

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UYARLANABİLİR SÜZGEÇ KULLANILARAK
SES YANKISININ GİDERİLMESİ

ERTUĞRUL SUNAN

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜH. ANABİLİM DALI

DANIŞMAN
YRD. DOÇ. DR. HATİCE SEZGİN

SAMSUN - 2007

UYARLANABİLİR SÜZGEÇ KULLANILARAK SES YANKISININ GİDERİLMESİ

ÖZET

Yankı, bir sinyalin gecikmiş değerlerinin şu andaki değerine girişim yapması ile oluşur. Ses yankısı da ses dalgalarının çevredeki nesnelere çarparak geri yansımaları sonucunda oluşur ve asıl ses dalgasına girişim yapar. Yankı yolu, bir kanal olarak düşünülür ve süzgeç kullanılarak etkisi giderilebilir. Ancak ses sinyalleri ve yankı yolu zamanla değişir ve gerçek zamanlı uygulamalarda sabit katsayılı süzgeçler kullanılarak yankı giderilemez. Bu nedenle uyarlanabilir süzgeçler kullanılır. Uyarlanabilir süzgeçler, her giriş değeri için katsayılarını dolayısı ile darbe tepkisini değiştiren süzgeçlerdir. Bu nedenle rasgele değişen gerçek zamanlı uygulamalarda tercih edilirler.

Bu çalışmada uyarlanabilir süzgeçler kullanarak tek ve çok yankılı sinyallerin yankıları giderildi. Öncelikle kayıtlı ses verisindeki yankılar, FIR ve IIR yapıdaki uyarlanabilir süzgeçler ile giderildi. Bunun için kanal dengeleme kullanıldı. Son olarak, telekonferans benzetimi yapılarak ses yankıları giderildi.

Anahtar Kelimeler: Ses yankısı, uyarlanabilir süzgeçler, LMS, kanal dengeleme , körs ters katlama

ACOUSTIC ECHO CANCELLATION BY USING ADAPTIVE FILTER

ABSTRACT

An echo consist of delayed and original signals. An acoustic echo is the reflecting of an acoustic wave off the vicinity of an object and reflected wave (echo) is interferenced with the original source. That is a source of contamination (it is the problem). Echo path is assumed as a channel and distorsion effect can be removed by echo-canceling filters. However acoustic signals and echo path may change in time, and in real-time applications, echoes cannot be removed by using constant-coefficient filters. Therefore, adaptive filters are widely used in real-time application. In real-time applications, coefficients of an adaptive filter and its impulse response can be simultaneously changed by specilized algorithms.

In this thesis, canceling echoes of single and multi echo signals by using adaptive filters has been evaluated by computer simulations. Initially, echoes in the (recorded) acoustic data have been filtered by FIR and IIR adaptive filters. Equalization has been applied to cancel the effect of the echoes in the channel. Finally, canceling acoustic echoes in teleconference has been succesfully demonstrated.

Keywords: Acoustic echo, adaptive filters, LMS, channel equalisation, blind deconvolution

TEŐEKKÜR

Çalıőmalarım sırasında desteęini ve yardımlarımı esirgemeyen, bilgi ve deneyimlerinden yararlandıęım çok deęerli hocam ve danıőmanım Yrd. Doç. Dr. Hatice SEZGİN'e ; tez çalıőmam sırasında manevi desteklerini esirgemeyen arkadaşlarım Cengiz TEPE, Begüm Korunur ENGİZ , Gaye Yeőim TAFLAN ve Hanife USTA'ya teőekkürlerimi sunarım.

İÇİNDEKİLER

1. GİRİŞ	1
2. LİTERATÜR ÖZETİ	2
3. MATERYAL VE METOT	3
3.1. Ses Ve Ses Dalgaları	3
3.2. Ses Sinyali(Audio)	7
3.2.1. Sayısal Ses sinyali	7
3.3. Uyarlanabilir Süzgeçler	14
3.3.1. Uyarlanabilir Süzgeç Yapıları	17
3.3.1.1. FIR Süzgeç Yapıları	18
3.3.1.2. IIR Süzgeç Yapıları	19
3.3.2. Uyarlanabilir Süzgeç Algoritmaları	21
3.3.2.1. Rasgele Gradyan Yaklaşımı	22
3.3.2.2. En Küçük Kareler Kestirimi	29
3.3.3. Uyarlanabilir Süzgeç Uygulamaları	30
3.3.3.1. Sistem/Kanal Tanımlama (Modelleme)	31
3.3.3.2. Ters Sistem Modelleme	32
3.3.3.3. Yankı Giderme	33
3.3.3.4. Gürültü/Girişim Giderme	35
3.3.3.5. Doğrusal Tahmin	36
3.3.3.6. Kanal Dengeleme	39
3.4. Uyarlanabilir Süzgeç Kullanarak Ses Yankısının Giderilmesi	40
3.4.1. Yankı ve Ses Yankısı	40
3.4.1.1. Ses Sinyaline Yankının Eklenmesi	42
3.4.2. İlinti Kullanarak Yankı Parametrelerinin Bulunması	46
3.4.3. Yankı Giderme Yöntemleri	52
3.4.3.1. Kanal Dengeleme İle Yankının Giderilmesi	52
3.4.3.2. Ters Süzgeçler	54
3.4.3.3. FIR Modellenmiş Bir Kanalın Dengelenmesi	56
3.4.3.4. IIR Modellenmiş Bir Kanalın Dengelenmesi	59
3.4.3.5. Uyarlanabilir Süzgeç İle Kanal Dengeleme Ve Yankı Giderme	60
3.4.3.6. Ters Katlama	61
3.4.3.7. Kör Ters Katlama	64
4. BULGULAR	67

4.1. FIR Modellenmiş Bir Kanalı Dengeleme	67
4.2. IIR Modellenmiş Bir Kanalı Dengeleme	71
4.3. Uyarlanabilir Süzgeç İle Kanal Dengeleme Ve Yankı Giderme	73
4.3.1. LMS Algoritması Kullanan Dengeleyici	74
4.3.2. NLMS Algoritması Kullanan Dengeleyici	82
4.3.3. RLS Algoritması Kullanan Dengeleyici	91
4.4. Ses Yankılarının Giderildiği Telekonferans Benzetimi	97
5. SONUÇ VE ÖNERİLER	101
6. KAYNAKLAR	102
7. EKLER	104
EK A. Matlab Ortamında Yazılan Fonksiyonlar	104
ÖZGEÇMİŞ	117

KISALTMALAR

- LMS : En küçük ortalama kare
NLMS : Normalize en küçük kare
RLS :Yinelemeli enküçük kare
FIR : Sonlu birim vuruş tepkeli
IIR : Sonsuz birim vuruş tepkeli
MSE : Ortalama kare hata

ŞEKİLLERİN LİSTESİ

Şekil Numarası

Şekil 3.1. Enine Dalga ve Boyuna Dalga	3
Şekil 3.2. Bir yay üzerinde ilerleyen enine ve boyuna dalgalar	4
Şekil 3.3. Bir tüp içinde ses dalgasının oluşturduğu basınç değişimi	4
Şekil 3.4. Farklı basınç bölgelerindeki molekül yoğunluğu	5
Şekil 3.5 Ses basıncının molekül yoğunluğuna etkisi	5
Şekil 3.6. Ses dalgasının zayıflaması	6
Şekil 3.7. Farklı örnekleme frekanslarının etkisi	8
Şekil 3.8. İdeal örneklemenin matematiksel modeli	9
Şekil 3.9. Örnekleme a)Asıl sinyal b)Birim vuruş katarı c)Örneklenmiş sinyal	10
Şekil 3.10.a)x(t) sinyalinin frekans tayfı b)Birim vuruş sinyalinin frekans tayfı	11
Şekil 3.11. Ayrık zamanlı sinyalin frekans tayfı	11
Şekil 3.12. Farklı frekanslarda örneklenmiş ayrık zamanlı sinyalin frekans tayfları	12
Şekil 3.13. Gerçek sinyallerin bant-sınırlı olmamasının örneklenmiş sinyal tayfindaki etkisi	13
Şekil 3.14. Bant-sınırlı olmayan sinyalin süzgeçlenerek bant-sınırlı hale getirilmesi	13
Şekil 3.15. Uyarlanabilir süzgeç	14
Şekil 3.16. Yüksek geçiren Sallen-Key süzgeci	15
Şekil 3.17. Uyarlanabilir analog süzgeç	16
Şekil 3.18. Melez yapıda bir uyarlanabilir süzgeç	17
Şekil 3.19. Transversal süzgeç yapısı	18
Şekil 3.20. Kafes(Lattice) süzgeç yapısı	19
Şekil 3.21. IIR süzgeç yapısı	20
Şekil 3.22. Transversal Laguerre süzgeci	21
Şekil 3.23. Bir adet w için MSE Eğrisi	23
Şekil 3.24. Gradyan kullanarak optimum süzgeç katsayılarına yakınsama	25
Şekil 3.25. En dik iniş algoritması kullanan uyarlanabilir süzgeç	25
Şekil 3.26. LMS algoritması kullanan uyarlanabilir süzgeç	26
Şekil 3.27. Kafes (Lattice) süzgeç yapısı	27

Şekil 3.28. Kafes yapıdaki uyarlanabilir süzgeç	27
Şekil 3.29. Sistem tanımlama	31
Şekil 3.30. Radyo kanalı tanımlama	32
Şekil 3.31. Ters sistem modelleme	32
Şekil 3.32. Yankı giderme	33
Şekil 3.33. Ses yankısı giderme	34
Şekil 3.34. Gürültü / Girişim giderme	35
Şekil 3.35. Ses gürültüsü giderme	35
Şekil 3.36. İleri tahmin edici	37
Şekil 3.37. Geri tahmin edici	38
Şekil 3.38. Kanal dengeleme	39
Şekil 3.39. Radyo kanalı dengeleme	39
Şekil 3.40. Tek yankılı kanal modeli	41
Şekil 3.41. Yankı sayısının ses basıncına etkisi	42
Şekil 3.42. Örnek Ses Sinyali(Metalik bir tın sesi)	43
Şekil 3.43. FIR modellenmiş kanal ile yankı eklenmesi	44
Şekil 3.44. Birden fazla yankılı sistemin IIR yapıda modellenmesi	44
Şekil 3.45. IIR modellenmiş kanal ile çoklu yankı eklenmesi	45
Şekil 3.46. r_{xx} :Giriş sinyalinin özilintisi	47
Şekil 3.47. r_{yx} : $y(n)$ ile $x(n)$ 'nin çapraz-ılintisi	47
Şekil 3.48. r_{xy} : $x(n)$ ile $y(n)$ 'nin özilintisi	48
Şekil 3.49. r_{yy} :Çıkış sinyali özilintisi	48
Şekil 3.50. Çok yankılı sinyalin özilintisi	49
Şekil 3.51. $r_{hh}(\tau)$: Kanal tepkesinin özilintisi	51
Şekil 3.52. Kanal dengeleme öbek çizgesi	53
Şekil 3.53. a) Minimum fazlı sistem b) Maksimum fazlı sistem	55
Şekil 3.54. Tek yankılı bir sistemin FIR yapıda modellenmesi	55
Şekil 3.55. FIR modellenmiş kanalın IIR süzgeç ile dengelenmesi	57
Şekil 3.56. IIR dengeleyici	57
Şekil3.57. FIR modellenmiş kanalın FIR süzgeç ile dengelenmesi	57
Şekil 3.58. FIR modellenmiş kanal için FIR yapıdaki dengeleyici	59
Şekil 3.59. IIR modellenmiş kanalın FIR süzgeç ile dengelenmesi	60
Şekil 3.60. IIR modellenmiş kanal için FIR yapıdaki dengeleyici	60
Şekil 3.61. Uyarlanabilir süzgeç kullanarak kanal dengeleme	60
Şekil 3.62. Nedenselliği korumak için $d(n)$, $x(n)$ ' nin gecikmiş olarak aktarılır	61

Şekil 3.63. Bir sistemin blok diyagram gösterimi	61
Şekil 3.64. Kör ters katlama	64
Şekil 3.65. Ters katlama akış çizgesi	66
Şekil 4.1. Tek yankılı sinyal	67
Şekil 4.2. Asıl ve yankısız sinyal(FIR-IIR dengeleme)	68
Şekil 4.3. Yankılı ve yankısız sinyal(FIR-IIR dengeleme)	68
Şekil 4.4. Hata sinyali (FIR-IIR dengeleme)	69
Şekil 4.5. Asıl ve yankısız sinyal(FIR-FIR dengeleme)	70
Şekil 4.6. Yankılı ve yankısız sinyal(FIR-FIR dengeleme)	70
Şekil 4.7. Hata sinyali(FIR-FIR dengeleme)	71
Şekil 4.8. Asıl ve Yankısız sinyal (IIR-FIR dengeleme)	72
Şekil 4.9. Yankılı ve yankısız sinyal (IIR-FIR dengeleme)	72
Şekil 4.10. Hata sinyali	73
Şekil 4.11. LMS algoritması ile kanal dengeleme($\mu=0.5$, $N=10$)	74
Şekil 4.12. Yankılı ve yankısız sinyaller ($\mu=0.5$, $N=10$)	75
Şekil 4.13. Hata sinyali ($\mu=0.5$, $N=10$)	75
Şekil 4.14. Farklı süzgeç uzunlukları için çıkış sinyalleri	77
Şekil 4.15. Farklı N değerlerinde oluşan hata sinyalleri	78
Şekil 4.16. Farklı μ değerleri için çıkış sinyalleri	79
Şekil 4.17. Farklı μ değerlerinde oluşan hata sinyalleri	80
Şekil 4.18. LMS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)	81
Şekil 4.19.LMS ile yankısı giderilmiş çok yankılı sinyal(Yankılı ve yankısız sinyaller)	81
Şekil 4.20. Hata sinyali ($\mu=0.5$, $N=10$)	82
Şekil 4.21. NLMS algoritması ile kanal dengeleme($\mu_0=0.5$, $N=10$)	83
Şekil 4.22. Yankılı ve yankısız sinyaller ($\mu_0=0.5$, $N=10$)	83
Şekil 4.23. Hata sinyali ($\mu_0=0.5$, $N=10$)	84
Şekil 4.24. Farklı süzgeç uzunlukları için çıkış sinyalleri	85
Şekil 4.25 Farklı N değerlerinde oluşan hata sinyalleri	86
Şekil 4.26. Farklı μ değerleri için çıkış sinyalleri	87
Şekil 4.27. Farklı μ değerlerinde oluşan hata sinyalleri	88
Şekil 4.28. NLMS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)	89
Şekil 4.29. NLMS ile yankısı giderilmiş çok yankılı sinyal(yankılı ve yankısız sinyaller)	90

Şekil 4.30. Hata sinyali ($\mu_0=0.5$, $N=10$)	90
Şekil 4.31. RLS algoritması ile kanal dengeleme ($N=10$)	91
Şekil 4.32. Yankılı ve yankısız sinyaller ($N=10$)	92
Şekil 4.33. Hata sinyali ($N=10$)	92
Şekil 4.34. Farklı süzgeç uzunlukları için çıkış sinyalleri	93
Şekil 4.35. Farklı N değerlerinde oluşan hata sinyalleri	94
Şekil 4.36. RLS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)	95
Şekil 4.37. RLS ile yankısı giderilmiş çok yankılı sinyal(yankılı ve yankısız sinyaller)	95
Şekil 4.38. Hata sinyali ($N=10$)	96
Şekil 4.39. konferans.m fonksiyonunun akış çizgesi	97
Şekil 4.40. Uzak uç ve yakın uç sinyalleri	98
Şekil 4.41. Yankılı ve yankısız sinyaller	98
Şekil 4.42. Oluşan hata sinyali	100

TABLOLARIN LİSTESİ

Tablo 4.1: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı	76
Tablo 4.2: Farklı μ değerleri için ilinti katsayısı ve hata karelerin toplamı	76
Tablo 4.3: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı	89
Tablo 4.4: Farklı μ değerleri için ilinti katsayısı ve hata karelerin toplamı	89
Tablo 4.5: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı	94

1.GİRİŞ

Ses sinyali mikrofon ile alınırken, asıl sinyale ek olarak yankılar da alınır. Yankı, ses sinyalinin anlaşılmasını zorlaştırır ve dinleyicileri rahatsız eder. Bu nedenle yankıların mümkün olduğunca giderilmesi gerekir. Yankı yolu bir kanal/sistem gibi düşünülerek modellenmesi yapılır ve süzgeçler kullanılarak yankı giderilir. Yankı yolu durağan yapıda ise bu işlem daha kolay yapılır. Bir deneme sinyali hoparlörden verilir ve ortamda bulunan bir mikrofon tarafından tekrar kaydedilir. Alınan sinyal, asıl sinyale ek olarak yankıları da içerir. Giriş ve çıkış sinyalleri belli olduğu için ters katlama kullanılarak yankı yolu sistemi belirlenebilir. Daha sonra bu ortamda kaydedilen ses sinyallerinin yankılarını gidermek için yankı yolu sisteminin tersi kullanılarak kanal dengeleme yapılır ve ses sinyali yankılardan arındırılır. Ancak çoğunlukla yankı yolu devingen yapıdadır ve zamanla değişir. Bu nedenle kanalı modellemek veya dengelemek için uyarlanabilir süzgeçlerin kullanılması gerekir. Ayrıca kör sinyal işleme uygulamalarında mutlaka uyarlanabilir süzgeçler kullanılmalıdır. Bu nedenlerle ses yankısının giderilmesi için uyarlanabilir süzgeç kullanımı tercih edilmiştir.

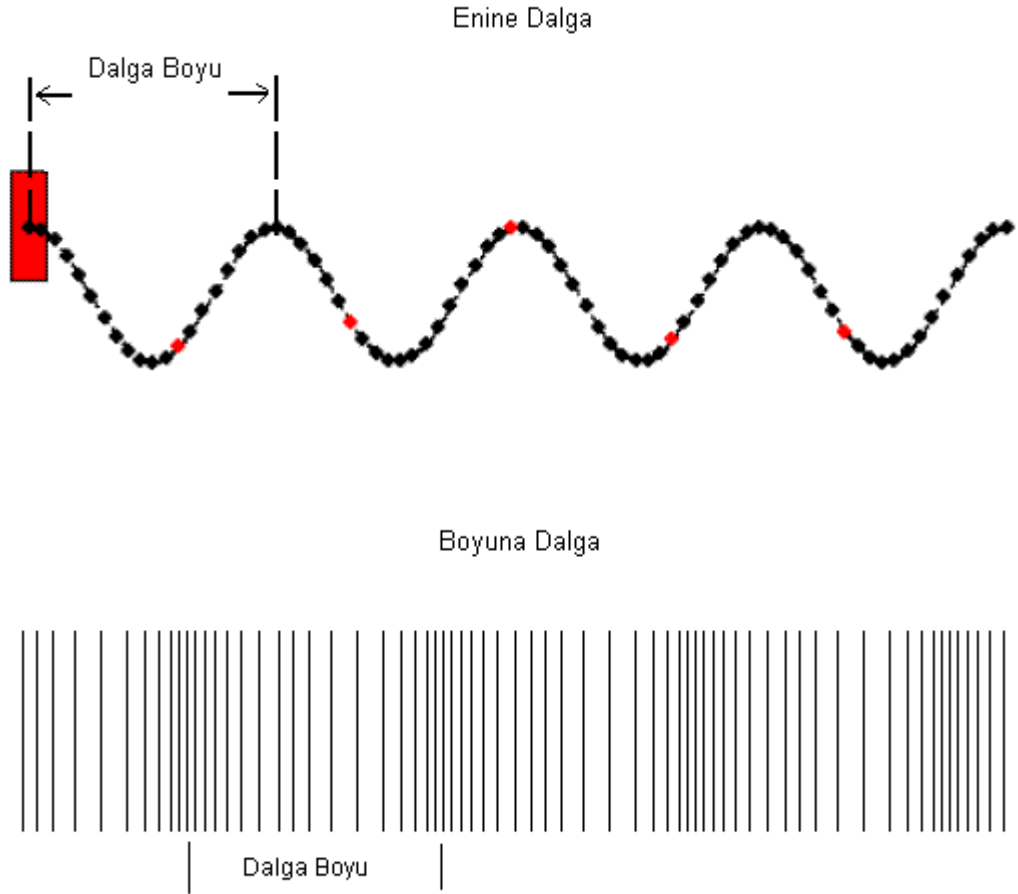
2. LİTERATÜR ÖZETİ

Uyarlanabilir süzgeçlerin temelleri, Wiener ve Hopf [1] tarafından bulunan ve kendi adlarıyla bilinen denklemlere dayanmaktadır. Daha sonra Kolmogorov[2], Krein[3] ve Wiener[4] tarafından en küçük ortalama kare kestirimler üzerine çalışmalar yapılmıştır. Widrow [5] , LMS algoritmasının karmaşık yapıdaki biçimini ilk olarak incelemiştir. 1960'ların başlarına kadar telefon kanallarının dengelenmesinde sabit dengeleyiciler ve katsayıları elle ayarlanan dengeleyiciler kullanılmıştır. 1965'te , Lucky bir transversal süzgecin katsayılarının otomatik olarak ayarlanması için sıfıra-zorlama algoritmasını geliştirmiştir[6]. LMS kriterini kullanarak, Gersho[7] ve Proakis ve Miller[8] uyarlamalı dengeleme problemini yeniden formüle etmişlerdir.

3. MATERYAL VE METOT

3.1. Ses ve Ses Dalgaları

Ses, madde içinde yayılan mekanik titreşimlerdir. Ses dalgalar halinde ilerler. Dalga, uzay veya uzay-zamanda yayılan ve sıklıkla enerjinin taşınmasına yol açan titreşime verilen isimdir. Bir ortamdaki parçacıkların, denge durumu etrafındaki titreşimleri sonucu oluşan dalgalara mekanik dalgalar denir. Dalgalar hareket biçimine göre enine ve boyuna dalgalar olarak ikiye ayrılır. Enine dalgalarda titreşim hareket yönüne dik iken boyuna dalgalarda hareket yönü ile aynı düzlemedir. Şekil 3.1 de enine ve boyuna dalga biçimleri gösterilmiştir.



Şekil 3.1. Enine Dalga ve Boyuna Dalga[9]

Şekil 3.2. 'de bir yay üzerindeki boyuna ve enine dalgalar gösterilmiştir.



Şekil 3.2. Bir yay üzerinde ilerleyen enine ve boyuna dalgalar[10]

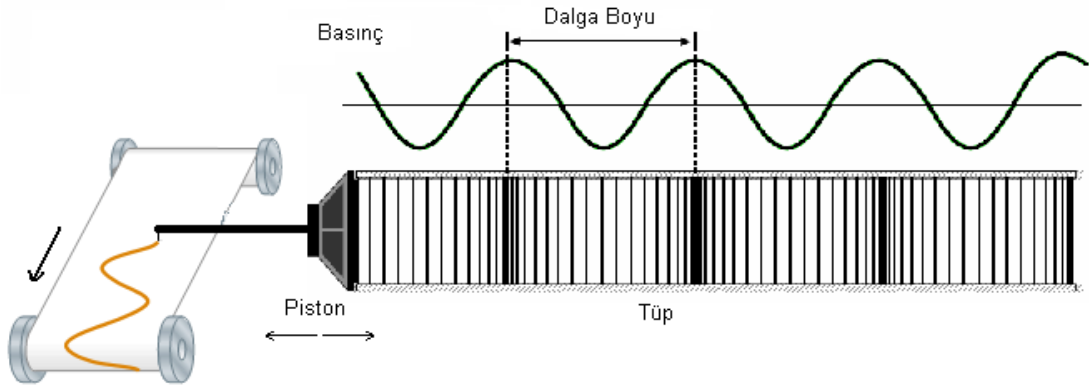
Ses dalgaları boyuna dalgalardır. Ses dalgaları ortamda yayılırken, ortamın parçacıkları, dalganın hareket doğrultusu boyunca yoğunluk ve hacim değişiklikleri üreterek yayılır. Bu değişiklikler yüksek ve alçak basınç bölgelerinin oluşumuna yol açar. Hava göz önüne alındığında, ses bir basınç dalgası olarak yayılır. Sıkıştırma hava içinde yayılır. Burada havadaki moleküller değil sadece basınç iletilir. Hava parçacıkları sadece birkaç mikron ileri ya da geri gider.

Ses dalgaları 20°C sıcaklıktaki havada yaklaşık saniyede 344 metre hızla ilerler. Havada sesin hızı sıcaklığa, hava basıncına, hava yoğunluğuna bağlı olarak değişir. Ses hızı frekansa bağlı olarak değişmez. Her frekansta ses aynı hızla ilerler. Sesin havada ki yayılma hızı

$$V_{\text{ses}} = 331.5 + 0.6T_c \quad (3.1)$$

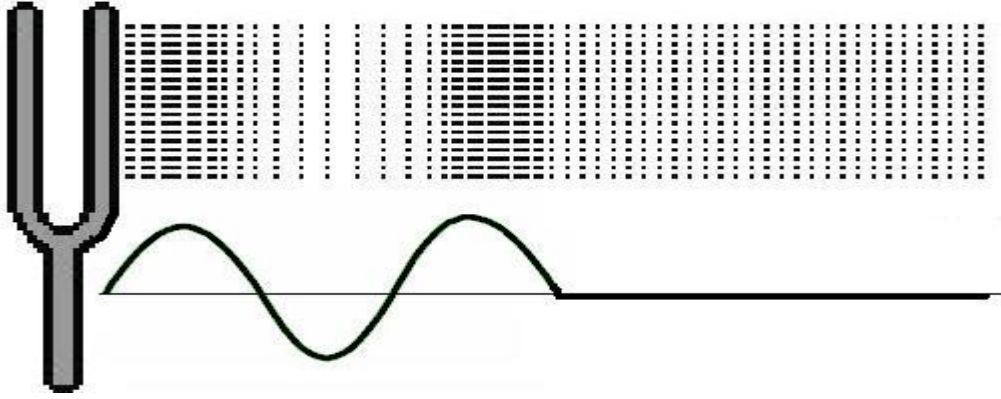
biçiminde ifade edilir. Burada T_c , santigrad ($^{\circ}\text{C}$) cinsinden sıcaklıktır.

Ses kaynağının hareketi sinüs biçimli ise havada ses dalgasından oluşan basınç değişimi de sinüs biçimli olur. Bu değişim Şekil 3.3.' de gösterilmiştir.

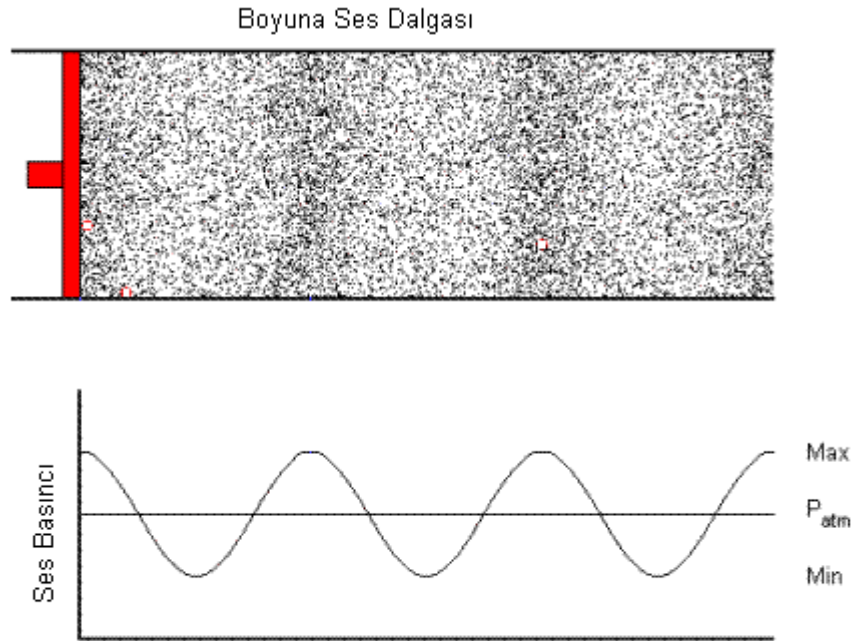


Şekil 3.3. Bir tüp içinde ses dalgasının oluşturduğu basınç değişimi[11]

Sıfır seviyesi normal durumdaki basınç seviyesini gösterir. Bu seviyenin üstündeki basınç bölgelerine normalden daha fazla molekül toplanırken alçak basınç bölgelerinde daha az molekül toplanır. Bu durum Şekil 3.4. ve Şekil 3.5.'de gösterilmiştir.



Şekil 3.4. Farklı basınç bölgelerindeki molekül yoğunluğu



Şekil 3.5 Ses basıncının molekül yoğunluğuna etkisi[9]

Ses, kulak zarıyla temasta bulunan havanın basıncının değişimi ile algılandığından, bir ses kaynağının belli bir noktada yarattığı ses basıncı seviyesi önemlidir. Ses basıncı, bir ses kaynağından çıkan ses dalgasının 1 m^2 alanda meydana getirdiği kuvvettir. Ses basıncı P ile gösterilir ve birimi Pascal veya Bar'dır. Ses basınç

seviyesi

$$P = 10 \log \left(\frac{P}{P_0^2} \right) = 20 \log \frac{P}{P_0} \quad (3.2)$$

formülü ile belirlenir. Burada

P : Ölçülen ses basıncı

P_0 : Duyma alt sınırını ifade eden referans ses basıncıdır (20 μ Pa).

Kulak, 20 μ Pa (duyma alt eşiği) ile 20 Pa aralığındaki ses basınçlarını algılar. Ses madde içinde ilerlerken enerjisini kaybeder ve sönümlenir. Bu durum Şekil.3.6.' da gösterilmiştir. Belli mesafede ölçülen ses basıncı belirlenmiş ise, uzaklık değiştiğinde ölçülecek ses basıncı

$$P_{\text{yeni}} = P_{\text{ref}} + 20 \log \left(\frac{d_{\text{ref}}}{d_{\text{yeni}}} \right) \quad (3.3)$$

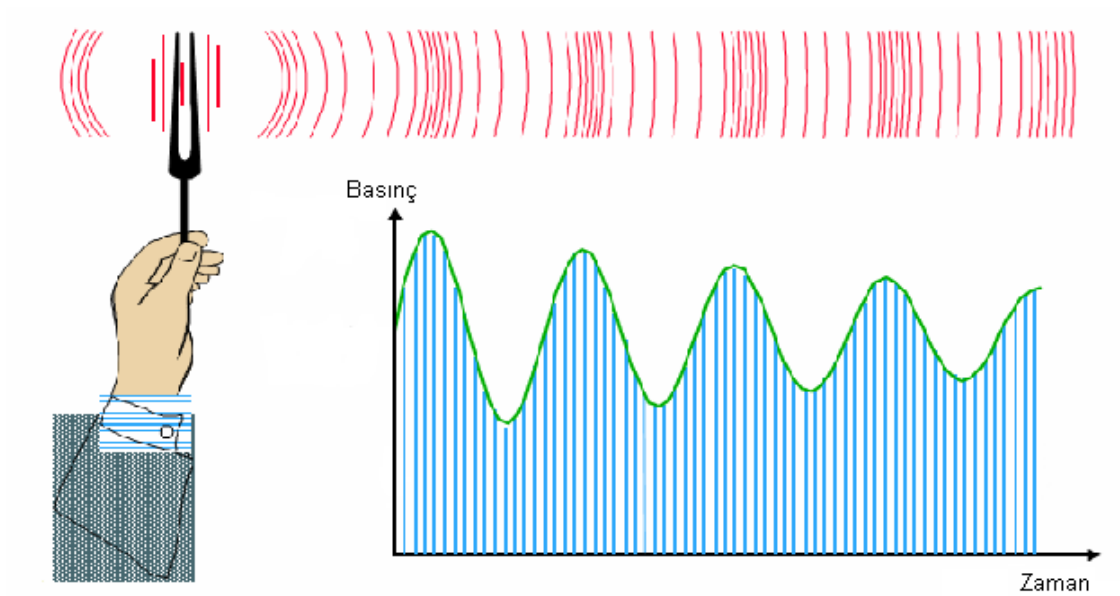
biçiminde hesaplanır. Burada

P_{yeni} : Yeni mesafedeki ses basınç seviyesi (dB)

P_{ref} : Referans seviyesindeki ses basınç seviyesi (dB)

d_{yeni} : Yeni mesafe (m)

d_{ref} : Referans mesafesidir (m).



Şekil 3.6. Ses dalgasının zayıflaması[12]

3.2. Ses Sinyali(Audio)

Ses dalgalarının, elektriksel sinyale dönüştürülmüş haline ses sinyali (audio) denir. Sesin kaydedilmesi, tekrar dinlenmesi ve üzerinde çeşitli işlemler yapılabilmesi için elektriksel sinyale dönüştürülmesi gereklidir.

Ses dalgaları havada ilerlerken alçak ve yüksek basınç alanları oluşturur. Ses sinyaline dönüştürülen bu basınç değişimleridir. Ses dalgalarını elektriksel ses sinyaline dönüştüren elemana mikrofon denir. Bütün mikrofonlar ses dalgalarına tepki gösteren çeşitli şekillerde yapılmış diyafram ya da benzeri bir elemana sahiptir. Mikrofona gelen ses dalgaları diyaframa çarpar ve ses basıncındaki değişikliklere göre diyafram içe veya dışa doğru hareket ederek mekanik titreşim yapar. Bu titreşimler sonucunda mikrofonun çıkış uçlarında bir gerilim meydana gelir. Bu şekilde ses sinyali elde edilir. Ses sinyali analog veya sayısal biçimde kaydedilip kullanılabilir. Günümüzde ses sinyallerinin işlenebilmesi ve sıkıştırılabilmesi için sayısal biçime dönüştürme tercih edilmektedir. Sayısal ses sinyali üzerinde yapılan bazı işlemler, yankı giderme, gürültü giderme, sıkıştırma, ses tanıma, konuşma tanıma vb. olarak sayılabilir.

3.2.1. Sayısal Ses Sinyali

Analog bir sinyalin sayısala dönüştürülmesi için yapılan işlemler

- Örnekleme
- Nicemleme (Kuantalama)
- Kodlama

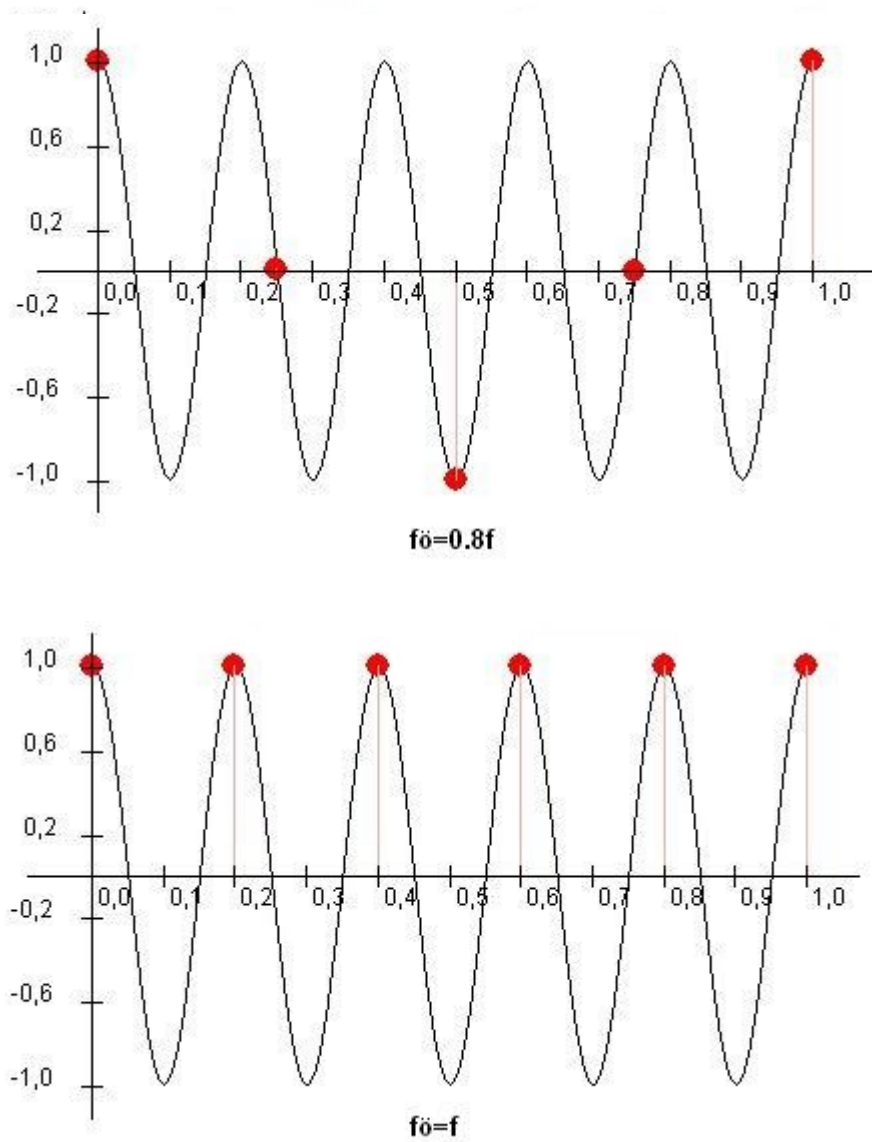
şeklindedir.

Örnekleme, analog sinyalden belirli zaman aralıklarında örnek alınması işlemidir. Örnekleme ile sürekli zamanlı sinyal ayrık zamanlı sinyale dönüştürülür. Örnekleme işlemi tek başına analog bir sinyalin sayısal bir sinyale dönüştürülmesi için yeterli değildir. Bunun nedeni örneklenmiş sinyalin sürekli bir değer aralığını kapsamaması ve dolayısıyla ayrık zamanlı sinyalin sonsuz sayıda değer alabilmesidir. Ayrık zamanlı sinyalin genlik değerlerinin, önceden belirlenmiş sınırlı sayıdaki genlik seviyelerinden en yakın olanına çekilmesi işlemine nicemleme(kuantalama) denir. Her nicemleme seviyesinin sayısal bir kod sözcüğü ile gösterilmesi neticesinde sayısal bir sinyal elde edilir. Analog sinyaller sayısala çevrildiğinde bilgi kaybının olmaması istenir. Bunun için analog sinyalden alınan örnek miktarının yeterince fazla olması gerekir. Ancak

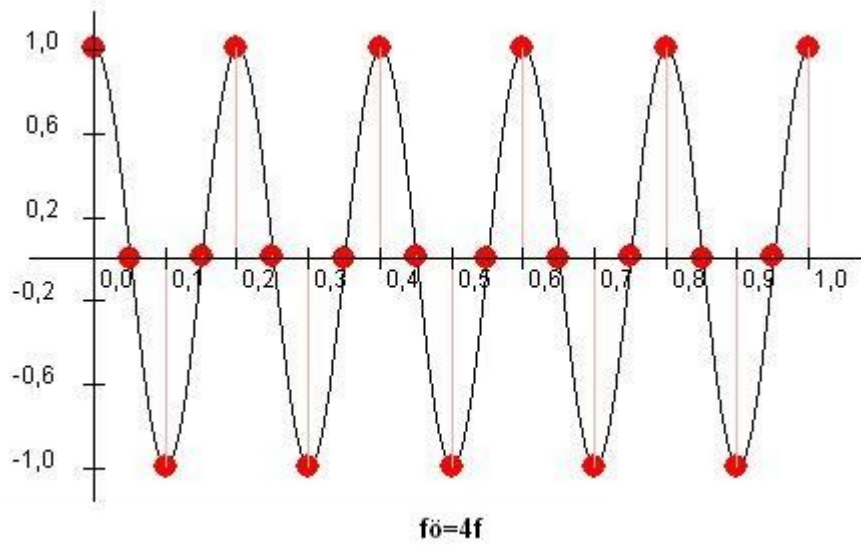
örnek sayısının fazla olması daha fazla bellek ve hesaplama gerektirir[14]. Bu nedenle örnekleme frekansının doğru seçilmesi gerekir. Bunun için Nyquist örnekleme kuramı kullanılır. Bu kurama göre, f_0 örnekleme frekansı ve f_{\max} analog sinyalin en yüksek frekans bileşeni olmak üzere;

$$f_0 > 2f_{\max} \quad (3.4)$$

seçilmelidir. Böylece analog sinyal kendi örneklerinden tekrar elde edilebilir. Aksi takdirde örtüşmeden dolayı analog sinyal tekrar elde edilemez. Şekil 3.7. 'de farklı frekanslarda örneklenmiş bir sinüs sinyali verilmiştir.



Şekil 3.7. Farklı örnekleme frekanslarının etkisi

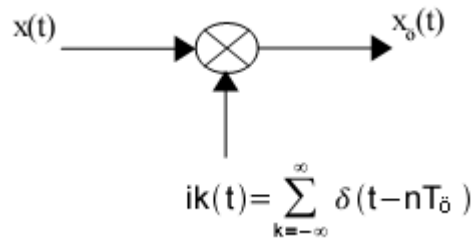


Şekil 3.7. Farklı örnekleme frekanslarının etkisi(Devamı)

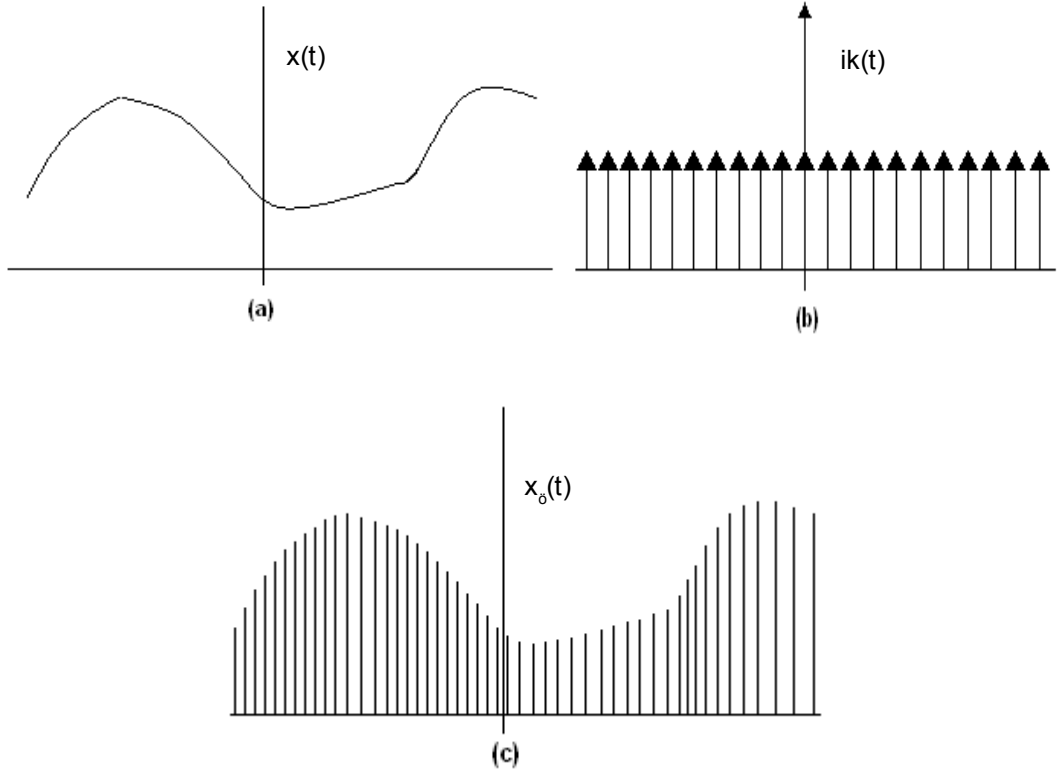
Görüldüğü gibi $f_0 < f_{\max}$ ise sinyal tekrar elde edilememekte ve bilgi kaybı olmaktadır. $f_0 = f_{\max}$ ise tekrar elde edilen sinyal bir DC sinyal olmaktadır. Bilgi kaybı olmadan sinyalin tekrar elde edilebilmesi için $f_0 \geq 2f_{\max}$ olmalıdır. Ancak uygulamada $f_0 > 2f_{\max}$ alınır. Örneklenmiş sinyal $x(n)$, analog sinyal $x(t)$ ile örnekleme zaman anlarında tanımlanmış bir birim vuruş katarının çarpımı biçiminde gösterilebilir[1]. Birim vuruş katarı

$$ik(t) = \sum_{k=-\infty}^{\infty} \delta(t - nT_0) \quad (3.5)$$

biçiminde yazılabilir. Burada δ birim vuruş sinyali ve T_0 örnekleme periyodudur.



Şekil 3.8. İdeal örneklemin matematiksel modeli

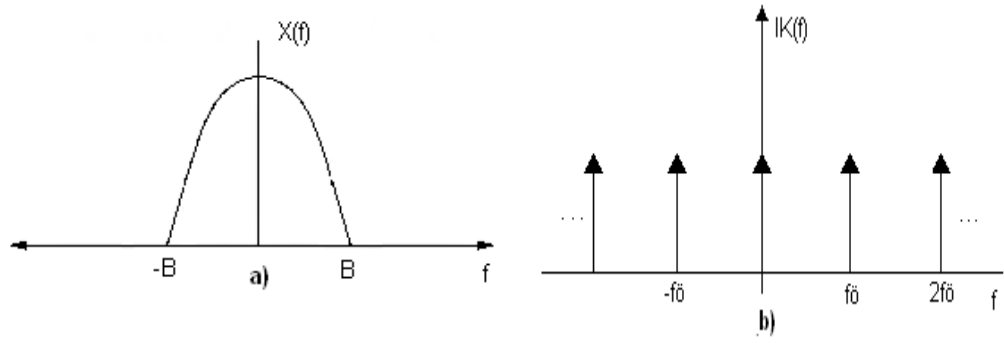


Şekil 3.9. Örnekleme a) Asıl sinyal b) Birim vuruş katarı c) Örneklenmiş sinyal

Şekil 3.8.' de ideal örnekleme için matematiksel model; Şekil 3.9.' da sürekli ve ayrık zamanlı sinyaller verilmiştir. Örnekleme frekansı önemli olduğu için sinyal frekans düzlemde de incelenmelidir. Şekil 3.10.a)' da bant sınırlı bir sinyalin frekans tayfı b)' de ise birim vuruş katarının frekans tayfı verilmiştir. Birim vuruş katarının frekans tayfı örnekleme frekansının tamsayı katlarında tanımlanmış bir birim vuruş katarı şeklindedir. Ve

$$IK(f) = \frac{1}{T_{\delta}} \sum_{n=-\infty}^{\infty} \delta(f - nf_{\delta}) \quad (3.6)$$

şeklinde tanımlanmaktadır[13].

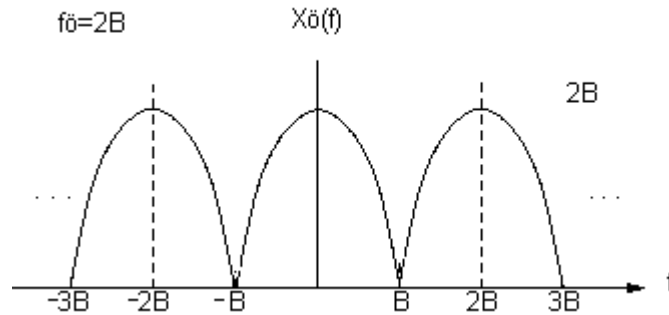


Şekil 3.10. a) x(t) sinyalinin frekans tayfı b) Birim vuruş sinyalinin frekans tayfı

Zaman düzleminde çarpma işlemi frekans düzleminde katlamaya karşılık geldiği için $x_0(t)$ sinyalinin frekans tayfı $x(t)$ 'nin frekans tayfı ile birim vuruş katarının frekans tayfının katlaması biçiminde olmaktadır. $x_0(t)$ 'nin frekans tayfı ;

$$\begin{aligned} X_0(f) &= X(f) * IK(f) = X(f) * \left[\frac{1}{T_0} \sum_{n=-\infty}^{\infty} \delta(f - nf_0) \right] \\ &= \frac{1}{T_0} \sum_{n=-\infty}^{\infty} X(f - nf_0) \end{aligned} \quad (3.7)$$

şeklinde bulunur. Şekil 3.11. 'de $x_0(t)$ 'nin frekans tayfı gösterilmiştir.

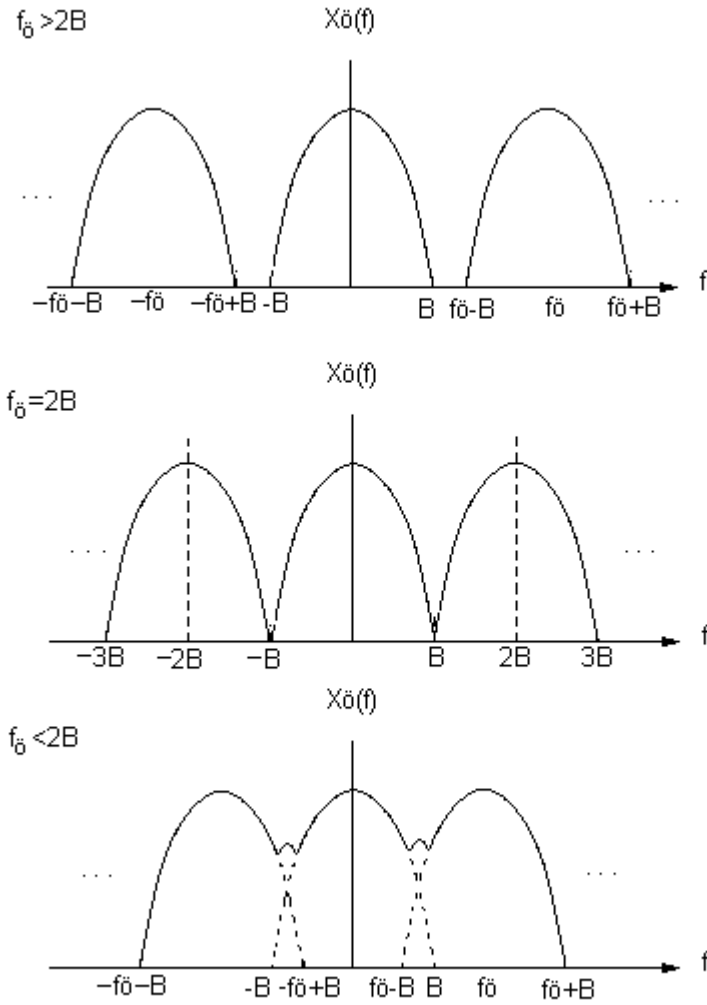


Şekil 3.11. Ayırık zamanlı sinyalin frekans tayfı

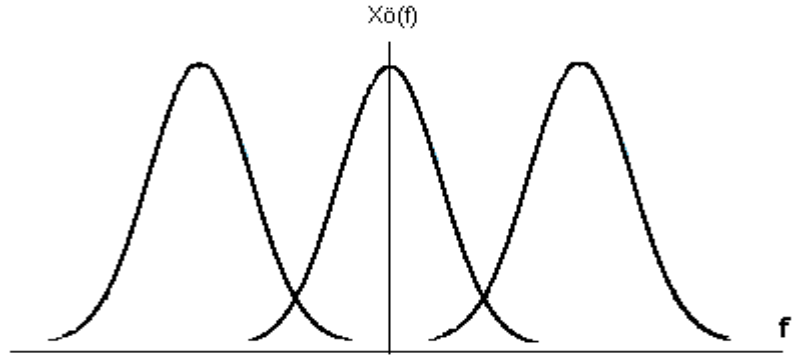
Ayrık zamanlı sinyalin frekans tayfı örnekleme frekansına bağlı olarak farklılık göstermektedir. Şekil 3.12 'de $f_0 < 2B$, $f_0 = 2B$ ve $f_0 > 2B$ örnekleme frekans aralıkları için frekans tayfı gösterilmiştir.

Şekil 3.12 'de görüldüğü gibi $f_0 < 2B$ ise frekans düzleminde örtüşme (aliasing) gerçekleşir. Bu durumda bilgi kaybı söz konusudur ve örneklenmiş sinyalden asıl sinyal tekrar elde edilemez. $f_0 = 2B$ alındığında örneklenmiş sinyalin frekans düzleminde tekrar eden bileşenleri uç uca bitişik olmaktadır.

Uygulamada kullanılan sinyaller zaman-sınırlı olduğundan bant-sınırlı olamazlar. Çünkü bir sinyalin bant-sınırlı olabilmesi için zaman uzayında sonsuz uzunlukta olması gereklidir[14].

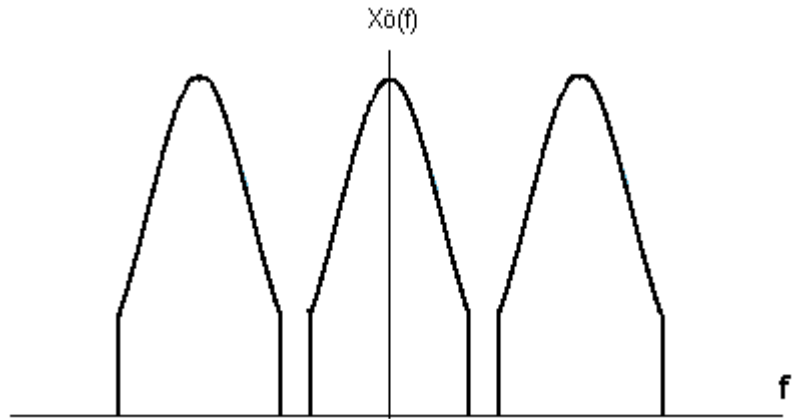


Şekil 3.12. Farklı frekanslarda örneklenmiş ayrık zamanlı sinyalin frekans tayfları



Şekil 3.13. Gerçek sinyallerin bant-sınırlı olmamasının örneklenmiş sinyal tayfındaki etkisi

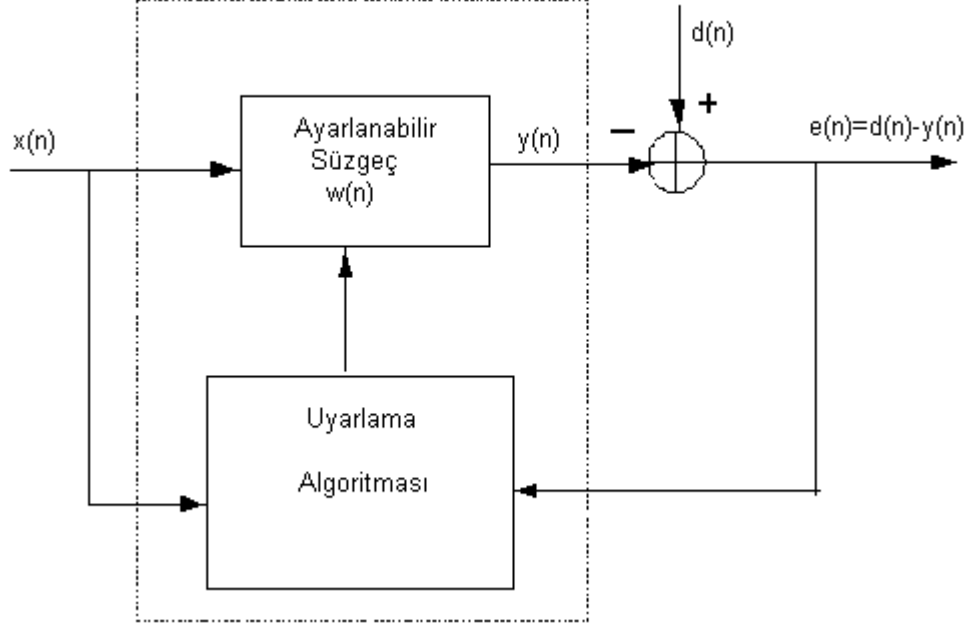
Gerçek sinyallerin bant-sınırlı olmamasının örneklenmiş sinyal tayfındaki etkisi Şekil 3.13.' de verilmiştir. Bant-sınırlı olmayan sinyaller doğrudan örneklendiğinde frekans tayfında bir örtüşme meydana gelmektedir. Bunu önlemek için analog sinyalin bant-sınırlı hale getirilmesi gerekmektedir. Örneklemeden önce analog sinyal bir alçak geçiren süzgeçten geçirilerek bant-sınırlı hale getirilir. Bu iş için kullanılan alçak geçiren süzgeç, örtüşme-önler (antialiasing) süzgeç denilmektedir[14]. Şekil 3.14.' de alçak geçiren süzgeçten geçirilmiş sinyalin frekans tayfı verilmiştir.



Şekil 3.14. Bant-sınırlı olmayan sinyalin süzgeçlenerek bant-sınırlı hale getirilmesi

3.3. Uyarlanabilir Süzgeçler

Giriş sinyalindeki değişimlere uygun olarak frekans tepkesini değiştiren süzgeçlere uyarlanabilir süzgeç (Adaptive Filter) denir[15]. Şekil 3.15 'de uyarlanabilir süzgeç yapısı verilmiştir.



Şekil 3.15. Uyarlanabilir süzgeç

Burada

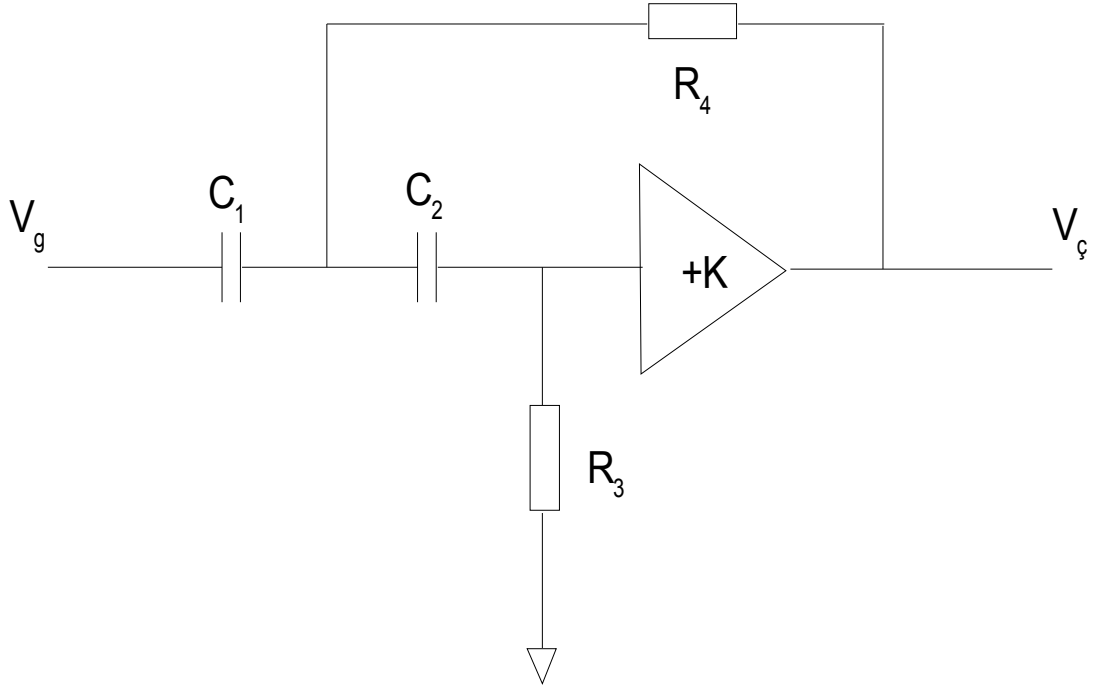
- $x(n)$:Giriş sinyali
- $y(n)$:Çıkış sinyali
- $d(n)$:Arzulanan (Referans) sinyal
- $e(n)$:Hata sinyalidir.

Uyarlanabilir süzgeç, ayarlanabilir süzgeç ve uyarlama algoritması olmak üzere iki ana kısımdan oluşur.

Ayarlanabilir süzgeçler, katsayılarının değiştirilerek frekans yanıtının değiştirilebildiği süzgeçlerdir. Süzgeç katsayıların değiştirilmesi uyarlama algoritması tarafından yapılır. Uyarlanabilir süzgeç çıkışı $y(n)$, arzulanan sinyal $d(n)$ ' den çıkartılarak hata sinyali elde edilir. Uyarlama algoritmasının görevi hata sinyali $e(n)$ i minimum yapacak şekilde süzgeç katsayılarını değiştirmektir. İdealde hata sıfır olur.

Uyarlanabilir süzgeçler sayısal, analog veya her ikisinin karışımı olan melez

yapıda olabilirler. Bu durumda sinyallerin işlenebilmesi için gerekli Analog/Sayısal ve Sayısal/Analog dönüşümlerinin yapılması gereklidir.

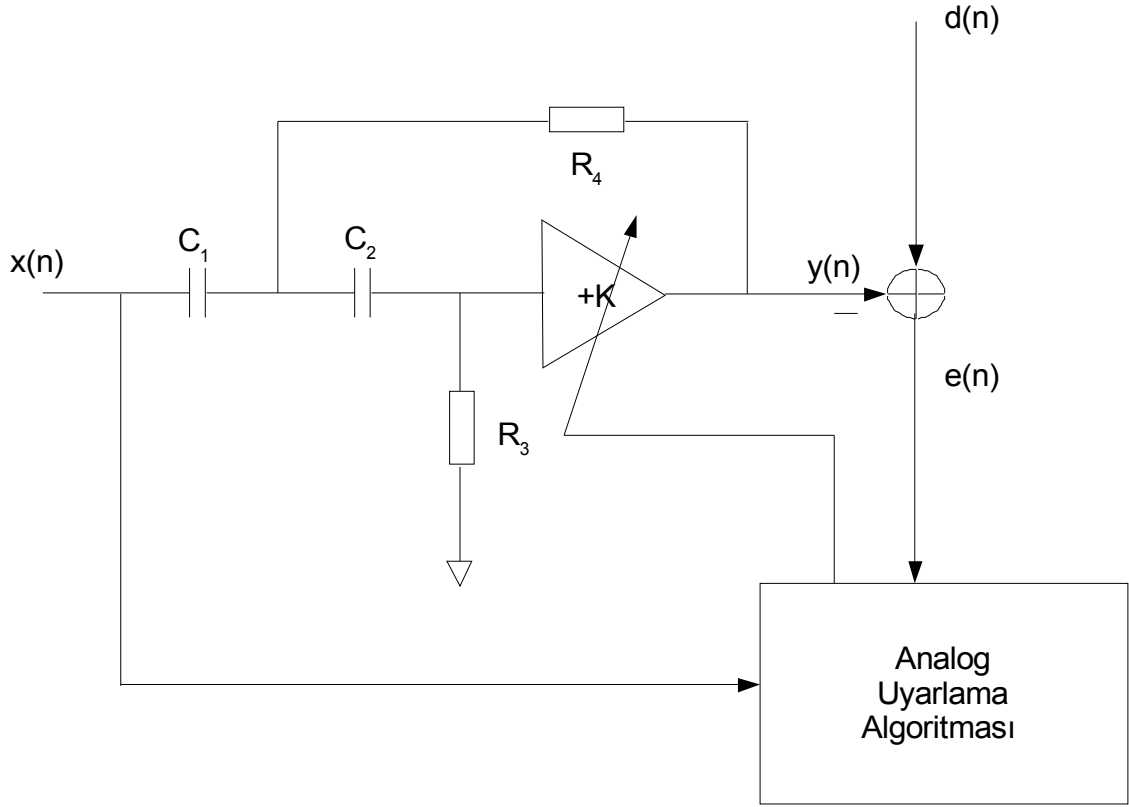


Şekil 3.16. Yüksek geçiren Sallen-Key süzgeci[16]

Şekil 3.16. 'da verilen yüksek geçiren Sallen-Key süzgecinin geçiş işlevi

$$\frac{V_{\check{c}}}{V_g} = \frac{K s^2}{s^2 + s \left(\frac{1-K}{R_4 C_1} + \frac{C_1 + C_2}{R_3 C_1 C_2} \right) + \frac{1}{R_3 R_4 C_1 C_2}} \quad (3.8)$$

şeklindedir. Burada K kazanç katı ayarlanabilir bir yükselteç yapılırsa (örneğin gerilim kontrollü yükselteç) bu durumda bir ayarlanabilir süzgeç elde edilir. K kazanç değeri değiştirilerek süzgeç tepkisi değiştirilebilir.

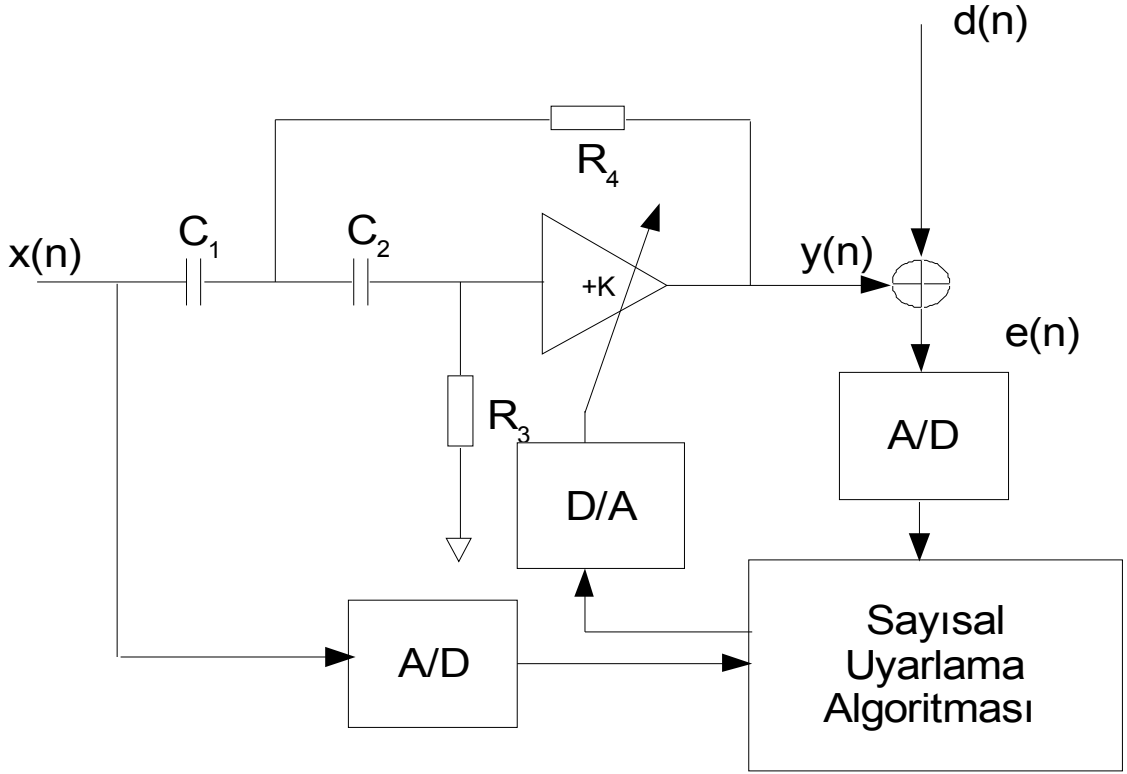


Şekil 3.17. Uyarlanabilir analog süzgeç[16]

Burada $x(n)$ giriş sinyali , $y(n)$ çıkış sinyali ve $d(n)$ arzulanan sinyal olmak üzere hata sinyali $e(n)$

$$e(n)=d(n)-y(n) \quad (3.9)$$

şeklinde hesaplanır. Uyarlama algoritması $e(n)$ ' i minimum yapacak şekilde K kazanç katını günceller. Burada hem süzgeç kısmı hemde uyarlama algoritması kısmı analog yapıda olduğundan bu süzgece uyarlanabilir analog süzgeç denir. Şekil 3.17.' de analog uyarlanabilir süzgeçlere bir örnek verilmiştir. Burada sistem çıkışı $y(n)$ sinyalidir. Eğer süzgeç kısmı analog yapıda olup algoritma kısmı sayısal olarak (mikroişlemci-DSP-mikrodenetleyici-mantık kapılar ile) gerçeklenirse bu durumda melez bir sistem tasarlanmış olur. Melez yapıdaki uyarlanabilir süzgeçlere bir örnek Şekil 3.18.' de verilmiştir.



Şekil 3.18. Melez yapıda bir uyarlanabilir süzgeç[16]

Sayısal uyarlanabilir süzgeçlerde ise hem süzgeç kısmı hemde algoritma kısmı sayısal yapıdadır.

3.3.1. Uyarlanabilir Süzgeç Yapıları

Daha önce belirtildiği gibi uyarlanabilir süzgeçler, ayarlanabilir süzgeç kısmı ve algoritma kısmı olarak iki ana kısımdan oluşur. Bu bölümde süzgeç kısmını oluşturan süzgeç yapıları anlatılmıştır.

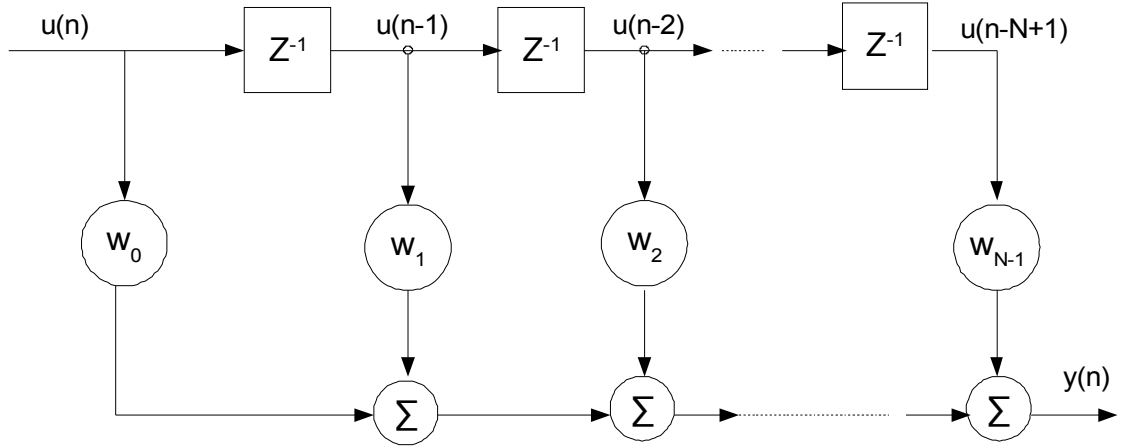
Sayısal süzgeçler darbe tepkelerine göre;

- Sonlu darbe tepkeli (FIR) süzgeçler
- Sonsuz darbe tepkeli (IIR) süzgeçler

olarak iki ana gruba ayrılırlar. FIR süzgeçlerde çıkış, sadece girişin o andaki ve eski değerlerine bağlı olup çıkışın önceki değerlerine bağlı değildir. IIR süzgeçlerde ise çıkış, giriş ve girişin gecikmişlerine ek olarak çıkışın eski değerlerine de bağlıdır. Bu geri besleme sayesinde yapılır.[17]

3.3.1.1. FIR Süzgeç Yapıları

a) Transversal Süzgeç Yapısı:



Şekil 3.19. Transversal süzgeç yapısı[18]

Şekil 3.19.' da transversal süzgeç yapısı verilmiştir. Transversal süzgeçler üç temel elemandan oluşur

- Birim gecikme elemanı
- Sabitle çarpıcı
- Toplayıcı.

Burada w katsayıları süzgeç katsayıları olup süzgecin derecesini belirlerler. N ise süzgecin derecesidir.

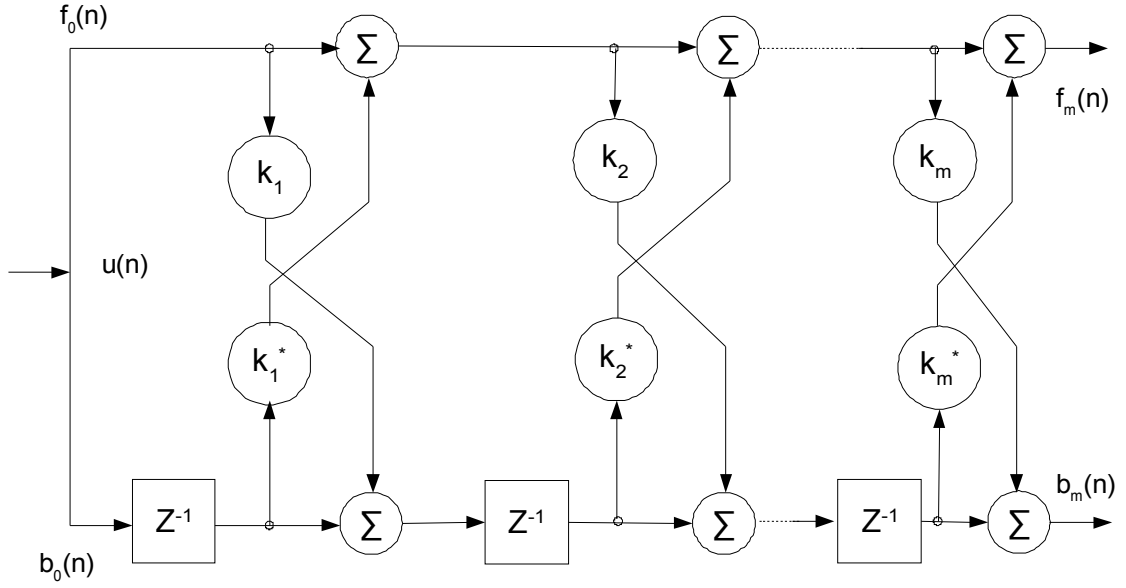
Süzgeç çıkışı

$$y(n) = \sum_{k=0}^{N-1} w_k u(n-k) \quad (3.10)$$

$$y(n) = w_0 u(n) + w_1 u(n-1) + \dots + w_{N-1} u(n-N+1) \quad (3.11)$$

şeklinde yazılabilir. (3.11) eşitliğine dikkat edilirse çıkış sadece girişin o andaki ve eski değerlerine bağlıdır. Bu nedenle darbe tepkisi sonludur. FIR yapıdaki süzgeçlere, geri besleme olmadığı için ileri yönlü süzgeçler de denilir.

b) Kafes (Lattice) Süzgeç Yapısı:



Şekil 3.20. Kafes(Lattice) süzgeç yapısı[18]

Şekil 3.20.' de kafes süzgeç yapısı verilmiş olup burada, f_m , m. ileri tahmin hatası ve b_m , m. geri tahmin hatasıdır. f_0 ve b_0 başlangıç değerleri olmak üzere süzgeç çıkışları

$$f_0(n) = b_0(n) = u(n) \quad (3.12)$$

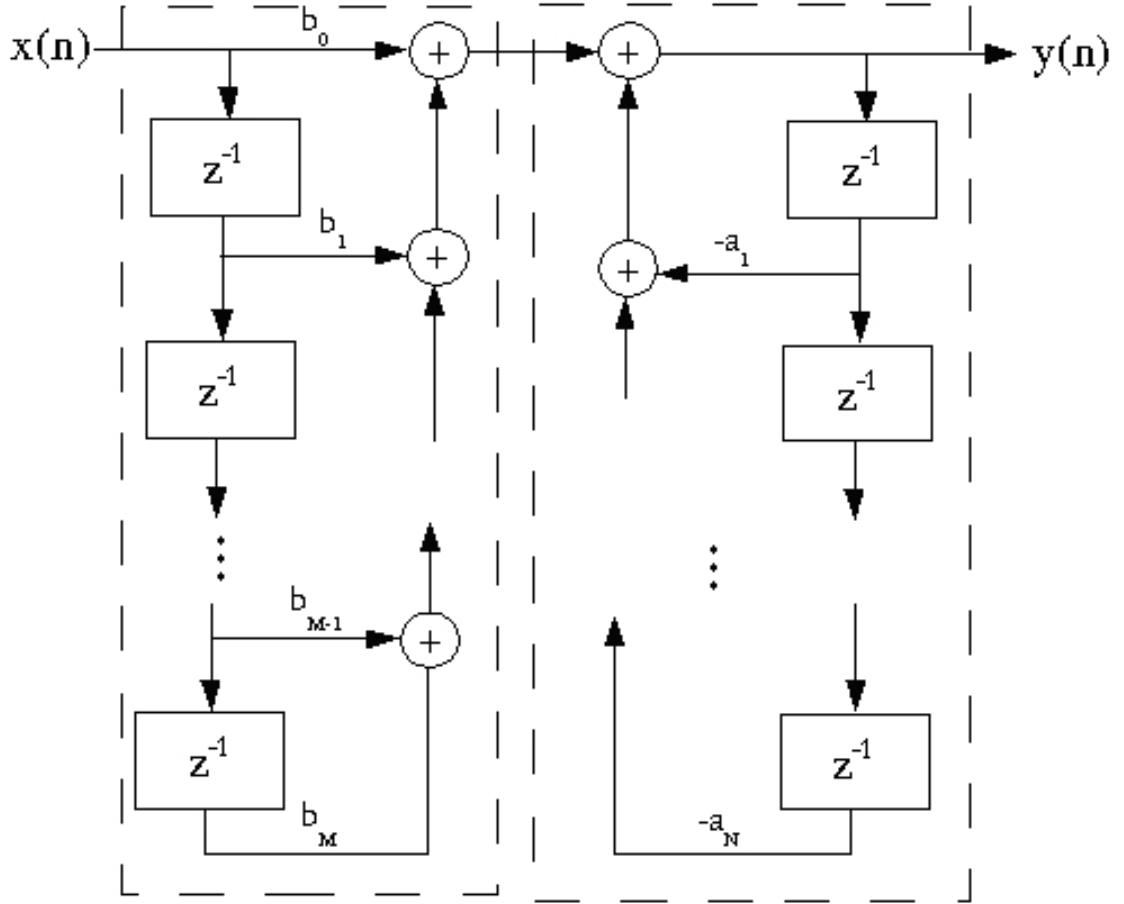
$$f_m(n) = f_{m-1}(n) + k_m^* b_{m-1}(n-1) \quad (3.13)$$

$$b_m(n) = b_{m-1}(n-1) + k_m f_{m-1}(n) \quad (3.14)$$

şeklindedir. Burada k_m^* , k_m ' nin karmaşık eşleniğidir.

3.3.1.2. IIR Süzgeç Yapıları

FIR süzgeç tepkisi sadece giriş sinyali ve onun geçmiş değerlerine bağlıdır. IIR süzgeç yapıları ise bunlara ek olarak çıkışın geçmiş değerlerine de bağlıdır. Bu geri besleme ile sağlanır. Ancak geri besleme kararsızlık ihtimalini ortaya çıkarır. FIR süzgeçler ise daima kararlıdır. Şekil 3.21.' de IIR süzgeç yapısı verilmiştir.



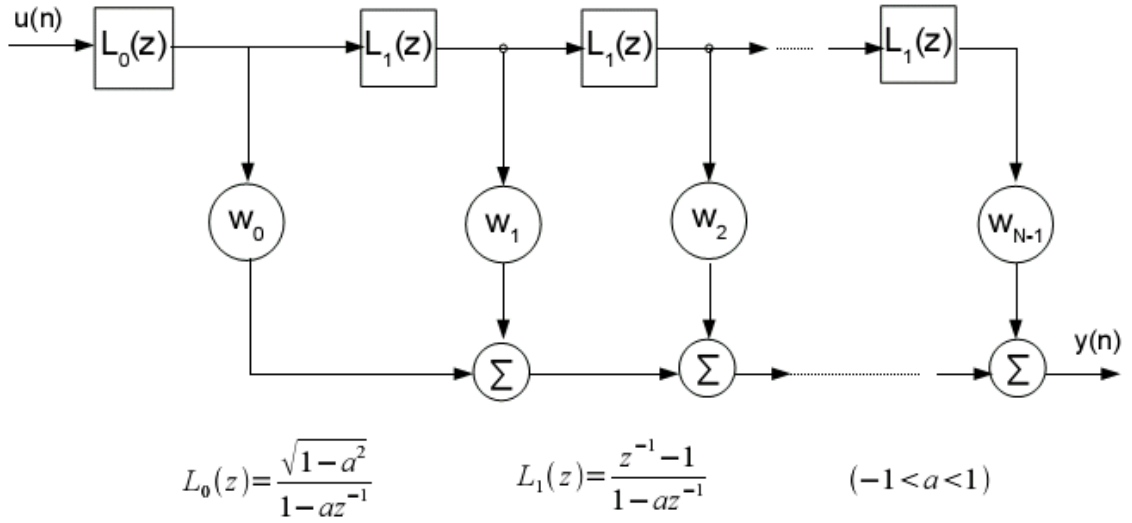
Şekil 3.21. IIR süzgeç yapısı[18]

IIR süzgecin çıkışı

$$y(n) = \sum_{i=0}^N b_i x(n-i) - \sum_{i=1}^M a_i y(n-i) \quad (3.15)$$

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_N x(n-N) - [a_1 y(n-1) + a_2 y(n-2) + \dots + a_M y(n-M)] \quad (3.16)$$

şeklindeir. Uyarlanabilir IIR süzgeçlerin kararlı olması istenir. Bu yapıdaki bir süzgeç Transversal Laguerre Süzgecidir ve yapısı Şekil 3.22. 'de verilmiştir.



Şekil 3.22. Transversal Laguerre süzgeci

Laguerre süzgeci yapı olarak ileri yönlü bir süzgece benzemektedir. Yani çıkışın önceki değerlerinin tekrar girişe verildiği geri besleme yokmuş gibi görünür. Ancak L_0 ve L_1 ' den dolayı çıkışın eski değerleride şu anki çıkışı etkiler. Bir sistemin kararlı olabilmesi için z düzleminde kutuplarının birim çemberin içinde olması gereklidir. L_0 ve L_1 ' in a noktasında birer adet kutupları vardır. Kararlılık için a sabit sayısı birim çemberin içinde olacak şekilde $-1 < a < 1$ aralığında seçilir. Bu durumda Laguerre süzgeci kararlıdır.

3.3.2. Uyarlanabilir Süzgeç Algoritmaları

Uyarlama algoritmalarının görevi hata sinyali en aza indirecek şekilde süzgeç katsayılarını değiştirmektir. Genel olarak uyarlama algoritmaları süzgeç katsayılarının eski değerlerini kullanarak yeni değerlerinin hesaplandığı yinelemeli (rekürsif) algoritmalarıdır. Uyarlama algoritmaları iki ana gruba ayrılabilir.

- Rasgele Gradyan Yaklaşımı (Stochastic Gradient Approach)
- En Küçük Kareler Kestirimi (Least Squares Estimation)

3.3.2.1. Rasgele Gradyan Yaklaşımı

Bu yaklaşımda algoritma hatanın ortalama karesini en aza indirmeye çalışır. Ortalama kare hata (Mean Square Error-MSE), hatanın karesinin beklenen değeri (Expected value) alınarak hesaplanır. Beklenen değer, istatistikte mümkün olasılıkların dağılımının aritmetik ortalaması veya kısaca ortalaması olarak tanımlanır. Beklenen değer

$$E[X] = \sum_{x=0}^n x f(x) \quad (3.17)$$

şeklinde hesaplanır. $f(x)$ frekans veya yoğunluk fonksiyonu olarak adlandırılır ve

$$f(x) = \frac{n!}{x!(n-x)!} p^x q^{n-x} \quad ; p+q=1 \quad (3.18)$$

şeklinde dir. Burada

- p : Arzulanan sonucun gelme olasılığı
- q : Arzulanan sonucun gelmeme olasılığı
- x : Belli bir sonucun gelmesini kaç defa istediğimiz
- n : Deneme sayısıdır.

Örnek olarak transversal süzgeç yapısı incelenirse hatanın karesinin beklenen değeri

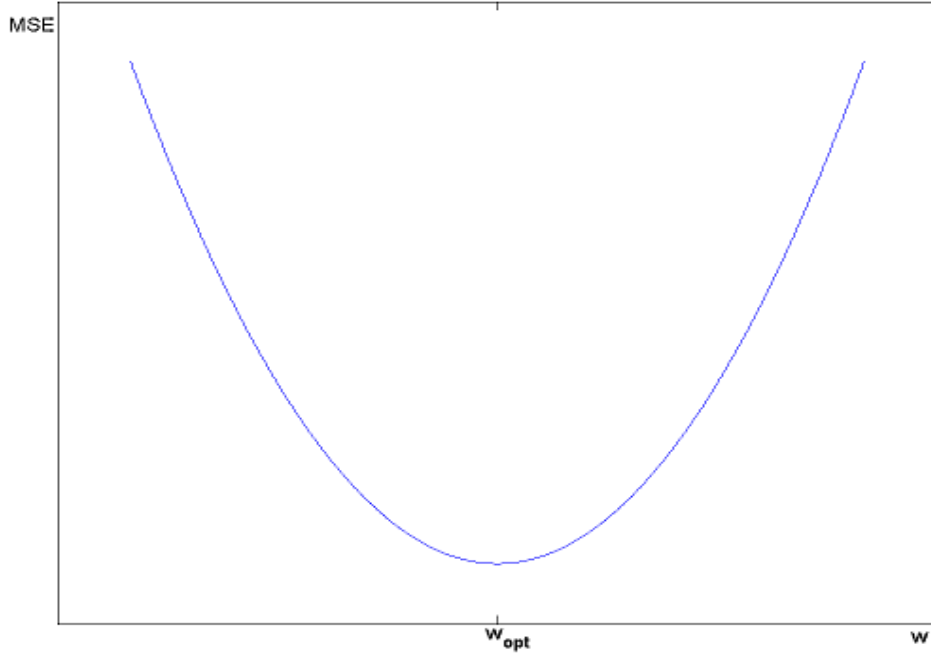
$$E[e_k^2] = E[(d_k - w^T x_k)^2] \quad (3.19)$$

$$\begin{aligned} &= E[d_k^2 - 2 w^T x_k d_k + w^T x_k x_k^T w] \\ &= E[d_k^2] - 2 w^T E[x_k d_k] + w^T E[x_k x_k^T] w \\ &= E[d_k^2] - 2 w^T P + w^T R w \end{aligned} \quad (3.20)$$

şeklinde bulunur. Burada

- $R = E[x_k x_k^T]$: Giriş vektörü $x(n)$ ' in özilinti matrisi,
- $P = E[x_k d_k]$: Giriş ile arzulanan sinyal $d(n)$ arasındaki çapraz ilinti matrisidir.

Bu yaklaşımda algoritma, (3.20) eşitliğinde verilen ortalama kare hata(MSE) değerini minimum yapacak w katsayılarını bulmaya çalışır. MSE değeri, w katsayılarının karesi ile değişmektedir. Bu nedenle tek bir genel minimumu vardır.



Şekil 3.23. Bir adet w için MSE Eğrisi[19]

Şekil 3.23.' de MSE' nin bir adet w katsayısına göre değişimi verilmiştir. Görüldüğü gibi MSE eğrisi bir adet genel minimumu olan bir eğridir. MSE' nin minimum olduğu w katsayıları optimum (en uygun) süzgeç katsayılarıdır. MSE' nin minimum olduğu noktayı bulmak için gradyanı (yönlü türevi) sıfıra eşitlenir. Bu durumda

$$\nabla_k = \frac{\partial}{\partial w} (E[e_k^2]) = -2P + 2Rw = 0$$

$$w_{opt} = R^{-1} P \quad (\text{Optimum süzgeç koşulu}) \quad (3.21)$$

eşitliği elde edilir. (3.21) eşitliğine Optimum süzgeç koşulu denir. Amaç optimum süzgeç katsayılarını bulmaktır. Uyarılama algoritmalarında, süzgeç katsayılarının şu anki değeri eski değerleri kullanılarak hesaplanır ve bunun için MSE' nin gradyanı kullanılır. Bu nedenle bu tür algoritmalara gradyan tabanlı algoritmalar denir. Bu tür

algoritmaların genel biçimi

$$w(n+1) = w(n) + \nabla_n \quad (3.22)$$

şeklindedir. ∇_n burada doğrultma amacıyla eklenmiştir.

Bu tür algoritmalarından üçü

- En dik iniş (Steepest Descent) algoritması
- En küçük ortalama kare (Least Mean Square-LMS) algoritması
- Gradyan uyarlanabilir kafes (Gradient Adaptive Lattice-GAL) algoritmasıdır.

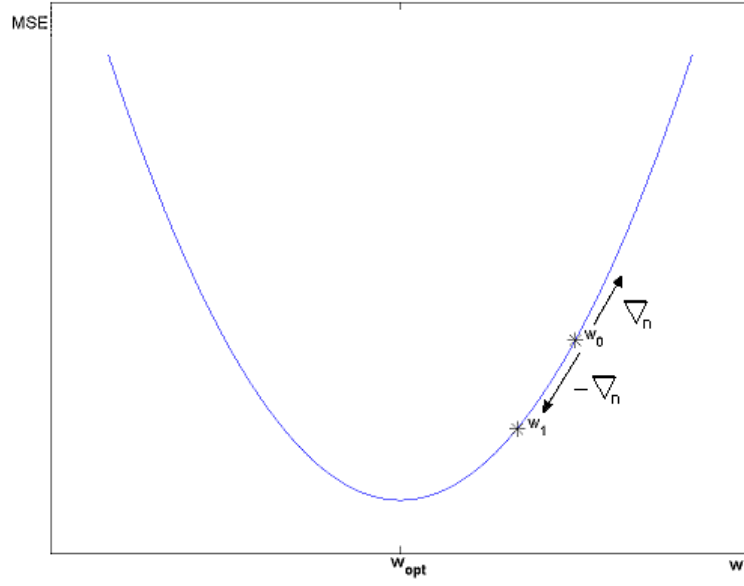
a) En Dik İniş Algoritması:

Uygulamada R ve P matrisleri tam olarak bulunamayabilir. Ayrıca hesaplamaları uzun zaman alır. Bunun yerine gradyanın kullanıldığı ve w' nın eski değerlerinden yararlanarak yeni değerlerinin bulunduğu yinelemeli hesaplama yöntemleri kullanılır. Genel biçim eşitlik 3.22.' de verilmiştir. Bir w_0 başlangıç değerinden başlanarak MSE' yi azaltacak şekilde w katsayıları ayarlanır. Gradyan yönlü türev olup yönü, değişimin en büyük olduğu yöndedir. Minimum arandığı için gradyanın tersi yönünde hareket edilmelidir. Yani gradyan negatif olarak eklenir. Bu durumda eşitliğimiz

$$w_k = w_{k-1} - \frac{1}{2} \mu \frac{\partial}{\partial w} (E[e_k^2]) \Big|_{w=w_{k-1}}$$

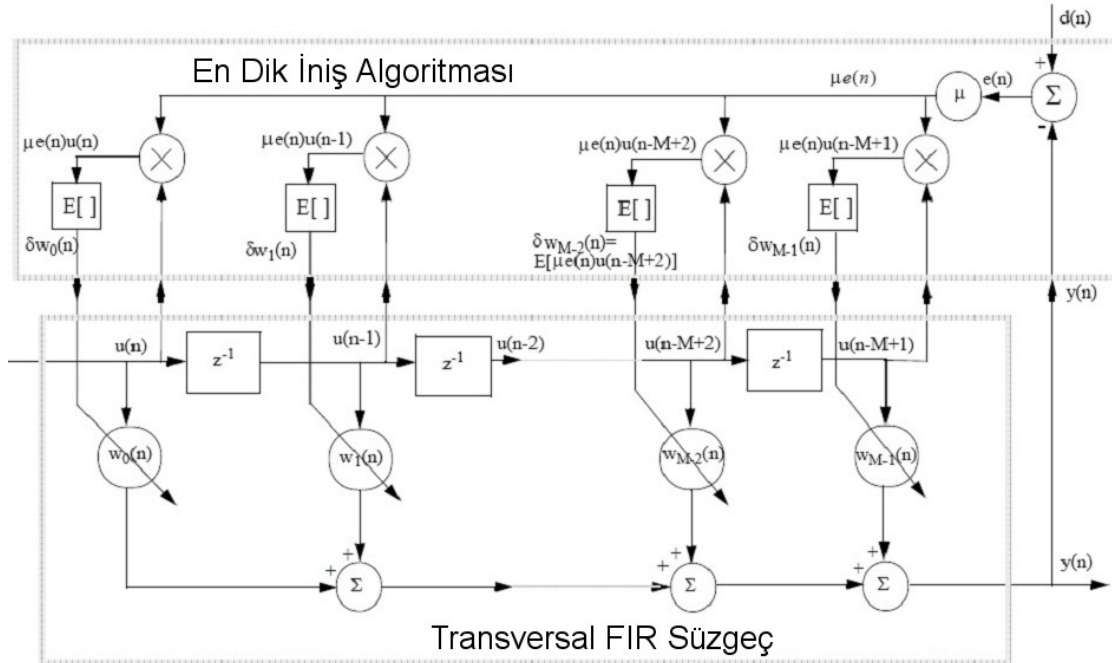
$$w_k = w_{k-1} - \frac{1}{2} \mu \nabla_{k-1} \quad (3.23)$$

şeklinde yazılabilir. μ , küçük pozitif bir sabit olup adım boyutudur ve algoritmanın hızını belirler. Şekil 3.24.' de gradyan kullanarak w_0' a nasıl yakınsandığı verilmiştir.



Şekil 3.24. Gradyan kullanarak optimum süzgeç katsayılarına yakınsama[19]

Şekil 3.25 'de transversal süzgeç yapısında ve en dik iniş algoritmasını kullanan uyarlanabilir süzgeç yapısı verilmiştir.



Şekil 3.25. En dik iniş algoritması kullanan uyarlanabilir süzgeç

b) LMS Algoritması:

LMS algoritmasında $E[e^2]$ 'nin gradyanı yerine e^2 'nin, yani hatanın karesinin gradyanı kullanılır. Bu daha kötü bir tahmin verir ancak hesaplamalar kolaylaşır. Hatanın karesinin gradyanı

$$\nabla_k = \frac{\partial}{\partial \mathbf{w}} (e_k^2) = 2 e_k \frac{\partial}{\partial \mathbf{w}} (d_k - \mathbf{w}_k^T \mathbf{x}_k) = 2 e_k (-\mathbf{x}_k) = -2 e_k \mathbf{x}_k \quad (3.24)$$

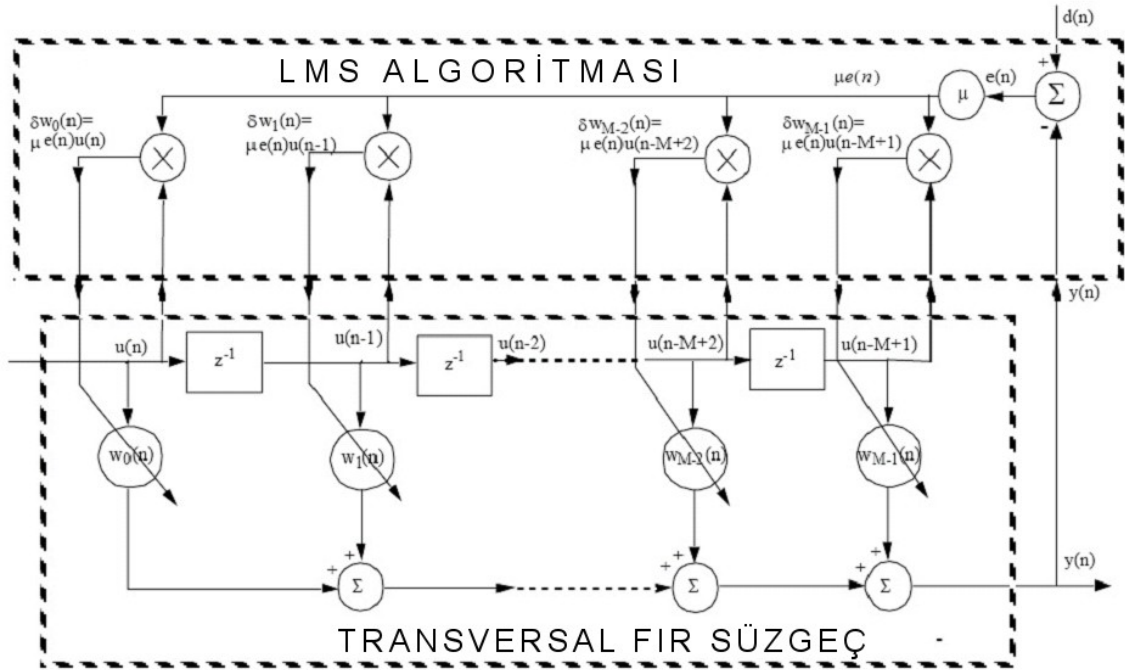
şeklindedir. Bu değer yerine konulursa

$$\mathbf{y}_k = \mathbf{w}_k^T \mathbf{x}_k \quad (3.25)$$

$$\mathbf{e}_k = \mathbf{d}_k - \mathbf{y}_k \quad (3.26)$$

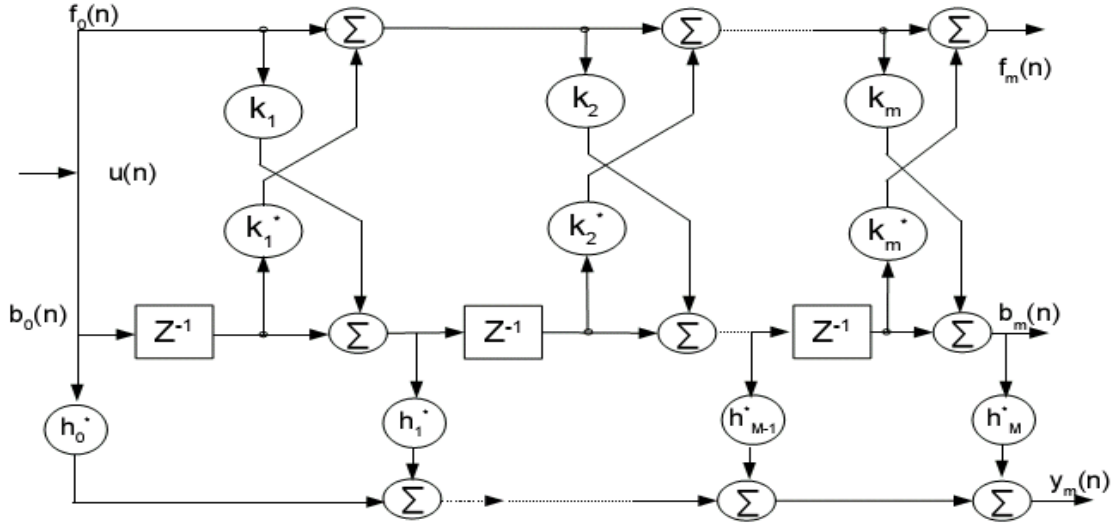
$$\mathbf{w}_{k+1} = \mathbf{w}_k + 2 \mu \mathbf{e}_k \mathbf{x}_k \quad (3.27)$$

eşitlikleri elde edilir. (3.26), (3.27), (3.28)'deki eşitlikler LMS algoritmasını oluşturur. Şekil 3.26.'da transversal süzgeç yapısında ve LMS algoritmasını kullanan uyarlanabilir süzgeç yapısı verilmiştir.



Şekil 3.26. LMS algoritması kullanan uyarlanabilir süzgeç

c) GAL Algoritması:

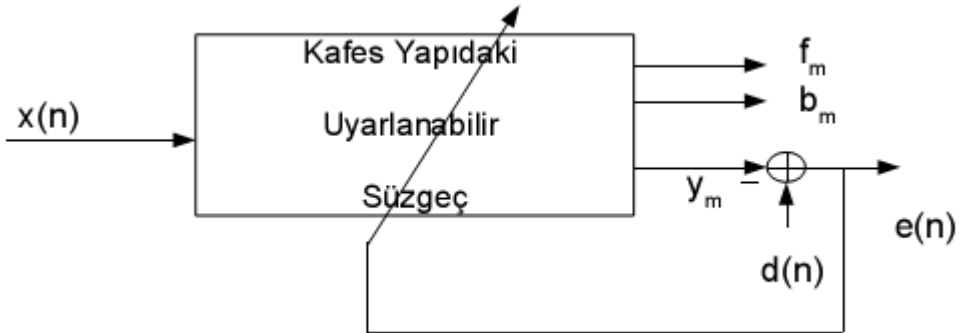


Şekil 3.27. Kafes (Lattice) süzgeç yapısı

Şekil 3.27.' de GAL algoritması kullanan kafes uyarlanabilir süzgeç yapısı verilmiştir. GAL algoritması kafes yapıdaki süzgecin k ve h katsayılarını güncellemek için kullanılır. Burada

- f_m : m . ileri tahmin hatası
- b_m : m . geri tahmin hatası
- y_m : m . süzgeç çıkışı ve arzulanan sinyal $d(n)$ ' nin tahminidir.

Kafes yapıdaki uyarlanabilir süzgeç şeması Şekil 3.28' de verilmiştir.



Şekil 3.28. Kafes yapıdaki uyarlanabilir süzgeç

f ve b katsayıları aşağıda verilen eşitlikler kullanılarak bulunur. Burada k_m^* , k_m' nin karmaşık eşleniğidir ve

$$f_0(n) = b_0(n) = u(n) \quad (3.28)$$

$$f_m(n) = f_{m-1}(n) + k_m^* b_{m-1}(n-1) \quad (3.29)$$

$$b_m(n) = b_{m-1}(n-1) + k_m' f_{m-1}(n) \quad (3.30)$$

eşitlikleri yazılabilir.

Maliyet (cost) fonksiyonunun gradyanı kullanılarak k katsayıları güncellenir. Bu fonksiyon f_m ve b_m' nin kareleri toplamının beklenen değerinin yarısıdır ve

$$J = \frac{1}{2} E[|f_m(n)|^2 + |b_m(n)|^2] \quad (3.31)$$

şeklinde yazılabilir. k katsayılarının güncellenmesi

$$\mathcal{E}_{m-1}(n) = \beta \mathcal{E}_{m-1}(n-1) + (1-\beta)(|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2) \quad (3.32)$$

$$k_m(n) = k_m(n-1) - \frac{\mu}{\mathcal{E}_{m-1}(n)} [f_{m-1}^*(n) b_m(n) + b_{m-1}(n-1) f_m^*(n)] \quad (3.33)$$

eşitlikleri ve

$$f_m(0) = b_m(0) = 0 \quad (3.34)$$

$$\mathcal{E}_{m-1}(0) = a \quad (3.35)$$

$$k_m(0) = 0 \quad (3.36)$$

başlangıç değerleri kullanılarak yapılır. Burada

- M : Süzgecin derecesi
- β : Sabit katsayı ($0 < \beta < 1$)
- μ : Sabit katsayı ($\mu < 0.1$)
- a : Küçük pozitif sabittir.

Burada $\varepsilon(n)$ ileri ve geri tahmin hatalarının toplam enerjisidir ve

$$\varepsilon_{m-1}(n) = \sum_{i=1}^n (|f_{m-1}(i)|^2 + |b_{m-1}(i-1)|^2) \quad (3.37)$$

şeklinde tanımlanır. $y_m(n)$ süzgeç çıkışıdır ve arzulanan sinyal $d(n)$ ' nin tahminidir. $y_m(n)$ ' nin $d(n)$ ' e yakınsayabilmesi için h katsayıları

$$h_m(0) = 0 \quad (3.38)$$

$$e_m(n) = d(n) - y_m(n) \quad (3.39)$$

$$y_m(n) = y_{m-1}(n) + h_m^*(n) b_m(n) \quad (3.40)$$

$$h_m(n+1) = h_m(n) + \frac{\mu}{\sum_{k=0}^m |b_k(n)|^2} b_m(n) e_m^*(n) \quad (3.41)$$

şeklinde güncellenir.

3.3.2.2. En Küçük Kareler Kestirimi

Bu yaklaşımda MSE yerine hataların karelerinin toplamı en küçük yapılmaya çalışılır. Maliyet işlevi

$$J = \sum_{i=1}^N |e(i)|^2 \quad (3.42)$$

şeklinindedir. Maliyet işlevinin en küçük değerini bulmak için yönlü türevi alınır ve sıfıra eşitlenir. Bu durumda

$$\begin{aligned} 0 &= \frac{\partial J}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_{LS}} \\ 0 &= [2\mathbf{U}^T \mathbf{U} \mathbf{w} - 2\mathbf{U}^T \mathbf{d}] \Big|_{\mathbf{w}=\mathbf{w}_{LS}} \\ \mathbf{X}_{uu} \mathbf{w}_{LS} &= \mathbf{X}_{du} \\ \mathbf{w}_{LS} &= \mathbf{X}_{uu}^{-1} \mathbf{X}_{du} \end{aligned} \quad (3.43)$$

eşitliği elde edilir. Burada

$$X_{uu} = U^T U \quad (3.44)$$

$$X_{du} = U^T d \quad (3.45)$$

şeklindedir. Amaç w katsayılarının, eşitlik (3.43)' de verilen w_{LS} değerine yakınsamasını sağlamaktır. Bu gruptaki algoritmalara örnekler

- Yinelemeli en küçük kareler (RLS) algoritması
- Yinelemeli en küçük kareler kafes (RLSL) algoritmasıdır.

RLS' de yukarıdaki düşünceden hareketle, w katsayılarının eski değerleri kullanılarak w' nın yeni değerleri bulunur. RLSL ise bu algoritmanın kafes yapısındaki süzgeçlere uyarlanmış halidir. RLS algoritmasının yakınsaması LMS' ye göre daha iyidir. Ancak daha fazla hesaplama yükü getirir.

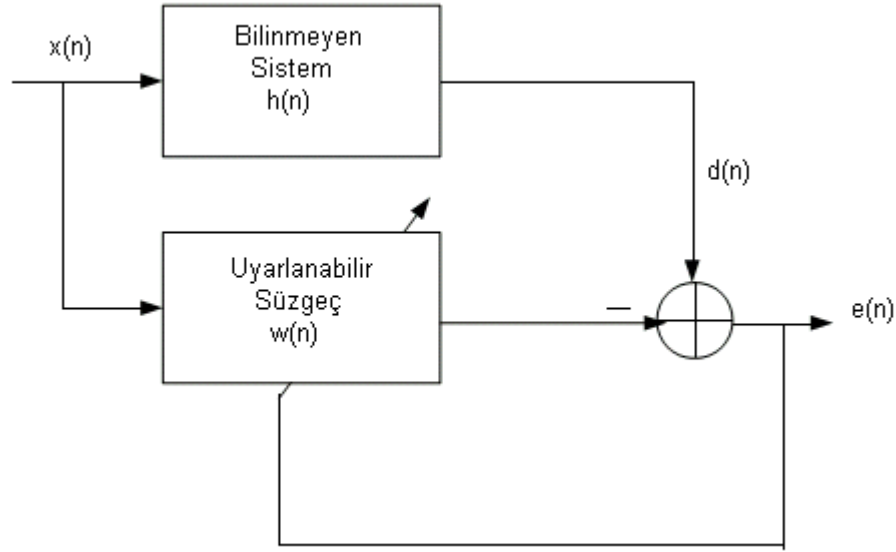
3.3.3. Uyarlanabilir Süzgeç Uygulamaları

Uyarlanabilir süzgeçlerin geniş bir uygulama alanı vardır. Bazı uygulama alanları

- Sistem/Kanal tanımlama (modelleme)
- Ters sistem modelleme
- Yankı giderme
- Gürültü/Girişim giderme
- Doğrusal tahmin
- Kanal dengeleme

şeklindedir[20].

3.3.3.1. Sistem/Kanal Tanımlama (Modelleme)



Şekil 3.29. Sistem tanımlama

Şekil 3.29.' da sistem tanıma öbek çizgesi verilmiştir. Giriş sinyali $x(n)$, hem bilinmeyen sistem girişine hem de süzgeç girişine verilir. Bilinmeyen sistem çıkışı arzulanan sinyal $d(n)$ olarak kullanılır. Hata sinyali $d(n)$ ile süzgeç çıkışı $y(n)$ arasındaki farktır. Algoritma hatayı sıfır yapacak şekilde w katsayılarını günceller. Burada

- Bilinmeyen sistemin çıkışı

$$d(n) = x(n) * h(n) \quad (3.46)$$

- Süzgeç çıkışı

$$y(n) = x(n) * w(n) \quad (3.47)$$

- Hata sinyali

$$e(n) = d(n) - y(n) \quad (3.48)$$

şeklindedir. Hata sinyali sıfır olursa;

$$e(n) = d(n) - y(n) = 0$$

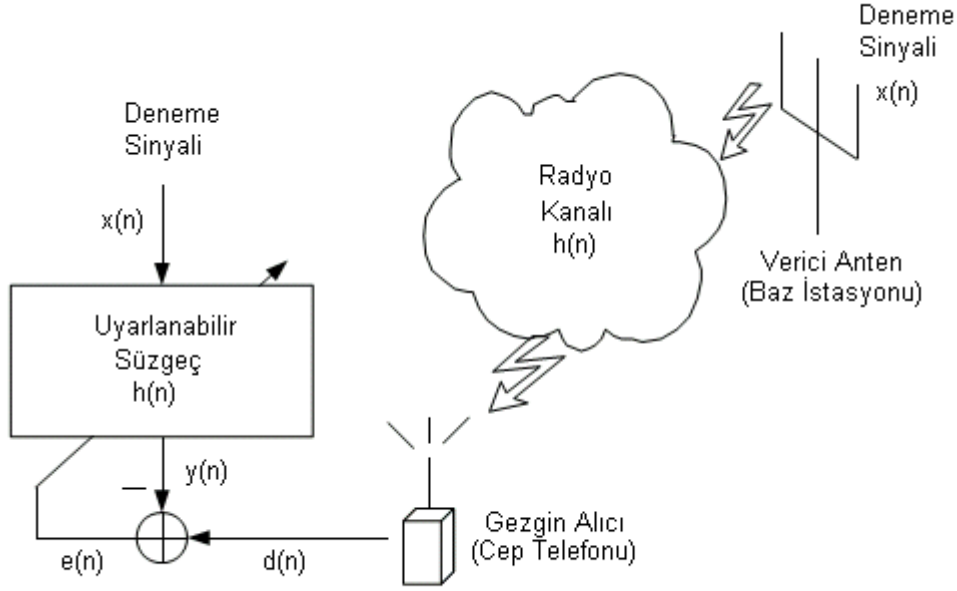
$$x(n) * h(n) - x(n) * w(n) = 0$$

$$x(n) * h(n) = x(n) * w(n)$$

$$h(n) = w(n) \quad (3.49)$$

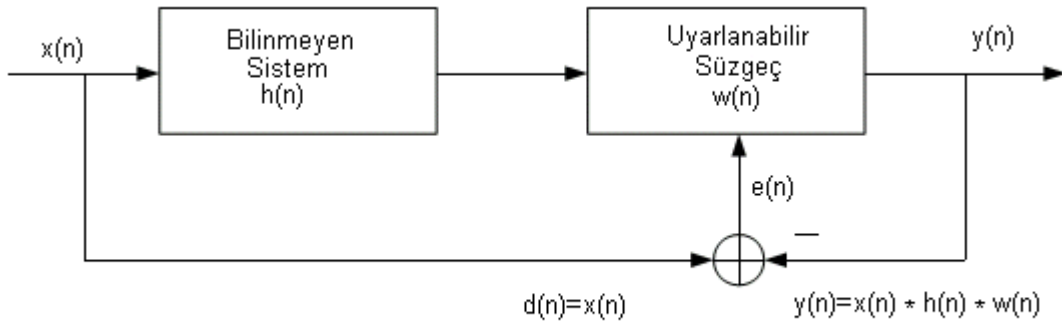
eşitlikleri yazılabilir. Hata sinyali sıfır yapıldığında, (3.49) eşitliğinde görüldüğü gibi

süzgeç katsayıları bilinmeyen sistem tepkisi $h(n)$ ' e eşittir. Bu durumda bilinmeyen sistem tanımlanmış olur. Ancak uygulamada asla hata sinyali sıfır olmaz ve bilinmeyen sisteme tam olarak yakınsanamaz. Şekil 3.30.' da sistem/kanal tanıma bir örnek olarak radyo kanalı tanımlama verilmiştir.



Şekil 3.30. Radyo kanalı tanımlama

3.3.3.2. Ters Sistem Modelleme



Şekil 3.31. Ters sistem modelleme

Şekil 3.31.' de ters sistem modelleme yapısı verilmiştir. Burada

$$y(n) = x(n) * h(n) * w(n)$$

$$e(n) = d(n) - y(n)$$

$$e(n) = x(n) - x(n) * h(n) * w(n) = 0$$

$$x(n) = x(n) * h(n) * w(n)$$

eşitlikleri yazılabilir. z dönüşümü alınırsa

$$X(z) = X(z) H(z) W(z)$$

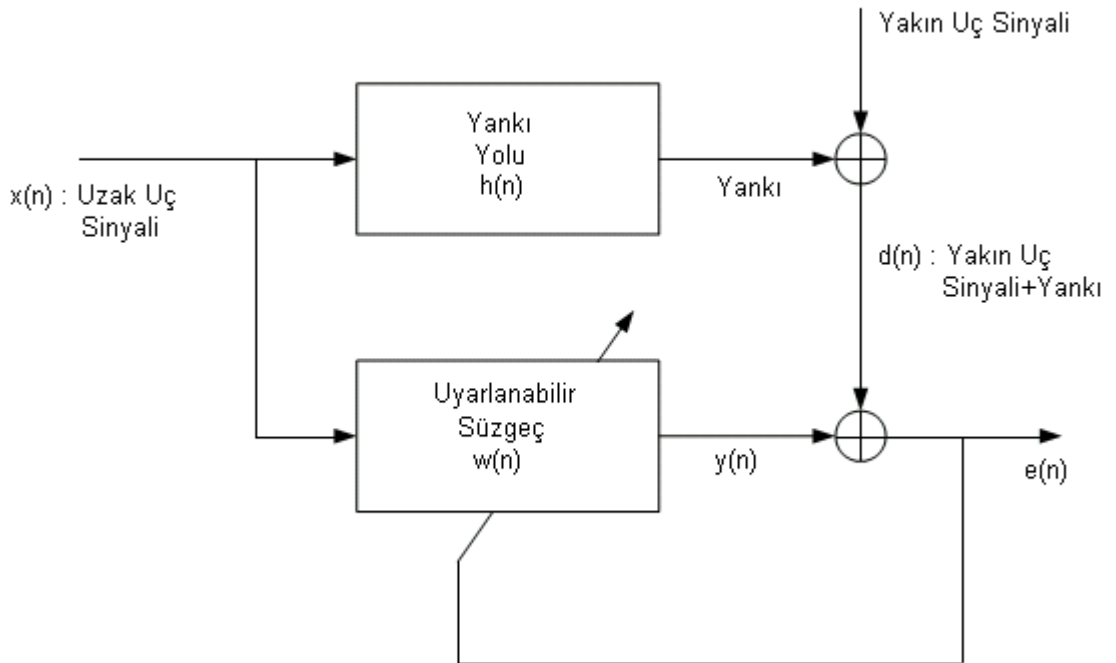
$$W(z) = \frac{1}{H(z)}$$

$$W(z) = \bar{H}(z)$$

$$w(n) = \bar{h} \quad (3.50)$$

elde edilir. Hata sinyali sıfır olduğunda (3.50) eşitliğinde görüldüğü gibi süzgeç katsayıları bilinmeyen sistem tepkesinin tersine eşittir. Bu şekilde bilinmeyen sistemin tersi bulunmuş olur. Ancak ters sistemin bulunabilmesi için sistemin tersinin alınabilir olması gerekir.

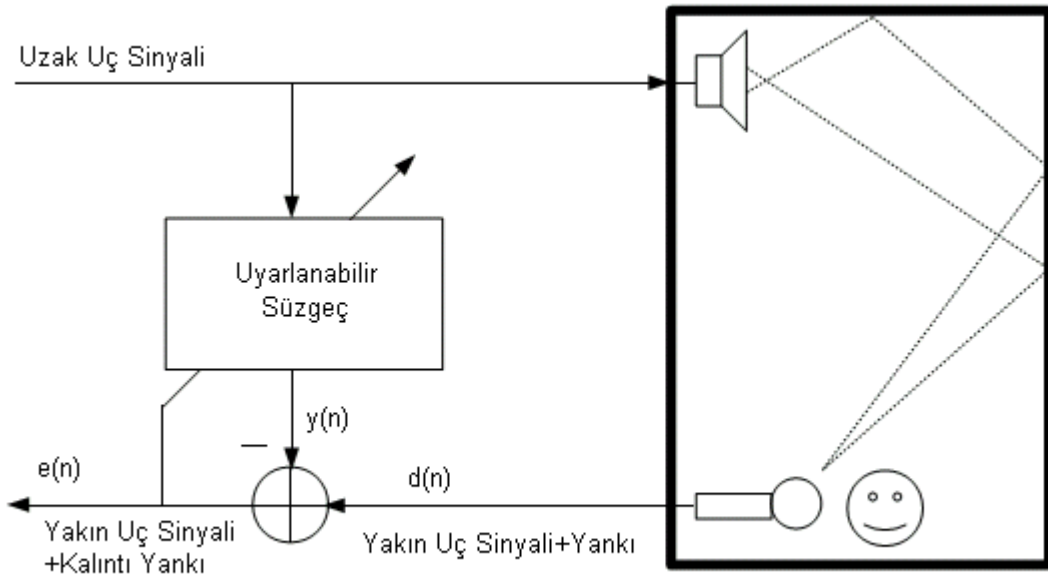
3.3.3.3. Yankı Giderme



Şekil 3.32. Yankı giderme

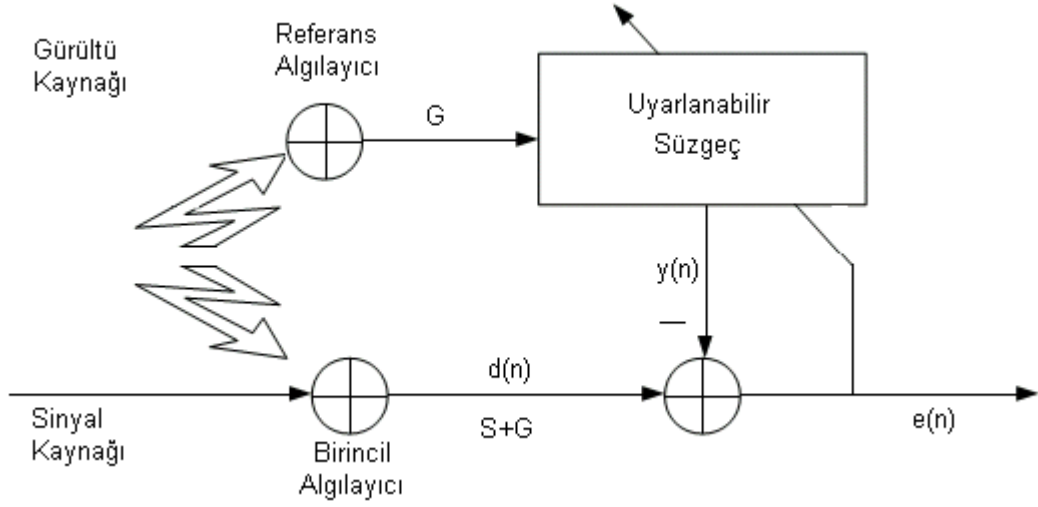
Şekil 3.32.' de görüldüğü gibi arzulanan sinyal, yankı sinyaline ek olarak yakın uç

sinyalinden oluşur. Bu sistem tanımlamadan farklı olan yanıdır. Normalde hata sinyalinin sıfır olabilmesi için süzgeç çıkışının, yakın uç sinyali+yankı olması gereklidir. Yakın uç sinyalinin istatistiksel olarak uzak uç sinyalinden bağımsız olduğu varsayılır. Bu da süzgecin, yankı yolunu yakın uç sinyali yokmuş gibi modellemeye çalışmasıyla sonuçlanır. Sonuçta süzgeç çıkışı yankı sinyaline yakınsar. Bu durumda hata sinyali de yakın uç sinyaline yakınsar. Burada sistem çıkışı hata sinyalidir. Aşağıda örnek olarak ses yankısı giderme örneği verilmiştir. Şekilde bir telekonferans sistemi tek taraflı olarak ele alınmıştır.



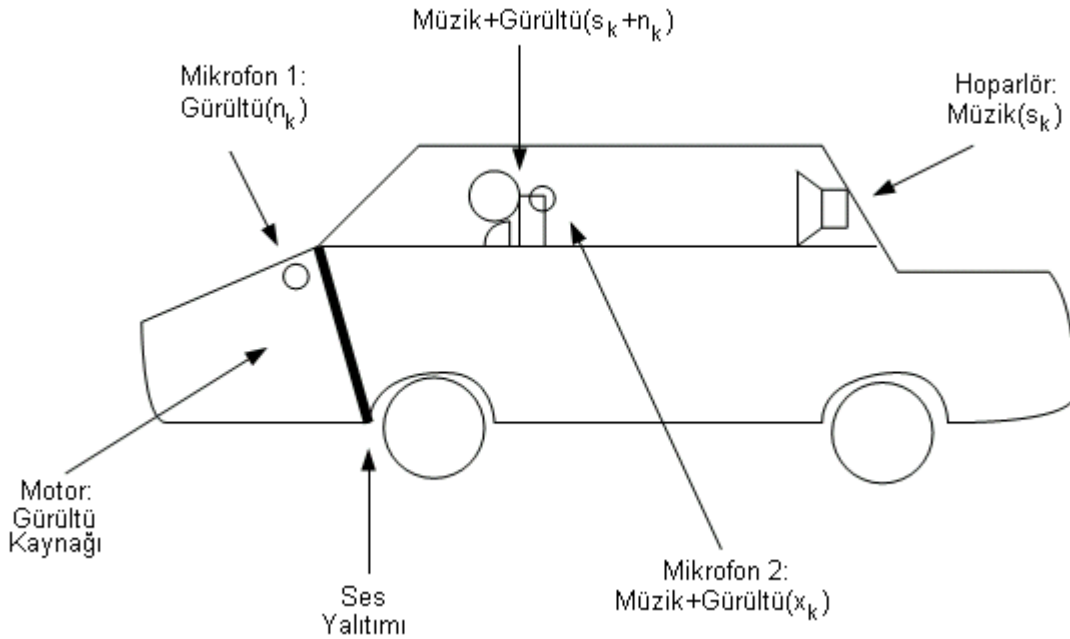
Şekil 3.33. Ses yankısı giderme

3.3.3.4. Gürültü/Girişim Giderme



Şekil 3.34. Gürültü / Girişim giderme

Şekil 3.34.' de gürültü/girişim giderme öbek çizgesi verilmiştir. Burada arzulanan sinyal , sinyal+gürültüden oluşur. Süzgeç girişine ise gürültü sinyali verilir. Referans algılayıcı mümkün olduğunca sinyal kaynağından yalıtılmalıdır. Yankı gidermeye benzer şekilde süzgeç çıkışı gürültü sinyaline yakınsar. Bu durumda hata sinyali de giriş sinyaline yakınsar. Şekil 3.35' de bir gürültü giderme örneği verilmiştir. Burada araba içinde motor gürültüsü giderilerek müzik kalitesi arttırılmaya çalışılır.



Şekil 3.35. Ses gürültüsü giderme[21]

3.3.3.5. Doğrusal Tahmin

Birçok uygulamada süzgeç giriş sinyalinden, istenen sinyalin elde edilmesi arzulanır. Örneğin k anında

$$d_k = x_{k+1} \quad (3.51)$$

olduğu düşünülün. Bu ileri tahmin problemi olarak bilinir. Ve girişin önceki değerleri kullanılarak bir zaman adımı ilerisi tahmin edilir. Tersine

$$d_k = x_{k-N} \quad (3.52)$$

olduğu zaman ise geri tahmin probleminden söz edilir. Bu durumda önceden incelenen bir veri örneği bulunmaya çalışılır.

e_k , k anındaki tahmin hata sinyali ise

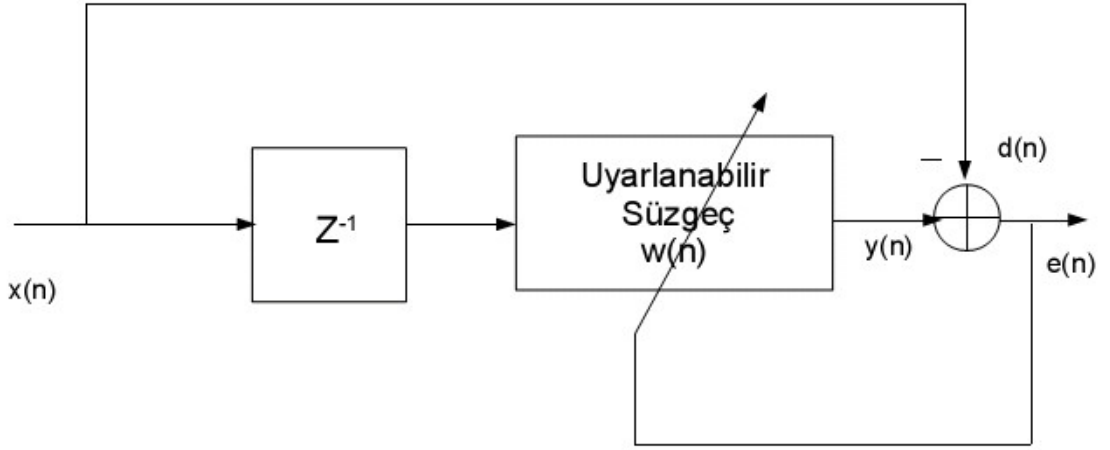
$$\begin{aligned} e_k &= d_k - y_k \\ e_k &= x_{k+1} - w_0 x_k - w_1 x_{k-1} - \dots - w_{N-1} x_{k-N+1} \\ x_{k+1} &= e_k + w_0 x_k + w_1 x_{k-1} + \dots + w_{N-1} x_{k-N+1} \end{aligned} \quad (3.53)$$

şeklinde ileri tahmin x_{k+1} bulunabilir. Geri tahmin ise

$$\begin{aligned} e_k &= x_{k-N} - w_0 x_k - w_1 x_{k-1} - \dots - w_{N-1} x_{k-N+1} \\ x_{k-N} &= e_k + w_0 x_k + w_1 x_{k-1} + \dots + w_{N-1} x_{k-N+1} \end{aligned} \quad (3.54)$$

şeklinde bulunur.

Uyarılma algoritması $e(n)$ tahmin hatasını sıfır yapmaya çalışır. Uygulamada e_k tahmin hatası sıfırlanamaz. Şekil 3.36' da ileri tahmin edicinin öbek çizgesi verilmiştir.



Şekil 3.36. İleri tahmin edici

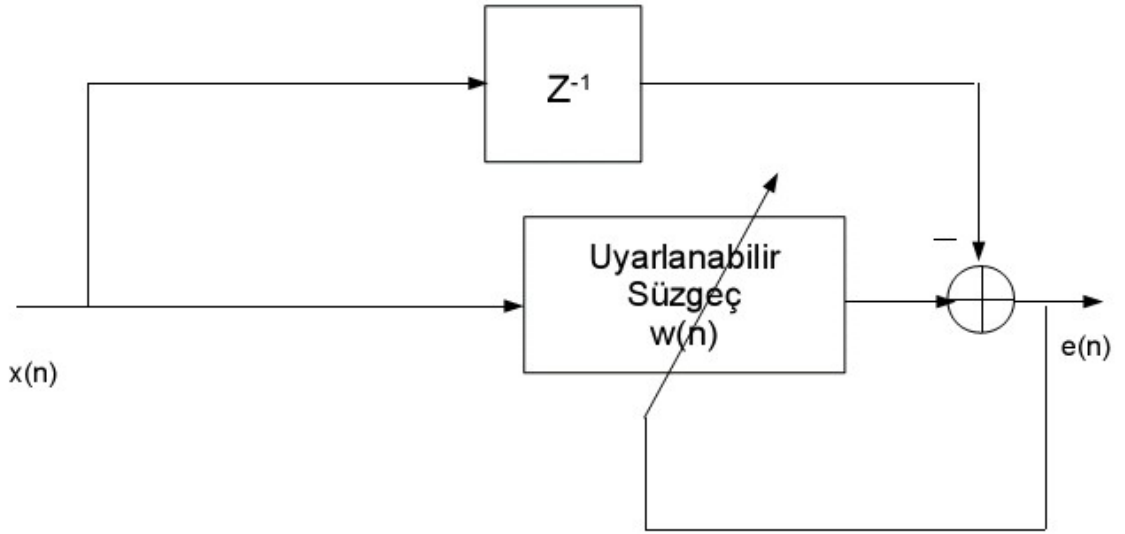
Burada

$$\begin{aligned}
 d(n) &= x(n) \\
 y(n) &= x(n-1) * w(n) \\
 e(n) &= x(n) - x(n-1) * w(n) = 0 \\
 x(n) &= x(n-1) * w(n) \\
 X(z) &= z^{-1} X(z) W(z) \\
 z X(z) &= X(z) W(z) \\
 x(n+1) &= x(n) * w(n)
 \end{aligned} \tag{3.55}$$

eşitlikleri yazılabilir. Uygulamada hata sıfır yapılamadığı için tahmine eklenir. Ve

$$x(n+1) = e(n) + x(n) * w(n) \tag{3.56}$$

elde edilir.



Şekil 3.37. Geri tahmin edici

Şekil 3.37.'de geri tahmin edicinin öbek çizgesi verilmiştir. Burada

$$\begin{aligned}
 d(n) &= x(n-1) \\
 y(n) &= x(n) * w(n) \\
 e(n) &= d(n) - y(n) \\
 e(n) &= x(n-1) - x(n) * w(n) = 0 \\
 x(n-1) &= x(n) * w(n)
 \end{aligned} \tag{3.57}$$

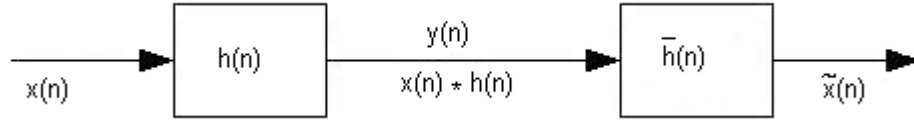
eşitlikleri yazılabilir. Uygulamada hata sıfır yapılamadığı için tahmine eklenir. Ve

$$x(n-1) = e(n) + x(n) * w(n) \tag{3.58}$$

elde edilir.

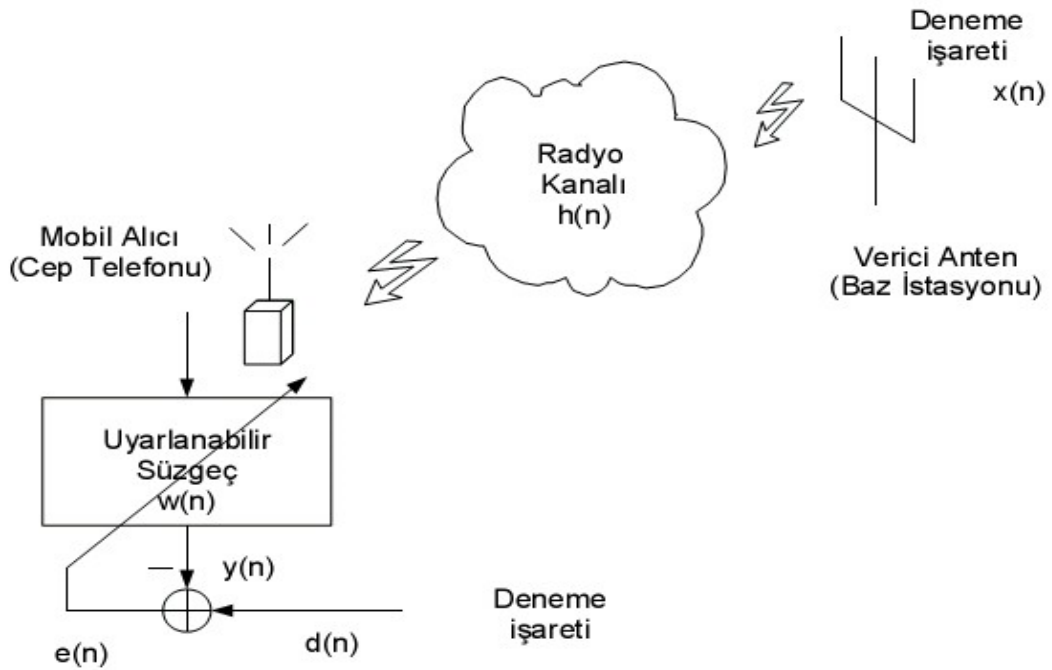
3.3.3.6. Kanal Dengeleme

Kanal dengelemenin amacı, kanalın sinyal üzerindeki olumsuz etkilerinin giderilmesidir. Bunun için ters sistem modellemeden yararlanır. Şekil 3.38.' de kanal dengeleme öbek çizgesi verilmiştir.



Şekil 3.38. Kanal dengeleme

$x(n)$ giriş sinyali, tepkisi $h(n)$ olan kanaldan geçirildiğinde sinyalde istenmeyen bozulmalar olur. Bu etkiyi yok etmek için kanal çıkışındaki sinyal, tepkisi kanal tepkesinin tersi olan bir başka sistemden geçirilirse tekrar giriş sinyali elde edilebilir. Böylece kanalın etkileri giderilmiş olur.



Şekil 3.39. Radyo kanalı dengeleme

Şekil 3.39.' da radyo kanalı dengeleme öbek çizgesi verilmiştir. Burada

$$\begin{aligned}
y(n) &= x(n) * h(n) * w(n) \\
d(n) &= x(n) \\
e(n) &= x(n) - x(n) * h(n) * w(n) = 0 \\
x(n) &= x(n) * h(n) * w(n) \\
X(z) &= X(z) H(z) W(z) \\
W(z) &= \frac{1}{H(z)} = \bar{H}(z) \\
w(n) &= \bar{h}(n)
\end{aligned} \tag{3.59}$$

elde edilir. $x(n)$ giriş sinyali kanaldan geçirildikten sonra katsayıları kanalın ters sistemine eşit olan süzgeçten geçirilerek kanalın etkileri yok edilmiş olur. Sistem çıkışı $y(n)$ sinyalidir.

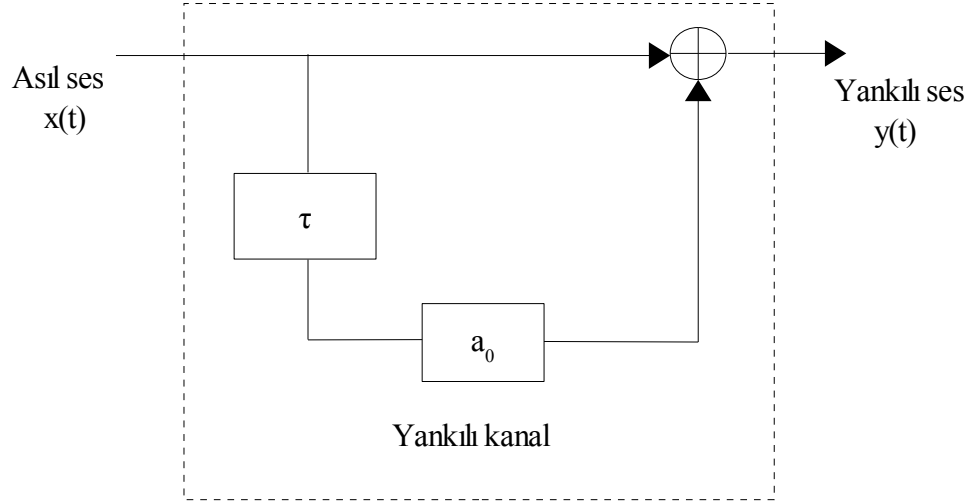
3.4. Uyarlanabilir Süzgeç Kullanarak Ses Yankısının Giderilmesi

3.4.1. Yankı ve Ses Yankısı

Bir sinyalin gecikmiş değerlerinin şu anki değerleri üzerine girişim yapması sonucu yankı oluşur. Benzer şekilde ses sinyali kimi zaman engellere çarpıp alıcıya gecikmeli olarak gelir. Bu gecikmeli ses dalgaları, ses kaynağından yeni çıkan ses dalgaları ile girişim yapar. Bu şekilde ses yankısı oluşur. Yankılar, gecikmelerine ve genliklerine bağlı olarak sesin anlaşılabilirliğini etkiler. Yankılar güçlü ise kulakta daha fazla hissedilir. Gecikme süreleri uzadıkça konuşmanın anlaşılabilirliği için konuşmacının ara ara sessiz kalması gerekebilir. Şekil 3.40 'da tek yankılı bir kanalın modeli verilmiştir. Burada

- $x(t)$: Asıl ses sinyali
- $y(t)$: Yankılı ses sinyali
- τ : Yankı gecikmesi

- a_0 : Yankı bağıl genliği
şeklindedir.



Şekil 3.40. Tek yankılı kanal modeli

Şekil 3.40.' da görüldüğü gibi yankılı ses sinyali

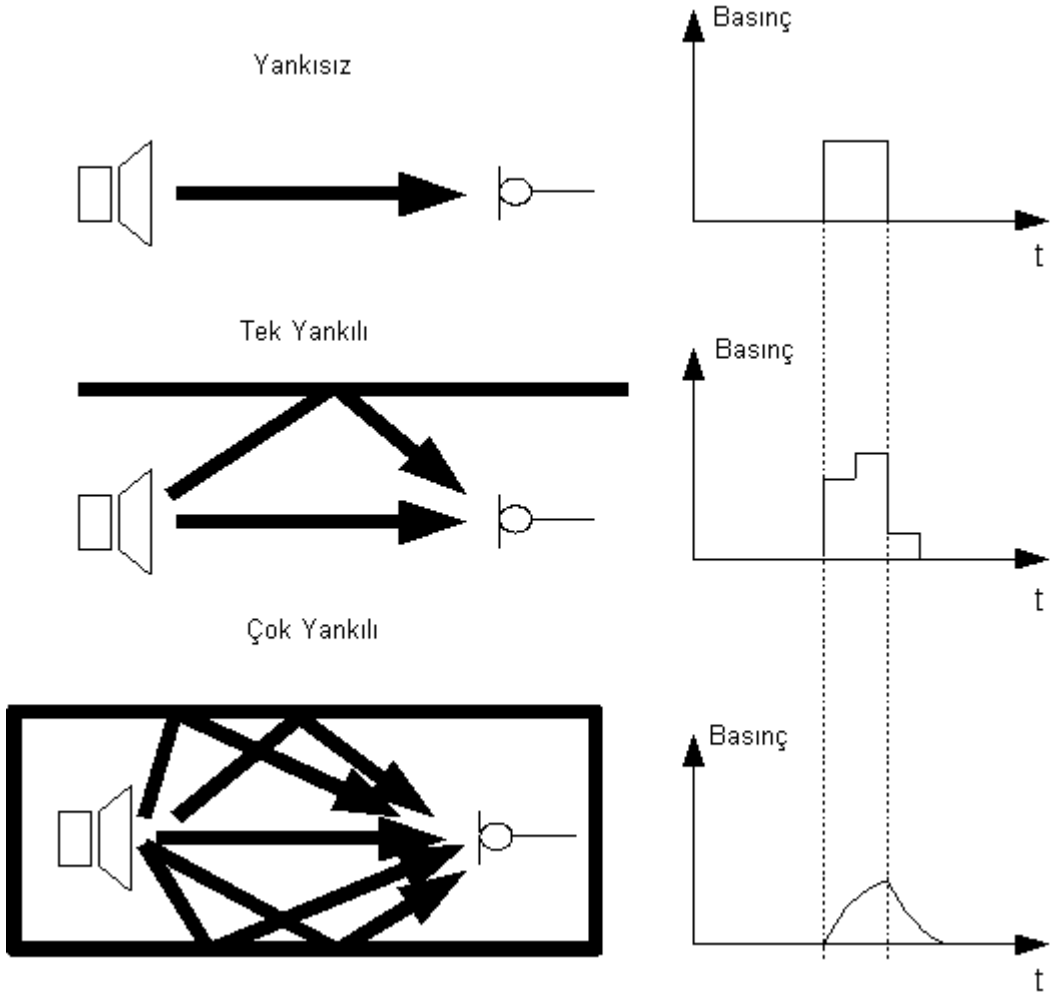
$$y(t) = x(t) + a_0 x(t - \tau) \quad (3.60)$$

şeklinde yazılabilir. Ayrık zamanlı ses sinyali için, yankı gecikmesinin karşılık geldiği örnek sayısı k ise benzer bağıntı

$$y(n) = x(n) + a_0 x(n - k) \quad (3.61)$$

şeklinde yazılabilir.

Genellikle yankı sayısı birden fazladır ve bu yankılar üst üste biner. Şekil 3.41.'de tek ve çok yankılı ses dalgalarının girişimi sonucu oluşan basınç değişimi verilmiştir.



Şekil 3.41. Yankı sayısının ses basıncına etkisi[22]

3.4.1.1. Ses Sinyaline Yankının Eklenmesi

Yankı, sinyalin k kadar gecikmişinin şu anki değere eklenmesiyle elde edilir. Ancak gecikmiş sinyal α kadar zayıflamıştır. Yankı kanalı FIR veya IIR yapıda modellenebilir. Şekil 3.40 'da FIR yapıdaki kanal modelinin öbek çizgesi verilmiştir. Kanal çıkışı

$$y(n) = x(n) + \alpha x(n - k) \quad (3.62)$$

şeklindedir. z dönüşümü yapılırsa

$$Y(z) = X(z) + \alpha X(z)z^{-k} \quad (3.63)$$

$$Y(z) = X(z)[1 + \alpha z^{-k}] \quad (3.64)$$

$$H(z) = \frac{Y(z)}{X(z)} = 1 + \alpha z^{-k} \quad (3.65)$$

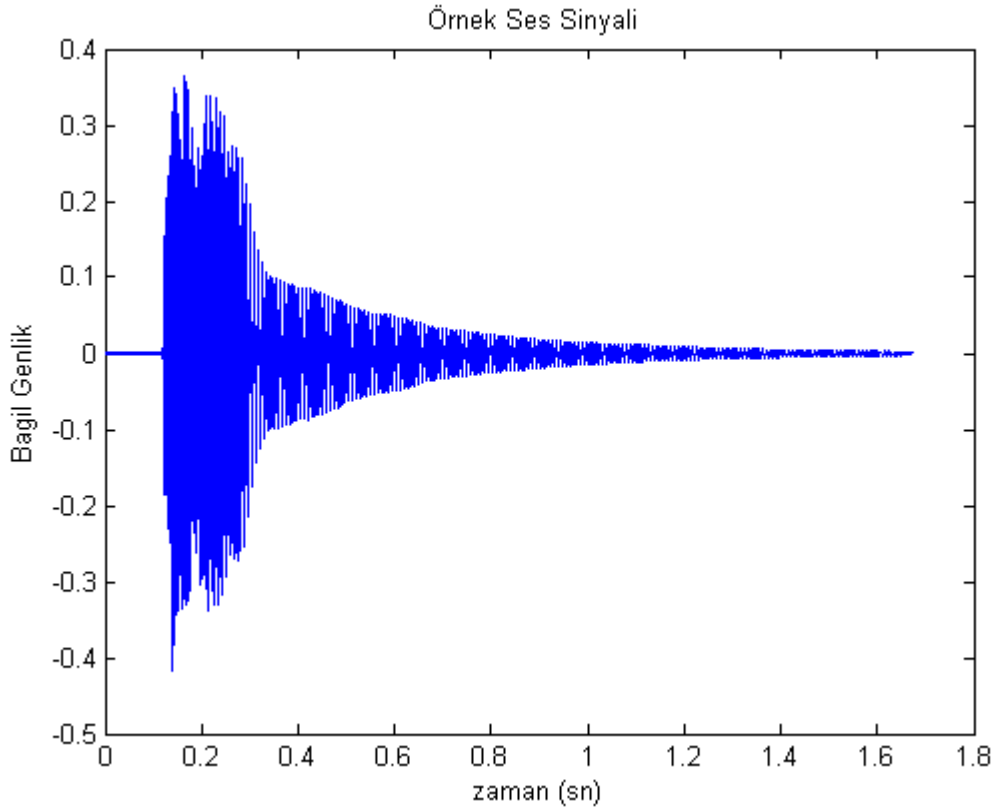
şeklinde kanalın transfer işlevi elde edilir. Ters z dönüşümü alınırsa

$$h(n) = \delta(n) + \alpha \delta(n-k) \quad (3.66)$$

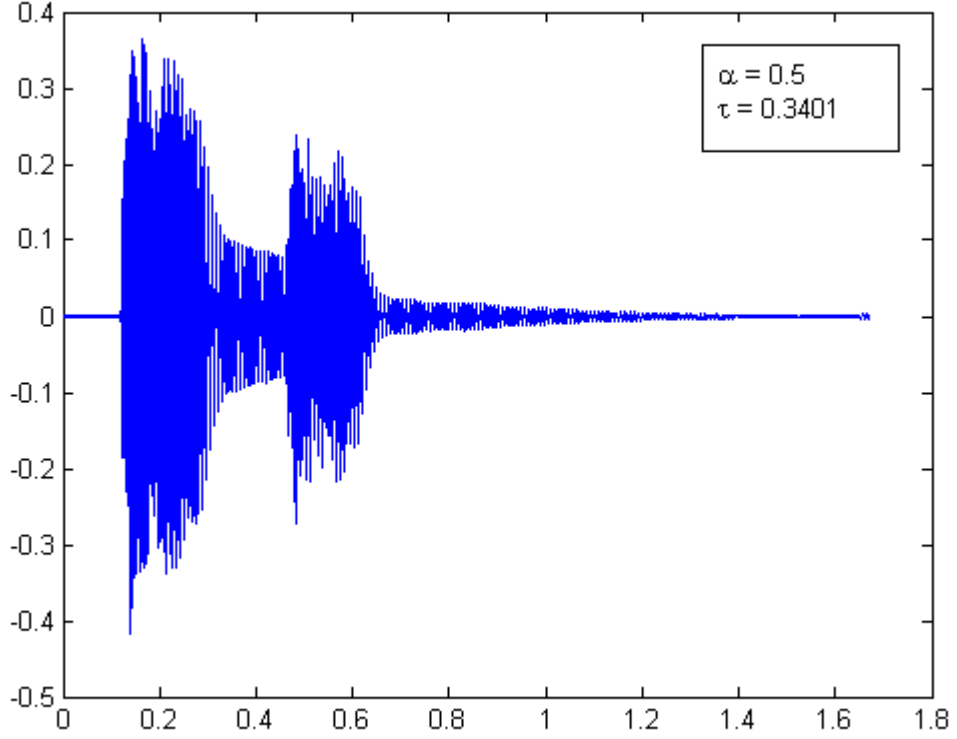
elde edilir. Veya vektör biçiminde yazılırsa

$$h(n) = [1 \underbrace{0 \ 0 \ 0 \ \dots \ 0 \ 0}_{(k-1) \text{ adet}} \ \alpha] \quad (3.67)$$

şeklinde gösterilebilir. Bu şekilde yankı ekleme doğrudan örnek geciktirme ile veya FIR süzgeçleme ile yapılabilir. Şekil 3.42.' de örnek ses sinyali verilmiştir.



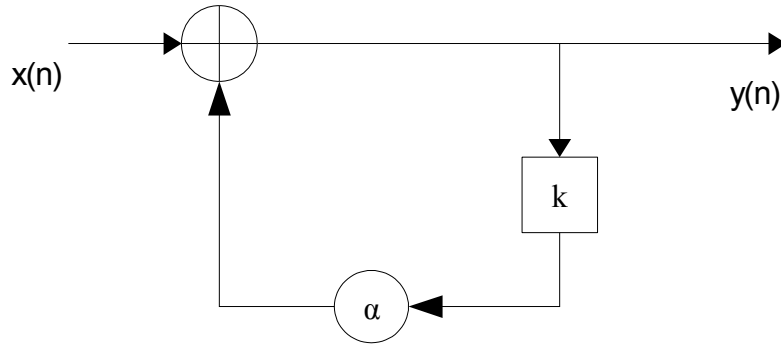
Şekil 3.42. Örnek Ses Sinyali(Metalik bir tın sesi)



Şekil 3.43. FIR modellenmiş kanal ile yankı eklenmesi

Şekil 3.43.' de tek yankı eklenmiş sinyal verilmiştir.

Aynı yankı farklı zayıflamalar ile birden fazla defa asıl sinyale eklenebilir. Böyle bir sistem IIR yapıda modellenebilir[23]. Şekil 3.44.' de bu yapı verilmiştir. Şekil 3.45.' de ise çok yankılı sinyal verilmiştir.



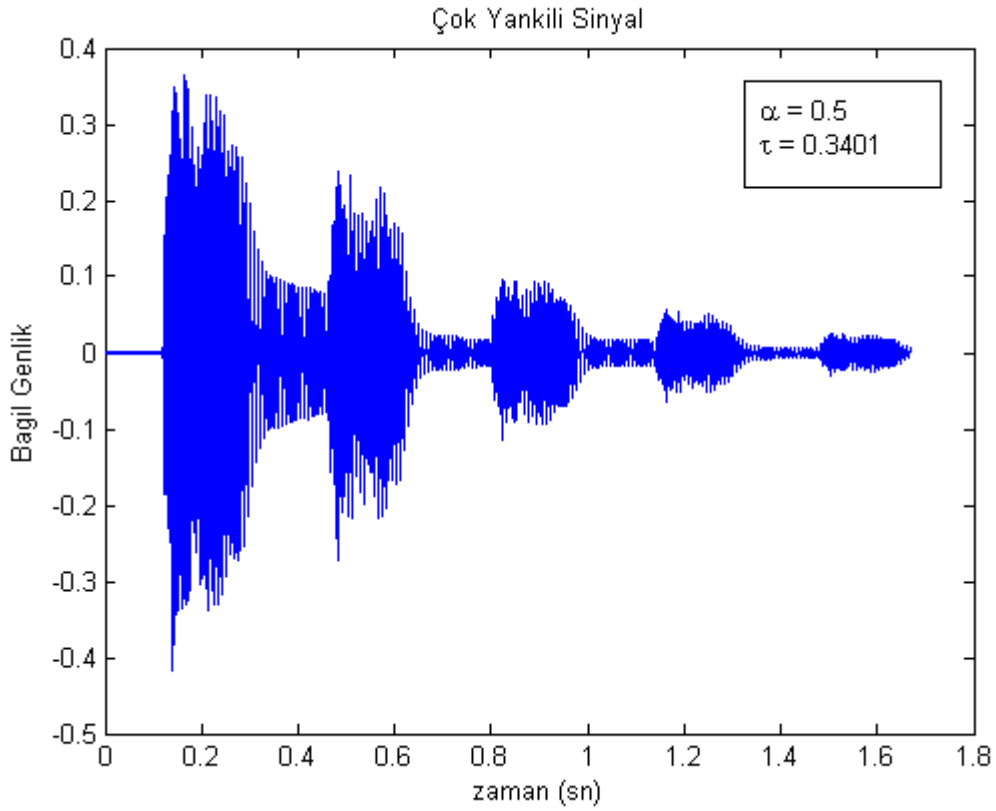
Şekil 3.44. Birden fazla yankılı sistemin IIR yapıda modellenmesi

Sistemin çıkışı

$$y(n) = x(n) + \alpha y(n-k) \quad (3.68)$$

$$H(z) = \frac{1}{1 - \alpha z^{-k}} \quad (3.69)$$

şeklindedir. Bu yapıda yankı ekleme IIR süzgeçleme ile yapılabilir.



Şekil 3.45. IIR modellenmiş kanal ile çoklu yankı eklenmesi

En genel şekilde çok yankılı bir sistemin çıkışı

$$H(z) = 1 + [a_1 z^{-k_1} + a_1^2 z^{-2k_1} + \dots] + [a_2 z^{-k_2} + a_2^2 z^{-2k_2} + \dots] + \dots + [a_n z^{-k_n} + a_n^2 z^{-2k_n} + \dots] + \dots \quad (3.70)$$

şeklinde yazılabilir.

3.4.2. İlinti Kullanarak Yankı Parametrelerinin Bulunması

İlinti, iki sürecin birbiri ile olan ilişkisini tanımlar. Sinyal işleme için iki sinyalin benzerliğinin bir ölçütüdür. Yüksek ilinti gösteren sinyallerde yüksek benzerlik, düşük ilinti gösteren sinyallerde ise düşük benzerlik vardır. Özilinti ise bir sinyalin farklı zaman gecikmeleri altında kendine olan benzerliğinin bir ölçütüdür[13].

İki sinyal arasındaki çapraz-ilinti

$$r_{xy}(t, t-\tau) = E[x(t)y^*(t-\tau)] \quad (3.71)$$

şeklinde tanımlanmaktadır. Burada $E[\]$ işleci beklenen değeri, $*$ ise karmaşık eşleniği ifade etmektedir. Ayırık zamanlı olarak ifade edilirse çapraz-ilinti

$$r_{xy}(\tau) = \sum_{n=-\infty}^{\infty} x(n)y(n-\tau) \quad ; \tau=0, \bar{+}1, \bar{+}2, \dots \quad (3.72)$$

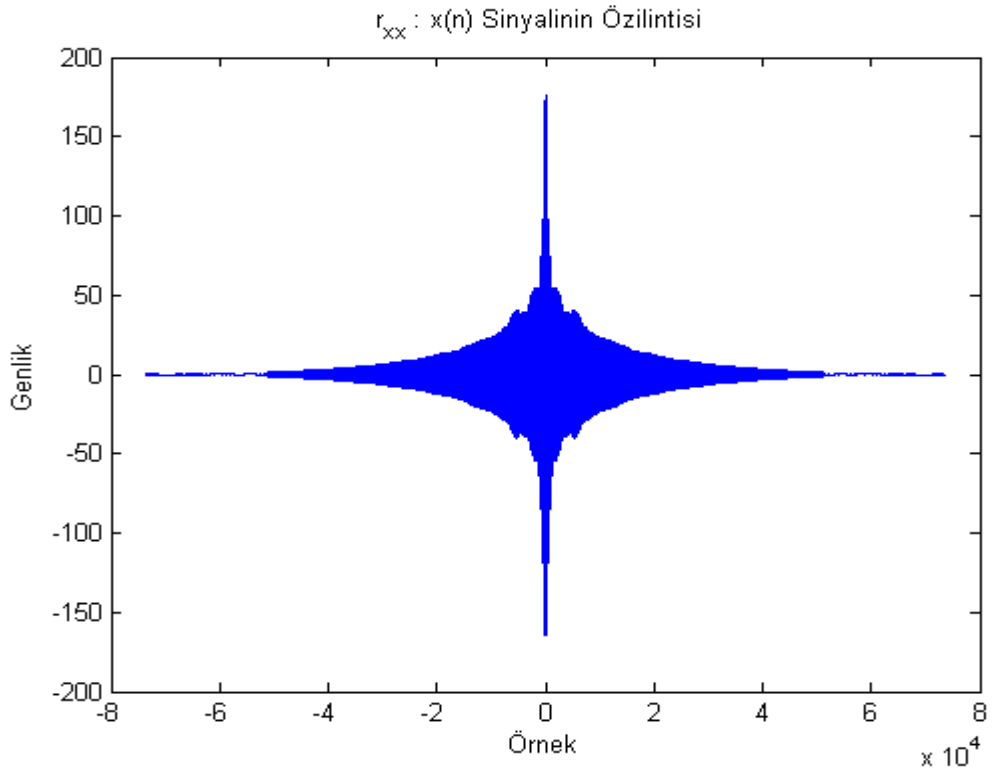
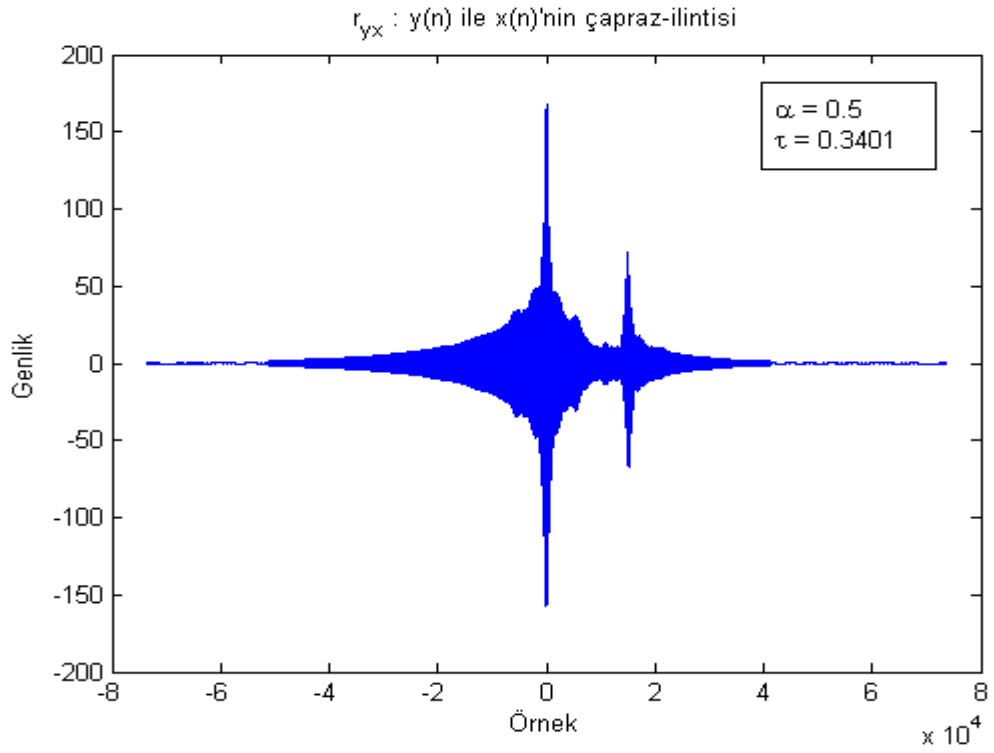
$$= \sum_{n=-\infty}^{\infty} x(n+\tau)y(n) \quad (3.73)$$

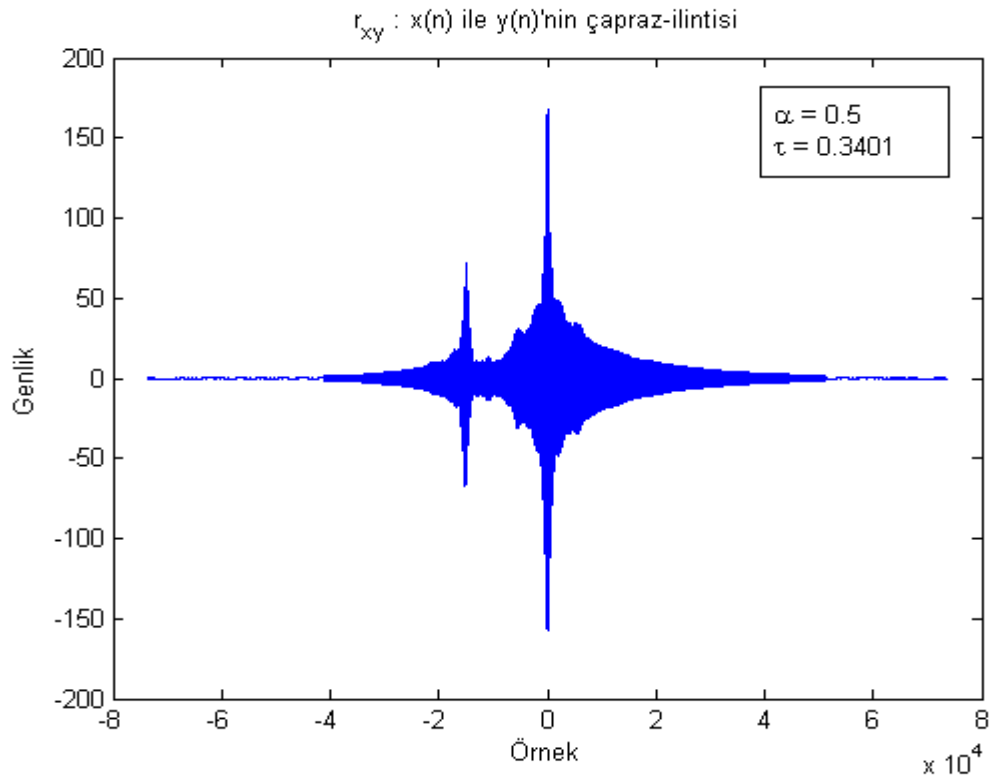
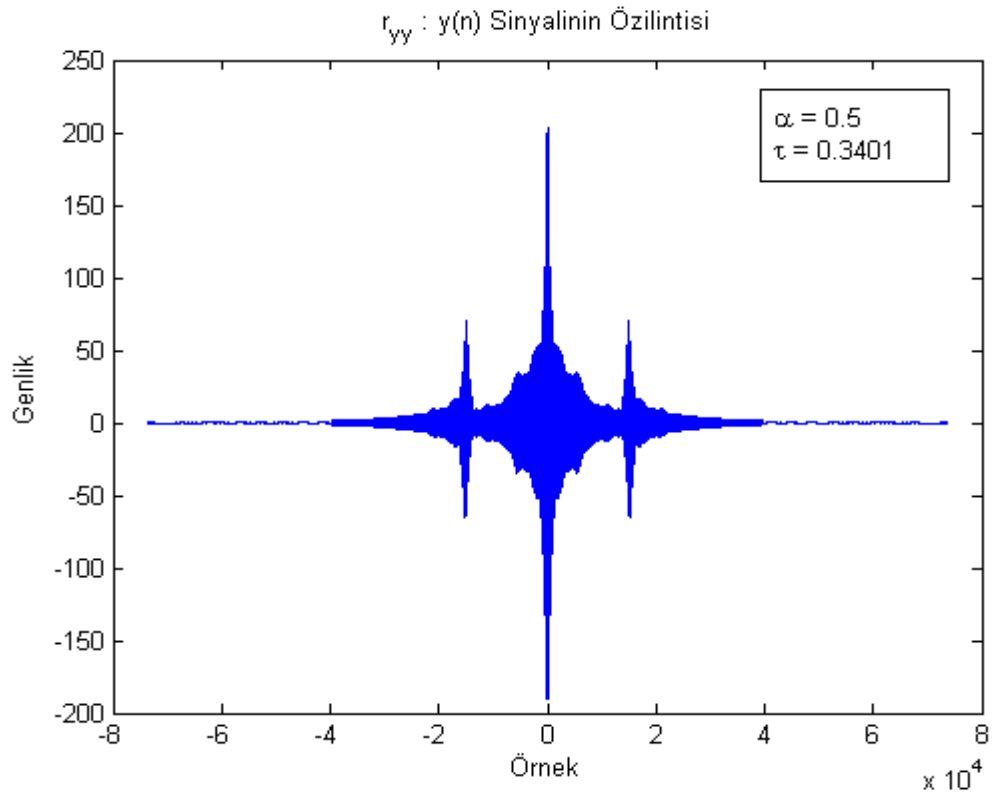
şeklinde yazılabilir. Özilinti fonksiyonu ise

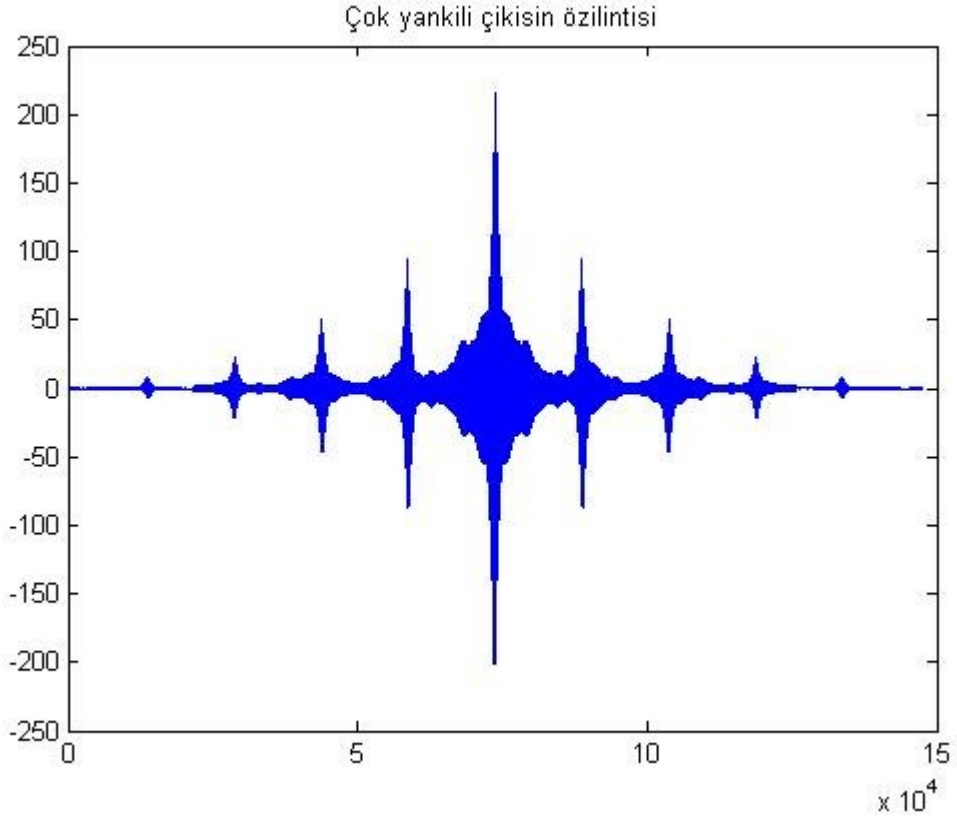
$$r_{xx}(\tau) = \sum_{n=-\infty}^{\infty} x(n)x(n-\tau) \quad (3.74)$$

$$= \sum_{n=-\infty}^{\infty} x(n+\tau)x(n) \quad (3.75)$$

şeklinde yazılabilir. Şekil 3.42.' de verilen örnek ses sinyali $x(n)$ giriş sinyalini ve Şekil 3.43. 'de verilen sinyal ise yankılı ses sinyali $y(n)$ 'yi temsil ederse bu iki sinyal arasındaki özilinti ve çapraz-ilinti grafikleri Şekil 3.46 ,Şekil 3.47 , Şekil 3.48 ve Şekil 3.49 'da verilmiştir. Şekil 3.50.' de ise çok yankılı sinyalin özilintisi verilmiştir.

Şekil 3.46. r_{xx} :Giriş sinyalinin özilintisiŞekil 3.47. r_{yx} : $y(n)$ ile $x(n)$ 'nin çapraz-ilintisi

Şekil 3.48. r_{xy} : $x(n)$ ile $y(n)$ 'nin özillentiŞekil 3.49. r_{yy} : Çıkış sinyali özillenti



Şekil 3.50. Çok yankılı sinyalin özilintisi

İlinti fonksiyonları, yankının gecikme miktarı (k) ve bağıl zayıflaması (α) hakkında bilgi vermektedir. Çıkışın özilinti grafiği incelenirse (Şekil 3.49) üç büyük tepe olduğu görülmektedir. Bu tepelerden en büyüğü $\tau=0$ anında, diğer iki tepe ise $\tau=k$ anında oluşmaktadır. En büyük tepe ile küçük tepeler arasındaki örnek sayısı gecikme miktarını vermektedir. Şekilden gecikme miktarının $k=15000$ örnek olduğu görülmektedir.

Tek yankılı bir kanalın tepkesi

$$h(n) = \delta(n) + \alpha \delta(n-k) \quad (3.76)$$

şeklindedir. $n=0$ ve $n=k$ için

$$h(0) = \delta(0) + \alpha \delta(-k) \quad (3.77)$$

$$h(0) = 1 \quad (3.78)$$

$$h(k) = \delta(k) + \alpha \delta(0) \quad (3.79)$$

$$h(k) = \alpha \quad (3.80)$$

elde edilir. $x(n)$ ve $y(n)$ sinyallerinin çapraz-ilişkisi ile katlaması arasında

$$r_{xy}(\tau) = x(n) * y(-n) \quad (3.81)$$

ilişkisi vardır. Buradan

$$r_{yy}(\tau) = y(n) * y(-n) \quad (3.82)$$

elde edilir. Çıkış giriş ile sistem tepkesinin katlamasıdır ve yerine konulursa

$$\begin{aligned} r_{yy}(\tau) &= [h(n) * x(n)] * [h(-n) * x(-n)] \\ r_{yy}(\tau) &= r_{hh}(\tau) * r_{xx}(\tau) \end{aligned} \quad (3.83)$$

elde edilir. Buradan

$$r_{yy}(\tau) = \left[\sum_{n=-\infty}^{\infty} h(n)h(n-\tau) \right] * \left[\sum_{n=-\infty}^{\infty} x(n)x(n-\tau) \right] \quad (3.84)$$

elde edilir. $\tau=0$ için

$$r_{yy}(0) = \left[\sum_{n=-\infty}^{\infty} h^2(n) \right] * r_{xx}(0) \quad (3.85)$$

$$r_{yy}(0) = \left[\sum_{n=-\infty}^{\infty} [\delta(n) + \alpha\delta(n-k)]^2 \right] * r_{xx}(0) \quad (3.86)$$

$$r_{yy}(0) = (1 + \alpha^2) * r_{xx}(0) \quad (3.87)$$

elde edilir. $\tau=k$ için ise

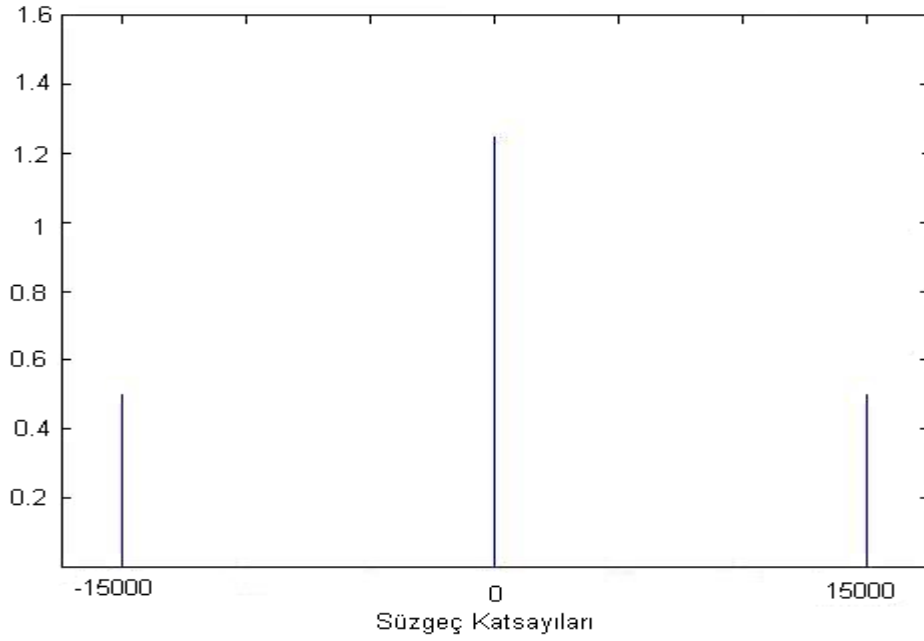
$$r_{yy}(k) = \left[\sum_{n=-\infty}^{\infty} h(n)h(n-k) \right] * r_{xx}(k) \quad (3.88)$$

$$r_{yy}(k) = \alpha * r_{xx}(k) \quad (3.89)$$

elde edilir. (3.87) ve (3.89) eşitlikleri kullanılarak α ve k parametreleri bulunabilir. $r_{hh}(\tau)$ özilinti fonksiyonu

$$r_{hh}(\tau) = \begin{cases} \alpha & , \tau = -k \\ 1 + \alpha^2 & , \tau = 0 \\ \alpha & , \tau = k \end{cases} \quad (3.90)$$

şeklindedir. Şekil 3.51.' de tek yankılı kanalın tepkesinin özilintisi verilmiştir.



Şekil 3.51. $r_{hh}(\tau)$: Kanal tepkesinin özilintisi

İlinti kullanarak yankı parametrelerini bulmanın farklı bir yolu daha vardır. Tek yankılı bir kanalın çıkışı

$$y(n) = x(n) + \alpha x(n - k) \quad (3.91)$$

şeklindedir. (3.91) eşitliğinin her iki tarafı $x(n - \tau)$ ile çarpılıp $n \in (-\infty, \infty)$ aralığında toplanırsa

$$\sum_{-\infty}^{\infty} y(n) x(n - \tau) = \sum_{-\infty}^{\infty} x(n - \tau) [x(n) + \alpha x(n - k)]$$

$$r_{yx}(\tau) = r_{xx}(\tau) + \alpha r_{xx}(\tau - k) \quad (3.92)$$

elde edilir. $\tau=0$ ve $\tau=k$ için

$$r_{yx}(0) = r_{xx}(0) + \alpha r_{xx}(k) \quad (3.93)$$

$$r_{yx}(k) = r_{xx}(k) + \alpha r_{xx}(0) \quad (3.94)$$

bulunur. Eşitlik (3.93) ve (3.94) kullanılarak k ve α bulunabilir[24].

3.4.3. Yankı Giderme Yöntemleri

Bu çalışmada ses yankısının giderilmesi sorunu iki durum için incelenecektir.

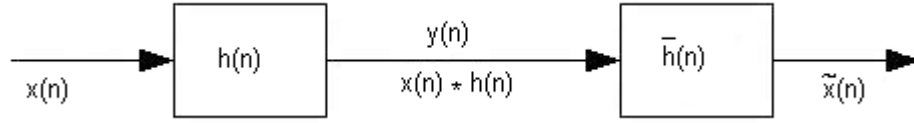
- Kayıtlı ses sinyalinden yankının giderilmesi
- Telekonferans uygulamasında yankının giderilmesi.

Birinci durumda, asıl ses sinyalinin elimizde olmadığı varsayılır ve yankı giderilmeye çalışılır. Bunun için yankı sinyali hakkındaki önbilgilerden yararlanır. Yankı yolu ters sistem olarak modellenir ve kanal dengeleme (equalization) ile yankının etkisi yok edilmeye çalışılır. Kanal dengeleme işlemini yapan yazılım veya donanıma dengeleyici (equalizer) denilir. Asıl ses sinyalinin elimizde olmadığı durumda kör dengeleme (blind equalization) yöntemleri kullanılır.

İkinci durumda, telekonferans uygulamasında yankı giderilmeye çalışılır. Burada yankının giderilmesi için uyarlanabilir süzgeçler kullanılır ve gerçek zamanlı olarak yankı giderilmeye çalışılır.

3.4.3.1. Kanal Dengeleme İle Yankının Giderilmesi

İletişim sistemlerinde vericiden aktarılan bilginin değişmeden alıcıya ulaşması istenir. Ancak bilgi kanaldan iletilirken çeşitli bozulmalara uğrar. Kanalın bu olumsuz etkilerini giderme işlemine kanal dengeleme denilir. Kanal dengelemede kanalın ters sistem modeli elde edilir ve alıcıdaki sinyal bu dengeleyiciden geçirilir. Bu şekilde asıl bilgi sinyali elde edilir. Ancak uygulamada tam bir kanal dengeleme yapılamaz.



Şekil 3.52. Kanal dengeleme öbek çizgesi

Şekil 3.52.' de kanal dengeleme öbek çizgesi verilmiştir. Burada

- $x(n)$: Giriş sinyali
- $h(n)$: Kanal tepkesi
- $\tilde{x}(n)$: Dengeleyici çıkışı
- $y(n)$: Kanal çıkışıdır.

Dengeleyici çıkışı, uygulamada giriş sinyalinden farklı olduğu için üstüne bir sinyal konularak farkı gösterilir. İstenen dengeleyici çıkışının mümkün olduğunca giriş sinyaline yakınsamasıdır.

Kanal çıkışı $y(n)$, giriş sinyali $x(n)$ ile kanal tepkesi $h(n)$ 'nin katlamasıdır. Kanal çıkışı

$$y(n) = x(n) * h(n) \quad (3.95)$$

yazılabilir. $y(n)$ sinyali kanal tepkesinin tersine eşit olan bir sistemden geçirilirse

$$\tilde{x}(n) = y(n) * \bar{h}(n) \quad (3.96)$$

$$\begin{aligned} &= x(n) * h(n) * \bar{h}(n) \\ &= x(n) * \delta(n) \end{aligned}$$

$$\tilde{x}(n) = x(n) \quad (3.97)$$

elde edilir.

Ancak her sistemin tersi alınamaz. Kanal dengeleme için kanalın tersinin alınabilmesi gerekir. z düzleminde kanalın geçiş işlevi ile ters sistemin geçiş işlevleri çarpımı bire eşittir. Yani

$$H(z)\bar{H}(z)=1 \quad (3.98)$$

$$\bar{H}(z)=\frac{1}{H(z)} \quad (3.99)$$

eşitlikleri yazılabilir. Ancak bu eşitliklerin yazılabilmesi yani sistemin tersinin alınabilmesi için sistemin geçiş işlevi ile ters sistemin geçiş işlevinin en azından kısmen örtüşen yakınsaklık bölgeleri olmalıdır[14].

3.4.3.2. Ters Süzgeçler

Bir sistemin transfer işlevinin bütün kutupları z-düzleminde birim çember içinde ise bu sistem kararlıdır. Sistemin geçiş işlevi

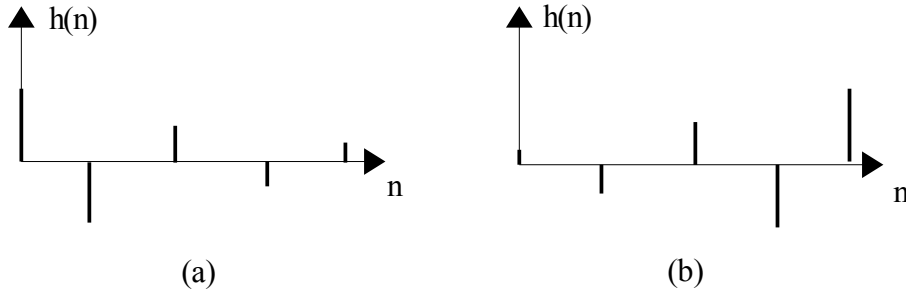
$$H(z)=\frac{N(z)}{D(z)} \quad (3.100)$$

şeklinde gösterilebilir. Burada N(z) 'nin kökleri sistemin sıfırları, D(z) 'nin kökleri sistemin kutuplarıdır. Ters sistemin geçiş işlevi H(z) 'nin tersidir. Ters sistemin geçiş işlevi

$$H_i(z)=\frac{D(z)}{N(z)} \quad (3.101)$$

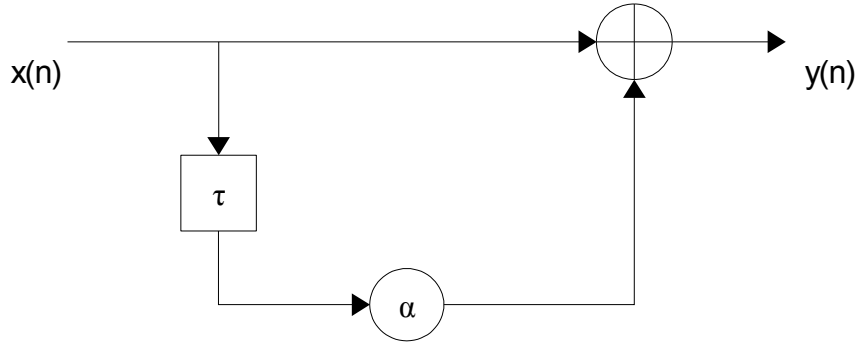
şeklinde dir. Ters sisteminde nedensel ve kararlı olması gerekir. Bunun için N(z) 'nin kökleri birim çember içinde olmalıdır. Yani tersi alınacak sistemin sıfırları birim çember içinde olmalıdır. Bir sistemin transfer işlevinin hem sıfırları hemde kutupları birim çember içinde ise bu sisteme minimum fazlı sistem denilir. Sıfırların bir kısmı birim çemberin dışında ise maksimum fazlı sistem denilir[25]. Verilen bir genlik tepkisi için minimum fazlı sistem her w frekansında en küçük faz genliğine sahip nedensel sistemdir. Başka bir deyişle belirli bir genlik karakteristiğini minimum fazda gerçekleyen sistemlerdir. Tam ters süzgeç uygulaması için tersi alınacak sistemin minimum fazlı olması gerekir[23]. Minimum fazlı sistemlerde, sistem tepkisi gittikçe azalırken, maksimum fazlı sistemlerde artmaktadır. Şekil 3.53.' de bu durum

görülmektedir.



Şekil 3.53. a) Minimum fazlı sistem b) Maksimum fazlı sistem[25]

Tek yankılı bir sistem , FIR süzgeç yapısında modellenmelidir. Şekil 3.54.' de bu yapının öbek çizgesi verilmiştir.



Şekil 3.54. Tek yankılı bir sistemin FIR yapıda modellenmesi

Sistemin çıkışı

$$y(n) = x(n) + \alpha x(n-1) \quad (3.102)$$

$$H(z) = 1 + \alpha z^{-1} \quad (3.103)$$

şeklindedir. Sistem yalnızca $z = -\alpha$ noktasında bir sifıra sahiptir. $|\alpha| < 1$ olduğu için sifırı birim çember içindedir. Bu FIR sistemin sifırı ters süzgecin kutbu olacaktır. Ters süzgecin kutbu da birim çember içinde olacağından doğrudan ters süzgeçleme yapılabilir[23].

Aynı yankı farklı zayıflamalar ile birden fazla defa asıl sinyale eklenebilir. Böyle bir sistem IIR yapıda modellenmelidir. Şekil 3.44. 'de bu sistemin öbek çizgesi verilmiştir. Sistemin çıkışı

$$y(n) = x(n) + \alpha y(n-k) \quad (3.104)$$

$$H(z) = \frac{1}{1 - \alpha z^{-k}} \quad (3.105)$$

şeklindedir. Sistem sadece bir adet kutba sahiptir. $|\alpha| < 1$ olduğu için kutup birim çember içindedir ve sistem kararlıdır. Ters sisteminse sıfırı birim çember içindedir ve doğrudan ters süzgeçleme yapılabilir.

Çok yankılı bir sistem

$$H(z) = 1 + [a_1 z^{-k_1} + a_1^2 z^{-2k_1} + \dots] + [a_2 z^{-k_2} + a_2^2 z^{-2k_2} + \dots] + \dots + [a_n z^{-k_n} + a_n^2 z^{-2k_n} + \dots] + \dots \quad (3.106)$$

şeklinde FIR yapıda modellenebilir. Bu sistemin tersi

$$\bar{H}(z) = \frac{1}{1 + [a_1 z^{-k_1} + a_1^2 z^{-2k_1} + \dots] + [a_2 z^{-k_2} + a_2^2 z^{-2k_2} + \dots] + \dots + [a_n z^{-k_n} + a_n^2 z^{-2k_n} + \dots] + \dots} \quad (3.107)$$

şeklindedir. Pratikte bütün a katsayıları 1' den küçüktür. Bu nedenle sistem kararlıdır. Görüldüğü gibi tek ve çok yankılı kanalların tersi doğrudan alınabilir ve kanal dengeleme yapılabilir.

3.4.3.3. FIR Modellenmiş Bir Kanalin Dengelenmesi

FIR modellenmiş bir kanalın öbek çizgesi Şekil 3.54. 'de verilmiştir. Bu kanalın geçiş işlevi

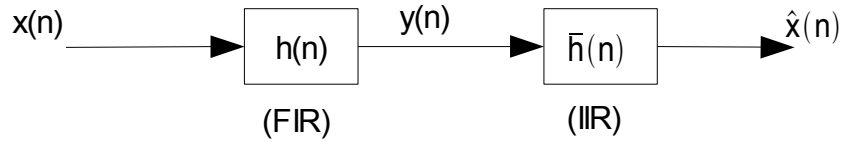
$$H(z) = 1 + \alpha z^{-k} \quad (3.108)$$

şeklindedir. Dengeleyicinin geçiş işlevi

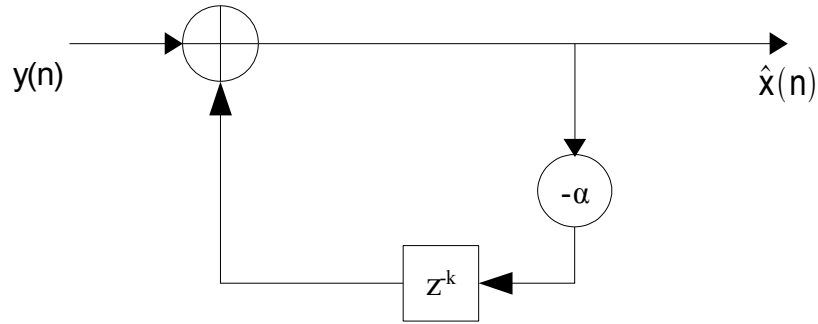
$$\bar{H}(z) = \frac{1}{1 + \alpha z^{-k}} \quad (3.109)$$

şeklindedir. Görüldüğü gibi dengeleyici IIR yapıdadır. Şekil 3.55.' de FIR yapıdaki

kanalın IIR dengeleyici ile dengelenmesi gösterilmiştir.



Şekil 3.55. FIR modellenmiş kanalın IIR süzgeç ile dengelenmesi



Şekil 3.56. IIR dengeleyici

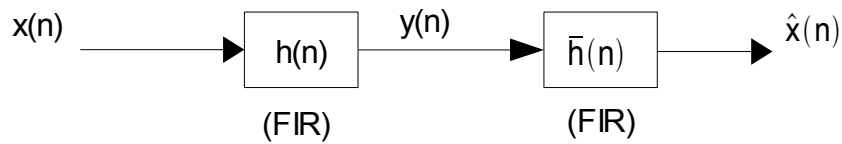
Şekil 3.56.' da IIR dengeleyici öbek çizgesi verilmiştir. Dengeleyicinin çıkışı

$$\tilde{X}(z) = Y(z) - a_0 z^{-k} \tilde{X}(z) \quad (3.110)$$

$$\hat{x}(n) = y(n) - \alpha \hat{x}(n-k) \quad (3.111)$$

şeklinde bulunur.

FIR modellenmiş bir kanal IIR dengeleyici yerine FIR dengeleyici kullanılarak dengelenebilir. Bunun için IIR dengeleyicinin geçiş işlevinin payı paydasına bölünür ve belli bir katsayıdan sonrası göz ardı edilir. Şekil 3.57.' de FIR yapıdaki kanalın FIR dengeleyici ile dengelenmesi gösterilmiştir.



Şekil3.57. FIR modellenmiş kanalın FIR süzgeç ile dengelenmesi

$$\begin{array}{r}
\begin{array}{|l}
1 \\
\hline
1+\alpha z^{-k} \\
-\alpha z^{-k} \\
\hline
-\alpha z^{-k} -\alpha^2 z^{-2k} \\
\hline
\alpha^2 z^{-2k} \\
\hline
-\alpha^2 z^{-2k} +\alpha^3 z^{-3k} \\
\hline
\alpha^3 z^{-3k} \\
\hline
\vdots
\end{array}
\end{array}
\left|
\begin{array}{l}
1+\alpha z^{-k} \\
\hline
1-\alpha z^{-k} +\alpha^2 z^{-2k} -\alpha^3 z^{-3k} +\alpha^4 z^{-4k} -\dots
\end{array}
\right.$$

$$\bar{H}(z) = 1 - \alpha z^{-k} + \alpha^2 z^{-2k} - \alpha^3 z^{-3k} + \alpha^4 z^{-4k} - \dots \quad (3.112)$$

Bölme işlemi sonunda elde edilen bölüm sonsuz uzunluktadır. $a_0 < 1$ iken, kuvvet derecesi (m) arttığında α^m 'nin değeri azalır. Böylece yüksek dereceden elemanlar gözardı edilebilir. Örneğin, ilk dört eleman alınarak FIR sistem gerçekleştirir

$$\bar{H}_{\text{FIR}}(z) = 1 - \alpha z^{-k} + \alpha^2 z^{-2k} - \alpha^3 z^{-3k} + \alpha^4 z^{-4k} \quad (3.113)$$

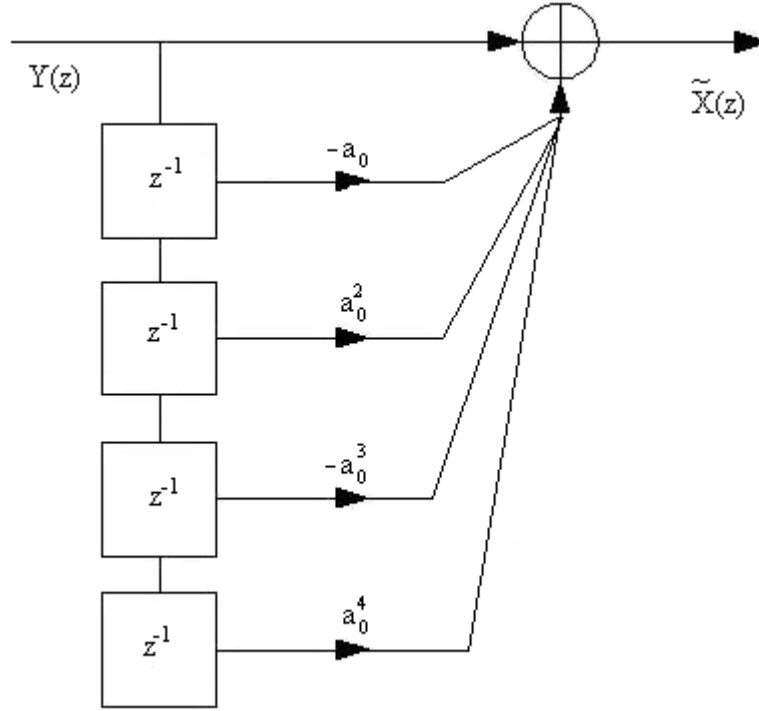
elde edilir. Bu durumda

$$\begin{aligned}
\tilde{X}(z) &= \bar{H}_{\text{FIR}}(z) Y(z) \\
&= Y(z) [1 - \alpha z^{-k} + \alpha^2 z^{-2k} - \alpha^3 z^{-3k} + \alpha^4 z^{-4k}]
\end{aligned} \quad (3.114)$$

şeklinde yazılabilir. Bu ifade zaman düzleminde yazılırsa

$$\tilde{x}(n) = y(n) - a_0 y(n-k) + a_0^2 y(n-2k) - \dots + a_0^4 y(n-4k) \quad (3.115)$$

elde edilir. Şekil 3.58.'de FIR yapıdaki kanal için FIR dengeleyicinin öbek çizgesi verilmiştir.



Şekil 3.58. FIR modellenmiş kanal için FIR yapıdaki dengeleyici

3.4.3.4. IIR Modellenmiş Bir Kanalin Dengelenmesi

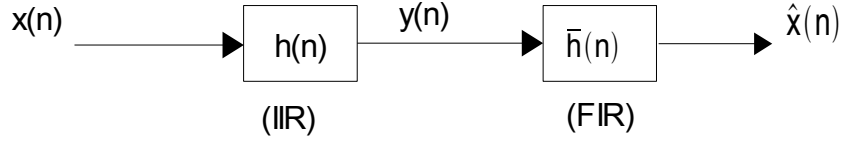
Yankı kanalı IIR yapıda modellenebilir. Şekil 3.44.'de bu yapıdaki kanalın öbek çizgesi verilmiştir. Sistemin geçiş işlevi

$$H(z) = \frac{1}{1 - \alpha z^{-k}} \quad (3.116)$$

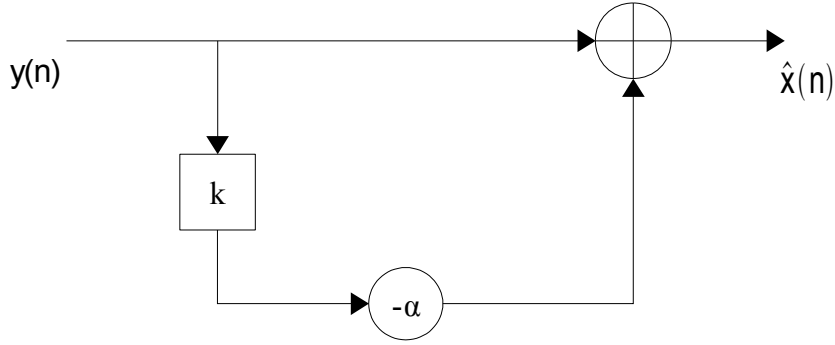
şeklindedir. Dengeleyicinin geçiş işlevi ise

$$\bar{H}(z) = 1 - \alpha z^{-k} \quad (3.117)$$

olarak bulunur. Şekil 3.59.' da IIR modellenmiş kanalın FIR süzgeç ile dengelenmesi verilmiştir. Şekil 3.60.' da ise IIR modellenmiş kanal için FIR yapıdaki dengeleyici öbek çizgesi verilmiştir.

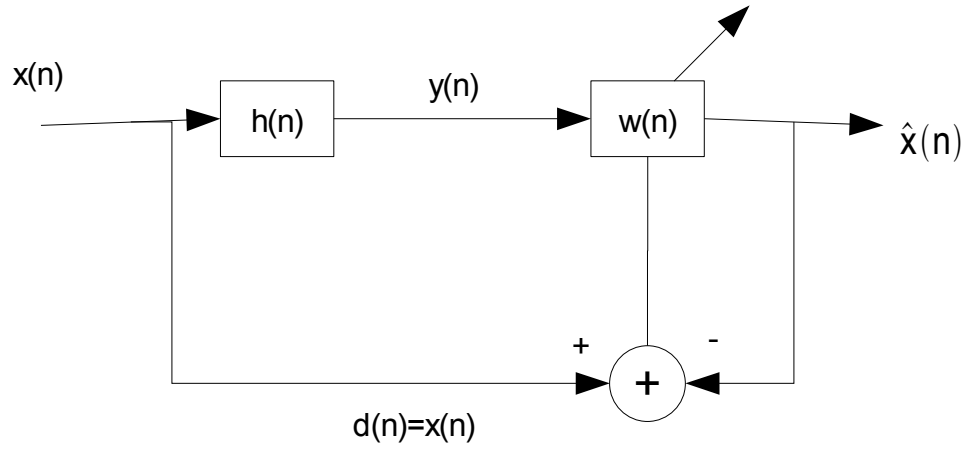


Şekil 3.59. IIR modellenmiş kanalın FIR süzgeç ile dengelenmesi



Şekil 3.60. IIR modellenmiş kanal için FIR yapıdaki dengeleyici

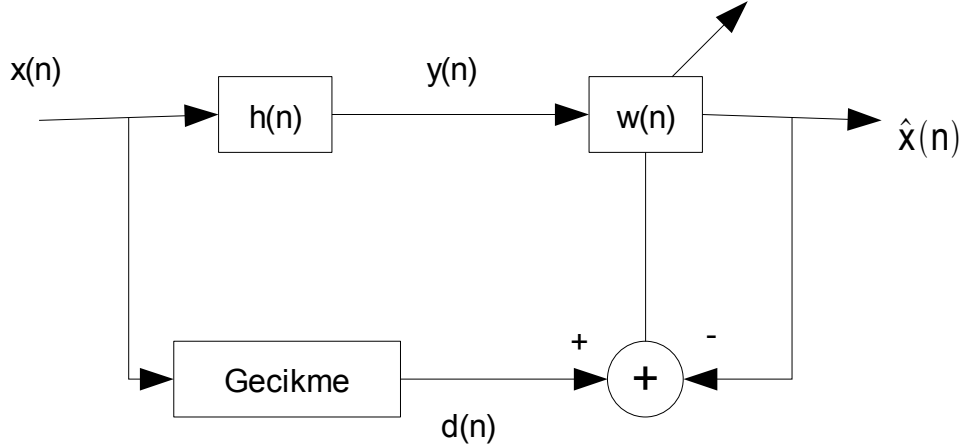
3.4.3.5. Uyarlanabilir Süzgeç İle Kanal Dengeleme ve Yankı Giderme



Şekil 3.61. Uyarlanabilir süzgeç kullanarak kanal dengeleme

Şekil 3.61.' da uyarlanabilir süzgeç ile kanal dengeleme öbek çizgesi verilmiştir. Bölüm 3.3.3.6. 'da anlatıldığı üzere uyarlama algoritması hata sinyalini en aza indirmeye çalışır. Hata sinyali sıfır olduğunda uyarlanabilir süzgeç tepkisi $w(n)$

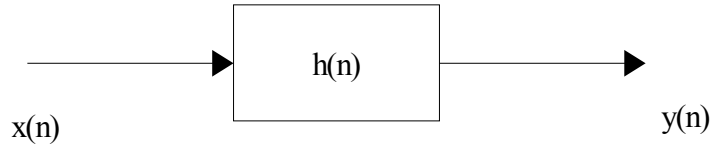
bilinmeyen sistem tepkisi $h(n)$ ' nin tersi olur. Sistemin nedenselliğini korumak için arzulanan sinyal $d(n)$, giriş sinyalinin gecikmiş olarak verilir[26]. Bu gecikmenin miktarı genellikle uyarlanabilir süzgecin derecesinin yarısı kadar alınır. Şekil 3.62.' de gecikme elemanı eklenmiş dengeleyici öbek çizgesi verilmiştir.



Şekil 3.62. Nedenselliği korumak için $d(n)$, $x(n)$ ' nin gecikmiş olarak aktarılır

Çoğu uygulamada giriş sinyali ve kanal hakkında bilgi sahibi olamayız. Bu durumda elimizde sadece çıkış sinyali hakkında bilgi ile kanalın istatistikleri hakkında bilgi bulunur. Mevcut bu bilgiler kullanılarak yapılan kanal dengelemeye kör kanal dengeleme (Blind Channel Equalization) veya kör ters katlama (Blind Deconvolution) denir[27].

3.4.3.6. Ters Katlama



Şekil 3.63. Bir sistemin öbek çizge gösterimi

Şekil 3.63.' de bir sistemin öbek çizge gösterimi verilmiştir. Girişi $x(n)$ ve birim vuruş tepkisi $h(n)$ olan bir sistemin çıkışı $x(n)$ ile $h(n)$ ' nin katlaması olarak

$$y(n)=h(n)*x(n) \quad (3.118)$$

şeklinde bulunur. Ayrık zamanlı sinyaller için katlama toplamı

$$y(n)=\sum_{i=-\infty}^{\infty} h(i)x(n-i)=\sum_{i=-\infty}^{\infty} x(i)h(n-i) \quad (3.119)$$

şeklindedir. Sistem N. dereceden ve nedensel ise eşitlik (3.119)

$$y(p)=\sum_{i=0}^{N-1} h(i)x(p-i)=\sum_{i=0}^{N-1} x(i)h(p-i) \quad ;(1 \leq p \leq n) \quad (3.120)$$

şeklini alır. Veya matris biçiminde

$$\begin{bmatrix} y(n) \\ y(n-1) \\ \cdot \\ \cdot \\ y(1) \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & \dots & 0 & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ 0 & 0 & 0 & \dots & h_{N-2} & h_{N-1} \end{bmatrix} \begin{bmatrix} x(n) \\ x(n-1) \\ \cdot \\ \cdot \\ x(2-N) \end{bmatrix} \quad (3.121)$$

$$\vec{y}(n)=H^T \vec{x}(n) \quad (3.122)$$

şeklinde gösterilir.

Eğer $y(n)$ biliniyor ancak $x(n)$ veya $h(n)$ ' den birisi bilinmiyor ise ters katlama ile bilinmeyen bulunabilir. $x(n)$ bilinmiyor ise

$$\begin{aligned} y(n) &= \sum_{i=0}^{N-1} h(i)x(n-i) \\ y(n) &= h(0)x(n) + \sum_{i=1}^{N-1} h(i)x(n-i) \\ x(n) &= \frac{1}{h(0)} \left[y(n) - \sum_{i=1}^{N-1} h(i)x(n-i) \right] \end{aligned} \quad (3.123)$$

şeklinde bulunur[28]. Burada

$$h(0) = \frac{y(0)}{x(0)} \quad (3.124)$$

dır. Matris biçiminde

$$\vec{x}(n) = H(H^T H)^{-1} \vec{y}(n) \quad (3.125)$$

olarak gösterilir. Eğer $h(n)$ bilinmiyorsa

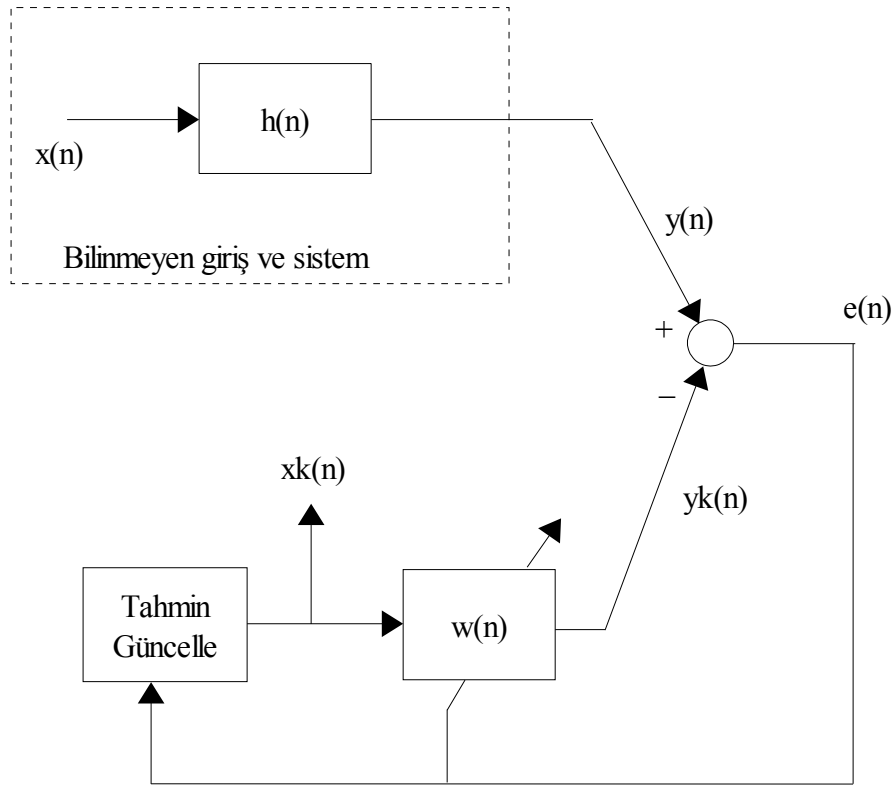
$$\begin{aligned} y(n) &= \sum_{i=0}^{N-1} x(i)h(n-i) \\ y(n) &= x(0)h(n) + \sum_{i=1}^{N-1} x(i)h(n-i) \\ h(n) &= \frac{1}{x(0)} \left[y(n) - \sum_{i=1}^{N-1} x(i)h(n-i) \right] \end{aligned} \quad (3.126)$$

şeklinde bulunur. Burada

$$x(0) = \frac{y(0)}{h(0)} \quad (3.127)$$

dır.

3.4.3.7. Kör Ters Katlama



Şekil 3.64. Kör ters katlama [29]

Şekil 3.64. 'de kör ters katlama öbek çizgesi verilmiştir. $y_k(n)$ tahmin edilen çıkış sinyali olup asıl çıkış sinyali $y(n)$ ' den çıkartılarak hata değeri elde edilir. Uyarlanabilir süzgecin uyarlama algoritması hatayı en küçük yapacak şekilde $w(n)$ katsayılarını günceller ve $w(n)$ katsayıları $h(n)$ katsayılarına yakınsar. Bu durumda $y_k(n)$, $y(n)$ ' ye ve $x_k(n)$ de $x(n)$ ' ye yakınsar. Tahmin edilen $x_k(n)$ sinyali

$$x_k(n+1) = \frac{1}{h_0} \left[y(n+1) - \sum_{i=1}^{N-1} w(i) x_k(n+1-i) \right] \quad (3.128)$$

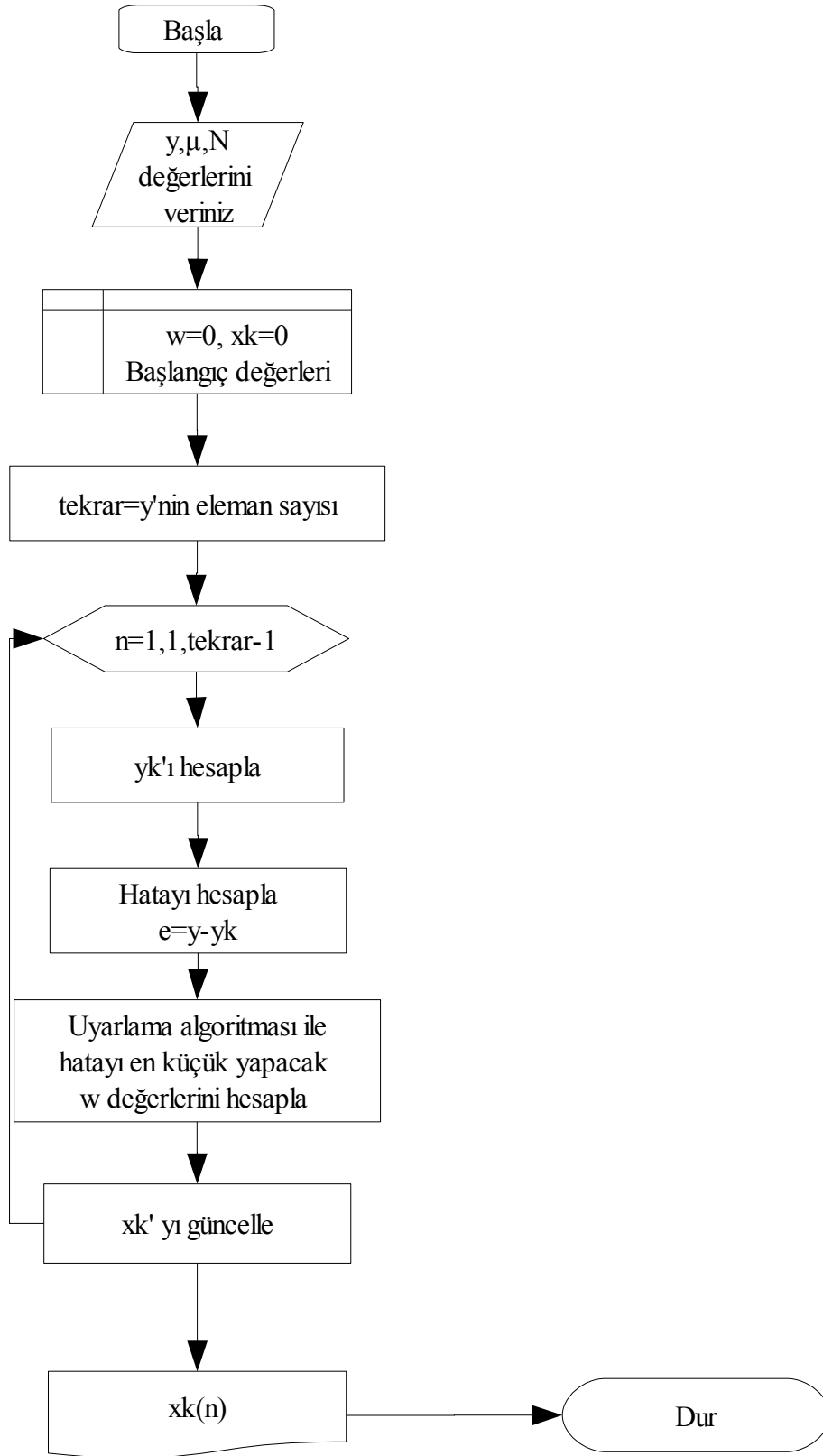
şeklinde güncellenir. Matris biçiminde

$$\vec{x}_k(n) = \frac{1}{h_0} [y(n+1) - \vec{w}^T \vec{x}] \quad (3.129)$$

olarak yazılabilir. Burada x durum değişkenleri olarak adlandırılır. x durum değişkenleri

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ \cdot \\ \cdot \\ x_N(n+1) \end{bmatrix} = \begin{bmatrix} 0 \\ x_1(n) \\ \cdot \\ \cdot \\ x_{N-1}(n) \end{bmatrix} + \delta \begin{bmatrix} h_0 \\ h_1 \\ \cdot \\ \cdot \\ h_{N-1} \end{bmatrix} \left[y(n+1) - \sum_{i=1}^{N-1} h_i x_i(n) \right] \quad (3.130)$$

şeklinde güncellenir. Burada δ uyarlama adımı, sıfır ile bir arasında değişen pozitif bir sabittir. Şekil 3.65.'de terskatlama akış çizgesi verilmiştir.



Şekil 3.65. Terskatlama akış çizgesi

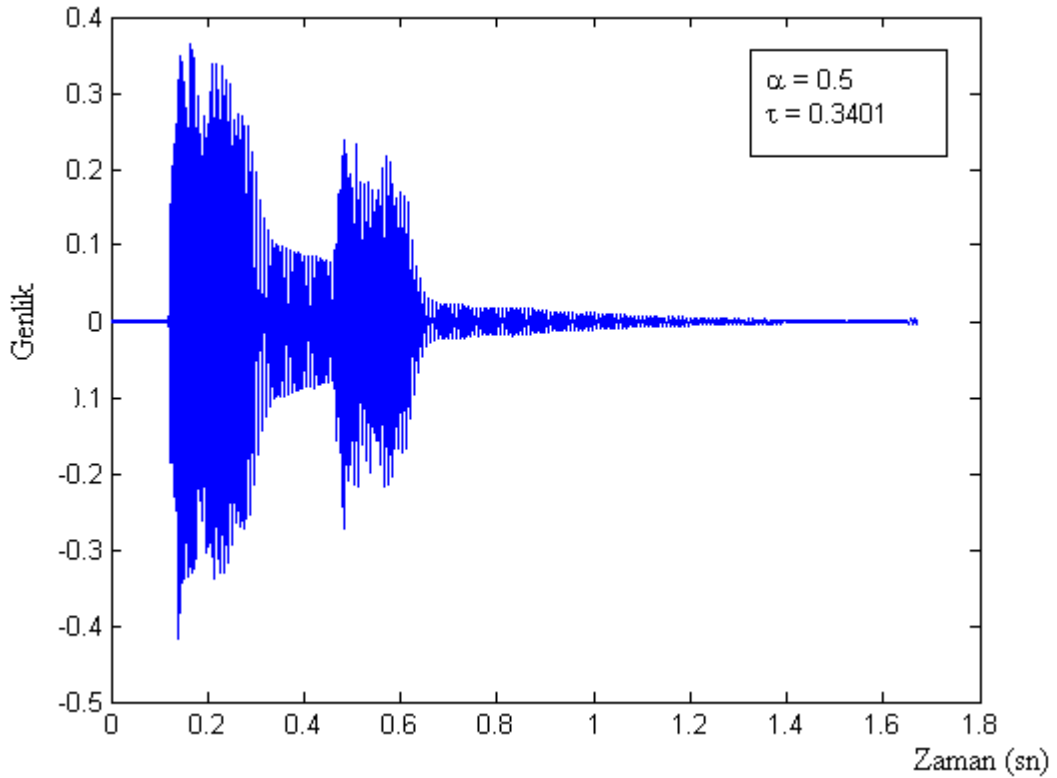
4. BULGULAR

Yankı gidermek için gerekli programlar Matlab ortamında yazılmış ve çalıştırılmıştır. Yankısız giriş sinyali olarak Şekil 3.42.' de verilen deneme sinyali kullanılmıştır.

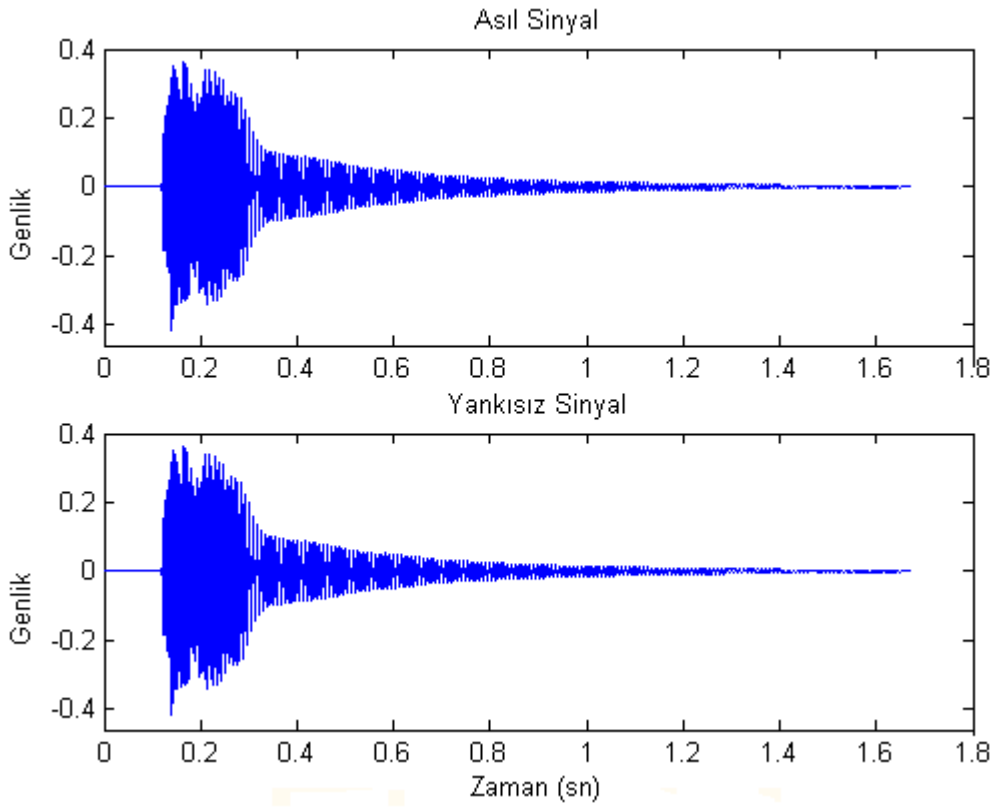
4.1. FIR Modellenmiş Bir Kanalın Dengelenmesi

Yankı yolu Şekil 3.54.'de verildiği gibi modellenmiş ve deneme sinyaline yapay olarak tek yankı eklenmiştir. Yankının gecikme miktarı yankılı sinyalin ($y(n)$) özilinti grafiğinden anlaşılabilir. Daha sonra Şekil 3.56.'da verilen IIR dengeleyici kullanılarak ses yankısı giderilmiştir. Bu iş için Ek A. 'da verilen denkleIIR.m fonksiyonunu kullanan FIR_IIR.m dosyası kullanılmıştır.

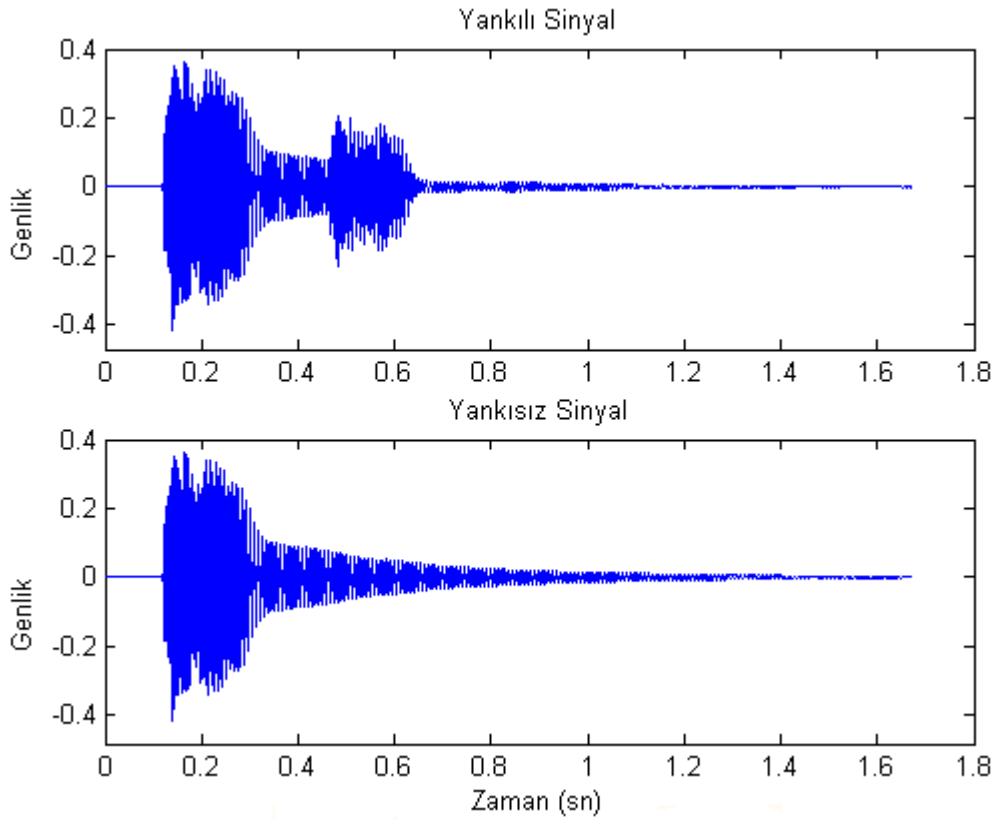
```
>> [x,f]=wavread('deneme.wav'); % asıl sinyal
>> y=yanki_FIR(x,0.5,15000); %Yankı ekle
>> yeni=denkleIIR(y,44100,0.5,15000); % Yankı gider
```



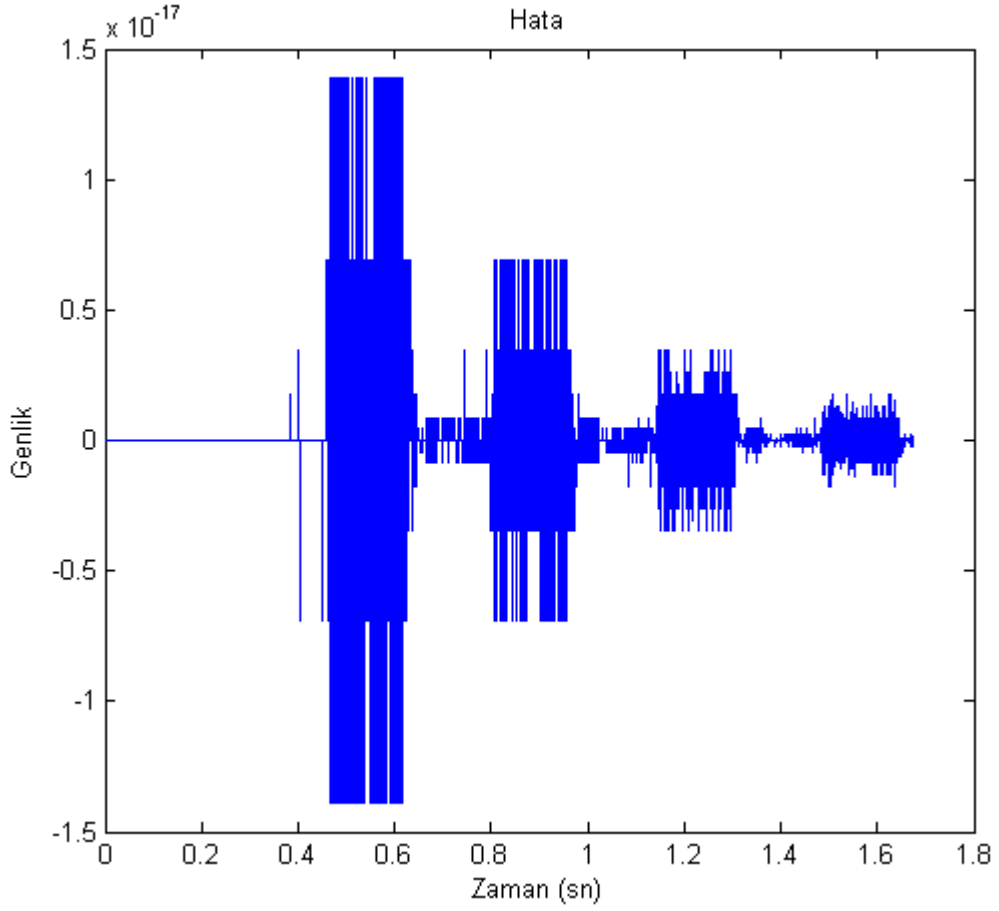
Şekil 4.1. Tek yankılı sinyal



Şekil 4.2. Asıl ve yankısız sinyal (FIR-IIR dengeleme)



Şekil 4.3. Yankılı ve yankısız sinyal (FIR-IIR dengeleme)



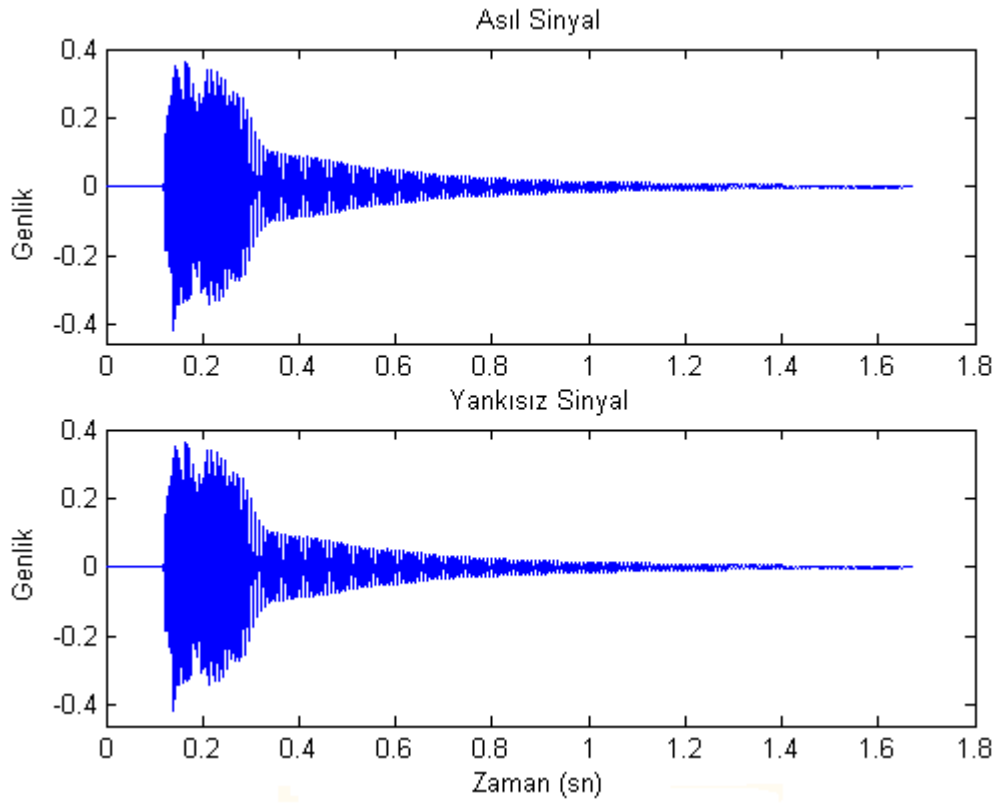
Şekil 4.4. Hata sinyali(FIR-IIR dengeleme)

Şekil 4.1.' de tek yankılı sinyal, Şekil 4.2.' de asıl ve yankısız sinyaller , Şekil 4.3.' de ise yankılı ve yankısız sinyaller verilmiştir .Asıl sinyal ile yankısı giderilmiş (yankısız) sinyal arasındaki fark hata sinyalidir ve Şekil 4.4.' de verilmiştir. Şekilde görüldüğü gibi hata çok küçük olup, 10^{-17} civarındadır. Ayrıca ikisi arasındaki ilinti katsayısı 1 çıkmaktadır. Bunlar yankının başarılı bir şekilde giderildiğini göstermektedir.

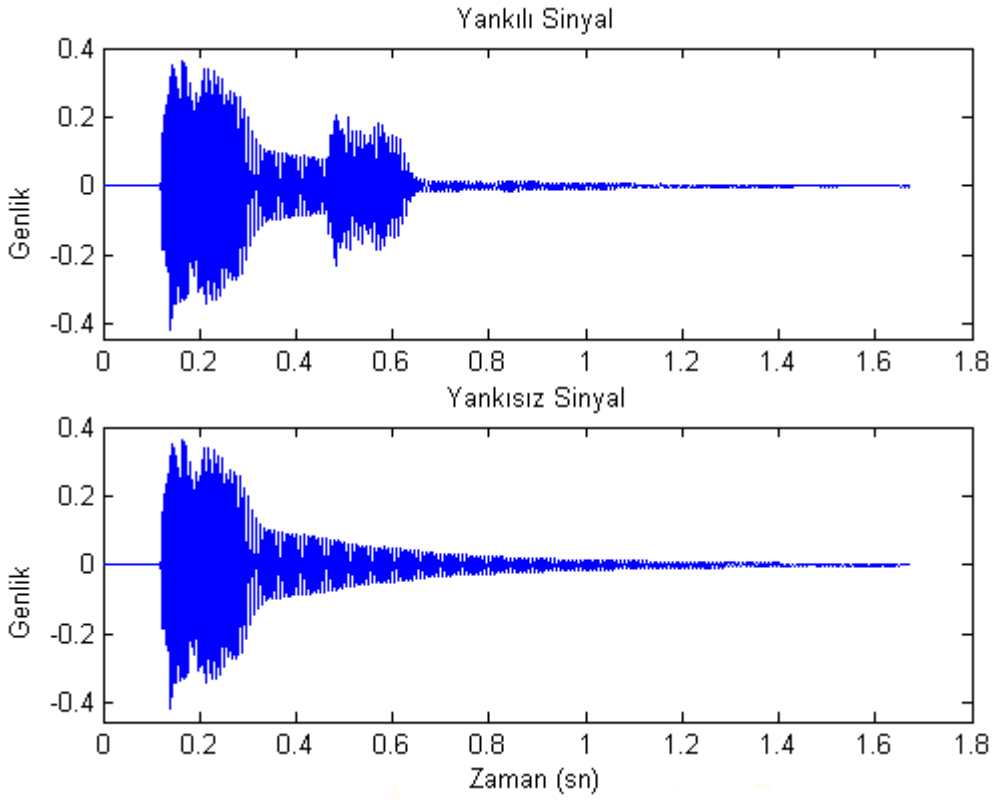
FIR modellenmiş bir yankı kanalı yine FIR yapıdaki bir dengeleyici kullanarak dengelenebilir. Şekil 3.58.' de verilen FIR dengeleyici yapısı kullanılabilir. Ek A.'da verilen denkleFIR.m fonksiyonunu kullanan FIR_FIR.m dosyası ile FIR dengeleme yapılmıştır.

```
>> y=yanki_FIR(x,0.5,15000); %Yankı ekle
>> yeni=denkleFIR(y,0.5,15000); %Yankıyı gider
```

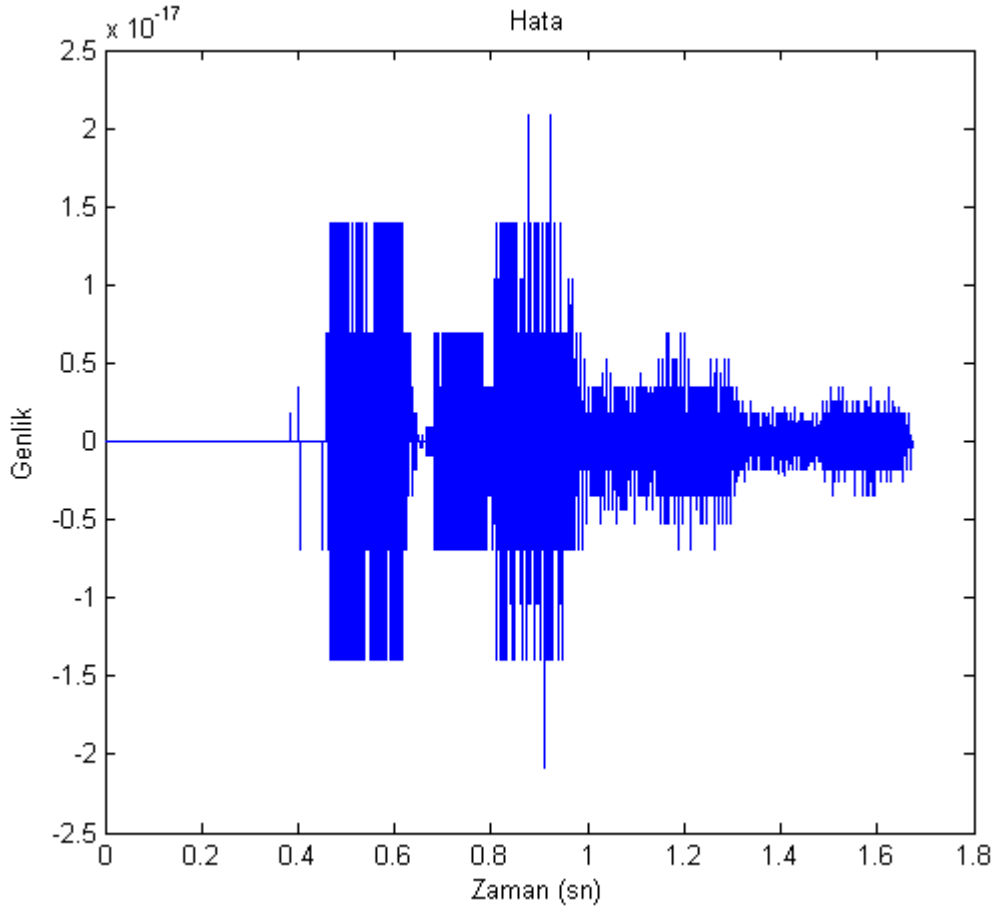
Asıl sinyal ile yankısız sinyalin ilinti katsayısı 1 çıkmaktadır.



Şekil 4.5. Asıl ve yankısız sinyal (FIR-FIR dengeleme)



Şekil 4.6. Yankılı ve yankısız sinyal(FIR-FIR dengeleme)



Şekil 4.7. Hata sinyali(FIR-FIR dengeleme)

Şekil 4.5.' de asıl ve yankısız sinyaller , Şekil 4.6.' da ise yankılı ve yankısız sinyaller ve Şekil 4.7.' de hata sinyali verilmiştir.

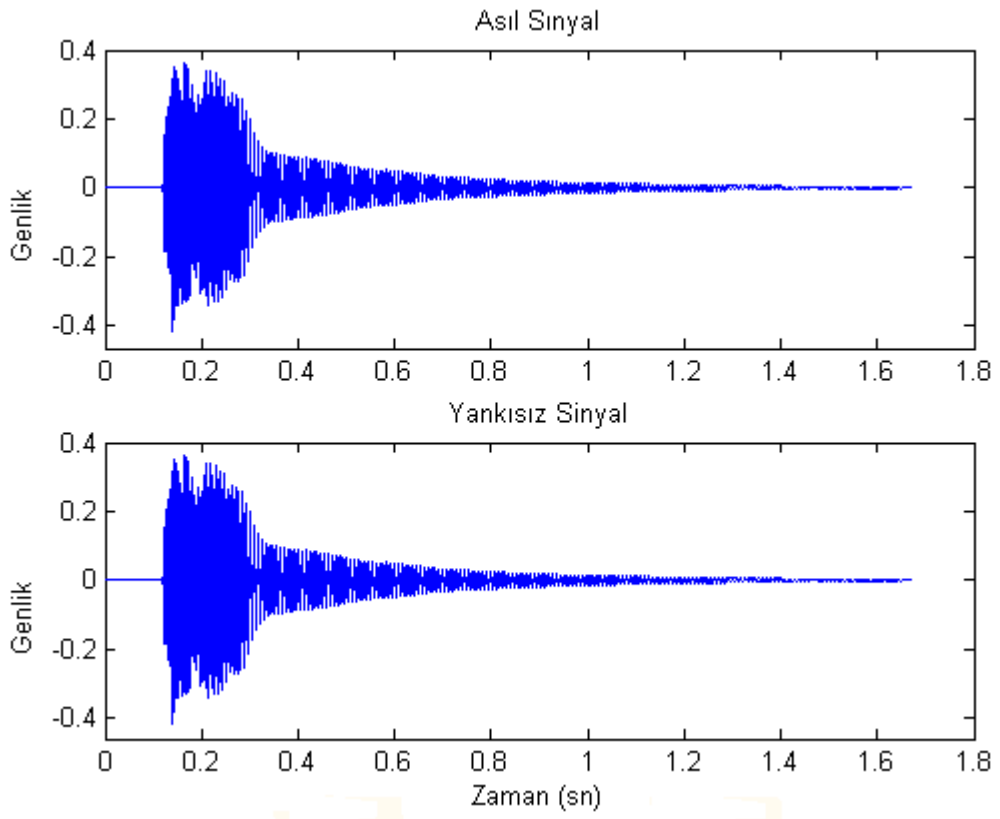
4.2. IIR Modellenmiş Bir Kanalı Dengeleme

Çok yankılı bir kanal Şekil 3.44.'de verildiği gibi modellenebilir. Buradaki yankıların özelliği yankı gecikmelerinin eşit olmasıdır. Çok yankılı bir sinyalin özilintisi Şekil 3.50.' de verilmiştir. Yankı gecikmeleri eşit olduğu için tek bir yankının gecikmesini belirlemek yeterlidir. FIR dengeleme için Ek A.'da verilen dengele_FIR.m fonksiyonunu kullanan IIR_FIR.m dosyası kullanılmıştır.

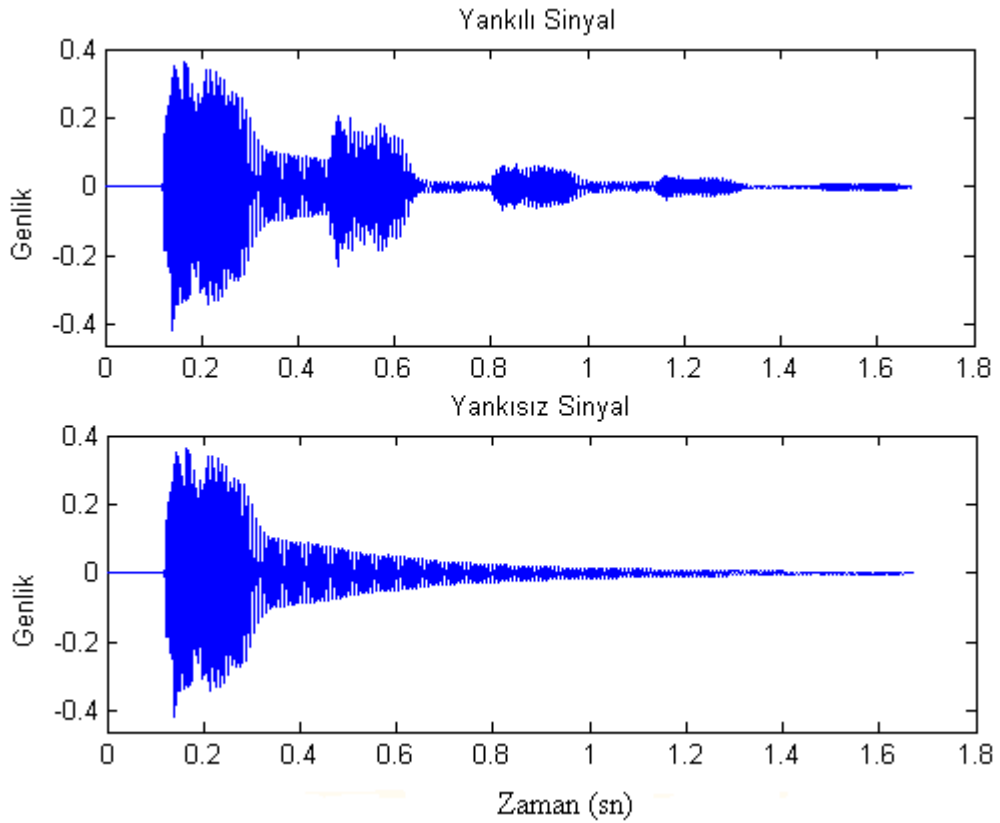
```
>> y=yanki_IIR(x,0.5,15000); %yankı ekle
```

```
>> yeni=dengele_FIR(y,0.5,15000);
```

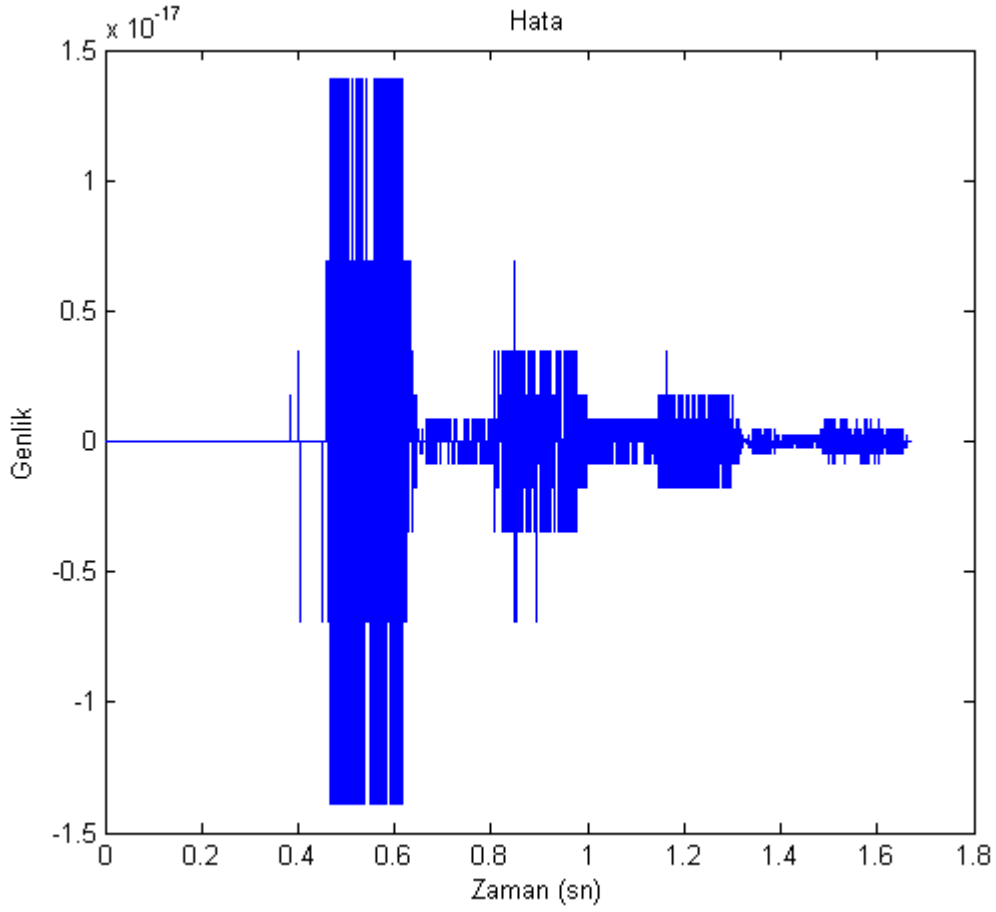
Asıl sinyal ile yankısız sinyalin ilinti katsayısı 1 çıkmaktadır.



Şekil 4.8. Asıl ve Yankısız sinyal (IIR-FIR dengeleme)



Şekil 4.9. Yankılı ve yankısız sinyal (IIR-FIR dengeleme)



Şekil 4.10. Hata sinyali

Şekil 4.8.' de asıl ve yankısız sinyaller , Şekil 4.9.' da ise yankılı ve yankısız sinyaller ve Şekil 4.10.' da hata sinyali verilmiştir.

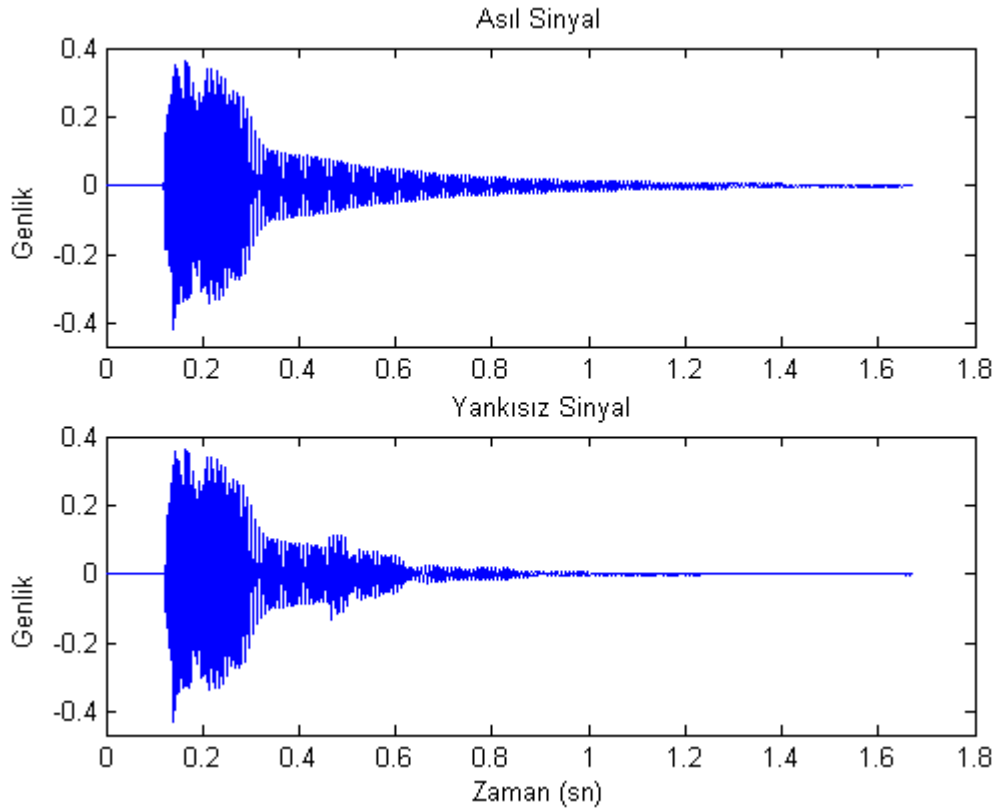
4.3. Uyarlanabilir Süzgeç İle Kanal Dengeleme Ve Yankı Giderme

Uyarlanabilir süzgeç kullanarak kanal dengeleme ile ses sinyalindeki yankılar giderilebilir. Bu çalışmada uyarlanabilir süzgeç yapısı olarak transversal FIR süzgeç yapısı kullanıldı. Uyarlama algoritmaları olarak da LMS, NLMS ve RLS algoritmaları kullanıldı ve sonuçlar karşılaştırıldı. Kanal dengeleme için Şekil 3.61. 'de verilen yapı kullanıldı.

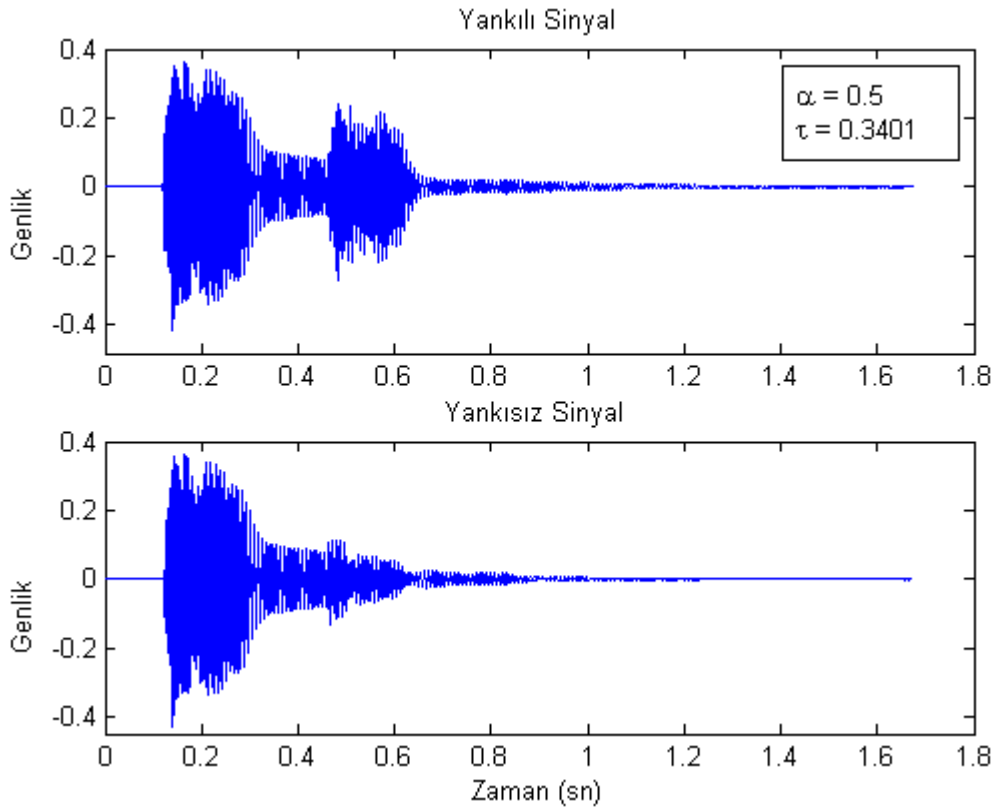
4.3.1. LMS Algoritması Kullanan Dengeleyici

İlk olarak tek yankılı kanalın dengelenmesi gerçekleştirildi. Yankılı sinyal $y(n)$ süzgecin girişi, asıl sinyal $x(n)$ ise arzulanan sinyal olarak verilir. Gerekli program Ek A.'da dengeleyici_lms.m fonksiyonunu kullanan lms_dengele.m dosyasında verilmiştir.

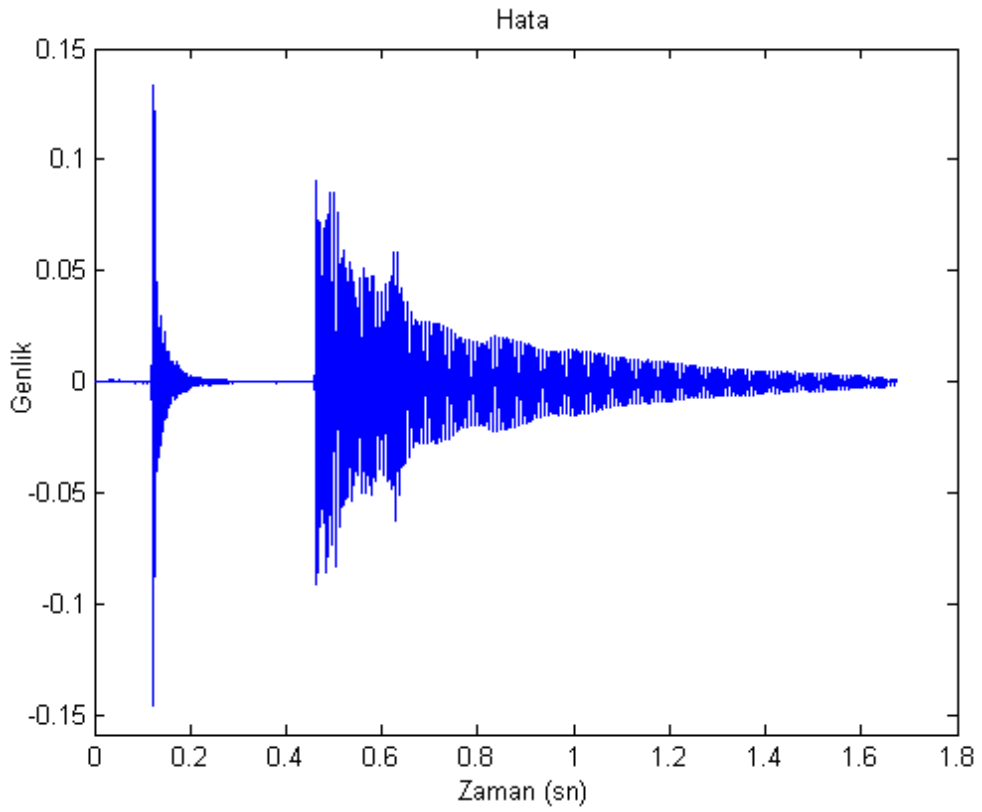
```
>> [x,f]=wavread('deneme.wav'); % Asıl sinyal
>> y=yanki_FIR(x,0.5,15000); % Yankı ekle
>> y=dengeleyici_lms(y,x,0.5,10); % Yankıyı gider
```



Şekil 4.11. LMS algoritması ile kanal dengeleme($\mu=0.5$, $N=10$)



Şekil 4.12. Yankılı ve yankısız sinyaller ($\mu=0.5$, $N=10$)



Şekil 4.13. Hata sinyali ($\mu=0.5$, $N=10$)

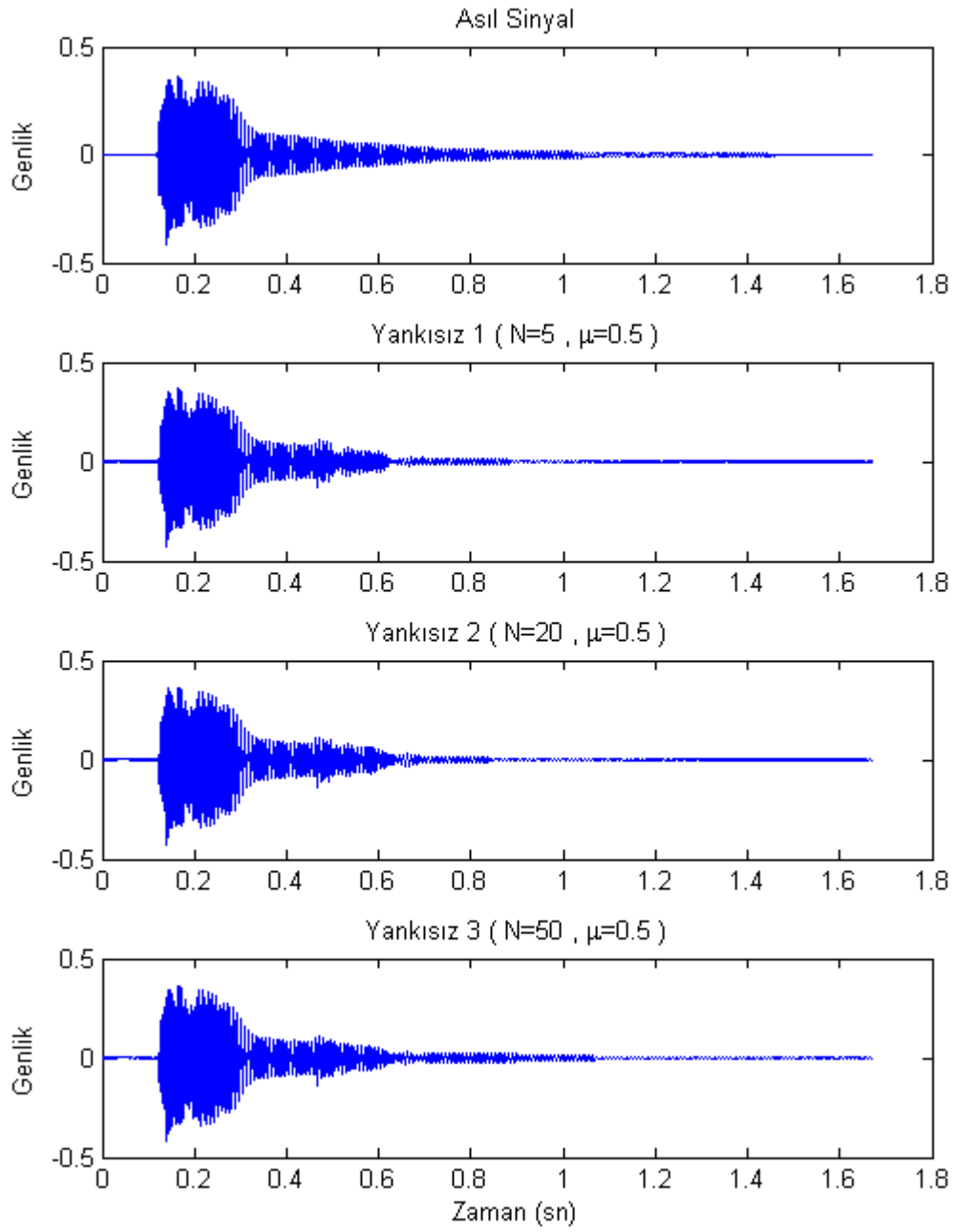
Şekil 4.11.' de asıl ve yankısız sinyaller , Şekil 4.12.' de ise yankılı ve yankısız sinyaller ve Şekil 4.13.' de hata sinyali verilmiştir. Farklı μ ve süzgeç uzunluğu N değerleri için program çalıştırılmış ve sonuçlar karşılaştırılmıştır. Şekil 4.14.' de $\mu=0.5$ olarak sabit tutulmuş ve N=5, N=20, N=50 değerleri için çıkışlar verilmiştir. Şekil 4.16.' da ise N=10 sabit tutularak $\mu=0.5$, $\mu=0.3$ ve $\mu=0.1$ değerleri için çıkışlar verilmiştir. Şekil 4.15. ve Şekil 4.17.' de ise hata sinyalleri verilmiştir. Tablo 4.1. ve Tablo 4.2.' de farklı N ve μ değerleri için çıkışların asıl sinyal ile ilinti katsayısı ve hata sinyalinin kareleri toplamı verilmiştir. N ve μ arttıkça ilinti katsayısı artmakta, hataların kareleri toplamı ise azalmaktadır. İlintinin yüksek çıkması yankısız sinyalin asıl sinyale daha çok benzediğini göstermektedir. Hata karelerin toplamının azalması da oluşan hatanın daha az olduğunu yani asıl sinyale daha fazla yaklaşıldığını gösterir.

Tablo 4.1: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı

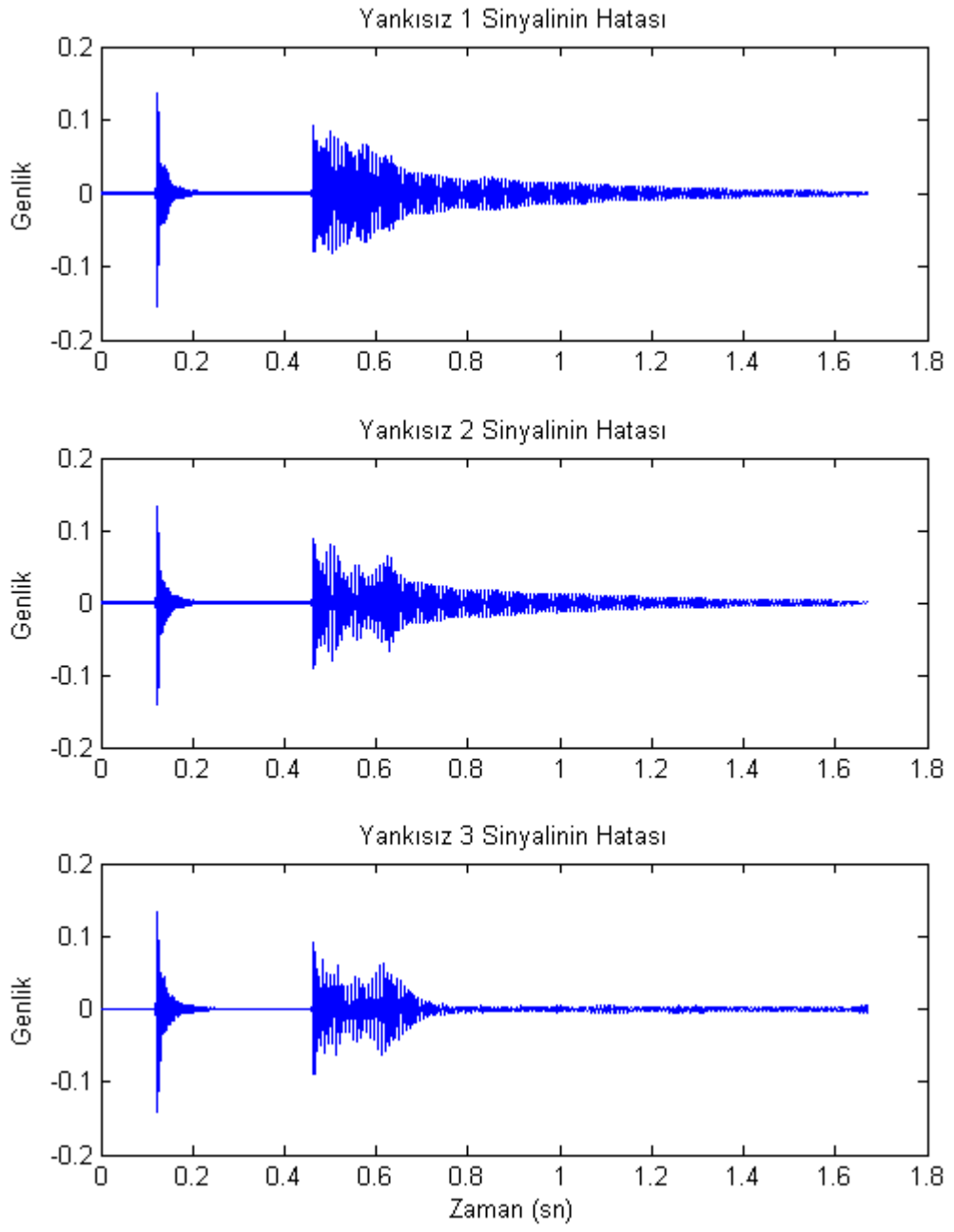
$\mu=0.5$	İlinti Katsayısı	Hata Karelerin Toplamı
N=5	0.9640	12.4251
N=10	0.9673	11.3082
N=20	0.9733	9.2615
N=50	0.9832	5.8426
N=100	0.9868	4.5996

Tablo 4.2: Farklı μ değerleri için ilinti katsayısı ve hata karelerin toplamı

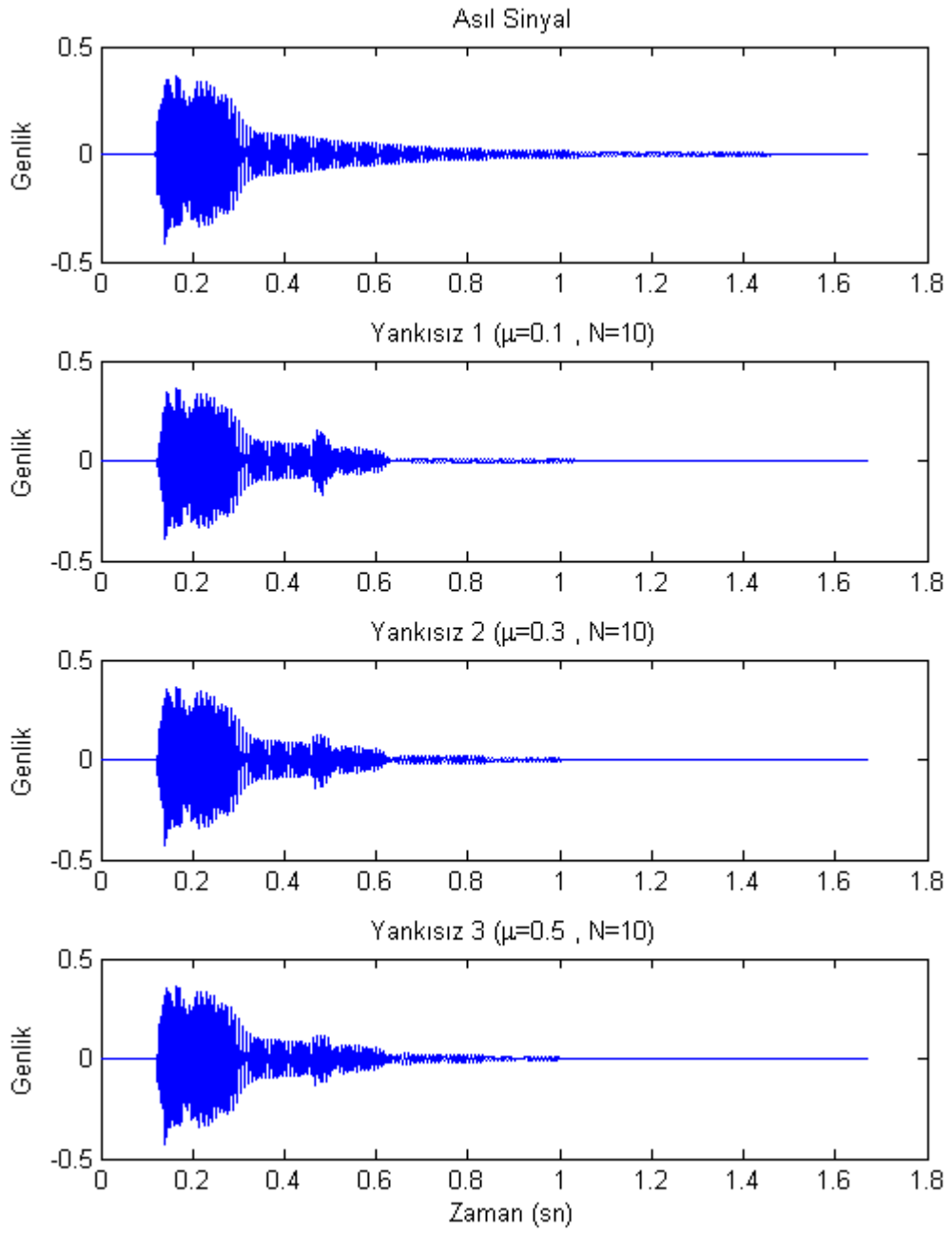
N=10	İlinti Katsayısı	Hata Karelerin Toplamı
$\mu=0.1$	0.9527	16.2372
$\mu=0.2$	0.9607	13.5357
$\mu=0.3$	0.9640	12.4199
$\mu=0.4$	0.9660	11.7613
$\mu=0.5$	0.9673	11.3082



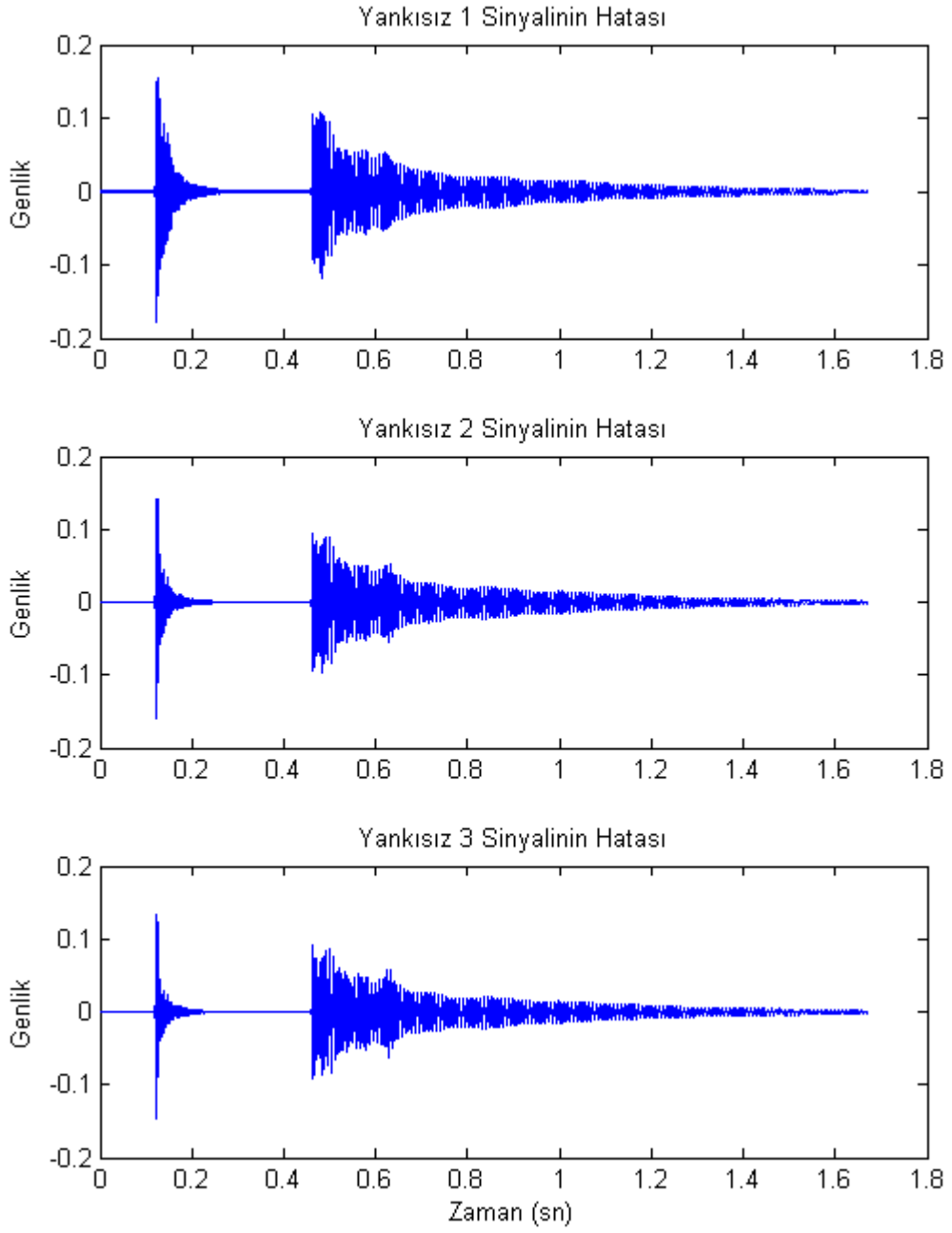
Şekil 4.14. Farklı süzgeç uzunlukları için çıkış sinyalleri



Şekil 4.15. Farklı N değerlerinde oluşan hata sinyalleri



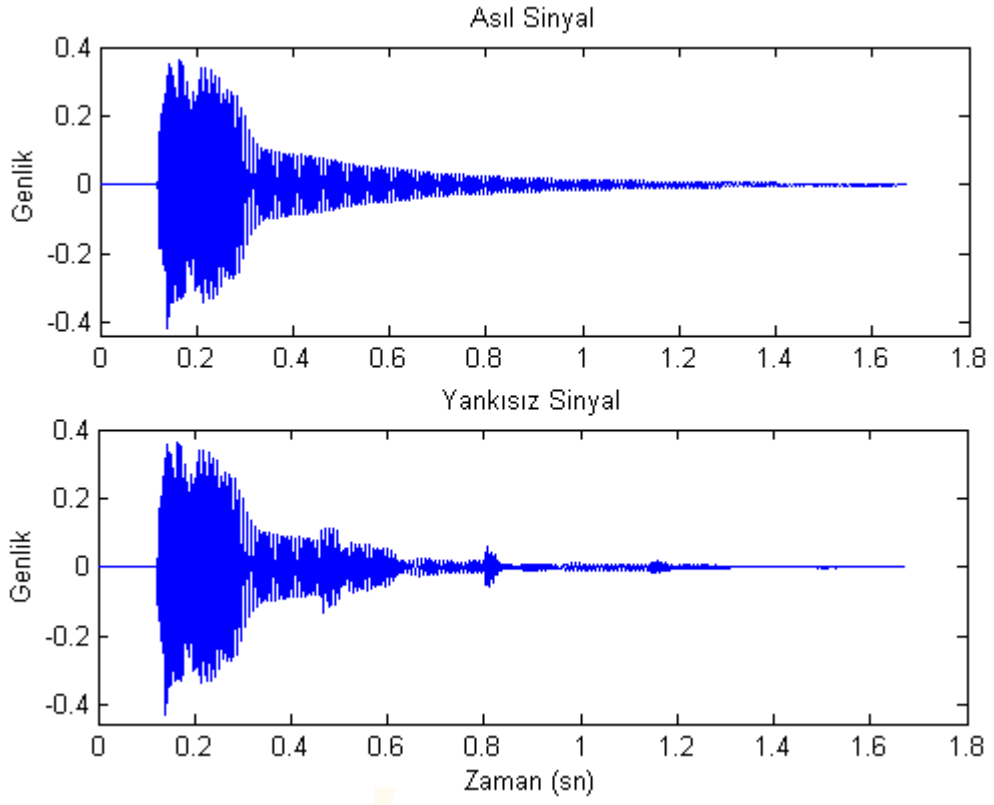
Şekil 4.16. Farklı μ değerleri için çıkış sinyalleri



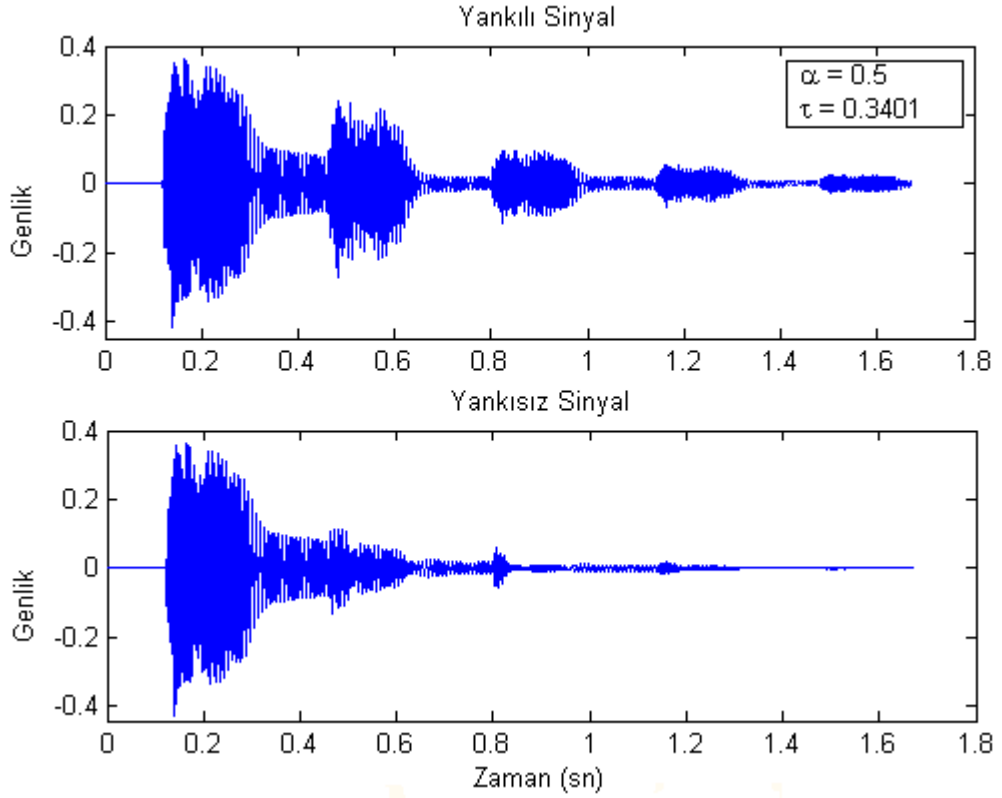
Şekil 4.17. Farklı μ değerlerinde oluşan hata sinyalleri

Çok yankılı sinyallerin yankılarını gidermek için de uyarlanabilir süzgeç kullanılabilir.

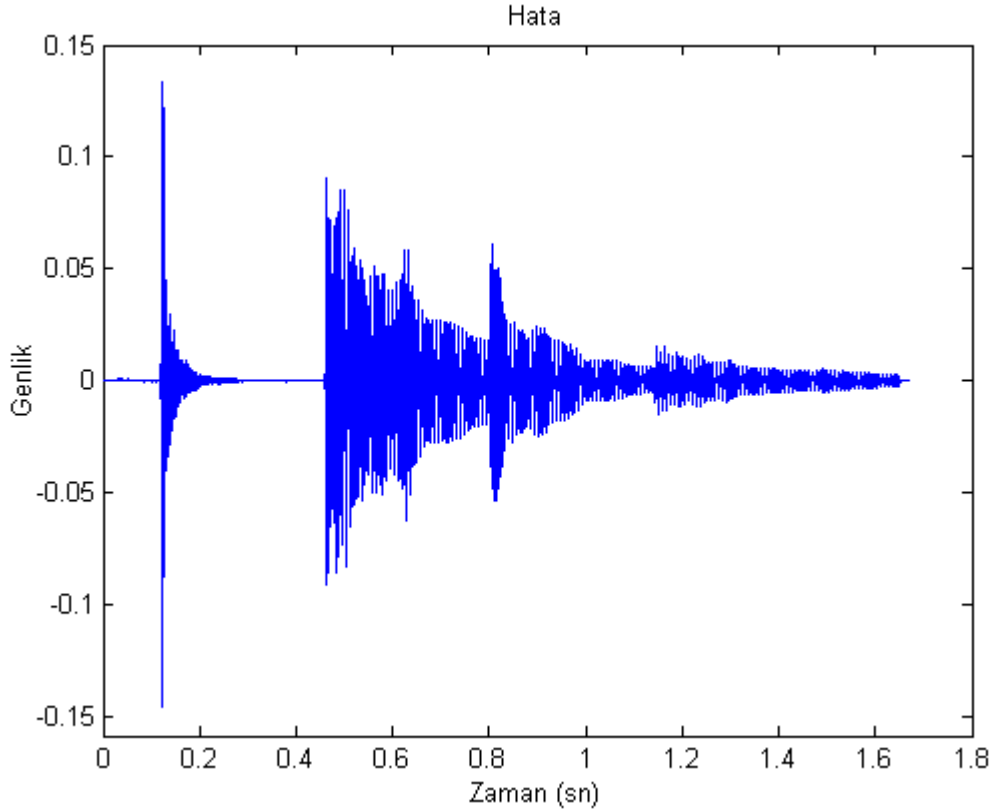
Uyarlanabilir süzgeç için $N=20$ ve $\mu=0.5$ seçilmiştir. Asıl sinyal ile yankısız sinyal arasındaki ilinti katsayısı 0.9780 ve hataların kareleri toplamı 7.6288 çıkmıştır. Şekil 4.18. ,4.19 ve 4.20.' de asıl, çok yankılı, yankısız ve hata sinyalleri verilmiştir.



Şekil 4.18. LMS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)



Şekil 4.19. LMS ile yankısı giderilmiş çok yankılı sinyal(Yankılı ve yankısız sinyaller)



Şekil 4.20. Hata sinyali ($\mu=0.5$, $N=10$)

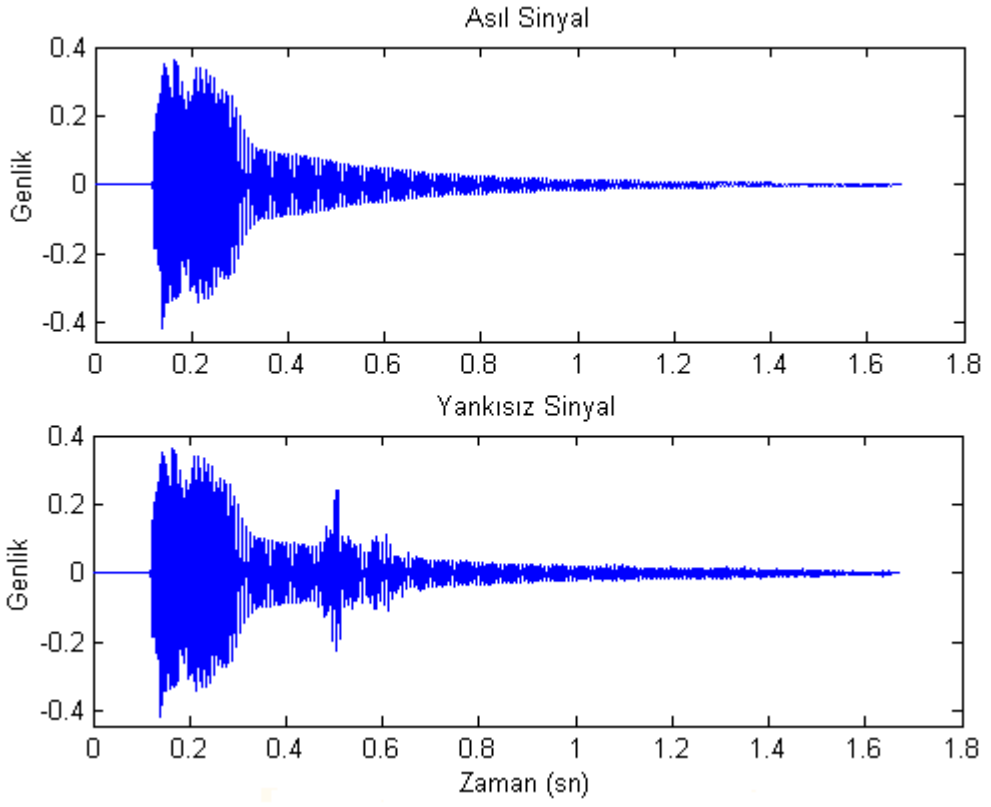
4.3.2. NLMS Algoritması Kullanan Dengeleyici

Uyarlanabilir süzgeç yapısı korunarak uyarlama algoritması olarak NLMS kullanılmıştır. LMS algoritmasında μ adım boyutu sabit kalırken Normalize LMS algoritmasında sürekli güncellenmektedir. β ,(0,2) açık aralığında bir sabit olmak üzere μ adım boyu

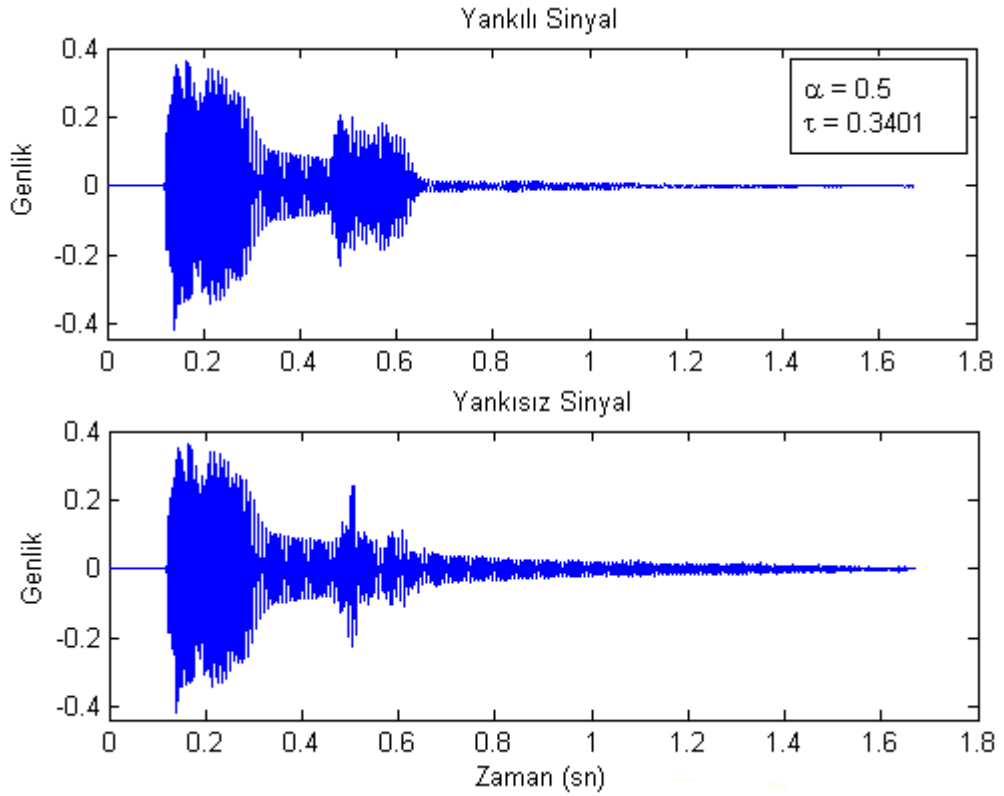
$$\mu(n) = \frac{\beta}{x^H(n)x(n)} = \frac{\beta}{\|x(n)\|^2} \quad (4.1)$$

şeklinde güncellenir. Bu iş için kullanılan program Ek A.' da dengeleyici_nlms.m fonksiyonunu kullanan nlms_dengele.m dosyasında verilmiştir. Şekil 4.21. ,4.22 ve 4.23.' de asıl, çok yankılı, yankısız ve hata sinyalleri verilmiştir.

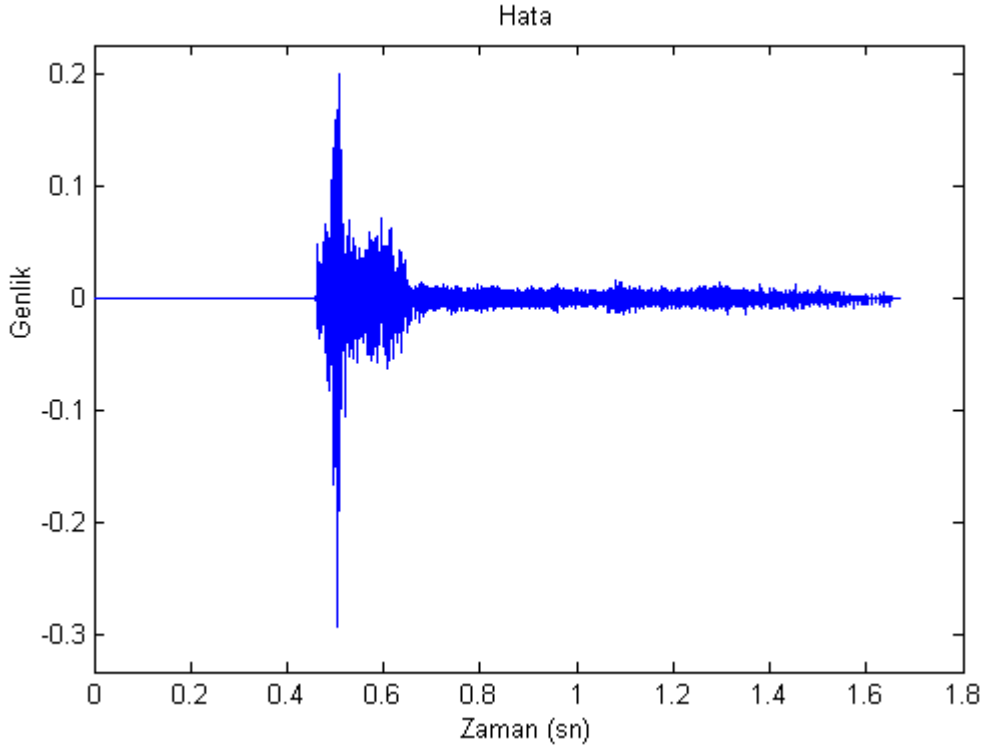
```
>> [x,f]=wavread('deneme.wav'); % Asıl sinyal
>> y=yanki_FIR(x,0.5,15000); % Yankı ekle
>> yeni=dengeleyici_nlms(y,x,0.5,10); % Yankı gider
```



Şekil 4.21. NLMS algoritması ile kanal dengeleme($\mu_0=0.5$, $N=10$)

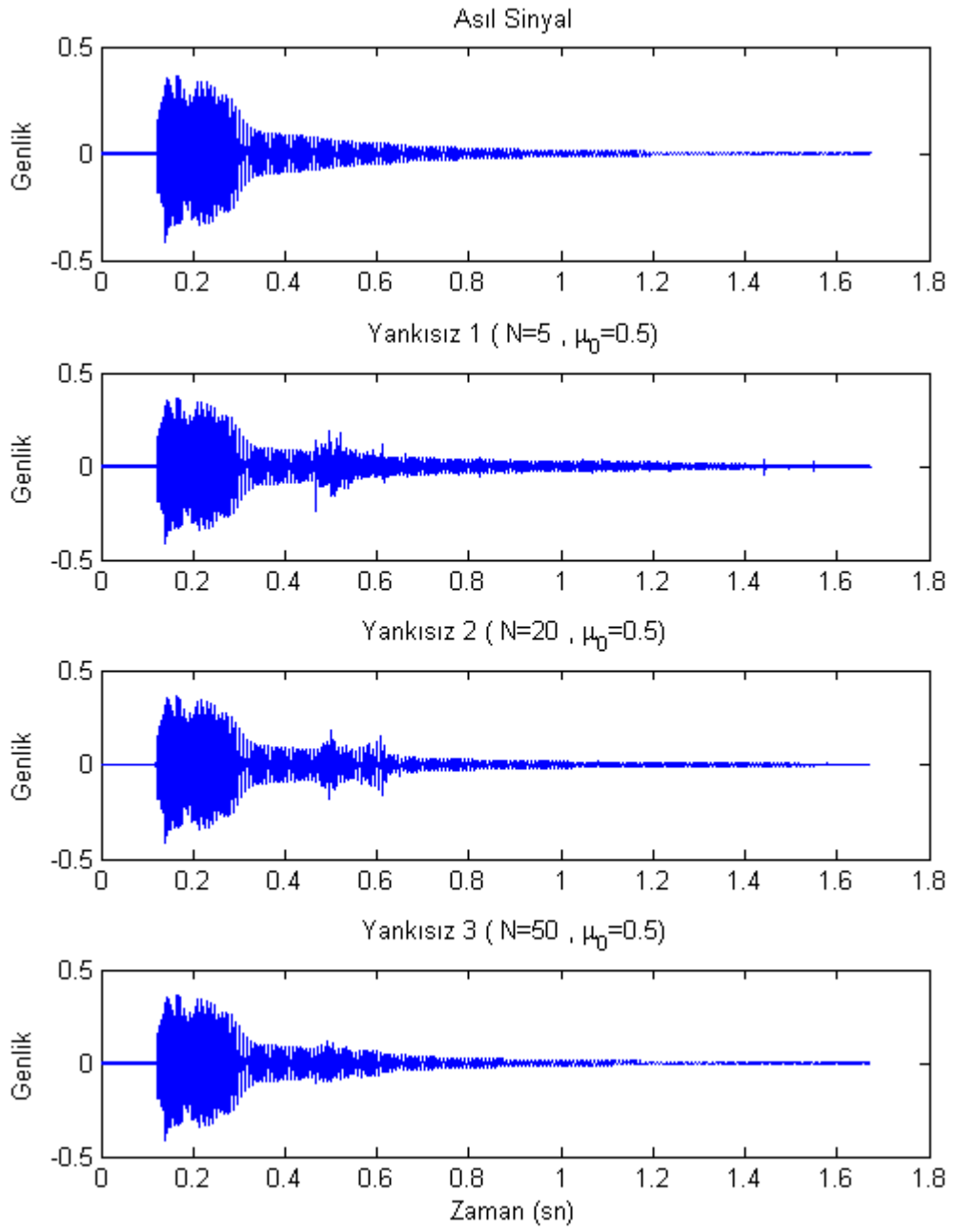


Şekil 4.22. Yankılı ve yankısız sinyaller ($\mu_0=0.5$, $N=10$)

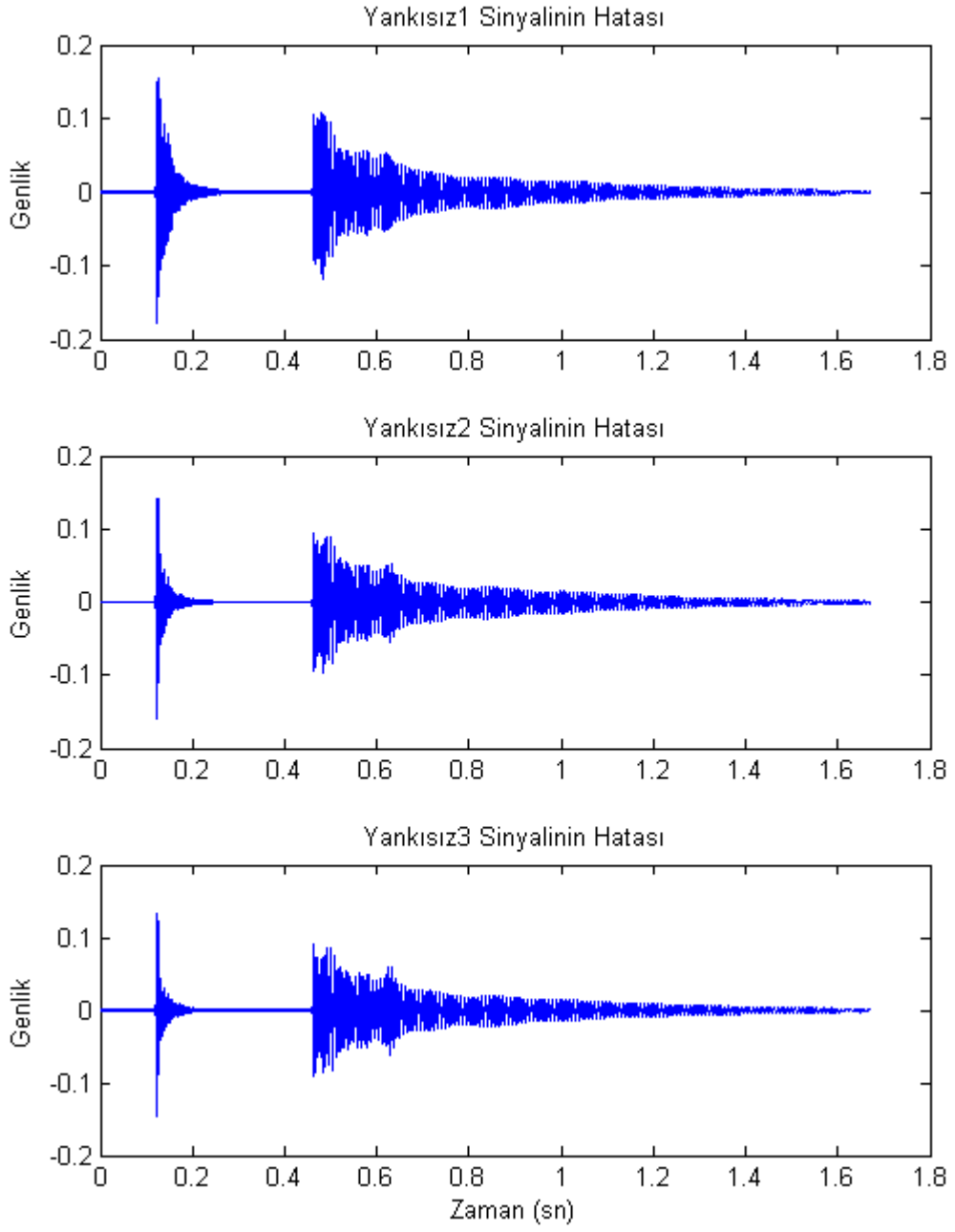


Şekil 4.23. Hata sinyali ($\mu_0=0.5$, $N=10$)

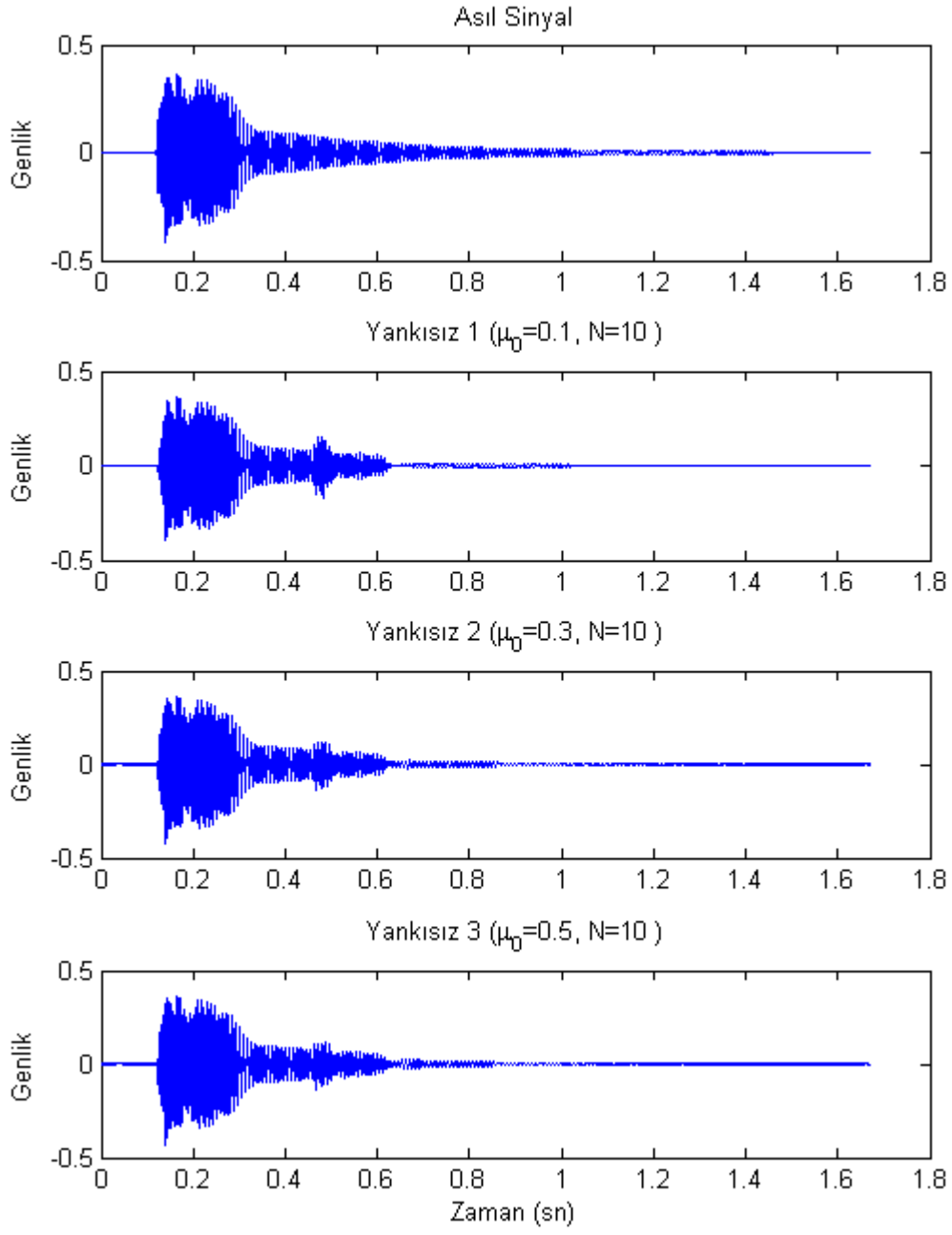
Farklı μ ve süzgeç uzunluğu N değerleri için program çalıştırılmış ve sonuçlar karşılaştırılmıştır. Şekil 4.24.'de $\mu_0=0.5$ olarak sabit tutulmuş ve $N=5$, $N=20$, $N=50$ değerleri için çıkışlar verilmiştir. Şekil 4.26.' da ise $N=10$ sabit tutularak $\mu=0.5$, $\mu=0.3$ ve $\mu=0.1$ değerleri için çıkışlar verilmiştir. Şekil 4.25. ve 4.27.' de ise hata sinyalleri verilmiştir. Tablo 4.3. ve Tablo 4.4. 'de farklı N ve μ değerleri için çıkışların asıl sinyal ile ilinti katsayısı ve hata sinyalinin kareleri toplamı verilmiştir. N arttıkça iyileşme görülmektedir. μ ' nün değişimi ile belirli bir düzelme görülememektedir. NLMS algoritması μ adım boyutunu sürekli olarak güncellemekte ve bu sayede LMS algoritmasına göre daha iyi sonuçlar vermektedir.



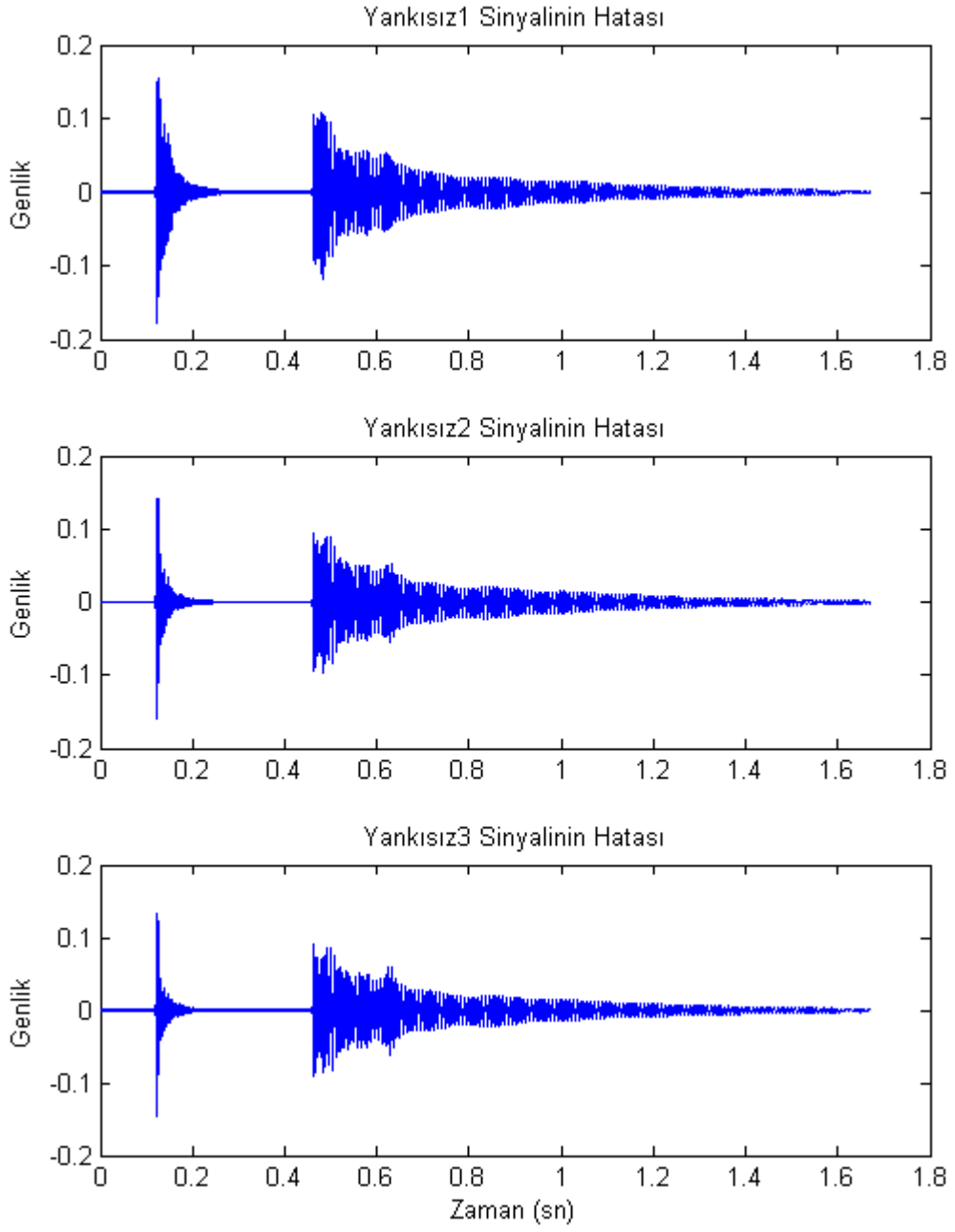
Şekil 4.24. Farklı süzgeç uzunlukları için çıkış sinyalleri



Şekil 4.25 Farklı N değerlerinde oluşan hata sinyalleri



Şekil 4.26. Farklı μ değerleri için çıkış sinyalleri



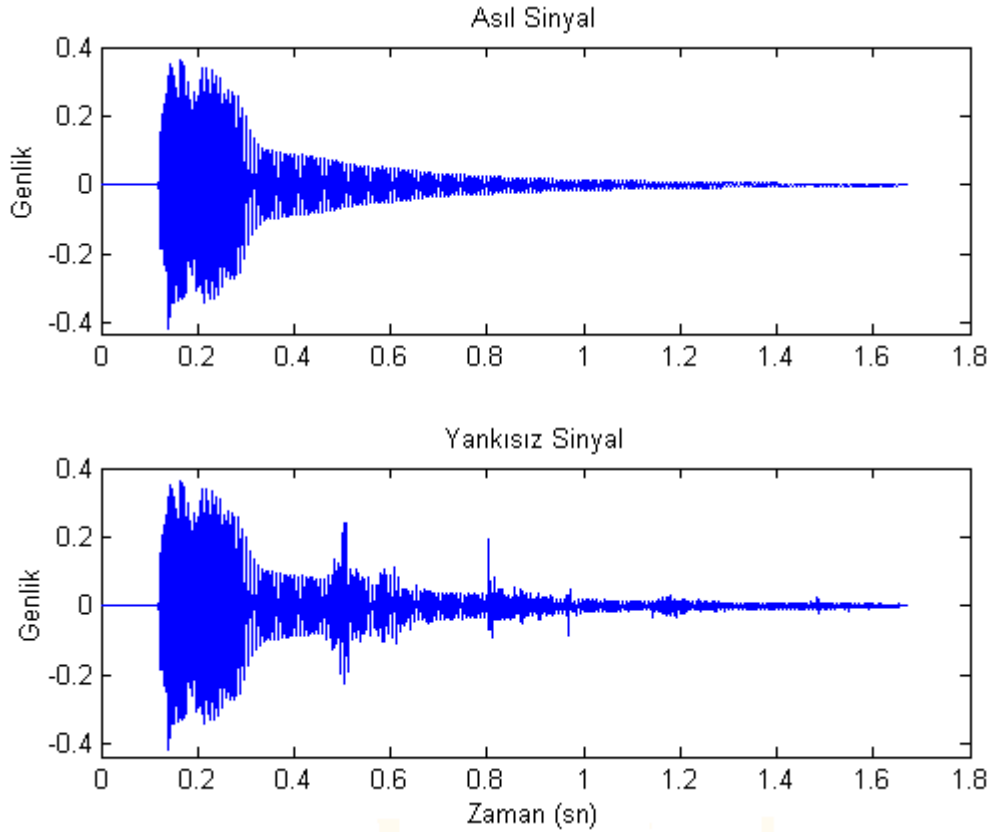
Şekil 4.27. Farklı μ değerlerinde oluşan hata sinyalleri

Tablo 4.3: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı

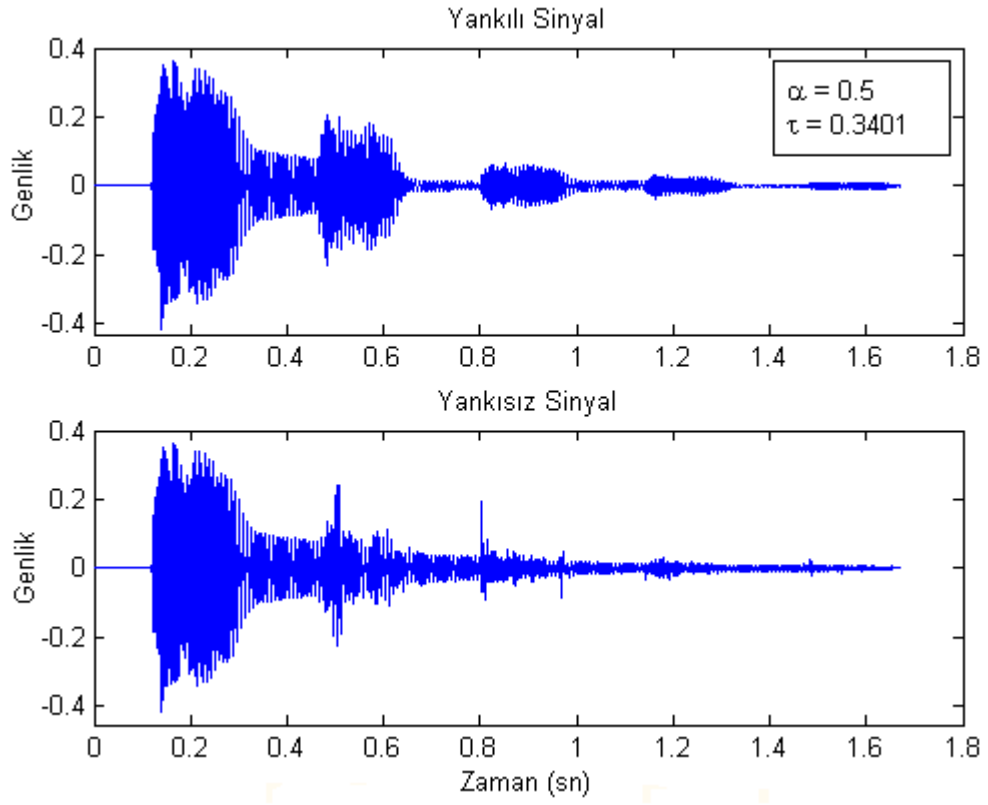
$\mu_0=0.5$	İlinti Katsayısı	Hata Karelerin Toplamı
N=5	0.9780	7.6963
N=10	0.9796	7.1754
N=20	0.9857	5.0733
N=50	0.9923	2.6946
N=100	0.9929	2.4868

Tablo 4.4: Farklı μ değerleri için ilinti katsayısı ve hata karelerin toplamı

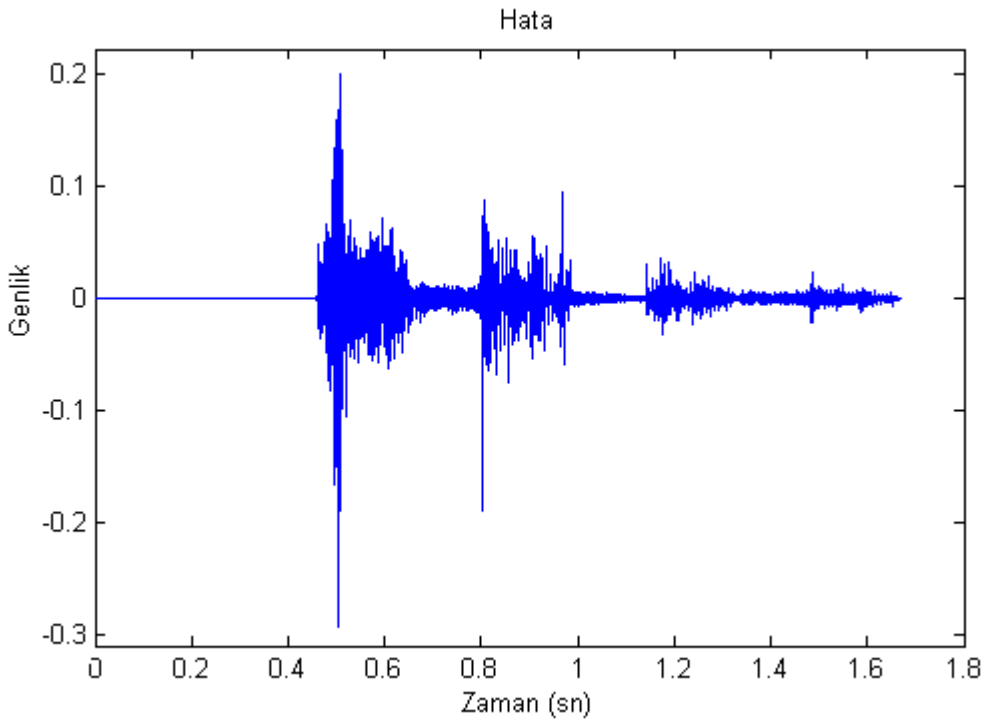
N=10	İlinti Katsayısı	Hata Karelerin Toplamı
$\mu=0.1$	0.9804	6.8446
$\mu=0.2$	0.9817	6.3881
$\mu=0.3$	0.9815	6.4699
$\mu=0.4$	0.9807	6.7652
$\mu=0.5$	0.9796	7.1754



Şekil 4.28. NLMS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)



Şekil 4.29. NLMS ile yankısı giderilmiş çok yankılı sinyal(yankılı ve yankısız sinyaller)



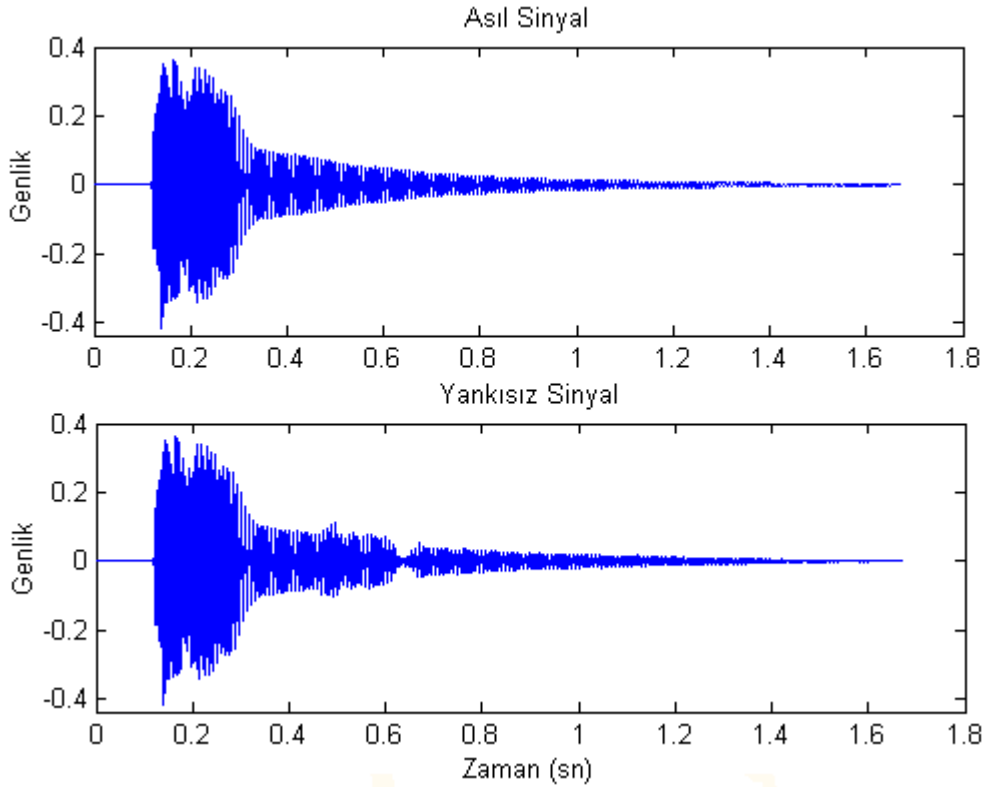
Şekil 4.30. Hata sinyali ($\mu_0=0.5$, $N=10$)

Çok yankılı sinyal, dengeleyiciden geçirilince Şekil 4.28., 4.29. ve 4.30' daki sonuçlar alınmıştır.

4.3.3. RLS Algoritması Kullanan Dengeleyici

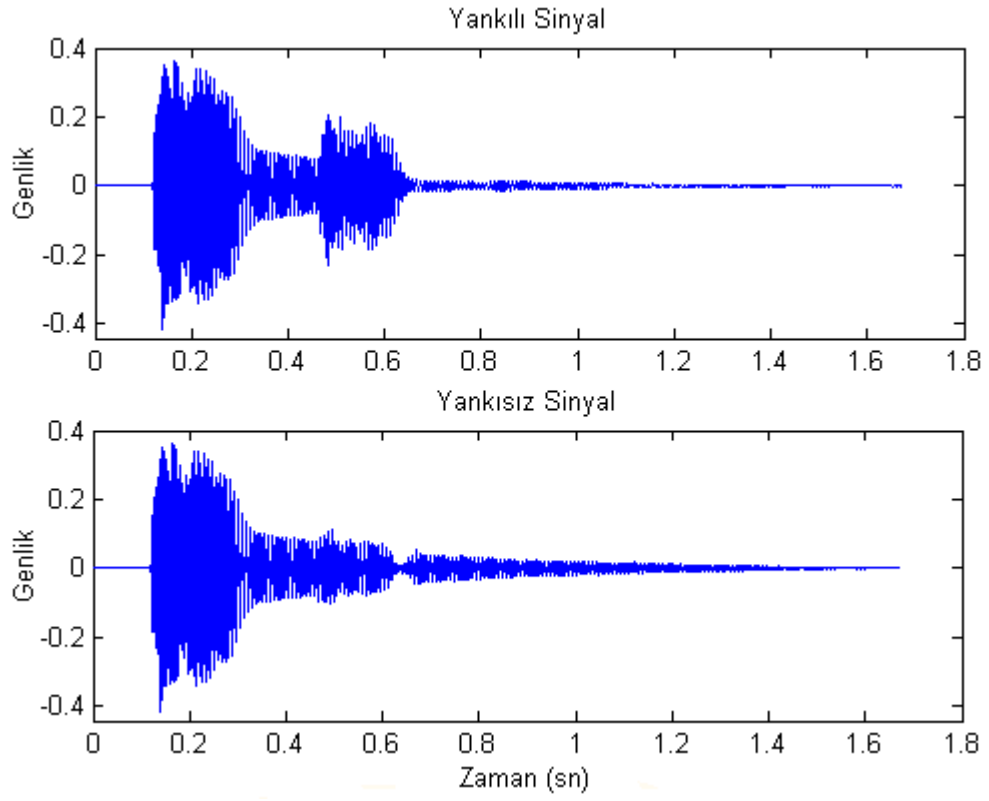
Uyarlama algoritması değiştirilerek RLS kullanılmıştır. Bunun için kullanılan program Ek A.'da dengeleyici_rls.m fonksiyonunu kullanan rls_dengele.m dosyasında verilmiştir.

```
>> [x,f]=wavread('deneme.wav'); % Asıl sinyal
>> y=yanki_FIR(x,0.5,15000); % Yankı ekle
>> yeni=dengeleyici_rls(y,x,10,0.5,0.99); %Yankı gider
```

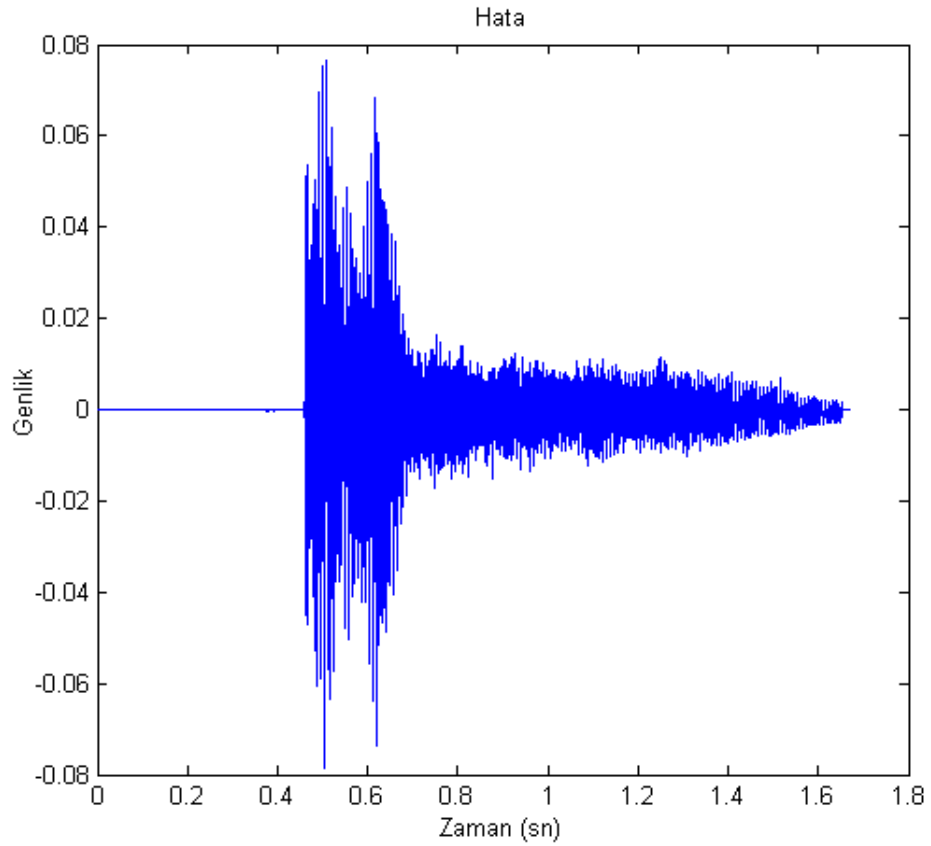


Şekil 4.31. RLS algoritması ile kanal dengeleme (N=10)

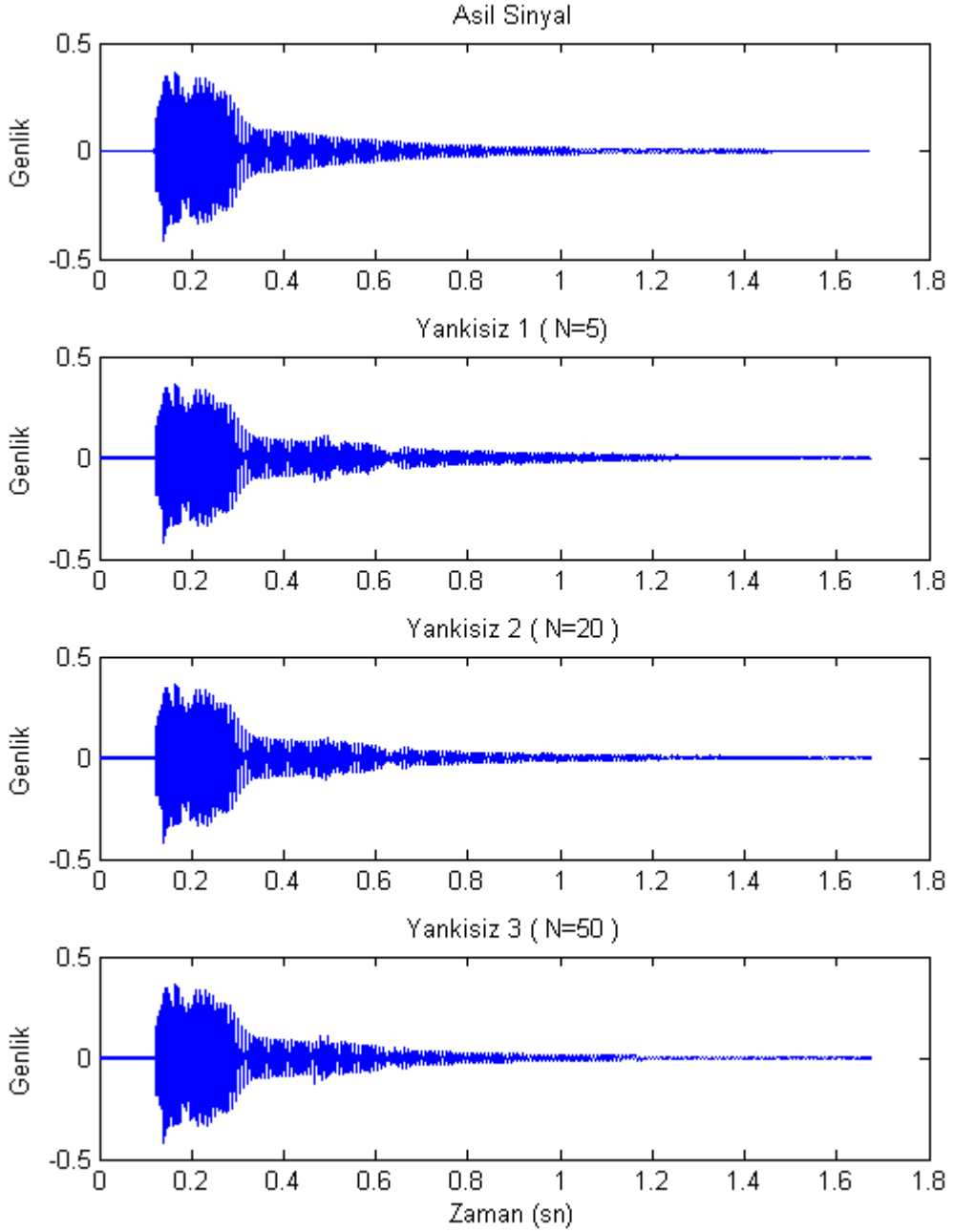
Şekil 4.33.'de yankısı giderilmiş sinyal ile asıl sinyal arasındaki fark sinyali verilmiştir.



Şekil 4.32. Yankılı ve yankısız sinyaller (N=10)



Şekil 4.33. Hata sinyali (N=10)

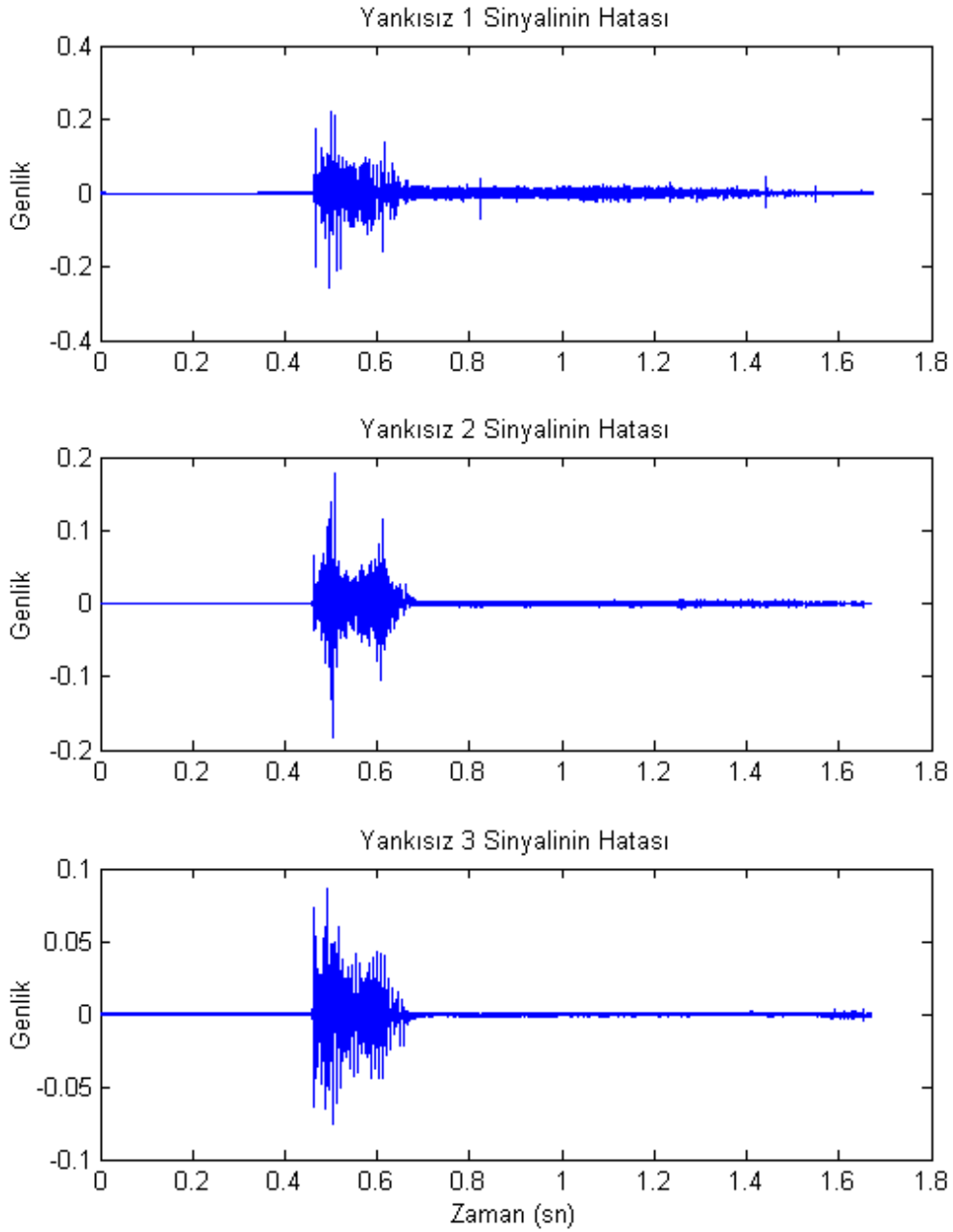


Şekil 4.34. Farklı süzgeç uzunlukları için çıkış sinyalleri

Farklı süzgeç uzunluğu N değerleri için program çalıştırılmış ve sonuçlar karşılaştırılmıştır. Şekil 4.34. 'de $N=5$, $N=20$, $N=50$ değerleri için çıkışlar verilmiştir. Tablo 4.5.' de farklı N değerleri için çıkışların asıl sinyal ile ilinti katsayısı ve hata sinyalinin kareleri toplamı verilmiştir. N arttıkça iyileşme görülmektedir.

Tablo 4.5: Farklı süzgeç uzunlukları için ilinti katsayısı ve hata karelerin toplamı

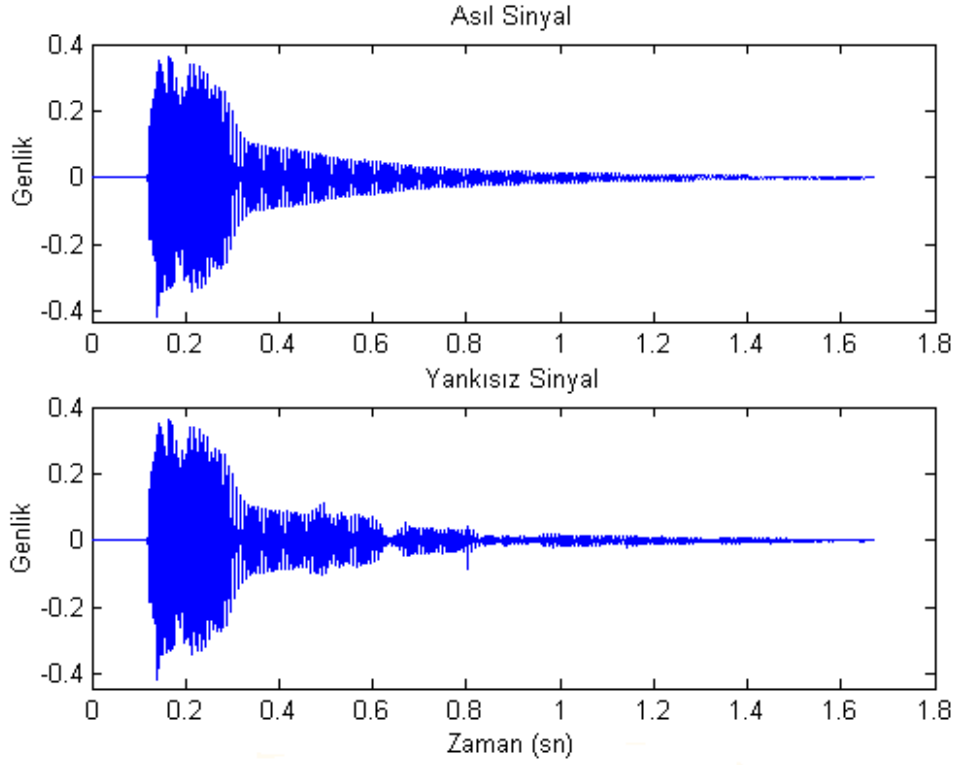
	İlinti Katsayısı	Hata Karelerin Toplamı
N=5	0.9809	6.6651
N=10	0.9843	5.4718
N=20	0.9864	4.7521
N=50	0.9859	4.9407
N=100	0.9871	4.5092



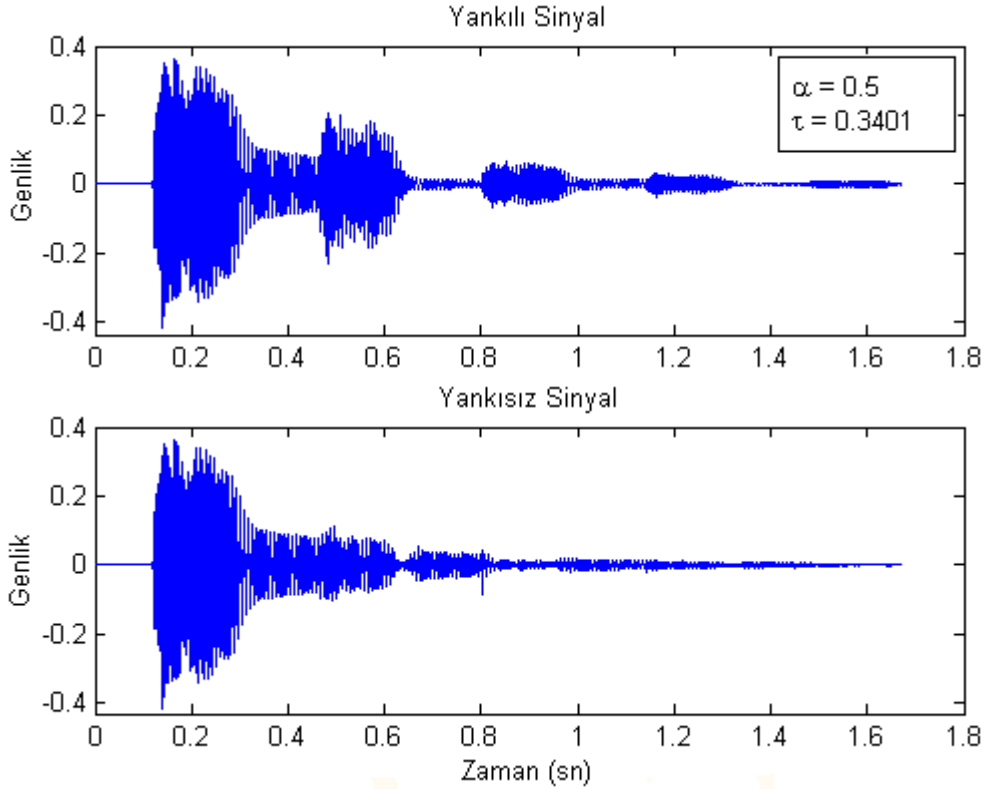
Şekil 4.35. Farklı N değerlerinde oluşan hata sinyalleri

Algoritma çok yankılı sinyallere de uygulanmıştır. Şekil 4.36., Şekil 4.37. ve Şekil

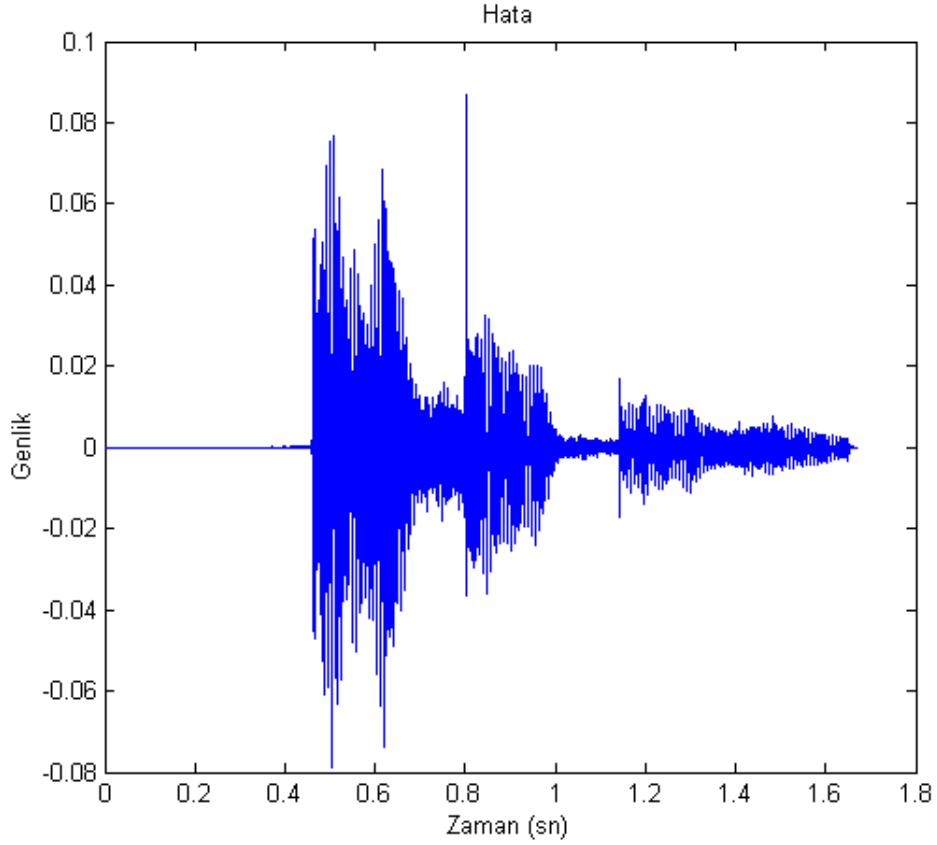
4.37.' de sonuçlar verilmiştir.



Şekil 4.36. RLS ile yankısı giderilmiş çok yankılı sinyal(Asıl ve yankısız sinyaller)



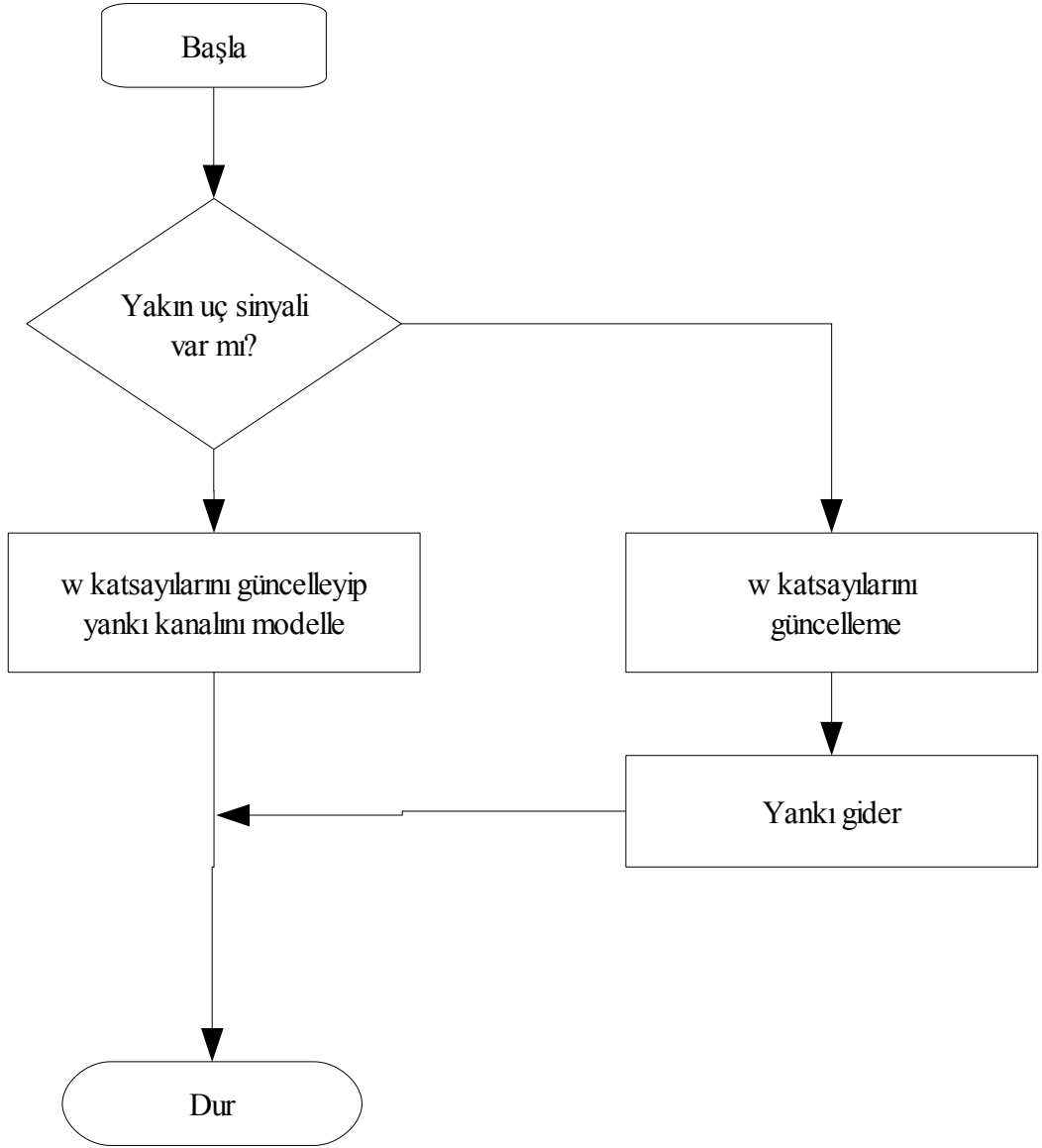
Şekil 4.37. RLS ile yankısı giderilmiş çok yankılı sinyal(yankılı ve yankısız sinyaller)



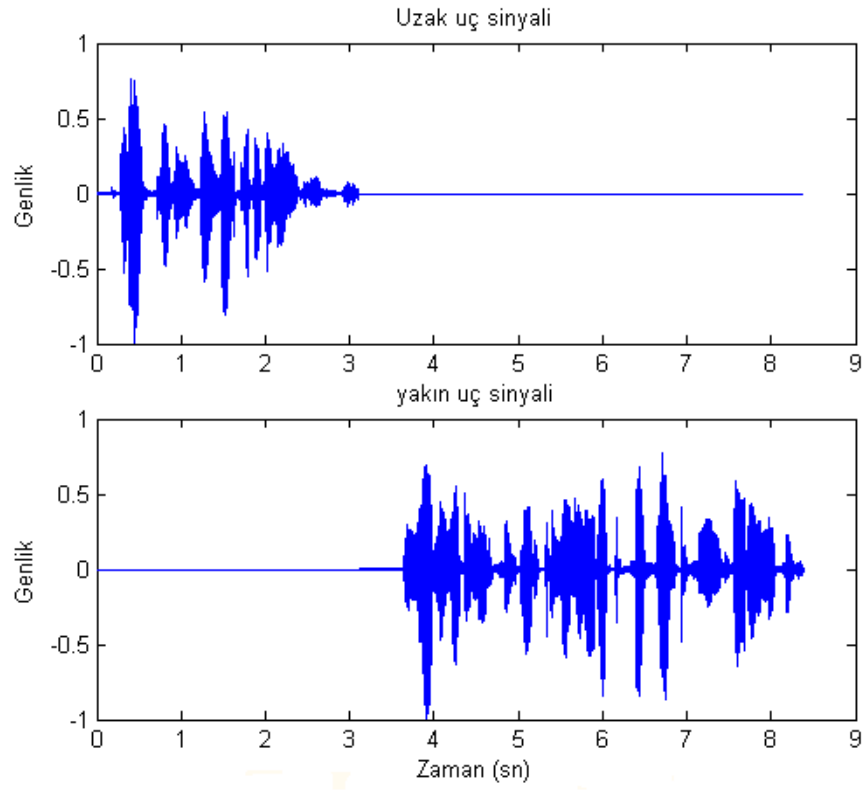
Şekil 4.38. Hata sinyali (N=10)

4.4. Ses Yankılarının Giderildiği Telekonferans Benzetimi

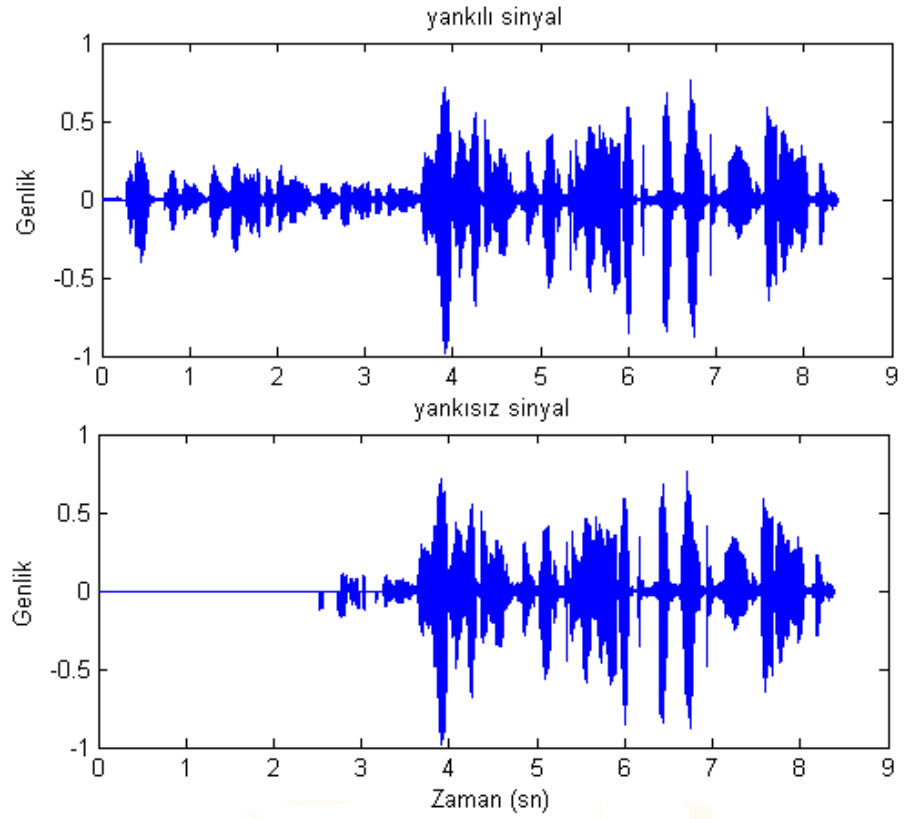
Şekil 3.32.'de verilen yapı kullanılarak uyarlanabilir süzgeç ile ses yankıları giderilmeye çalışılmıştır. Telekonferans uygulamasında tek taraf dikkate alınmıştır. Bu yapı Şekil 3.33.'de verilmiştir. Uzak uç konuşmacısı konuşmasını bitirdikten sonra oluşan yankı yakın uç konuşmacısının ses sinyaline eklenmiştir. Amaç yakın uç sinyalini bu yankıdan kurtarmaktır. Programın akış çizgesi Şekil 4.39.'da verilmiştir. Benzetimde, çoklu yankı olduğu varsayılmıştır.



Şekil 4.39. konferans.m fonksiyonunun akış çizgesi



Şekil 4.40. Uzak uç ve yakın uç sinyalleri



Şekil 4.41. Yankılı ve yankısız sinyaller

Şekil 4.40.' da yakın ve uzak uç sinyalleri, Şekil 4.41.' de yankılı ve yankısız sinyaller ve Şekil 4.42.' de hata sinyali verilmiştir. Yakın uç sinyali olarak “Biraz tezinizden bahsedebilir misiniz?” ve uzak uç sinyali olarak “Uyarlanabilir süzgeç kullanarak ses yankısının giderilmesi” şeklinde konuşma sinyalleri kullanılmıştır. Her iki sinyalin süreleri sıfır değerleri eklenerek eşitlenmiştir. Telekonferans sırasında, aynı anda çift yönlü konuşma olursa süzgeç katsayıları yanlış güncellenir. Bu nedenle uyarlanabilir yankı gidericilerde çift yönlü konuşma algılayıcılar kullanılır. Yakın uç sinyali olduğunda katsayılar güncellenmez. Eğer

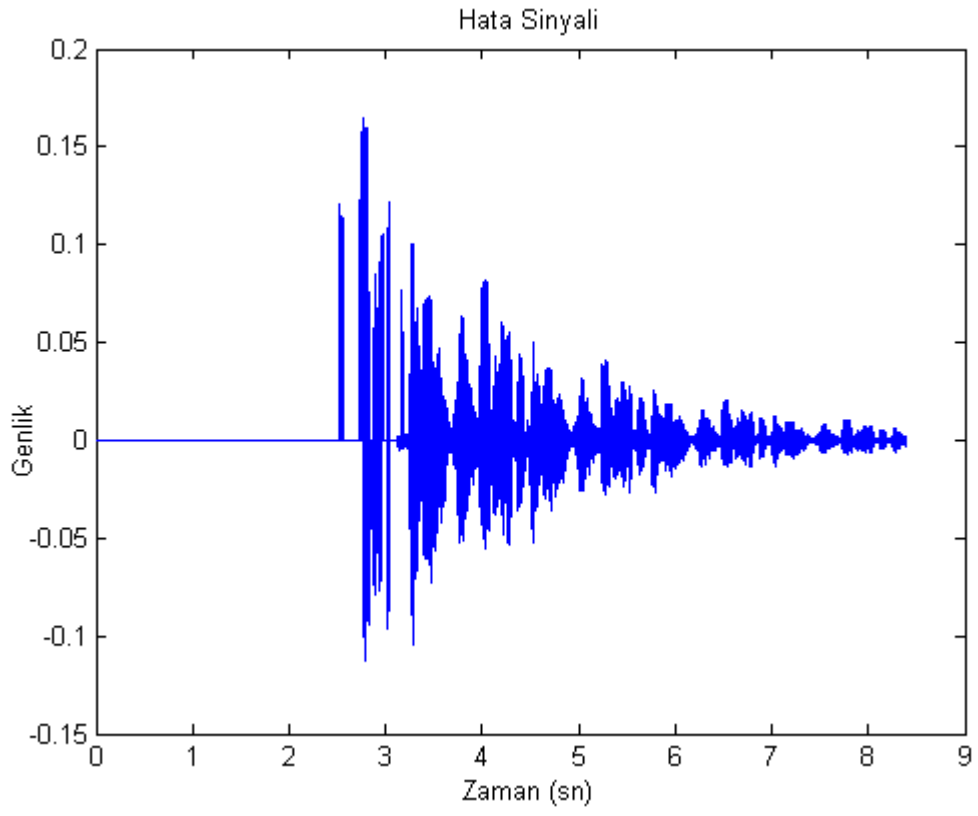
$$|d(n)| > \frac{1}{2} |x(n)| \quad (4.2)$$

ise yakın uç sinyali vardır ve güncelleme yapılmaz. Ancak uygulamada (4.2) eşitsizliği kullanılmaz. Yerine

$$|d(n)| > \frac{1}{2} \max\{|x(n)|, \dots, |x(n-N+1)|\} \quad (4.3)$$

eşitsizliği kullanılır. Yapılan denemelerde (4.3) eşitliğindeki 0.5 çarpanının iyi sonuçlar vermediği görülmüştür. Bu çarpan büyütüldüğünde daha iyi sonuçlar alınmıştır. Ancak bu değer çok büyük alınmamalıdır. Denemelerde çarpan 1.2-1.5 aralığında alındığında daha iyi sonuçlar elde edilmiştir.

Tele konferans benzetimi için Ek A.' da verilen konferans.m fonksiyonunu kullanan konferans_ana.m dosyası kullanılmıştır.



Şekil 4.42. Oluşan hata sinyali

5. SONUÇ VE ÖNERİLER

Bu çalışmada ses sinyalindeki yankılar giderilmeye çalışılmıştır. Gerçek zamanlı uygulamalarda yankı kanalı sürekli olarak zamanla değişmekte ve sabit katsayılı süzgeçler yankı gidermede yetersiz kalmaktadır. Bu nedenle uyarlanabilir süzgeçler tercih edilmektedir. Uyarlanabilir süzgeç yapısı kadar uyarlama algoritmasında çok önemlidir.

Bu çalışmada süzgeç yapısı olarak sadece doğrusal transversal süzgeç kullanılmıştır. Başka doğrusal veya doğrusal olmayan süzgeç yapıları kullanılarak da yankılar giderilmeye çalışılabilir. Ancak diğer süzgeç yapıları daha karmaşık yapıda olup hesaplama ve bellek yükünü arttırmaktadırlar. Kullanılan süzgecin derecesi arttıkça daha iyi sonuçlar alınır. Ancak bu durumda bellek ihtiyacı ve uyarlama süresi artmaktadır. Özellikle zaman aralıkları sabit olmayan çok yankılı sinyaller doğrusal yapıda olmayıp doğrusal süzgeçler ile tam olarak yankıdan arındırılamazlar. Yapay sinir ağları bu iş için uygundur. Örnek olarak telefon hatlarındaki yankıları gidermek için kullanılan MADALINE verilebilir.

Uyarlama (öğrenme) algoritmaları olarak LMS, NLMS, RLS kullanılmış ve en iyi sonuçlar RLS algoritması ile elde edilmiştir. RLS algoritması ile en az hata ve en büyük ilinti katsayısı alınmıştır. Ayrıca LMS algoritması gerçek zamanlı uygulamalar için uygun değildir. RLS algoritması yinelemeli (rekürsif) yapıda olduğundan gerçek-zamanlı uygulamalarda tercih edilmelidir. Kullanılan algoritmadaki adım boyutu gibi katsayılar da uyarlama süresi ve yakınsama miktarını etkilemektedir. Bu katsayıların seçimi de çok önemlidir. LMS algoritmasındaki adım boyutunun tespiti için henüz kesin bir çözüm yoktur.

Uyarlanabilir süzgeçler sadece anlık olarak uyarlama yapıp verinin tamamı için kullanılamaz. Özellikle kayıtlı veriler için verinin tamamına uygulanabilen algoritmalar vardır. Bu tür algoritmalara küme algoritmaları denir.

6. KAYNAKLAR

1. **Wiener , N. and E.Hopf, 1931.** On A Class Of Singular Integral Equations, Proc. Prussian Acad. Math-Phys. Ser. , 636
2. **Kolmogorov ,A.N. ,1939.** Sur L'interpolation Et Extropolation Des Suites Stationairs, C.R. Acad. Sci. ,Paris,vol,208, 2043-2045
3. **Krein,M.G. ,1945.** On a Problem Of Extrapolation Of A.N.KOLMOGOROV, C.R. (Dokl.) Akad. Nauk SSSR,vol. 46, 306-309
4. **Wiener, N. ,1949.** Extrapolation ,Interpolation And Smoothing Of Time Series, With Engineering Applications, MIT Press, Cambridge Mass.
5. **Widrow , B. , J. Mccool ,and M. Ball, 1975.** The Complex LMS Algorithm, Proc IEEE vol.63 ,719-720
6. **Lucky, R.W., 1965.** Automatic Equalisation For Digital Communication, Bell Syst. Tech. J. ,vol.44, 547-588
7. **Gersho, A., 1969.** Adaptive Equalisation Of Highly Dispersive Channels For Data Transmission, Bell Syst. Tech. J., vol.48 , 55-70
8. **Proakis ,J.G., and J.H. Miller , 1969.** An Adaptive Receiver For Digital Signaling Through Channels With Intersymbol Interference , IEEE Trans. Information Theory ,vol. IT-15, 484-497
9. http://www.isvr.soton.ac.uk/SPCG/Tutorial/Tutorial/Tutorial_files/Web-basics-nature.htm
10. <http://www.glenbrook.k12.il.us/gbssci/phys/Class/sound/u1111b.html>
11. **1998.** Basic Frequency Analysis of Sound,Lecture Note,English BA 7669-11, Brüel&Kjaer, 4
12. **1998.** Basic Concepts of Sound , Lecture Note, English BA 7666-11, Brüel&Kjaer, 8
13. **Ertürk, S. , 2005.** Sayısal Haberleşme, Birsen Yayınevi.
14. **Ertürk, S. , 2005.** Sayısal İşaret İşleme 2.Baskı, Birsen Yayınevi,156-168.
15. **Sunan, Ertuğrul, 2006.** Uyarlanabilir Süzgeçler Ve Uygulama Alanlarının İncelenmesi, Yüksek Lisans Semineri.Samsun Ondokuz Mayıs Üni.
16. **Seguine Dennis.** Adjustable Sallen and Key High-Pass Filters , Application Note.Cypress Mikrosystems.

<http://www.khalus.com.ua/psoc/pdf/appnotes/an2030.pdf>

17. **Birlikbaş, Sema, 2000.** Adaptif Dijital Filtre Yapıları , Öğrenme Algoritmaları Ve Uygulamaları ,Yüksek Lisans Tezi, Elazığ ,Fırat Üniversitesi.
18. **Haykin Simon, 2002.** Adaptive Filter Theory, Fourth Edition, Prentice Hall,7-12
19. **Scott,C., Nowak, R., 2004.** Introduction To Adaptive Filtering
<http://cnx.rice.edu/content/m11433/latest/>
20. **Moonen Marc .** An Introduction To Adaptive Signal Processing
http://homes.esat.kuleuven.be/~moonen/asp_course.html
21. **Jones,D. L.** Adaptive Interference (Noise) Cancellation
<http://cnx.rice.edu/content/m11835/latest/>
22. **1988.** Measurements In Building Acoustics, **Brüel&Kjaer** , 5
23. **Oral, Canan, 2004.** Radar Ve Dedectör Gibi Bozucu Sistemlerde Kök Momenti Kullanılarak Minimum Fazlılaştırma Ve Ters Süzgeç Uygulamaları, Yüksek Lisans Tezi , Samsun Ondokuz Mayıs Üni.
24. **Proakis, J. G.,Manolakis D. G. ,1989.** Introduction To Digital Signal Processing , Macmillan Publishing Company .
25. **Vaseghi S. V. 2006.** Advanced Digital Signal Processing and Noise Reduction , 3rd . Ed.,John Wiley & Sons Ltd,371-379
26. **2007.** Filter Design Toolbox 4, The Mathworks Inc.
27. **Cichocki, A., Amari, S., 2002.** Adaptive Blind Signal And Image Processing , Learning Algorithms And Applications, John Wiley & Sons Ltd.
28. **Ambardar A.,1995.** Analog And Digital Signal Processing, PWS Publishing Company ,170
29. **Bellanger M. G., 1987.** Adaptive Digital Filters And Signal Analysis,Marcel Dekker, Inc.,314-316

7. EKLER

EK A. Matlab ortamında yazılan fonksiyon ve betikler

```

%*****
%LMS.m
%*****
%LMS algoritması
%xn: Giriş
%dn: Arzulanan sinyal
%mu: (0,1) arası pozitif sabit
% N: Süzgeç katsayılarının sayısı
% w: Süzgeç katsayılarının başlangıç değeri

function [wn,en,yn]=LMS(xn,dn,mu,N,w);
if nargin<5
    w=zeros(N,1); end
tekrar=length(xn);
w=w(:);
xn=xn(:);
dn=dn(:);
for i=N:tekrar
    x=xn(i:-1:i-N+1);
    y(i)=w'*x;
    e(i)=dn(i)-y(i);
    w=w+mu*x*e(i);
end
wn=w;
en=e(:);
yn=y;

```

```

%*****
%NLMS.m
%*****
%NLMS algoritması

function [wn,en,yn]=NLMS(xn,dn,mu1,N,w);
if nargin<4
    w=zeros(N,1); end
tekrar=length(xn);
w=w(:);
xn=xn(:);
dn=dn(:);
mu=mu1;
for i=N:tekrar
    x=xn(i:-1:i-N+1);
    y(i)=w'*x;
    e(i)=dn(i)-y(i);
    w=w+mu*x*e(i);
    n=(norm(x))^2;
    if n==0
        mu=mu1;
    else
        mu=mu1/n;
    end
end
end
wn=w;
en=e(:);
yn=y(:);

```

```

%*****
%RLS.m
%*****
% RLS algoritması
%lamda: 1'den küçük ama 1'e çok yakın sabit
%delta:yüksek SNR için küçük pozitif sabit
%      :düşük SNR için büyük pozitif sabit

function [wn,hata,yeni]=RLS2(xn,dn,N,delta,lamda,w);
if nargin<6
    w=zeros(N,1); end
p=(1/delta)*eye(N);
tekrar=length(xn);
w=w(:);
xn=xn(:);
dn=dn(:);
for i=N:tekrar
    x=xn(i:-1:i-N+1);
    pp=p*x;
    k=pp/(lamda+x'*pp);
    y(i)=w'*x;
    ksi(i)=dn(i)-y(i);
    w=w+k*ksi(i);
    p=(1/lamda)* p-(1/lamda)* k* x' * p;
end
wn=w;
hata=ksi(:);
yeni=y(:);

```

```

%*****
% dengele_FIR.m
%*****
% IIR-FIR dengeleyici

function yankisiz=dengele_FIR(x,alfa,k);
a=[1];
b=[1 zeros(1,k-1) -alfa];
yankisiz=filter(b,a,x);
yankisiz=yankisiz';

%*****
% denkleFIR.m
%*****
% FIR-FIR dengeleyici

function yden=denkleFIR(yank,ybag,ornek);%yden=denkleFIR(yank,ybag,tg);
yson=yank;
y5=[zeros(1,5*ornek) yson'];
y11=[zeros(1,ornek) yson'];
y2=[zeros(1,2*ornek) yson'];
y3=[zeros(1,3*ornek) yson'] ;
y4=[zeros(1,4*ornek) yson'];
ny=length(yson);
yden=(yson(1:ny))'-ybag*y11(1:ny)+(ybag)^2*y2(1:ny)-
(ybag)^3*y3(1:ny)+(ybag)^4*y4(1:ny)-(ybag)^5*y5(1:ny);

```

```

%*****
% denkleIIR.m
%*****
%FIR-IIR dengeleyici
%girişler; yank: yankılı ses sinyali;
%ybag: yankı bağıl genliği,
%çıkış: yeni: denkleştirici çıkışı

function yeni=denkleIIR(yank,f,ybag,ornek);
yson=yank;
yeni=filter(1,[1 zeros(1,ornek-1) ybag ],yson);

%*****
%dengeleyici_lms.m
%*****

function [yeni,wn]=dengeleyici_lms(xn,dn,mu,N,w);
if nargin<5
    w=zeros(N,1); end
dn=dn(:)';
[wn,e,yeni]=LMS(xn,dn,mu,N,w);

%*****
% dengeleyici_nlms.m
%*****

function [yeni,wn]=dengeleyici_nlms(xn,dn,mu1,N,w);
if nargin<5
    w=zeros(N,1); end
dn=dn(:)';
[wn,e,yeni]=NLMS(xn,dn,mu1,N,w);

```

```

%*****
% dengeleyici_rls.m
%*****
% RLS kullanan uyarlanabilir dengeleyici
%lamda: 1'den küçük ama 1'e çok yakın sabit
%delta:yüksek SNR için küçük pozitif sabit
%      :düşük SNR için büyük pozitif sabit

function [yeni,wn]=dengeleyici_rls(xn,dn,N,delta,lamda,w);
if nargin<6
    w=zeros(N,1); end
dn=dn(:)';
[wn,e,yeni]=RLS(xn,dn,N,delta,lamda,w);

%*****
%konferans.m
%*****

function [wh,eh,yeni]=konferans(xn,dn,mu,N);
tekrar=min(length(xn),length(dn));
w=zeros(N,1);
xn=xn(:);
dn=dn(:);
for i=N:tekrar
    x=xn(i:-1:i-N+1);
    y(i)=w'*x;
    e(i)=dn(i)-y(i);
    if abs(dn(i))>1/2*max(x);
        continue
    else
        w=w+mu*x*e(i);
    end
end

```

```

end
e=e(:);
if length(xn)<length(dn)
    eh=[e(N:tekrar)' dn(tekrar+1:length(dn))'];
else
    eh=[e(N:tekrar)' xn(tekrar+1:length(xn))'];
end
wh=w;

%*****
%cizdir.m
%*****
% Grafik çizimi

function cizdir(x,y,yankisiz,f)
x=x(:);
y=y(:);
t=linspace(0,length(x)/f,length(x));
yankisiz=yankisiz(:);
pos = get(gcf,'Position');
set(gcf,'Position',[pos(1), pos(2)-100,pos(3),(pos(4)+85)])
subplot(211)
dugme1=uicontrol('Style','pushbutton','Units','centimeters','Position',[4 0.2 2 0.7 ],
'String','Grafik1','Callback','wavplay(y,f)');
plot(t,y)
title('Yankili Sinyal')
ylabel('Genlik')
subplot(212)
plot(t,yankisiz)
title('Yankisiz Sinyal')
ylabel('Genlik')
xlabel('Zaman (sn)')

```

```

dugme2=uicontrol('Style','pushbutton','Units','centimeters','Position',[9 0.2 2
0.7],'String','Grafik2','Callback','wavplay(yankisiz,f)');
figure
pos2 = get(gcf,'Position');
set(gcf,'Position',[pos2(1), pos2(2)-100,pos2(3),(pos2(4)+85)])
subplot(211)
dugme3=uicontrol('Style','pushbutton','Units','centimeters','Position',[4 0.2 2 0.7 ],
'String','Grafik1','Callback','wavplay(x,f)');
plot(t,x)
title('Asil Sinyal')
ylabel('Genlik')
subplot(212)
dugme4=uicontrol('Style','pushbutton','Units','centimeters','Position',[9 0.2 2 0.7 ],
'String','Grafik2','Callback','wavplay(yankisiz,f)');
plot(t,yankisiz)
title('Yankisiz Sinyal')
ylabel('Genlik')
xlabel('Zaman (sn)')
figure
pos3 = get(gcf,'Position');
set(gcf,'Position',[pos3(1), pos3(2)-100,pos3(3),(pos3(4)+60)])
plot(t,x-yankisiz)
title('Hata')
xlabel('Zaman (sn)')
ylabel('Genlik')
%axis([0,length(x),-0.001,0.001])

```

```
%*****
```

```
%FIR_IIR.m
```

```
%*****
```

```
%FIR-IIR dengeleme
```

```
close all;
```

```
clear all;
```

```
clc
```

```
alfa=0.4;
```

```
k=15000;
```

```
[x,f]=wavread('echo-off.wav');
```

```
y=yanki_FIR(x,alfa,k);
```

```
yankisiz=denkleIIR(y,alfa,k);
```

```
cizdir(x,y,yankisiz,f)
```

```
%*****
```

```
%FIR_FIR.m
```

```
%*****
```

```
%FIR-FIR dengeleme
```

```
close all;
```

```
clear all;
```

```
clc
```

```
alfa=0.4;
```

```
k=15000;
```

```
[x,f]=wavread('echo-off.wav');
```

```
y=yanki_FIR(x,alfa,k);
```

```
yankisiz=denkleFIR(y,alfa,k);
```

```
cizdir(x,y,yankisiz,f)
```

```

%*****
%IIR_FIR.m
%*****

%IIR-FIR dengeleme

close all;
clear all;
clc
alfa=0.4;
k=15000;
[x,f]=wavread('echo-off.wav');
y=yanki_IIR(x,alfa,k);
yankisiz=dengele_FIR(y,alfa,k);
cizdir(x,y,yankisiz,f)

%*****
%lms_dengele.m
%*****

%LMS uyarlanabilir dengeleme

close all;
clear all;
clc
alfa=0.5;
k=15000;
mu=0.5;
N=10;
[x,f]=wavread('echo-off.wav');
y=yanki_IIR(x,alfa,k);
yankisiz=dengeleyici_lms(y,x,mu,N);
cizdir(x,y,yankisiz,f)

```

```

%*****
%nlms_dengele .m
%*****

%NLMS uyarlanabilir dengeleme

close all;
clear all;
clc
alfa=0.4;
k=15000;
mu=0.5;
N=10;
[x,f]=wavread('echo-off.wav');
y=yanki_IIR(x,alfa,k);
yankisiz=dengeleyici_nlms(y,x,mu,N);
cizdir(x,y,yankisiz,f)

%*****
%rls_dengele .m
%*****

%RLS uyarlanabilir dengeleme

close all;
clear all;
clc
alfa=0.4;
k=15000;
N=10;
delta=0.1;
lamda=0.99;
[x,f]=wavread('echo-off.wav');
y=yanki_IIR(x,alfa,k);

```

```

yankisiz=dengeleyici_rls(y,x,N,delta,lamda);
cizdir(x,y,yankisiz,f)

%*****
%konferans_ana.m
%*****
%Telekonferans benzetimi

close all;
clear all;
clc
[x,f]=wavread('uzakuc.wav');
[p,f]=wavread('yakinuc.wav');
[w,e,yankili,xn,d]=konferans(x,p,0.025,1000,8000);

t=linspace(0,length(xn)/f,length(xn));

pos = get(gcf,'Position');
set(gcf,'Position',[pos(1), pos(2)-100,pos(3),(pos(4)+85)])
subplot(211)
dugme1=uicontrol('Style','pushbutton','Units','centimeters','Position',[4 0.2 2 0.7 ],
'String','Grafik1','Callback','wavplay(xn,f)');
plot(t,xn)
title('Uzak uç sinyali')
subplot(212)
plot(t,d)
title('yakin uç sinyali')
dugme2=uicontrol('Style','pushbutton','Units','centimeters','Position',[9 0.2 2
0.7],'String','Grafik2','Callback','wavplay(d,f)');

figure
pos2 = get(gcf,'Position');

```

```
set(gcf,'Position',[pos2(1), pos2(2)-100,pos2(3),(pos2(4)+85)])
subplot(211)
dugme3=uicontrol('Style','pushbutton','Units','centimeters','Position',[4 0.2 2 0.7 ],
'String','Grafik1','Callback','wavplay(yankili,f)');
plot(t,yankili)
title('yankili sinyal')
subplot(212)
dugme4=uicontrol('Style','pushbutton','Units','centimeters','Position',[9 0.2 2 0.7 ],
'String','Grafik2','Callback','wavplay(e,f)');
plot(t,e)
title('yankisiz sinyal')

figure
plot(t,d-e)
title('Hata Sinyali')
```

ÖZGEÇMİŞ

1978 yılında Erzincan'da doğdu. İlk ve orta öğrenimini Erzincan'da tamamladı. 1997 yılında lisans öğrenimine Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği bölümünde başladı. 2002 yılında mezun oldu. 2004 yılında yüksek lisans öğrenimine başladı. Şu anda Ondokuz Mayıs Üniversitesi Elektrik-Elektronik Mühendisliği bölümünde Araştırma Görevlisi olarak çalışmaktadır.