



T.C.  
NECMETTİN ERBAKAN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ



DİJİTAL SES EFEKTLERİNİN YAZILIMSAL  
OLARAK OLUŞTURULMASI

Enes SULAĞ

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Kasım-2018  
KONYA  
Her Hakkı Saklıdır

## TEZ KABUL VE ONAYI

Enes SULAK tarafından hazırlanan “DİJİTAL SES EFEKTLERİNİN YAZILIMSAL OLARAK OLUŞTURULMASI” adlı tez çalışması 06/11/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

### Jüri Üyeleri

### İmza

#### Danışman

Prof. Dr. Sabri KOÇER

.....

#### Üye

Dr. Öğr. Üyesi Muhammed Fahri ÜNLERŞEN

.....

#### Üye

Dr. Öğr. Üyesi Hüseyin DOĞAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Ahmet AVCI  
FBE Müdürü

Bu tez çalışması Necmettin Erbakan Üniversitesi Bap Koordinatörlüğü tarafından 161319010 nolu proje ile desteklenmiştir.

## **TEZ BİLDİRİMİ**

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## **DECLARATION PAGE**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Enes Sulak  
Tarih: 06.11.2018

# ÖZET

## YÜKSEK LİSANS TEZİ

### DİJİTAL SES EFEKTLERİNİN YAZILIMSAL OLARAK OLUŞTURULMASI

**Enes SULAK**

**Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı**

**Danışman: Prof. Dr. Sabri KOÇER**

**2018, 66 Sayfa**

**Jüri**

**Prof. Dr. Sabri KOÇER**

**Dr. Öğr. Üyesi Muhammed Fahri ÜNLERŞEN**

**Dr. Öğr. Üyesi Hüseyin DOĞAN**

Bu çalışmada, dijital ses efektleri incelenmektedir. Çoğu müzisyen dijital ses efektlerini kullanmaktadır. Bu efektler müzisyenlerin istedikleri tonları ve sesleri elde etmelerini sağlamaktadır. Dijital Ses Çalışma Ortamları ve Sanal Stüdyo Teknolojileri ses efektleri yazılımlarının geliştirilmesi ve yaygınlaşmasını sağlamışlardır. Bu tezin amacı genel doğrusal gitar efektlerinden olan gecikme, yankı, wah-wah ve flanger gibi efektlerinin gerçek zamanlı olarak modellenmesi ve algoritmalar kullanarak geliştirilmesidir.

Bu tez, ses sinyalinin girmek istediği efektleri teori, tasarım ve uygulamasına odaklanmaktadır. Ses mühendislerinin, önceden tanımlanmış blokların veya parametrelerin değerlerini değiştirerek istenilen ses efektlerini kolayca yapılandırmasına yardımcı olmaktadır. Parametreler, istenen efektler için transfer fonksiyonunu şekillendirir ve kontrol eder. Matematiksel formülasyon yapmak yerine, parametreler deneme ve yanılma ile ayarlanabilir ve ince ayar ile mükemmel ses efektleri gözlemlenmiştir.

**Anahtar Kelimeler:** Flanger, Gecikme, Dijital Ses Efektleri, Sinyal İşleme, Yankı, Wah-wah

**ABSTRACT**

**MS THESIS**

**CREATING DIGITAL SOUND EFFECTS IN SOFTWARE**

**Enes SULAK**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF  
NECMETTİN ERBAKAN UNIVERSITY  
THE DEGREE OF MASTER OF SCIENCE / DOCTOR OF PHILOSOPHY  
IN INDUSTRIAL ENGINEERING**

**Advisor: Prof. Dr. Sabri KOÇER**

**2018, 66 Pages**

**Jury**

**Advisor: Prof. Dr. Sabri KOÇER**

**Assist. Prof. Muhammed Fahri ÜNLERŞEN**

**Assist. Prof. Hüseyin DOĞAN**

In this study, digital sound effects are examined. Most musicians use digital sound effects. These effects enable musicians to get the tones and sounds they want. Digital Audio Working Environments and Virtual Studio Technologies have enabled the development and expansion of sound effects software. The aim of this thesis is to develop the effects of general linear guitar effects such as delay, echo, wah-wah and flanger in real time using algorithms.

This thesis focuses on the theory, design and application of the effects of the audio signal. It helps sound engineers to easily configure desired sound effects by changing the values of predefined blocks or parameters. The parameters shape and control the transfer function for the desired effects. Rather than mathematical formulation, parameters can be adjusted by trial and error, and fine sound effects have been observed with fine tuning.

**Keywords:** Flanger, Delay, Digital Sound Effects, Signal Processing, Echo, Wah-wah

## ÖNSÖZ

Bu çalışmada, bilgisayar ortamında dijital ses efektleri incelenmiş ve doğrulanmıştır. Bilgisayar ortamında bazı dijital ses efektlerinin temel dalga formları ve temel filtrelerin nasıl çalıştığı gösterilmiştir. Dijital ses efektleri, bilgisayar ortamlarında kodlanarak ve tasarlanarak incelenmiştir.

Tez çalışmamda planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Prof. Dr. Sabri KOÇER'e sonsuz teşekkürlerimi sunarım.

Enes SULAK  
KONYA-2018



# İÇİNDEKİLER

<b>ÖZET .....</b>	<b>iv</b>
<b>ABSTRACT.....</b>	<b>v</b>
<b>ÖNSÖZ .....</b>	<b>vi</b>
<b>İÇİNDEKİLER .....</b>	<b>vii</b>
<b>KISALTMALAR .....</b>	<b>ix</b>
<b>1. GİRİŞ .....</b>	<b>1</b>
<b>2. KAYNAK ARAŞTIRMASI .....</b>	<b>3</b>
2.1. İlgili Çalışmalar .....	3
<b>3. DİJİTAL SES İŞLEME VE SES EFEKTLERİ .....</b>	<b>5</b>
3.1. Ses Efektleri ve Tarihsel Gelişimi .....	5
3.1.1. Analog Gecikme .....	6
3.1.2. Erken Gecikme Sistemleri .....	6
3.1.3. Dijital Gecikme .....	7
3.1.4. Yazılım Gecikme .....	7
3.2. Dijital Ses İşleme .....	8
3.2.1. Dijital Ses Teorisi .....	8
3.2.2. Temel Ses Efektlerinin Yapı Taşları .....	10
3.3. Doğrusal Ses Efektleri.....	11
3.3.1. FIR Gecikmesi .....	11
3.3.2. Flanger .....	12
3.3.3. Yankılanma .....	13
3.3.3.1. Yankılanma Ortam Modellemesi .....	14
3.3.4. Dijital All-Pass Filtresi (Dijital Tüm-Geçiren Filtre) .....	15
3.3.5. Yapay Yankı .....	18
3.3.5.1 Renksiz Yapay Yankılanma.....	19
3.3.6. Erken Yankılanma .....	21
3.3.7. Geç Yankılanma .....	21
3.4. Wah-Wah Efekt .....	22
<b>4. ARAŞTIRMA BULGULARI VE TARTIŞMA .....</b>	<b>24</b>
4.1. MATLAB ile Ses Efekt İşleme .....	24
4.1.1. Temel Dijital Ses Filtreleme Efektleri .....	24
4.1.2. Ekolayzır Filtreleri (Shelving ve Peak Filtreleri) .....	24
4.1.2. Zamanla Değişen Filtreler.....	28
4.1.3. Gecikme Tabanlı Efektler .....	29
4.1.3.1. Vibrato .....	29
4.1.3.2 Comb Filtresi Gecikmeli Efektler.....	30

4.2.	MATLAB ile Doğrusal Olmayan Ses İşleme .....	31
4.2.1.	Sınırlayıcı (Limiter) .....	31
4.2.2.	Kompresörler ve Geniştiriciler (Compressor ve Expander ) .....	32
4.3.	Yankılanma ve Mekansal Efektler .....	33
4.3.1.	Yatay Kaydırma (Panning) .....	33
4.3.2.	Schroeder Yankılayıcı .....	34
4.3.3.	Konvolüsyon Yankılanma .....	35
4.4.	MATLAB Simulink İle Gerçek Zamanlı Ses İşleme .....	37
4.4.1.	Gecikme (Delay) .....	37
4.4.2.	Eko Efekt .....	38
4.4.3.	Flanger Efekt .....	41
4.4.4.	Koro Efekt .....	43
4.4.5.	Reverb (Yankı) Efekt .....	46
<b>5.</b>	<b>SONUÇ VE ÖNERİLER.....</b>	<b>49</b>
5.1.	Sonuçlar .....	49
5.2.	Öneriler .....	50
<b>KAYNAKLAR .....</b>	<b>51</b>	
<b>EKLER .....</b>	<b>53</b>	
<b>ÖZGEÇMİŞ .....</b>	<b>66</b>	



## KISALTMALAR

### Kısaltmalar

ASIC	: Uygulamaya Özel Tümüleşik Devre (Application Specific Integrated Circuit)
DAFx	: Dijital Ses Efektleri (Digital Audio Effects)
DSP	: Dijital Sinyal İşleme, Dijital Sinyal İşlemcisi (Digital Sinyal Processing, Digital Signal Processor)
FFT	: Hızlı Fourier Dönüşümü (Fast Fourier Transform)
FIR	: Sonlu Darbe Yanıtı (Finite Impulse Response)
FPGA	: Alan Programlanabilir Kapı Dizisi (Field Programmable Gate Array)
IIR	: Sonsuz Darbe Yanıtı (Inifinite Impulse Response)
LFO	: Düşük Frekanslı Osilatör (Low Frequency Oscillators)
LTI	: Doğrusal Zamanla Değişmeyen (Linear Time-Invariant)
RAM	: Rastgele Erişimli Hafıza (Random Access Memory)
PCM	: Darbe Kod Modülasyonu (Pulse Code Modulation)

## 1. GİRİŞ

Elektro gitar, metal tellerin titreşimini elektrik sinyaline dönüştürme yeteneğiyle birçok analog ve dijital efekt için ev sahipliği yapmaktadır. Gitar efektleri dünyası ikiye ayrılır: analog efektler ve dijital efektler.

Analog efektler, yüksek doğruluk, kalite ve organik sesler sağlarken, dijital efektler tek bir kutuda veya yazılımda hareketlilik, nicelik ve efekt paketi sağlar. Analog sinyal işleme, bu iki tip arasında bir öncüdür. Dijital işlemenin çeşitli dezavantajları olsa da, hala tüm gerçek sinyallerin doğada analog olduğu bir gerçektir. Fakat günümüzde gelişen teknoloji sayesinde dijital sinyal işleme uygulamaları yaygınlaşmış ve dijital sinyal işlemeye olan önem artmıştır. Analog işlemenin karmaşık ve zor olması dijital işlemeye olan ilgiyi daha da arttırmaktadır. Dijital sinyale müdahale etmek analog sinyale göre daha kolaydır ve bu bize kolaylıkla ses üzerinde işlem yapmamıza olanak sağlamaktadır. Bununla birlikte, dijital sinyal işleme, bileşenler olarak ADC ve DAC'ye ihtiyacımız vardır. ADC, gerçek dünya analog sinyalini dijitale dönüştürmek için kullanılır ve DAC, gerçek dünyaya sunmak için işlenmiş dijital sinyali analog forma dönüştürmek için kullanılır. Ayrık zaman, ayrık frekans veya diğer ayrık alan sinyalleri gibi herhangi bir analog sinyal türü, dijital işlemeye izin vermek için bir dizi sıra formunda dijital sinyaller olarak uygun şekilde temsil edilirler.

Dijital sinyal işleme, mikro işlemcilerin sinyal işlemeye yönelik hesaplama gücünü oluşturur ve böylece analog işleme göre birçok avantaj sağlar. Bir elektrogitar çaların kullandığı en yaygın efektler: Gecikme, Yankı, Wah-Wah ve Flanger'dır. Bunlar, diğer özelleştirilmiş efektleri türetilen ana efektlerdir.

Dijital sinyal işleme, girişin ses sinyali, video sinyali, görüntü, multimedya veya başka bir şey olup olmadığına bakmadan sayıların giriş sırasını işler. Dijital sinyal işleme, sinyallerin sayısal halidir. Ölçüm, filtreleme, yükseltme, zayıflama, gecikme, sıkıştırma veya sinyal oluşturma gibi sinyal üzerinde çeşitli işlemler gerçekleştirilebilir.

Bu çalışma, lineer gitar ses efektleri ve doğrusal olmayan gitar ses efektlerinden gecikme, yankı, wah-wah, flanger gibi ses efektlerinin gerçek zamanlı uygulamaları üzerinedir. Kişisel bilgi işlem ve dijital sinyal işleme biçimlerindeki geliştirmelerle ortaya çıkan, dijital efektlerin, analog efektlere göre başarılı bir şekilde taklit etme, simüle etme ve sentezlemeye ihtiyacı vardır.

Bu tez çalışmasının ikinci bölümünde kaynak araştırması ve ilgili çalışmalardan bahsedilmektedir. Üçüncü bölümünde ses efektleri ve tarihsel gelişimi, dijital ses işleme, ses efektleri hakkında bilgiler vermektedir. Tezin dördüncü bölümünde MATLAB ortamındaki efekt çıktıları gösterilmekte ve MATLAB ortamında gerçek zamanlı ses işleme örnekleri üzerinde durmaktadır. Son bölüm olan beşinci bölümde ise sonuç kısmı ve öneriler bulunmaktadır.



## 2. KAYNAK ARAŞTIRMASI

### 2.1. İlgili Çalışmalar

Dijital sinyal işleme, ses ve konuşma işleme, sonar ve radar sinyal işleme, anten ışın oluşturma, spektral kestirim, istatistiksel sinyal işleme, dijital görüntü işleme, iletişim, kontrol, veri sıkıştırma, finansal sinyal işleme, biyomedikal, sismik veri işleme gibi hızla büyüyen uygulama alanlarındaki temel teknolojilerden biridir. Dijital sinyal işleme uygulamalarının belirli tüketici sınıfı örnekleri şunlardır; konuşma sıkıştırma ve iletimi için cep telefonları ve ses efektleri üretimi için Hi-fi ses sistemi, Ses Takviye Sistemi, MRI gibi Tıbbi görüntüleme ve Bilgisayar grafikleri vb.

Bir çeşit dijital sinyal işleme gerçekleştiren ürünlerin sayısı ve çeşitliliği konusunda artan bir eğilim vardır. Dijital sinyal işleme ilaç, iletişim, endüstri veya tüketici piyasası gibi çeşitli alanlarda kullanılan birçok üründe kilit bileşeni oluşturmaktadır.

Günümüzde dijital sinyal işleme için kullanılan, alan programlanabilir kapı dizileri, güçlü genel amaçlı mikroişlemciler ve dijital sinyal denetleyicileri gibi diğer teknolojiler ortaya çıkmaktadır. Dijital sinyal denetleyicileri çoğunlukla motor kontrol gibi endüstriyel kontrol uygulamaları için kullanılmaktadır. Genellikle gerçek zamanlı olmayan uygulamalar sinyal işleme görevlerini gerçekleştirmek için genel amaçlı bilgisayarları kullanırlar.

Yüksek kaliteli elektrogitar ses ekipmanları pahalıdır ve insanlar bir pakette yüksek kaliteli efektler elde etmek için daha ucuz ve daha etkili çözümler aramaktadırlar. Sonuç olarak, yazılım üzerinden dijital efektlerinin modellenmesi ortaya çıkmıştır. Konuyla ilgili teknik araştırma makaleleri ve kitaplar; Udo Zölzer tarafından DAFx, dijital ses efektleri üzerine araştırma makaleleri içermektedir (Zölzer, 2011). Bu makaleler ve ses efektlerinin açıklamaları, uygulamalar için uygun için başlangıç referansıdır.

Bu alanda literatür taraması yaptığımızda yukarıda da belirttiğimiz gibi DAFx dijital ses efektleri üzerinde çalışmak için en önemli kaynaktır. Bununla birlikte Udo Zölzer'in DAFX ( Digital Audio Effects) isimli 2002 yılında basılan kitabı değerli bir kaynaktır.

Bu alanda yapılan çalışmalar: Aalborg Üniversitesi, Elektronik Sistemler Bölümü tarafından hazırlanan “Akustik Gitar için Çok Efektli İşlemci” (Multi-Effect Processor for Acoustic Guitar) bu alanda bize yol gösteren önemli çalışmalardan birisi olmuştur.

Orta Doğu Teknik Üniversitesi’nden Zeki ENGİN tarafından hazırlanan Digital Modelling Of Guitar Audio Effects (Gitar Ses Etkilerinin Dijital Modellenmesi) tezi bu alanda yapılmış en önemli çalışmalardan birisidir. Bu çalışmada gitar ses efektleri dijital olarak modellenmiştir.

Saurabh R Prasad, Pawan K. Gaikwad ve Yashwant V Joshi tarafından hazırlanan ve Uluslararası Hesaplamalı Bilim ve Uygulamaları Dergisi’nde (IJCSA) yayınlanan “MATLAB Gui Kullanarak Ses Sinyalinin Gerçek Zamanlı Özel Etkileri Üretimi Ve Gürültü Filtrasyonu” isimli makaleleri MATLAB ile ses işleme üzerine hazırlanmış bir çalışma olması açısından önemlidir. Ayrıca bu çalışmada gerçek zamanlı ses işleme gerçekleştirilmiştir.

Markus Pfaff, David Malzner, Johannes Seifert, Johannes Traxler, Horst Weber, ve Gerhard Wiendl tarafından hazırlanan ve Dijital Ses Efektleri Konferansı’nda (DAFx-07) yayınlanan “Donanım/Yazılım Tasarım Yaklaşımı Kullanarak Dijital Ses Etkilerinin Uygulanması” adlı makalesi bazı efektlerin temel yapıları hakkında bilgiler vermektedir.

Novak A. , Simon L. ve Lotton P. tarafından hazırlanan “Ses Efektleri için Senkronize Edilmiş Süzme Yöntemi Kullanılarak Doğrusal Olmayan Sistemlerin Analizi, Sentezi ve Sınıflandırılması” adlı çalışma da “Sinyal işlemede gelişmeler” isimli EURASIP dergisinde yayınlanmış önemli makaleler arasındadır.

“Gerçek efekt analizine dayanan sinyal işlemcileri için dijital müziksel efektlerin modellenmesi” (“Modelling digital musical effects for signal processors, based on real effect manifestation analysis.”) isimli makale Dijital Ses Efektleri Konferansı’nda yayınlamış önemli bir eserdir.

“Vakum tüplü gitar amplifikatörlerinin modellenmesi için dijital tekniklerin gözden geçirilmesi.” (A review of digital techniques for modeling vacuum-tube guitar amplifiers. ) isimli makale Pakarinen J. ve Yeh D. T. Tarafından hazırlanmış ve Bilgisayar Müzik Dergisi tarafından yayınlanmıştır. Gitar üzerine modelleme üzerine olması açısından önem arz etmektedir.

Stark A. M. , Davies M. E. ve Plumbley M. D. hazırladıkları “Müzikal sesin gerçek zamanlı eşzamanlı analizi.” çalışması Dijital Ses Efektleri Konferansı’nda yayınlanmıştır. Bu çalışma gerçek zamanlı analiz çalışmaları için önemli bir kaynaktır.

### 3. DİJİTAL SES İŞLEME VE SES EFEKTLERİ

Bu bölümde tez çalışmasında kullanılacak gerekli kavramlar tanıtılmıştır. Ses efektlerinin tarihsel gelişimi ve Dijital ses işleme hakkında bilgiler verilecektir.

#### 3.1. Ses Efektleri ve Tarihsel Gelişimi

Ses, tek boyutlu bir sinyaldir, yani fiziksel miktar, sadece bir bağımsız değişken olan zaman ile değişir. Bu nedenle dijital alanda ses, vektörlerle temsil edilir. Müzik, video oyunları, hareketli görüntü, televizyon, animasyon vb. ve ayrıca canlı performans gibi multimedyaaların sanatsal ve diğer içeriğini vurgulamak için yapay efektler yapay olarak oluşturulur veya geliştirilir. Özel efektlerin çoğu, üç temel işlemle üretilir: Gecikme, skaler çarpma ve ekleme (Prasad, Gaikwad, & Joshi 2015).

Gecikme, her ses efektinin ayrılmaz bir parçasıdır. Bir birim gecikme elemanı, gelen numune akışını bir numune periyodu için saklar ve daha sonra çıkış örnek akışını serbest bırakır. Geciktirilmiş sinyal, bir defada veya istenen etkiyi yaratmak için birçok kez oynatılabilir. Gecikme süresi ve gecikme geri bildirimlerinin sayısı gerçekleşen ses efektini değiştirir.

20. yüzyılın ilk yıllarında bir gitarın ürettiği sesin daha yüksek ses seviyesine ulaşırma arzusu, daha küçük müzisyen gruplarının gruplar oluşturması, bu enstrümanın gelişimini büyük ölçüde hızlandırdı. 1920'li yılların sonlarında, çelikten yapılmış, üst düzey gitarlar, kalın iplik dizilerinin daha yüksek gerginliğini taşıyacak şekilde değiştirilmiş yapılar üretildi (Prasad, Gaikwad & Joshi, 2015). Tek bir dairesel ses deliği yerine, daha büyük bir ses projeksiyonu için iki f şekilli delikler kullanılmıştır.

Basit manyetikler ile donatılan elektrik akustik gitarları takip eden ilk katı gövdeli elektrik gitar 1948 yılında Fender tarafından tanıtıldı. Bu aynı zamanda elektronik ses efektlerinin de başlangıcıdır. Çoğu zaman gitar yükselteçleri (amplifikatör) , ses değiştirici aygıtlarla birleştirdi. Aynı zamanda enstrüman ve yükselteçleri, pedallar halinde birbirine bağlamak da yaygındı (Bacon, 2007). 1950'li yıllarda, tiz ve bas, yankılanma veya elektronik tremolo üniteleri için basit ton kontrolüne ek olarak sunulan amplifikatörler geliştirildi. 1960'larda gitaristler, bir amplifikatörün çıktısını bir diğerine besleyerek geri besleme ve bozulma ile deney yapmaya başladılar (Picarella, 2007).

### 3.1.1. Analog Gecikme

Ne analog ne de dijital ses teknolojilerinin bulunmadığı eski zamanlarda, gecikmeli bir eko kullanan müzikler, doğal olarak yankılanmış bir odada kaydedilmeliydi. Ancak, bu hem müzisyenlere hem de ses mühendislerine büyük rahatsızlık vermekteydi. Bu problem, istenilen aralık veya genlikte yankıları sentezleyebilen bir makineyi keşfetmek için bir ilham kaynağı olmuştur (Prasad, Gaikwad, & Joshi 2015).

### 3.1.2. Erken Gecikme Sistemleri

İlk gecikme etkisi üretimi “bant bazlı gecikme” olarak tanımlanır ve kayıt ortamı olarak manyetik bant kullanılır. Burada, gecikme süresi, bant döngüsünün uzunluğunun ayarlanmasıyla kontrol edilebilir. Bu teknik, Pierre Schaeffer gibi eski besteciler arasında en popüler olanıydı; Hammadde olarak kaydedilen sesleri kullanarak müzikal kompozisyonun deneysel bir tekniğidir. 20. yüzyılın Karlheinz Stockhausen gibi büyük bestecileri, uzun kayıtların çoklu kayıt cihazlarıyla kullanılmasını içeren ve canlı bir oyuncunun girişini işlemek için kullanılan karmaşık bir sistem geliştirdi. Ses mühendisleri bu teknikleri hızla uyarladılar ve aynı zamanda doğal yankıyı simüle etmek için tasarlanmış bir elektromekanik dönüştürücü olan "Plaka Reverb" olarak bilinen benzer teknikleri icat ettiler (Prasad, Gaikwad & Joshi 2015).

1940'ların sonlarında ve 1950'lerin başlarında, ses mühendisleri kendilerini kayıt stüdyolarında ve canlı performansta geçerli olan yeni cihazları icat etmeye başladılar. Yaygın dikkat çeken bant gecikmesi, fazlama ve çok kanallı kayıt oluşturmayı başarıyla denendi. Ray Butts, Amerikan gitaristi Chet Atkins'in ülke müziklerinde yaygın olarak kullanılan bir teyp yankısına sahip bir taşınabilir gitar amplifikatörü olan "EchoSonic" adlı bir simge markayı geliştirdi (Millard, 2004). Bant yankıları 1950'lerde piyasaya sunuldu. Bir “eko makinesi”, sesi tekrarlamak ve simüle edilmiş bir eko üretmek için elektronik aletler için kullanan ses işleme makinesinin ilk adıydı. Bu cihaz çok popüler oldu ve çağdaş ünlü gitaristler Brian May, Jimmy Page ve Syd Barrett tarafından kullanıldı. Bir başka analog geciktirme cihazı, bir bant halkası kullanılan "Echoplex" olarak adlandırılmıştır (Battenberg & Avizienis, 2011). Bu cihazda kayıt süresi ve kayıttan yürütme kafası arasındaki mesafe değiştirilerek gecikme uzunluğu ayarlanabilir. Bir başka benzer cihaz olan Roland'ın “Space Echo” olarak bilinen RE-201'i, 1973'de

Roland Corporation tarafından üretildi. Bu cihaz, deęişken bant hızına sahip bir kayıt kafası ve çoklu oynatma kafalarından oluşuyor ve analog ses geciktirme efektleri üretimi için kullanılıyordu. Bu cihazda, birbirini takip eden ekolar arasındaki zaman, bant hızı (Persson, F.Augustsson & Andersson, 2012) deęiştirerek ayarlandı. Eko sinyali, eko öncesi sinyale geri gönderilen eko sinyali miktarını deęiştirilerek ayarlandı (Prasad, Gaikwad & Joshi, 2015).

### 3.1.3. Dijital Gecikme

1970'lerin ve 1980'lerin sonlarında ucuz dijital elektronik devreler ve cihazların kolay kullanılabilirlięi, dijital gecikme üretme cihazlarının geliştirilmesine yol açmıştır. Başlangıçta pahalı raf tipi üniteler şeklinde mevcuttu, ancak daha sonra maliyetler düştü ve elektronik parçalar küçüldü, ayak pedalı şeklinde kullanılabilir hale geldiler (Jot & Chaigne, 1991). Pedalda sunulan ilk dijital gecikme 1984 yılında Boss DD-2 idi.

Dijital gecikme üretim makineleri sürekli zaman sinyalini örneklemede ve bir depolama arabelleęine kaydeden bir dizi dijital sinyal işlemcisinden geçmektedir. Saklanan ses örneklerinin oynatılması, kullanıcı tarafından ayarlanan parametrelere dayanarak gerçekleşmektedir. Geciktirilmiş çıkış, ya deęiştirilmiş sinyal ile ya geri besleme ya da ileri besleme şeklinde karıştırılır ve son olarak çıktı için dijital-analog dönüştürücüye tatbik edilmektedir.

### 3.1.4. Yazılım Gecikme

Dijital gecikme sisteminde bir sonraki adım, yazılım tabanlı gecikme sistemlerinin icadıydı. Yazılım gecikmeleri, dijital donanım gecikmelerinde daha fazla esneklik avantajından yararlanır. Yazılım gecikmeleri, ses arabelleęi için neredeyse sınırsız depolama olanaęı sağlayan kişisel bilgisayarlarda bulunan sistem belleęini kullanır. Yazılım gecikme algoritmalarının doğal verimlilięi, gecikme uygulamasını dięer eski teknolojilere kıyasla benzersiz kılmıştır. Bu nedenle, yazılım tabanlı gecikmeler, elektronik türlerdeki müzisyenler arasında çok popülerdir. Günümüzde ses sınavları, genellikle bilgisayarlı yazılım kontrollü dijital ses düzenleme ve karıştırma ortamında gerçekleşmektedir.



## 3.2. Dijital Ses İşleme

Dijital ses işleme, büyük miktarda esneklik sunmaktadır. Analog teknikleri kullanarak mümkün olmayan bir şekilde sesi işlemek için neredeyse sonsuz bir olasılık sunmaktadır. Son yirmi yılda, müzik stüdyosu, ağırlıklı olarak analog ekipmandan, neredeyse tüm dijital ekipmanlara, bir bilgisayar veya dijital ses iş istasyonuna dayanarak büyük ölçüde değişti. Geleneksel bir kayıt stüdyosunda, manyetik bant kullanan analog donanımlar, mikserler, efekt üniteleri ve devasa çok kanallı kayıt cihazları ile dolu birkaç oda bulunur. Bu alanda uğraşan bir teknisyen, bir kayıt için mükemmel yankı efektini bulmak için büyük ekipmanların olduğu bir stüdyoda günlerini geçirmesi gerekebilir.

Dijital sinyal işleme ve bilgisayar işlemci gücünün üstel bir hızda artmasıyla birlikte, bilgisayardaki en basit kişisel bilgisayarlarda bile tüm bu analog yankı ünitelerinin ve amplifikatörlerin bir bilgisayar tıklamasıyla yazılımlara aktarılması mümkün hale getirmiştir (Mike Hudson, 2012).

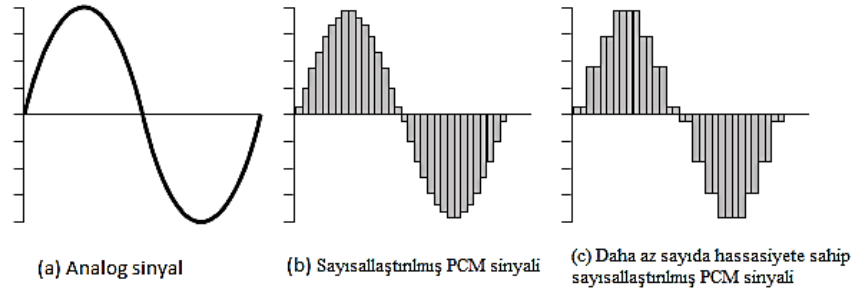
Bu önemli gelişme yeni enstrümanlar için yol açtı ve müzisyenler ve teknisyenler, yüzlerce yeni müzik türünün doğuşunu farklı bir şekilde üretmeye başladılar. Son on yıllık müzik teknolojinin gelişiminin, müziği doğrudan nasıl etkilediğine dair güzel bir örnek sunmaktadır (Smith, 1997).

Dijital sinyal işlemenin temel kavramları ve teorisi fazla değişmemiş olabilir ancak uygulandıkları teknoloji olağanüstü bir oranda gelişmeye devam etmektedir. Gömülü multimedya işleme teknolojisindeki değişim teknolojisinin ne kadar hızlı ilerlediğini göstermiştir. Bununla birlikte, daha yüksek hız, daha düşük maliyet ve daha küçük boyuttaki artışlara ihtiyaç duyulduğundan, dijital ses işleme uygulamalarının geleneksel sıralı dijital ses işleme işlemcilerin de sınırları görülmeye başlanmıştır (Mike Hudson, 2012).

### 3.2.1. Dijital Ses Teorisi

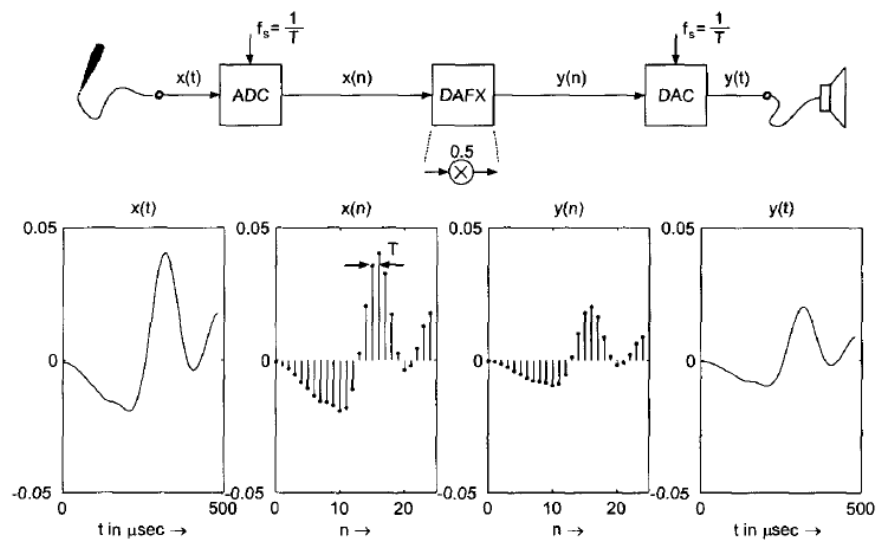
Genel olarak, dijital olarak işlenen sesin ilk aşaması, analog ses sinyalini bir analogdan dijital çevirici (ADC) kullanarak dijital bir gösterime dönüştürmektir. Sayısal sinyal işleme, daha sonra dijital sinyalden bir analog sinyali oluşturmak dijital-analog dönüştürücü (DAC) ile gerçekleştirilir. ADC örnekleri, analog sinyalin genliğini zaman

noktalarında, örneğin Nyquist'in örnekleme teoremi, örnekleme oranı veya  $f_s$  olarak kullanılacaktır. Amplitüd değeri tipik olarak, ses için, 16 ila 24 bit uzunluğunda bir ikili değer (PCM:Pulse Code Modulation) ile temsil edilir. Genel olarak, bit derinliği, daha hassas ve daha düşük kuantizasyon hatası bir sinyale sahip olduğunda gürültü artar. Şekil 3.1, hassasiyetin daha küçük bir bit uzunluğu ile nasıl kaybedebileceğini göstermektedir.



Şekil 3.1. (a) Analog sinyal (b) Sayısallaştırılmış PCM sinyali  
(c) Daha az sayıda hassasiyete sahip sayısallaştırılmış PCM sinyali

En temel efekt, bir ses kontrolünü, her dijital örnek değerini bir değerinden daha az bir değere çarparak gerçekleştirilebilir. Çoğu dijital ses örnek verisi, kesirli ikişer tamamlayıcı biçim olarak gösterildiğinden, -1'den +1'e kadar olan değerler, birden küçük herhangi bir sayı ile çarpılarak -1 ila +1 aralığında kalır. Şekil 3.2 'deki sonuç, girişin genliğinin yarısı olan bir çıkış sinyali olacaktır (Mike Hudson, 2012).



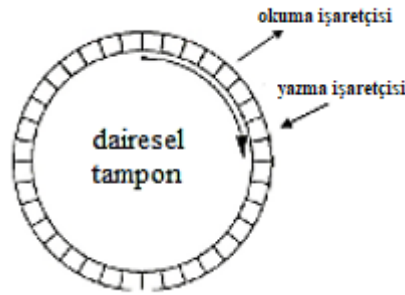
Şekil 3.2. ADC tarafından örnekleme ve niceleme, dijital ses efektleri ve DAC tarafından yeniden yapılandırma (Zölzer, 2011).

### 3.2.2. Temel Ses Efektlerinin Yapı Taşları

Sayısal sinyal işleme, genellikle örnek oranı cinsinden verilen, zaman içinde farklı noktalarda çok sayıda örneğin tekrarlanan çarpma ve toplamından (çoğul biriktirme) oluşur. Bazı yaygın dijital ses efektlerini gerçekleştirmek için, bu üç ana bileşenin oluşturulması gerekir:

- Değişken gecikme çizgisi
- Çarpan
- Akümülatör

Bu temel işlemler, çıktı üzerinde istenen etkiyi elde etmek için çeşitli yapılarda düzenlenmiştir. Değişken gecikme çizgisi, tipik olarak, önceki örneklenmiş değerlerin sabit bir miktarını depolayan bir miktar RAM tamponu olabilir. Yazma işaretçisinin belirtilen arabellek uzunluğuna ulaşması durumunda RAM'in bir döngüsel arabellek olarak uygulanması gerekir ve en eski örneğin üzerine yazılır. Bu, genellikle yazım işaretçisine atıfta bulunularak, tasarımcı tarafından yerleri belirtilebilen çoklu okuma işaretçilerine sahip olmak suretiyle gerçekleştirilir (Mike Hudson, 2012). Bu konsept Şekil 3.3'te okunabilir işaretçinin yazma işaretçisini geride bıraktığı yerde görselleştirilebilir.



Şekil 3.3. Dairesel tampon kavramı

### 3.3. Doğrusal Ses Efektleri

#### 3.3.1. FIR Gecikmesi

Sinyal işleme uygulamalarında, gecikme, hem zaman hem de frekans alanlarındaki etkileri beklenen en çok kullanılan yapı bloğudur. Genel olarak, alandaki ses mühendisleri, sonuçta frekans etki alanında meydana gelen değişiklikleri ihmal eden taraf üzerindeki etkilerle ilgili zaman alanı değişikliklerini dikkate alırlar (Zölzer, 2011). Gecikme ses efekti, zaman dilimindeki ayrık zaman sinyalini, hem zaman hem de frekans alanında (Şekil 3.4) matematiksel olarak aşağıda ifade edildiği gibi, bir tamsayı  $d$  sayısı ile değiştirerek uygulanabilir.

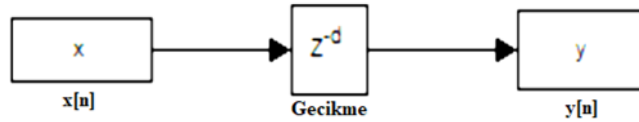
$$y[n] = x[n - d] \quad (3.1)$$

$$Y[e^{j\omega}] = X(e^{j\omega}) \cdot e^{-j\omega d} \quad (3.2)$$

$$H[e^{j\omega}] = e^{-j\omega d} \quad (3.3)$$

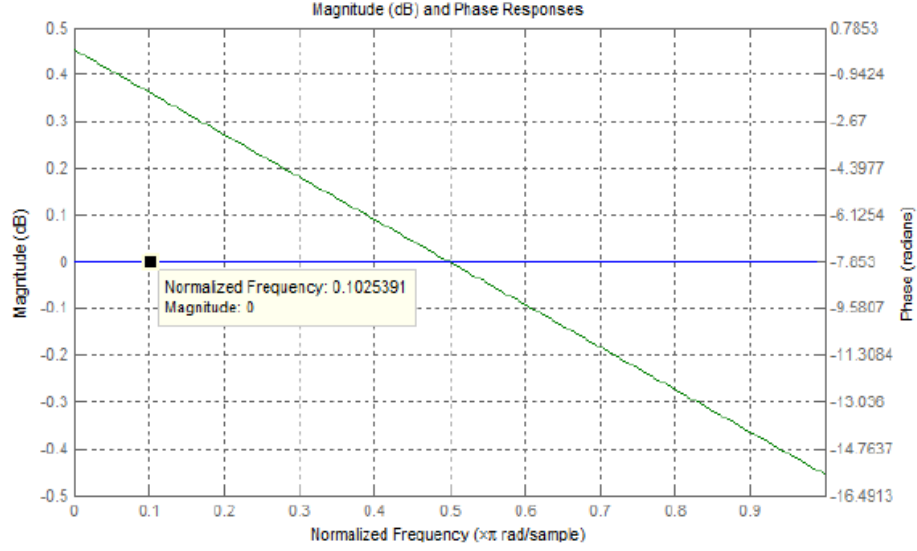
$$H[z] = z^{-d} \quad (3.4)$$

Ayrık Zamanlı Fourier Dönüşümü ve Z-dönüşümünden görülebileceği gibi, gecikme sinyali zaman içinde kaydırır ve aynı zamanda frekans alanında bir faz kaymasına neden olur. Beklendiği gibi, faz, frekans ile doğrusal olarak değişir, burada gecikme miktarı,  $d$ , eğim üzerinde karar verir. Şekil 3.5'den görebildiğimiz gibi, filtrenin büyüklüğü tüm kullanıcılar için 0 dB'de kalıyor.



Şekil 3.4 Temel Yapı Bloğu: Gecikme.

Bu filtrenin doğrusal olarak faz değiştirirken büyüklüğünün etkilenmediği anlamına gelir.



Şekil 3.5 Gecikmenin Büyüklüğü ve Faz Yanıtı (d = 5).

### 3.3.2. Flanger

Flanger temelde, gecikme, kesirli veya tamsayı zaman indeksi ile değiştiği, değişken gecikmeli dijital filtredir. Gecikmedeki değişiklik miktarı ve modeli, ne tür bir etki oluşturduğunu tanımlar. Bir ses flanger gecikmesi genellikle 0 ila 15 ms arasında sinüzoidal olarak değişir. Bu, giriş üzerinde zamanla sinüzoidal olarak değişen küçük değişiklikler oluşturur. Farklı gecikme işlevleri, jet uçak sesleri, özelleştirilmiş flanger desenleri ile oluşturulan bulanık ve korkak sesler gibi farklı efekt ve değişikliklere yol açabilir (Zeki, 2015). Değişken gecikme aşağıdaki gibi matematiksel olarak gösterilebilir.

$$A = \text{Floor}\left(t_{\text{flangemax}} \cdot \frac{T_s}{2}\right) \quad (3.5)$$

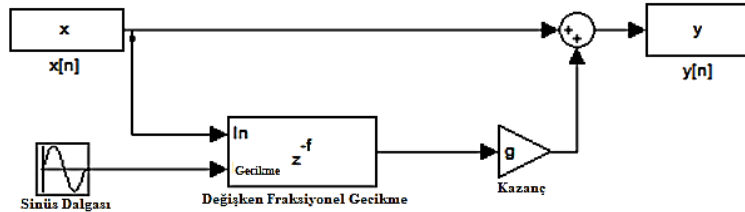
$$M[n] = \text{Floor}(A + A \sin(2\pi F_{\text{flange}} n T_s)) \quad (3.6)$$

$$z[n] = x[n - M[n]] \quad (3.7)$$

$$y[n] = x[n] + g \cdot z[n] \quad (3.8)$$

$$y[n] = x[n] + g \cdot x[n - M[n]] \quad (3.9)$$

Flanger efektinin kökenleri, iki eşzamanlı kayıtların doğrudan bant karışmasından gelir. Karıştırma bandının hızındaki küçük farklar, orijinal karışımında değişken gecikme uygulanmasıyla matematiksel olarak aynı olan fazlama etkisine yol açar. Flanger efektini, ses efektlerinden, değişken gecikme sürenizi ve hızınızı değiştireceğiniz koro benzeri küçük varyasyon efektlerine değiştirir. Sinyal akış şeması Şekil 3.6'daki gibi çizilmiştir.



Şekil 3.6 Flanger Efekt Sinyal Akışı

### 3.3.3. Yankılanma

Bir oda veya bir konser salonunda, yankılanma olduğunda ses dağınık halde oluşur ve bir ortamdan geri gelir. Yankılanmada çevredeki malzemelerin özellikleri faz, genlik ve frekans karakteristikleri etkilidir.

Bir yankı algılama algoritması tasarlarırken, statik ses kaynağı ve dinleme konumlarıyla ilgili bilgiler önemlidir. Yankı sistemleri üç kısma ayrılır: doğrudan ses, erken yansımalar ve geç yankılanma. Literatürde tanımlandığı gibi, doğrudan ses, önce dinleyiciye ulaşan ses dalgasıdır (Schroeder & Logan,1961). Erken yansımalar ses kaynağından hızlı gelen yansımalarıdır. Geç yansımalar, çevreden gelen farklı yansımalarıdır.

Akustik ses modellemede, oda veya konser salonu gibi kapalı bir ortamın işlenmesinde, ses yayılırken ve çevreden geri döndüğünde yankı oluşur. Çevresel materyallerin özellikleri, yankılanma fazı, genliği ve frekans karakteristikleri için etkilidir (Zeki, 2015).

### 3.3.3.1. Yankılanma Ortam Modellemesi

Yankılanma ortamları bir oda, konser salonu, mağara, arena, hangar vb. simülasyonlarını içerir. Farklı yankılama algoritmaları, farklı yankı ortamlarını modellemek için tasarlanmıştır. Bu algoritmaların temelleri, odanın yansımalarının ve saçılmasının göz önünde bulundurulduğu oda akustiğinin modellemesidir. İnsan kulağı, odanın gerçek şekli arasında sadece çıkarılan çıktı ile tam olarak ayrılmamaktadır. Bu, şekil parametresinin bulunmamasının sebebidir, ancak yankı sistemini tasarlarken göz önünde bulundurulan sadece genlik ve faz zayıflamalarıdır.

Yankılanmanın ilk uygulamaları arasında, Manfred Schroeder'ın yapay yankılanma üzerindeki çalışmaları en önemlisidir (Schroeder & Logan,1961). 1960'lardaki çalışmasında, gecikme tabanlı all-pass filtreler ve özyinelemeli comb filtreleri kullanarak ekoların karmaşık modellerini tasarladı (Schroeder & Logan,1961). Matematiksel olarak, gecikme çizgilerini kullanarak temel özyinelemeli yankılanma denklemi 3.10'da gösterilmiştir.

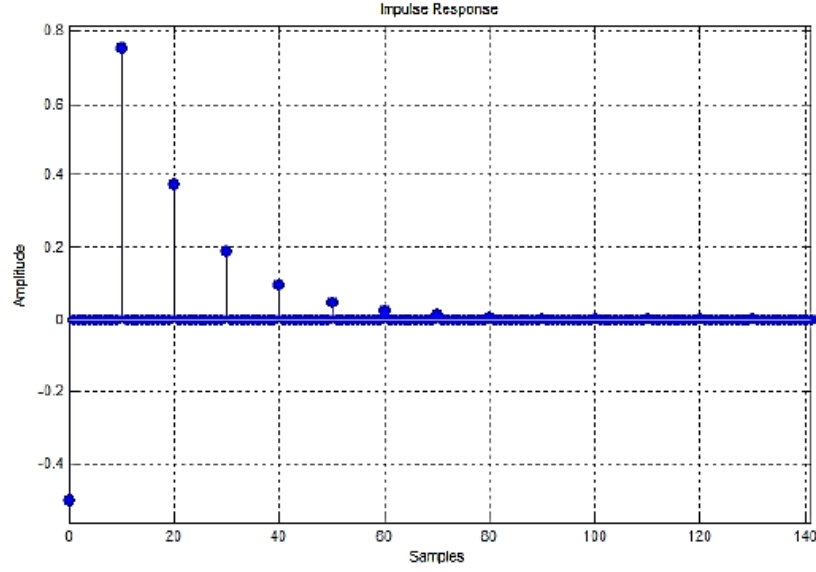
$$y[n] = -g \cdot x[n] + x[n - m] + g \cdot y[n - m] \quad (3.10)$$

Bu özyinelemeli girdi çıktı ilişkisinin ne olduğunu anlamının en iyi yolu, özyineli yankılama sisteminin dürtü yanıtına bakmaktır. Ayrıca, her bir çıktı örneğine toplanan toplamdan daha az bir kazancı olan bir geri besleme döngüsü olduğunu da görebiliriz.

Bu geri bildirim döngüsü, oda, konser salonu gibi bir yankı ortamının akustiğini oluşturmaktadır. Yankılanmanın tepki cevabı matematiksel olarak denklem 3.11'de gösterilmiştir. Kazanç parametresi olan  $g$ , yankılanmanın dürtü yanıtının üssel bozulma oranına karar verir.

$$h[n] = -g \cdot \delta[n] + (1 - g^2)[\delta[n - m] + g \cdot \delta[n - 2 \cdot m] + \dots] \quad (3.11)$$

Yankılanma dürtü yanıt fonksiyonu, bir yankı etkisi yaratır. Özyinelemeli yankı denkleminin darbe tepkisi Şekil 3.7'de gösterildiği gibidir. Dinleyici, etraftaki duvarlardan veya diğer nesnelere saçılma olduğunu düşünür.



Şekil 3.7 Özyinelemeli Yankı Denklemine Darbe Tepkisi.

### 3.3.4. Dijital All-Pass Filtresi (Dijital Tüm-Geçiren Filtre)

Dijital all-pass filtresi, frekans cevabının büyüklüğünü değiştirmez, ancak giriş sinyalinin  $x[n]$  fazını değiştirir. Denklem 3.12'de, dijital all-pass filtrenin ortak bir formu matematiksel olarak gösterilmiştir.

$$A_p^g(z) = -\frac{g + z^{-1}}{1 + g \cdot z^{-1}} \quad (3.12)$$

$$A_p^g(e^{j\omega}) = -\frac{g + e^{-j\omega}}{1 + g \cdot e^{-j\omega}} \quad (3.13)$$

$$|A_p^g(e^{j\omega})| = -\frac{|g + e^{-j\omega}|}{|1 + g \cdot e^{-j\omega}|} \quad (3.14)$$

$$|A_p^g(e^{j\omega})| = -\frac{|g + \cos(\omega) - j \cdot \sin(\omega)|}{|1 + g \cdot \cos(\omega) - j \cdot g \cdot \sin(\omega)|} \quad (3.15)$$

$$|A_p^g(e^{j\omega})| = -\frac{\sqrt{(g + \cos(\omega))^2 + \sin^2(\omega)}}{\sqrt{(1 + g \cdot \cos(\omega))^2 + g^2 \cdot \sin^2(\omega)}} \quad (3.16)$$

$$|A_p^g(e^{j\omega})| = 1 \quad (3.17)$$

Bir dijital all-pass filtresi, bir analog all-pass filtrenin tam olarak dönüşümüdür. Dijital all-pass filtresi, sinyal akış yolunda farklı parametrelerle kademeli olarak



kullanıldığında fazör görevi görür. Dijital all-pass filtrenin düz bir büyüklük yanıtı vardır, ancak doğrusal faz yanıtı yoktur. Bu özellik, bu filtreyi, büyüklük özelliklerini değiştirmek istemediğimiz etkilere uygulanabilir kılmaktadır. Dijital all-pass filtreyi kullanarak, aynı anda faz karakteristiklerini değiştirebilir ve zamanla değişen efektler elde edebiliriz (Zeki,2015). Değişen gitar ses efektleri için, dijital all-pass filtre bir yapı taşı olarak kullanılır. Matematiksel terimlerle, all-pass filtrenin faz cevabı denklem 3.21'de gösterilmiştir.

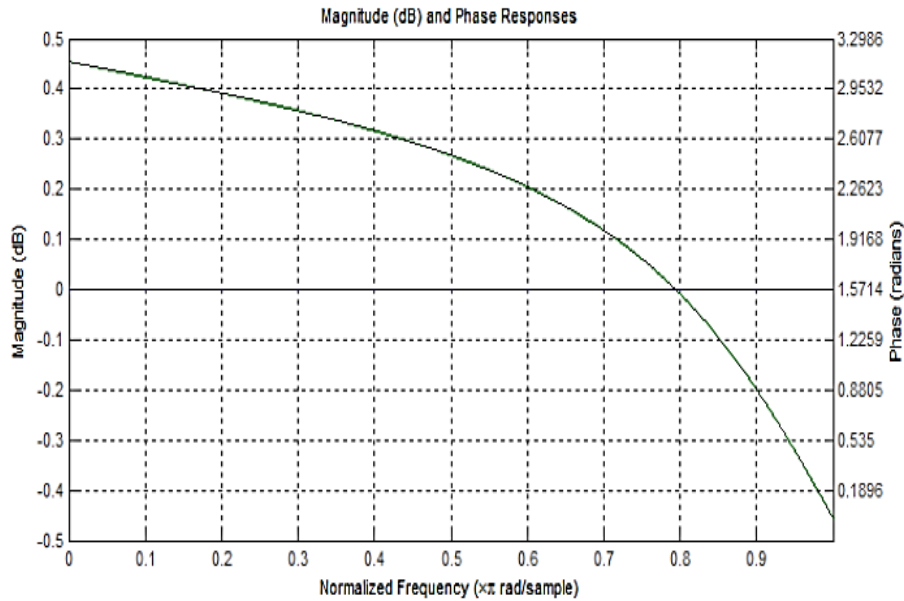
$$A_p^g(z) = -\frac{g + z^{-1}}{1 + g \cdot z^{-1}} \quad (3.18)$$

$$\text{Arg}(A_p^g(e^{jw})) = \text{Arg}(-g - e^{-jw}) - \text{Arh}(1 + g \cdot e^{-jw}) \quad (3.19)$$

$$\text{Arg}(A_p^g(e^{jw})) = \text{Arg}(-g - \cos(w) + j \cdot \sin(w)) - \text{Arg}(1 + g \cdot \cos(w) - j \cdot g \cdot \sin(w)) \quad (3.20)$$

$$\text{Arg}(A_p^g(e^{jw})) = \arctan\left(\frac{\sin(w)}{-g - \cos(w)}\right) - \arctan\left(\frac{-g \cdot \sin(w)}{1 + g \cdot \cos(w)}\right) \quad (3.21)$$

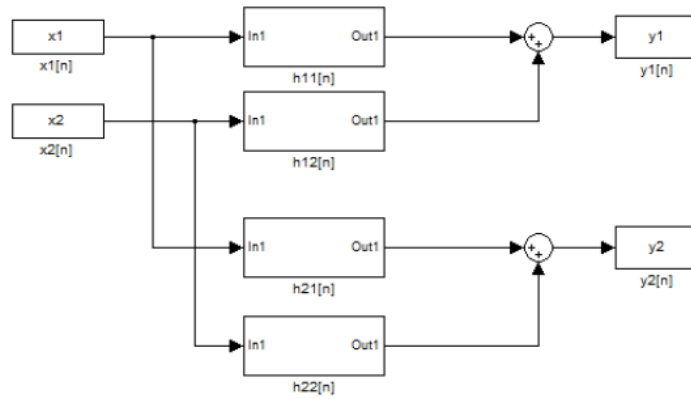
Şekil 3.8'e baktığımızda, all-pass filtresini karakterize eden düz bir frekans cevabı görebiliyoruz. Faz yanıtı, frekansla azaltan denklem 3.21'in doğrudan temsidir.



Şekil 3.8 Dijital All-Pass Filtre Frekansı ve Faz Yanıtı Çizimi.

All-pass filtresi genel yankı etkileri için yapı taşıdır ve ses efektleri literatüründe yaygın olarak kullanılır.

Yankılanma problemi Julius O. Smith III tarafından Fiziksel Ses Sinyal İşleme adlı çalışmasında tartışılmıştır (Yeh, Abel & Smith, 2010). Bu çalışmada, bir ses kaynağından yayılan doğrudan sinyalin, bir zayıflama ölçekleme veya alçak geçiren filtre ile seri halde tek bir gecikme çizgisi kullanılarak simule edilebileceği belirtilmektedir. Model, kaynak sinyalinin genliğini ve fazını değiştirebilen, hava emilimi ve yansıtma materyalinin etkilerini modellemek için tasarlanmış bir filtreden geçmektedir (Battenberg & Avizienis, 2011). Bu yöntemin dezavantajı, kaynağı sadece dinleyiciye (noktadan noktaya) sesli akustiğe göre modelleyebilmemizdir. İki kaynak için böyle bir transfer fonksiyon sistemi ve iki kulaklı bir dinleyici (girişler/çıkışlar) Şekil 3.9'daki gibi MATLAB ortamında modellenilebilir.



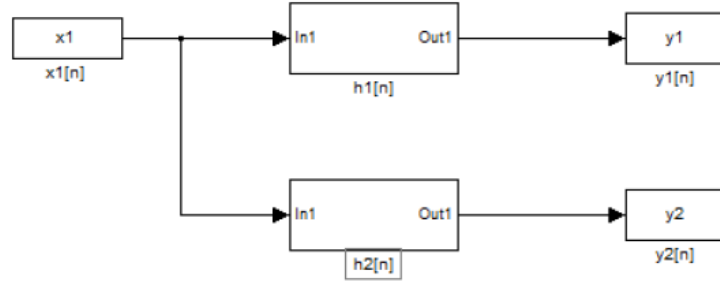
Şekil 3.9 Transfer Fonksiyonu Modelleme ile Tam Yankı.

Şekilde görülebileceği gibi  $x_1[n]$  ve  $x_2[n]$  giriş ses kaynaklarıdır.  $h_{11}[n]$  ve  $h_{12}[n]$  ilk kulağa çıkış olan  $y_1[n]$  için transfer fonksiyonlarıdır.  $h_{21}[n]$  ve  $h_{22}[n]$  de ikinci kulak çıkışına  $y_2[n]$ 'e için transfer fonksiyonlarıdır. Bu durumda  $y_1[n]$  ve  $y_2[n]$  çıkışları, soldaki ve sağdaki iki kulak olarak görülebilir. Transfer fonksiyonu, aşağıdaki gibi z-alanında bir matris ilişkisi olarak denklem 3.22'de gösterilmiştir.

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} H_{11}(z) & H_{12}(z) \\ H_{21}(z) & H_{22}(z) \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_2(z) \end{bmatrix} \quad (3.22)$$

Elektrogitar yankılanması durumunda, tek bir ses kaynağımız vardır. Elektrogitar girişi ve aynı girişi yapan iki ayrı sistem, sırasıyla sol ve sağ kulak çıkışlarına  $y_1[n]$  ve

$y_2 [n]$  olarak veriyor. Bu tür bir sistem MATLAB'da Şekil 3.10'da olduğu gibi gösterilebilir.



Şekil 3.10 Transfer Fonksiyonu Modelleme ile Tam Gitar Yankısı

Elektrogitar yankılanma modeli için tam denklem 3.23'de verilmiştir. Gitar girişi sırasıyla  $x [n]$ , sol ve sağ için sırasıyla  $h_1 [n]$  ve  $h_2 [n]$ , sol ve sağ kulak için iki çıkış  $y_1 [n]$  ve  $y_2 [n]$  olarak belirtilir.

$$y_i[n] = \sum_{i=1}^2 x[n] * h_i[n] = \sum_{m=0}^{M_i} x[n].h_i[n - m], i = 1,2. \quad (3.23)$$

Her kaynak ve dinleyici arasındaki modelleme yanıt cevabı, hesaplama açısından karmaşık olabilir. Bu nedenlerle, yankılanmayı modelleme yöntemleri daha kullanışlı hale gelir ve bu yöntem yapay yankılama olarak adlandırılır. Akustik yankılanma ortamının modellenmesi, çarpım ve eklemelerin sayısının, odanın genişliğinin, boyunun ve uzunluğunun her bir inçte çok daha yüksek olmasından dolayı gerçek zamanlı uygulamaları kısıtlayan gecikmeyi de beraberinde getirmektedir. Hesaplamaları çok daha kolay hale getirmek, yankı desenlerini modellemek ve bu problemlerin üstesinden gelmek için çözüm olarak yapay yankı algoritmaları getirilmiştir.

### 3.3.5. Yapay Yankı

Yapay yankı, kaynaklardan gelen sinyalin davranışını inceleyen yankılanmayı yeniden oluşturmak için yöntemlerin geliştirildiği, yankılanmanın algısal modellemesidir. Bu yöntemler, algoritmalarda gerçek yankı yollarını taklit eden basit modellerin kullanıldığında etkilidir (Zeki, 2015).

Yapay yankı için yankı yoğunluğu önemlidir. Yankı yoğunluğu, ses kaynağından birim zaman başına düşen yansıma ve geri dönen ses dalgalarının sayısıdır. Yankı yoğunluğu, kaynaktan çıkan ilk ses dalgasından başlayan, zamanla orantılı olan kare şeklindedir. Bu kısımda Manfred R. Schroeder'ın Renksiz Yapay Yankı ve Doğal Sondaj Yapay Yankısı hakkında bilgi vermek gerekmektedir (Zeki, 2015).

### 3.3.5.1 Renksiz Yapay Yankılanma

1961'de Manfred R. Schroeder ve B. F. Logan, Audio Engineering Society Dergisi'nde "Renksiz Yapay Yankı" adlı bir makale sundu (Schroeder & Logan, 1961). Bu yazıda dijital yankılama terimi tanımlandı. Modern yankı teknikleri, dijital yankı uygulamalarına dayanmaktadır. Bu makale, bu nedenle dijital ses efektlerinde bir kilometre taşı olarak kabul edilir.

Yapay yankılanma tasarımına duyulan ihtiyaç, analogik alandaki yankılamaların maliyetli ve verimsiz olması durumundan ileri gelmektedir. Bir odadaki çoklu yansımaları simüle etmek için yapılan araştırma, bu yansımaları simüle etmek için yapı taşlarının tasarımı gerekliliğini ortaya koymaktadır. Yankı frekans cevabındaki değişiklikler uygulanarak "renkli" yankılanmalar meydana gelir. Schroeder ve Logan'ın çalışması, aynı yankı etkisi için bir all-pass filtresi öneren bu problemin çözümünü amaçlamaktadır.

Elektronik yankılanmalar, gerçek yankı odalarını doğrudan modelleme konusunda belirli avantajlara sahiptir. Modelleme ve yankı odalarına sahip olmak için gerekli olan maliyet herkesin karşılayabileceği bir şey değildir. Diğer yandan, gömülü ses kodeğine sahip herhangi bir mikroişlemci mimarisinde, elektronik bir yankılanma uygulanması kolaydır. Bununla birlikte, yankılanmayı temsil etmenin ideal bir yolu, genel olarak bir LTI (Doğrusal Zamanla Değişmeyen) sistemi olarak kolayca temsil edilebilmesi açısından yapay olmasıdır.

Geniş odaların frekans cevabını simüle etmek yankılama algoritması tasarımlarında zorlu bir sorundur. Schroeder makalesinde, bir odanın, Denklem 3.24'deki gibi matematiksel olarak tarif edilebilen normal titreşim modlarıyla karakterize olabileceğini belirtmektedir.

$$\text{Mod Sayısı} = \frac{4\pi V}{c^3} f^2 \quad (3.24)$$

Denklem 3.24'den, modların sayısının, oda şekli ve frekansla orantılı olarak bağımsız olduğunu görebiliyoruz. Bu denklemde  $V$  oda hacmidir,  $c$  sesin hızıdır ve  $f$  frekanstır. Birçok modun yüksek yoğunluklu ekoları nedeniyle üst üste geldiği an olan kritik frekans olan  $f_c$  frekansı vardır. Kritik frekans, Denklem 3.25'te matematiksel olarak gösterilebilir.

$$f_c = 2000 \sqrt{\frac{T}{V}} \quad (3.25)$$

Denklem 3.25'te,  $T$  saniye cinsinden yankılanma süresidir ve  $V$  m<sup>3</sup> cinsinden odanın hacmidir. Frekans, bu frekans aralığına yaklaştığında ve bu frekans aralığının üstünde olduğunda, genel olarak düzensiz genlik frekans yanıtı ile sonuçlanan modların artan sayısı nedeniyle bireysel saçılmaları ve yankılanmaları duymak zordur. Bir insan kulağı için, bu dalgalanmaları frekans cevabında ayırt etmek zordur, bu yüzden onları genel bir değişim olarak algılayabiliriz. Yankı odalarını test etme yöntemi, farklı dürtü tepkilerine yol açan farklı frekanslara sahip sinüs dalgaları uygulamak ve odanın duvarlarından yansıyan çıkış sinyallerini duymaktır. Bir başka çok etkili yöntem ise, daha akıcı dürtü yanıtları bulmak için psiko-akustik olarak geliştirilmiş test sinyallerini, özel sesleri kullanmaktır. Renksiz yankılanana kadar, yapay elektronik yankı maddelerinin tek dezavantajı, bir odanın düzgün frekans tepkisini modellemenin zor uygulanabilirliğidir. Renkli yankılanma algoritması denklemi 3.26'da gösterilmiştir. 3.26'nın büyüklük cevabı ve  $z$  dönüşümü aşağıdaki denklemlerde verilmiştir.

$$y[n] = x[n - d] + g \cdot y[n - d] \quad (3.26)$$

$$H(e^{j\omega}) = \frac{e^{j\omega d}}{1 - g \cdot e^{-j\omega d}} \quad (3.27)$$

$$H(z) = \frac{z^{-d}}{1 - g \cdot z^{-d}} \quad (3.28)$$

$$H(z) = \frac{z^{-10}}{1 - 0,5 \cdot z^{-10}} \quad (3.29)$$

$$H(e^{j\omega}) = \sum_{k=1}^{\infty} g^{k-1} \cdot e^{-j\omega kd} \quad (3.30)$$

$$|H(e^{jw})|^2 = \frac{1}{1 + g^2 - 2.g.\cos(w.d)} \quad (3.31)$$

$$|H(e^{jw})|^2 = \frac{1}{1 + g^2 - 2.g.\cos(10w)} \quad (3.32)$$

$$h[n] = \sum_{k=1}^{\infty} g^{k-1} \cdot \delta[n - k.d] \quad (3.33)$$

Darbe tepkisi ve Z-Dönüşümü'nden de görülebileceği gibi, frekans cevabının renkli bir şekilde tarak filtresi gibi radyal olarak etkilenmesi açısından renklendirildiğini söyleyebiliriz. Renkli yankı algoritması, girişin frekans karakteristiğinin değişmesi amaçlanmadığında kullanılmaz.

### 3.3.6. Erken Yankılanma

Erken yansımalar, ses kaynağından işitme cihazına (kulak) yaklaşık 100 ms sonra gelen yansımalarıdır. Bu 100 ms'lik kısa süre, erken yansımaları modellemek için dişli gecikme çizgilerini (TDL) kullanmamızı sağlar. Erken yansımalar, ses kaynaklarının 3D uzaysal konumlandırmasını kulaklarımıza göre oluşturabildiğinden algısal olarak önemlidir, bu nedenle mümkünse, mekansal özelliklerin erken yansılardan modellenmesi daha iyidir (Zeki, 2015).

Ardışık gecikme çizgileri (TDL) ile erken yansımaları modellemek, istenen yankılanmayı elde etmek için en yaygın yaklaşımdır. Hava emilimi ve yansıma deseni modeli, önce giriş sinyalini filtreleyerek ve daha sonra TDL'den geçirerek yapılabilir. Erken yansımalar ve geç yankılar eş zamanlıdır. Erken yankılanmalar ortadan kalkmaya başladığı noktada, geç yankılanma görünmeye başlar.

### 3.3.7. Geç Yankılanma

Geç yankılanma, iki önemli özelliğe sahip bir yankı dürtü yanıtı gerektirir: akıcı zayıflama ve akıcı bir frekans cevabı (Schroeder & Logan,1961). Doğal yankılanma katlanarak azalır, bu bize genel olarak sorunsuz bir dürtü yanıtı verir. Buna ek olarak, zayıflayan bir gürültü süreci örnek alınabilir.

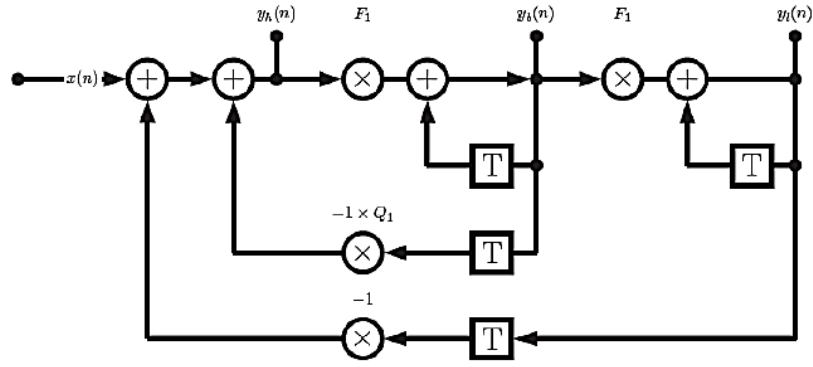
Frekans cevabını pürüzsüz ve düzensizliklerden arındırmak için, mod yoğunluğunun özellikle yüksek olması gerekir. Bu seçim, genel olarak düzgün bir frekans cevabına neden olacak şekilde düzgün yayılmak için modlara yol açar (Zeki, 2015).

Moorer, geç yankılanma dürtü yanıtı olarak üstel olarak azalan beyaz gürültünün yaratılması yöntemini denedi (Schroeder & Logan,1961). Beyaz gürültü geç yankılanma için gerekli olan koşulların her ikisine de karşılık gelir: akıcı zayıflama ve akıcı frekans tepkisi. İdeal geç yankılanma ile karşılaştırıldığında, üssel olarak bozulan beyaz gürültü dürtü yanıtının ortaya çıkmasıyla elde edilen gerçek geç yankılamadan daha renkli olduğunu söyleyebiliriz. Geç yankılanmada, Schröder'in önerdiği gibi basamaklı all-pass filtreler kullanarak, girişin frekans yanıtını değiştirmeden istenen geç yankı algoritmalarını elde edebilmemizdir (Schroeder & Logan,1961). Bu amaç için bile, all-pass bölümleri için optimize edilmiş özel donanımlar bugünün ses uygulamaları için üretilmiştir. All-pass filtre tasarımı, ses mühendislerinin tasarım prosedürünü ikiye ayırmalarını sağladı: İlk olarak yankılanma süresini ve yoğunluğunu tasarlamak ve ikinci olarak girişe istenen tepkiyi filtrelemek suretiyle girişe istenen rengi eklemek. Çevreyi sanallaştırmak ve ayırmak için Schroeder, rezonans yanıtını değiştirmeden ideal, sanal bir yankılanma olarak tanımladığı renksiz yankılanma terimini tanımlamıştır (Schroeder & Logan,1961).

### 3.4. Wah-Wah Efekt

Wah-wah efekti, ismine benzer şekilde, temel insan sesi "Wah" ile giriş sinyalinin bir varyasyonunu üretir. Wah-wah efekti aslında bant geçiren filtrenin merkez frekansının (fc) zamanla değiştiği, zamana bağlı bir bant geçiş filtresidir. Değişen merkez frekansı ile bu zamana bağlı filtreleme, wah-wah adı verilen bu benzersiz etkiyi yaratır. Wah-wah genellikle rock elektrogitar parçalarında kullanılır ve gitar kumandasına ayak kumandalı pedal olarak sunulur. Bu, gitaristin merkez frekansına karar vermesini sağlar.

Wah-wah efekti zamana göre değişen bant geçiren filtresinin yapısı durum değişken filtresidir. Durum değişken filtresinin ana avantajı, aynı anda düşük geçişli, bant geçiren ve yüksek geçişli filtre sağlama yeteneğidir. Wah-wah efektinde kullanılan durum değişken filtre yapısı Şekil 3.11'de gösterilmiştir.



Şekil 3.11 Wah-Wah Etkisinde Kullanılan Durum Değişken Filtresi.

Şekil 3.11'e baktığımızda giriş  $x[n]$ , düşük geçişli çıkış  $y_l[n]$ , bant geçişi çıkış  $y_b[n]$  ve yüksek-geçiş çıkışı  $y_h[n]$ 'yi görebiliriz. Çıkışlar, T'nin bir birim gecikme bloğunu temsil ettiği durum değişken filtresi kullanılarak oluşturulur. Çıktı denklemleri Denklem 3.34'den başlayarak verilmiştir.

$$y_1[n] = F_1 y_b[n] + y_1[n - 1] \quad (3.34)$$

$$y_b[n] = F_1 y_h[n] + y_b[n - 1] \quad (3.35)$$

$$y_h[n] = x[n] - y_1[n - 1] - Q_1 y_b[n - 1] \quad (3.36)$$

Genel olarak, wah-wah efekti, bu aralığın ötesindeki harmonikleri tanıtan ve çoğaltan distorsiyonla birlikte kullanılır; bu nedenle, daha yüksek bant genişliğini kapsayacak şekilde 500 ila 3000 Hz arasında değişen bir bant geçişi kesme frekansı seçilmektedir.

Wah-wah gitar ses efektleri en ünlü pedallarından biri Jim Dunlop Orijinal GCB95 modelidir. Bu wah pedalı 1966 Thomas modelinden ve temel olarak tasarlanan en eski wah pedalı. Wah pedalının altındaki temeller zamanla değişen bant geçiş filtrelemesidir, bu nedenle filtrenin şekli ve aralığı çıktı sesinde önemli bir rol oynar.



## 4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu bölümde tez için geliştirilen uygulamaların sonuç çıktıları MATLAB ortamında ve Simulink üzerinden verilmiştir.

### 4.1. MATLAB ile Ses Efekt İşleme

Temel dalga formlarını ve temel filtrelemeleri ile bazı dijital ses efektleri MATLAB ortamında incelenmiştir. Ses efektlerinden gecikme, yankı, wah-wah ve flanger efektlerinin modellenmesi gerçekleştirilmiştir. MATLAB kullanılarak efektlerin ses sinyali işleme yöntemleri ve filtreleri ile çalışma mantıkları incelenmiş ve doğrulanmıştır. Ses efektlerini işleme biçimlerini aşağıdaki gibi belli kategorilere ayırabiliriz.

Temel Filtreleme	: Alçak Geçiren, Yüksek Geçiren Filtre vb. , Ekolayzır
Zaman Değişken Filtreleri:	Phaser, Wah-wah
Gecikmeler	: Eko, Vibrato, Flanger, Chorus(Koro)
Modülatörler	: Ring modülasyon, Tremolo, Vibrato
Özel Efektler	: Panning, Reverb

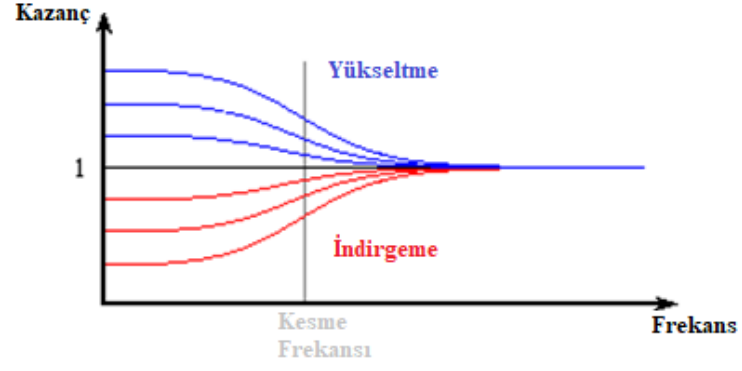
Bu kategorilerde yer alan efekt biçimleri MATLAB ile gerçekleştirilmiş ve ses efekt işleme için temel yapıyı oluşturmuştur.

#### 4.1.1. Temel Dijital Ses Filtreleme Efektleri

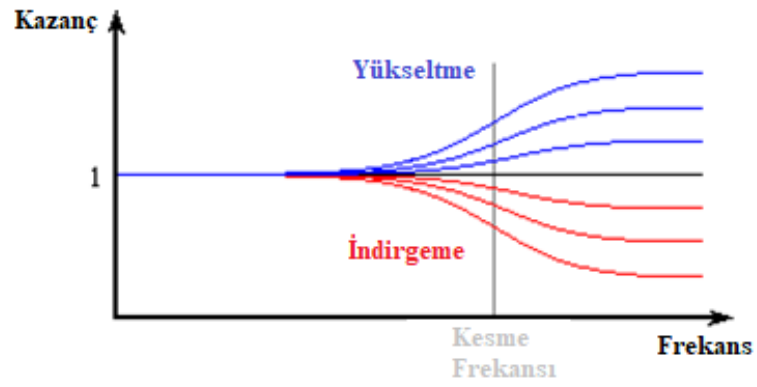
Ekolayzırlar: Ekolayzırlar, bazı frekans bantlarını artırır/ azaltır, diğerlerini değiştirmeden bırakır. Bir dizi shelving ve peak filtresi kullanılarak oluşturulmuştur. Genellikle birinci veya ikinci dereceden filtreler kullanılır.

#### 4.1.2. Ekolayzır Filtreleri (Shelving ve Peak Filtreleri)

Ekolayzır için Shelving Filtresi: Düşük frekanslı veya yüksek frekanslı Shelving Filtreler Şekil 4.1 ve Şekil 4.2 de gösterilmiştir.

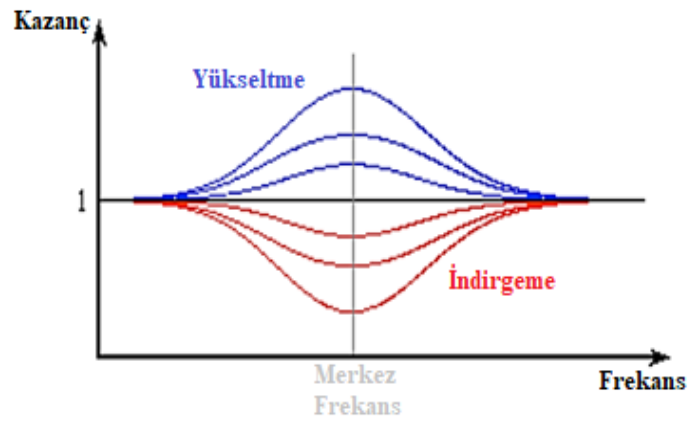


Şekil 4.1 Alçak Geçiren Shelving Filtre Frekansı Tepkisi



Şekil 4.2 Yüksek Geçiren Shelving Filtre Frekansı Tepkisi

Ekolayzır için Peak Filter: Bir kesme frekansı  $F_c$ , bir bant genişliği  $f_b$  ve kazanç  $G$  ile orta frekans bantlarını artırır veya keser (Şekil 4.3).



Şekil 4.3 Zirve (Peak) Filtre Frekansı Tepkisi

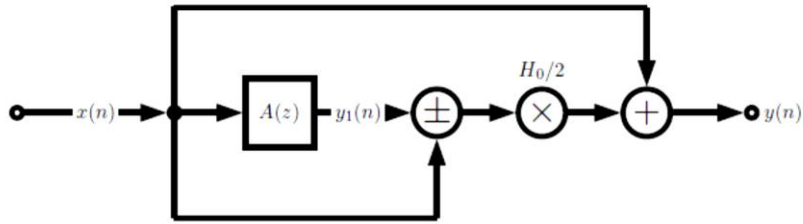
Birinci derece Shelving filtresi transfer fonksiyonu ile tanımlanabilir (Denklem 4.1):

$$H(z) = 1 + \frac{H_0}{2} (1 \mp A(z)) \quad (4.1)$$

Burada  $A(z)$  birinci mertebeden bir allpass filtredir. Tüm frekansları geçer ancak fazı değiştirir (Denklem 4.2):

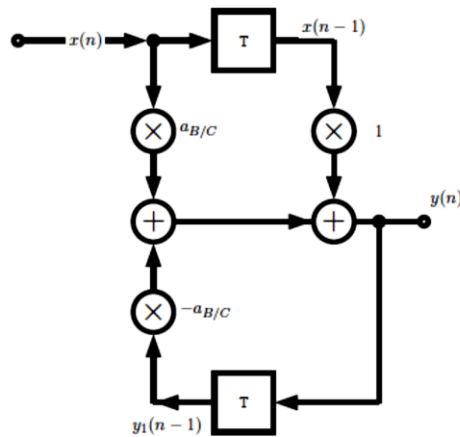
$$A(z) = \frac{z^{-1} + a_{B/C}}{1 + a_{B/C} z^{-1}} \quad B = \text{yükseltilmiş}, C = \text{kesilmiş} \quad (4.2)$$

Shelving filtreleri sinyal akış grafiği Şekil 4.4 te gösterilmiştir.



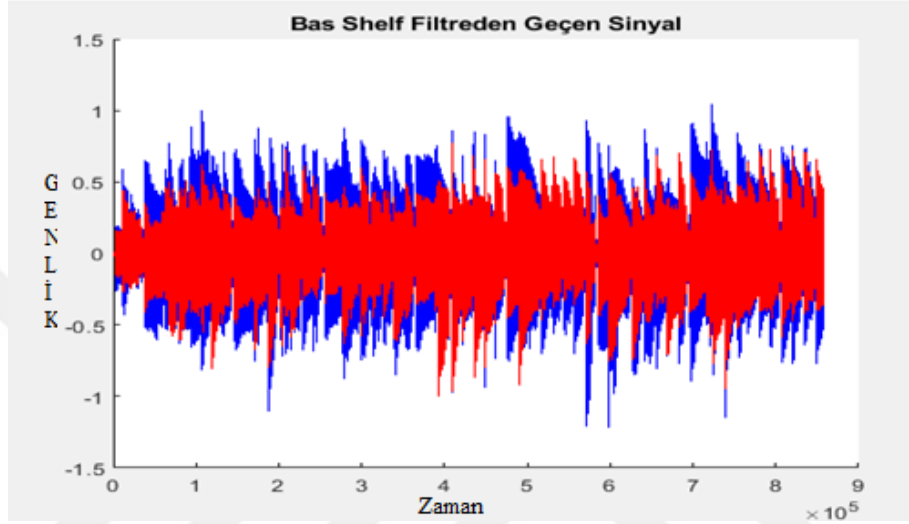
Şekil 4.4 Shelving Filtreleri Sinyal Akış Grafiği

Burada  $A(z)$  Şekil 4.5 de verilmiştir.



Şekil 4.5  $A(z)$

Bas Shelving Filtrenin Ek-1 de sunulan MATLAB koduna göre çıktısının gösterimi Şekil 4.6 da gösterilmiştir. Akustik gitar solosuna karşı bas shelf filtre sonuçları görülmektedir. Burada kırmızı ile gösterilen gerçek ses , mavi ile gösterilen filtre sonucunda çıkan sestir. Tiz Shelving Filtrenin Ek-1 sunulan MATLAB koduna karşılık çıktısının gösterimi Şekil 4.7 da gösterilmiştir. Kırmızı ile gösterilen gerçek ses , yeşil ile gösterilen filtre sonucunda çıkan sestir.



Şekil 4.6 Bas Shelving Filtrenin MATLAB çıktısı

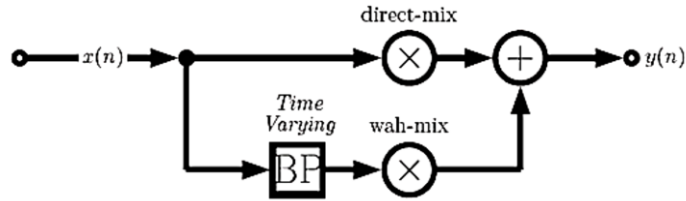


Şekil 4.7 Tiz Shelving Filtrenin MATLAB çıktısı

#### 4.1.2. Zamanla Değişen Filtreler

Bazı ortak etkiler, birkaç farklı yoldan yalnızca bir filtrenin zamana göre değişimi ile gerçekleşir.

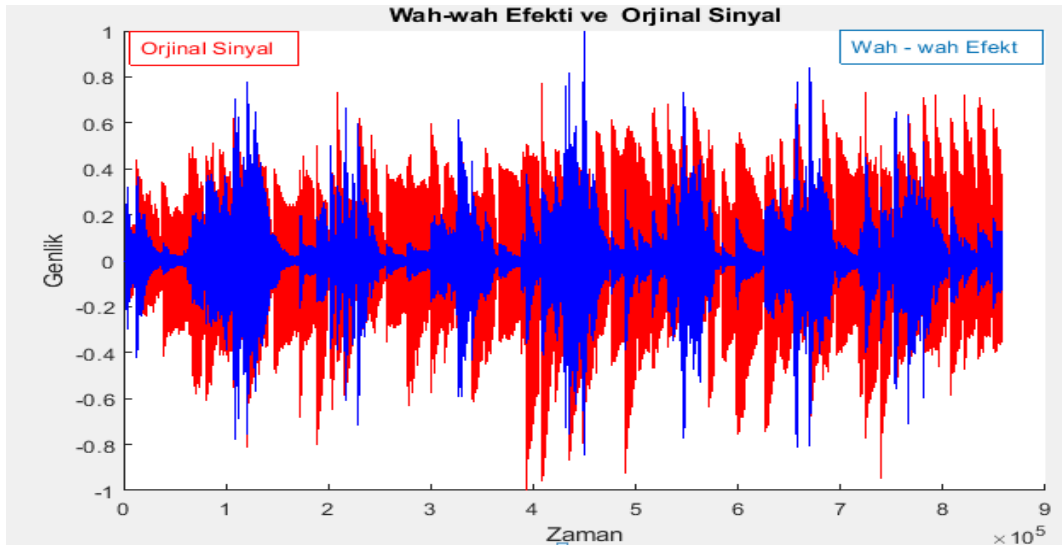
Wah-wah :Zamanla değişen merkez (rezonans) frekansa ve küçük bir bant genişliğine sahip bir bant geçiren filtredir. Doğrudan sinyal ile karıştırılmış filtrelenmiş sinyaldir. Bir wah-wah için sinyal akışı Şekil 4.6 gösterilmiştir.



Şekil 4.6 Bir wah-wah için sinyal akış grafiği

Burada BP zamanla değişen frekans bant geçiren filtredir.

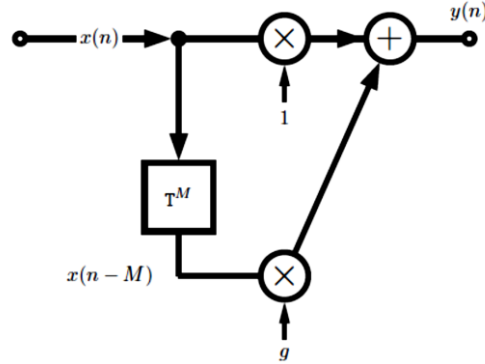
MATLAB kodu Ek-2 de sunulan Wah-wah efektinin uygulama çıktısı Şekil 4.7'de gösterilmiştir. Kırmızı renk ile gösterilen dışardan verdiğimiz gitar solosudur. Mavi renk ise gösterilen wah-wah efekti sonucunda çıkan sinyalin zamanda genlik gösterimidir.



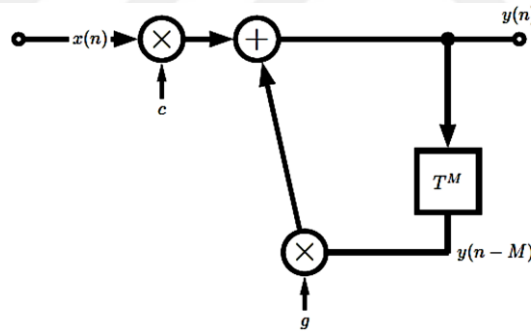
Şekil 4.7 Wah - Wah Efektinin MATLAB çıktısı

### 4.1.3. Gecikme Tabanlı Efektler

Birçok ses efekti, gecikme yapısı kullanılarak uygulanabilir. Vibrato, Flanging, Koro ve Echo gecikme tabanlı efektlerdir. FIR ve IIR filtreleri kullanılır. FIR ve IIR'nin birleşimi Universal Comb Filtresi verir. FIR comb Filtresi Sinyal Akış Diyagramı Şekil 4.7 ve IIR Comb Filtresinin Akış Diyagramı Şekil 4.8 de gösterilmiştir.



Şekil 4.7. FIR Comb Filtresi Sinyal Akış Diyagramı

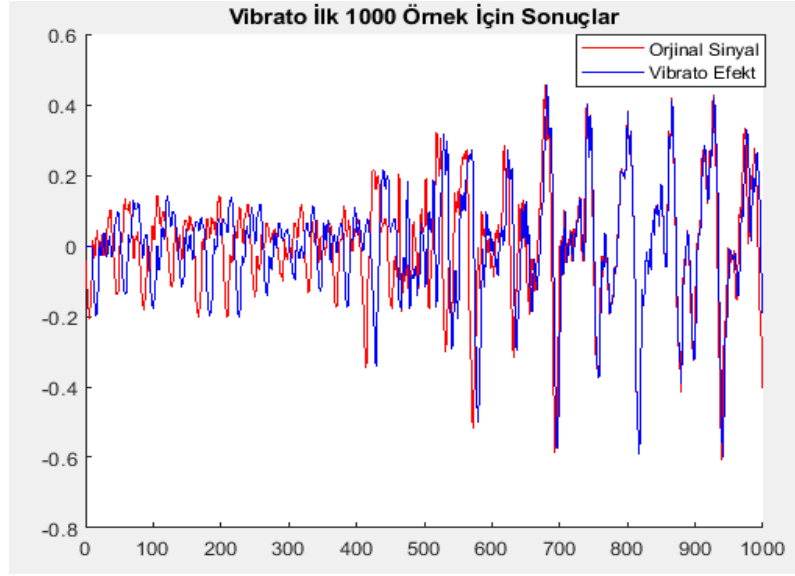


Şekil 4.8. IIR Comb Filter Signal Akış Diyagramı

#### 4.1.3.1. Vibrato

Basit gecikme tabanlı bir efekttir. Vibrato, ses perdelerinde hızlı değişimlerle sağlanan titretmeli efekttir. Bu efekti gerçekleştirmek için periyodik olarak zaman gecikmesini değiştirmemiz gereklidir.

MATLAB kodu Ek-3 de sunulan kodun uygulama sonucunda vibrato efektinin çıktıları Şekil 4.9 da görülmektedir.



Şekil 4.9 Vibrato Efeği MATLAB Çıktısı

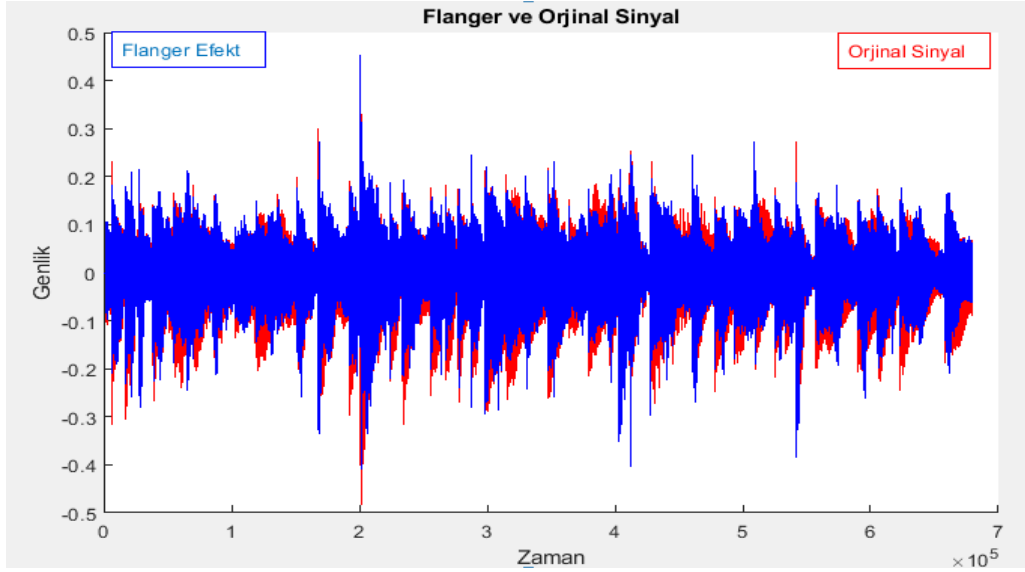
#### 4.1.3.2 Comb Filtresi Gecikmeli Efektler

Bu efektler comb filtresini kullanırlar ve flanger, koro, slapback (sesin hızlı tekrarlanması) ve eko bu gruba girmektedir. Bu efektlerin gecikme aralıkları ve modülasyonları Tablo 4.1 de gösterilmiştir.

Tablo 4.1. Efektlerin Gecikme Aralıkları ve Modülasyonları

Efekt	Gecikme Aralığı (ms)	Modülasyon
Yankılayıcı	0 ... 20	Yok
Flanger	0 ... 15	Sinüsoidal (~ 1 Hz)
Koro	10 ... 25	Rastgele
Slapback	25 ... 50	Yok
Eko	> 50	Yok

Flanger efektinin MATLAB kodu Ek-4 de sunulan Flanger.m kodunun sonucu Şekil 4.10 da gösterilmiştir. Kırmızı orijinal sinyalimiz, mavi ise flanger efekti sonrası oluşan sinyalimizdir.



Şekil 4.10 Flanger Efekt MATLAB çıktısı

## 4.2. MATLAB ile Doğrusal Olmayan Ses İşleme

Doğrusal olmayan işlemciler orijinal sinyalin mevcut değil (kasıtlı veya kasıtsız) harmonik ve inharmonic frekans bileşenleri oluşturarak karakterize eder.

Dinamik İşleme: Harmonik bozulmayı en aza indirmeyi amaçlamaktadır. Örnekler: Kompresörler, Sınırlayıcılar.

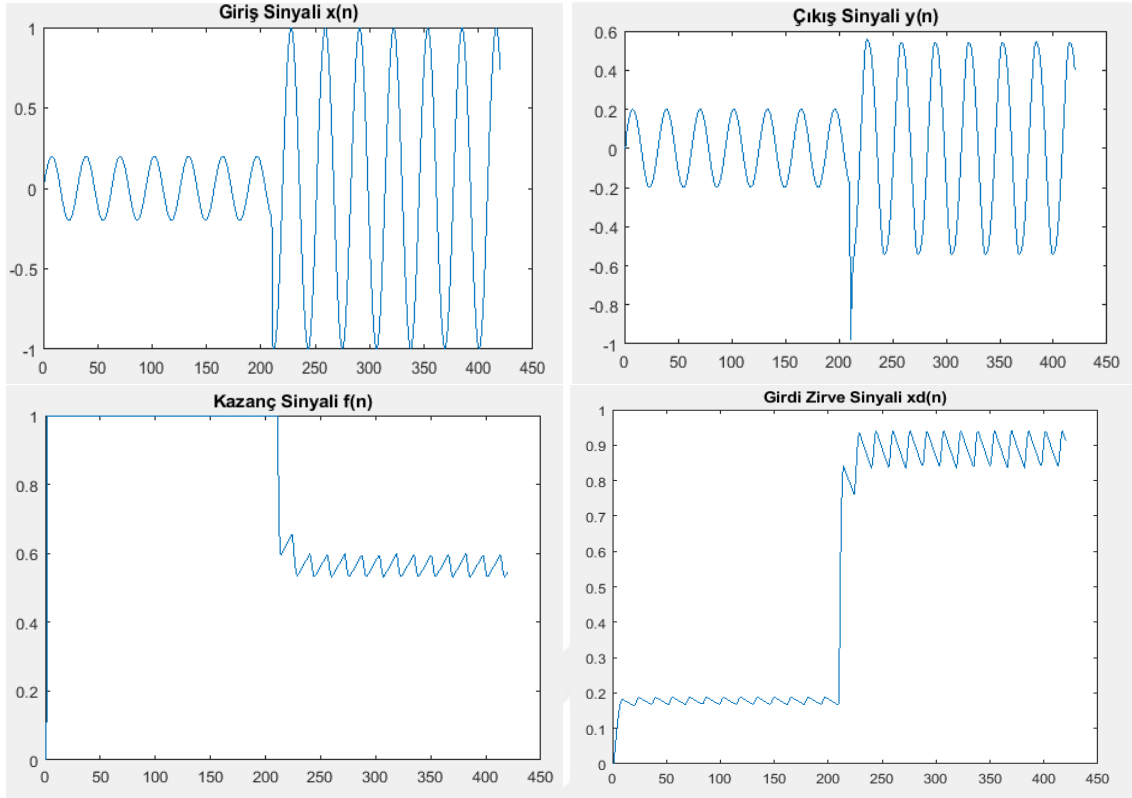
Kasıtlı doğrusal olmayan harmonik işleme: Güçlü harmonik bozulma hedeflenmektedir. Örnekler: Distortion (bozulma) gibi birçok elektro gitar efekti.

### 4.2.1. Sınırlayıcı (Limiter)

Bir sınırlayıcı, bir sinyaldeki yüksek zirveleri kontrol eden bir cihazdır, ancak ana sinyalin dinamiklerini olabildiğince az değiştirmeyi amaçlamaktadır. Bir sınırlayıcı, zirve seviyesi ölçümünü kullanır ve eşğin üstünde olması durumunda seviyeyi ölçeklendirmek için çok hızlı tepki vermeyi amaçlar. Zirveyi düşürerek genel sinyal artırabilir. Limitleme yalnızca tek bir enstrümanda değil aynı zamanda CD mastering (frekansların dengelenmesi), radyo yayını vb. için de kullanılmaktadır.

MATLAB ile oluşturduğumuz sınırlayıcı kodumuzun (Ek-5) çıktıları Şekil 4.11’de verilmiştir. Giriş sinyalini , çıkış sinyalini , kazanç sinyalini ve giriş için zirve olan sinyallerin çıktıları gösterilmiştir.





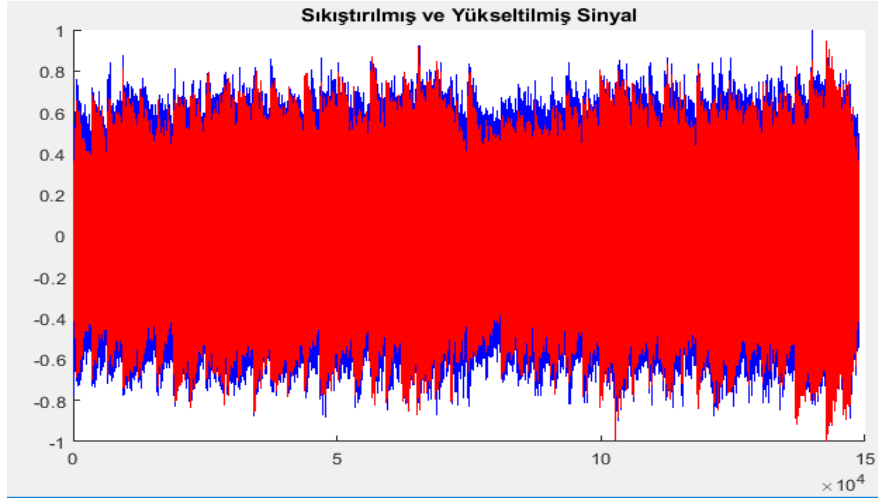
Şekil 4.11 Sınırlayıcının MATLAB Çıktıları

#### 4.2.2. Kompresörler ve Genişleticiler (Compressor ve Expander )

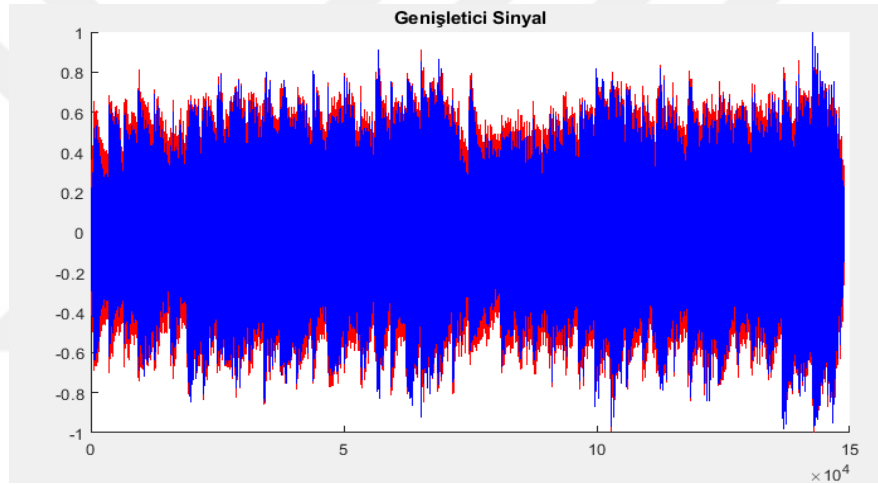
Özünde ses seviyesini dengeleyen bir alettir. Eğer gitar çalarken basslar çok yüksek tizler çok alçak geliyorsa bu dengeleme işini kompresörler üstlenir.

Kompresörler, giriş sinyalinin dinamiklerini azaltmak için kullanılır. Yüksek sesli parçaları, statik eğilime göre azaltılır. Vokallerde ve gitar efektlerinde kullanılırlar. Genişleticiler düşük sinyal seviyelerinde çalışırlar ve sinyallerin dinamikliklerini artırır. Daha canlı bir ses karakteristiği oluşturmak için kullanılır.

MATLAB ortamında çalıştırdığımız kodumuzun (Ek-6) kompresörün (compressor) ve genişleticinin (expander) sonuçları Şekil 4.12 ve Şekil 4.13’de gösterilmiştir. Kırmızı orjinal ses yani giriş sinyalidir. Mavi renk ise kompresör ve genişletici sonucunda çıkan sinyallerdir.



Şekil 4.12 Kompresörün MATLAB Çıktısı



Şekil 4.13 Genişleticinin MATLAB Çıktısı

### 4.3. Yankılanma ve Mekansal Efektler

#### 4.3.1. Yatay Kaydırma (Panning)

Yatay kaydırma, stereo veya çok kanallı bir ses alanında mono bir sinyalin yayılmasıdır. Panning, enstrümanlar merkezini sola ve sağa kaydırarak bir karışımda boşluk ekler.

Monofonik bir ses kaynağının bir stereo ses görüntüsü üzerinden haritalandırılması ve böylece ses bir hoparlör (R) içinde başlayıp n adım adımlarında diğer hoparlöre (L) taşınmaktadır.

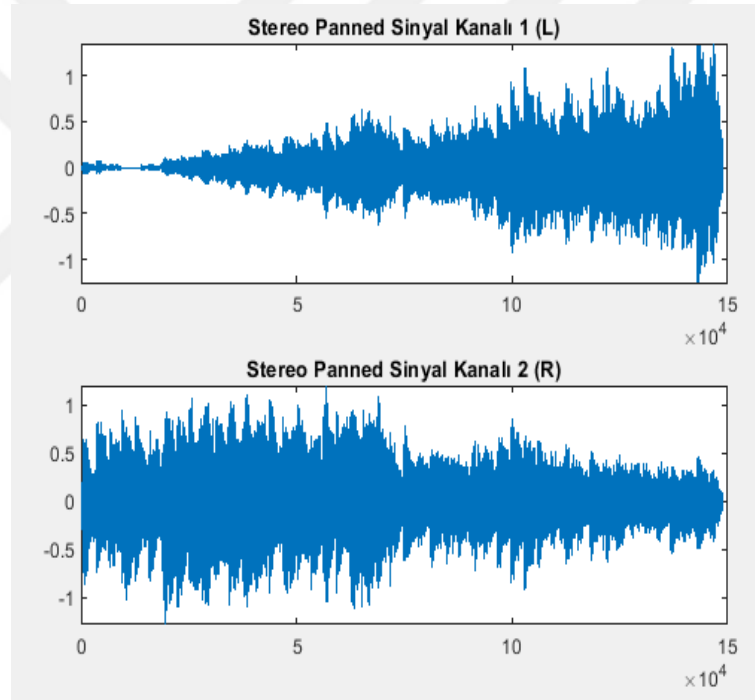
Basit bir 2D döndürme ile elde edilebilir; süpürme açısı =  $\theta$

$$A_{\theta} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad (4.3)$$

$$\begin{bmatrix} g_L \\ g_R \end{bmatrix} = A_{\theta} \cdot x \quad (4.4)$$

Burada  $x$ , mono sesin bir parçasıdır.

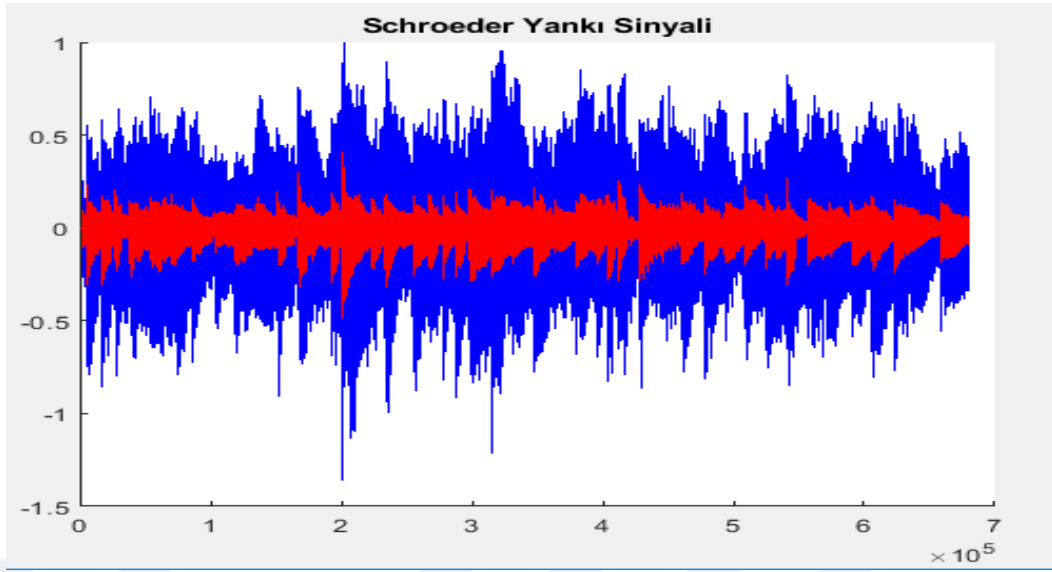
MATLAB ortamında kodlanarak ( Ek- 7 ) modellenmesi sonucunda çıkan yatay kaydırma (panning) sonuçları aşağıda Şekil 4.14’de gösterilmiştir.



Şekil 4.14 MATLAB Panning Sonuçları

### 4.3.2. Schroeder Yankılayıcı

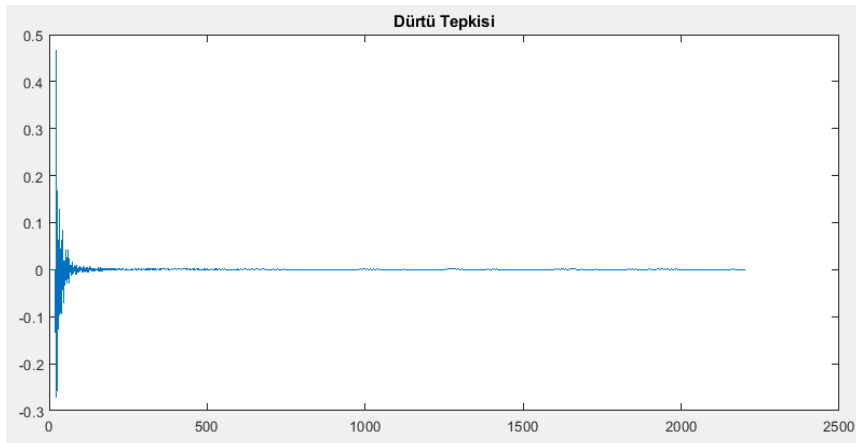
Üçüncü bölümde yankılama kısmında detaylıca bahsettiğimiz Schoreder yankılayıcının MATLAB ortamında kodlanması Ek-8 de verilmiştir ve Schroeder yankılayıcısının ses sinyalimize karşı yanıtı Şekil 4.15’de gösterilmiştir. Kırmızı renk orijinal sinyalimizdir. Mavi renk ise Shroeder yankılayıcı sonucunda çıkan sinyalimizdir.



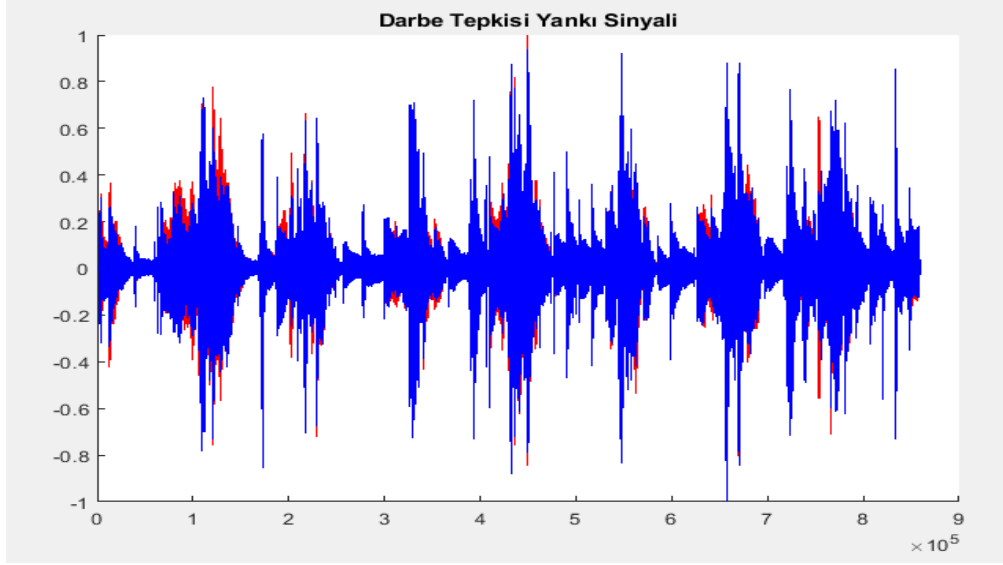
Şekil 4.15. Schroeder Yankısının MATLAB Sonuçları

### 4.3.3. Konvolüsyon Yankılanma

MATLAB ortamında kodlanması Ek-9 da gösterilen, salon dürtü tepkisine (Şekil 4.16) ses sinyalimizin yanıtı Şekil 4.16'da gösterilmiştir. Salon ortamında çalınıyormuş gibi etki sağlar.



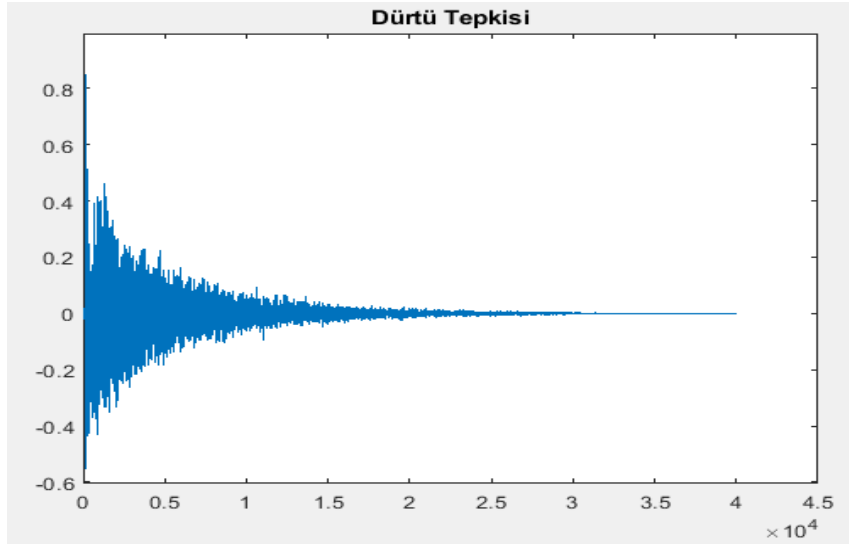
Şekil 4.16. Salon Dürtüsü



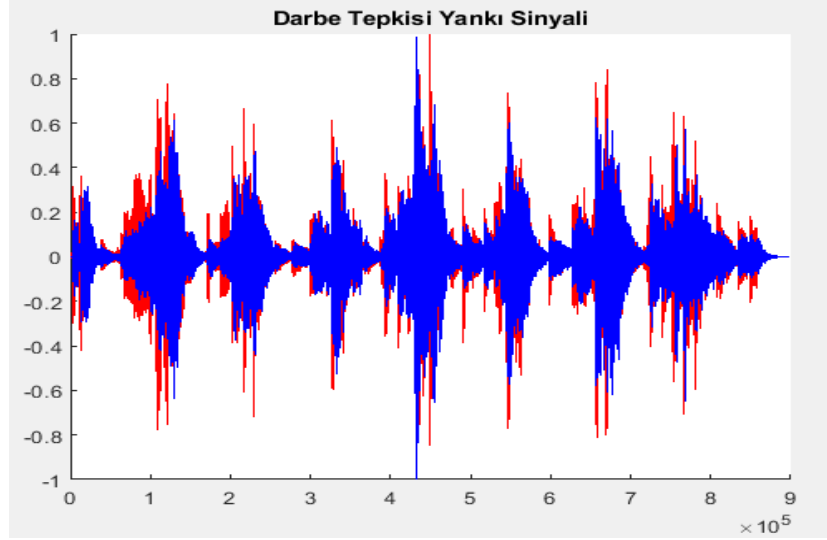
Şekil 4.17. Salon Dürtüsüne göre Yankılanmanın MATLAB Çıktısı

Burada mavi renk ile gösterilen yankılanmanın çıktı sinyalidir. Kırmızı renk ise orijinal sinyalimizdir.

MATLAB ortamında kodlanması Ek-9 da gösterilen, Katedral dürtü tepkisine (Şekil 4.18) ses sinyalimizin yanıtı Şekil 4.19'da gösterilmiştir. Katedralde çalınıyormuş gibi yankı etkisi sağlar.



Şekil 4.18. Katedral Dürtüsü

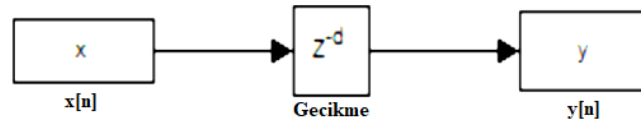


Şekil 4.19. Katedral Dürtüsüne göre Yankılanmanın MATLAB Çıktısı

#### 4.4. MATLAB Simulink İle Gerçek Zamanlı Ses İşleme

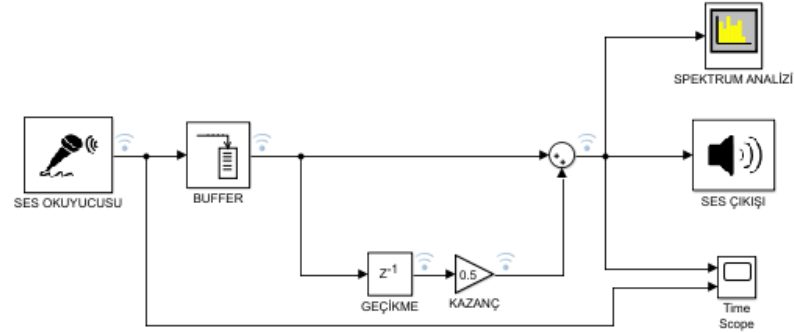
##### 4.4.1. Gecikme (Delay)

Sinyal işleme uygulamalarında, gecikme beklenildiği gibi zaman ve frekans etkilere sahip en sık kullanılan yapı taşıdır. Matematiksel olarak gecikme ses efekti bir tamsayı  $d$  miktarı zaman alanında ayrık zamanlı sinyal değiştirme ile uygulanabilir. (Şekil 4.20)



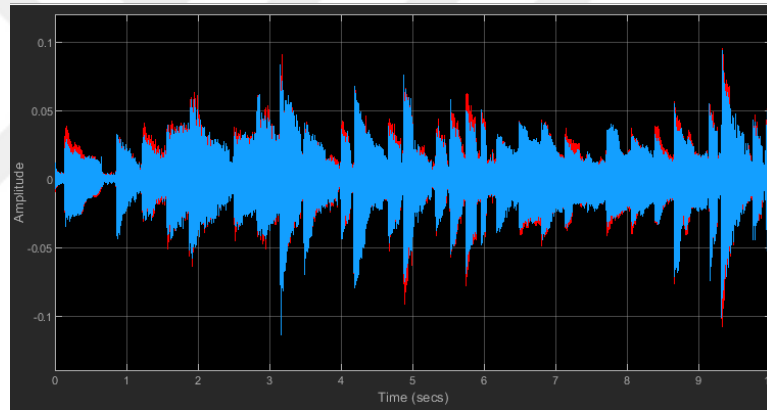
Şekil 4.20 Gecikme Blok Diyagramı

MATLAB Simulink ortamında gecikme diyagramı Şekil 4.21’de gösterilmiştir.



Şekil 4.21 MATLAB Simulink Ortamında Gecikme

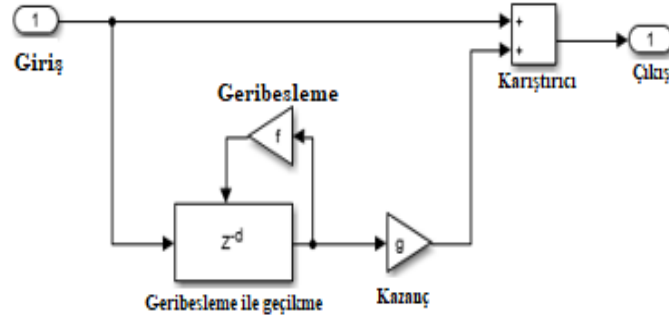
Simulink ortamında gerçek zamanlı ses işleme gecikmesinin zaman bazında genlik çıktısı Şekil 4.22 de gösterilmiştir.



Şekil 4.22 MATLAB Simulink Ortamında Gecikme Çıktısı

#### 4.4.2. Eko Efeito

Ses sinyalini geciktirerek ve tekrar ekleyerek eko efektini modelleyebiliriz. Bir geciktirme efekti vermek için gecikme hattına genellikle geri bildirim eklenmektedir. Eko efektinin blok şeması Şekil 4.23’de gösterilmiştir. Blok şeması, bir yankı efektinin yüksek seviyeli bir uygulamasını göstermektedir.



Şekil 4.23. Eko Efekt Blok Şeması

Eko efekti örneğinin, simülasyon çalışırken modifiye edilebilen dört ayarlanabilir parametresi vardır:

Gecikme: Saniyede ses sinyaline uygulanan gecikme

Kazanç: Gecikmiş sesin doğrusal kazancı

Geri besleme seviyesi: Gecikme hattına geri besleme kazancının uygulanması

Karıştırıcı: Orijinal sinyale eklenen işlenmiş sinyalin oranı

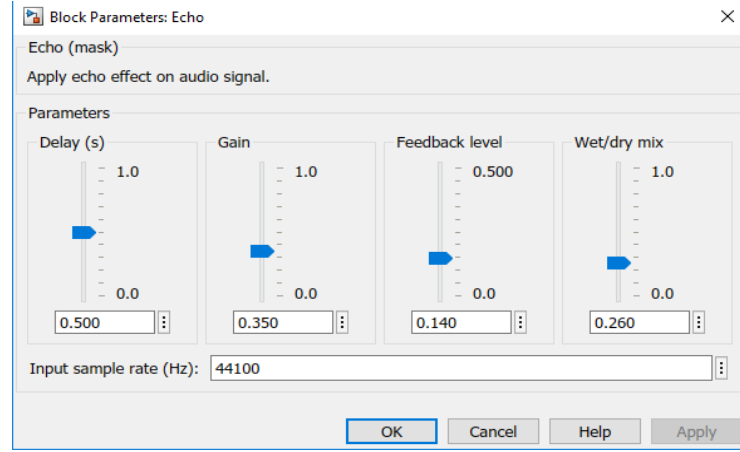
MATLAB Simulink ile oluşturduğumuz projenin görüntüsü Şekil 4.24’de gösterilmiştir.



Şekil 4.24. Eko Efektinin Simulink Modeli

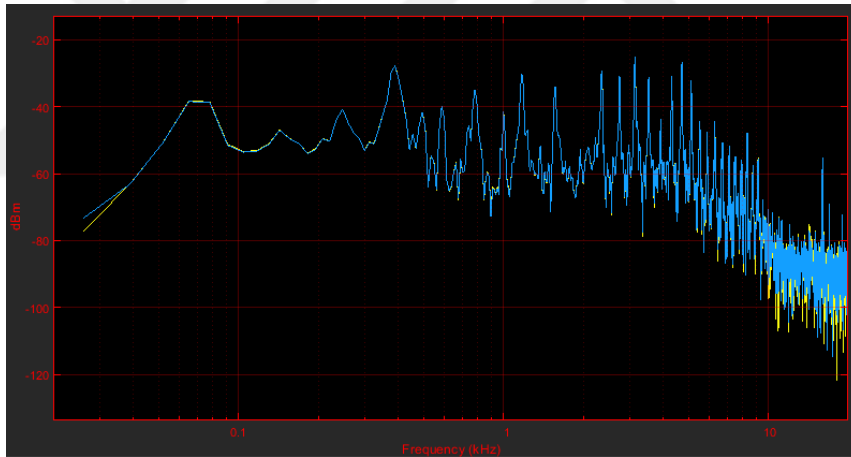
Eko efektinin Şekil 4.23 de gördüğümüz blok diyagramındaki gecikme, kazanç, geri besleme seviyesi, orijinal ve işlenmiş sinyallerin karıştırılması gibi parametrelerini Simulink ortamında projemiz çalışırken de değiştirebiliyor ve istediğimiz parametre ayarını gerçekleştirebiliyoruz (Şekil 4.25).





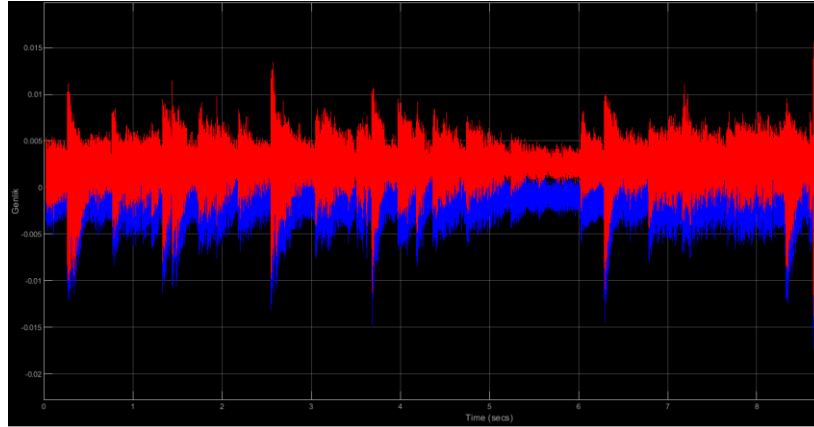
Şekil 4.25. Eko Efekti Parametreleri

Mikrofondan verdiğimiz gitar solosuna karşı gerçek zamanlı ses işleme simulinkinin spektrum analize göre çıktısı Şekil 4.26’da gösterilmiştir.



Şekil 4.26. Eko Efektinin Spektrum Analize göre Çıktısı

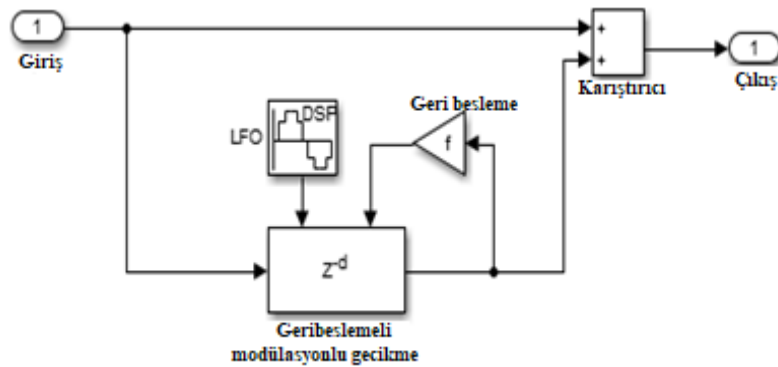
Zaman bazında genlik değişiminin gösterimi ise Şekli 4.27’de gösterilmiştir. Gerçek zamanlı olarak eko parametrelerini değiştirip gitar solosunu istediğimiz eko seviyesine getirebiliriz.



Şekil 4.27. Eko Efektinin Zaman Bazlı Genlik Gösterimi

#### 4.4.3. Flanger Efektı

Ses girişini düşük frekanslı bir osilatör (LFO) tarafından modüle edilen bir miktarda geciktirerek flanger (flanşlama) efektini modelleyebiliriz. Flanger’da kullanılan gecikme çizgisi de bir geri besleme yoluna sahip olabilmektedir. Eko efektinin blok şeması Şekil 4.28’de gösterilmiştir.



Şekil 4.28. Flanger Efektı Blok Şeması

Flanger efekti örneğinde, simülasyon çalışırken modifiye edilebilen beş ayarlanabilir parametre vardır:

Gecikme: Saniyede ses sinyaline uygulanan temel gecikme

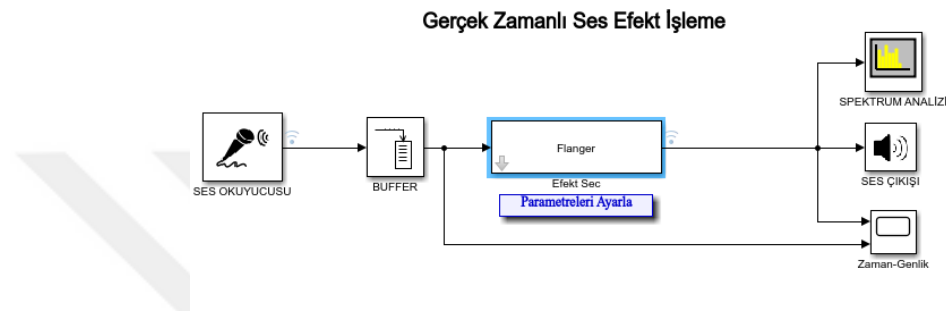
Derinlik: LFO Genliği

Oran: LFO'nun Frekansı, Hz olarak

Geri besleme seviyesi: Gecikme hattına geri besleme kazancının uygulanması

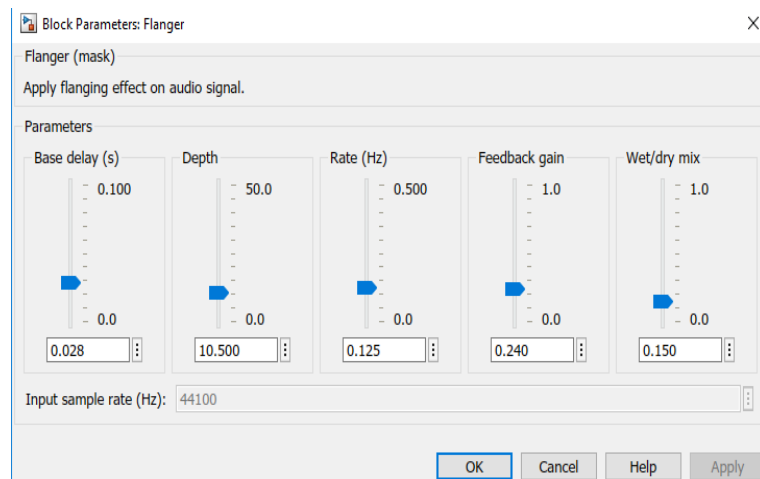
Karıştırıcı: Orijinal sinyale eklenen işlenmiş sinyalin oranı

MATLAB Simulink ile oluşturduğumuz projenin görüntüsü Şekil 4.29'da gösterilmiştir.



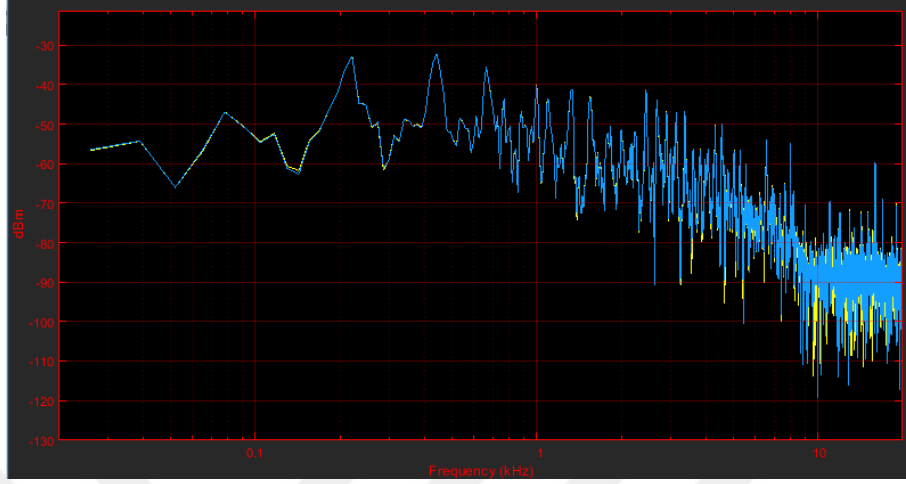
Şekil 4.29. Flanger Efektinin Simulink Modeli

Flanger efektinin Şekil 4.28'de gördüğümüz blok diyagramındaki gecikme, derinlik (LFO genliği), LFO'nun frekansı, geri besleme seviyesi, karıştırıcı gibi parametrelerini değiştirmek için Şekil 4.30'da gösterilen ara yüzü kullanıyoruz. Buradan parametreleri istediğimiz seviyeye çekebiliyoruz.

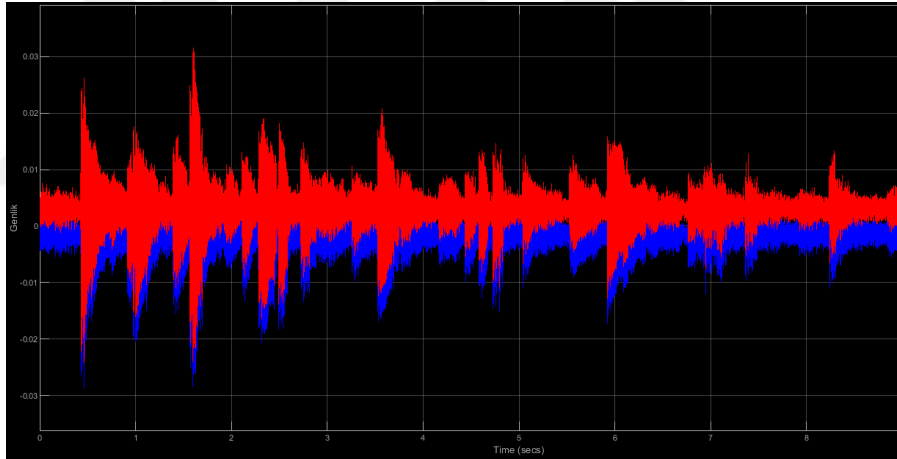


Şekil 4.30. Flanger Efektinin Parametreleri

Mikrofondan verdiğimiz gitar solosuna karşı projenin spektrum analize göre flanger çıktısı Şekil 4.31’de gösterilmiştir. Zaman bazlı genlik gösterimi ise Şekli 4.32’de gösterilmiştir.



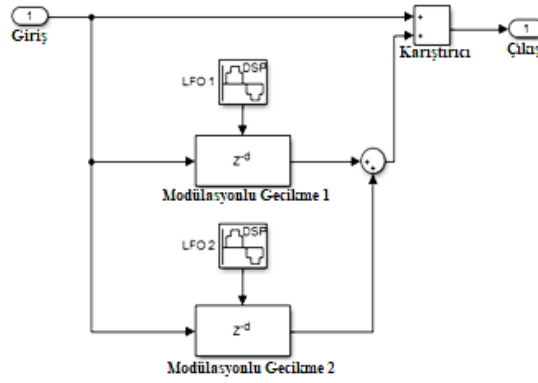
Şekil 4.31. Flanger Efektinin Spectrum Analize göre Çıktısı



Şekil 4.32. Flanger Efektinin Zaman bazlı genlik gösterimi

#### 4.4.4. Koro Efektı

Koro etkisi genellikle her biri düşük frekanslı bir osilatör tarafından modüle edilen çoklu bağımsız gecikmelere sahiptir. Eko efektinin blok diyagramı Şekil 4.33’de gösterilmiştir.



Şekil 4.33. Koro Efektı Blok Diyagramı

Koro efektinin, simülasyon çalışırken modifiye edilebilen altı ayarlanabilir parametresi vardır:

Gecikme: Saniyede ses sinyaline uygulanan temel gecikme

Derinlik 1: İlk gecikme dalına uygulanan modülatörün genliği

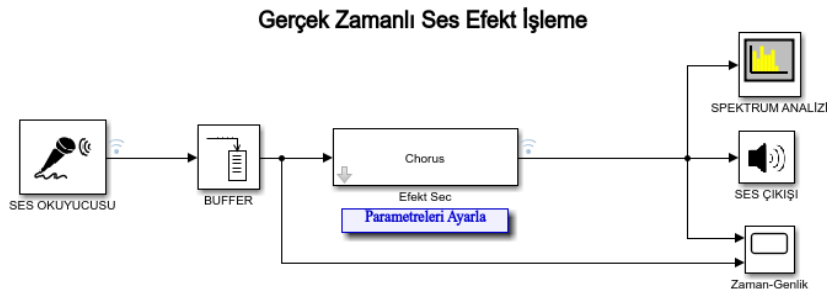
Oran 1: İlk gecikme dalına uygulanan modülatör frekansı, Hz cinsinden

Derinlik 2: İkinci gecikme dalına uygulanan modülatörün genliği

Oran 2: İkinci gecikme dalına uygulanan modülatör frekansı, Hz cinsinden

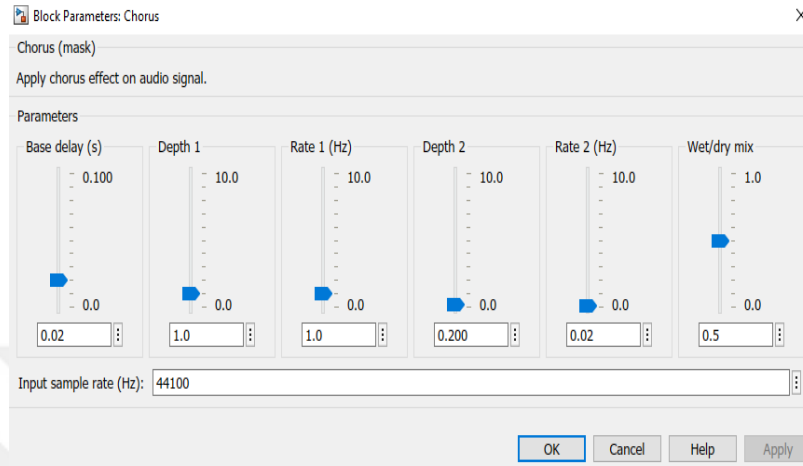
Karıştırıcı: Orijinal sinyale eklenen işlenmiş sinyalin oranı

MATLAB Simulink ile oluşturduğumuz projenin görüntüsü Şekil 4.34'de gösterilmiştir.



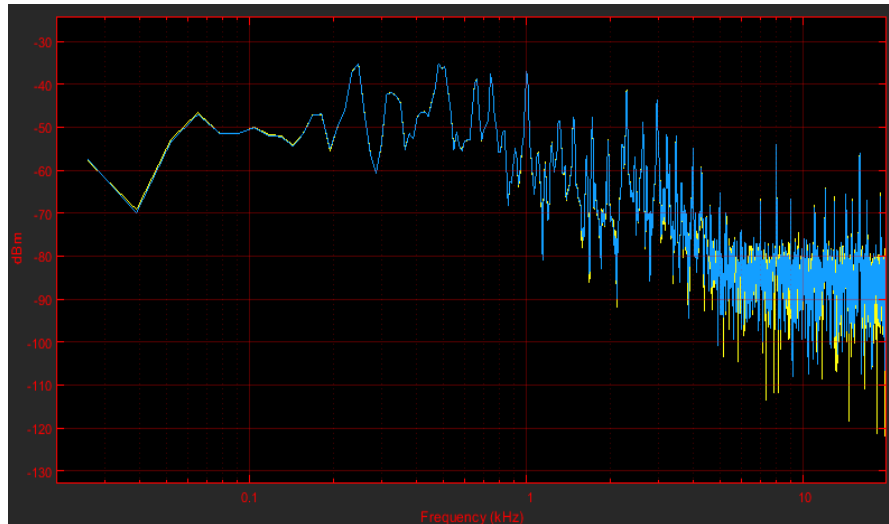
Şekil 4.34. Koro Efektinin Simulink Modeli

Koro efektinin Şekil 4.33’de gördüğümüz blok diyagramındaki parametreleri değiştirmek için Şekil 4.35’de gösterilen ara yüzü kullanıyoruz. Buradan parametreleri istediğimiz seviyeye çekebiliyoruz.



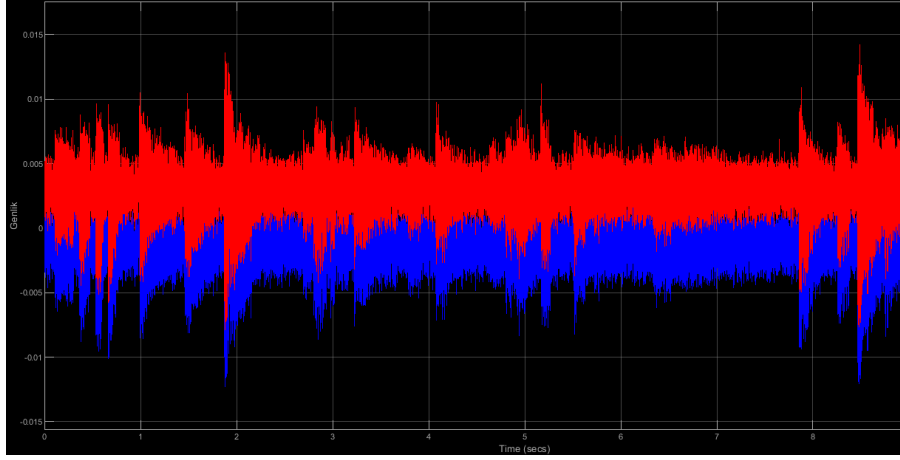
Şekil 4.35. Koro Efektini Parametreleri

Mikrofondan verdiğimiz gitar solosuna karşı projenin spektrum analizine göre koro efekti çıktısı Şekil 4.36’da gösterilmiştir.



Şekil 4.36. Koro Efektinin Spektrum Analize göre çıktısı

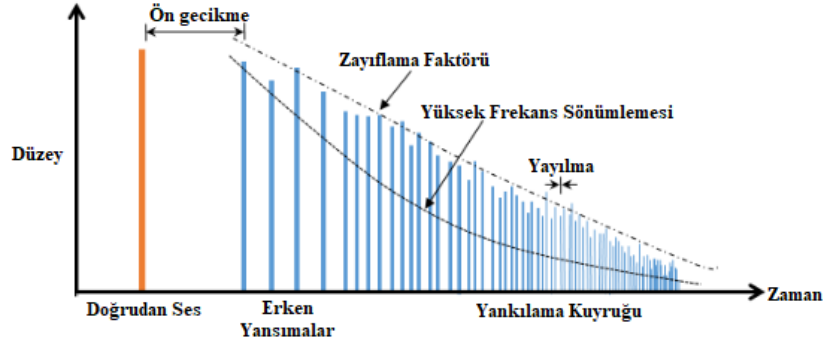
Koro efektinin zaman bazlı genlik gösterimi ise Şekli 4.37'de gösterilmiştir.



Şekil 4.37. Koro Efektinin Zaman Bazlı Genlik Gösterimi

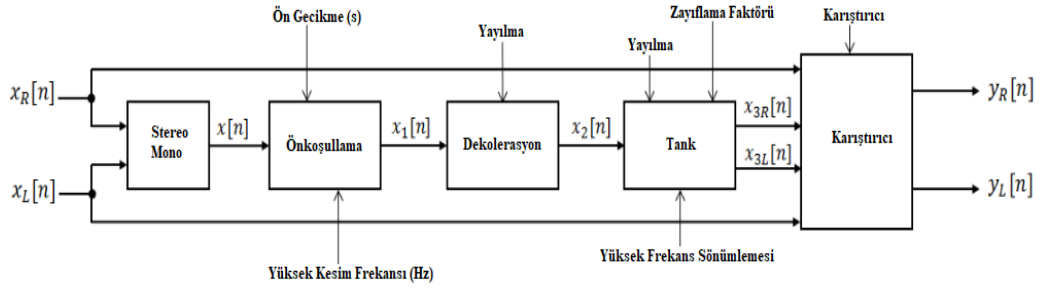
#### 4.4.5. Reverb (Yankı) Efektini

Mono veya stereo ses sinyallerine yankılanma ekler. Yankı efektinin zaman düzey gösterimi Şekil 4.38'de gösterilmiştir.



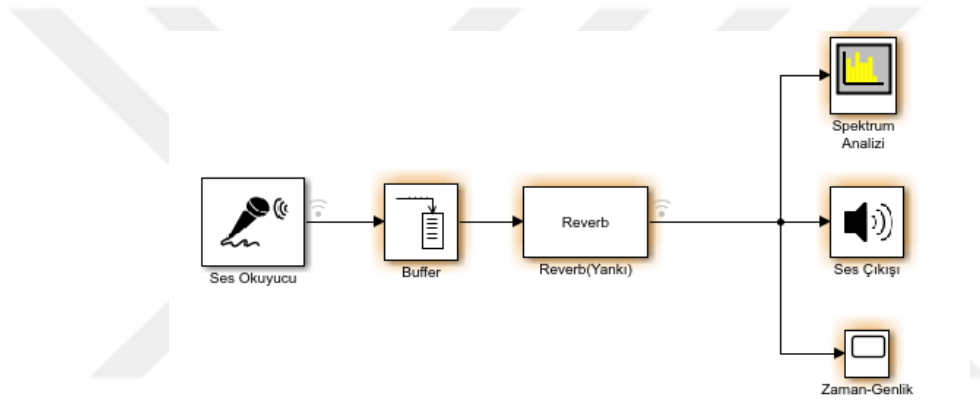
Şekil 4.38. Yankı Efektinin Zaman Bazlı Düzey Gösterimi

Reverberasyon eklemek için kullanılan algoritma, Şekil 4.39'da tarif edilen yankılama topolojisini izler ve 29,761 Hz'lik bir örnekleme hızına dayanır. Algoritmanın beş aşaması vardır.



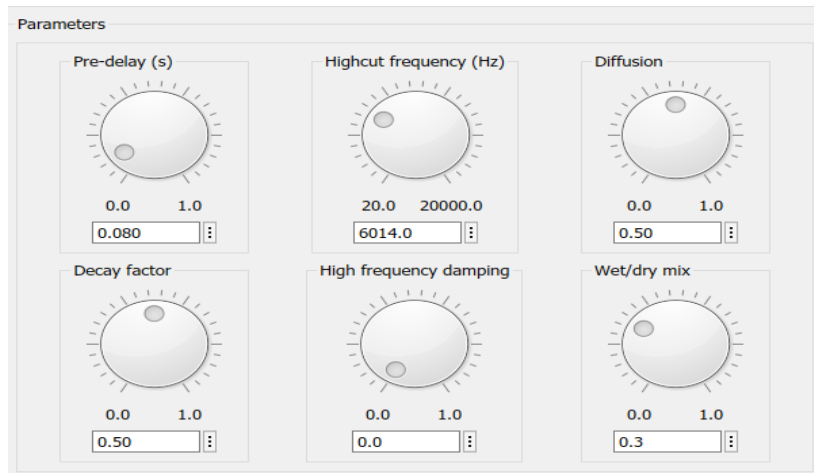
Şekil 4.39. Yankı Etkisi Algoritması Aşamaları

MATLAB Simulink ile oluşturduğumuz Reverb efektinin görüntüsü Şekil 4.40'da gösterilmiştir.



Şekil 4.40. Reverb Etkinin Simulink Modeli

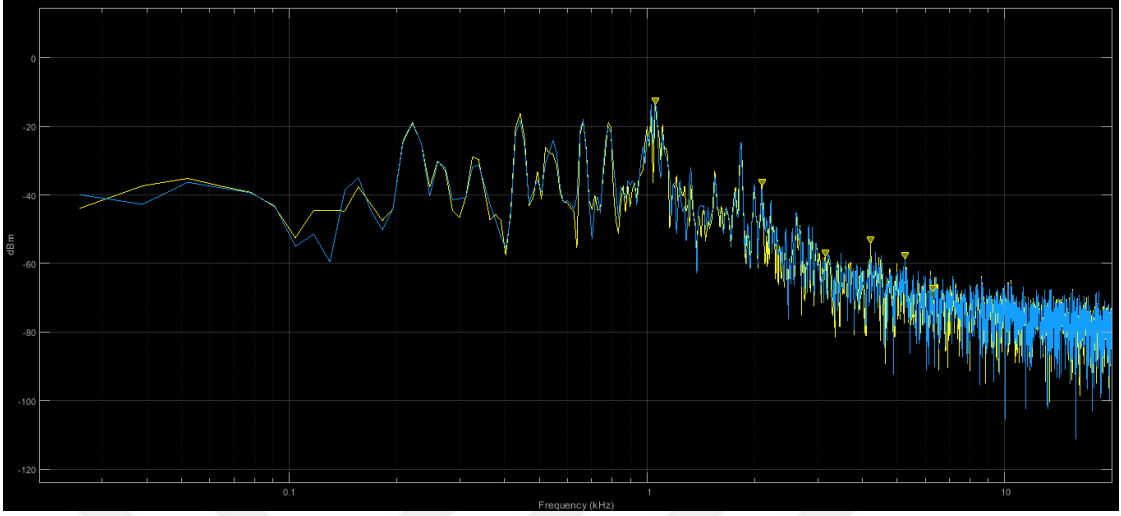
Yankı efektinin Şekil 4.39'da gördüğümüz blok diyagramındaki parametreleri değiştirmek için Şekil 4.41'de gösterilen parametre ara yüzünü kullanıyoruz. Buradan parametreleri istediğimiz seviyeye çekebiliyoruz.



Şekil 4.41. Reverb Etkisi Parametreleri

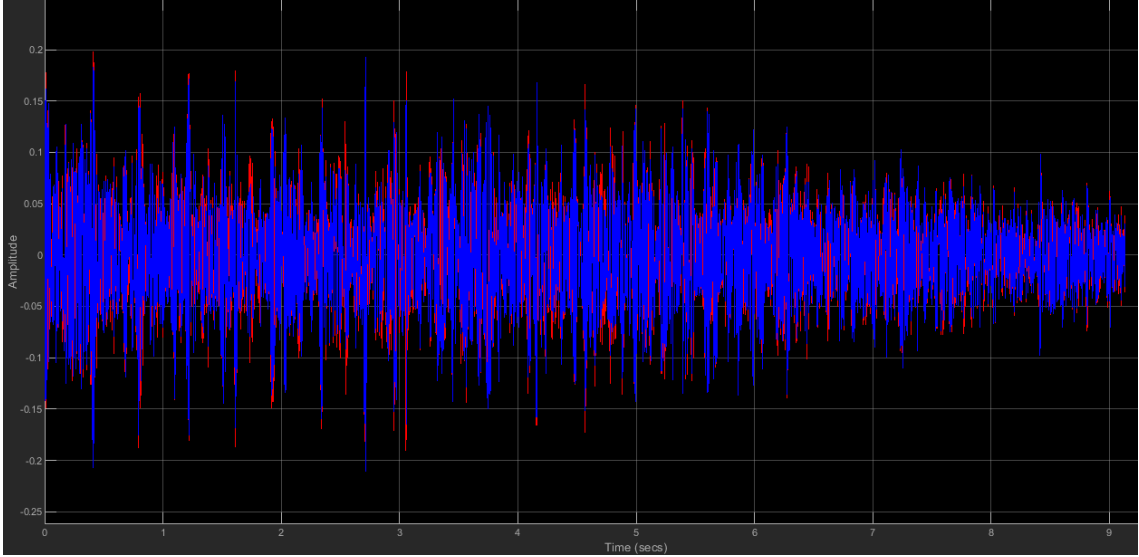


Mikrofondan verdiğimiz gitar solosuna karşı projenin spektrum analizine göre reverb efekti çıktısı Şekil 4.42’de gösterilmiştir.



Şekil 4.42. Reverb Efektinin Spectrum Analize göre çıktısı

Koro efektinin zaman bazlı genlik gösterimi ise Şekli 4.43’de gösterilmiştir.



Şekil 4.43. Reverb Efektinin Zaman Bazlı Genlik Gösterimi

## 5. SONUÇ VE ÖNERİLER

### 5.1. Sonuçlar

Kişisel bilgi işlem ve dijital sinyal işleme biçimlerindeki geliştirmelerle ortaya çıkan, dijital efektlerin, analog efektlere göre başarılı bir şekilde taklit etme, simüle etme ve sentezlemeye ihtiyacı vardır. Bu çalışmada dijital ses efektleri Matlab ortamında incelenmiştir. Ses efektlerinden delay, yankılanma, wah-wah ve flanger efektlerinin modellenmesi gerçekleştirilmiştir. Matlab kullanılarak efektlerin çalışma mantıkları incelenmiş ve doğrulanmıştır. Matlab üzerinden her bir efektin ses sinyali işlenerek çıktıları incelenmiştir.

Matlab uygulamaları ile genel doğrusal gitar efektlerinden olan gecikme, yankı, wah-wah ve flanger gibi efektlerinin gerçek zamanlı olarak Matlab kullanarak modellenmesi gerçekleştirilmiştir. Bu uygulamalar, ses sinyaline girilmek istenilen efektleri teori, tasarım ve uygulamasına odaklanmıştır. Matlab tabanlı tasarım yaklaşımı, ses mühendisinin, önceden tanımlanmış Simulink bloklarının veya efekt parametrelerinin değerlerini değiştirerek istenilen ses efektlerini kolayca yapılandırmasına yardımcı olur. Simulink veya ses efekt parametreleri, istenen efektler için transfer fonksiyonunu şekillendirir ve kontrol etmenize olanak sağlar.

Bu çalışmada, lineer gitar ses efektleri ve doğrusal olmayan gitar ses efektlerinden gecikme, yankı, wah-wah, flanger gibi ses efektleri gerçek zamanlı olarak incelenmiştir. Kişisel bilgi işlem ve dijital sinyal işleme biçimlerindeki geliştirmelerle ortaya çıkan, dijital efektlerin, analog efektlere göre başarılı bir şekilde taklit etme, simüle etme ve sentezlemeye ihtiyacı vardır. Bu simüle etme işini dijital ses işleme teknikleri ve algoritmalarıyla başarılı bir şekilde gerçekleştirebiliyoruz.

Simulink ile çalışmak, teorik ve gerçek zamanlı dijital ses işlemenin öğrenimini kolaylaştırmakta ve aynı zamanda öğrenme sürecini daha keyifli hale getirmektedir. Simulink, filtre tasarımı ve ses efektlerinin gerçek zamanlı olarak işlenmesine olanak sağlamaktadır. Ayrıca analiz için grafikler elde etmemizi sağlamaktadır. Simulink kütüphanesi de bize büyük kolaylık sağlamıştır. Ses sistemleri araç kutusu projemizi oluşturmamızı sağlamıştır. Simulink'in bu özellikleri sinyal işleme ve ses işleme çalışmalarında önemli bir çalışma platformu olduğunu göstermektedir.

Simulink üzerinde gerçek zamanlı olarak eko, flanger ve koro gibi efektler üzerinde çalışılmıştır. Matlab Simulink sayesinde gerçek zamanlı olarak zaman bazlı genlik sonuçlarına ve spektrum üzerinde sinyalin giriş ve çıkış sonuçları gözlemlenmiştir.

## 5.2. Öneriler

Bu çalışma ile ses efektlerinin daha efektif bir şekilde çalışabilmesini sağlanmıştır ve ses efektleri üzerinde yapılacak çalışmalar için bir yol gösterici olacaktır. Çalışma mevcut literatürde Türkçe kaynak eksikliğini giderecek ve ilerde bu alanda yapılacak Türkçe tezlerde önemli bir yol gösterici olacaktır. Çalışmada daha sonra sistem performansını yükseltmek için daha farklı teknikleri kullanılabilir. Farklı ses işleme algoritmaları ve yöntemleri ile bu çalışmadan yola çıkarak performans değerlendirilmesi yapılabilir. Ayrıca bu çalışma günümüzde kullanılan güncel donanımsal programlamalar ile ses efekt üzerine yeni cihaz üretimleri ve daha efektif algoritmalar geliştirilmesine öncü olacaktır.

Genel amaçlı mikroişlemciler, Mikrodenetleyici, Alan Programlanabilir Kapı Dizileri (FPGA'lar), Özel ASIC'ler vb. gibi çok çeşitli işlemcileri kullanarak sinyal işleme yapılabilmektedir. Uygulama alanlarının artan popüleritesi nedeniyle, DSP'e olan ilgi de günümüzde giderek artmaktadır. DSP işlemcileri, üst düzey verimlilik, hız, enerji verimliliği vb. farklı avantajları nedeniyle oldukça popüler hale gelmiştir. DSP'nin analog karşılığı gibi ayırt edici özelliği yazılım kontrolünün kabiliyetidir; Aynı dijital donanımı ancak farklı bir yazılım programında tutmak, çeşitli uygulamalar ile mümkündür. Ayrıca DSP işlemcileri alan programlanabilirdir. Ses efektlerinin dijital ortama dönüşümü gerçekleştirildi. Bu dijital ortamdaki modeller C diline ve donanımsal programlama dillerine dönüştürülebilir. Bu çalışmadan yola çıkarak DSP ile ses efektleri işlenebilir ve optimum sonuçlar için çalışmalar gerçekleştirilebilir.

## KAYNAKLAR

- Millard, A., (2004) "Playing with Poier-The Electric Guitar", A History of an American Icon, 123
- Battenberg, E., & Avizienis, R. (2011, September). Implementing real-time partitioned convolution algorithms on conventional operating systems. In Proceedings of the 14th International Conference on Digital Audio Effects. Paris, France.
- Jot, J. M., & Chaigne, A. (1991) "Digital delay networks for designing artificial reverberator", Audio Engineering Society Convention 90
- Schroeder, M. R., & Logan, B. F. (1961). " Colorless" artificial reverberation. IRE Transactions on Audio, 9(6), 209-214.
- Yeh, D. T., Abel, J. S., & Smith, J. O. (2010). Automated physical modeling of nonlinear audio circuits for real-time audio effects—Part I: Theoretical development. IEEE transactions on audio, speech, and language processing, 18(4), 728-737.
- Zölzer, U. (Ed.). (2011). DAFX: digital audio effects. John Wiley & Sons.
- Smith, S. W. (1997) The scientist and engineer's guide to digital signal processing.
- Mukhopadhyay, M., & Ranjan, O. (2013) "An Intuitive Design Approach For Implementing Real Time Audio Effects".
- Zeki, E. (2015). Digital Modelling Of Guitar Audio Effects (Doctoral Dissertation, Middle East Technical University).
- Prasad, S. R., Gaikwad, P. K., & Joshi, Y. V. (2015). Real Time Special Effects generation and noise filtration of audio signal using Matlab GUI. International Journal on Computational Science & Applications (IJCSA), 5(6), 93-109.
- Oboril, D., Balik, M., Schimmel, J., Smekal, Z., & Krkavec, P. (2000, December). Modelling digital musical effects for signal processors, based on real effect manifestation analysis. In Conference on Digital Audio Effects (pp. 1-6).
- Välimäki, V., Bilbao, S., Smith, J. O., Abel, J. S., Pakarinen, J., & Berners, D. (2011). Virtual analog effects. DAFX: Digital Audio Effects, 473-522.
- Herrera, P., & Bonada, J. (1998, November). Vibrato extraction and parameterization in the spectral modeling synthesis framework. In Proceedings of the Digital Audio Effects Workshop (DAFX98) (Vol. 99).
- Nordström, J. (2017). Real time digital signal processing using Matlab.
- Hudson, M. (2012), "The Design And Realisation Of An Fpga Based Audio Processor"

- Kamen, E., & Heck, B. S. (2007). *Fundamentals of Signals and Systems Using the Web and MatLab: AND Mathworks, MATLAB Sim SV 07*. Prentice Hall Press.
- Porat, B. (1996). *A course in digital signal processing*. John Wiley & Sons, Inc..
- Kuo, S. M., Lee, B. H., & Tian, W. (2013). *Real-time digital signal processing: fundamentals, implementations and applications*. John Wiley & Sons.
- Pakarinen, J., & Yeh, D. T. (2009). A review of digital techniques for modeling vacuum-tube guitar amplifiers. *Computer Music Journal*, 33(2), 85-100.
- Stark, A. M., Davies, M. E., & Plumbley, M. D. (2009, September). Real-time beat-synchronous analysis of musical audio. In *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy* (pp. 299-304).
- Dattorro, J. (1997). Effect design, part 1: Reverberator and other filters. *Journal of the Audio Engineering Society*, 45(9), 660-684.
- Pirkle, W. (2012). *Designing Audio Effect Plug-Ins in C++: With Digital Audio Signal Processing Theory*. Focal Press.
- Novak, A., Simon, L., Lotton, P., & Gilbert, J. (2010, September). Chebyshev model and synchronized swept sine method in nonlinear audio effect modeling. In *Proc. 13th Int. Conference on Digital Audio Effects (DAFx-10)*.
- Byun, K., Kwon, Y. S., Koo, B. T., Eum, N. W., Jeong, K. H., & Koo, J. E. (2009, June). Implementation of digital audio effect SoC. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on* (pp. 1194-1197). IEEE.
- Brandtsegg, Ø., Reiss, J., & Sarkar, S. (2017). Investigation of a Drum Controlled Cross-Adaptive Audio Effect for Live Performance.
- Eichas, F., Fink, M., & Zölzer, U. (2015, November). Feature design for the classification of audio effect units by input/output measurements. In *Proc. 18th Int. Conf. Digital Audio Effects (DAFx-15), Trondheim, Norway* (pp. 27-33).
- Welch, T. B., Wright, C. H., & Morrow, M. G. (2016). *Real-time digital signal processing from MATLAB to C with the TMS320C6x DSPs*. CRC Press.
- Giannakopoulos, T., & Pirkakis, A. (2014). *Introduction to audio analysis: a MATLAB® approach*. Academic Press.
- Fliege, N. J. (1994). *Multirate digital signal processing (Vol. 994)*. New York: John Wiley.
- Bacon, T. ,(2007), 'Guitar', Grove Music Online ed. L. Macy
- Picarella, J. ,2007 Grove Music Online ed. L. Macy

## EKLER

### EK-1 Shelving Filtresinin MATLAB Fonksiyon ve Çalıştırma Kodu

```

function [b, a] = shelving(G, fc, fs, Q, type)
%
% Verilen bir genlik ve kesme frekansı olan bir shelving filtresi için
% türetme katsayıları.
% Zolzer'ın DAFX kitabı (s. 50-55).

% Kullanımı: [B,A] = shelving(G, Fc, Fs, Q, type);
%
% G logarithmic kazancıdır (dB cinsinden)
% FC merkez frekanstır
% Fs örnekleme oranıdır
% Q eğimi sqrt (2) teriminin yerine geçecek şekilde ayarlar
% tipi, filtre türünü tanımlayan bir karakter dizisidir
% Seçenekler şunlardır: 'Base_Shelf' veya 'Treble_Shelf'
%
% Yazar: Jeff Tackett 08/22/05
%

%Hata Kontrolü
if((strcmp(type,'Base_Shelf') ~= 1) && (strcmp(type,'Treble_Shelf') ~=
1))
    error(['Unsupported Filter Type: ' type]);
end

K = tan((pi * fc)/fs);
V0 = 10^(G/20);
root2 = 1/Q; %sqrt(2)

% Kesim durumunda kazancı ters çevir
if(V0 < 1)
    V0 = 1/V0;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   BASE BOOST
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(( G > 0 ) && (strcmp(type,'Base_Shelf')))

    b0 = (1 + sqrt(V0)*root2*K + V0*K^2) / (1 + root2*K + K^2);
    b1 = (2 * (V0*K^2 - 1) ) / (1 + root2*K + K^2);
    b2 = (1 - sqrt(V0)*root2*K + V0*K^2) / (1 + root2*K + K^2);
    a1 = (2 * (K^2 - 1) ) / (1 + root2*K + K^2);
    a2 = (1 - root2*K + K^2) / (1 + root2*K + K^2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   BASE CUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif (( G < 0 ) && (strcmp(type,'Base_Shelf')))

    b0 = (1 + root2*K + K^2) / (1 + root2*sqrt(V0)*K +
V0*K^2);
    b1 = (2 * (K^2 - 1) ) / (1 + root2*sqrt(V0)*K +
V0*K^2);

```

```

    b2 = (1 - root2*K + K^2) / (1 + root2*sqrt(V0)*K +
V0*K^2);
    a1 = (2 * (V0*K^2 - 1) ) / (1 + root2*sqrt(V0)*K +
V0*K^2);
    a2 = (1 - root2*sqrt(V0)*K + V0*K^2) / (1 + root2*sqrt(V0)*K +
V0*K^2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   TREBLE BOOST
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif (( G > 0 )&& (strcmp(type,'Treble_Shelf')))

    b0 = (V0 + root2*sqrt(V0)*K + K^2) / (1 + root2*K + K^2);
    b1 = (2 * (K^2 - V0) ) / (1 + root2*K + K^2);
    b2 = (V0 - root2*sqrt(V0)*K + K^2) / (1 + root2*K + K^2);
    a1 = (2 * (K^2 - 1) ) / (1 + root2*K + K^2);
    a2 = (1 - root2*K + K^2) / (1 + root2*K + K^2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   TREBLE CUT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

elseif (( G < 0 )&& (strcmp(type,'Treble_Shelf')))

    b0 = (1 + root2*K + K^2) / (V0 + root2*sqrt(V0)*K +
K^2);
    b1 = (2 * (K^2 - 1) ) / (V0 + root2*sqrt(V0)*K +
K^2);
    b2 = (1 - root2*K + K^2) / (V0 + root2*sqrt(V0)*K +
K^2);
    a1 = (2 * ((K^2)/V0 - 1) ) / (1 + root2/sqrt(V0)*K +
(K^2)/V0);
    a2 = (1 - root2/sqrt(V0)*K + (K^2)/V0) / (1 + root2/sqrt(V0)*K +
(K^2)/V0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   All-Pass Filtre
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else
    b0 = V0;
    b1 = 0;
    b2 = 0;
    a1 = 0;
    a2 = 0;
end

%değerleri döndürme
a = [ 1, a1, a2];
b = [ b0, b1, b2];

```

**Shelving Filtresinin Fonksiyon Kullanma Kodu:**

```

infile = 'Gitar_SEN_GELMEZ_OLDUN.wav';

% Wav dosyasını okuma
[ x, Fs ] = audioread(infile);

% Shelving Filter parametreleri

G = 4;
fcb = 300;
Q = 3;
type = 'Base_Shelf';

[b, a] = shelving(G, fcb, Fs, Q, type);

yb = filter(b,a, x);

% Çıkış sinyalini yazma

filename = 'out_wah_sen_gelmez_oldun.wav';
audiowrite(filename,yb,Fs)

% Orjinal sinyali ve filtreden çıkan sinyali aynı plot a çizme

figure(1)
hold on
plot(yb,'b');
plot(x,'r');
title('Bas Shelf Filtreden Geçen Sinyal');

fct = 600;
Q = 3;
type = 'Treble_Shelf';
[b, a] = shelving(G, fct, Fs, Q, type);

yt = filter(b,a, x);

% Tiz Shelf Filtrenin Sonucunu Yazma

filename = 'out_wah_sen_gelmez_treble.wav';
audiowrite(filename,yt,Fs)

figure(2)
hold on
plotWave_YW(1,yt,Fs,'freq');

plot(yt,'g');
plot(x,'r');
title('Tiz Shelf Filtreden Geçen Sinyal');

```



## EK-2 Wah-wah Efektinin MATLAB Kodu

```

% wah_wah.m durum deęiřkeni bant geęiři
% Ronan O'Malley tarafından yazılmıř%
% 2 Ekim 2005
% Dar geęiř bandı ile BP filtresi, Fc spektrum yukarı ve ařaęı salınır

% DAFX bۆlüm 2. bۆlümünden'den alınan Fark denklemi

% y1(n) = F1*yb(n) + y1(n-1)
% yb(n) = F1*yh(n) + yb(n-1)
% yh(n) = x(n) - y1(n-1) - Q1*yb(n-1)
%
% Fc 500 ila 5000 Hz arasında deęiřir
% Saniyede 44100 örnek

infile = 'Gitar_SEN_GELMEZ_OLDUN.wav'

% read in wav sample
[ x, Fs ] = audioread(infile);

%%%%%%%% Efekt Katsayıları %%%%%%%%%%
%%%%%%%%%
%%%
% sۆnۆmleme faktۆrű

% Sۆnۆmleme faktۆrűnű kűçűltmek, geęiř bandını kűçűltmek

damp = 0.05;

% Deęiřken bant geęiren filtrenin minimum ve maksimum merkez kesme
frekansı
minf=500;
maxf=3000;

% wah frekansı, saniyede kaę Hz ile ęevrildięini
Fw = 2000;
%%%%%%%%%
%%%

% ۆrnek bařına merkez frekansta% deęiřim (Hz)
% delta=0.1;
delta = Fw/Fs;

% Saniyede 44100 ۆrnekte 0.1 => saniyede 4,4 kHz Fc kayması anlamına
gelmelidir
% merkez frekans deęerleri űçgen dalga oluřturur.
Fc=minf:delta:maxf;
while(length(Fc) < length(x) )
    Fc= [ Fc (maxf:-delta:minf) ];
    Fc= [ Fc (minf:delta:maxf) ];
end

% Giriř boyutuna űçlű trim
Fc = Fc(1:length(x));

```

```

% fark denklemleri katsayıları
F1 = 2*sin((pi*Fc(1))/Fs); % Fc her değiştiğinde yeniden
hesaplanmalıdır.
Q1 = 2*damp; % Bu geçiş bantlarının boyutunu belirler.

yh=zeros(size(x)); % boş vektörler oluşturur.
yb=zeros(size(x));
yl=zeros(size(x));

% negatif sinyallerin referansını önlemek için ilk örnek
yh(1) = x(1);
yb(1) = F1*yh(1);
yl(1) = F1*yb(1);

% Fark denklemini örneklere uygular
for n=2:length(x)

    yh(n) = x(n) - yl(n-1) - Q1*yb(n-1);
    yb(n) = F1*yh(n) + yb(n-1);
    yl(n) = F1*yb(n) + yl(n-1);

    F1 = 2*sin((pi*Fc(n))/Fs);
end

%normalizasyon

maxyb = max(abs(yb));
yb = yb/maxyb;

% çıktıyı wav dosyasına yaz

filename = 'Gitar_SEN_GELMEZ_OLDUN_çıkış.wav';
audiowrite(filename,yb,Fs)

figure(1)
hold on
plot(x,'r'); xlabel('Zaman'); ylabel('Genlik');
plot(yb,'b'); xlabel('Zaman'); ylabel('Genlik');

% Original Sinyal Kırmızı , Efekt Sonucu Çıkan Sinyal Mavidir.
title(' Wah-wah Efekt i ve Orijinal Sinyal ');

```

### EK-3 Vibrato Efektinin MATLAB Kodu

```

% Vibrato
function y=vibrato(x,SAMPLERATE,Modfreq,Width)

Delay=Width; % sn cinsinden giriş örneğinin temel gecikmesi
DELAY=round(Delay*SAMPLERATE); % örnekte temel gecikme
WIDTH=round(Width*SAMPLERATE); % örnekte modülasyon genişliği
if WIDTH>DELAY
    error(' temel gecikmeden daha büyük gecikme!!!');
end
MODFREQ=Modfreq/SAMPLERATE; % örnekte modülasyon frekansı
LEN=length(x); % # WAV dosyasındaki örneklerin
L=2+DELAY+WIDTH*2; % tüm gecikme uzunluğu
Delayline=zeros(L,1); % gecikme için bellek ayırma
y=zeros(size(x)); % çıkış vektörü için bellek ayırma

for n=1:(LEN-1)
    M=MODFREQ;
    MOD=sin(M*2*pi*n);
    ZEIGER=1+DELAY+WIDTH*MOD;
    i=floor(ZEIGER);
    frac=ZEIGER-i;
    Delayline=[x(n);Delayline(1:L-1)];
    %--- linear interpolasyon (Linear Ekleme) -----
    ---
    y(n,1)=Delayline(i+1)*frac+Delayline(i)*(1-frac);

    %--- Allpass interpolasyonu -----
    %y(n,1)=(Delayline(i+1)+(1-frac)*Delayline(i)-(1-frac)*ya_alt);
    %ya_alt=ya(n,1);
end

infile = 'Gitar_SEN_GELMEZ_OLDUN.wav';

% vibrato örneğinde oku
[ x, Fs ] = audioread(infile);

% vibrato için parametreleri ayarla
Modfreq = 10; %10 Khz
Width = 0.0008; % 0.8 milisaniyeler

% Vibrato Gerçekleştir

yvib = vibrato(x, Fs, Modfreq, Width);

% çıktıyı wav dosyasına yaz
filename = 'out_vibrato_sen_gelmez.wav';
audiowrite(filename,yvib,Fs)

% Orijinal sinyali ve vibrato sinyalini çiz
figure(1)
hold on
plot(x(1:1000), 'r');
plot(yvib(1:1000), 'b');
title('Vibrato İlk 1000 Örnek İçin Sonuçlar');

```

**EK-4 Flanger Efektinin MATLAB Kodu**

```

% Ronan O'Malley
% 5 Ekim 2005

% M 0 - 5 Hz'de 0-3 ms veya 0-15 ms arasında bir gecikme süresiyle tek
bir gecikme oluşturan komut dosyası
% Tamponlama kullanılmadıkça bu sentezlenemez
% Uzatma imkanı:
% - izin verilen aralıkları mükemmelleştirmeli
% - Numunelerde gecikmeyi milisaniyeye dönüştürmek için örnekleme
frekanslı hesaplamalar yapmak (bazı 44.1kHz örneklerine ihtiyaç duyar)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all;

infile = 'Gitar_SEN_GELMEZ_OLDUN.wav';
outfile='out_gitar_SEN_GELMEZ_OLDUN_flanger.wav';

% akım örnek 11kHz yani 0-3 ms 0 - 33 örnek
[ x, Fs ] = audioread(infile);

% Efekti değiştirecek parametreler
max_time_delay=0.003; % Saniyede 3 ms maksimum gecikme
rate=1; %Hz'de flanş (flange) oranı
index=1:length(x);

% sin referans salınan gecikme oluşturmak için
sin_ref = (sin(2*pi*index*(rate/Fs)))'; % sin(2pi*fa/fs);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

max_samp_delay=round(max_time_delay*Fs);
% max gecikme, örneklerde %maksimum gecikme

y = zeros(length(x),1); % boş vektör oluştur

y(1:max_samp_delay)=x(1:max_samp_delay);
% Negatif numunelerin referansını önlemek için

amp=0.7; % önerilen katsayı sayfa 71'den itibaren DAFX

% her örnek için
for i = (max_samp_delay+1):length(x)
    cur_sin=abs(sin_ref(i)); % abs of akım sin val 0-1
    cur_delay=ceil(cur_sin*max_samp_delay);
% 1-max_samp_delay'den gecikme oluştur ve tam sayıyı sağla
y(i) = (amp*x(i)) + amp*(x(i-cur_delay)); %gecikmeli örnek ekle
end

% yazma çıkışı
audiowrite(outfile,y,Fs)
figure(1)
hold on
plot(x,'r');xlabel('Zaman'); ylabel('Genlik');
plot(y,'b');xlabel('Zaman'); ylabel('Genlik');
title('Flanger ve Orjinal Sinyal');

```

**EK-5 Sınırlayıcı (Limiter) MATLAB Kodu**

```

% Limiter.m

clear all;

% Genlikli bir sinüs dalgası yarat yarısı süresi boyunca azalt.

anzah=210;
for n=1:anzah,
    x(n)=0.2*sin(n/5);
end;
for n=anzah+1:2*anzah;
    x(n)=sin(n/5);
end;

% Limitleme Gerçekleştir

slope=1;
tresh=0.5;
rt=0.01;
at=0.4;

xd(1)=0; % x Tepeler
for n=2:2*anzah;
    a=abs(x(n))-xd(n-1);
    if a<0, a=0; end;
    xd(n)=xd(n-1)*(1-rt)+at*a;
    if xd(n)>tresh,
        f(n)=10^(-slope*(log10(xd(n))-log10(tresh)));
        % Lineer hesabi: f=10^(-LS*(X-LT))
    else f(n)=1;
    end;
    y(n)=x(n)*f(n);
end;

figure(1)
plot(x)
title('Giriş Sinyali x(n)');

figure(2)
plot(y)
title('Çıkış Sinyali y(n)');

figure(3)
plot(f)
title('Kazanç Sinyali f(n)');

figure(4)
plot(xd)
title('Girdi Zirve Sinyali xd(n)');

```

## EK-6 Kompresörler ve Genişleticiler MATLAB Kodu

```

function y=compexp(x, comp, a, Fs)
% Kompresörler / Genişleticiler
% comp - sıkıştırma: 0 > comp > -1, genişletme: 0 < comp < 1
% a- filtre parametresi < 1
h=filter([(1-a)^2],[1.0000 -2*a a^2],abs(x));
h=h/max(h);
h=h.^comp;

y=x.*h;
y=y*max(abs(x))/max(abs(y));

% Kompresor.m
clear all;

filename='Gitar_SEN_GELMEZ_OLDUN.wav';

[ x, Fs ] = audioread(filename);
comp = -0.5; % kompresörü ayarla
a = 0.5;
y = compexp(x, comp, a, Fs);

% yazma çıkışı
filenameout='Gitar_SEN_GELMEZ_OLDUN_out.wav';
audiowrite(filenameout,y,Fs);
figure(1);
hold on
plot(y,'b');
plot(x,'r');
title('Sıkıştırılmış ve Yükseltilmiş Sinyal');

%
% Genişletici.m
clear all;

filename='Gitar_SEN_GELMEZ_OLDUN.wav';
outfile='Gitar_SEN_GELMEZ_OLDUN_out.wav';

% örnek dalga formunu oku
[ x, Fs ] = audioread(filename);

comp = 0.5;
a = 0.5;
y = compexp(x, comp, a, Fs);

% Çıktıyı yaz
audiowrite(outfile,y,Fs)

figure(1);
hold on
plot(x,'r'); % Orjinal sinyal kırmızı çiz.
plot(y,'b'); % Çıktı sinyalini mavi çiz.

title('Genişletici Sinyal');

```

**EK-7 Yatay Kaydırma (Panning) MATLAB Kodu**

```

% matpan.m
% DAFZ p 140
% Stereodan mono sese matris tabanlı kaydırma yapmak.

clear all;

% örnek dalga formunu oku

filename='Gitar_SEN_GELMEZ_OLDUN.wav';

[ monox, Fs ] = audioread(filename);
initial_angle = -40; %derece olarak
final_angle = 40;   %derece olarak
segments = 32;
angle_increment = (initial_angle - final_angle)/segments * pi / 180;
                                                    % radyan

lenseg = floor(length(monox)/segments) - 1;
pointer = 1;
angle = initial_angle * pi / 180; % radyan
y=[[[];[]];

for i=1:segments
    A = [cos(angle), sin(angle); -sin(angle), cos(angle)];
    stereox = [monox(pointer:pointer+lenseg)'];
    monox(pointer:pointer+lenseg)';
    y = [y, A * stereox];
    angle = angle + angle_increment; pointer = pointer + lenseg;
end

% yazma çıkışı
filenameout='Gitar_SEN_GELMEZ_OLDUN_out.wav';
audiowrite(filenameout,y,Fs);

figure(1);
hold on
subplot(2,1,1);
plot(y(1,:));
title('Stereo Panned Sinyal Kanalı 1 (L)'); % Sol kanal çıkışı

subplot(2,1,2);
plot(y(2,:));
title('Stereo Panned Sinyal Kanalı 2 (R)'); % Sağ kanal çıkışı

```

**EK-8 Schroeder Yankılayıcı MATLAB Kodu**

```

function [y,b,a]=schroeder1(x,n,g,d,k)
% Schroeder'in tasarımına dayanan ve tüm seri geçiş filtrelerinden
oluşan bir reverberator.
%
Yapısı: [y,b,a] = schroeder1(x,n,g,d,k)
%
%     x = giriş sinyali
%     n = allpass filtrelerinin sayısı
%     g = allpass filtrelerinin kazancı (bu, stabilite için 1'den az
olmalıdır)
%     d = her bir allpass filtresinin gecikme uzunluğunu içeren bir
vektör
%     k = direkt sinyalin kazanç faktörü
%     y = çıkış sinyali
%     b = transfer fonksiyonunun pay katsayıları
%     a = Transfer fonksiyonunun payda katsayıları
%
% not: d'nin n ile aynı uzunlukta olduğundan emin olun.
% Gautham J. Mysore - gauthamjm@yahoo.com
%

% ilk allpass filtresi üzerinden giriş sinyalini göndermek
[y,b,a] = allpass(x,g,d(1));

% Her bir allpass filtresinin çıkışını bir sonraki allpass filtresinin
girişine gönderir
for i = 2:n,
    [y,b1,a1] = allpass(y,g,d(i));
    [b,a] = seriescoefficients(b1,a1,b,a);
end

% Ölçekli direkt sinyali ekle
y = y + k*x;
% çıkış sinyalini normalize etmek
y = y/max(y);

function [y,b,a]=allpass(x,g,d)

% Bu bir allpass filtre işlevidir.
%
%Yapısı: [y,b,a] = allpass(x,g,d)
%
%     x = giriş sinyali
%     g = ileriye dönük kazanç (geri besleme kazancı bunun
negatifidir) (bu, istikrar için 1'den az olmalıdır)
%     d = gecikme uzunluğu
%     y = çıkış sinyali
%     b = transfer fonksiyonunun pay katsayıları
%     a = Transfer fonksiyonunun payda katsayıları
%
% Gautham J. Mysore - gauthamjm@yahoo.com
%
```



```

%Geri bildirim kazancı 1'den fazlaysa, 0,7 olarak ayarlayın.
if g>=1
    g=0.7;
end

%g ve d'ye bağlı olarak b ve transfer fonksiyonunun katsayılarını
ayarlayın.
b=[g zeros(1,d-1) 1];
a=[1 zeros(1,d-1) g];

%giriş sinyalini filtreleyin
y=filter(b,a,x);

filename='gitar_solo.wav';

[ x, Fs ] = audioread(filename);

% allpass filtresinin sayısını ayarla
n = 6;
% allpass filtrelerinin kazancını ayarla
g = 0.9;
%Her bir allpass filtresinin numunelerin sayısında gecikme ayarlanması
%Rastgele bir milisaniye kümesi hesaplayın ve örnek oranını kullanın
rand('state',sum(100*clock))
d = floor(0.05*rand([1,n])*Fs);
% direkt sinyalin kazanımını ayarlamak
k= 0.2;

[y b a] = schroeder1(x,n,g,d,k);

% çıktıyı yaz

filenameout='gitar_solo_out.wav';
audiowrite(filenameout,y,Fs);

figure(1);
hold on
plot(y, 'b');
plot(x, 'r');
title('Schroeder Yankı Sinyali');

```

## EK-9 Konvolüsyon Yankılanma MATLAB Kodu

```

clear all;

% örnek dalga formunu oku
filename='Gitar_SEN_GELMEZ_OLDUN.wav';

[ x, Fs ] = audioread(filename);

% impuls yanıtı dalga formunu oku
filename1='impulse_room.wav';
% filename1='impulse_cathedral.wav';

[ imp, Fsimp ] = audioread(filename1);

% FFT ile konvolüsyon yap
y = fconv(x,imp);

% Çıktı Yaz
filenameout='out_Reverb_salon_sen_gelmez_oldun.wav';
audiowrite(filenameout,y,Fs);

figure(1);
hold on
plot(x,'r');
plot(y,'b');
title('Darbe Tepkisi Yankı Sinyali');

figure(2);

plot(imp);
title('Dürtü Tepkisi');

function [y]=fconv(x, h)
% FCONV konvolüsyon
% [y] = FCONV(x, h) x ve h dönüştürür ve çıkışı normalleştirir
% +-1.
%
% x = giriş vektör
% h = giriş vektör
% Stephen G. McGovern, 2003-2004.

Ly=length(x)+length(h)-1; %
Ly2=pow2(nextpow2(Ly)); % 2'nin en küçük gücünü bul> Ly
X=fft(x, Ly2); % Hızlı Fourier Dönüşümü
H=fft(h, Ly2);
Y=X.*H; %
y=real(ifft(Y, Ly2)); % Ters hızlı Fourier dönüşümü
y=y(1:1:Ly); % Sadece ilk N elemanını al
y=y/max(abs(y)); % Çıkışı normalize et

```

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Enes SULAK  
**Uyruğu** : T.C.  
**Doğum Yeri ve Tarihi** : KONYA/Ereğli - 16.03.1990  
**Telefon** : 0544 3177828  
**e-mail** : enessulak@gmail.com

### EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Selçuklu Atatürk Lisesi, Selçuklu, KONYA	2007
Üniversite	: Fırat Üniversitesi, Merkez, ELAZIĞ	2012
Yüksek Lisans	: Necmettin Erbakan Üniversitesi, Meram, KONYA	

### İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2015-2018	Selçuk Üniversitesi	Bilgi İşlem Destek Elemanı

### YABANCI DİLLER

İngilizce Yazma – Okuma – Konuşma: Orta

### BİLGİSAYAR BİLGİSİ

- MVC
- Asp.Net
- C #
- Entity Framework
- Java
- Vhdl
- Veri Tabanı (MySQL – Oracle)

### KURS ve SEMİNERLER

2013 C Sınıfı İş Sağlığı ve Güvenliği Uzmanlığı ( Selçuksem )  
 2015 Kosgeb Girişimcilik Sertifikası

### YAYINLAR

Multifunction Audio Effect Processing - International Advanced Researches & Engineering Congress 2017 Proceeding Book - Sayfa:1218