaki beklenen karesel hata minimize olacak şekilde ağırlıklar seçilir. Beklenen karesel hata ifadesi aşağıdaki gibidir;

$$E\left\{\left|d(n) - \mathbf{w}^H \mathbf{x}(n)\right|^2\right\} = E\{d^2(n)\} - 2\mathbf{w}^H \mathbf{r} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (2.88)$$

$x(n)$, n anındaki dizi çıkışı, $d(n)$ referans işareti, $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^H(n)\}$ işaret kovaryans matrisi, $\mathbf{r} = E\{\mathbf{x}(n)d^*(n)\}$ ise referans işareti ile veri işareti arasındaki karşıt ilişki vektörüdür. (2.88) ifadesini minimum yapan ağırlık vektörü, w 'ye göre gradyan vektörünün sıfıra eşitlenmesiyle bulunur.

$$\nabla_{\mathbf{w}} \left(E\left\{\left|d(n) - \mathbf{w}^H \mathbf{x}(n)\right|^2\right\} \right) = -2\mathbf{r} + 2\mathbf{R}\mathbf{w} = 0 \quad (2.89)$$

Optimum ağırlık ifadesi w_{opt} ise;

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{r} \quad (2.90)$$

Optimum ağırlıklar doğrudan kovaryans matrisinin tersi alınarak hesaplanabilir. Bu algoritma referans işarete ihtiyaç duyar ve yoğun hesaplama gerektirir. Fakat buna rağmen LMS algoritmasından daha hızlıdır.

3. HUZME BİÇİMLENDİRME

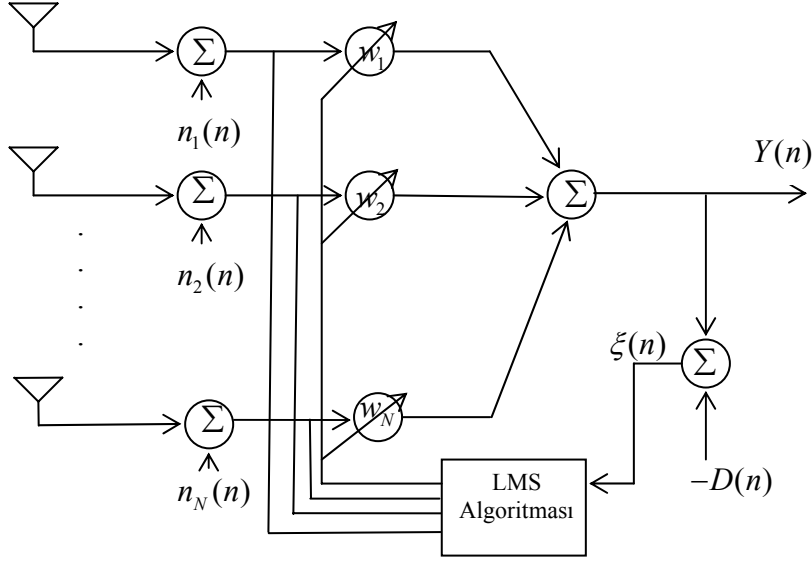
Uyarlamalı algoritmalar çok geniş bir kullanım alanına sahiptir. Haberleşmede, uyarlamalı kanal dengelemede, gürültü gidermede, spektrum analizinde ve daha bir çok uygulamada kullanılan uyarlamalı algoritmalar anten dizileriyle işaret işleme ve uyarlamalı olarak huzme biçimlendirmede de rahatlıkla kullanılabilirler.

Optimum Wiener çözümüne iteratif olarak ulaşmayı amaçlayan bu algoritmaların kuşkusuz en basiti LMS algoritmasıdır. LMS algoritması basit ve az işlem yüküne sahip olmasına karşın oldukça etkili bir algoritmadır. LMS algoritmasının giriş verisine bağlı olarak geç yakınsaması gibi bazı olumsuz yönlerinden kaçınılmak istenildiği durumlarda RLS algoritması kullanılabilir. RLS algoritması çok hızlı bir algoritma olmasına karşın oldukça fazla işlem yoğunluğuna sahiptir. Bu iki algoritma aynı sonucu elde etmeye çalışırken farklı yöntemler kullanırlar; RLS algoritması Wiener çözümüne deterministik bir yöntemle ulaşmaya çalışır. Bu bölümde bu iki algoritma ayrıntılı olarak analiz edilmiş ve performansları simülasyonlarla sunulmuştur. Bu bölümde ayrıca referans işaretine gerek duymadan çalışan LCMV huzme biçimlendirici, LCMV huzme biçimlendiriciyi gerçeklemenin değişik bir yolu olan Genelleştirilmiş Yan Lob Engelleme yapısı anlatılmakta, LMS ve RLS algoritmalarıyla gerçekleştirilen simülasyon sonuçları değerlendirilmektedir.

3.1 LMS Algoritması

LMS algoritması 1960 yılında Widrow ve Hoff tarafından geliştirilmiştir [18,22]. LMS algoritması, Hızlı İniş Algoritmasından (Steepest Descent Algorithm) yola çıkılarak bulunmuştur; bir rasgele gradyan algoritmasıdır ve bu yönüyle deterministik gradyan kullanan Hızlı İniş Algoritmasından ayrılır. LMS algoritmasının en önemli özelliği basit olmasıdır; korelasyon fonksiyonlarını kullanmaz, matris tersi işlemine ihtiyaç duymaz. Bu nedenle LMS algoritması son 40 yılda uyarlamalı işaret işleme gereken alanlarda ve doğal olarak uyarlamalı anten dizileriyle işaret işlemede oldukça yaygın kullanım alanına sahip olmuştur.

3.1.1 LMS algoritmasıyla huzme biçimlendirme



Şekil 3.1 LMS algoritmasıyla huzme biçimlendirme.

\$w\$; anten çıkışlarına uygulanan kompleks ağırlık katsayı vektörünü ve \$X\$; \$n\$ anındaki dizi çıkışı veri vektörünü göstermek üzere dizi çıkış ifadesi (3.3) eşitliği ile verilir.

$$\mathbf{w} = [w_1(n) \ w_2(n) \ \dots \ w_N(n)]^T \quad (3.1)$$

$$\mathbf{X} = [x_1(n) \ x_2(n) \ \dots \ x_N(n)]^T \quad (3.2)$$

$$Y(n) = \mathbf{w}^H \mathbf{X} \quad (3.3)$$

Dizi çıkışındaki hata ifadesi ise;

$$\xi(\mathbf{w}) = E[(D - \mathbf{w}^H \mathbf{X})(D^* - \mathbf{X}^H \mathbf{w})] = \sigma_d^2 - \mathbf{w}^H \mathbf{r} - \mathbf{r}^H \mathbf{w} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (3.4)$$

Bu ifadede \$\sigma_d^2\$ referans işaretinin varyansını, \$\mathbf{r} = E[\mathbf{X}D^*]\$ dizi çıkış işaret vektörü ile referans işareti arasındaki karşit ilişki vektörünü, \$\mathbf{R} = E[\mathbf{X}\mathbf{X}^H]\$ dizi çıkış işaret vektörünün korelasyon matrisini göstermektedir. Hatanın \$w\$'ye göre gradyan ifadesi ise aşağıdaki gibidir;

$$\nabla_{\mathbf{w}^H} \xi = -\mathbf{r} + \mathbf{R} \mathbf{w} \quad (3.5)$$

Buradan optimum hata ifadesi;

$$\mathbf{R}\mathbf{w}_o = \mathbf{r} \Rightarrow \mathbf{w}_o = \mathbf{R}^{-1}\mathbf{r} \quad (3.6)$$

Bu optimum katsayıların kullanımıyla bulunacak MMSE hata ise;

$$\xi_o = \sigma_d^2 - \mathbf{w}_o^H \mathbf{R}\mathbf{w}_o = \sigma_d^2 - \mathbf{r}^H \mathbf{w}_o \quad (3.7)$$

Optimum w katsayılarına ulaşarak bu optimum ξ_o hataya ulaşmayı amaçlayan yöntemlerden birisi Hızlı İniş Algoritması tekniğidir. Bu yönteme göre ağırlıkları güncelleme ifadesi;

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \alpha(-\nabla \xi_{w^H}) \Rightarrow \mathbf{w}(n) = \mathbf{w}(n-1) + \alpha[\mathbf{r} - \mathbf{R}\mathbf{w}(n-1)] \quad (3.8)$$

LMS algoritması ise r ve R ifadelerinin gerçek değerlerini kullanmak yerine sadece kestirimlerini kullanır. Dolayısıyla LMS algoritmasında bu ifadeler şu şekilde hesaplanır;

$$\hat{\mathbf{r}} = \frac{1}{K} \sum_{n=N-K}^N \mathbf{X}(n)D^*(n) \quad (3.9)$$

$$\hat{\mathbf{R}} = \frac{1}{K} \sum_{n=N-K}^N \mathbf{X}(n)\mathbf{X}^H(n) \quad (3.10)$$

$$\hat{\nabla} \xi_{w^H}(n) = -\hat{\mathbf{r}} + \hat{\mathbf{R}}\hat{\mathbf{w}} = -\mathbf{X}(n)D^*(n) + \mathbf{X}(n)\mathbf{X}^H(n)\hat{\mathbf{w}}(n) \quad (3.11)$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \alpha\mathbf{X}(n)[D^*(n) - \mathbf{X}^H(n)\hat{\mathbf{w}}(n-1)] \quad (3.12)$$

En son ifadede parantez içi $Y(n) = \mathbf{w}^H \mathbf{X}$ bağıntısı dikkate alınarak tekrar düzenlenirse LMS algoritması ağırlık güncelleme ifadesi (3.13) ile verilir.

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \alpha\mathbf{X}(n)[D^*(n) - \hat{Y}^*(n)] = \hat{\mathbf{w}}(n-1) + \alpha\mathbf{X}(n)e^*(n) \quad (3.13)$$

Ağırlık güncelleme ifadesindeki α terimi LMS algoritmasının adım boyu olarak isimlendirilir. LMS algoritmasının yakınsama hızını α adım boyu belirler; çok küçük α değerleri için algoritma oldukça yavaş yakınsar. Daha büyük α adım boyu değerlerini seçmek algoritmanın yakınsama hızını artırabilir ancak bu durumda katsayıların yakınsadığı değer optimum değerden daha çok uzaklaşır, dolayısıyla hata artar. Ancak α değeri belirli sınırlar içinde kaldığı sürece algoritma yakınsar. α parametresi için sınır değerleri ;

$$0 < \alpha < \frac{2}{\lambda_{\max}} \quad (3.14)$$

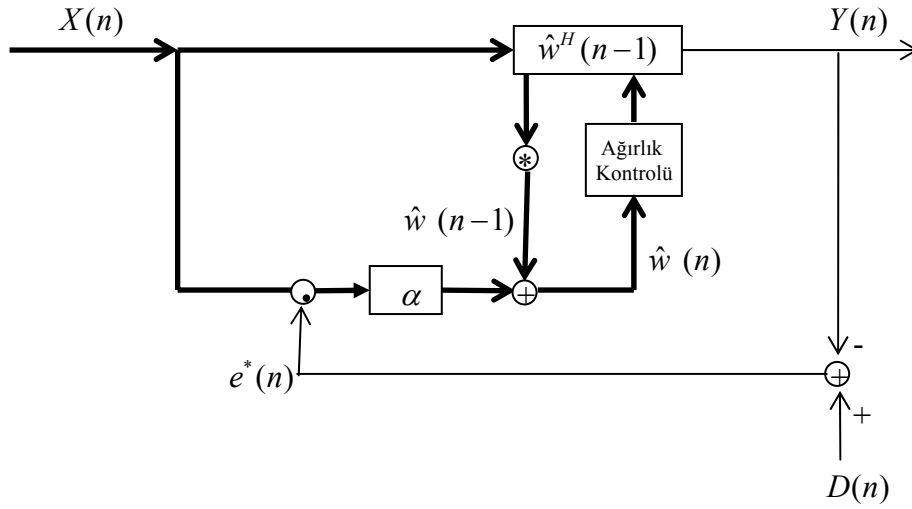
Bu ifadede λ_{\max} , R alınan işaret vektörü korelasyon matrisinin en büyük özdeğeridir. α adım boyu için istenen aralık aşağıdaki gibi de verilebilir;

$$0 < \alpha < \frac{2}{\sum_{i=1}^N \lambda_i} \quad (3.15)$$

(3.15) ifadesinde paydadaki terim R kovaryans matrisinin özdeğerler toplamıdır.

Ağırlık güncelleme ifadesinde başlangıç koşulu olarak değişik değerler kullanılabilir. Genellikle şu seçimler yapılır;

$$\hat{\mathbf{w}}(0) = 0; \quad (3.16)$$



Şekil 3.2 LMS algoritması blok diyagramı.

$$\hat{\mathbf{w}}(0) = \frac{\mathbf{A}}{M}; \quad (3.17)$$

$$\hat{\mathbf{w}}(0) = \mathbf{w}_q; \quad (3.18)$$

\mathbf{A} ; istenen yön için yönlendirme (steering) vektörünü, M anten sayısını belirtir, \mathbf{w}_q ise Quiescent ağırlık vektörüdür [26].

$$\mathbf{A} = [1 \ e^{-j\phi_1} \ e^{-j\phi_2} \ \dots \ e^{-j\phi_{M-1}}]^T \quad (3.19)$$

$$\mathbf{w}_q = \mathbf{A}(\mathbf{A}^H \mathbf{A})^{-1} \quad (3.20)$$

3.1.2 LMS algoritması simülasyon sonuçları

Algoritmanın değişik durumlarda performansını ölçmek için yapılan simülasyonlarda antenler arası mesafe $d = \lambda / 2$ olan ULA geometrili anten dizisi kullanılmıştır. ULA anten dizi verisi olarak, istenen yönden gelen işaret ve değişik yönlerden gelen interferans işaretleri sıfır ortalamalı gauss dağılımlı rasgele işaret olarak üretilmiştir. Herbir antende alınan işarete eklenen gürültü bileşeni de yine sıfır ortalamalı gauss dağılıma sahip kompleks bir gürültüdür. Simülasyonlarda basitlik olması açısından referans işareti istenen işaretle aynı alınmış, değişik yönlerden gelen işaretler birbirleriyle korelasyonsuz tutulmuş, eşit güçte oldukları varsayılmıştır.

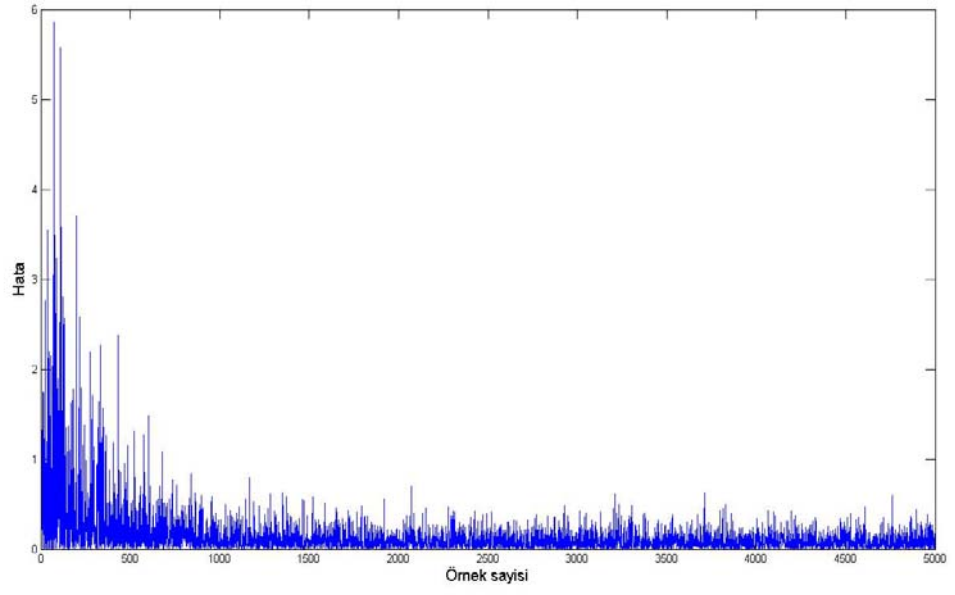
Simülasyon 1:

İlk simülasyon uygulamasında 4 anten kullanılmıştır. İstenen s_o işaretin geliş açısının 30° , -20° , 0° ve 60° geliş açılarında sahip 3 tane interferans kaynağının olduğu varsayılmaktadır.

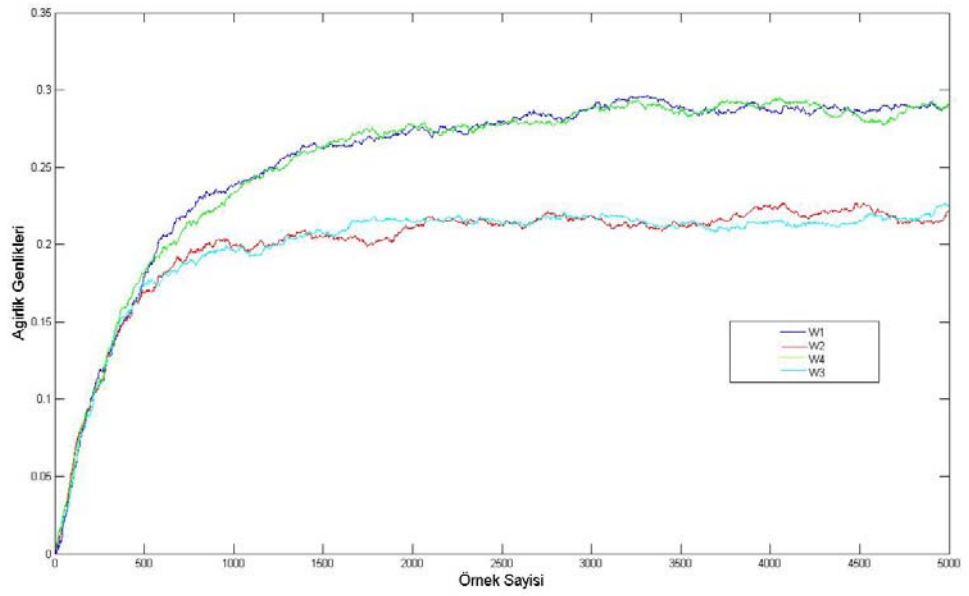
α adım boyu parametresi 0.0005 ve gürültü varyansı 0.3 olarak alınmıştır. Şekil 3.3'deki öğrenme eğrisinden seçilen α değerinin optimum değere yakın bir değer olduğu görülmektedir; çünkü yakınsadığı değerle çizdirilen Şekil 3.5'deki anten dizisi huzme biçiminde hemen hemen interferans işaretleri açılarında derin sıfırlar görülmektedir; öyleki bu sıfırlar yaklaşık 50 dB aşağıdadır. Algoritma yaklaşık 1500 iterasyonda yakınsamaktadır. Şekil 3.4'de görüldüğü gibi ağırlık katsayıları 1500. iterasyondan sonra fazla bir değişim göstermemektedir.

Algoritmanın yakınsadığı katsayılar;

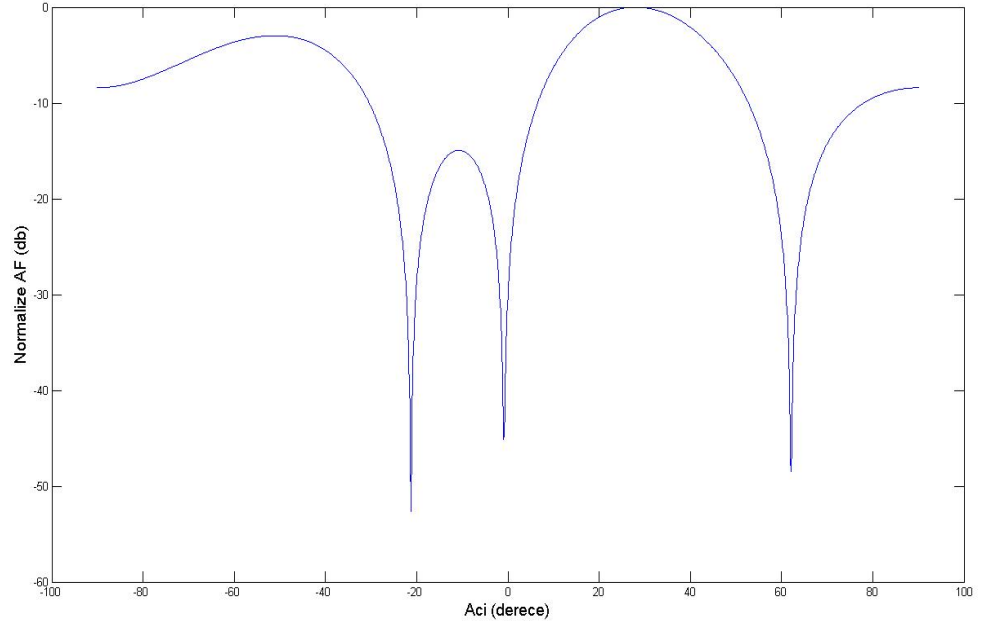
$$w_1 = 0.2875 - 0.0002i ; w_2 = -0.1357 - 0.1727i ; w_3 = -0.1730 - 0.1395i ; w_4 = 0.0028 + 0.2869i$$



Şekil 3.3 Hatanın zamanla değişimi.

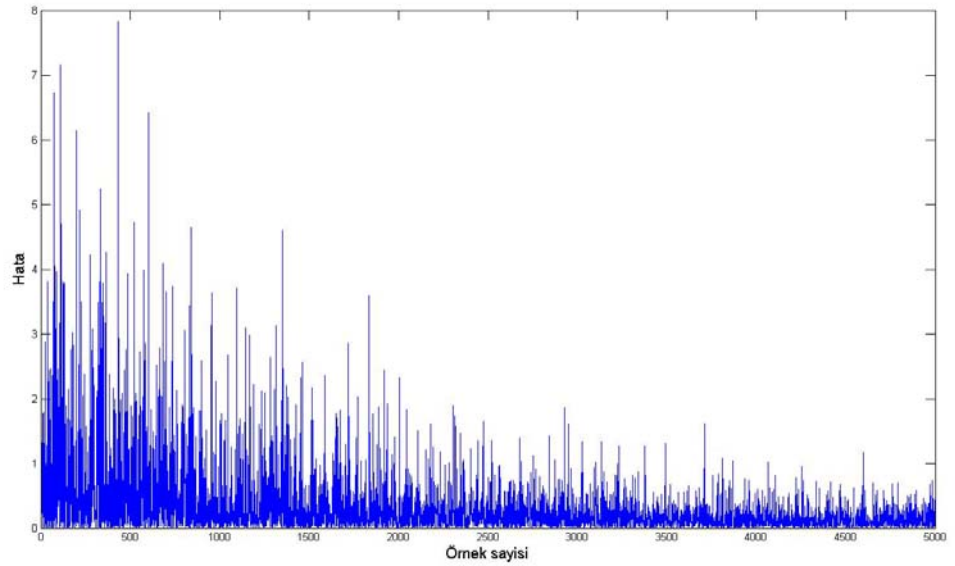


Şekil 3.4 Ağırlık genliklerinin zamanla değişimi.



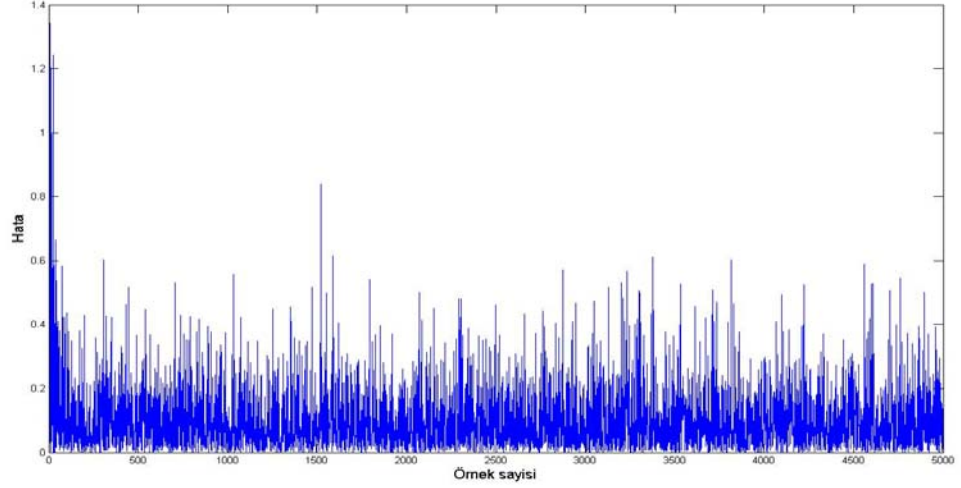
Şekil 3.5 Huzme biçimi.

α adım boyu parametresini değiştirerek öğrenme eğrisinin nasıl etkileneceğini inceleyerek;



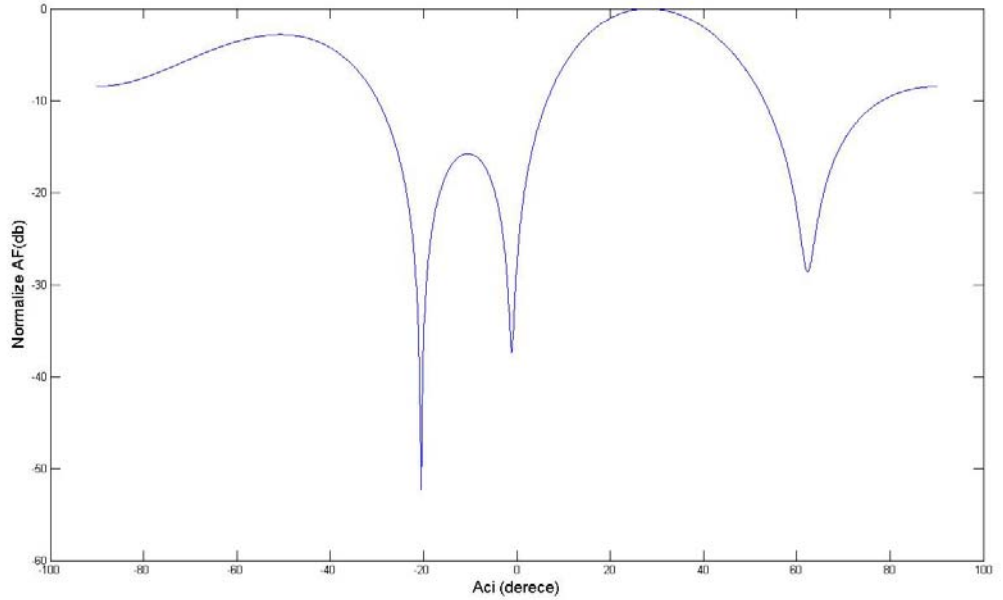
Şekil 3.6 Hatanın zamanla değişimi ($\alpha = 0.0001$).

$\alpha = 0.0001$ alındığında Şekil 3.6'da görüldüğü gibi algoritma çok yavaş yakınsamaktadır 5000 iterasyon sonunda hala yakınsamamıştır.



Şekil 3.7 Hatanın zamanla değişimi ($\alpha = 0.01$).

$\alpha = 0.01$ alındığında ise Şekil 3.7'de görüldüğü gibi algoritma çok hızlı yakınsamıştır fakat bu değer optimum değerden oldukça uzaktır ve hata oldukça fazladır. Bu durumda elde edilen anten dizi huzme biçimi ise Şekil 3.8'de görülmektedir. Algoritma interferans işaretlerinin açılarını doğru bulmasına rağmen yeterince derin sıfırlar koyamamıştır.



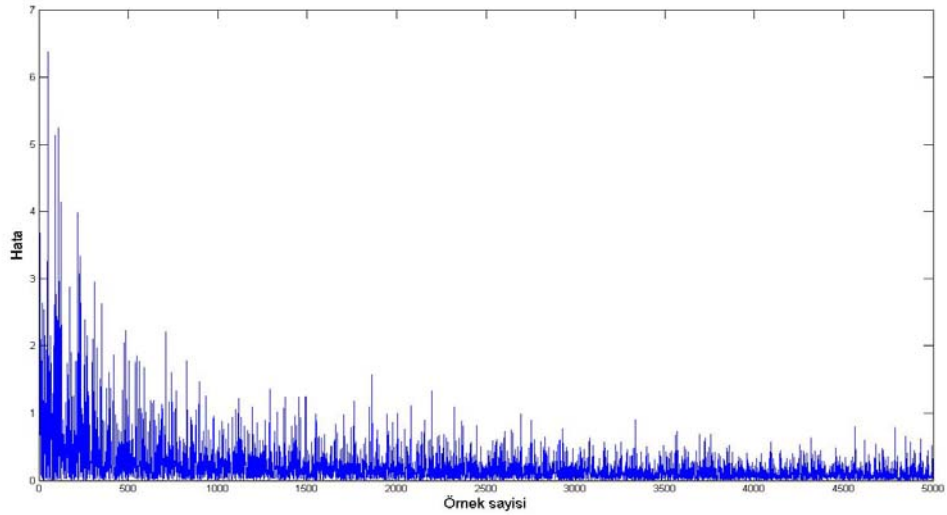
Şekil 3.8 Huzme biçimi ($\alpha = 0.01$).

Simülasyon 2:

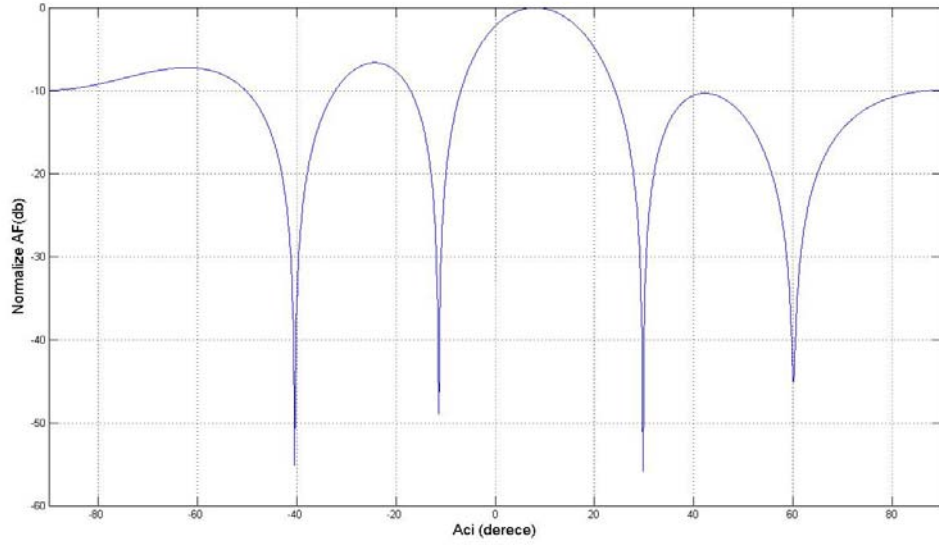
İkinci simülasyonda 5 anten kullanılmıştır, istenen s_0 işaretin geliş açısı 0° ve $-40^\circ -10^\circ 30^\circ$ ve 60° geliş açlarına sahip 4 tane interferans işaret kaynağı olduğu varsayılmaktadır.

LMS algoritması adım boyu $\alpha = 0.0005$ için yakınsama eğrisi Şekil 3.9'da görülmektedir. α değeri oldukça küçük seçildiği için algoritma geç yakınsamıştır; yaklaşık 3000 iterasyon sürmüştür; fakat Şekil 3.10'da görüldüğü gibi yakınsadığı değerle elde edilen huzme biçiminde interferans işaretlerinin açılarını tam olarak yakalamış, oldukça derin sıfırlar koymuştur.

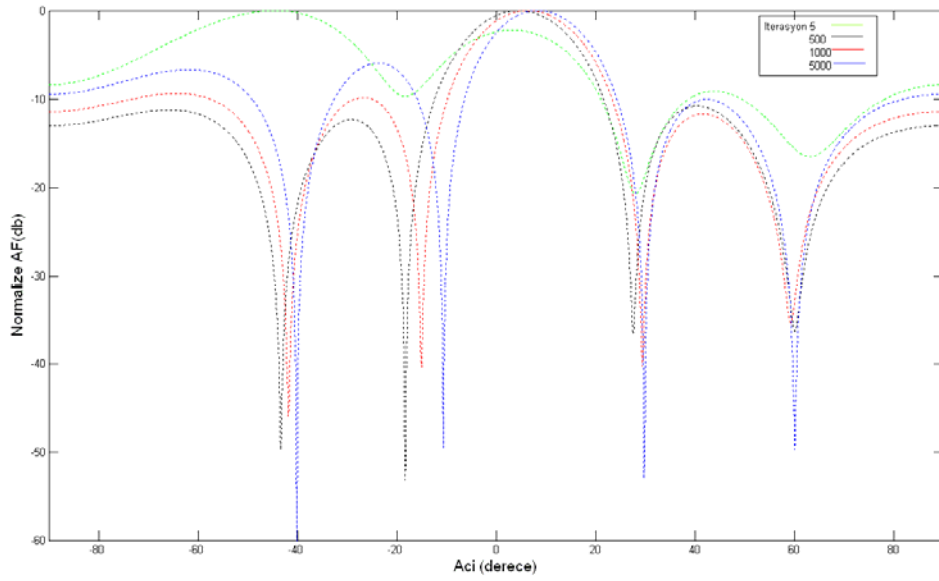
Şekil 3.11'de algoritmanın değişik iterasyon sayılarında bulunduğu ağırlık katsayılarıyla çizdirilmiş huzme biçimleri gösterilmektedir. Algoritma daha henüz 5. iterasyonda interferans işaretlerinin sayısını algılamakta ve sıfırlar koymaya çalışmaktadır; fakat henüz interferans işaretlerinin geliş açılarını bulamamıştır. İterasyon sayısı arttıkça algoritma interferans işaretlerinin geliş açılarını tam olarak bulmakta ve bu açılarda sıfırlar koymaya çalışmaktadır ama yine de bu sıfırlar yeteri kadar derin değildir. 1000. iterasyonda hala bazı işaretlerin geliş açılarında hatalar vardır. 5000. iterasyonda elde edilen katsayılar tüm interferans işaretlerinin açılarını hatasız bulmakta ve yaklaşık 50 dB aşağı çekmektedir.



Şekil 3.9 Hatanın zamanla değişimi.



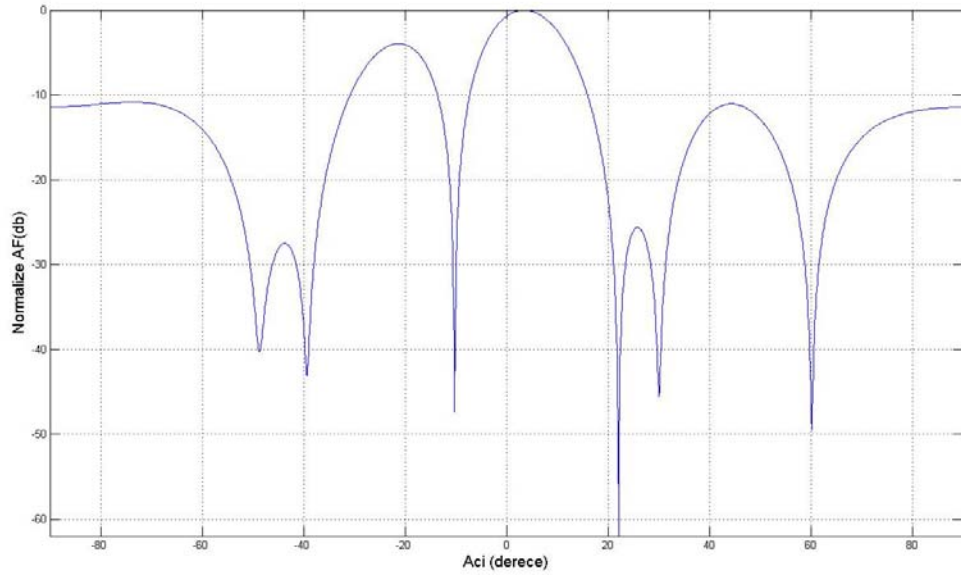
Şekil 3.10 Huzme biçimi.



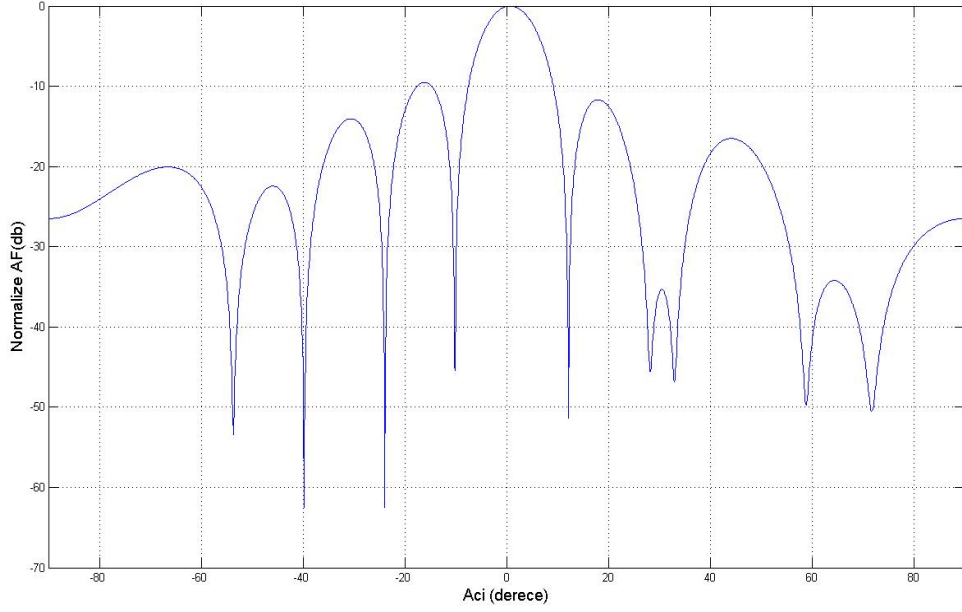
Şekil 3.11 Huzme biçiminin zamanla değişimi.

Şunu hemen belirtelim ki şu ana kadar anten sayısı ile işaret sayısı birbirine eşit alındı; şüphesiz ki bu durum olabilecek en kötü senaryoya karşı düşmektedir ve simülasyonlardan görüldüğü gibi algoritma böyle bir durumda dahi interferans işaretlerinin geliş açısı yönünde derin sıfırlar koymayı başarabilmektedir. Algoritmanın istediğimiz gibi çalışabilmesi için anten sayısı interferans işaret sayısından fazla olmalıdır. Şimdi 2. simülasyonda kullanılan senaryo için anten sayısını artırarak nasıl bir huzme biçimi elde edileceğini inceleyelim.

Şekil 3.12’de 7 anten Şekil 3.13’de ise 10 anten kullanılmıştır. Anten sayısını artırdığımızda huzme biçiminde lob sayısının artması ve ana lobun daralması beklediğimiz bir sonuçtur. Ana lobun daralması daha önce 5 antenli yapıda 0° ’den gelen istenen işaret yönünü tam olarak yakalayamamış huzme biçiminin 0° ’ye daha da yaklaşmasına hatta 10 antenli durum için tam olarak yakalamasına fırsat vermiştir. İnterferans işaretlerinin geliş açıları ise tam olarak yakalanmış ve derin sıfırlar konulmuştur. İnterferans işaretleri geliş açılarının dışında bazı açılar için de derin sıfırlar konulmuştur, bu durum lob sayısının artması gerektiğinin doğal bir sonucudur. Aynı zamanda şekillerden yan lobların daha aşağıya bastırıldığı da görülmektedir. Çok antenli durum için kuşkusuz algoritma çok daha kısa sürede yakınsayacaktır; ancak bu durum her bir anten için alıcıda eklenen gürültü bileşen gücünün işaret gücüne oranla çok düşük olduğu varsayımına dayanmaktadır.



Şekil 3.12 Huzme biçimi ($M = 7$).



Şekil 3.13 Huzme biçimi ($M = 10$).

3.2 RLS Algoritması

Hatanın karesinin beklenen değerini minimize etmeye çalışan gradyan temelli algoritmalar yerine RLS algoritması LS (Least Squares: En küçük kareler) yöntemini kullanır; bu yöntem giriş verisinin istatistiksel özellikleriyle ilgilenmez. İstatistiksel yaklaşımlı yöntemlerde aynı istatistiksel özelliklere sahip giriş verileri için aynı sonuçlar elde edilmesine karşın LS yönteminde farklı olan her giriş verisi için farklı sonuçlar elde edilecektir. Bu yöntem fazla işlem yoğunluğu içermesine karşın oldukça etkilidir ve çok hızlı yakınsama sağlar.

LS kestirim yöntemi 1795 yılında Gauss tarafından bulunmuştur. 1950' li yıllarda LS yöntemini özyinelemeli (recursive) olarak gerçekleyebilmek için bir çok bilim adamı uğraş vermiştir; Follin, Swerlin, Kalman, Bucy [26].

3.2.1 RLS algoritmasıyla huzme biçimlendirme

Anten dizisi çıkış ifadesi yazılıp her bir an için istenen işaretle farkı alınıp hata hesaplanırsa;

$$Y(n) = \mathbf{w}^H(n)\mathbf{X}(n) \quad (3.21)$$

$w(n)$ anten dizisi ağırlık katsayı vektörünü $X(n)$ ise anten dizisi giriş vektörünü göstermektedir. $D(n)$ istenen işaret ve M anten sayısı olmak üzere;

$$\mathbf{w}(n) = \begin{bmatrix} w_1(n) \\ w_2(n) \\ \cdot \\ \cdot \\ w_M(n) \end{bmatrix}; \quad \mathbf{X}(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \cdot \\ \cdot \\ x_M(n) \end{bmatrix}$$

$$e(n) = D(n) - \mathbf{w}^H(n)\mathbf{X}(n); \quad n = 1, 2, \dots, N \quad (3.22)$$

Her bir an için ağırlıklandırılmış hataların toplamını veren ifade ise:

$$\xi_\lambda(n) = \sum_{n=1}^N \lambda^{N-n} |e(n)|^2 = \sum_{n=1}^N \lambda^{N-n} [(D(n) - \mathbf{w}^H(n)\mathbf{X}(n))(D^*(n) - \mathbf{X}^H(n)\mathbf{w}(n))] \quad (3.23)$$

(3.23) ifadesinin $\mathbf{w}^H(n)$ 'e göre gradyanının sıfıra eşitlenmesinden;

$$\hat{\mathbf{w}}_{lse}(n) = \boldsymbol{\Phi}^{-1}(n)\boldsymbol{\Phi}_{xd^*}(n) \quad (3.24)$$

$$\boldsymbol{\Phi}(n) = \sum_{n=1}^N \lambda^{N-n} \mathbf{X}(n)\mathbf{X}^H(n) \quad (3.25)$$

$$\boldsymbol{\Phi}_{xd^*}(n) = \sum_{n=1}^N \lambda^{N-n} \mathbf{X}(n)D^*(n) = \mathbf{X}(n)D^*(n) + \lambda\boldsymbol{\Phi}_{xd^*}(n-1) \quad (3.26)$$

$\boldsymbol{\Phi}(n)$ giriş verisinin ağırlıklandırılmış korelasyon matrisini, $\boldsymbol{\Phi}_{xd^*}(n)$ ise giriş vektörü ile istenen işaret arasındaki ağırlıklandırılmış korelasyon vektörünü belirtmektedir. $\boldsymbol{\Phi}(n)$ ifadesindeki toplam açılıp özyinelemeli olarak hesaplanabilecek bir ifadeye dönüştürülüp Lemma matris tersi yöntemi kullanılırsa $\boldsymbol{\Phi}^{-1}(n)$ için özyinelemeli hesaplanabilecek ifade şu şekilde bulunur;

$$\boldsymbol{\Phi}(n) = \lambda\boldsymbol{\Phi}(n-1) + \mathbf{X}(n)\mathbf{X}^H(n) \quad (3.27)$$

$$\boldsymbol{\Phi}^{-1}(n) = \lambda^{-1}\boldsymbol{\Phi}^{-1}(n-1) - \frac{\lambda^{-2}\boldsymbol{\Phi}^{-1}(n-1)\mathbf{X}(n)\mathbf{X}^H(n)\boldsymbol{\Phi}^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{X}^H(n)\boldsymbol{\Phi}^{-1}(n-1)\mathbf{X}(n)} \quad (3.28)$$

$$(3.28) \text{ ifadesinde } \mathbf{g}(n) = \frac{\lambda^{-1}\boldsymbol{\Phi}^{-1}(n-1)\mathbf{X}(n)}{1 + \lambda^{-1}\mathbf{X}^H(n)\boldsymbol{\Phi}^{-1}(n-1)\mathbf{X}(n)} \text{ ve } \boldsymbol{\Phi}^{-1}(n) = \mathbf{P}(n) \text{ olarak}$$

tanımlanırsa giriş verisi korelasyon matris tersini özyinelemeli olarak hesabını veren ifade;

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{g}(n)\mathbf{X}^H(n)\mathbf{P}(n-1) \quad (3.29)$$

$$(3.29) \text{ ifadesinde her iki taraf } X(n) \text{ ile çarpılıp } \mathbf{g}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{X}(n)}{1 + \lambda^{-1} \mathbf{X}^H(n) \mathbf{P}(n-1) \mathbf{X}(n)}$$

bilgisi kullanılarak aşağıdaki eşitlik elde edilebilir;

$$\mathbf{g}(n) = \mathbf{P}(n) \mathbf{X}(n) \quad (3.30)$$

$\mathbf{g}(n)$; RLS algoritmasının kazanç vektörü olarak isimlendirilir. Son olarak özyinelemeli olarak ağırlık katsayıların hesabına izin veren ifadeyi bulabilmek için (3.26) ve (3.29) ifadeleri (3.24) ifadesinde yerine konularak;

$$\hat{\mathbf{w}}_{lse}(n) = \hat{\mathbf{w}}_{lse}(n-1) + \mathbf{g}(n) [D^*(n) - \mathbf{X}^H(n) \hat{\mathbf{w}}_{lse}(n-1)] \quad (3.31)$$

Bir önceki andaki katsayılar kullanılarak bulunan çıkıştaki hata $\varepsilon_p(n)$ olarak tanımlanırsa, (3.31) ifadesi şu şekilde düzenlenebilir;

$$\varepsilon_p(n) = D(n) - \hat{\mathbf{w}}_{lse}^H(n-1) \mathbf{X}(n) \quad (3.32)$$

$$\hat{\mathbf{w}}_{lse}(n) = \hat{\mathbf{w}}_{lse}(n-1) + \mathbf{g}(n) \varepsilon_p^*(n) \quad (3.33)$$

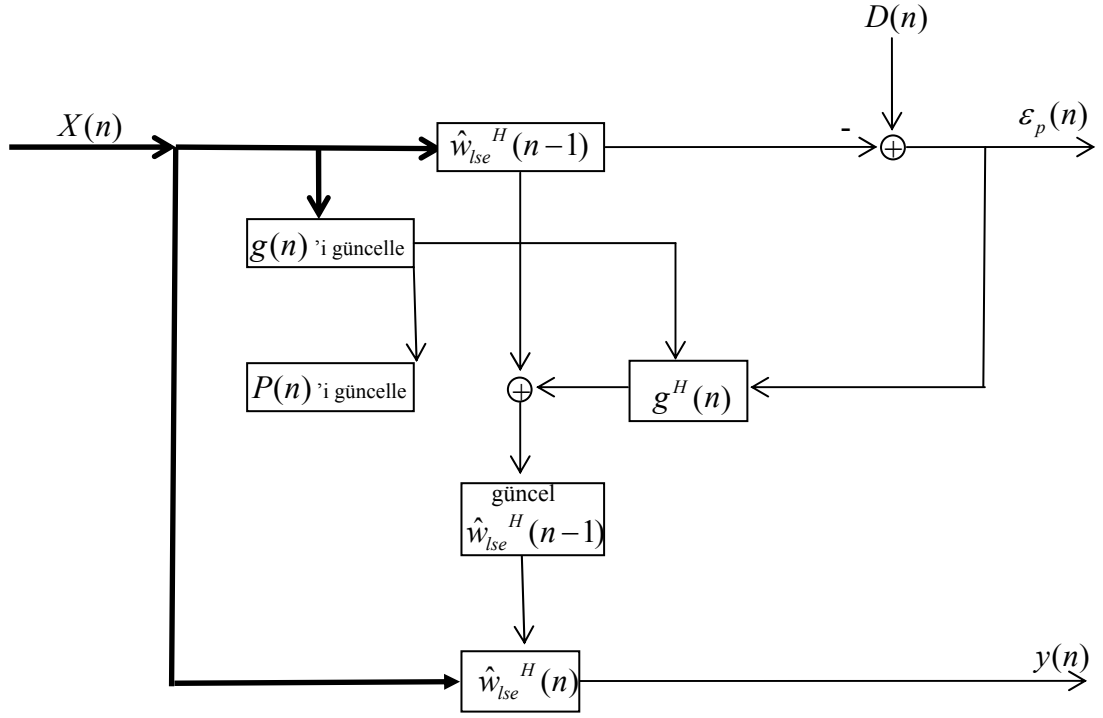
RLS algoritması şu şekilde özetlenebilir:

- Başlangıç koşulu:

$$P(0) = \frac{1}{\delta} \mathbf{I}; \quad I: M \times M \text{ boyutunda birim matris; } \delta: \text{ sabit küçük bir değer}$$

$$\hat{\mathbf{w}}(0) = \frac{\mathbf{A}}{M}; \quad A: \text{ istenen işaret için yönlendirme vektörü; } M: \text{ anten sayısı}$$

- $\mathbf{g}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{X}(n)}{1 + \lambda^{-1} \mathbf{X}^H(n) \mathbf{P}(n-1) \mathbf{X}(n)}$
- $\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{g}(n) \mathbf{X}^H(n) \mathbf{P}(n-1)$
- $\varepsilon_p(n) = D(n) - \hat{\mathbf{w}}_{lse}^H(n-1) \mathbf{X}(n)$
- $\hat{\mathbf{w}}_{lse}(n) = \hat{\mathbf{w}}_{lse}(n-1) + \mathbf{g}(n) \varepsilon_p^*(n)$



Şekil 3.14 RLS algoritması akış diyagramı.

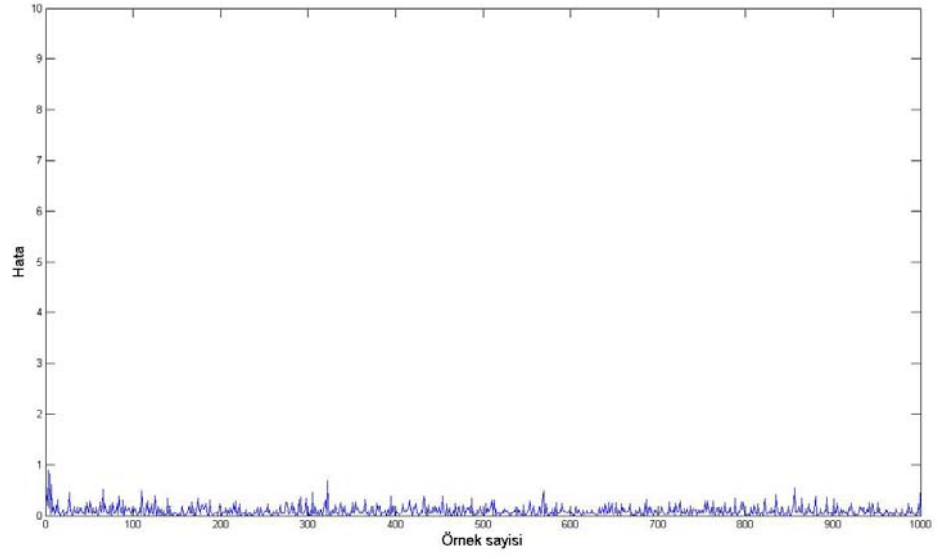
3.2.2 RLS algoritması simülasyon sonuçları

Simülasyon 1:

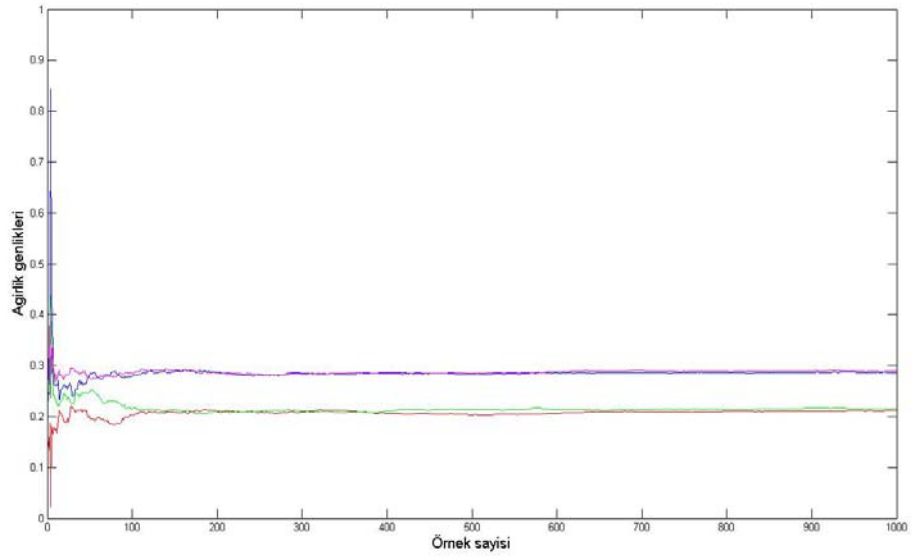
İstenen s_o işaretin geliş açısının 30° , -20° , 0° ve 60° geliş açılarına sahip 3 tane interferans kaynağının olduğu varsayılmaktadır.

4 anten kullanılmıştır ve gürültü varyansı 0.3 olarak alınmıştır. Aynı koşullarda LMS algoritması yaklaşık 1500 iterasyon sonunda yakınsamasına karşın Şekil 3.15'deki RLS öğrenme eğrisinden görüldüğü gibi RLS algoritması çok hızlı yakınsamıştır. Öğrenme eğrisi şeklinden algoritmanın hemen yakınsadığı görülse de ağırlık katsayılarının zamanla değişimi çizdirilip yakınsadığı iterasyon değerinin görülmesi daha sağlıklı olacaktır.

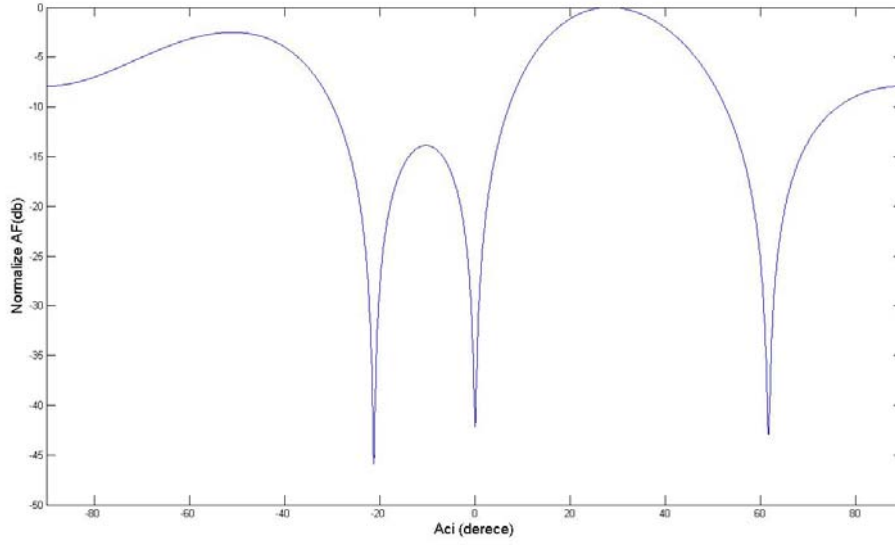
Şekil 3.16'da ağırlık katsayı genliklerinin zamanla değişimi görülmektedir. Ağırlık genliklerindeki dalgalanma yaklaşık 120 iterasyona kadar sürmektedir ve bu değerden sonra önemli bir değişim olmamaktadır. Bu an için elde edilen katsayılar kullanılarak çizdirilmiş huzme biçimi ise Şekil 3.17'de görülmektedir; bu şekil LMS simülasyonunda elde edilen huzme biçimi ile hemen hemen aynıdır.



Şekil 3.15 Hatanın zamanla değişimi.



Şekil 3.16 Ağırlık genliklerinin zamanla değişimi.

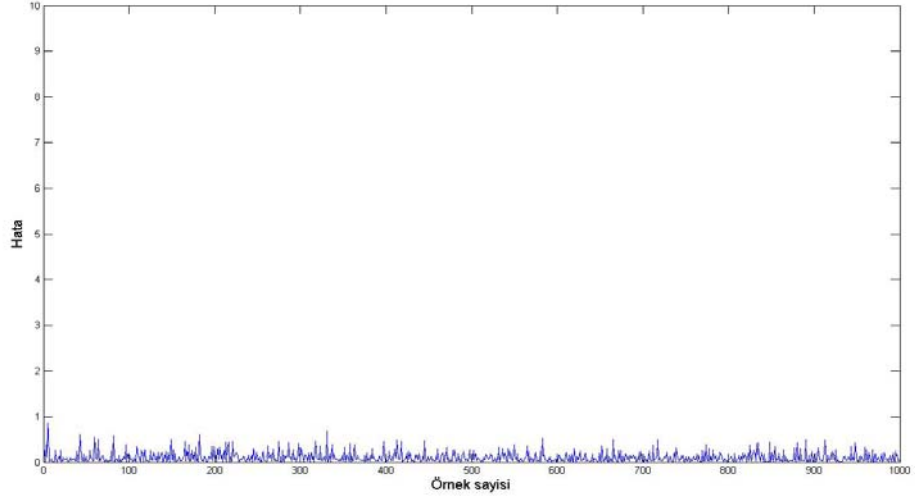


Şekil 3.17 Huzme biçimi.

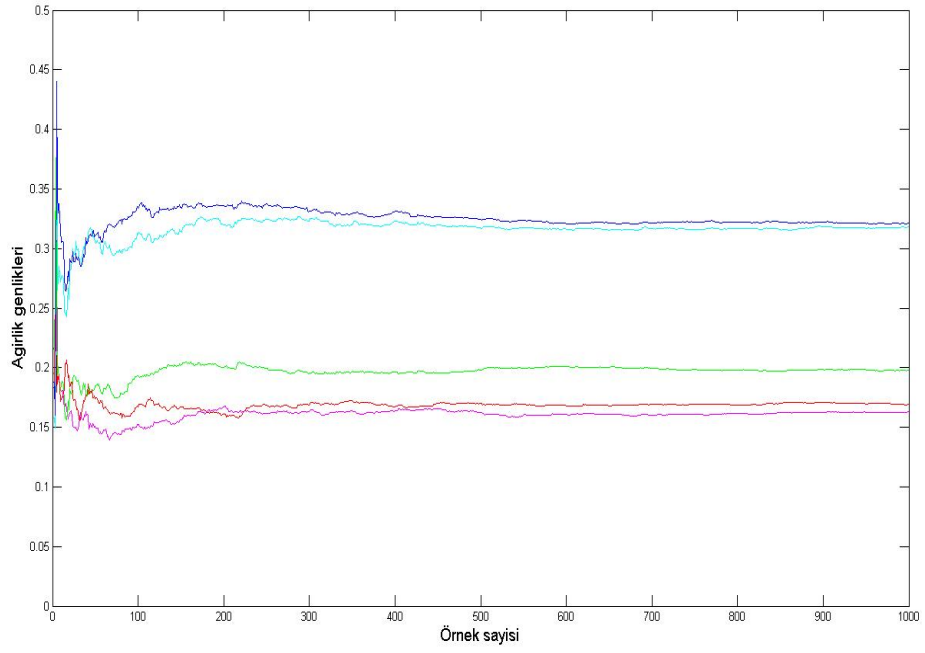
Simülasyon 2:

İstenen s_o işaretin geliş açısı 0° ve $-40^\circ -10^\circ 30^\circ$ ve 60° geliş açlarına sahip 4 tane interferans işaret kaynağı olduğu varsayılmaktadır.

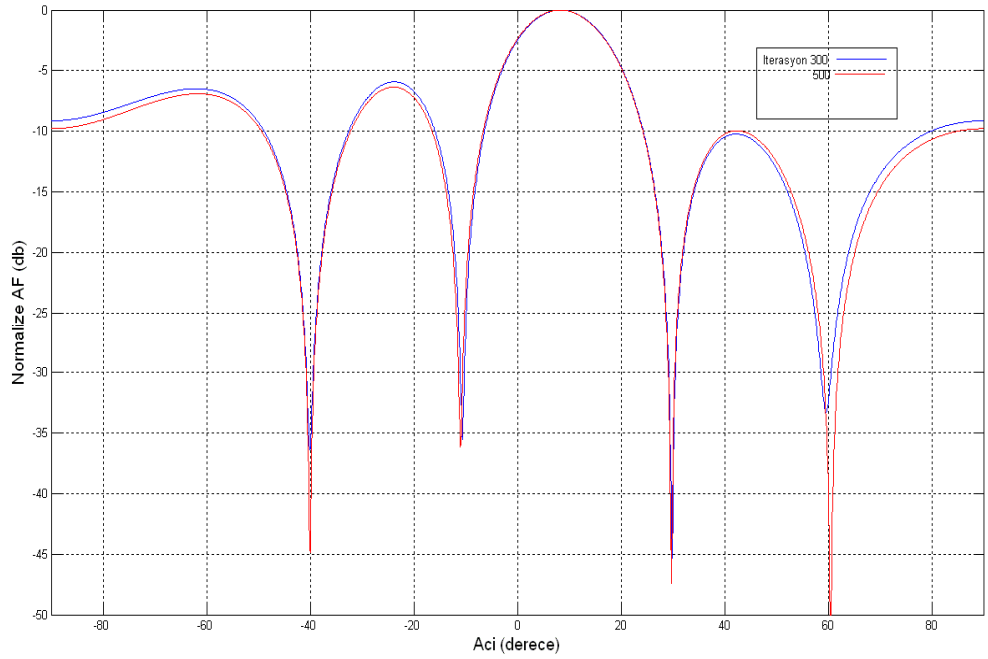
Şekil 3.19'daki ağırlık genlikleri değişiminden algoritmanın yaklaşık 300 iterasyonda yakınsadığı görülmektedir ,fakat bazı katsayılar da dalgalanmalar mevcuttur ve bu katsayılar da yaklaşık 500. iterasyonda dengeye oturmaktadır. Hem 300. iterasyonda hem de 500. iterasyonda elde edilen katsayılarla çizdirilmiş huzme biçimleri Şekil 3.20'de görülmektedir. Şekilden görüldüğü gibi algoritma hem 300. hem de 500. iterasyonda interferans işaretlerinin açılarını tam olarak yakalayabilmiştir. Ancak dengeye oturan 500. iterasyon katsayıları interferans işaretleri geliş açıları yönünde daha derin sıfırlar koymuştur.



Şekil 3.18 Hatanın zamanla değişimi.



Şekil 3.19 Ağırlık genliklerinin zamanla değişimi.



Şekil 3.20 Huzme biçimi.

3.3 LMS ve RLS Algoritmalarının Karşılaştırılması

LMS ve RLS algoritmaları kullanılarak yapılan simülasyon sonuçlarından da görüldüğü gibi RLS algoritması LMS algoritmasına göre çok daha hızlı yakınsamaktadır. Bunun sebebi LMS algoritmasının gözlem işareti korelasyon matrisinin özdeğerlerine bağımlı olmasıdır. Gözlem işareti korelasyon matrisi özdeğer dağılımı ne kadar fazlaysa LMS algoritmasının yakınsaması o derece uzun sürer. Diğer taraftan RLS algoritması gözlem işaretinin istatistiksel özellikleriyle ilgilenmez, deterministik olarak işaretin kendisiyle ilgilenir ve gözlem işareti korelasyon matrisinin özdeğer dağılımından etkilenmez. Ancak RLS algoritması gözlem işareti korelasyon matrisinin tersini deterministik olarak hesaplama yoluna gittiğinden işlem yükü oldukça fazladır. LMS algoritması geç yakınsamasına rağmen basit olması sebebiyle çok yaygın kullanım alanına sahiptir. LMS algoritmasında adım boyu katsayısı zamanla değişken olarak tasarlanıp algortmada değişiklikler yapıldığında performansı oldukça iyileşmektedir; algoritmanın performansı artırılmış NLMS (Normalize LMS) gibi bir çok değişik versiyonu mevcuttur [11-19].

3.4 Genelleştirilmiş Yan Lob Engelleyici

Haberleşme sistemlerinde alıcı, her zaman referans işareti hakkında bilgi sahibi olmayabilir. Böyle bir durumda alınan işareten istenen işaretin elde edilebilmesi için istenen işaret hakkında başka bilgilerin kullanılması gerekir ve genellikle bu bilgi istenen işaretin geliş açısıdır.

LCMV huzme biçimlendirici referans işarete ihtiyaç duymadan istenen işareti elde edebilir; bunun için her bir anten çıkışına uygulanan ağırlık katsayıları hatayı minimize ederken aynı zamanda istenen işaret yönünden gelen işaretlerin alımını da maksimum yapmalıdır. Frost [24] uyarlamalı olarak LCMV huzme biçimlendirici yapısını LMS algoritmasıyla sunmuştur. LCMV huzme biçimlendiriciyi gerçeklemenin bir başka yolu ise Genelleştirilmiş Yan Lob Engelleyici (GSLC: Generalized Sidelobe Canceller) yapısıdır; bu yapı Griffiths ve Jim [39] tarafından sunulmuştur. Bu bölümde temel LCMV yapısı ve Frost'un LMS algoritması anlatıldıktan sonra GSLC yapısı ayrıntılı olarak ele alınıp LMS ve RLS algoritmaları kullanılarak gerçekleştirilmiş GSLC simülasyon sonuçları sunulacaktır.

3.4.1 LCMV huzme biçimlendirici

LCMV huzme biçimlendirici yapısında anten çıkışlarına uygulanan ağırlık katsayı vektörü aşağıdaki eşitliği sağlamalıdır;

$$\mathbf{w}^H \mathbf{C} = \mathbf{g}^H \quad (3.34)$$

(3.34) ifadesinde \mathbf{w}^H , $1 \times M$ boyutlu vektör, \mathbf{C} , $M \times N$ boyutlu sınırlandırma matrisi \mathbf{g}^H ise $1 \times N$ boyutlu bir vektördür; M anten sayısını belirtirken N ise sınırlandırma sayısı olarak düşünülebilir. \mathbf{C} matrisinin ilk sütunu genellikle istenen işaret için yönlendirme vektörü olarak seçilir ve bu durumda \mathbf{g} vektörünün ilk elemanı 1'dir çünkü istenen işaret yönünden gelen tüm işaretlerin kayıpsız alınması istenmektedir. \mathbf{C} matrisinin diğer sütunlarında istenildiği gibi sınırlamalar konulabilir örneğin interferans işaretleri yönünde \mathbf{g} vektöründe sıfır konulabilir ve interferans işaretlerine çok yakın açılar için de sıfır konularak daha düzgün bir huzme biçimi elde edilebilir. LCMV huzme biçimlendiricide ağırlık katsayıları sınırlandırma koşullarını sağlarken aynı zamanda anten dizisi çıkış işaret gücünü minimum yapmalıdır;

$$E[y^2] = \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (3.35)$$

(3.35) ifadesinde R anten dizisi tarafından alınan işaret vektörünün korelasyon matrisini belirtir. (3.34) ve (3.35) ifadelerinin ortak çözümü Lagrange çarpanı yöntemiyle yapılabilir; maliyet fonksiyonu ve maliyet fonksiyonunu minimum yapan çözüm ifadesi aşağıdaki gibidir:

$$\mathbf{J} = \mathbf{w}^H \mathbf{R} \mathbf{w} + [\mathbf{w}^H \mathbf{C} - \mathbf{g}^H] \boldsymbol{\kappa} + \boldsymbol{\kappa}^H [\mathbf{C}^H \mathbf{w} - \mathbf{g}] \quad (3.36)$$

$$\mathbf{R} \mathbf{w} + \mathbf{C} \boldsymbol{\kappa} = 0 \quad (3.37)$$

$$\mathbf{w} = -\mathbf{R}^{-1} \mathbf{C} \boldsymbol{\kappa} \quad (3.38)$$

Bu ifadelerde $\boldsymbol{\kappa}$, $N \times 1$ boyutlu Lagrange çarpanıdır. (3.38) ifadesinin (3.34) ifadesinde yerine konulmasından;

$$-\boldsymbol{\kappa}^H \mathbf{C}^H \mathbf{R}^{-1} \mathbf{C} = \mathbf{g}^H \quad (3.39)$$

(3.39) ifadesi $\boldsymbol{\kappa}^H$ için çözümlü (3.38) eşitliğinde yerine konulursa LCMV için istenilen ağırlık katsayı ifadesi (3.40) eşitliği ile verilir.

$$\mathbf{w}_{lcmv}^H = \mathbf{g}^H [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{C}^H \mathbf{R}^{-1} \quad (3.40)$$

3.4.1.1 Frost'un uyarlamalı LCMV algoritması

Frost LCMV yapısındaki koşulları sağlayan ağırlık katsayılarının uyarlamalı olarak bulunmasını LMS algoritmasıyla gerçekleştirmiştir. Bunun için LMS algoritması ağırlık güncelleme ifadesinde ve başlangıç koşullarında bazı değişiklikler yapmıştır. Frost'un LMS algoritması aşağıdaki eşitliklerle verilmektedir.

$$\hat{\mathbf{w}}(0) = \mathbf{w}_q \quad (3.41)$$

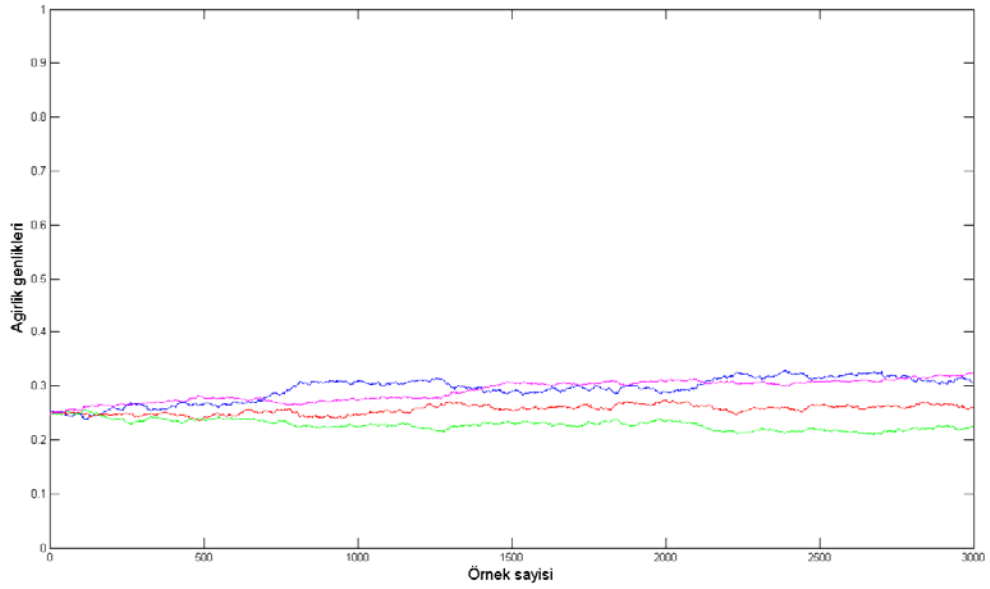
$$\mathbf{w}_q = \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{g} \quad (3.42)$$

$$\mathbf{P}_c = \mathbf{I} - \mathbf{C} (\mathbf{C}^H \mathbf{C})^{-1} \mathbf{C}^H \quad (3.43)$$

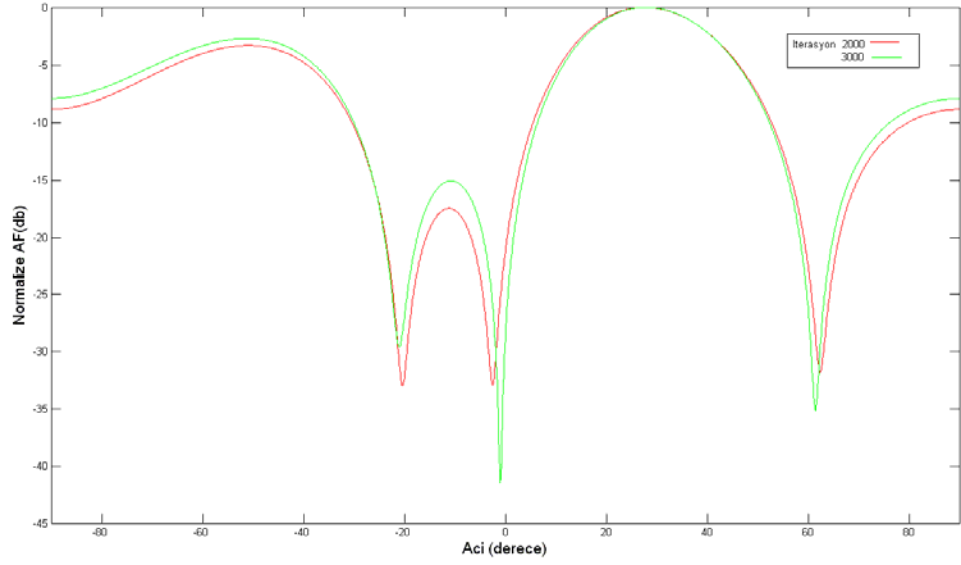
$$y(n) = \hat{\mathbf{w}}^H(n-1) \mathbf{X}(n) \quad (3.44)$$

$$\hat{\mathbf{w}}(n) = \mathbf{P}_c [\hat{\mathbf{w}}(n-1) - \alpha \mathbf{X}(n) y^*(n)] + \mathbf{w}_q \quad (3.45)$$

Bu ifadelerde $X(n)$ anten dizisi tarafından alınan işaret vektörüdür. w_q , (3.42) eşitliği ile tanımlanmaktadır ve Quiescent ağırlık vektörü [26] olarak bilinir; P_c ise sınırlandırma alt uzayına projeksiyon yapan matris olarak dik bir matris olarak düşünülebilir, I birim matrisi belirtmektedir. Frost'un LMS algoritmasında (3.41) ifadesinden de görüldüğü gibi ağırlık katsayıları başlangıç koşulu sıfır değildir. Algoritma referans işaretini kullanmadan istenilen işareti çekmeyi amaçlamaktadır ve sadece istenen işaretin geliş yönü bilgisini kullanmaktadır, bu nedenle algoritma referans işaret bilgisini kullanan LMS algoritmasına göre daha geç yakınsamaktadır. Aşağıda Frost'un LMS algoritması kullanarak elde edilmiş simülasyon sonuçları görülmektedir, istenen işaret geliş açısı 30° , -20° , 0° ve 60° interferans işaretleri geliş açıları.



Şekil 3.21 Ağırlık genliklerinin zamanla değişimi.



Şekil 3.22 Huzme biçimi.

3.4.2 Genelleştirilmiş Yan Lob Engelleyici (GSLC)

LCMV huzme biçimlendiriciyi gerçekleştirmenin bir başka yolu ise M boyutlu uzayı biri sınırlandırma (constraint) alt uzayı ve diğeri buna dik bir alt uzay olmak üzere iki alt uzaya bölmektir. Sınırlandırma alt uzayı $M \times N$ boyutlu C matrisinin sütunlarıyla belirtilirken dik alt uzay ise $M \times (M - N)$ boyutlu B matrisinin sütunlarıyla belirtilir. B matrisinin sütunları birbirleriyle lineer olarak bağımsızdır ve her bir sütun C matrisinin her bir sütununa diktir.

$$C^H B = 0 \quad (3.46)$$

(3.40) eşitliği ile verilen w_{lcmv}^H ağırlık katsayı ifadesi birbirine dik iki ağırlık katsayı vektörüne ayrılabilir;

$$w_{lcmv}^H = w_c^H - w_p^H \quad (3.47)$$

w_c^H , w_{lcmv}^H ağırlık katsayı vektörünün C sınırlandırma matrisi alt uzayına izdüşümü olarak düşünülebilir, aynı şekilde w_p^H de w_{lcmv}^H 'nin B matrisi üzerine izdüşümüdür. Sınırlandırma matrisi üzerine izdüşüm matrisi (3.48) eşitliği ile verilir;

$$P_c = C [C^H C]^{-1} C^H \quad (3.48)$$

$$\mathbf{w}_c^H = \mathbf{w}_{lcmv}^H \mathbf{P}_c \quad (3.49)$$

(3.40) ve (3.48) eşitlikleri (3.49) eşitliğinde yerine konularak (3.50) eşitliği elde edilebilir;

$$\mathbf{w}_c^H = \mathbf{g}^H [\mathbf{C}^H \mathbf{C}]^{-1} \mathbf{C}^H = \mathbf{w}_q^H \quad (3.50)$$

(3.50) ifadesinden w_c 'nin daha önce (3.42) eşitliği ile verilen w_q Quiescent ağırlık vektörüne eşit olduğu görülmektedir.

$$\mathbf{w}_c = \mathbf{w}_q \quad (3.51)$$

w_{lcmv}^H ağırlık katsayı vektörünün B matrisi alt uzayına izdüşüm matrisi ise (3.52) eşitliği ile verilir;

$$\mathbf{P}_b = -\mathbf{B} [\mathbf{B}^H \mathbf{B}]^{-1} \mathbf{B}^H \quad (3.52)$$

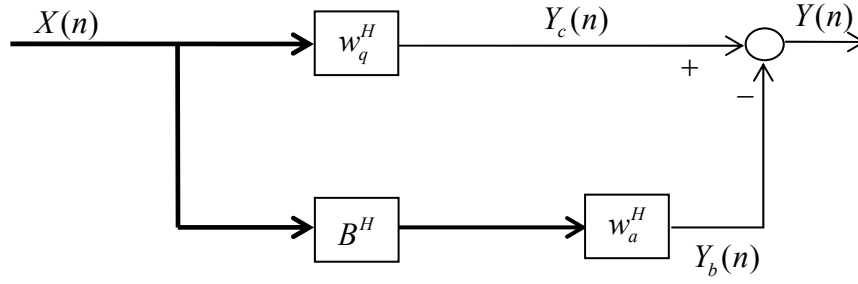
Bu durumda w_p ;

$$\mathbf{w}_p = \mathbf{P}_b \mathbf{w}_{lcmv} \quad (3.53)$$

$$\mathbf{w}_p^H = \mathbf{g}^H [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{C}^H \mathbf{R}^{-1} \mathbf{B} [\mathbf{B}^H \mathbf{B}]^{-1} \mathbf{B}^H \quad (3.54)$$

B matrisi (3.46) ifadesi gözönüne alındığında engelleme matrisi olarak da düşünülebilir çünkü B matrisi C sınırlandırma matrisi alt uzayından herhangi bir bileşen içeremez. B matrisi aynı zamanda tek değildir (3.46) ifadesini sağlayan herhangi bir matris B matrisi olarak kullanılabilir.

Şekil 3.23'de GSLC yapısı görülmektedir $X(n)$ anten dizisi tarafından alınan işaret vektörü w_q^H ağırlık vektörü ile çarpılarak GSLC üst taraf çıkışı oluşturulmaktadır, bu çıkış istenen yönden gelen işareti ve gürültü bileşenini içerir. $X(n)$ vektörü aynı zamanda B^H engelleme matrisi ile çarpılarak GSLC alt taraf çıkışı oluşturur, alt taraf çıkışı istenen işaretle ilgili hiçbir bileşen içermez. w_a^H ağırlık vektörü ise (3.54) ile verilen w_p^H ağırlık vektörüdür ve GSLC yapısının uyarlamalı ağırlık katsayılarını belirtir. w_a^H ağırlık vektörü herhangi bir uyarlamalı algoritma tarafından kontrol edilerek GSLC yapısının üst taraf çıkışındaki gürültü bileşenleri minimize edilebilir.



Şekil 3.23 GSLC yapısı.

Anten çıkışlarına uygulanacak ağırlıklar ise aşağıdaki eşitlikle verilir;

$$\mathbf{w} = \mathbf{w}_q - \mathbf{B}\mathbf{w}_a \quad (3.55)$$

Çıkış gücü ifadesi ise;

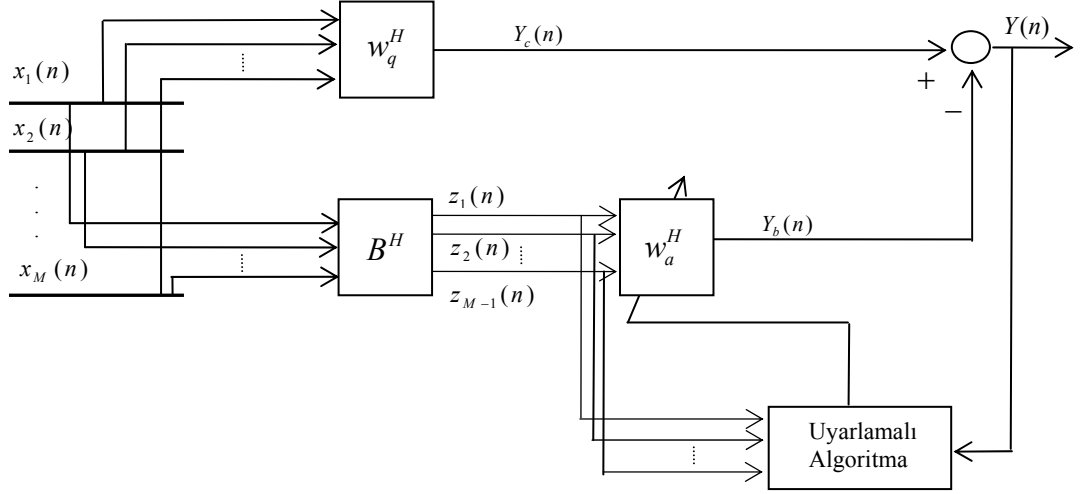
$$P_o = [\mathbf{w}_q - \mathbf{B}\mathbf{w}_a]^H \mathbf{R} [\mathbf{w}_q - \mathbf{B}\mathbf{w}_a] \quad (3.56)$$

Çıkış gücünü minimum yapan w_a katsayılarını bulmak için ise (3.56) ifadesinin w_a 'ya göre gradyanının sıfıra eşitlenmesinden;

$$[\mathbf{w}_q^H - \mathbf{w}_a^H \mathbf{B}^H] \mathbf{R} \mathbf{B} = 0 \quad (3.57)$$

$$\hat{\mathbf{w}}_a^H = \mathbf{w}_q^H \mathbf{R} \mathbf{B} [\mathbf{B}^H \mathbf{R} \mathbf{B}]^{-1} \quad (3.58)$$

En basit durum için C sınırlandırma matrisi istenen işaret için yönlendirme vektörüne eşit alınabilir. Bu durumda B^H matrisi ile çarpılmış GSLC alt taraf çıkışı M anten sayısı olmak üzere $M - 1$ boyutlu olur. Böyle bir uyarlamalı GSLC yapısı Şekil 3.24'de görülmektedir. Uyarlamalı algoritma olarak LMS ve RLS algoritmalarının kullanıldığı GSLC huzme biçimlendirme simülasyon sonuçları bir sonraki bölümde sunulacaktır.



Şekil 3.24 Uyarlamalı GSLC yapısı.

3.4.3 Simülasyon sonuçları

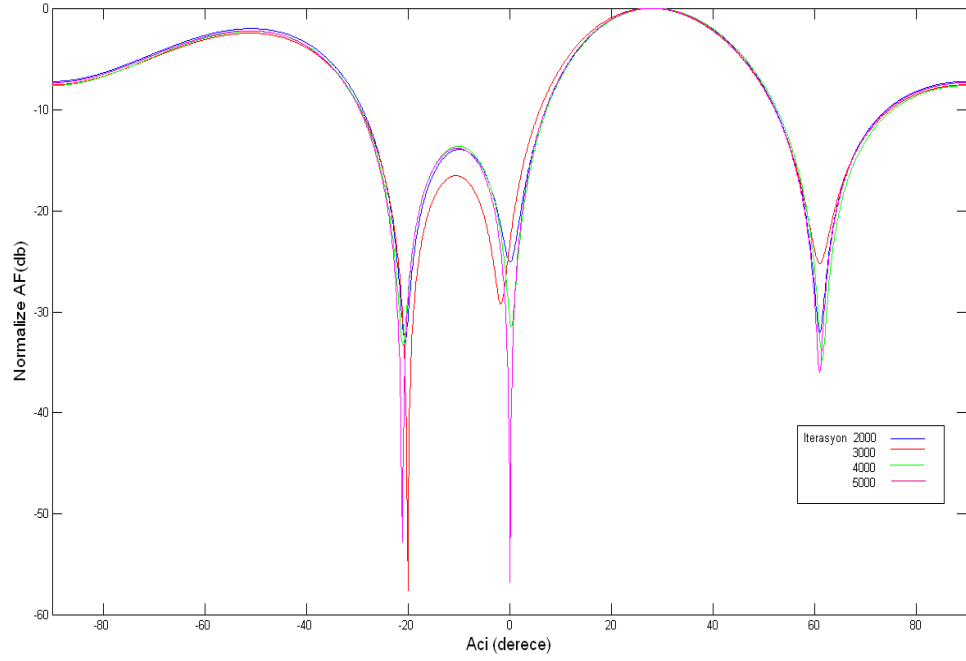
Simülasyon 1:

İstenen s_o işaretin geliş açısının 30° , -20° , 0° ve 60° geliş açılarında sahip 3 tane interferans kaynağının olduğu varsayılmaktadır.

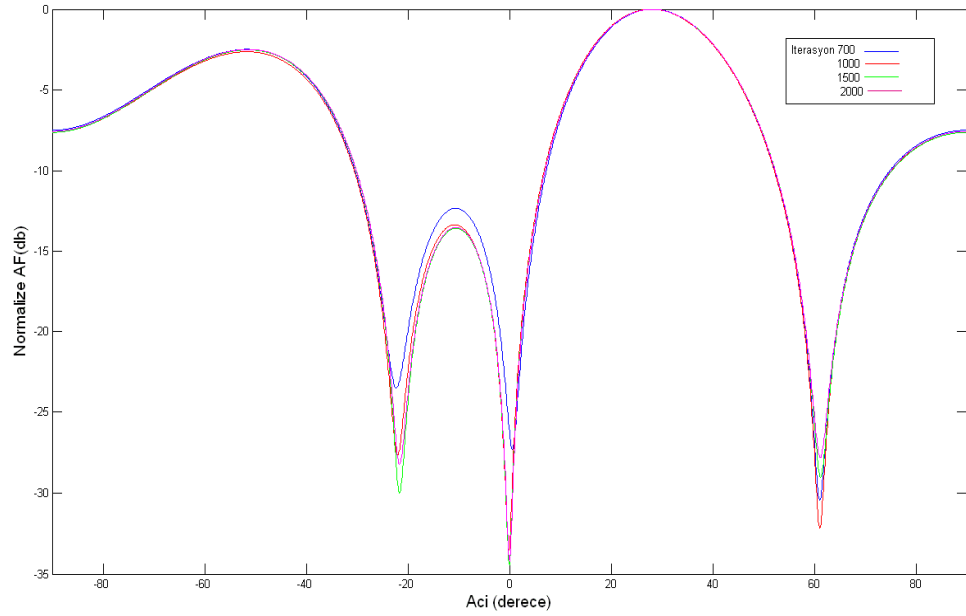
Şekil 3.25’de LMS algoritmasıyla gerçekleştirilmiş GSLC huzme biçimleri görülmektedir. LMS algoritması adım boyu $\alpha = 0.001$ olarak alınmıştır. LMS algoritması yavaş yakınsayan bir algoritma olduğundan düşük iterasyonlar için iyi huzme biçimleri vermemiştir. Ancak 2000 iterasyondan itibaren algoritma tüm interferans açılarını hatasız yakalamayı başarmaktadır ve iterasyon sayısı arttıkça interferans yönlerinde daha derin sıfırlar koymaya çalıştığı görülmektedir, 5000. iterasyon için algoritma iki interferans işaretini yaklaşık 60 dB aşağı çekmeyi başarmaktadır.

Şekil 3.26’da ise RLS algoritmasıyla gerçekleştirilmiş GSLC huzme biçimleri görülmektedir. RLS algoritması çok hızlı bir algoritma olduğundan interferans işaretleri açılarını daha kısa sürede yakalamıştır. Değişik iterasyon sayılarında elde edilen ağırlık katsayılarıyla çizdirilen huzme biçimleri arasında fazla bir fark görülmemektedir, iterasyon sayısı arttıkça interferans yönlerindeki işaretleri biraz daha aşağıya çekmektedir fakat hepsinde açılar hemen hemen aynıdır. 3000 ve 4000. iterasyondaki huzme biçimleri ise Şekil 3.27’de görülmektedir; interferans işaretleri

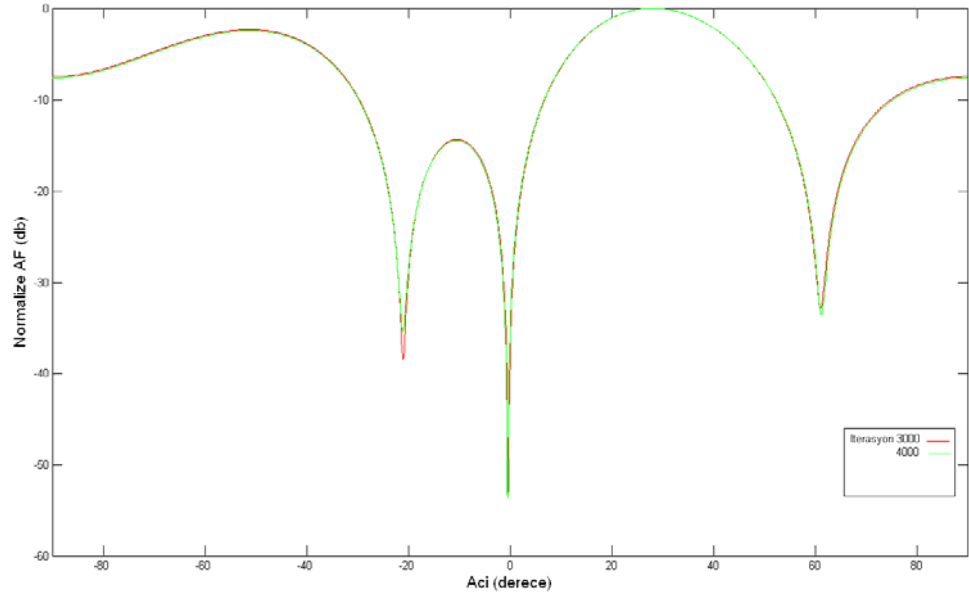
daha da aŖađıya bastırılmıŖtır, bu iterasyon sayılarından daha yukarı ıkıldığında ise huzme biimleri hemen hemen aynı kalmaktadır.



Ŗekil 3.25 LMS-GSLC huzme biimi.



Ŗekil 3.26 RLS-GSLC huzme biimi (a).



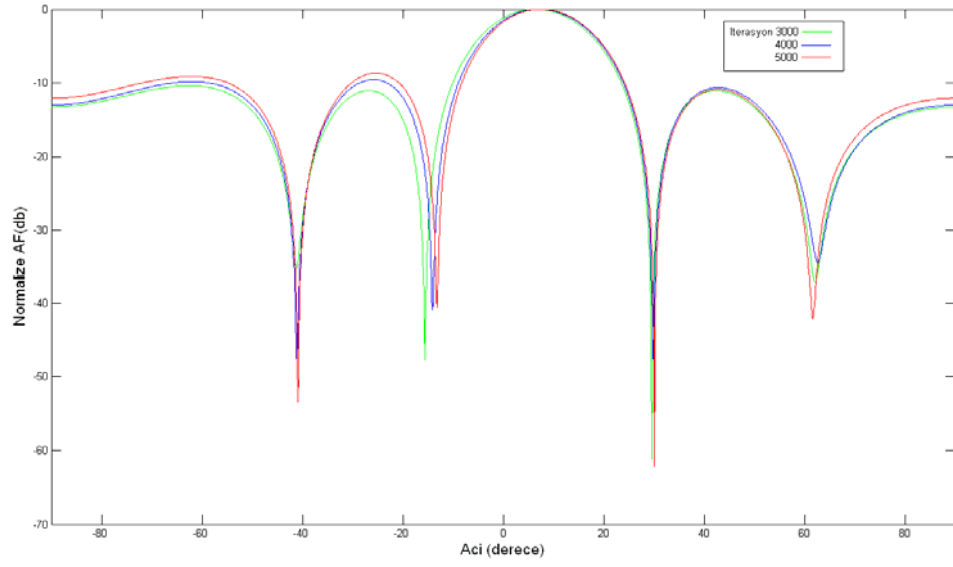
Şekil 3.27 RLS-GSLC huzme biçimi (b).

Simülasyon 2:

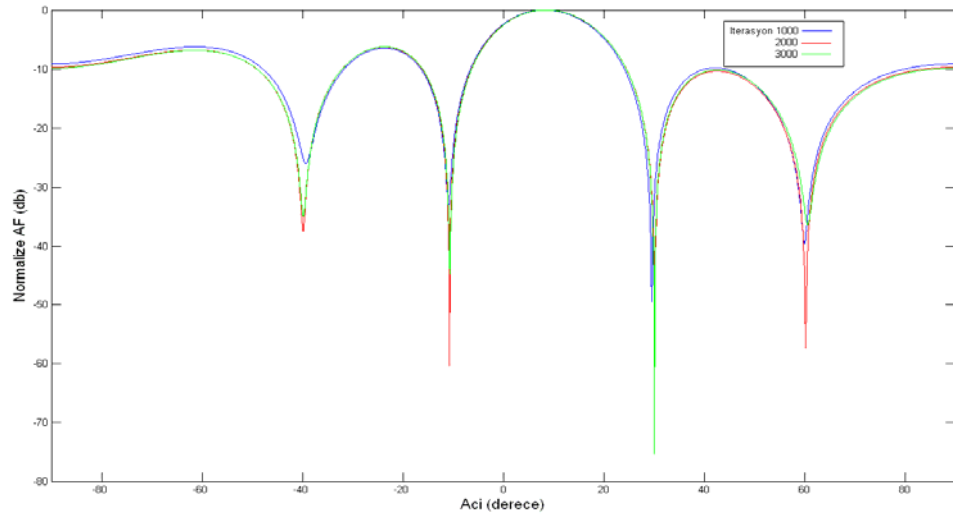
İstenen s_o işaretin geliş açısı 0° ve -40° -10° 30° ve 60° geliş açılarna sahip 4 tane interferans işaret kaynağı olduğu varsayılmaktadır.

Şekil 3.28'deki LMS-GSLC huzme biçiminden görüldüğü gibi iterasyon sayısı arttıkça interferans işaretleri oldukça iyi yakalanmaktadır ve oldukça derin sıfırlar konulmaktadır.

Şekil 3.29'dan görüldüğü gibi RLS algoritması yine çok hızlı yakınsamıştır ve LMS algoritmasına göre çok düşük iterasyonlarda çok iyi huzme biçimleri vermiştir.



Şekil 3.28 LMS-GSLC huzme biçimi.



Şekil 3.29 RLS-GSLC huzme biçimi.

4. KAFES (LATTICE) KESTİRİCİLERLE HUZME BİÇİMLENDİRME

Kafes kestiriciler bir çok avantajı beraberinde getirmektedir; modüler yapıya sahip olması, filtre karakteristiği analizini kolaylaştırması bu avantajlar arasında sayılabilir. Kafes kestiricilerin uyarlamalı işaret işlemeye uygulanması ilk olarak 1971 yılında Itakura ve Saito tarafından yapılmıştır [11]. Kafes kestiriciler jeofizik işaret işleme ve ağ (network) teorisinde de oldukça yaygın kullanıma sahiptir.

Bu bölümde ilk önce klasik Kafes kestirici yapısı anlatılmakta, daha sonra anten dizileriyle işaret işlemeye uygulanmış bir Kafes algoritması üzerinde durulmakta ve bu algoritmayla elde edilmiş huzme biçimlendirme simülasyonları sunulmaktadır. Alternatif olarak bu Kafes algoritması bir önceki bölümde anlatılan Genelleştirilmiş Yan Lob Engelleyici yapısının uyarlamalı kısmında kullanılıp simülasyon sonuçları değerlendirilmektedir.

4.1 İleri ve Geri Kestirici Hataları

Alınan gözlem işaret dizisinde, belirli uzunluktaki gözlem verisini kullanarak bir sonraki veriyi kestirmeye ileri yönde kestirim denir. Aynı şekilde belirli uzunluktaki gözlem dizisini kullanarak geçmişteki bir veriyi kestirmeye de geri yönde kestirim denir. Geri yönde kestirimin uygulamada herhangi bir anlamı olmasa da geri yönde kestirim hataları ile ileri yönde kestirim hataları arasında bir ilişki mevcuttur ve bu yapı işaret işlemede büyük avantajlar sunmaktadır.

M uzunluklu bir $x(n-1)$ gözlem vektörü aşağıdaki gibi tanımlanmaktadır. Bu durumda ileri kestirim ve ileri kestirim hata ifadeleri sırasıyla (4.1) ve (4.2) ile verilmektedir.

$$\mathbf{x}(n-1) = \begin{bmatrix} x(n-1) \\ x(n-2) \\ \cdot \\ \cdot \\ x(n-M) \end{bmatrix} \quad (4.1)$$

$$\hat{x}(n) = \sum_{k=1}^M w_{f,k}^* x(n-k) \quad (4.2)$$

$$f_M(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^M w_{f,k}^* x(n-k) \quad (4.3)$$

(4.3) ifadesi daha düzgün bir biçimde yazılabilir:

$$f_M(n) = x(n) - \sum_{k=1}^M w_{f,k}^* x(n-k) = \sum_{k=0}^M a_{M,k}^* x(n-k) \quad (4.4)$$

$$a_{M,k} = \begin{cases} 1, & k = 0 \\ -w_{f,k} & k = 1, 2, \dots, M \end{cases} \quad (4.5)$$

w_f ileri kestirim katsayıları Wiener-Hopf denklemleriyle bulunabilir:

$$\mathbf{w}_f = \begin{bmatrix} w_{f,1} \\ w_{f,2} \\ \cdot \\ \cdot \\ w_{f,M} \end{bmatrix} \quad (4.6)$$

$$\mathbf{R} = E[\mathbf{x}(n-1)\mathbf{x}^H(n-1)] = \begin{bmatrix} r(0) & r(1) & \cdot & \cdot & \cdot & r(M-1) \\ r^*(1) & r(0) & \cdot & \cdot & \cdot & r(M-2) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ r^*(M-1) & r^*(M-2) & \cdot & \cdot & \cdot & r(0) \end{bmatrix} \quad (4.7)$$

$$\mathbf{r} = E[\mathbf{x}(n-1)x^*(n)] = \begin{bmatrix} r^*(1) \\ r^*(2) \\ \cdot \\ \cdot \\ r^*(M) \end{bmatrix} \quad (4.8)$$

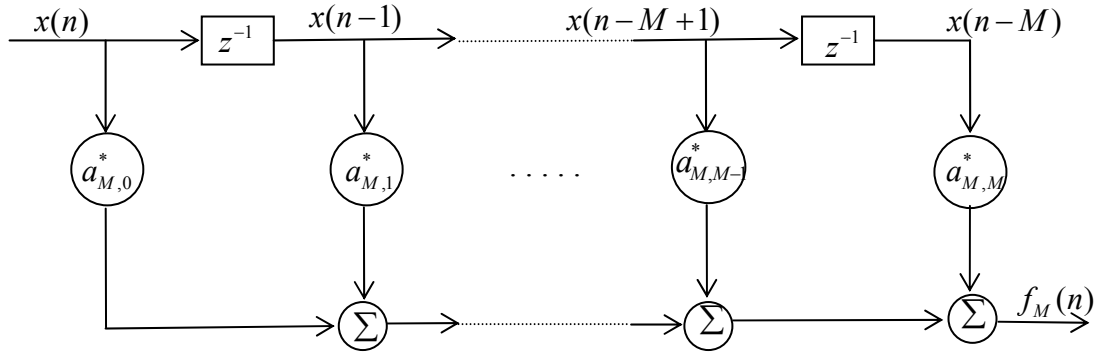
$$\mathbf{R}\mathbf{w}_f = \mathbf{r} \Rightarrow \mathbf{w}_f = \mathbf{R}^{-1}\mathbf{r} \quad (4.9)$$

$$\mathbf{P}_{fM} = r(0) - \mathbf{r}^H \mathbf{w}_f \quad (4.10)$$

(4.9) ve (4.10) ifadeleri tek bir eşitlikte birleştirilebilir:

$$\begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_f \end{bmatrix} = \begin{bmatrix} P_{fM} \\ 0 \end{bmatrix} \quad (4.11)$$

(4.11) ifadesinde $0, M \times 1$ boyutlu sıfır vektörünü; $R, x(n-1)$ vektörünün korelasyon matrisini, r ise $x(n-1)$ vektörü ile $x(n)$ işareti arasındaki ilişki vektörünü, P_{fM} ileri kestirim hata gücünü belirtir.



Şekil 4.1 İleri kestirici hata filtresi.

Aynı şekilde geri kestirim ve geri kestirim hata ifadeleri şu şekilde verilir:

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-M+1) \end{bmatrix} \quad (4.12)$$

$$\hat{x}(n-M) = \sum_{k=1}^M w_{b,k}^* x(n-k+1) \quad (4.13)$$

$$b_M(n) = x(n-M) - \hat{x}(n-M) = x(n-M) - \sum_{k=1}^M w_{b,k}^* x(n-k+1) \quad (4.14)$$

$$b_M(n) = \sum_{k=0}^M c_{M,k}^* x(n-k) \quad (4.15)$$

$$c_{M,k} = \begin{cases} 1, & k = M \\ -w_{b,k+1}^* & k = 0, 1, \dots, M-1 \end{cases} \quad (4.16)$$

$f_M(n)$ ve $b_M(n)$ sırasıyla ileri ve geri kestirici hataları, $a_{M,k}$ ve $c_{M,k}$ ise sırasıyla ileri ve geri kestirici filtre katsayılarıdır; çünkü $f_M(n)$ ve $b_M(n)$ sırasıyla $x(n)$ ve $x(n-1)$ giriş vektörlerinin filtre çıkışı olarak düşünülebilir. Bu durumda filtre katsayıları sırasıyla $a_{M,k}$ ve $c_{M,k}$ olur. Bu ifadelerdeki M indisi filtre uzunluğunu belirtmektedir.

w_b geri kestirim katsayılarını bulmak için Wiener-Hopf eşitliği kullanılırsa;

$$\mathbf{r}^B = \begin{bmatrix} r^*(M) \\ r^*(M-1) \\ \cdot \\ \cdot \\ r^*(1) \end{bmatrix} \Rightarrow \mathbf{r}^{B*} = \begin{bmatrix} r(M) \\ r(M-1) \\ \cdot \\ \cdot \\ r(1) \end{bmatrix} \quad (4.17)$$

$$\mathbf{r} = E[\mathbf{x}(n)x^*(n-M)] = \begin{bmatrix} r(M) \\ r(M-1) \\ \cdot \\ \cdot \\ r(1) \end{bmatrix} = \mathbf{r}^{B*} \quad (4.18)$$

$$\mathbf{R} \mathbf{w}_b = \mathbf{r} \Rightarrow \mathbf{R} \mathbf{w}_b = \mathbf{r}^{B*} \quad (4.19)$$

$$P_{bM} = r(0) - [\mathbf{r}^B]^T \mathbf{w}_b \quad (4.20)$$

R korelasyon matrisi ileri ve geri kestirim için aynıdır, r ilişki vektörü ise geri kestirim için \mathbf{r}^{B*} halini almıştır, P_{bM} ise geri kestirim hata gücüdür.

(4.19) ve (4.20) ifadeleri aynı şekilde tek bir eşitlikle gösterilebilir:

$$\begin{bmatrix} \mathbf{R} & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} -\mathbf{w}_b \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ P_{bM} \end{bmatrix} \quad (4.21)$$

Aynı zamanda ortalama ileri ve geri kestirim hata güçlerinin tanımlanması faydalı olacaktır:

$$P_{fM} = E[|f_M(n)|^2] \quad (4.22)$$

$$P_{bM} = E[|b_M(n)|^2] \quad (4.23)$$

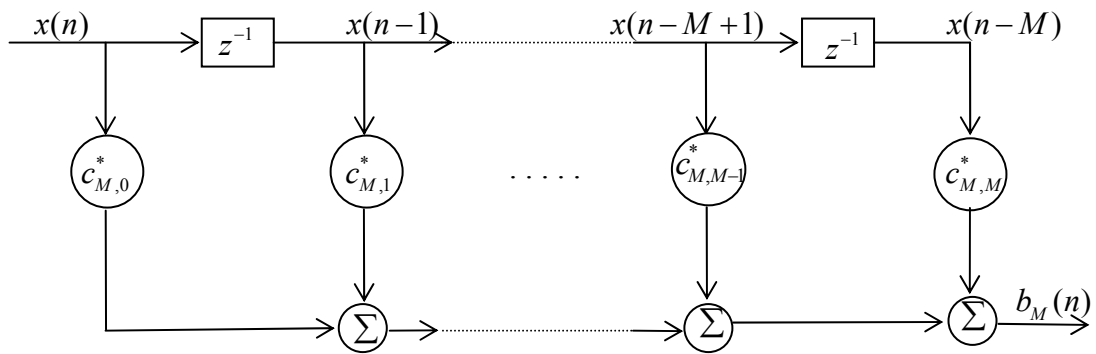
İleri ve geri kestirim ifadelerinden görüldüğü gibi her iki durum için giriş R korelasyon matrisi aynıdır. Bu durum ileri ve geri kestirim hata filtre katsayıları arasında bir ilişkiyi de beraberinde getirmektedir.

$$c_{M,k} = a_{M,M-k}^* \quad k = 0, 1, \dots, M \quad (4.24)$$

$$\mathbf{a}_{M,k} = \begin{bmatrix} 1 \\ a_{M,1} \\ a_{M,2} \\ \vdots \\ a_{M,M} \end{bmatrix} \Rightarrow \mathbf{c}_{M,k} = \begin{bmatrix} a_{M,M}^* \\ a_{M,M-1}^* \\ \vdots \\ 1 \end{bmatrix} \quad (4.25)$$

(4.25) ifadesi kullanılarak aynı zamanda ortalama ileri ve geri kestirim hata güçlerinin eşit olduğu gösterilebilir:

$$P_{fM} = P_{bM} = P_M \quad (4.26)$$



Şekil 4.2 Geri kestirici hata filtresi.

$b_m(n)$ her bir seviyedeki geri kestirim hatalarının $m = 0, 1, \dots, M$, m uzunluklu filtre çıkışları olduğu düşünülürse;

$$b_0(n) = x(n)$$

$$b_1(n) = a_{1,1}x(n) + a_{1,0}x(n-1)$$

$$b_2(n) = a_{2,2}x(n) + a_{2,1}x(n-1) + a_{2,0}x(n-2)$$

·
·

$$b_M(n) = a_{M,M}x(n) + a_{M,M-1}x(n-1) + \dots + a_{M,0}x(n-M)$$

$$\mathbf{b}(n) = \begin{bmatrix} b_0(n) \\ b_1(n) \\ \cdot \\ \cdot \\ b_M(n) \end{bmatrix} \text{ ve } \mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \cdot \\ \cdot \\ x(n-M) \end{bmatrix} \text{ olmak üzere;}$$

$$\mathbf{b}(n) = \mathbf{L} \mathbf{x}(n) \quad (4.27)$$

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & 0 \\ a_{1,1} & 1 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{M,M} & a_{M,M-1} & \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad (4.28)$$

(4.27) ifadesi Gram-Schmidt dikleştirme algoritması olarak bilinir. Bu algoritmaya göre birbirleriyle korelasyonlu $x(n)$ giriş vektörü elemanları birbirine dik $b(n)$ geri kestirim hatalarına dönüştürülür; $x(n)$ vektörü ile $b(n)$ vektörü aynı bilgiyi içermektedir.

L matrisinin satırları her bir seviyedeki geri kestirim hata filtresi katsayılarından oluşmaktadır. Ana diyagonal elemanları 1'dir ve ana diyagonalin üstündeki tüm elemanlar 0'a eşittir. Her bir seviyedeki geri kestirim hatalarının korelasyon matris ifadesi ise aşağıdaki ifadeyle verilebilir;

$$\mathbf{D} = E[\mathbf{b}(n)\mathbf{b}^H(n)] \quad (4.29)$$

$$\mathbf{D} = E[\mathbf{b}(n)\mathbf{b}^H(n)] = E[\mathbf{L}\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{L}^H] = \mathbf{L} E[\mathbf{x}(n)\mathbf{x}^H(n)] \mathbf{L}^H = \mathbf{L}\mathbf{R}\mathbf{L}^H \quad (4.30)$$

(4.30) ifadesindeki R terimi $x(n)$ giriş vektörünün korelasyon matrisidir. L matrisi alt diyagonal bir matris ve L^H matrisi ise üst diyagonal bir matris olacağından (4.30) ifadesi diyagonal bir matrise eşit olacaktır ve (4.31) eşitliği ile verilir;

$$\mathbf{D} = \text{diag}(P_0, P_1, \dots, P_M) \quad (4.31)$$

(4.31) ifadesindeki P_i terimi $i = 0, 1, \dots, M$ her bir seviyedeki geri kestirim ortalama hata gücünü belirtmektedir.

4.2 Kafes Kestiriciler

Kafes kestirici ifadelerini oluşturmada önce giriş vektörü parçalı şekilde ifade edilirse;

$$1. \mathbf{x}_{m+1}(n) = \begin{bmatrix} \mathbf{x}_m(n) \\ x(n-m) \end{bmatrix} \quad ; \quad \mathbf{x}_m(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ \cdot \\ \cdot \\ x(n-m+1) \end{bmatrix}$$

$$2. \mathbf{x}_{m+1}(n) = \begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix} \quad ; \quad \mathbf{x}_m(n-1) = \begin{bmatrix} x(n-1) \\ x(n-2) \\ \cdot \\ \cdot \\ x(n-m) \end{bmatrix}$$

Farklı derecelerdeki hata katsayıları arasında bir ilişki vardır ve bir seviyedeki katsayılarından diğer bir seviye katsayılarına Levinson-Durbin algoritmasıyla geçilebilir. Bu algoritmaya göre katsayılar arasındaki ilişki aşağıdaki ifadeyle verilir;

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \quad (4.32)$$

(4.32) ifadesinde κ_m bir sabittir ve yansıma katsayısı olarak isimlendirilir. \mathbf{a}_m ifadesinin hermityen transpozesiyle $\mathbf{x}_{m+1}(n)$ vektörü çarpılırsa;

$$\mathbf{a}_m^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} \mathbf{a}_{m-1}^H \\ 0 \end{bmatrix}^H \mathbf{x}_{m+1}(n) + \kappa_m^* \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}^H \mathbf{x}_{m+1}(n) \quad (4.33)$$

$$\mathbf{a}_m^H \mathbf{x}_{m+1}(n) = f_m(n) \quad (4.34)$$

$$\begin{bmatrix} \mathbf{a}_{m-1}^H \\ 0 \end{bmatrix}^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} \mathbf{a}_{m-1}^H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_m(n) \\ x(n-m) \end{bmatrix} = \mathbf{a}_{m-1}^H \mathbf{x}_m(n) = f_{m-1}(n) \quad (4.35)$$

$$\begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} 0 & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix} = \mathbf{a}_{m-1}^{BT} \mathbf{x}_m(n-1) = b_{m-1}(n-1) \quad (4.36)$$

(4.34), (4.35), (4.36) ifadeleri birleştirilirse (4.33) ifadesi aşağıdaki gibi verilir;

$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1) \quad (4.37)$$

Geri kestirim hatası derece güncelleme ifadesini elde edebilmek için ise (4.32) ifadesinin konjügesi alınıp vektörlerin ters çevrilmesi kolaylık sağlayacaktır;

$$\mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + \kappa_m^* \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} \quad (4.38)$$

Yine aynı şekilde (4.38) ifadesinin hermityen transpozesi alınıp $\mathbf{x}_{m+1}(n)$ giriş vektörü ile çarpılırsa;

$$\left(\mathbf{a}_m^{B*}\right)^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}^H \mathbf{x}_{m+1}(n) + \kappa_m \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix}^H \mathbf{x}_{m+1}(n) \quad (4.39)$$

$$\left(\mathbf{a}_m^{B*}\right)^H \mathbf{x}_{m+1}(n) = \mathbf{a}_m^{BT} \mathbf{x}_{m+1}(n) = b_m(n) \quad (4.40)$$

$$\begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} 0 & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} x(n) \\ \mathbf{x}_m(n-1) \end{bmatrix} = \mathbf{a}_{m-1}^{BT} \mathbf{x}_m(n-1) = b_{m-1}(n-1) \quad (4.41)$$

$$\begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix}^H \mathbf{x}_{m+1}(n) = \begin{bmatrix} \mathbf{a}_{m-1}^H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_m(n) \\ x(n-m) \end{bmatrix} = \mathbf{a}_{m-1}^H \mathbf{x}_m(n) = f_{m-1}(n) \quad (4.42)$$

(4.40), (4.41) ve (4.42) ifadeleri birleştirilirse (4.39) ifadesi aşağıdaki gibi verilir;

$$b_m(n) = b_{m-1}(n-1) + \kappa_m f_{m-1}(n) \quad (4.43)$$

İleri ve geri kestirim hataları güncelleme ifadeleri tek bir eşitlikte birleştirilebilir;

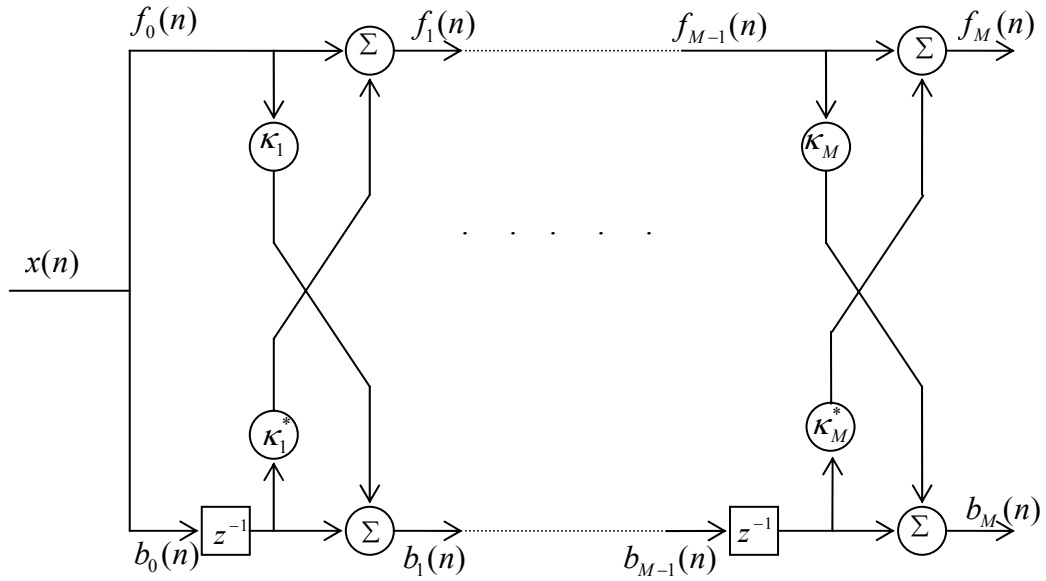
$$\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \kappa_m^* \\ \kappa_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n-1) \end{bmatrix} \quad m = 1, 2, \dots, M \quad (4.44)$$

Başlangıç koşulu için ise $m = 0$;

$$f_0(n) = b_0(n) = x(n) \quad (4.45)$$

Şekil 4.3'de Kafes kestirici hata akış diyagramı görülmektedir. Bu yapı ve Kafes ifadeleri göz önüne alındığında şu sonuçlar elde edilebilir: Kafes kestirici yapısı sürekli olarak ileri ve geri kestirici hatalarını eş zamanlı olarak üretmektedir; her bir kafes seviyesi birbiriyle ilişkisizdir çünkü geri kestirim hataları geniş anlamda durağan giriş işaretleri için birbirine diktir; kafes yapısı modülerdir, kestirim seviyesi

artırılmak istenildiği zaman önceki seviyelerdeki hesaplamalar değişikliğe uğramadan artırılabilir. Bu yönleriyle kafes yapısı bir çok avantaj sunmaktadır.



Şekil 4.3 İleri ve geri kestirim hataları işaret akış diyagramı.

4.3 Ortak Süreç Kestirimi

Kafes yapısı ileri ve geri kestirim hatalarını üretirken aynı zamanda $x(n)$ giriş işaretinden istenen bir $d(n)$ işaretinin kestirimini yapabilir. Bu yapı ortak süreç kestirimi olarak bilinmektedir. $d(n)$ işaretinin kestirimi geri kestirim hataları kullanılarak yapılmaktadır, geri kestirim hataları birbirine dik olduklarından ortak süreç yapısı daha kolay bir çözüm sunmaktadır.

Ortak süreç yapısı işaret akış biçimi Şekil 4.4'de görülmektedir. Birbiriyle korelasyonlu $x(n)$ giriş işareti birbirlerine dik $b_m(n)$ geri kestirim hatalarına dönüştürüldükten sonra $h_m(n)$ ortak süreç kestirim katsayılarıyla çarpılarak $\hat{d}(n)$ istenen işaret kestirimi elde edilmektedir.

$$\mathbf{b}(n) = \begin{bmatrix} b_0(n) \\ b_1(n) \\ \vdots \\ b_M(n) \end{bmatrix} \quad \text{ve} \quad \mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_M \end{bmatrix} \quad \text{olmak üzere;}$$

$$\hat{d}(n) = \mathbf{h}^H \mathbf{b}(n) \quad (4.46)$$

$$\mathbf{D} = E[\mathbf{b}(n)\mathbf{b}^H(n)] \quad (4.47)$$

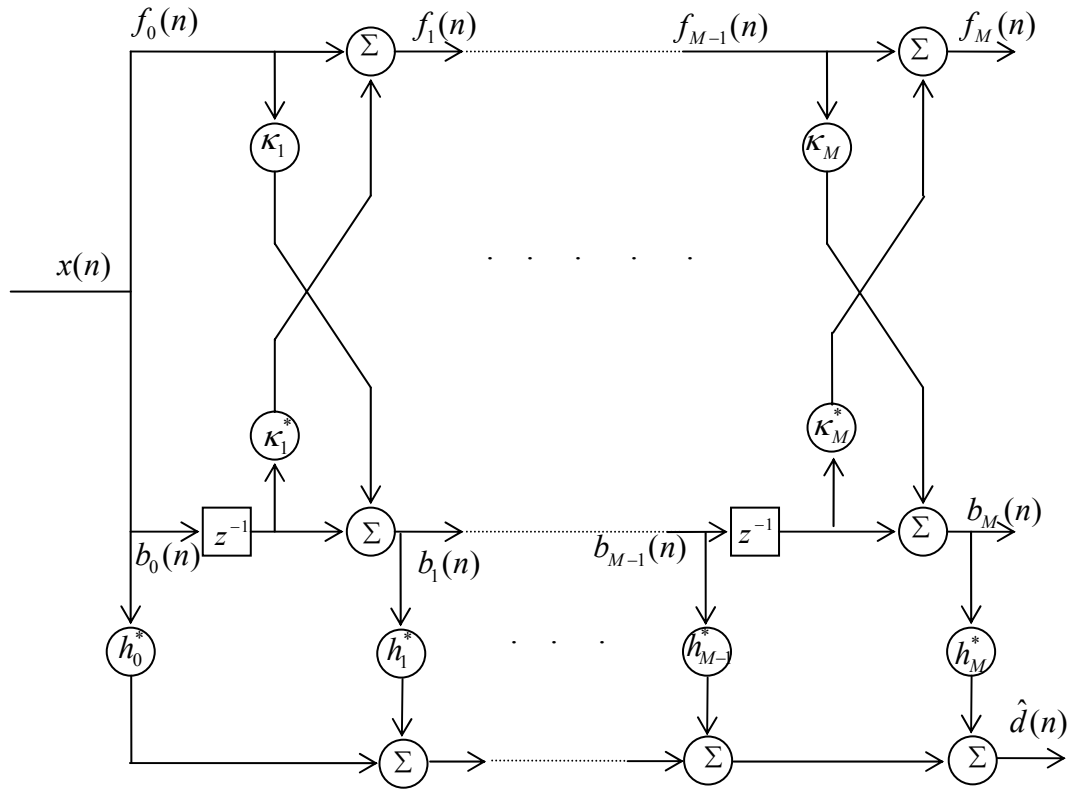
$$\mathbf{z} = E[\mathbf{b}(n)d^*(n)] \quad (4.48)$$

(4.47) ifadesinde D geri kestirim hatalarının korelasyon matrisini göstermektedir, (4.48) ifadesinde z ise istenen $d(n)$ işaretiyle geri kestirim hataları arasındaki çapraz ilişki vektörüdür. h ortak süreç kestirim katsayılarını bulmak için Wiener-Hopf eşitliği kullanılırsa;

$$\mathbf{D}\mathbf{h} = \mathbf{z} \Rightarrow \mathbf{h} = \mathbf{D}^{-1}\mathbf{z} \quad (4.49)$$

(4.30) ifadesinden D matrisinin tersine geçilirse;

$$\mathbf{D}^{-1} = \mathbf{L}^{-H}\mathbf{R}^{-1}\mathbf{L}^{-1} \quad (4.50)$$



Şekil 4.4 Ortak süreç yapısı işaret akış diyagramı.

(4.50) ifadesi (4.49)'da ve (4.27) ifadesi (4.48)'de yerine yazılırsa;

$$\mathbf{h} = \mathbf{L}^{-H}\mathbf{R}^{-1}\mathbf{L}^{-1}\mathbf{z} \quad (4.51)$$

$$\mathbf{z} = E[\mathbf{L} \mathbf{x}(n) d^*(n)] = \mathbf{L} E[\mathbf{x}(n) d^*(n)] = \mathbf{L} \mathbf{r} \quad (4.52)$$

(4.52) ifadesindeki \mathbf{r} , $\mathbf{x}(n)$ giriş vektörü ile istenen $d(n)$ işareti arasındaki çapraz ilişki vektörünü belirtmektedir. Son olarak (4.52) ifadesi (4.51)'de yerine konulduğunda istenen Wiener çözüm katsayıları bulunmuş olur;

$$\mathbf{h} = \mathbf{L}^{-H} \mathbf{R}^{-1} \mathbf{L}^{-1} \mathbf{z} = \mathbf{L}^{-H} \mathbf{R}^{-1} \mathbf{L}^{-1} \mathbf{L} \mathbf{r} = \mathbf{L}^{-H} \mathbf{R}^{-1} \mathbf{r} = \mathbf{L}^{-H} \mathbf{w} \Rightarrow \mathbf{w} = \mathbf{L}^H \mathbf{h} \quad (4.53)$$

Ortak süreç kestirimini uyarlamalı olarak yapmak için bir çok algoritma mevcuttur. Bu algoritmalar arasında gradyan tabanlı GAL algoritması RLS tabanlı hata geri beslemeli hızlı algoritmalar sayılabilir; ayrıca zaman güncellemesini yapmak için LMS algoritması da Kafes kestiricinin içinde kullanılabilir [11-19].

4.4 Kafes Yapısının Anten Dizilerine Uygulanması

Şu ana kadar anlatılan kafes yapısı gecikmeli işaretler için geçerlidir, yani bu yapıda belirli bir anda gelen işaret önceki işaretlerle beraber değerlendirilir, yeni giriş vektörü bir önceki andaki giriş vektörü verilerini de içermektedir. Oysa ki anten dizileriyle alınan belirli bir andaki işaret vektörüyle bir önceki andaki işaret vektörü arasında böyle bir ilişki yoktur. Dolayısıyla kafes yapısının anten dizilerine uygulanabilmesi için bazı değişiklikler yapılması gerekmektedir. 2002 yılında Haidong Wu ve Jiangzhou Wang uyarlamalı anten dizileri için böyle bir kafes yapısını sunmuştur [40], bu bölümde bu Kafes algoritması anlatılacak algoritmanın performansı simülasyonlarla değerlendirilecektir.

4.4.1 Kafes algoritmasıyla huzme biçimlendirme

M antenden oluşmuş bir anten dizisini düşünelim. $x_m(n)$ $m = 1, 2, \dots, M$, her bir antenin aldığı işaret olmak üzere $\mathbf{X}_m(n)$ vektörü aşağıdaki gibi tanımlanmaktadır;

$$\mathbf{X}_m(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_m(n) \end{bmatrix} \quad (4.54)$$

$a_{m,k}$ ve $c_{m,k}$ sırasıyla ileri ve geri kestirim hata katsayıları olmak üzere $k = 0, 1, \dots, m$, $m = 1, 2, \dots, M$; $A_m(n)$ ve $C_m(n)$ her bir an için her bir seviyedeki ileri ve geri kestirim hata katsayı vektörleri şu şekilde tanımlanır;

$$\mathbf{A}_m(n) = \begin{bmatrix} 1 \\ a_{m,2}(n) \\ \cdot \\ \cdot \\ a_{m,m}(n) \end{bmatrix} \quad (4.55)$$

$$\mathbf{C}_m(n) = \begin{bmatrix} c_{m,1}(n) \\ c_{m,2}(n) \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \quad (4.56)$$

$$a_{m,0} = 1, \quad c_{m,m} = 1 \quad m = 1, 2, \dots, M \quad (4.57)$$

$f_m(n)$ ve $b_m(n)$ her bir andaki her bir seviyedeki ileri ve geri kestirim hataları ise;

$$f_m(n) = x_1(n) - \sum_{k=2}^m a_{m,k}^*(n-1) x_k(n) \quad (4.58)$$

$$b_m(n) = x_m(n) - \sum_{k=1}^{m-1} c_{m,k}^*(n-1) x_k(n) \quad (4.59)$$

$g_m(n)$ ise $f_m(n)$ ve $b_m(n)$ için derece güncelleme ifadelerinde kullanılacaktır ve aşağıdaki gibi tanımlanmaktadır;

$$\mathbf{g}_m(n) = \begin{bmatrix} 0 \\ \mathbf{C}_{m-1}(n-1) \end{bmatrix}^H \mathbf{X}_m(n) \quad (4.60)$$

$$f_m(n) = f_{m-1}(n) + \kappa_m^{f*}(n-1) g_m(n) \quad (4.61)$$

$$b_m(n) = g_m(n) + \kappa_m^{b*} f_{m-1}(n) \quad (4.62)$$

$f_m(n)$ ve $b_m(n)$ için derece güncelleme ifadeleri (4.61) ve (4.62) eşitlikleriyle verilmektedir, bu eşitliklerdeki κ_m^f ve κ_m^b birer skalerdir; (4.63) ve (4.64) ile verilen ileri ve geri hata maliyet fonksiyonlarını minimize edecek şekilde tanımlanırlar;

$$\xi_m^f(n) = \sum_{k=1}^n \lambda^{n-k} |f_m(k)|^2 = |f_m(n)|^2 + \lambda \xi_m^f(n-1) \quad (4.63)$$

$$\xi_m^b(n) = \sum_{k=1}^n \lambda^{n-k} |b_m(k)|^2 = |b_m(n)|^2 + \lambda \xi_m^b(n-1) \quad (4.64)$$

$$\kappa_m^f(n) = - \frac{\sum_{k=1}^n \lambda^{n-k} f_{m-1}^*(k) g_m(k)}{\sum_{k=1}^n \lambda^{n-k} |g_m(k)|^2} \quad (4.65)$$

$$\kappa_m^b(n) = - \frac{\sum_{k=1}^n \lambda^{n-k} g_m^*(k) f_{m-1}(k)}{\sum_{k=1}^n \lambda^{n-k} |f_{m-1}(k)|^2} \quad (4.66)$$

Bu ifadelerdeki λ parametresi ağırlıklandırma faktörüdür ve pozitif bir sabittir, genellikle 1'e çok yakın olarak seçilir. (4.58) ve (4.59) $f_m(n)$ ve $b_m(n)$ tanım ifadeleri ve (4.60) $g_m(n)$ tanım ifadesi (4.61) ve (4.62) $f_m(n)$ ve $b_m(n)$ derece güncelleme ifadelerinde yerine konulursa ileri ve geri hata kestirim katsayıları için Levinson-Durbin algoritmasındaki gibi derece güncelleme ifadeleri elde edilir;

$$\mathbf{A}_m(n-1) = \begin{bmatrix} \mathbf{A}_{m-1}(n-1) \\ 0 \end{bmatrix} + \kappa_m^f(n-1) \begin{bmatrix} 0 \\ \mathbf{C}_{m-1}(n-1) \end{bmatrix} \quad (4.67)$$

$$\mathbf{C}_m(n-1) = \begin{bmatrix} 0 \\ \mathbf{C}_{m-1}(n-1) \end{bmatrix} + \kappa_m^b(n-1) \begin{bmatrix} \mathbf{A}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (4.68)$$

Şekil 4.5'de ileri ve geri kestirim hata seviyelerinin güncellenmesi görülmektedir. İlk seviyedeki hatalar birinci antenin aldığı işarete eşitlenir. Diğer antenlerin aldığı işaretler ise sisteme (4.60) ifadesinden de görüldüğü gibi $g_m(n)$ aracılığıyla dahil edilir.

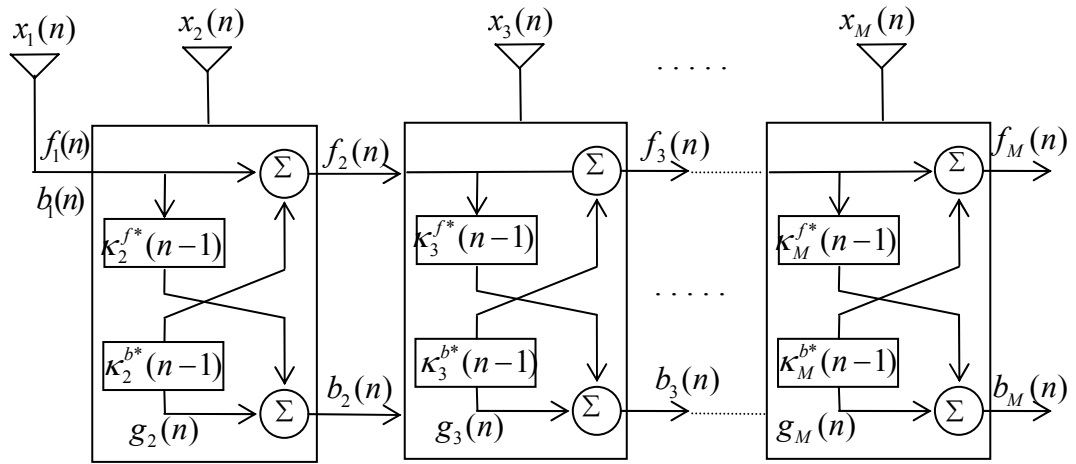
$$f_1(n) = b_1(n) = x_1(n) \quad (4.69)$$

İstenen işaretin kestirimi ise aynı şekilde ortak süreç kestirim ifadeleriyle yapılır;

$$\mathbf{b}_M(n) = \begin{bmatrix} b_1(n) \\ b_2(n) \\ \vdots \\ b_M(n) \end{bmatrix} \quad \text{ve} \quad \mathbf{h}_M(n) = \begin{bmatrix} h_1(n) \\ h_2(n) \\ \vdots \\ h_M(n) \end{bmatrix} \quad \text{olmak üzere;}$$

$$\hat{d}(n) = \mathbf{h}_M^H(n) \mathbf{b}_M(n) \quad (4.70)$$

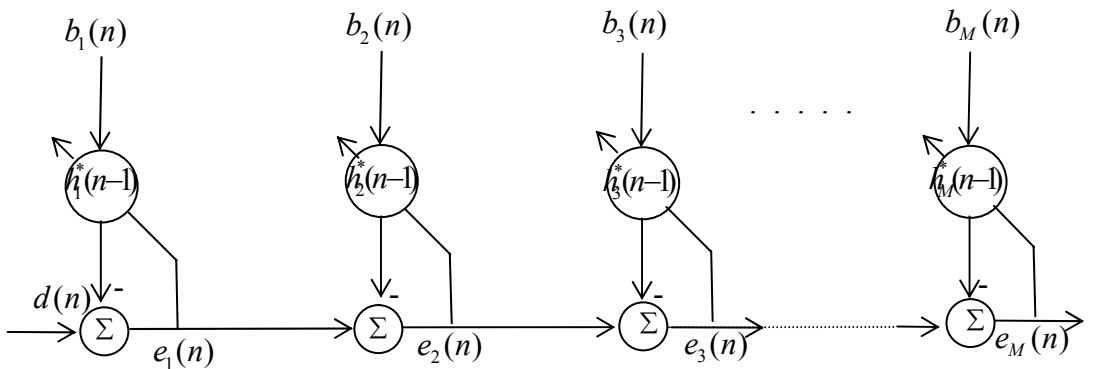
$$e_m(n) = e_{m-1}(n) - h_m^*(n-1)b_m(n) \quad (4.71)$$



Şekil 4.5 Kafes algoritması ileri ve geri hata işaret akış diyagramı.

(4.71) ifadesinde $e_m(n)$ her bir seviyede yapılan kestirim ile istenen $d(n)$ işareti arasındaki hatayı belirtmektedir. Bu ifadede 0. derecedeki hata istenen referans işaretine eşit alınır.

$$e_0(n) = d(n) \quad (4.72)$$



Şekil 4.6 Kafes algoritması ortak süreç kestirimi.

$h_M(n)$ ortak süreç kestirim katsayıları aşağıdaki hata maliyet fonksiyonunu minimize edecek şekilde bulunur;

$$\xi_m(n) = \sum_{k=1}^n \lambda^{n-k} |e_m(k)|^2 = \sum_{k=1}^n \lambda^{n-k} |d(k) - \mathbf{h}_M^H(k) \mathbf{b}_M(k)|^2 \quad (4.73)$$

$$h_m(n) = \frac{\sum_{k=1}^n \lambda^{n-k} e_{m-1}^*(k) b_m(k)}{\sum_{k=1}^n \lambda^{n-k} |b_m(k)|^2} \quad (4.74)$$

Anten çıkışlarına uygulanacak ağırlık katsayılarının bulunması için ortak süreç kestirim katsayılarından Wiener çözümüne geri dönülmesi gerekmektedir. Bu dönüşüm işlemi (4.53) ifadesinde verilmektedir. Tekrar L matrisi elde edilip bu ifade yazılırsa;

$$\mathbf{L}_M(n) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ c_{2,1}^*(n-1) & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ c_{M,1}^*(n-1) & c_{M,2}^*(n-1) & \dots & 1 \end{bmatrix} \quad (4.75)$$

$$\mathbf{w}_M(n) = \mathbf{L}_M^H(n) \mathbf{h}_M(n) \quad (4.76)$$

Algoritmanın zaman güncelleme ifadeleri ise aşağıdaki eşitliklerle verilmektedir;

$$\xi_m^f(n) = |f_m(n)|^2 + \lambda \xi_m^f(n-1) \quad (4.77)$$

$$\xi_m^b(n) = |b_m(n)|^2 + \lambda \xi_m^b(n-1) \quad (4.78)$$

$$\xi_m^g(n) = |g_m(n)|^2 + \lambda \xi_m^g(n-1) \quad (4.79)$$

$$f_m(n) = f_{m-1}(n) + \kappa_m^{f*}(n-1) g_m(n) \quad (4.80)$$

$$b_m(n) = g_m(n) + \kappa_m^{b*}(n-1) f_{m-1}(n) \quad (4.81)$$

$$\kappa_m^f(n) = \kappa_m^f(n-1) - \frac{g_m(n)}{\xi_m^g(n)} f_m^*(n) \quad (4.82)$$

$$\kappa_m^b(n) = \kappa_m^b(n-1) - \frac{f_{m-1}(n)}{\xi_m^g(n)} f_m^*(n) \quad (4.83)$$

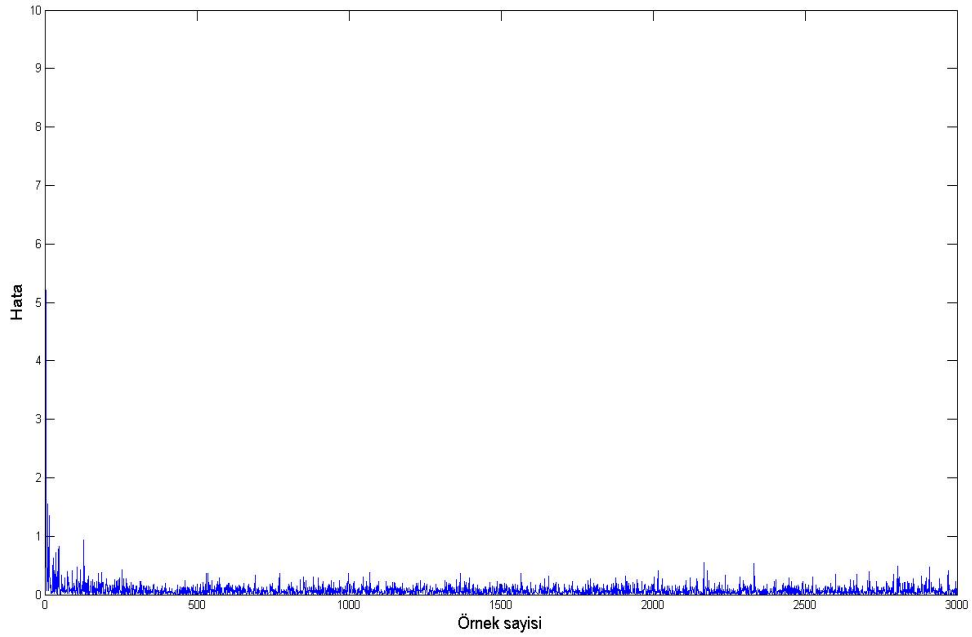
$$h_m(n) = h_m(n-1) + \frac{b_m(n)}{\xi_m^b(n)} e_m^*(n) \quad (4.84)$$

4.4.2 Kafes algoritması simülasyon sonuçları

Simülasyon 1:

İstenen s_o işaretin geliş açısının 30° , -20° , 0° ve 60° geliş açlarına sahip 3 tane interferans kaynağının olduğu varsayılmaktadır.

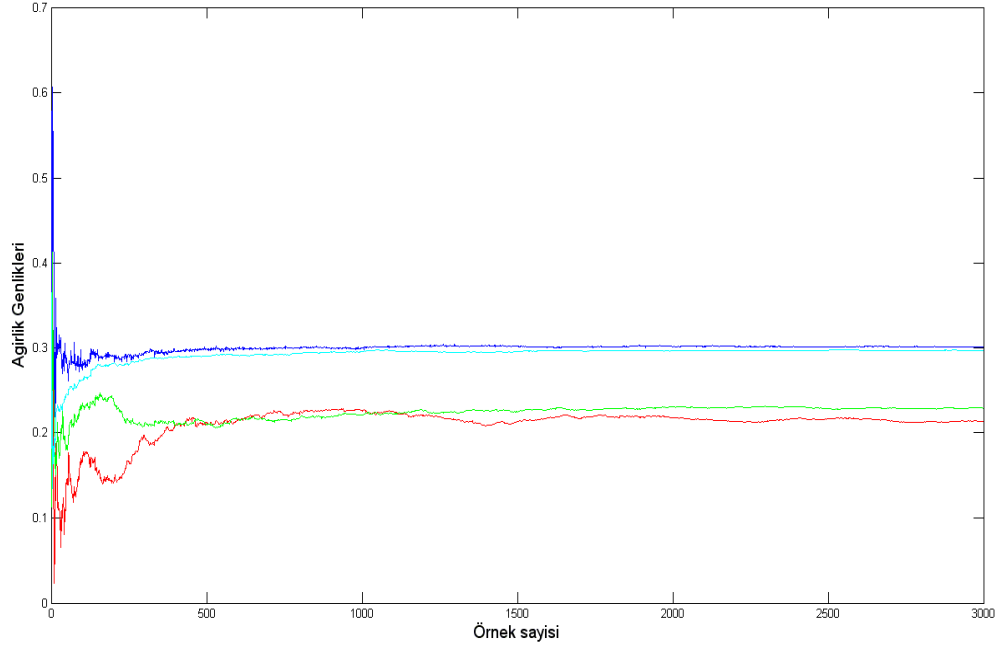
Şekil 4.7’de görüldüğü gibi Kafes algoritması çok hızlı yakınsamıştır, hata çok kısa bir süre içinde 0.5 değerinin altına düşmüştür. Bu yakınsama eğrisi RLS algoritmasıyla elde edilen sonuca oldukça yakındır. Ancak algoritmanın gerçek yakınsama zamanı, ağırlık katsayılarının zamanla değişimi çizdirilip incelenirse daha iyi görülebilir.



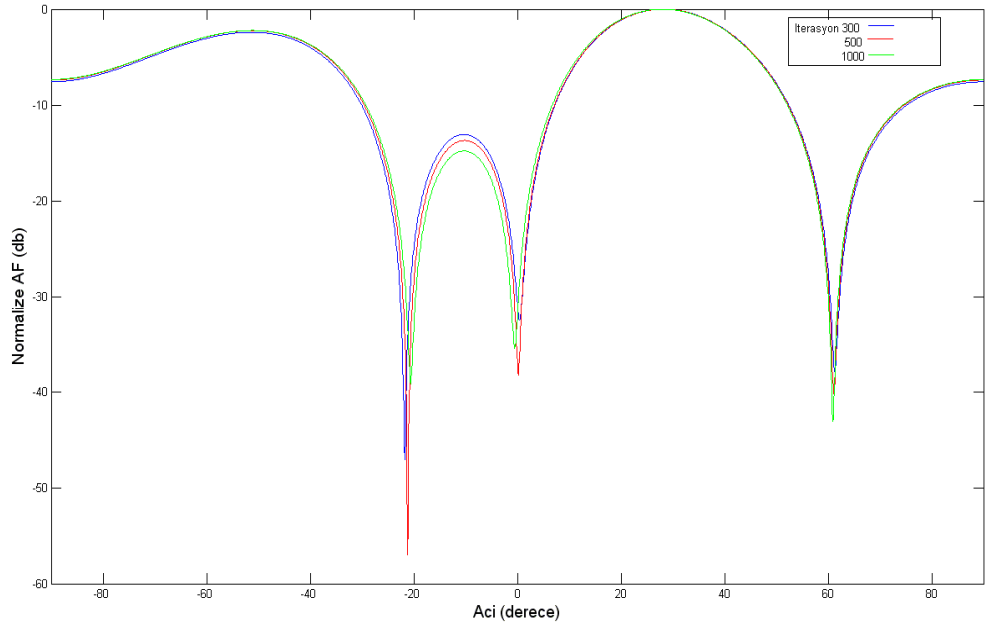
Şekil 4.7 Hatanın zamanla değişimi.

Şekil 4.8’de ağırlık genliklerinin zamanla değişimi görülmektedir. Bazı katsayılar çok hızlı yakınsamasına karşın diğerlerinin yakınsaması biraz uzun sürmüştür. Fakat yine de 300. iterasyondan sonra ağırlık genliklerinde fazla bir değişim olmamaktadır. Algoritmanın değişik anlarda elde ettiği ağırlıklar kullanılarak çizdirilmiş huzme biçimleri ise Şekil 4.9’da görülmektedir. Hem 300, hem de 500. iterasyonda algoritma tüm işaretlerin geliş açılarını yakalamıştır. 500. iterasyonda interferans

kaynakları yönünden gelen işaretleri daha da aşağıya çekerek bastırmıştır. 1000. iterasyonda ise daha dengeli bir huzme biçimi elde edilmiştir. Bu huzme biçimlerinden algoritmanın 300. iterasyondan sonra birbirine çok yakın huzme biçimleri elde ettiği söylenebilir.



Şekil 4.8 Ağırlık genliklerinin zamanla değişimi.

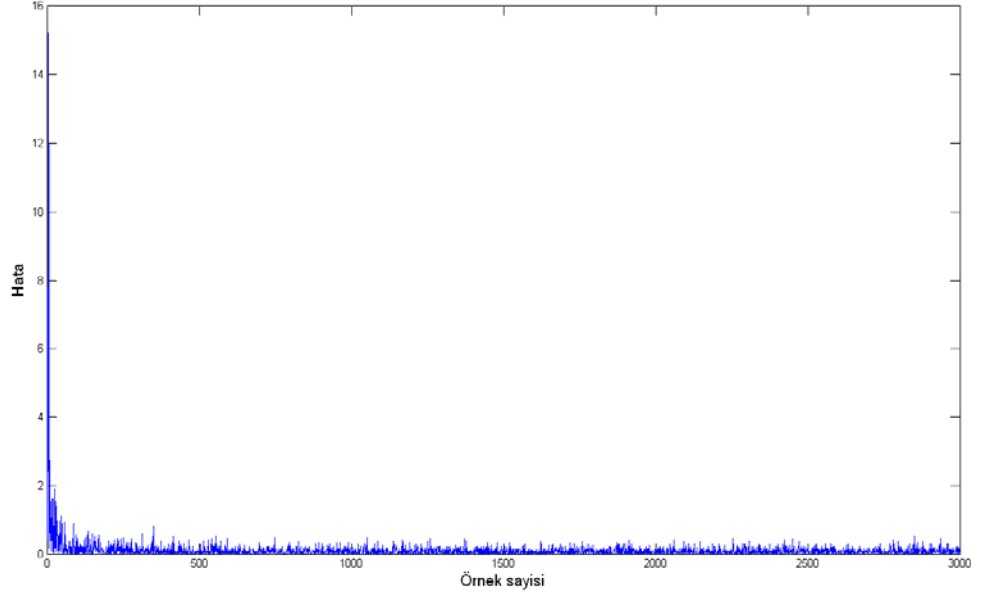


Şekil 4.9 Huzme biçiminin zamanla değişimi.

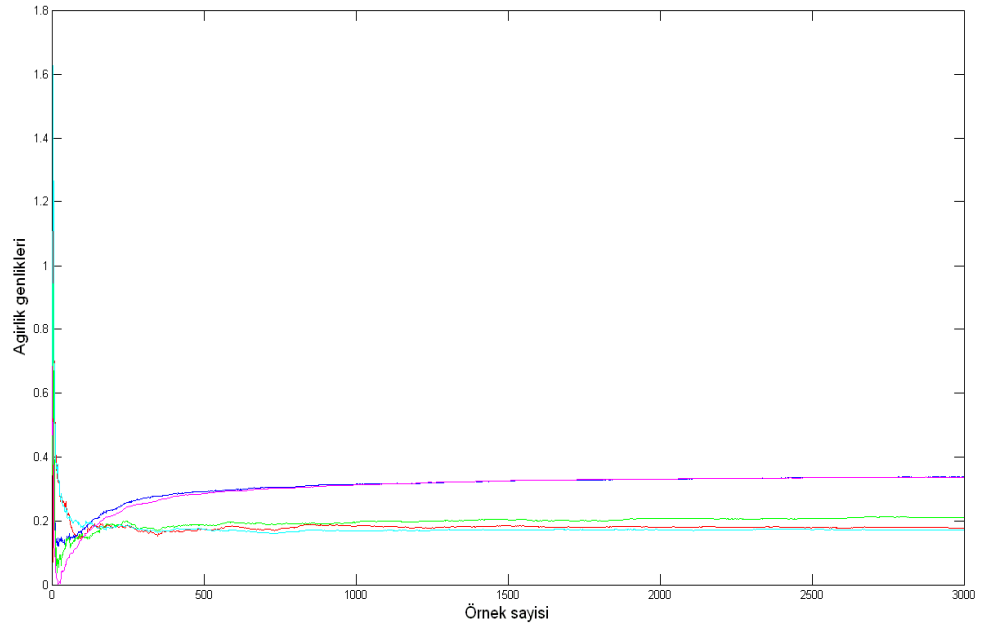
Simülasyon 2:

İstenen s_o işaretin geliş açısı 0° ve $-40^\circ -10^\circ 30^\circ$ ve 60° geliş açlarına sahip 4 tane interferans işaret kaynağı olduğu varsayılmaktadır.

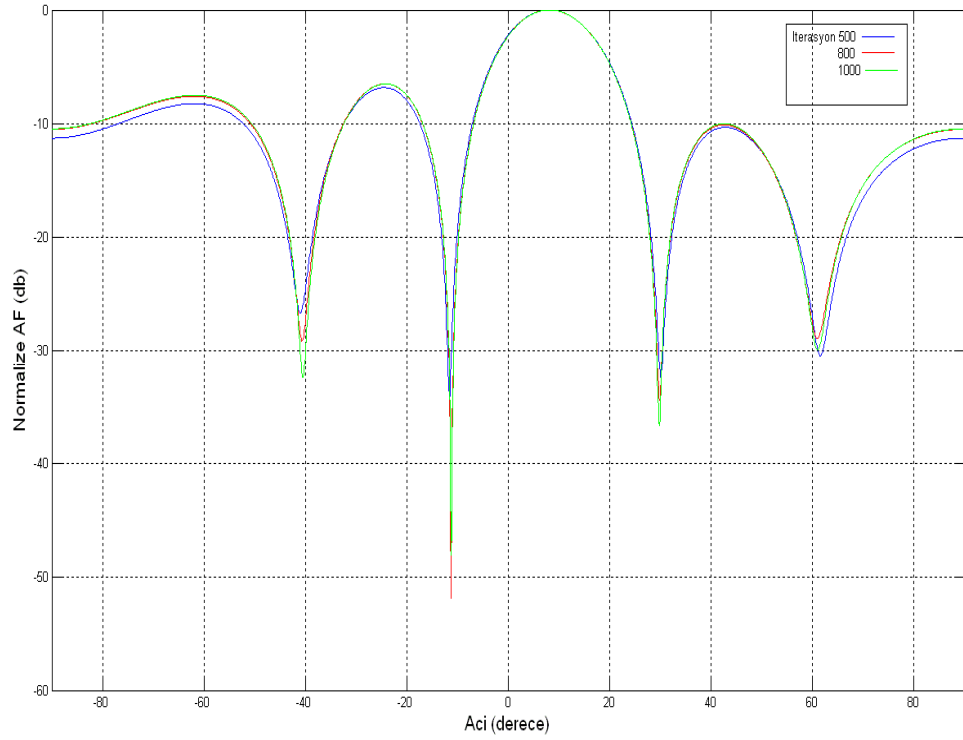
Şekil 4.10'da görüldüğü gibi algoritma yine çok kısa bir sürede yakınsamıştır. Ağırlık genliklerinin değişimi ise Şekil 4.11'de görülmektedir. Ağırlık genlikleri 500. iterasyondan sonra fazla değişim göstermemektedir. Değişik anlarda elde edilen katsayılar kullanılarak çizdirilmiş huzme biçimleri ise Şekil 4.12 de görülmektedir. 500. iterasyondan sonra algoritma interferans işaretlerinin geliş açılarını tam olarak yakalamıştır ve iterasyon sayısı arttıkça yine aynı şekilde algoritma interferans işaretlerini daha aşağı bastıran huzme biçimleri oluşturmaktadır.



Şekil 4.10 Hatanın zamanla değişimi.



Şekil 4.11 Ağırlık genliklerinin zamanla değişimi.

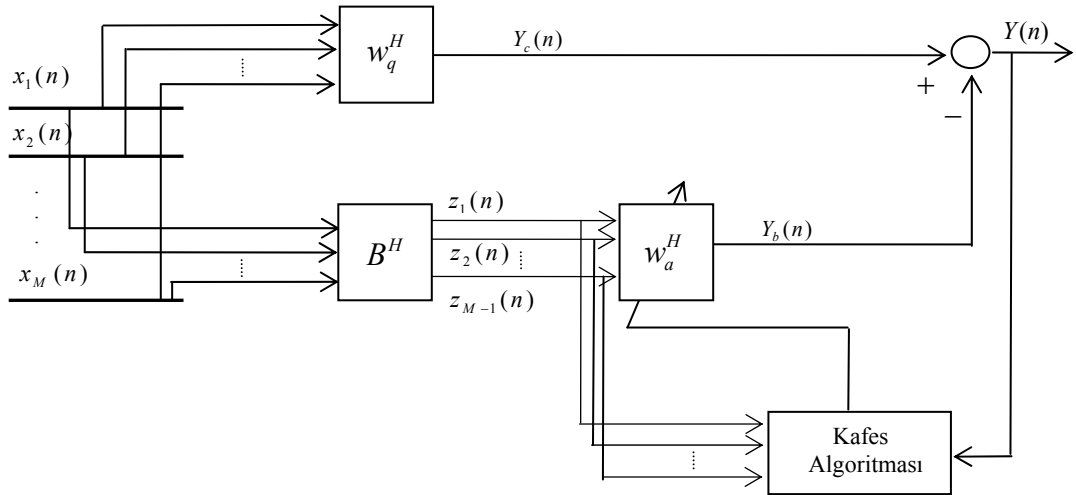


Şekil 4.12 Huzme biçiminin zamanla değişimi.

4.5 Uyarlamalı Kafes Algoritmasıyla Genelleştirilmiş Yan Lob Engelleyici

Bu çalışmada önerilen yöntem, bir önceki bölümde anlatılan Genelleştirilmiş Yan Lob Engelleyici (GSLC) huzme biçimlendiricinin uyarlamalı kısmında Kafes algoritmasının kullanılmasıdır. Kafes algoritmasıyla gerçekleştirilmiş GSLC huzme biçimlendirici Şekil 4.13'de görülmektedir. Bu yapıda Kafes algoritması uyarlamalı w_a^H ağırlıklarını güncelleyerek GSLC yapısının üst tarafında elde edilen $Y_c(n)$ çıkışındaki gürültü bileşenini gidermeye çalışmaktadır.

C sınırlandırma matrisi olarak istenen işaret için yönlendirme vektörü kullanılmıştır. Dolayısıyla M antenli durum için B^H engelleme matrisi çıkışı $M-1$ boyutlu olacaktır, z_1, z_2, \dots, z_{M-1} ile gösterilen bu çıkışlar bir önceki bölümde anlatılan Kafes algoritmasındaki x_1, x_2, \dots, x_{M-1} anten çıkışları olarak kabul edilmektedir.



Şekil 4.13 Kafes algoritmasıyla GSLC yapısı.

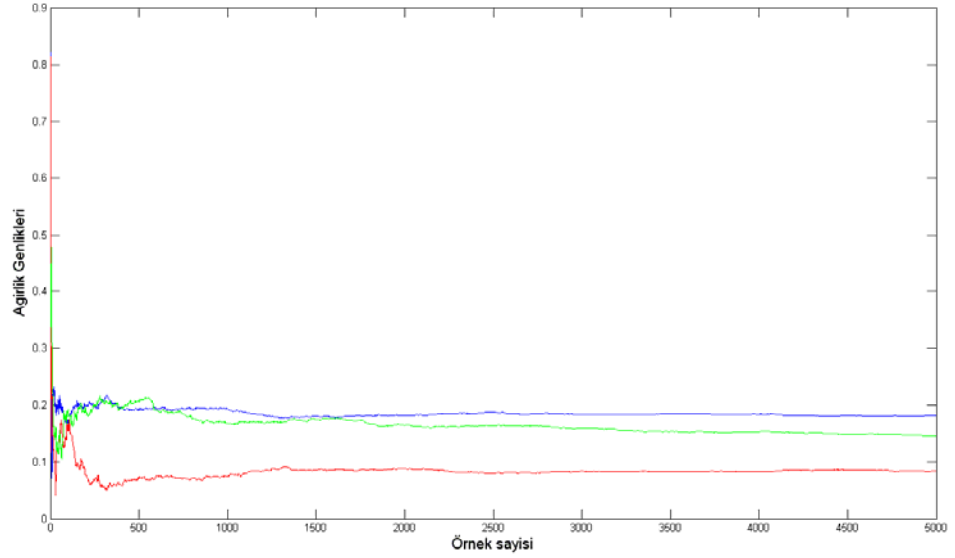
4.5.1 Kafes-GSLC simülasyon sonuçları

Simülasyon 1:

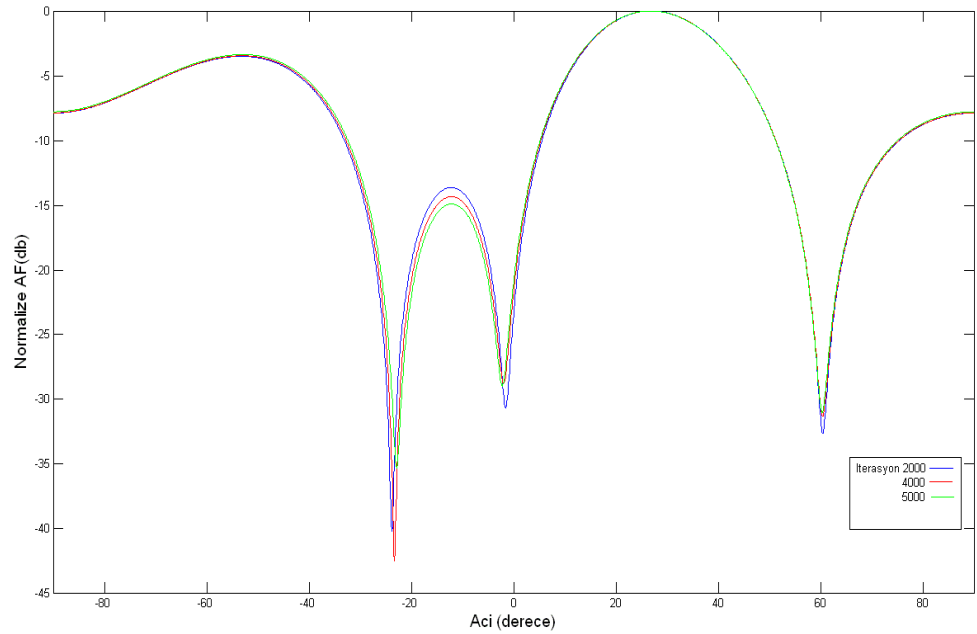
İstenen s_o işaretin geliş açısının 30° , -20° , 0° ve 60° geliş açılara sahip 3 tane interferans kaynağının olduğu varsayılmaktadır.

Şekil 4.14'de w_a ağırlıklarının zamanla genlik değişimleri görülmektedir. Yaklaşık 1000 iterasyonda ağırlıklar dengeye oturmaktadır. Değişik iterasyonlarda elde edilen

ağırlıklarla çizdirilmiş huzme biçimi ise Şekil 4.15’de görülmektedir. Şekilden de görüldüğü gibi 1000. iterasyondan sonra elde edilen ağırlıklar hemen hemen aynı huzme biçimlerini vermektedir. Algoritma bütün interferans işaretleri geliş açılarını bulmada aynı başarıyı göstermese de bazı açılar için oldukça iyi sonuç vermiştir.



Şekil 4.14 w_a ağırlık katsayılarının değişimi.

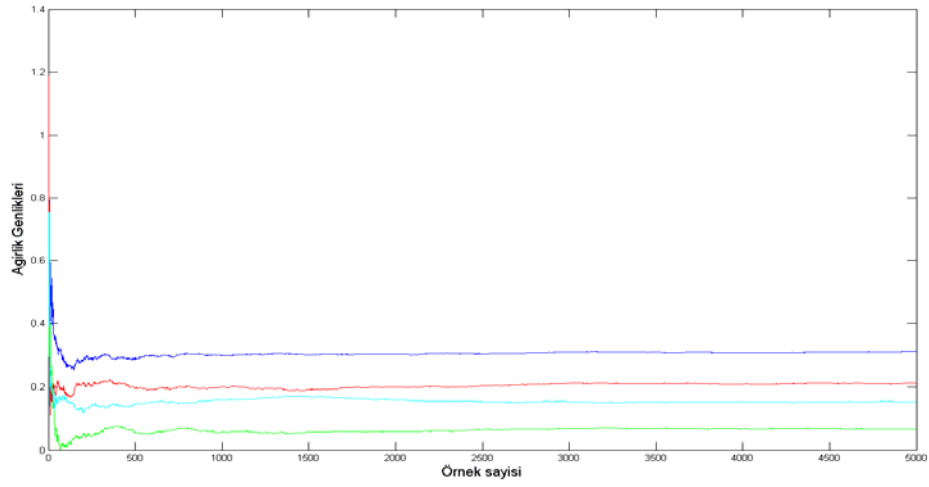


Şekil 4.15 Kafes-GSLC huzme biçimi.

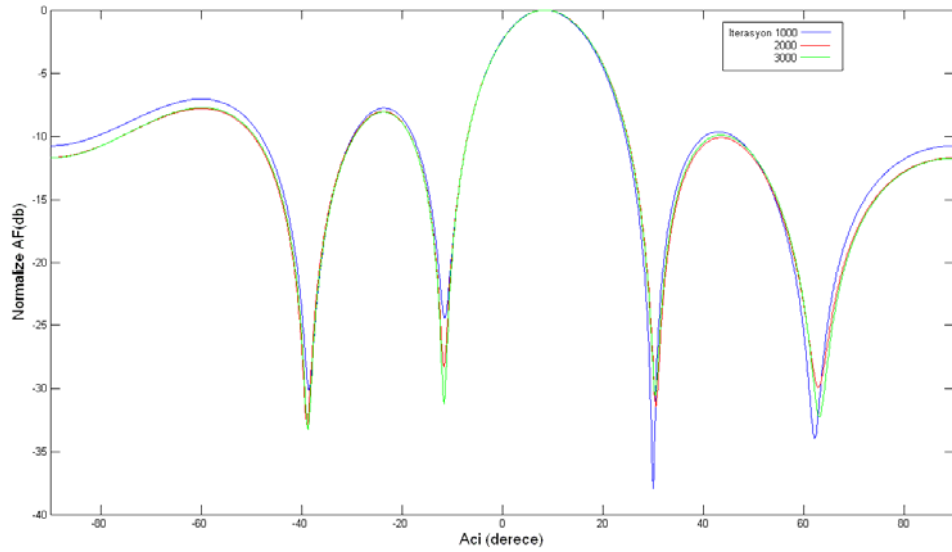
Simülasyon 2:

İstenen s_o işaretin geliş açısı 0° ve -40° -10° 30° ve 60° geliş açlarına sahip 4 tane interferans işaret kaynağı olduğu varsayılmaktadır.

Şekil 4.16'daki w_a ağırlıklarının değişiminden yine ağırlıkların yaklaşık olarak 1000 iterasyonda yakınsadığı söylenebilir. Değişik iterasyonlarda elde edilen ağırlıklarla çizdirilmiş Şekil 4.17'deki huzme biçimlerinden algoritmanın bu simülasyon için çok daha iyi sonuç verdiği görülmektedir; bütün interferans işaretleri geliş açıları hemen hemen yakalanmıştır.



Şekil 4.16 w_a ağırlık katsayılarınının değişimi.



Şekil 4.17 Kafes-GSLC huzme biçimi.

5. SONUÇLAR

Bu tez çalışmasında ULA geometrili uyarlamalı anten dizileri incelenmiştir. Çok kullanılan bazı huzme biçimlendirme teknikleri ayrıntılı olarak ele alınıp performansları simülasyonlarla sunulmuştur.

LMS algoritması basit olmasına rağmen çok etkili bir algoritmadır. LMS algoritmasıyla elde edilen huzme biçimlerinden, interferans işaretleri geliş açıları yönünde derin sıfırlar konulduğu görülmüştür. Ancak LMS algoritmasının yakınsama karakteristiği anten dizisi tarafından alınan işaret korelasyon matrisi özdeğer dağılımına bağlıdır. Geniş bir aralığa yayılmış özdeğer dağılımı için LMS algoritması çok geç yakınsamaktadır.

RLS algoritması, anten dizisi işaret korelasyon matris tersini özyinelemeli olarak hesaplamaktadır. Bu yönüyle RLS algoritması LMS algoritmasından farklıdır ve simülasyon sonuçlarından oldukça hızlı yakınsadığı görülmüştür. RLS algoritmasının önemli bir başka özelliği ise giriş verisine bağımlı olmamasıdır; bu nedenle RLS algoritması anten dizisi işaret korelasyon matrisi özdeğer dağılımından etkilenmez. Simülasyonlardan RLS algoritmasının çok kısa sürede interferans işaretleri geliş yönlerinde oldukça derin sıfırlar koyduğu görülmüştür. RLS algoritması bu derece iyi çalışmasına rağmen hesaplama yükü oldukça fazla ve karmaşık bir algoritmadır.

Kafes kestiriciler işaret işlemede önemli avantajlar sunmaktadır. Dar bantlı işaretler için Kafes kestirici kullanımı çok yaygın olmamasına karşın bu tez çalışmasında böyle bir Kafes kestirici kullanılarak huzme biçimlendirme performansı değerlendirilmiştir. Simülasyon sonuçlarından Kafes algoritmasının yakınsama karakteristiğinin RLS algoritmasına oldukça benzediği görülmüştür. Kafes algoritması huzme biçimlerinde, oldukça kısa sürede interferans işaretleri geliş açıları yakalanmış ve derin sıfırlar konulmuştur.

Haberleşme sistemlerinde alıcı tarafın istenen işaret hakkında bilgi sahibi olması her zaman mümkün değildir. Bu durumda alıcı taraf bazı bilgileri kullanarak istenen işareti çekme yoluna gitmektedir ve genellikle bu bilgi istenen işaretin geliş açısı

bilgisidir. LCMV huzme biçimlendirme referans işaretine gerek duymadan çalışan huzme biçimlendirme yöntemlerinden birisidir ve bu çalışmada ayrıntılı olarak ele alınmıştır. Genelleştirilmiş Yan Lob Engelleyici, LCMV huzme biçimlendiriciyi gerçekleminin değişik bir yoludur. Genelleştirilmiş Yan Lob Engelleyici yapısında anten dizisi tarafından alınan işaret uzayı iki kısma ayrılır. Uyarlamalı olan kısım diğer kısımda elde edilen işaretin gürültü bileşenini gidermeye çalışır, bu aynı zamanda huzme biçiminde yan lobların bastırılması anlamına gelmektedir.

Bu çalışmada Genelleştirilmiş Yan Lob Engelleyici yapısının uyarlamalı kısmında LMS ve RLS algoritmaları kullanılmış, huzme biçimlendirme performansları simülasyonlarla değerlendirilmiştir. RLS algoritması yine çok hızlı yakınsamıştır ve düzgün huzme biçimleri elde etmeyi başarmıştır. Yavaş yakınsayan LMS algoritması ise belirli bir iterasyondan sonra interferans işaretlerini hatasız yakalamayı başarmıştır ve oldukça derin sıfırlar koyan huzme biçimleri vermiştir. Bu çalışmada alternatif bir yöntem olarak sunulan Kafes algoritmasıyla gerçekleştirilmiş Genelleştirilmiş Yan Lob Engelleyici huzme biçimlendirici simülasyonlarından, Genelleştirilmiş Yan Lob Engelleyici yapısının uyarlamalı kısmında kullanılan Kafes algoritmasının interferans işaretleri geliş açılarını hemen hemen yakaladığı görülmüştür. Bu yapıda Kafes algoritması, interferans işaretlerini LMS algoritması kadar iyi bastıramamasına rağmen LMS algoritmasından çok daha hızlı yakınsamıştır ve düzgün huzme biçimleri vermeyi başarmıştır.

5.1 İlerideki Çalışmalar

Kafes algoritmasında bazı değişiklikler yapılarak Genelleştirilmiş Yan Lob Engelleyici içerisinde performansı artırılabilir ve daha iyi sonuçlar elde edilebilir. Bu çalışmanın ilerisinde, daha gelişmiş kafes algoritmaları, özellikle çok kanallı (multi-channel) veriler için geliştirilmiş kafes yapıları kullanılarak performans artırılması düşünülmektedir.

KAYNAKLAR

- [1] **Van Atta, L.**, 1959. Electromagnetic reflection, Technical report, U.S. Patent 2908002.
- [2] **Howells, P.W.**, 1959. Intermediate frequency sidelobe canceller, Technical report U.S. patent 3202990.
- [3] **Howells, P.W.**, 1976. Explorations in fixed and adaptive resolution at GE and SURC, IEEE trans. antenna and propagation, Special issue on adaptive antennas, vol. 24, pp. 575-584.
- [4] **Walden, R.H.**, 1999. Performance trends for analog to digital converters, IEEE commn. mag., pp. 96-101.
- [5] **Reed, J.**, 2002. Software radio: A modern approach to radio engineering, Prentice Hall, New York.
- [6] **Mitola, J.**, 1995. Software radios, IEEE Commun. Mag.
- [7] **Doufexi, A. & Armour, S.**, 2004. Range and throughput enhancement of wireless local area networks using smart sectorised antennas. IEEE trans. wireless communications, vol. 3, pp. 1437-1443.
- [8] **Skolnik, M.**, 2001. Introduction to radar systems, McGraw-Hill: third edition, New York.
- [9] **Jeng, S. & Lin, H.**, 1999. Smart antenna system and its application in low-earth-orbit satellite communication systems, IEEE proc. microwaves antennas and propagation, vol. 146, pp. 125-130.
- [10] **Durgin, G.**, 2003. Space-time wireless channels, Prentice Hall, New York.
- [11] **Haykin, S.**, 2002. Adaptive filter theory: Fourth edition, Prentice Hall, New Jersey.
- [12] **Farhang-Boroujeny, F.**, 1999. Adaptive filters: Theory and applications , John Wiley and Sons, New York.
- [13] **Bellanger, M.G.**, 2001. Adaptive digital filters: Second edition, Revised and expanded , Marcel Dekker , New York.
- [14] **Diniz, P.S.R.**, 2008. Adaptive filtering algorithms and practical implementation: Third edition, Springer, New York.
- [15] **Stoica, P. & Moses, R.**, 1997. Introduction to Spectral Analysis, Prentice Hall, New Jersey.
- [16] **Poularikas, A.D. & Ramadan Z.M.**, 2006. Adaptive filtering primer with Matlab, Taylor&Francis, London.
- [17] **Hayes, M.H.**, 1996. Statistical digital signal processing and modeling, John Wiley and Sons, New York.

- [18] **Widrow, B. & Stearns, S.D.**, 1985. Adaptive signal processing, Prentice Hall, New Jersey.
- [19] **Manolakis, D.G.**, 2005. Statistical and adaptive signal processing: Spectral estimation, Signal modeling, Adaptive filtering and Array processing, Artech House, Boston.
- [20] **Friis, H.T. & Feldman, C.B.**, 1937. A multiple unit steerable antenna for short-wave reception, Bell syst. Tech. J., vol. 16, pp. 3337-419.
- [21] **Applebaum, S.P.**, 1976. Adaptive arrays, IEEE trans. antennas and propagation, vol. 24, pp. 585-598.
- [22] **Widrow, B. & Griffiths L.J.**, 1967. Adaptive antenna systems, Proc. IEEE, vol. 55, no. 12, pp. 2143-2159.
- [23] **Griffiths, L., J.**, 1969. A simple adaptive algorithm for real time processing in antenna arrays, Proc. IEEE, vol.57, pp. 1969-1704.
- [24] **Frost, O.L.**, 1972. An algorithm for linearly constrained adaptive array processing, Proc. IEEE, vol. 60, pp.926-935.
- [25] **Compton, R. & Huff, R. & Swarner, W. & Ksienski, A.**, 1976. Adaptive arrays for communication systems: An overview of research at Ohio State University, IEEE trans. antennas and propagation, vol. 60, pp. 599-607.
- [26] **Van Trees, H.L.**, 2002. Optimum array processing: Part IV of detection, estimation and modulation theory, John Wiley and Sons, New York.
- [27] **Allen, B. & Ghavami, M.**, 2005. Adaptive array systems :Fundamentals and applications, John Wiley and Sons, London.
- [28] **Monzingo, R.A. & Miller, T.W.**, 2004. Introduction to adaptive arrays, Scitech Publishing, North Carolina.
- [29] **Johnson D.H. & Dudgeon, D.E.**, 1993. Array signal processing: Concepts and techniques, Prentice Hall, New Jersey.
- [30] **Blaunstein, N. & Christodoulou, C.**, 2007. Radio propagation and adaptive antennas for wireless communication links, John Wiley and Sons, New York.
- [31] **Janaswamy, R.**, 2001. Radiowave propagation and smart antennas for wireless communications, Kluwer Academic Publishers, New York.
- [32] **Url**<http://www.iec.org/online/tutorials/smart_ant/topic03.html>, alındığı tarih 15.01.2009.
- [33] **Weiner, M.M.**, 2006. Adaptive antennas and receivers, Taylor&Francis, New York.
- [34] **Zooghby, A.**, 2005. Smart antenna engineering, Artech House, Boston.
- [35] **Gross, F.B.**, 2005. Smart antennas for wireless communications with Matlab, McGraw-Hill, New York.
- [36] **Godara, L., C.**, 2000. Smart antennas, CRC Press, Florida.

- [37] **Lee, H. & Li, F.**, 1994. An eigenvector technique for detecting the number of emitters in a cluster. *IEEE trans. signal process.*, vol. 42, pp. 2380-2388.
- [38] **Saraç, U. & Harmancı, F.K. & Akgül, T.**, 2006. Detection and localization of multiple emitters in multipath environments, *Signal processing and communications applications*, IEEE vol. 14 , Issue , pp. 1 – 4.
- [39] **Griffiths, L.J. & Jim, C.W.**, 1982. Alternative approach to linearly constrained beamforming, *IEEE trans. antennas propagat.*, vol. AP-30, pp. 27-34.
- [40] **Haidong, W. & Jiangzhou, W.**, 2002. On the use of lattice algorithm for adaptive antennas, *Communications, 2002. ICC 2002. IEEE international conference on* vol. 1, Issue, pp. 177-181.

ÖZGEÇMİŞ

Ad Soyad : Harun Ünaldı

Doğum Yeri ve Tarihi : İstanbul 25.05.1980

Lisans Üniversite : Yıldız Teknik Üniversitesi