



**AKDENİZ ÜNİVERSİTESİ
GÜZEL SANATLAR ENSTİTÜSÜ**

MÜZİK ANASANAT DALI

**YAZILIM TABANLI TINI EFEKTİ ÜRETİMİ ÜZERİNE ÖRNEK
BİR ARAŞTIRMA**

EKİN BARAN YILDIZ

YÜKSEK LİSANS TEZİ

PROF. DR. HASAN ARAPGİRLİOĞLU

ANTALYA, 2023



**AKDENİZ ÜNİVERSİTESİ
GÜZEL SANATLAR ENSTİTÜSÜ**

MÜZİK ANASANAT DALI

**YAZILIM TABANLI TINI EFEKTİ ÜRETİMİ ÜZERİNE ÖRNEK BİR
ARAŞTIRMA**

EKİN BARAN YILDIZ

YÜKSEK LİSANS TEZİ

DANIŞMAN PROF. DR. HASAN ARAPGİRLİOĞLU

ANTALYA, 2023

KABUL VE ONAY SAYFASI

Ekin Baran YILDIZ tarafından hazırlanan “Yazılım Tabanlı Tını Efektü Üretimi Üzerine Örnek Bir Araştırma” başlıklı bu çalışma, jürimiz tarafından Müzik Anasanat Dalı’nda Yüksek Lisans Tezi olarak oybirliği/oyçokluğu ile kabul edilmiştir.
22/06/2023

Jüri Başkanı	Prof. Dr. Atilla Kadri Şendil	
Jüri Üyesi (Danışman)	Prof. Dr. Hasan Arapgirlioğlu	
Jüri Üyesi	Dr. Bekir Tanyeri	

Bu çalışma, Akdeniz Üniversitesi Lisansüstü Eğitim-Öğretim Yönetmeliği’nin ilgili maddeleri uyarınca yukarıdaki jüri tarafından uygun bulunmuştur.

.....Güzel Sanatlar Enstitüsü Müdürü

TEŞEKKÜR

Bu çalışmanın bütün hazırlanma aşamasında bilgi, birikim ve desteğini hiç esirgemeyen danışman hocam Prof. Dr. Hasan ARAPGİRLİOĞLU'na teşekkürlerimi sunarım.



T.C.
AKDENİZ ÜNİVERSİTESİ
GÜZEL SANATLAR ENSTİTÜSÜ

Öğrencinin	Adı Soyadı	Ekin Baran YILDIZ
	Numarası	202053011003
	Anasanat Dalı	Müzik
	Danışmanı	Prof. Dr. Hasan ARAPGİRLİOĞLU
	Tezin Adı	Yazılım Tabanlı Tını Efekti Üretimi Üzerine Örnek Bir Araştırma

ÖZ

Bu tez çalışması, dijital tını efektlerinin bilgisayar dili ve yazılım programları kullanılarak nasıl oluşturulabileceğine dair bir çalışma ortaya koymaktadır. Çalışmanın amacı, müzik endüstrisindeki teknolojik gelişmelerin kullanımını ele alarak, yazılım tabanlı efekt oluşturulması konusunda bir örnek sunmaktadır.

Çalışmanın yöntemi, öncelikle hangi efektin elde edileceğinin belirlenmesi ve buna göre hangi ses dosyalarının kullanılacağına tespit edilmesiyle başlamaktadır. Ardından, Matlab programı kullanılarak ses dosyalarının işlenmesi gerçekleştirilmiştir ve bu işlemlerin Matlab içerisindeki grafiksel analizleri ortaya konulmuştur. Çalışma kapsamında farklı efekt türleri incelenerek, bunların nasıl oluşturulabileceğine dair yöntemler sunulmuştur.

Bu çalışmanın sonucunda, dijital tını efektlerinin üretiminde kullanılan yazılımların ve sinyal işleme tekniklerinin müzik prodüksiyonunda önemli bir rol oynayabileceği gösterilmiştir. Ayrıca, çalışmanın yöntemi ve sonuçları, dijital tını efektleri konusunda ilerlemek isteyen araştırmacılara yol gösterici olabilir.

Anahtar Sözcükler: Yazılım, Matlab, Simulink, Yankı, Gecikme, Sinyal, Audio Toolbox

T.R.
AKDENİZ UNIVERSITY
INSTITUTE OF FINE ARTS

Student' s	Name Surname	Ekin Baran YILDIZ
	Number	202053011003
	Department	Music
	Supervisor	Prof. Dr. Hasan ARAPGİRLİOĞLU
	Thesis Title	A Sample Research on Software Based Timbre Effect Generation

ABSTRACT

This thesis presents a study on how digital timbre effects can be created using computer language and software programs. The aim of the study is to examine the use of technological developments in the music industry and provide an example of software-based effect creation.

The method of the study begins by determining which effect is to be obtained and identifying which sound files will be used accordingly. Then, the sound files are processed using the Matlab program, and the graphical analysis of these processes within Matlab is presented. Different types of effects are examined in the scope of the study, and methods for their creation are presented.

The results of this study indicate that software and signal processing techniques used in the production of digital timbre effects can play a significant role in music production. Furthermore, the methodology and results of the study can provide guidance for researchers interested in advancing the field of digital sound effects.

Keywords: Software, Matlab, Simulink, Reverb, Delay, Signal, Audio Toolbox

İÇİNDEKİLER DİZİNİ

KABUL VE ONAY SAYFASI	i
TEŞEKKÜR	ii
ÖZ	iii
ABSTRACT	iv
TABLolar DİZİNİ	viii
ŞEKİLLER DİZİNİ	viii
GİRİŞ	1
BÖLÜM I	2
1.1 Problem Durumu	2
1.2 Araştırmanın Amacı ve Önemi	3
1.3 Sayıtlılar	3
1.4 Sınırlılıklar	3
1.5 Tanımlar	4
BÖLÜM II	6
KAVRAMSAL ÇERÇEVE	6
2.1 Dijital Tını Efektleri	6
2.1.1 Chorus	6
2.1.2 Flanger	8
2.1.3 Phaser	10
2.1.4 Reverb (Yankı)	12
2.1.5 Delay (Gecikme)	16
2.2 Dijital Sinyal İşleme ve MATLAB	19
2.2.1 Fourier Dönüşümü	20
2.2.2 IIR ve FIR Filtreleri	21
2.2.3 Matlab Üzerinde Dijital Sinyal İşleme	23
2.2.4 Python ve C Dilleri ile Dijital Efekt Oluşturma	24
2.2.5 Simulink: Matlab Tabanlı Grafiksel Simülasyon Ortamı	25
2.2.6 Simulink Kütüphaneleri	27
2.3 Ses İşleme Algoritmaları: Teori ve Uygulamalar	28
2.3.1 Gerçek Zamanlı Ses İşleme	30

2.3.2 Ses İşleme Algoritmalarının Matematiksel Temelleri ve Gerçek Zamanlı Uygulamaları.....	32
2.4 Audio Toolbox	34
BÖLÜM III.....	36
VERİ ANALİZİ VE LİTERATÜR TARAMASI	36
3.1 Literatür Taraması.....	36
3.2 Veri Toplama	36
3.3 Veri Analizi ve Yöntem	36
BÖLÜM IV.....	37
BULGULAR VE YORUMLAR	37
4.1 Birinci Alt Probleme İlişkin Bulgu ve Yorumlar.....	37
4.1.1 Matlab ile Ses İşleme	37
4.1.2 Matlab Simulink Yazılımı ile Model Oluşturma.....	40
4.1.3 Matlab Simulink ile Ses Dosyalarının Analizi ve Simülasyonu	41
4.1.4 Modelleme Süreçlerine Efekt Bloklarının Eklenmesi.....	43
4.1.5 Matlab Audio Test Bench ile Tını Efektleri Analizi ve Uygulaması	46
4.2 İkinci Alt Probleme İlişkin Bulgu ve Yorumlar.....	50
4.2.1 Matlab Simulink ile Gerçek Zamanlı Tını Efekt Uygulaması ve Analizi	51
4.2.1.1 Reverberator Bloğunun Ses Dosyası Üzerindeki Etkilerinin Grafikselsel Analizi	51
4.2.1.2 Graphic EQ Bloğunun Ses Dosyası Üzerindeki Etkilerinin Grafikselsel Analizi	57
4.2.2 Matlab Audio Test Bench ile Tını Efekt Pluginlerinin Grafikselsel Analizleri	58
4.2.2.1 Chorus Efekt Plugininin Test Süreci ve Grafikselsel Analizi.....	58
4.2.2.2 Echo Efekt Plugininin Test Süreci ve Grafikselsel Analizi	63
4.2.2.3 Flanger Efekt Plugininin Test Süreci ve Grafikselsel Analizi	67
4.2.2.4 Phaser Efekt Plugininin Test Süreci ve Grafikselsel Analizi.....	71
4.2.2.5 Parametric Equalizer Plugininin Test Süreci ve Grafikselsel Analizi	76
4.3 Üçüncü Alt Probleme İlişkin Bulgu ve Yorumlar.....	80
4.3.1 Matlab Audio Test Bench ile Tını Efekt Pluginlerinin Harici Çıktılara Aktarılması	81

4.3.2 Oluşturulan Pluginin DAW içerisine Entegre Edilerek Kullanılması.....	83
BÖLÜM V	88
SONUÇ, TARTIŞMA VE ÖNERİLER	88
5.1 Sonuç.....	88
5.2 Tartışma	89
5.3 Öneriler	90
KAYNAKÇA	91
EKLER	92
EK-1 Tını Efektleri Kullanılarak Oluşturulan ve Miksaj Yapılan Ses Dosyalarının QR Kodu	92
EK-2 Üretilen Chorus Plugininin Arayüz Görseli	92
EK-3 Üretilen Echo Plugininin Arayüz Görseli.....	93
EK-4 Üretilen Flanger Plugininin Arayüz Görseli.....	93
EK-5 Üretilen Phaser Plugininin Arayüz Görseli	94
EK-6 Üretilen Parametric Equalizer Pluginin Arayüz Görseli	95
EK-7: YAYIMLAMA VE FİKRÎ MÜLKİYET HAKLARI BEYANI.....	96
EK-8: ETİK BEYANI.....	98
EK-9: TEZ ORJİNALLİK RAPORU	99
EK-10: Master's/Proficiency in Art/PhD Thesis/ Art Work Report Originality Report.....	100

TABLOLAR DİZİNİ

Sıra	Başlık	Sayfa
Tablo 1	Chorus ve Flanger efekti gecikme süreleri	11
Tablo 2	Reverb parametreleri ve açıklamaları	13

ŞEKİLLER DİZİNİ

Sıra	Tanımlama	Sayfa
Şekil 1	Chorus efekti ana sinyal ve kopyalanmış sinyal grafiği	7
Şekil 2	Flanger efekti ana sinyal ve kopyalanmış sinyal grafiği	9
Şekil 3	Phaser çalışma prensibi	10
Şekil 4	Ses sinyali efekt işleme şeması	11
Şekil 5	Doğrudan ses, erken yansımalar ve geç yankılanma grafiği	14
Şekil 6	Gecikme efekti çalışma prensibi ve temel parametreleri	16
Şekil 7	Ses dosyasının konumu	38
Şekil 8	Ses dosyasının Matlab üzerindeki çıktısı	38
Şekil 9	“audioread” komutu Matlab çıktısı	39
Şekil 10	Ses dosyasının Workspace üzerindeki çıktısı	39
Şekil 11	Simulink modelleme örnekleri	40
Şekil 12	Örnek bir ses dosyası okuyucu modeli	40
Şekil 13	‘From Multimedia File’ bloğu	41

Şekil 14	‘Audio Device Writer’ bloğu	42
Şekil 15	‘From Multimedia File’ blok parametreleri	42
Şekil 16	‘Audio Device Writer’ blok parametreleri	43
Şekil 17	Simulink “Library Browser” sekmesi	44
Şekil 18	Reverberator blok parametreleri	44
Şekil 19	Graphic EQ blok parametreleri	45
Şekil 20	Model blok dizilimi	45
Şekil 21	Modelin spectrum analyzer çıktısı	46
Şekil 22	“audioTestBench” komutu	47
Şekil 23	“Add an Object Under Test” sekmesi	47
Şekil 24	“audiopluginexample.Chorus” plugin parametreleri	48
Şekil 25	“Input” sekmesi giriş sinyal parametreleri	48
Şekil 26	“Output” sekmesi çıkış sinyal parametreleri	49
Şekil 27	Audio Test Bench üzerinden “Spectrum Analyzer” grafik çıktısı	49
Şekil 28	Reverberator ve Spectrum Analyzer blokları eklenmiş bir ses dosyası okuyucu modeli	51
Şekil 29	“CLAP.wav” isimli ses dosyasının Spectrum çıktısı	52
Şekil 30	“CLAP.wav” isimli ses dosyasının Spectrogram çıktısı	42
Şekil 31	Ham sinyal Spectrum Analyzer grafik çıktısı	53
Şekil 32	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	54
Şekil 33	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	55
Şekil 34	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	56

Şekil 35	İşlenmiş sinyal Spectrogram grafik çıktısı	56
Şekil 36	İşlenmiş sinyal Graphic EQ grafik çıktısı	57
Şekil 37	Ham sinyal Spectrum Analyzer grafik çıktısı	59
Şekil 38	Ham sinyal Time Scope çıktısı	59
Şekil 39	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	60
Şekil 40	İşlenmiş sinyal Time Scope grafik çıktısı	61
Şekil 41	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	62
Şekil 42	İşlenmiş sinyal Time Scope grafik çıktısı	62
Şekil 43	Ham sinyal Time Scope grafik çıktısı	63
Şekil 44	Ham sinyal Spectrum Analyzer grafik çıktısı	64
Şekil 45	Ham sinyal Spectrogram grafik çıktısı	64
Şekil 46	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	65
Şekil 47	İşlenmiş sinyal Time Scope grafik çıktısı	65
Şekil 48	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	66
Şekil 49	İşlenmiş sinyal Time Scope grafik çıktısı	66
Şekil 50	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	67
Şekil 51	İşlenmiş sinyal Time Scope grafik çıktısı	68
Şekil 52	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	69
Şekil 53	İşlenmiş sinyal Time Scope grafik çıktısı	69
Şekil 54	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	70
Şekil 55	İşlenmiş sinyal Time Scope grafik çıktısı	71
Şekil 56	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	72

Şekil 57	İşlenmiş sinyal Time Scope grafik çıktısı	72
Şekil 58	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	73
Şekil 59	İşlenmiş sinyal Time Scope grafik çıktısı	74
Şekil 60	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	75
Şekil 61	İşlenmiş sinyal Time Scope grafik çıktısı	75
Şekil 62	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	76
Şekil 63	İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı	77
Şekil 64	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	78
Şekil 65	İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı	78
Şekil 66	İşlenmiş sinyal Spectrum Analyzer grafik çıktısı	79
Şekil 67	İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı	80
Şekil 68	Audio Plugin Generator parametreleri	82
Şekil 69	Audio Plugin Generator dosya formatları	82
Şekil 70	Chorus.vst dosya yolu	83
Şekil 71	Chorus pluginin Studio One içerisindeki gösterimi	84
Şekil 72	Studio One “Console” sekmesi	84
Şekil 73	Ham sinyallerin master kanalı üzerinden Span grafik çıktısı	86
Şekil 74	İşlenmiş sinyallerin master kanalı üzerinden Span grafik çıktısı	87

GİRİŞ

Günümüz dünyasında müzik üretimi, teknolojik gelişmelerin bir sonucu olarak bilgisayar programcılığının önemini ortaya koymaktadır. Son yirmi yıldaki DAW yazılımlarının gelişimi, müzik prodüksiyonunun dijital dönüşümündeki önemli bir adım olarak kabul edilebilir. Bu yazılımlar, analog kayıt cihazlarının yerini alarak müzisyenlere ve müzik yapımcılarına daha fazla esneklik ve kontrol sağlamıştır.

DAW yazılımlarının ortaya çıkışı, ilk ses kaydı ile başlayan uzun bir sürecin sonucudur. İlk kayıt teknolojisi, 19. Yüzyılın ortalarında yaratılmış ve o zamandan günümüze hızla ilerlemiştir. Önce fonografin icadı, ardından manyetik bant kullanımı yaygınlaşmıştır. Bu teknolojiler sayesinde müzik kaydedilebilir ve saklanabilir hale getirilmiştir.

Dijital teknolojinin gelişmesiyle birlikte müzik üretimi dijitalleşmeye başlamış ve MIDI-Audio ses teknolojileri bilgisayar aracılığı ile müzik üretimini mümkün hale getirmiştir. Pluginler ve diğer yazılımlar da bu sürece katkı sağlamıştır. Bu gelişmeler sonucunda müzik üretimi alanında birçok farklı yazılım dili ve programı ortaya çıkmıştır.

BÖLÜM I

Bu bölümde problem durumuna, araştırmanın amacına, araştırmanın önemine, alt problemlere ve tanımlara yer verilmiştir.

1.1 Problem Durumu

Elde edilen teknolojik gelişmeler ve kolay bir şekilde erişilebilir eğitim kaynakları, herkesin kendi standartlarına uygun birçok uygulama, oyun ve elektronik cihaz üretmesine olanak sağlamaktadır. Müzik üretimi de bu teknolojik gelişmelerin bir sonucu olarak, bilgisayar programlama alanının günümüz dünyasındaki önemini gözler önüne sermektedir.

Bu tez çalışması, dijital efektlerin gelişimi, sinyal işleme teknikleri ve yazılım yoluyla müzik prodüksiyonundaki efektlerin nasıl üretilebileceği konularını ele almaktadır. Tını efektlerinin analog dönemdeki kullanımı ile dijital dönemdeki kullanımı arasındaki farklar, gecikme, yankı, phaser, flanger, chorus gibi tını efektlerinin üretim yöntemleri, Matlab’da dijital sinyal işleme ile ilgili konular detaylı bir şekilde incelenmektedir.

Bu çalışmada üzerinde durulan dijital tını efektleri, müzik sektöründe uzun yıllardan beri hem analog hem de dijital yöntemlerle kullanılmaktadır. Tını efektleri, oluşturulan müzik prodüksiyonunun önemli bir parçasıdır ve çalışmanın derinliği, yeri ve oluşturmak istediği etki, kullanılan efektlerle doğru orantılıdır. Bu çalışmada ele alınan dijital tını efektleri, geçmişte kullanılan yöntemlerin dijitalleştirilmiş sürümleri olarak görülebilir. Örneğin, bir yankı efektinin derinliğini ve genişliğini değiştirmek için eskiden bulunan ortamın fiziksel konumunu değiştirmek gerekiyorken, günümüz teknolojisi sayesinde, bütün mekanların taklit edilmiş efektleri bilgisayar ortamında kullanıma hazır bir şekilde bulunmaktadır.

Bu araştırmanın problemi;

- Yazılım tabanlı tını efekti üretim süreci ne şekilde gerçekleşmektedir?

Bu problem durumuna göre araştırmanın alt problemleri ise aşağıdaki gibidir;

1. Efektlerin oluşturulması için hangi aşamalar izlenmelidir?
2. Efekt parametreleri üzerinde durularak, farklı parametre değerlerinin etkileri nasıl analiz edilir?
3. Üretilen efektler müzik prodüksiyonu içerisinde nasıl kullanılabilir?

1.2 Araştırmanın Amacı ve Önemi

Bu araştırmanın amacı, müzik endüstrisinde dijital teknolojinin kullanımının artması ve bu sebeple dijital tını efektlerinin de daha yaygın bir şekilde kullanılmasıdır. Bu nedenle, efekt üretimi alanında uzmanlaşmak isteyenler için bir rehber niteliği taşıyan bu çalışma, yeni tekniklerin geliştirilmesine katkı sağlayabilir.

Bu çalışmanın önemi, dijital tını efektlerinin bilgisayar dili ve yazılım programları kullanılarak nasıl oluşturulabileceğine dair bir çalışma ortaya koymaktır. Tını efektlerinin bilgisayar ortamında üretilmesi, eski tip analog cihazların kullanımına göre daha kullanışlı ve anlaşılabilir olması sebebiyle önem arz etmektedir.

1.3 Sayıtlılar

Bu araştırmada, üretilen tını efektlerinin ve kullanılan Matlab yazılımının dünya standartlarında olduğu varsayılmaktadır. Ancak, bu varsayımın doğruluğu tam olarak kanıtlanmamıştır ve bu nedenle farklı görüşler olabilir. Ayrıca, araştırma içerisinde “Ses Efektleri” yerine “Tını Efektleri” terimi kullanılmıştır. Bu tercih, araştırmacının kendi tercihi olmakla birlikte, literatürde daha yaygın olarak kullanılan “Ses Efektleri” teriminin yerine geçmesi açısından dikkate alınabilir.

1.4 Sınırlılıklar

Bu tezde dijital tını efektlerinin oluşturulması ve kullanımına yönelik bir araştırma planlanmıştır. Ancak çalışmanın bazı sınırlılıkları da bulunmaktadır. Bu çalışma C dili içerisinde yer alan Matlab yazılımı ile sınırlandırılmıştır ve farklı yazılımların kullanımı çalışmaya dahil edilmemiştir. Ayrıca çalışmada ele alınan efektler, Matlab yazılımının programlandırılmış Simulink, Audio Toolbox ve Audio Test Bench araçlarının içerisinde bulunan; Reverberator, Graphic EQ, Chorus, Echo, Flanger, Phaser ve Parametric Equalizer örnekleri kullanılarak çalışmaya dahil edilmiştir ve başka bir efekt örneği bu çalışmada yer almamaktadır. Bunun yanı sıra çalışma sadece müzik sektöründe kullanılan efektler üzerine odaklanmıştır ve diğer sektörlerdeki kullanım alanlarına değinilmemiştir.

1.5 Tanımlar

- Modulation: Modülasyon, bir sinyalin özelliklerini başka bir sinyalin aracılığı ile değiştirilme işlemidir.
- Synthesis: Sentez, ses ve müzik oluşturma sürecine verilen isimdir. Sentez, yeni seslerin oluşturulabilmesi için ses kaynaklarının kombinasyonunu içerir.
- Classification: Sınıflandırma, verileri belli kategorilere ve sınıflara ayırma işlemine verilen isimdir.
- Signal Processing Algorithms: Sinyal işleme algoritmaları, sinyallerin analizi ve işlenmesi için kullanılan matematiksel yöntem ve tekniklere verilen isimdir.
- Digital Audio Effects: Dijital ses efektleri, ses sinyallerinin işlenerek özel efektlerin eklenmesini sağlayan, dijital teknoloji kullanılarak üretilen efektlerdir.
- Software-Based Effects: Yazılım tabanlı efektler, dijital ses işleme yazılımları kullanılarak oluşturulan efektlere verilen isimdir.
- Plugin: Eklenti, bir yazılımın içerisine entegre edilen ve özel işlevler sağlayan yazılım bileşenidir.

- Virtual Studio Technology (VST): Sanal stüdyo teknolojisi, yazılım tabanlı ses işleme ve müzik prodüksiyonu için kullanılan standart bir dosya formatıdır.
- Dynamic Effects: Dinamik efektler, ses sinyalinin dinamik aralığını ve karakteristik özelliklerini değiştiren efektlerdir.
- Digital Signal Processing (DSP): Dijital sinyal işleme, analog sinyallerin dijital ortama aktarılmasını ifade eder.
- User Interface (UI): Kullanıcı arayüzü, kullanıcının bir yazılım içerisinde etkileşimde bulunabilmesini sağlayan görsel veya metin tabanlı arayüzlere verilen isimdir.
- Parameter: Parametre, bir fonksiyon, işlem veya sistem tarafından kullanılan ayarlanabilir değerlere verilen isimdir.
- Matlab: Matlab, matematiksel hesaplamalar, veri analizleri, ses işleme fonksiyonları gibi uygulamalar için kullanılan bir programlama dili yazılım ortamıdır.
- C, C++ : Çok amaçlı bir programlama dilidir.
- Python: Python, çok amaçlı bir programlama dili ve yaygın kullanılan bir yazılım geliştirme aracıdır.
- Audio Mixing: Ses miksajı, oluşturulan ses kaynaklarının birleştirilerek denge ve uyumun sağlandığı süreçtir.
- Equalization (EQ): Eşitleme, ses sinyallerindeki frekans tepkilerini değiştirerek, belirli frekanslar üzerinde oynama işlemidir.
- Mixing Console: Miksaj konsolu, ses sinyallerinin düzenlendiği ve kontrol edildiği analog veya dijital ses aracıdır.
- Digital Audio Workstation (DAW): Dijital ses işleme istasyonu, müzik veya ses üretim sürecinde ve düzenlemesinde kullanılan yazılımlara verilen isimdir.
- Timbre: Timbre, bir müzik aletinin karakteristik özelliklerini veya tınısını ifade eden bir kavramdır.

BÖLÜM II

KAVRAMSAL ÇERÇEVE

Bu bölümde araştırmanın ilgilendiği konu başlıklarına tarihsel süreçleri ve gelişimleri ile beraber detaylı bir şekilde yer verilmiştir.

2.1 Dijital Tını Efektleri

Dijital tını efektleri, modern müzik sektörü ve müziğin kullanıldığı tüm alanlarda temel araçlar olarak kullanılmaktadır. Kuru bir ses sinyalini yaratıcı ve etkileyici bir hale getirmek ya da sinyal kalitesini arttırmak için tını efektleri kullanılmaktadır. Tını efektleri, etkilerine göre dinamik, modülasyon ve zaman bazlı olarak kategorilere ayrılmaktadır. Dinamik efektler, ses sinyalinin yükseklik ve alçaklığı ile ilgilidir ve ses sinyalindeki dalgalanmaların düzenlenmesi, azaltılması, artırılması ve belirli bir seviyeden sonra kapatılması gibi işlemleri gerçekleştirir. Dinamik efektler, stüdyo miks aşamasında, pedal efekt zincirlerinde, sahne performanslarında tercihen genellikle öncül olarak yer almaktadır.

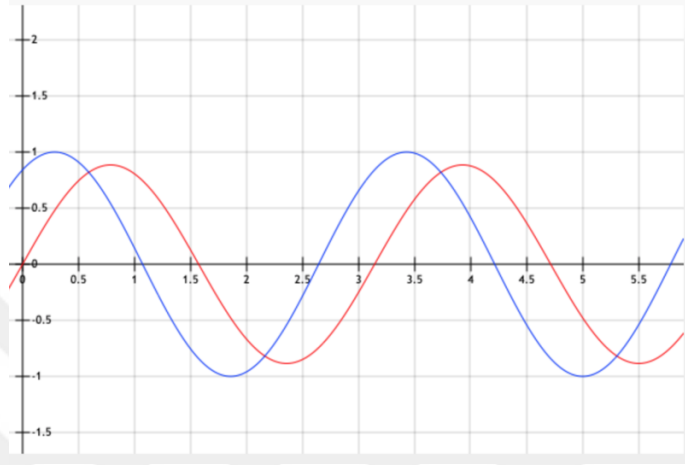
Modülasyon efektleri ise efekt sıralamasında dinamik efektlerden sonra gelmekte ve prodüksiyonlarda, canlı performanslarda sıkça kullanılmaktadır. Modülasyon efektleri arasında Chorus, Flanger ve Phaser olmak üzere üç temel modülasyon efekti bulunmaktadır.

2.1.1 Chorus

Chorus efekti, aynı perde ve tınıya sahip birkaç ayrı sesin art arda çalmasıyla ortaya çıkar. Chorus, bu zamanlama ve perde farklılıklarını simüle ederek, tek bir enstrüman kaynağının bir grup enstrüman tarafından çalınıyormuş gibi ses çıkarmasını sağlar (Varghese&Yadav&Bhatia,2021).

Chorus efektinin en belirgin özelliği ana sinyalin kopyası olan sinyalde yaşanan gecikmedir. Bu gecikme genellikle 5-20 milisaniye arasında sürer. Bu

gecikmeden kaynaklı oluşan tını farklılıkları ve faz çakışmaları gerçek bir koroda oluşan perde farklılarına benzediği için efekt Chorus olarak adlandırılmıştır. Chorus çalışma prensibi, sinyal grafiği olarak Şekil 1’de gösterilmiştir.



Şekil 1 Chorus efekti ana sinyal ve kopyalanmış sinyal grafiği

Bahsi geçen gecikme süresi bir LFO (Low Frequency Oscillator) ile sürekli olarak değiştirilir. Bir başka deyişle, gecikme süresi bir osilatör aracılığı ile modüle edilir. “Modülasyon efekti” terime de buradan gelmektedir. Bu modülasyon sinyalin perdesinde küçük kaymalara yol açar (Önen, 2016).

Chorus efektinin kendi içerisinde tekrarlanması halinde ortaya çıkan bir başka efekt ise “ Multi Voice Chorus” olarak adlandırılır. Bu efekt giriş sinyalinin birden fazla gecikmeli kopyasını oluşturur ve her biri birbirinden bağımsız hareket eder.

Chorus efektini belirleyen temel parametreler:

1. **Delay:** Bu parametre, efektin zaman aralığını belirler. Daha kısa bir zaman aralığı daha yoğun bir chorus efekti anlamına gelir.
2. **Decay:** Bu parametre, efektin zamanla azalmasını belirler. Daha yüksek bir Decay değeri efektin daha hızlı azalması anlamına gelir.

- 3. Depth:** Bu parametre, efektin derinliğini belirler. Daha yüksek bir depth değeri efektin daha fazla olduğu anlamına gelir.

Bu parametreler istenilen chorus efektini oluşturmak için uygun şekilde ayarlanabilir.

Chorus efektinin matematiksel parametreleri:

$$y_chorus(t) = y(t) + depth * \sin(2 * \pi * f * t_delay * n) * \exp(-decay * n)$$

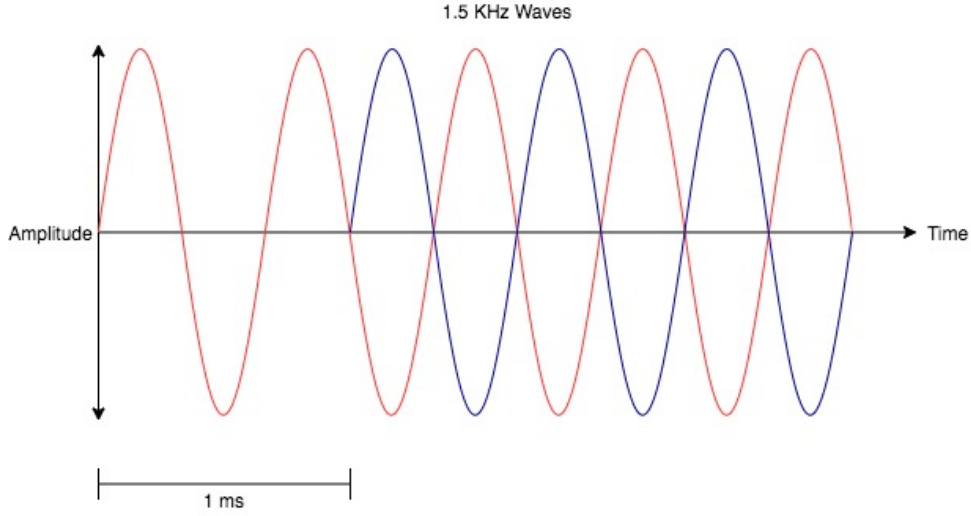
Burada:

1. $y(t)$: kaynak ses sinyali
2. $depth$: chorus efekt derinliği
3. f : çaplamanın frekansı
4. t_delay : çaplamanın zaman aralığı
5. $decay$: çaplamanın azalma hızı
6. n : çaplamanın sayısı (sürekli bir değişken)

Bu formül, her bir çaplamanın kaynak ses sinyalini belirleyen faktörleri içerir ve chorus efektinin çaplamanın zamanla azalmasını ve derinliğini modelleyen fonksiyonlarla birlikte hesaplanır.

2.1.2 Flanger

Flanger efekti chorus efekti ile çok benzer bir yapıya sahiptir. Aralarındaki fark gecikme süreleri ile alakalıdır. Chorus efekti uygulanan bir sinyalin, kopya sinyalle arasındaki gecikme süresi 5-20 milisaniye arasındayken flanger efektinin gecikme süresi 1-10 milisaniye arasındadır. Flanger çalışma prensibi, sinyal grafiği olarak Şekil 2'de gösterilmiştir.



Şekil 2 Flanger efekti ana sinyal ve kopyalanmış sinyal grafiği

Flanger efektini belirleyen temel parametreler:

1. **Delay:** Bu parametre, ses sinyallerinin zaman aralığını belirler. Daha kısa bir zaman aralığı daha yoğun bir flanger efekti anlamına gelir.
2. **Depth:** Bu parametre, ses sinyallerinin derinliğini belirler. Daha yüksek bir depth değeri ses sinyallerinin daha fazla birbirine karıştığı anlamına gelir.
3. **Rate:** Bu parametre, ses sinyallerinin birbirine karışma hızını belirler. Daha yüksek bir rate değeri ses sinyallerinin daha hızlı birbirine karıştığı anlamına gelir.
4. **Feedback:** Bu parametre, ses sinyallerinin geri beslemesini belirler. Daha yüksek bir feedback değeri ses sinyallerinin daha fazla geri beslendiği anlamına gelir.

Bu parametreler, istenilen flanger efektini oluşturmak için uygun bir şekilde ayarlanabilir.

Flanger efektinin matematiksel parametreleri:

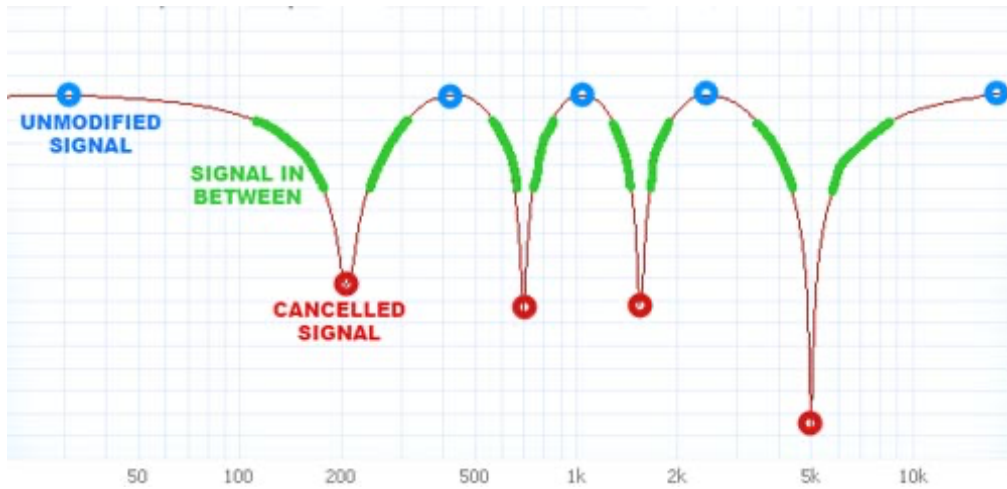
$$y(t) = x(t) + (1 - f) * x(t - d)$$

Bu formülde, $x(t)$ giriş ses sinyalini, $y(t)$ çıkış ses sinyalini, f geri besleme faktörünü, d zaman aralığını ve $(1 - f)$ çarpma faktörünü temsil eder. Bu formül, flanger efektinin temel işleme mekanizmasını açıklar ve ses sinyallerinin birbirine nasıl karıştığını gösterir. Flanger efekti için diğer parametreler de bu formül içinde yer alabilir ve flanger efektinin görüntüsünü daha da ayrıntılı hale getirebilir.

2.1.3 Phaser

Phaser, girişteki sinyalin bir kopyasını çıkartır ve “all-pass” (tam geçirgen) olarak adlandırılan bir dizi filtreden geçirir. All-pass filtreler sinyalin frekans içeriği üzerinde herhangi bir değişiklik yapmaz. Diğer yandan, all-pass filtrelerin frekans faz tepkisi lineer olmadığından dolayı, ham (orijinal) sinyal ve işlenmiş sinyal arasında frekanslara göre değişiklik gösteren faz farklılıkları oluşur. Diğer bir deyişle, ham sinyaldeki farklı frekanslar farklı sürelerde gecikmeye uğramış olur ve bu da çıkış sinyalinde çoğu zaman armonik olarak birbirleriyle uyumlu olmayan çukur noktaları oluşturur. Bahsi geçen gecikme süresi bir LFO (Low Frequency Oscillator) aracılığı ile periyodik olarak değiştirilir, bunun sonucunda çıkış sinyalindeki çukur noktalarının yerleri değişir. Ortaya çıkan bu efekt “phasing effect” ya da “phaser effect” olarak adlandırılır (Önen, 2007).

Phaser çalışma prensibi, sinyal grafiği olarak şekil 3 üzerinde gösterilmiştir.



Şekil 3 Phaser çalışma prensibi

Chorus, Flanger ve Phaser efektlerini birbirinden ayıran en temel fark gecikme süreleridir. Chorus efektinin gecikme milisaniyesi, Flanger efektine göre daha fazladır. Phaser efektinin ise kopyalanan sinyal arasında gecikme farkı yoktur.

Gecikme sürelerinin milisaniye olarak karşılıklarını gösteren veriler Tablo 1 üzerinde gösterilmiştir.

Efekt	Gecikme (ms)
Chorus	5-20
Flanger	1-10

Tablo 1 Chorus ve Flanger efekti gecikme süreleri

Kuru bir şekilde gelen ses sinyalinin, efekt uygulandıktan sonra çıkış yönüne doğru hareketini gösteren şema Şekil 4 üzerinde gösterilmiştir.

Ses Girişi → Ses Efektli → Ses Çıkışı

Şekil 4 Ses sinyali efekt işleme şeması

Phaser efektini belirleyen temel parametreler:

1. **Rate:** Bu parametre, Phaser filtresinin döngü hızını belirler. Daha yüksek bir rate değeri, daha hızlı bir Phaser efekti anlamına gelir.
2. **Depth:** Bu parametre, Phaser filtresinin derinliğini açıklar. Daha yüksek bir depth değeri, sesin daha fazla Phaser efekti altında olduğunu gösterir.
3. **Feedback:** Bu parametre, Phaser filtresinin geri beslemesini belirler. Daha yüksek bir feedback değeri Phaser efektinin daha fazla güçlendiği anlamına gelir.
4. **Stages:** Bu parametre, Phaser filtresinin aşamalarını belirler. Daha fazla aşama, daha karmaşık bir Phaser efekti anlamına gelir.

Bu parametreler, istenilen Phaser efektini oluşturmak için uygun bir şekilde ayarlanabilir.

Phaser efektinin matematiksel parametreleri:

$$y(t) = x(t) + c * \sin(2 * \pi * f * t + a * \sin(2 * \pi * r * t))$$

Bu formülde, $x(t)$ giriş ses sinyalini, $y(t)$ çıkış ses sinyalini, f modülasyon frekansını, r modülasyon hızını, a modülasyon derinliğini ve c geri besleme faktörünü açıklar. Bu formül, Phaser efektinin temel çalışma prensibini tanımlar ve ses sinyallerinin nasıl faz değişikliği yapılarak modüle edildiğini gösterir. Phaser efekti için diğer parametreler de bu formül içinde yer alabilir ve Phaser efektinin görüntüsünü daha da ayrıntılı hale getirebilir.

2.1.4 Reverb (Yankı)

Yankı geçmiş dönemlerden beri müziğin önemli bir bileşeni olmuştur. Yankı, bir ses dalgasının işitilmeden geçtiği mesafenin büyüklüğü ile ilgilidir. Konser salonları, sinema salonları, kayıt stüdyoları mimari açıdan yankının kontrol

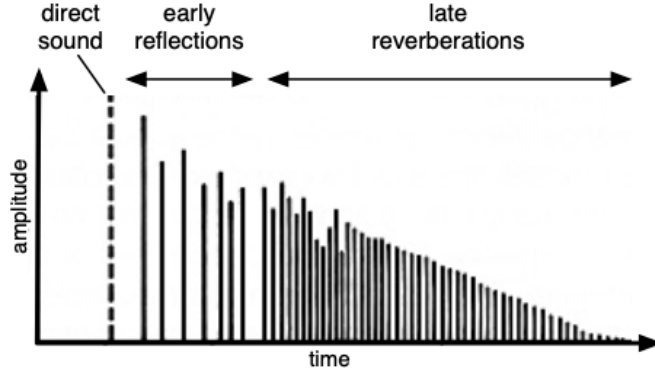
edilebilmesi için tasarlanmıştır. Bu ses dalgalarını dinleyiciye aktarmadan önce yapılan yankı, duyulan sesi daha efektif bir hale getirmektedir.

Yankı efekti müzikte bir derinlik etkisi yarattığı için zamanla canlı performanslardan müzik kayıt alanlarına geçiş yaptı. Yankı efekti, kayıt alınan odanın da gürültüsü alınarak yapılan bir efekttir. Yankı çeşitleri Hall, Chamber, Spring, Plate, Room olarak 5 çeşide ayrılır. Bunların arasındaki farklılıklar büyüklükleri ile ilgilidir. Reverb parametreli ve açıklamaları Tablo 2 üzerinde gösterilmiştir.

Tür	Yankının işleyeceği oda.
Uzunluk/Zaman	Yankının ne kadar sürdüğü.
Erken Gecikme	Tetiklemeden sonra yankının ne kadar hızlı başlayacağı.
Erken Yansıma/Kuyruk	Ses başladıktan sonra duyulan ilk yansımalar.
Boyut	Yankının işleyeceği odanın büyüklüğü.
Genişlik	Yankı efektinin stereo olarak genişliği
Difüzyon	Yansımaların birbirine ne kadar yankın olduğu
Ekolayzır (EQ)	Yankı efektinin frekans kontrolü.
Sönümleme	Yankının işleyeceği odanın yumuşaklığı.
Kuru/İslak	Yankılanan sese göre orijinal sesin ne kadar duyulduğu.
Çıkış	Yankının genel seviyesi.

Tablo 2 Reverb parametreleri ve açıklamaları

Yankılanan sinyalin azalma süresi, genellikle alanın gelen sinyaller aracılığı ile uyarılmasından sonra yankılanma seviyesinin 60 dB azalması için geçen süreyi açıklayan RT60 tarafından tanımlanır. Şekil ve yüzey yapılarına bağlı olarak yansıyan yankılar, belirli bir ses sinyaline belirli renklendirme (frekans geliştirme veya azaltma) ekleyen doğrudan sinyalin tam kopyaları değildir. Bir odanın bu sinyale yanıtı ise üç bölüme ayrılır: doğrudan ses, erken yansımalar ve geç yankılanma. Doğrudan ses, erken yansımalar ve geç yankılanmaya ilişkin grafik Şekil 5 üzerinde gösterilmiştir.



Şekil 5 Doğrudan sesi erken yansımalar ve geç yankılanma grafiği

Yankı efekti, bir ses kaynağından yansıyan seslerin zaman dağılımı ve frekans spektrumlarındaki değişikliklerini modellemek için bir takım matematiksel filtre ve algoritmalarından oluşur. Bu filtreler ve algoritmalar, ses yansımalarının yavaş yavaş azalmasını, frekans spektrumlarındaki değişiklikleri ve yansımaların zaman dağılımını simüle eder. Yankı efekti, birbirinden farklı parametreler ile kontrol edilebilir ve bu parametreler efektin tipini, süresini, hızını ve frekans cevaplarını belirler. Örneğin, yankı efekti için bir ortam modelleme parametresi, bir konser mekanı, kilise veya açık hava sahnesi gibi farklı mekanlardaki yansımaları taklit etmeye çalışır. Süre parametresi ise, yankı efektinin süresini belirler ve yankı efektinin ne kadar süre boyunca devam edeceğini kontrol eder. Genel olarak yankı efekti, ses sinyallerinin doğal ortamdaki yansımalarını taklit etmeye yarayan bir teknolojidir ve müzik sektöründe yaygın bir biçimde kullanılır. Yankı efekti, seslerin doğal, gerçekçi ve etkileyici hale getirilmesine yardımcı olur.

Yankı efekti, bir ses kaynağından çıkan sinyalin yankılanarak çevredeki nesnelerden geri dönmesi sonucu oluşan doğal bir etkidir. Bu efekt, kaydedilen sesin daha geniş ve doğal bir ortamda çalınmış gibi duyulmasını sağlar. Yankı efekti, kaydedilen sinyalin zaman içindeki değişimlerini taklit eden algoritmalar kullanır ve bu sayede farklı yankı ayarları ile farklı ortamların simülasyonu yapılabilir (Varghese&Yadav&Bhatia,2021).

Yankı efektini belirleyen temel parametreler:

1. **Decay Time:** Bu parametre, yankının sesin güçsüzleşmesi için ne kadar zaman harcayacağını belirler.
2. **Damping:** Bu parametre, yankının hangi frekanslarının güçsüzleşeceğini belirler.
3. **Room Size:** Bu parametre, yankının ne kadar geniş bir alanda uygulanacağını belirler.
4. **Predelay:** Bu parametre, yankının sesin yansımından önceki zaman aralığını belirler.
5. **Wet/Dry Mix:** Bu parametre, yankının orijinal sesle ne kadar karıştırılacağını belirler.

Bu parametreler istenilen yankı efektini oluşturmak için uygun şekilde ayarlanabilir.

Yankı efektinin matematiksel bir formülü yoktur. Bu efekt için sadece modelleme yöntemi kullanılabilir. Modelleme çeşitleri çok karmaşık ve teknik bilgi içerdiğinden dolayı diferansiyel denklemler, Fourier dönüşümler, gecikme hattı filtreleme veya impuls yanıtı tekniği gibi yöntemleri içerir. Yankı efekti bir ortamda yansıyan seslerin bütünü olarak düşünülebilir ve bu yansıyan seslerin grafiği, yansıyan mekanın fiziksel özellikleri, yansıma yerine olan uzaklık ve yansıma yerinin iletkenliği gibi faktörlere göre değişkenlik gösterir. Bu sebeple, yankı efekti için tek bir formül belirlemek ve hesaplamak çok zordur, uygulanan şartlara göre her efekt ayrı modellenmelidir.

2.1.5 Delay (Gecikme)

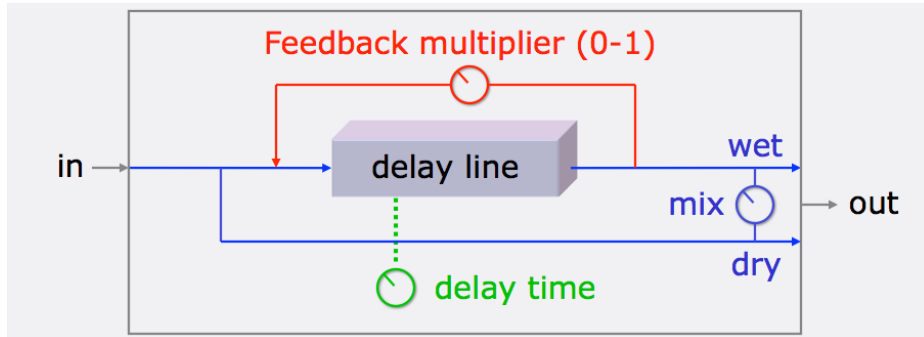
Gecikme efekti, müziğin ya da ses kaynağının bir önceki çalışmasının tekrar etmesi sonucunda var olan bir tını efektidir. Bu efekt, ses dalgalarının belirli bir sürede gecikme uğraması ile oluşur ve çeşitli şekillerde kullanılabilir.

Gecikme efekti, 1960'ların sonlarına doğru analog teknolojiler ile oluşturulan gecikmeli çalma (tape delay) tekniği ile kendini göstermiştir. Gecikme efekti, müziğin dinamikliğini ve derinliğini arttırmak, ritmik dokunuşlar oluşturmak ve şarkının atmosferini zenginleştirmek için kullanılabilir.

Gecikme efekti, bir ses sinyalindeki gecikmeyi kullanarak oluşturulan bir efekt türüdür. Bu efekt, insan işitme sisteminin gecikmeli sinyalleri nasıl aldığını ve yorumladığını temel alır. Bu etkiler, tekrarlar, akustik alanın bazı özellikleri veya spektral alanın dönüşümler olarak anlaşılabilir. Müzikte zaman gecikmesine dayalı ses efektlerinin ilk kullanımları 1940'lara kadar uzanır (Earl,2006).

Gecikme efektinin birden fazla çeşidi bulunmaktadır. Bunların içinde tek gecikmeli efektler (single delay), çoklu gecikmeli efektler (multi-tap delay), modüle edilmiş gecikme efektleri (modulated delay), hall efektleri (artificial reverb) ve feedback delay efektleri bulunabilir.

Gecikme efektinin çalışma prensibi ve parametreleri ile ilgili grafik Şekil 6 üzerinde gösterilmiştir.



Şekil 6 Gecikme efekti çalışma prensibi ve temel parametreleri

Gecikme efekti, müzikte sıklıkla kullanılan bir efekt türüdür. Bu efekt, bir ses sinyalindeki gecikmeyi kullanarak oluşturulur ve tekrar eden yankılan veya yankıların oluşmasına neden olur. Delay efekti, müzikte farklı amaçlar için kullanılabilir. Örneğin, bir gitarist, gecikme efektini kullanarak tek başına çaldığı parçada daha dolgun bir ses elde edebilir veya vokalist, gecikme efektini kullanarak vokal performansını daha etkileyici hale getirebilir. Ayrıca, gecikme efekti müzik prodüksiyonunda da yaygın olarak kullanılır ve kaydedilen enstrümanların veya vokallerin daha zengin ve derin bir ses kalitesine sahip olmasını sağlar (Price,2019).

Gecikme efektini belirleyen temel parametreler:

1. **Delay Time:** Gecikme efektinin temel özelliği, sinyalin ana kaydından bir süre sonra tekrarlanmasıdır. Bu zamanı ölçmek için kullanılan parametre “delay time” yani gecikme süresidir. Gecikme süresi çoğunlukla milisaniye (ms) cinsinden ifade edilir ve çoğunlukla 20 ms ile 1000 ms arasında değişebilir. Kısa gecikme süreleri, sesin daha pes ve doygun bir hale gelmesine, uzun gecikme süreleri ise yankı etkisi üretmemize yardımcı olur.
2. **Feedback:** Gecikme efektinin diğer önemleri parametresi ise “feedback” yani geri bildirimdir. Geri bildirim, çoğaltılmış sinyalin ana sinyale geri beslenmesini sağlar ve daha uzun yankı süreleri üretmek için kullanılır. Bu parametre aynı zamanda, efektin ne kadar yoğun duyulacağını belirler.
3. **Mix:** Mix yani karıştırma oranı, ana sinyal ile çoğaltılmış sinyal arasındaki dengeyi belirler. Bu parametre genellikle yüzde (%) olarak ifade edilir ve 0% ila 100% arasında değişebilir. Karıştırma oranının artması, efektin daha belirginleşmesi ve ana sinyalin daha az duyulmasına neden olur.
4. **Modulation:** Modülasyon, gecikme efektine eklenen bir parametre olabilir ve efektin daha ilginç ve ilgi çekici hale gelmesine katkıda bulunur. Modülasyon, çoğaltılmış sinyalin yavaşça değişen frekans veya ses hacmi gibi özelliklerine bir dalgalanma ekler.
5. **Filter:** Bazı gecikme efektleri, sinyalin belirli frekanslarını çıkartarak veya arttırarak belirli bir karakter sağlamak için bir filtre uygulayabilir.

- 6. Timing:** Bazı gecikme efektleri, gecikme süresini belirli bir zamanda ayarlayarak, ritmik bir yankı veya atak oluşturabilir. Bu genellikle ritimli enstrümanlar için kullanılır.

Bu parametreler, birbirinden farklı birçok gecikme efektinde farklı yollarla uygulanabilir ve birçok müzik türünde ve kayıt platformlarında kullanılabilir.

Gecikme efektinin matematiksel parametreleri:

$$y(n) = x(n) + F * x(n-d)$$

Burada $y(n)$, çoğaltılmış sinyal, $x(n)$ ana sinyal, F geri besleme değeri, d gecikme süresidir. Bu formül, gecikme süresi boyunca ana sinyale geri beslemeli olarak eklenen çoğaltılmış sinyalin hesaplanmasını sağlar. Geri besleme değeri, çoğaltılmış sinyalin ne kadar geri beslenmesinin sağlanacağını kontrol eder. Bu formül, sadece basit bir delay efekti için geçerli olur ve farklı parametreler, farklı özellikler eklenerek daha karmaşık bir hale getirilebilir.

Delay etkisi, bir sinyali zaman içinde kaydırarak elde edilir ve sinyal üzerinde çeşitli işlemler yaparak yinelenen gecikmeler oluşturulabilir. Matematiksel olarak, delay etkisi, orijinal sinyale bir gecikme süresi boyunca orijinal sinyalin bir çoğaltımının eklenmesiyle elde edilir. Bu, basitçe, sinyalin geçmişteki bir versiyonunun şimdiki sürümüne eklenmesi anlamına gelir. Bu işlem, geri besleme oranı ve gecikme süresi gibi parametrelerle kontrol edilir ve sinyalin reverb, chorus veya echo gibi farklı efektlerle değiştirilmesine olanak tanır (van der Hulst,2001).

2.2 Dijital Sinyal İşleme ve MATLAB

Dijital sinyal işleme, analog sinyalleri örnekleme, nicelleme ve dijital işleme yoluyla dijital bir forma dönüştürme işlemidir. Bu süreç, dijital sinyallerin kaydedilmesi, işlenmesi ve iletilmesi için birçok uygulama tarafından kritik bulunur. Matlab, bu sürecin birçok yönünü kolaylaştıran bir programlama ortamı sağlar.

Matlab, çok çeşitli işlevler ve araçlar sağlayarak dijital sinyal işlemeye birçok fayda sağlar. Bu işlevler, örnekleme, nicelleştirme, filtreleme, çizim ve analiz gibi temel işlemleri içerir. Matlab, sinyalleri ve sistemleri analiz etmek için çeşitli grafiksel araçlar sağlar ve hatta kullanıcıların kendi özel sinyal işleme yöntemlerini uygulamak için kendi işlevlerini yazmalarına izin verir.

Matlab, kullanıcıların örnekleme ve nicelleştirme işlemlerini gerçekleştirmesini kolaylaştırır. Matlab, sinyallerin ve sistemlerin çeşitli özelliklerini hesaplamak için kullanılan birçok işlevi sağlar. Bunlar, FFT (Fast Fourier Transform) gibi spektral analiz fonksiyonlarını, FIR (Finite Impulse Response) ve IIR (Infinite Impulse Response) filtreleri gibi dijital filtre tasarım fonksiyonlarını ve Signal Processing Toolbox gibi gelişmiş fonksiyon kitaplıklarını içerir.

Matlab ayrıca birçok grafik aracına sahiptir. Zaman ve frekans alanlarındaki sinyalleri ve sistemleri görüntülemek için kullanılabilen çizim, kök ve frekans fonksiyonları gibi temel grafik fonksiyonlarını içerir. Matlab ayrıca sinyal spektrumlarını, spektrogramları ve zaman-frekans analizini görselleştirmek için Spektrogram ve Pwelch gibi gelişmiş grafiksel işlevler sağlar. Genel olarak Matlab, dijital sinyal işleme uygulamaları için güçlü bir araçtır. Matlab, kullanıcılara örnekleme, nicelleştirme, filtreleme, spektral analiz, filtreleme ve grafikleme işlevleri gibi temel işlemlerin yanı sıra gelişmiş özellikler sağlar. Matlab, kullanıcıların sinyallerinin nasıl işlendiğini özelleştirmek için kendi özel işlevlerini oluşturmalarına izin verir. Örneğin, Matlab kullanarak FIR ve IIR filtre tasarımı, FFT spektral analizi, veri sıkıştırma, ses işleme ve görüntü işleme konularına daha ayrıntılı bir şekilde değışilebilir. Ayrıca Matlab optimizasyon fonksiyonları, öğrenme algoritmaları ve sinir ağları gibi ilgili diğer konulardan da bahsedebilir.

DSP (Digital Signal Processing), analog sinyallerin dijital veriye dönüştürülmesini, dijital verilerin depolanmasını ve işlenmesini içerir. Bu süreçte, dijital sinyaller analog sinyallerin birebir taklidi değildir; bunun yerine, sinyaller örneklenir, nicelleştirilir ve dijital veriler haline getirilir. DSP, birçok farklı uygulama alanında kullanılmaktadır, örneğin, telekomünikasyon, radyo yayıncılığı, müzik ve tıp gibi alanlarda kullanılmaktadır (Proakis,2006).

Matlab, sayısal hesaplama, veri analizi ve görselleştirme için kullanılan bir programlama dili ve ortamıdır. Matlab, matris işlemleri, fonksiyon çizimi, veri analizi ve algoritmalarının uygulanması gibi birçok matematiksel işlemi kolaylaştıran bir araçtır. Matlab ayrıca, sinyal işleme, görüntü işleme ve yapay zeka gibi alanlarda da yaygın olarak kullanılır. Matlab'ın avantajlarından biri de açık kaynak kodlu olmasıdır. Bu sayede kullanıcılar kendi fonksiyonlarını yazabilirler. Matlab ayrıca grafiksel kullanıcı ara yüzleri (GUI) oluşturmak için de kullanılabilir. Bu özellikler sayesinde Matlab, mühendislik, bilim ve matematik alanlarında yaygın olarak kullanılan bir araçtır. Sinyal işleme, bir sinyalin matematiksel olarak analiz edilmesi, işlenmesi ve yorumlanmasıdır. Sinyal işleme, ses, görüntü, radyo dalgaları ve diğer fiziksel olayların matematiksel modellerinin oluşturulmasında kullanılır. Bu modeller daha sonra sinyallerin özelliklerini çıkarmak veya değiştirmek için kullanılabilir. Sinyal işleme teknikleri, sinyallerin frekansı, fazı, genliği ve zamanı gibi özelliklerini analiz etmek için kullanılır. Bu teknikler arasında filtreleme, Fourier dönüşümü ve wavelet dönüşümü gibi yöntemler bulunur. Sinyal işleme teknolojisi, telekomünikasyon, tıp görüntüleme sistemleri, akustik mühendisliği ve diğer birçok alanda yaygın olarak kullanılır (Varghese&Yadav&Bhatia,2021).

2.2.1 Fourier Dönüşümü

Fourier dönüşümü, zamana bağlı bir sinyalin frekansa bağlı bir gösterime dönüştürülmesiyle ilgili matematiksel bir işlemidir. Bu işlem, zaman ve frekans arasındaki iki değişkenli dönüşümün bir örneğidir. Fourier dönüşümleri, ses, video, radyo ve diğer iletişim sistemlerinde kullanılan sinyalleri analiz etmek ve sentezlemek için önemlidir.

Bir sinyalin Fourier dönüşümü bize sinyalin sürekli frekans spektrumunu verir. Bir sinyalin zamanı, frekans bileşenlerine bölünerek her bileşenin frekansını, genliğini ve faz açısını belirtir. Fourier dönüşümleri genellikle sinyallerin frekans içeriğini belirlemek ve sinyal işleme algoritmalarını geliştirmek için kullanılır.

Matematiksel olarak, bir sinyalin Fourier dönüşümü, sinyalin belirli frekanslardaki fazının ve genliğinin karmaşık bir sayısal temsidir. Dönüşüm genellikle aşağıdaki formülle temsil edilir:

$$F(w) = \text{integral} (x(t) * e^{(-j\omega t)} dt), -\infty < t < \infty$$

Burada $x(t)$ zamana bağlı sinyaldir ve $F(w)$ sinyalin frekansa bağlı gösterimidir. ‘j’ karmaşık sayı anlamına gelir ve ‘w’ frekans anlamına gelir. Entegrasyon, sinyalin tüm zaman aralığı boyunca hesaplanması anlamına gelir. Fourier dönüşümü, ters Fourier dönüşümü kullanılarak da tersine çevrilebilir. Arzu edilirse, ters Fourier dönüşümü, sinyali frekansa bağlı temsilinden orijinal zamana bağlı sinyale geri dönüştürmek için kullanılabilir.

Fourier dönüşümü, bir sürekli zamanlı sinyalin frekans alanındaki bileşenlerine ayırmak için kullanılan bir matematiksel araçtır. Dönüşüm, bir sinyalin zamana bağlı olarak değişen fonksiyonu yerine, frekans bileşenlerinin bir dizisi olarak temsil edilmesini sağlar. Bu, bir sinyalin frekans özelliklerini anlamamıza ve sinyalin nasıl işleneceğine karar vermemize olarak tanır.

2.2.2 IIR ve FIR Filtreleri

Infinite Impulse Response (IIR), bir tür sinyal işleme filtresidir. Bu filtreler, giriş sinyalini belirli matematiksel işlemlerle işleyerek bir çıkış sinyali üretir.

IIR filtrelerinin en önemli özelliği, bir geri besleme mekanizması kullanarak geçmiş değerlere bağımlı olmalarıdır. Dolayısıyla bu filtrelerin çıkışları, önceki çıkış değerleri ve giriş sinyalleri ile ilişkilendirilir. Finite Impulse Response (FIR)

filtreleriyle karşılaştırıldığında, IIR filtreleri daha az hesaplama gücü ve depolama alanı gerektirir. Ancak IIR filtreleri, FIR filtrelerinden daha az kararlıdır ve özellikle yüksek frekanslarda diğer filtrelerden daha fazla bozulmaya neden olabilir.

IIR filtreleri, özellikle analog filtre tasarımı ile ilgilenen mühendisler tarafından yaygın olarak kullanılmaktadır. Bu filtreler, dijital filtreler tasarlamak için de kullanılabilir ve Matlab gibi sinyal işleme araçları, IIR filtreleri tasarlamak ve uygulamak için hazır işlevler sağlar.

Finite Impulse Response (FIR) filtreleri, dijital sinyal işleme uygulamalarında yaygın olarak kullanılan dijital filtrelerdir. Bir FIR filtresi, bir giriş sinyaline örneklenmiş bir ayırık sinyal veya dürtü yanıtı uygulayan bir filtre türüdür.

FIR filtreleri, sabit katsayılı sistem fonksiyonlarını kullanarak filtreleme gerçekleştirir. Bu fonksiyonun sırası, filtrelemede kullanılan katsayı sayısına karşılık gelir. FIR filtreleri, IIR filtrelerinden daha az karmaşık olma eğilimindedir. FIR filtrelerinin avantajları arasında doğrusal faz yanıtı, kararlılık, doğrusallık, tasarım basitliği ve dürtü yanıtı uyumluluğu yer alır. Bu özellikler, FIR filtrelerini çeşitli uygulamalar için popüler bir filtreleme seçeneği haline getirir.

FIR filtreleri haberleşme sistemleri, akustik sinyal işleme, biyomedikal sinyal işleme, görüntü işleme gibi birçok alanda kullanılmaktadır. FIR filtre tasarımı, Matlab gibi dijital sinyal işleme yazılımları kullanılarak kolayca yapılabilir.

Bir FIR filtresinin matematiksel modeli, sabit katsayılı doğrusal bir sistem fonksiyonu olarak ifade edilebilir. Bu fonksiyon, filtrelemede kullanılan katsayıların bir ürünü olarak ifade edilir. FIR filtre tasarımının amacı, belirli bir frekans yanıtı elde etmek için uygun katsayıları seçmek veya hesaplamaktır.

Özetle, FIR filtreleri, sayısal sinyal işleme uygulamalarında yaygın bir filtreleme seçeneği olarak kullanılan sabit katsayılı filtrelerdir. FIR filtrelerinin avantajları arasında doğrusallık, kararlılık, doğrusal faz tepkisi ve değiştirilebilirlik yer alır. FIR filtre tasarımı, Matlab gibi dijital sinyal işleme yazılımları kullanılarak kolayca yapılabilir.

FIR ve IIR filtrelerinin en temel farkı, FIR filtrelerinin geçmiş giriş değerlerine bağlı olmayan, yalnızca şu anki giriş değerine ve geçmiş çıktı değerlerine bağlı olan

geçici yanıt (Impulse Response) fonksiyonlarına sahip olmasıdır. Bu nedenle FIR filtrelerine geçici yanıtta sonlu uzunluklu (Finite Impulse Response) filtreler denir. IIR filtreleri ise, yalnızca şu anki giriş değerine ve geçmiş giriş-çıkış değerlerine bağlı olmayan, geçmiş çıkış değerleriyle de bağlantılı olan geçici yanıt fonksiyonlarına sahiptir. Bu nedenle IIR filtrelerine geçici yanıtta sonsuz uzunluklu (Infinite Impulse Response) filtreler denir (Altın,2013).

2.2.3 Matlab Üzerinde Dijital Sinyal İşleme

Matlab’da dijital sinyal işleme genellikle şu adımlarla gerçekleştirilir:

1. **Sinyal Alımı:** Dijital sinyal işlemede, analog sinyalin öncelikle örnekleme yoluyla dijital forma dönüştürülmesi gerekir. Bu prosedürde örneklenen sinyaller, Matlab’da vektörler olarak tanımlanır.
2. **Sinyal Görselleştirme:** İşlenmekte olan sinyali görselleştirebilir ve gözlemleyebilirsiniz. Sinyalin zaman bölgesi ve frekans bölgesi temsillerinin grafikleri oluşturulur. Bu prosedür, bir dizi Matlab grafik araçları ve işlevleri kullanır.
3. **Sinyal Ön İşleme:** Ön işleme, sinyal işleme, filtreleme, kaynak gürültüsünü azaltma, örnek azaltma ve sinyal yükseltmeyi içerir. Sinyali daha doğru bir şekilde analiz etmek ve sonraki işlemenin verimliliğini arttırmak için bir ön işleme adımı gereklidir.
4. **Sinyal İşleme:** Sinyal işleme adımları, sinyaller üzerinde istenilen işlemleri gerçekleştirir. Bu işlemler, örneğin, frekans analizi, sinyal modülasyonu, ses işleme, görüntü işleme, veri sıkıştırma ve diğer işlemleri içerir. Matlab, çeşitli araçlar ve işlevler sağlayarak bu işlemleri gerçekleştirmemize yardımcı olur.
5. **Sinyal İşleme Sonrası:** Son işleme, sinyal işlemeden sonra daha fazla analiz için sinyalleri yeniden düzenleme, filtreleme ve hazırlama işlemidir. Bu noktada işlenen sinyal tekrar görselleştirilir ve sonuçlar yorumlanır.

Matlab, dijital sinyal işlemeyi kolaylaştırmak için birçok araç ve işlev sağlar. Kullanıcılar bu işlemleri bir grafik kullanıcı arayüzü ve Matlab kodu aracılığıyla gerçekleştirebilirler.

FIR filtreleri, dijital sinyallerin işlenmesinde yaygın olarak kullanılan bir tür filtredir. Bu filtreler, giriş sinyalini belirli bir şekilde değiştirerek çıkış sinyalini oluştururlar. FIR filtreleri, özellikle düşük geçişli, yüksek geçişli ve bant geçiren filtreler gibi belirli frekans aralıklarındaki sinyalleri kesmek veya geçirmek için kullanılırlar. FIR filtreleri tasarlamak ve uygulamak oldukça kolaydır. Matlab gibi yazılım araçları kullanarak FIR filtreleri tasarlamak mümkündür. Bu sayede, FIR filtrelerinin tasarımı için gerekli adımlar ayrıntılı olarak tasarlanabilir. FIR filtreleri, diğer filtrelere göre daha az karmaşıktır ve daha iyi performans gösterirler. Ayrıca, FIR filtreleri lineer faz karakteristiğine sahiptirler ve bu nedenle zaman gecikmesi olmadan çalışabilirler. Bu özellikleri sayesinde FIR filtreleri, ses işleme, görüntü işleme ve diğer birçok uygulama alanında yaygın olarak kullanılırlar (Varghese&Yadav&Bhatia,2021).

2.2.4 Python ve C Dilleri ile Dijital Efekt Oluşturma

Python, üst düzey nesne yönelimli bir programlama dilidir. Hazır fonksiyonların farklı operasyon alanları için kullanılabilmesi için geniş bir kütüphane sistemleri yelpazesi sunar. Bu şekilde Python başta veri bilimi, yapay zeka, web geliştirme, yazılım testi gibi alanlar olmak üzere pek çok alanda kullanılmaktadır. Python, basit okunabilir söz dizimi ile programlamayı kolaylaştırır. Öğrenmesi ve kullanması kolay olduğu için eğitim amaçlı kullanım için de idealdir.

C programlama dili, özellikle sistem programlama için tasarlanmış düşük seviyeli bir programlama dilidir. Temel olarak bellek yönetimi ve donanım kontrolleri gibi süreçleri yönetir. C, verimlilik ve hız açısından diğer dillerden üstündür. İşletim sistemleri, derleyiciler, araçlar, grafik arayüzler ve ağ protokolleri dahil olmak üzere birçok sistem yazılımı C dilinde yazılmıştır. C ayrıca birçok programlama paradigması için işlevsellik sağlar, bu nedenle çok çeşitli projelerde kullanılabilir.

Müzik efektleri oluşturmak için hem Python hem de C kullanılabilir. Bu dillerin sağladığı farklı kütüphaneler, farklı müzik işleme işlevleri için kullanılabilir. Örneğin, Python'un PyDub kitaplığı, ses dosyaları için çeşitli efektler ve işlevler sağlar. C dilinde ses işleme için Poppler kütüphanesini kullanabilirsiniz. Poppler, ses dosyalarını okumak ve düzenlemek için uygun bir arayüz sağlar. Bu, Python ve C programlama dillerinin müzikal efektler oluşturmak gibi çeşitli görevler için kullanılmasına izin verir. Her dilin kendi güçlü ve zayıf yönleri vardır ve seçim projenizin özelliklerine bağlıdır.

Python ve C, farklı tasarımlarla ve felsefelerle geliştirilmiş iki farklı programlama dili olmasına rağmen, birbirleriyle karşılaştırılabilir. Python, açık kaynak kodlu, yüksek seviyeli bir dilken, C daha düşük seviyeli ve derleyici tabanlıdır. Python, temiz bir sözdizimine sahiptir ve C'ye göre daha okunaklıdır. C dilinde bellek yönetimi doğrudan programcının sorumluluğundadır, Python ise bellek yönetimini otomatik olarak gerçekleştirir. Ayrıca, Python, C'nin aksine kod yazma süresini kısaltır ve geliştirme sürecini hızlandırır. Bununla birlikte, C dilinde yazılan programlar daha hızlı ve daha verimlidir. Hangi dilin kullanılacağı, projenin gereksinimlerine ve özelliklerine bağlıdır. C dilinde ses işleme kütüphaneleri oldukça sınırlıdır. Ancak, yine de C dili, özellikle düşük seviye işlemler için oldukça güçlüdür. Bununla birlikte, Python dilinde birçok ses işleme kütüphanesi mevcuttur. Örneğin, Python'da PyAudio, SoundFile, librosa, scikit-sound, wave ve daha birçok kütüphane vardır. Bu kütüphaneler, Python programcılarına ses işleme işlerinde büyük kolaylıklar sağlar (Smith&Rodet,2014).

2.2.5 Simulink: Matlab Tabanlı Grafiksel Simülasyon Ortamı

Simulink, MathWorks tarafından geliştirilmiş bir grafik simülasyon ortamıdır. Simulink, fiziksel sistemleri modellemek ve simüle etmek için bir araçtır. Elektrik mühendisliği, makine mühendisliği ve otomotiv mühendisliği gibi birçok alanda kullanılmaktadır. Simulink, blok diyagramlardan modeller oluşturmanıza olanak sağlar. Bu blok şemalar, çalışma fonksiyonlarını, bağlantıları ve sinyal akışını açıkça gösterir. Simulink bir grafik kullanıcı arayüzüne sahiptir ve Matlab programlama dili

ile entegredir. Simulink, Matlab fonksiyonları ve matris işlemleri dahil olmak üzere Matlab programlama dili ile tamamen uyumludur. Matlab kullanıcıları bu nedenle Matlab kodunu Simulink modellerine entegre edebilir ve Matlab araç çubuğundan modelleri yönetebilir.

Simulink ayrıca modelleme ve simülasyon için araçlar sağlar. Matlab ve Simulink'te bulunan araçların çoğu, sinyal işleme, optimizasyon ve veri analizi gibi birçok alanda kullanılabilir. Simulink, yalnızca modelleme ve simülasyon sırasında araçlar sağlamakla kalmaz, aynı zamanda kod oluşturma ve kod doğrulama için de araçlar sağlar. Bu, modelinizi test etmek ve doğrulamak için kod oluşturmaya otomatikleştirmenizi sağlar.

Simulink ayrıca diğer modelleme araçlarıyla da etkileşime girer. Modelleme araçlarından verileri Simulink'e aktarabilir ve Simulink modellerinin çıktılarını diğer araçlarda kullanabilirsiniz. Bu özellik, sistemin gerçek zamanlı simülasyonunu sağlar. Simulink, sürücülere, motorlara ve diğer donanım modüllerine bağlanmanızı sağlayan bir araçtır.

Özetle Simulink, Matlab entegre bir grafik simülasyon ortamıdır. Simulink, blok şemaları kullanarak fiziksel sistemleri modellemenizi ve simüle etmenizi sağlar. Simulink, modelleme ve simülasyon araçlarının yanı sıra kod oluşturma ve doğrulama araçları sağlar. Simulink, gerçek zamanlı simülasyonu etkinleştirmek için diğer modelleme araçlarıyla da çalışır.

Simulink, Matlab yazılımı için bir arayüz görevi gören blok diyagramı aracıdır. Simulink, sistem modellemesi, simülasyonu ve model tabanlı tasarımın gerçekleştirilmesi için kullanılan bir grafiksel modelleme aracıdır. Simulink modeli, bir sistemdeki tüm bileşenleri, girdileri, çıktıları ve geribildirim döngülerini tanımlayan blokların birbirine bağlanmasıyla oluşturulur. Bu bloklar, diferansiyel denklemler, doğrusal denklemler, mantık işlemleri, filtreler ve diğer matematiksel işlemleri gerçekleştirebilir. Simulink, modelin geliştirilmesi ve tasarımın optimizasyonu için farklı araçlar ve işlevler sağlamaktadır (Little&Moler,1984).

Simulink, grafik arayüz kullanarak gerçek zamanlı matematiksel modeller ve simülasyonlar tasarlamak ve çalıştırmak için bir araçtır. Bu araç, müzik endüstrisindeki ses düzenleme, efekt oluşturma ve kayıt stüdyolarında kullanılabilir. Simulink'te kullanılan bloklar, ses işleme, sinyal modülasyonu ve filtreleme gibi işlemleri gerçekleştirmek için kullanılabilir. Bu sayede müzik efektleri oluşturabilir veya ses sinyalleri üzerinde çeşitli işlemler gerçekleştirebilirsiniz.

2.2.6 Simulink Kütüphaneleri

Simulink, model tabanlı tasarım geliştirmek için kullanılan bir yazılımdır. Bu yazılımın en önemli özelliklerinden biri, kullanıcıların model oluşturmak için kullanabilecekleri pek çok hazır kitaplık ile birlikte gelmesidir. Kitaplıklar, modelinizi tasarlarken ihtiyacınız olan blokları içerir. Bu bloklar, kullanıcıların modelleri hızlı ve kolay bir şekilde tasarlamasına olanak tanır.

Simulink kütüphaneleri, ihtiyaç duyulan blokların bulunduğu yerlerdir. Bu kütüphaneler, Sinks, Sources, Linear, Nonlinear, Math, Signal Attributes, Logic and Bit Operations ve Lookup Tables gibi farklı kategorilere ayrılmıştır. Her bir kategori, kullanıcıların model tasarımı için ihtiyaç duyacakları blokları içerir. Örneğin, Signal Attributes kategorisinde, kullanıcının modelinde yer alan sinyallerin özelliklerini değiştirebilecek bloklar yer alırken, Math kategorisinde ise matematiksel işlemler için kullanılabilecek bloklar yer almaktadır.

Ayrıca Simulink kütüphanesinde modelinizi tasarlarken ihtiyaç duyacağınız özel bloklar bulunmaktadır. Bu bloklar, kullanıcıların birçok tasarım zorluğunun üstesinden gelmesine yardımcı olur. Örneğin, Embedded Coder Support Package for BeagleBone Black Hardware bloğu, kullanıcının model tasarımını BeagleBone Black Hardware üzerinde çalışacak hale getirmesine olanak tanır. Ayrıca, User-Defined Functions bloğu, kullanıcının kendi fonksiyonlarını tasarlamasına olanak sağlar.

Simulink kütüphaneleri, çeşitli endüstrilerdeki mühendislerin ihtiyaçlarını karşılayan birçok kullanıma hazır blok ve model içerir. Bu kütüphaneler otomotiv,

havacılık, savunma, enerji, medikal ve telekomünikasyon gibi sektörlerde kullanılabilir.

Simulink kitaplığı, temel matematik işlemlerinden gelişmiş sinyal işleme ve denetleyici tasarımına kadar çeşitli konularda bloklar içerir. Müzik efektleri için kullanılabilecek kitaplıklar, sinyal işleme, filtreleme, ekolayzırlar ve daha fazlası için bloklar içerir.

Örneğin, Simulink sinyal işleme kitaplığı, sinyal işleme uygulamaları için birçok blok içerir. Bu bloklar, Fourier dönüşümleri. FIR ve IIR filtreleri ve sıralı çarpma gibi birçok sinyal işleme işlevini gerçekleştirebilir. Bu bloklar, müzikal efektler oluşturmak için birleştirilebilir.

Simulink kitaplığında ayrıca müzikal efektler oluşturmak için kullanılabileceğiniz önceden oluşturulmuş modeller bulunur. Örneğin, Simulink Audio Equalizer modeli, basit ekolayzırlar oluşturmak için bloklar içerir. Bu modelde, sinyal işleme blokları, filtre blokları, ekolayzır blokları vb. kullanılarak bir ekolayzır tasarımı gerçekleştirilir.

Simulink kütüphanesi, birçok gelişmiş özelliği içeren hazır bloklarla müzik efektlerinin oluşturulmasını basitleştirir ve zamandan tasarruf sağlar.

2.3 Ses İşleme Algoritmaları: Teori ve Uygulamalar

Ses işleme, ses sinyallerin analiz etmek, sentezlemek ve işlemek için algoritmalar kullanır. Bu algoritmalar farklı amaçlar için kullanılabilir ve ses sinyalinin farklı yönlerini işleyebilir. Bu amaçlar arasında konuşma tanıma, müzik çalma, gürültü azaltma, ses sentezi, tını efektleri oluşturma ve daha pek çok şey yer alır.

Ses işleme algoritmaları, ses sinyalini işleyerek bir ses kaynağından bilgi almayı amaçlar. Belirli özellikleri belirlemek için ses sinyali matematiksel işlemler kullanılarak analiz edilir. Bu özellikler frekans, süre, enerji, tını, vuruş vb. içerebilir.

Ses işleme algoritmaları çeşitli matematiksel teknikler kullanır. Bu teknikler arasında Fourier dönüşümü, cepstral analiz, filtreleme, modülasyon, sentezleme, sınıflandırma ve daha pek çok şey yer alır.

Fourier dönüşümü, bir ses sinyalinin frekans bileşenlerine ayrıştırmak için kullanılabilen bir tekniktir. Bu teknik, bir ses sinyalinin spektrumunu çıkarmak için kullanılır. Cepstral analiz, bir ses sinyalinin tınısını belirlemek için kullanılan bir tekniktir. Bu teknik, bir ses sinyalinin mel-spektrumunu çıkarmak için kullanılır.

Filtreleme, ses sinyallerindeki gürültüyü azaltmak için kullanılan bir tekniktir. Bu teknik, bir ses sinyalinin bileşenlerini belirli bir frekans aralığında çıkarmamızı sağlar. Modülasyon, ses sinyallerini diğer sinyallerle karıştırmak için kullanılan bir tekniktir. Bu teknik, ses sinyaline başka bir sinyal eklenerek elde edilir.

Sentezleme, yeni ses sinyalleri oluşturmak için kullanılan bir tekniktir. Bu teknik, belirli frekans ve zaman bileşenlerinden yeni ses sinyalleri oluşturmayı mümkün kılar. Sınıflandırma, ses sinyallerinin belirli özelliklere göre farklı sınıflarda sınıflandırılmasını sağlar. Bu teknoloji, konuşma tanıma ve müzik sınıflandırma gibi uygulamalar için kullanılabilir.

Ses işleme algoritmaları, ses verilerinin dijital olarak işlenmesinde kullanılan matematiksel ve bilgisayar bilimleri yöntemlerini ifade eder. Bu algoritmalar, ses sinyali filtreleme, kaydetme, sıkıştırma, dinleme, algılama ve diğer birçok uygulama için kullanılır.

Ses işleme algoritmaları, işlemenin çeşitli aşamalarında kullanılan bir dizi teknik ve adım içerir. Bu işleme adımları, örnekleme, çerçeveleme, FFT, FIR ve IIR filtreleme, özellik çıkarma, algılama, kodlama ve sıkıştırmayı içerir.

Örnekleme, ses sinyalinin dijital olarak temsil edilebilmesi için bir analog sinyali düzenli aralıklarla örneklemei ifade eder. Bu örnekleme işlemi, ses sinyallerinin dijital olarak işlenmesi için gereklidir. Öte yandan çerçeveleme, bir ses sinyalinin onları işlemek için kullanılan daha küçük parçalara ayırma işlemidir.

FFT, bir ses sinyalinin frekans içeriğini analiz etmek için kullanılan bir tekniktir. FIR ve IIR filtreleme, sinyal işlemede yaygın olarak kullanılan dijital filtreleme yöntemleridir. Özellik çıkarımı, algılama ve sınıflandırma için kullanılan

ses sinyallerinin özelliklerini analiz etme yöntemidir. Konuşma tanıma, bir ses sinyalinin belirli bir kelime veya ifade gibi önceden tanımlanmış bir şeyle eşleşip eşleşmediğini belirleme işlemidir. Ses kodlama ve sıkıştırma, dijital ses sinyallerini saklama ve iletme yöntemleridir.

Ses işleme algoritmaları birçok uygulama alanında kullanılmaktadır. Ses işleme algoritmalarına yönelik araştırmalar devam etmektedir ve yeni teknikler ve yöntemler geliştirildikçe bunların ses işleme alanındaki uygulamaları da artmaktadır.

Ses işleme alanında sıklıkla kullanılan tekniklerden biri, sinyal işleme tekniklerinin bir alt dalı olan zaman-frekans analizidir. Zaman-frekans analizi, bir sinyalin frekans bileşenlerinin zamana göre nasıl değiştiğini gösteren bir gösterimdir. Zaman-frekans analizi sıklıkla, özellikle müzik işlemede, müzik sinyallerinin frekans spektrumunun zaman içindeki değişimini incelenmesi amacıyla kullanılmaktadır. Bu amaçla, zaman-frekans analizinde, sinyaller genellikle kısa zamanlı Fourier dönüşümü (STFT) veya sürekli dalga dönüşümü (CWT) gibi teknikler kullanılarak zaman-frekans uzayına dönüştürülür. Bu zaman-frekans gösterimleri, müzik işlemede yaygın olarak kullanılan birçok teknik ve algoritmanın temelini oluşturur (Yılmaz,2017).

2.3.1 Gerçek Zamanlı Ses İşleme

Gerçek zamanlı ses işleme, ses sinyallerini işlemeye yönelik bir teknolojidir ve özellikle canlı müzik performansı, konuşma tanıma, müzik üretimi ve otomatik alarm sistemleri gibi uygulamalarda önemlidir. Gerçek zamanlı ses işleme, sinyalin gerçek zamanlı olarak işlenmesi gerektiğinden çok hızlı ve verimli işleme teknikleri gerektirir.

Gerçek zamanlı ses işleme, gelen sinyalleri hızlı bir şekilde işlemek için özel teknikler gerektirir. Bu teknikler arasında zaman-bölge yöntemleri, frekans-bölge yöntemleri, dalga biçimi analizi, spektral analiz ve Fourier dönüşümü bulunur.

Tarih boyunca, gerçek zamanlı ses işleme için çeşitli algoritmalar ve teknikler geliştirilmiştir. Genetik algoritmalar, yapay sinir ağları, bulanık mantık, çevreleyen

algoritma, nörofizyoloji vb. teknolojiler yer almaktadır. Bu teknolojiler, sinyal işleme ve analizi için kullanılan çeşitli yöntemler ve algoritmalar sunar.

Gerçek zamanlı ses işleme, birçok farklı uygulama için kullanılabilir. Müzik prodüksiyonu, kayıtları düzenlemek ve efektler eklemek için gerçek zamanlı ses işleme teknikleri kullanılır. Canlı müzik performansları için gerçek zamanlı ses işleme teknolojisi, çeşitli efektler ve yankılanmalar ekleyebilir. Otomatik alarm sistemleri, gerçek zamanlı ses işleme teknolojisini kullanarak bilinmeyen veya potansiyel olarak tehlikeli sesleri algılayabilir.

Gerçek zamanlı ses işleme teknikleri, işlem gücü, yüksek veri işleme gücü ve yüksek hesaplama gücü gerektiren algoritmaları içerir. Bu nedenle, gerçek zamanlı ses işleme uygulamaları genellikle özel donanım ve yazılımlar kullanılarak gerçekleştirilir.

Bu nedenle gerçek zamanlı ses işleme, düşük gecikmeli, yüksek hızlı geri bildirim uygulamaları için kullanılan bir ses işleme teknolojisidir. Bu tür uygulamalar, FFT, DCT, FIR ve IIR filtreleri, cepstral analiz, frekans analizi, zamanlama analizi, spektral analiz, gürültü azaltma, özellik çıkarma ve sınıflandırma yöntemleri gibi teknikleri kullanır. Bu teknikler, özel donanım ve yazılım kullanılarak gerçekleştirilir.

Gerçek zamanlı ses işlemesi, genellikle ses sinyallerini yakalayan ve bunları yazılım tarafından işlenebilecek dijital bir formata dönüştüren bir donanım bileşeni olan bir ses arayüzü aracılığıyla gerçekleştirilir. Ses arayüzü, dijital sinyalleri hoparlörler veya diğer ses cihazlarından çıkarmak için analog sinyallere dönüştürebilir. Gerçek zamanlı ses işlemesi, bir ses sinyalinin ses arayüzüne girişi ile işlenmesi ve çıktı vermesi arasındaki zaman gecikmesi olan düşük gecikme süresi gerektirir. Gecikme süresi, sinyal işleme algoritmalarının optimize edilmesi, dijital sinyal işlemcileri (DSP'ler) gibi özel donanımların kullanımı veya düşük gecikmeli ses arayüzlerinin kullanımıyla azaltılabilir (Karthick&Kumar,2019).

2.3.2 Ses İşleme Algoritmalarının Matematiksel Temelleri ve Gerçek Zamanlı Uygulamaları

Ses sinyalleri matematiksel olarak zamanla değişen dalga biçimleri olarak görüntülenebilir ve bu dalga biçimleri matematiksel olarak işlenebilir. Örneğin, sinyallerin frekans analizi için Fourier dönüşümleri, sinyallerin zaman-frekans karakteristikleri için Gabor dönüşümleri gibi çeşitli matematiksel dönüşümler kullanılabilir. Ses sinyallerinin işlenmesinde kullanılan birçok filtreleme ve işleme tekniği matematiksel olarak da ifade edilebilmektedir. Örneğin, düşük geçiren filtreler, yüksek geçiren filtreler, bant geçiren filtreler ve daha karmaşık filtreler matematiksel ifadeler kullanılarak açıklanabilir. Ayrıca, matematiksel sinyal işleme teorisi (Signal Processing) gibi alanlar, ses işleme algoritmalarının matematiksel temelini oluşturan temel kavramları ve teknikleri sağlar.

Bazı ses işleme algoritmalarının formülleri ve formülleri vardır. Ancak her algoritma kendi özelliklerine ve uygulama alanlarına göre farklı formüller kullanabilmektedir.

Örneğin, Fourier dönüşümü yaygın olarak kullanılan bir matematiksel işlemidir. Ses sinyalleri genellikle zaman veya frekans alanında da temsil edilir. Fourier dönüşümü, zaman alanındaki bir sinyali frekans alanına dönüştürme işlemidir. Bu işlem genellikle bir dizi matematiksel formül ve integral denklem kullanılarak yapılır.

Başka bir örnek, ses sinyallerini temizlemek için kullanılan bir spektral alt bant gürültü azaltma algoritması olabilir. Bu algoritma, sinyalin spektral yoğunluğunu belirleyen matematiksel bir formül kullanarak bir sinyali gürültüden arındırmaya çalışır.

Ses sıkıştırma algoritmaları ayrıca matematiksel formüller kullanır. Örneğin, popüler bir sıkıştırma algoritması olan MP3, ses sinyalini bir dizi matematiksel işleme analiz eder ve veri depolamayı azaltmak için bazı frekans bileşenlerini kaldırır. Ancak, her algoritma için formüller ve ifadeler farklıdır ve bazıları oldukça karmaşıktır. Bu nedenle, her bir algoritmanın ayrıntılı bir matematiksel analizi, belirli bir algoritmanın ve kapsamın seçilmesini gerektirir.

Bazı algoritmaların matematiksel ifadeleri:

1. **Fourier Dönüşümü:** Bir sinyalin frekans bileşenlerini analiz etmek için kullanılır. Matematiksel olarak, bir sürekli zamanlı sinyal $f(t)$ için Fourier dönüşümü şu şekildedir:

$$F(\omega) = \int f(t) e^{-j\omega t} dt$$

Burada, $F(\omega)$ frekans bileşenleri (spectrum) temsil eder ve ω radyan frekansıdır. $e^{-j\omega t}$ ise kompleks eksponansiyeldir.,

2. **Wavelet Dönüşümü:** Bu, bir sinyalin frekans ve zamansal bileşenlerini analiz etmek için kullanılan bir dönüşümdür. Matematiksel olarak, bir sürekli zamanlı sinyal $f(t)$ için Wavelet dönüşümü şu şekildedir:

$$W(a,b) = \int f(t) * \psi((t-b)/a) dt$$

Burada, $W(a,b)$ wavelet katsayılarını temsil eder. a ve b , wavelet fonksiyonunun ölçek ve konum parametreleridir. ψ ise ana wavelet fonksiyonudur.

3. **LPC (Lineer Prediktif Kodlama):** Bu, bir sinyalin lineer filtreleme kullanarak modelleme yöntemidir. Matematiksel olarak, bir sinyal $x(n)$ için LPC modeli şu şekildedir:

$$x(n) = \sum a(i) * x(n-i) + e(n)$$

Burada, $a(i)$ lineer filtre katsayılarıdır. $e(n)$ ise kalan hata sinyalidir.

Bu örnekler, ses işleme alanında kullanılan formüllerden yalnızca birkaç örnektir. Ses işleme alanında kullanılan matematiksel formüllerin sayısı çok fazladır ve her biri farklı bir amaca hizmet eder.

2.4 Audio Toolbox

Matlab içerisinde bulunan Audio Toolbox, ses işleme ve analizi için önde gelen yazılım aracıdır. Matlab yazılımının bir aracı olarak sunulan bu yazılım, profesyonel ses mühendisleri, akustikçiler ve araştırmacılar tarafından kullanılır.

Ses işleme, insanlar tarafından duyulan seslerin dijital ortamda işlenmesini ve analiz yapılmasını sağlayan bir alandır. Bu süreçte ses sinyali, örnekleme, niceleme, FFT, gibi temel işlemlerden geçilerek dijital sinyale dönüştürülür ve çeşitli uygulamalar için kullanılır.

Ses işleme temelleri, ses sinyallerinin dijital işlenmesini ve analizini içeren tekniklerin ve yöntemlerin açıklamasıdır. Bu temel kavramlar, örnekleme, niceleme, FFT'ler, filtreler ve zaman/frekans analizini içerisinde bulundurur. (Ozerov&Vincent&Bimbot,2012).

Audio Toolbox, ses sinyalleri oluşturma, kaydetme ve oynatma gibi temel işlemler için bir dizi işlev sağlar. Ayrıca efektler, ölçümler, format dönüştürme ve filtreleme gibi diğer işlemler için de çözümler sunar.

Audio Toolbox ile ses sinyallerini ölçmek ve analiz etmek için birçok işlev sağlar. Bu özellikler arasında ses yoğunluğu, frekans spektrumu, harmonik analiz, biçim analizi ve zaman/frekans analizi yer alır. Bu işlevler, ses sinyalleri hakkında kapsamlı bilgi sağlar ve çeşit çeşit uygulamalar için kullanılabilir. Audio Toolbox ayrıca oldukça çok tını efekti işlevi sağlar. Bu efektler yankılanma, gürültü azaltma, ses normalleştirme, ekolayzır ve konuşma sentezini içerir.

Audio Toolbox ayrıca ses sinyallerinin format dönüşümü için birçok fonksiyon sağlar. Bu işlem, ses sinyallerinin çeşitli biçimlerde kaydedilebilmesi veya oynatılabilmesi gerektiğinde kullanılır. Audio Toolbox, WAV, MP3, AIFF ve diğer popüler ses formatlarını desteklemektedir. (Buza&Todorean&Nica,2006).

Audio Toolbox, kullanıcıların özel ses işleme uygulamaları oluşturabilmelerini sağlar. Bu uygulamalar, farklı uygulama senaryoları için yapılandırılabilir ve özelleştirilebilir. Audio Toolbox, geniş bir kullanıcı kitlesine uygun bir arayüz sunmakta ve farklı kullanım senaryolarına uygun çözümler sunmaktadır. Kullanıcılar,

özel ses işleme uygulamaları oluşturabilir ve Audio Toolbox'ı Matlab yazılımının diğer özellikleriyle entegre edebilir.



BÖLÜM III

VERİ ANALİZİ VE LİTERATÜR TARAMASI

3.1 Literatür Taraması

Dijital tını efektleri alanında yapılmış önceki araştırmalar incelenerek bu alanda hangi yöntemlerin kullanıldığı ve hangi sonuçların elde edildiği araştırılmıştır.

3.2 Veri Toplama

Araştırma kapsamında farklı dijital tını efektleri oluşturulmuş ve bu efektlerin özellikleri kayıt altına alınmıştır.

3.3 Veri Analizi ve Yöntem

Araştırmada, C ve C++ dili içerisinde görev yapan Matlab (Versiyon: R2023a) yazılımı kullanılmıştır. Matlab içerisindeki “Simulink” aracı, efektlerin nasıl yazılabileceğini görsel olarak göstermiş ve bu parametrelerin grafiksel olarak incelenmesi sağlanmıştır. Kullanılan ses dosyalarının hem ham hem de işlenmiş sinyal grafikleri gösterilerek, bu grafiklerdeki değişimler, analizleri ile ortaya konulmuştur. Analiz sonuçları efektlerin nasıl yazılabileceğine dair bir yöntem sunmakta ve bu yöntemin müzik prodüksiyonunda kullanılmasına ilişkin potansiyel artılarını tartışmaktadır.

Bu araştırma nicel bir araştırma yöntemi kullanılarak hazırlanmıştır. Araştırmanın nicel verileri Matlab yazılımının Simulink ve Audio Toolbox araçları kullanılarak elde edilmiştir. Simulink ve Audio Toolbox araçları, efektlerin nasıl yazılabileceğini görsel olarak göstermiş ve bu parametrelerin grafiksel olarak incelenmesini sağlamıştır. Kullanılan ses dosyalarının hem kuru hem de ıslak sinyal grafikleri gösterilerek, bu grafiklerdeki değişimler analiz edilmiş ve yorumlanmıştır.

BÖLÜM IV

BULGULAR VE YORUMLAR

Bu bölüm, elde edilen verilerin analizi yapılarak yorumlanmış ve anlamlılık esasına göre araştırmanın amacına uygun şekilde ortaya konulmuştur. Bu bölümün içerisindeki analizler iki farklı ses dosyası kullanılarak oluşturulmuştur. Bu ses dosyaları gitar ve clap seslerinden oluşmaktadır. Gitar, zengin armonik yapısı ve frekans aralığı ile karakteristik bir enstrümandır bu sebeple gitar sesinin kullanımı melodik ifadeleri vurgulamak için tercih edilmiştir. Clap sesinin kullanımı ise ritmik ifadelerin vurgulanması ve dinamik etkilerin anlaşılması amacıyla tercih edilmiştir.

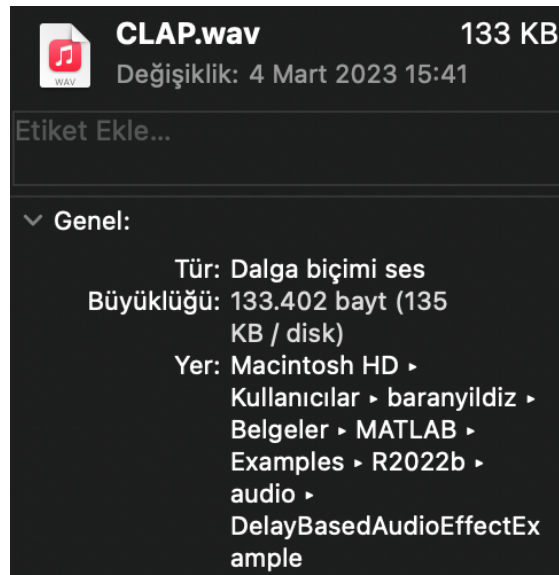
4.1 Birinci Alt Probleme İlişkin Bulgu ve Yorumlar

“Efektlerin oluşturulması için hangi aşamalar izlenmelidir?”

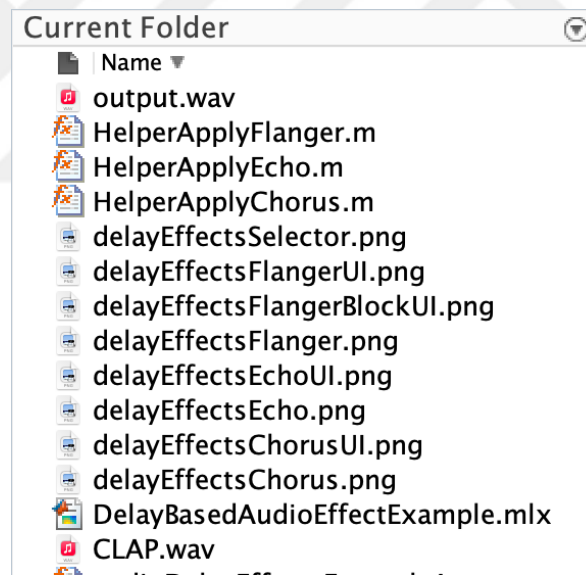
Efektlerin oluşturulabilmesi için öncelikle hangi efektin elde edileceği amaçlanmış ve buna göre hangi ses dosyaları ile çalışılması gerektiği belirlenmiştir. Ardından nasıl bir efekt oluşturulacağına dair bir plan hazırlanmıştır.

4.1.1 Matlab ile Ses İşleme

Matlab programında bir ses dosyasını işlemek için öncelikle kullanılacak ses dosyasının bilgisayarın disk bölümündeki konumu belirlenmiştir. Ardından bu ses dosyası Matlab programına aktarılacak için, dosya fare yardımı ile Matlab programının “Çalışma Klasörü (Current Folder)” bölümüne sürüklenmiştir. Bu işlem sonucunda dosyanın bilgisayar üzerindeki konumu ile Matlab programının çalışma kütüphanesi arasında bir bağlantı kurulmuştur. Bu sayede dosya Matlab programında işlenebilir hale getirilmiştir ve buna göre çeşitli işlemler gerçekleştirilmiştir. Bu işlem sıralaması Şekil 7 ve Şekil 8 üzerinde gösterilmiştir.



Şekil 7 Ses dosyasının konumu

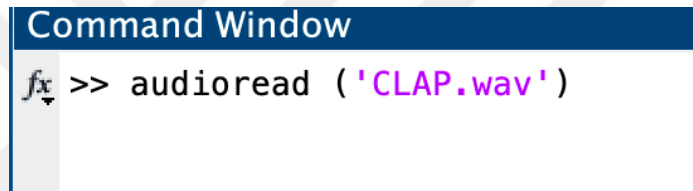


Şekil 8 Ses dosyasının Matlab üzerindeki çıktısı

Dosya belirlendikten sonra Matlab programındaki “audioread” komutu kullanılarak dosya okunmuş ve program içerisindeki “Workspace” kısmına aktarılmıştır. Bu işlem ses dosyasının program içerisinde kullanılabilmesini sağlamaktadır.

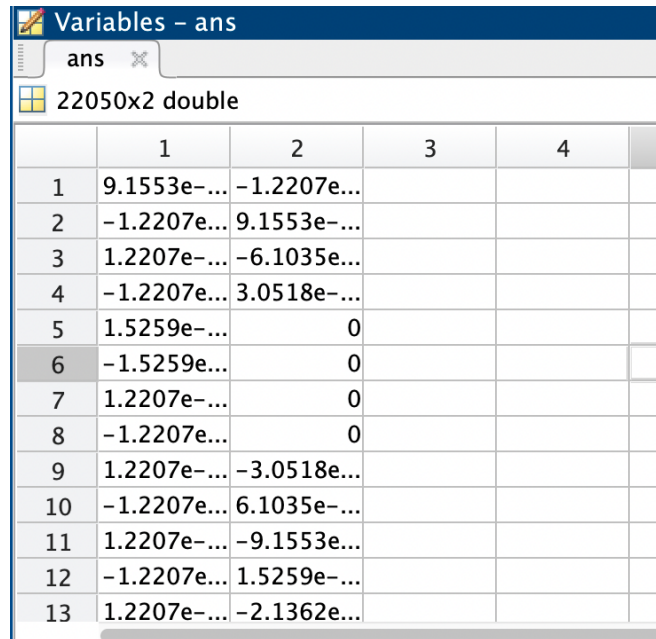
“Audioread” komutu ses dosyasının formatını ve özelliklerini yazılımın gereği olarak otomatik olarak algılamakta ve okunması gereken yolu belirttiğimizde ses dosyasının okunmasını sağlamaktadır. Örneğin, “audioread('CLAP.wav')” şeklinde bir komut, “CLAP.wav” isimli ses dosyasının okunmasını sağlar ve dosyanın özellikleri bir değişkene atanır.

Bunun sonucunda ses dosyasının özellikleri program içinde kullanılabilir hale getirilir ve istenilen analiz veya işlemler gerçekleştirilebilir. Böylelikle ses dosyasının özellikleri program içerisinde kullanılarak farklı efektlerin uygulanması, frekans analizi yapılması gibi işlemler gerçekleştirilebilir hale getirilmiştir. Bu işlem sıralamaları Şekil 9 ve Şekil 10 üzerinde gösterilmiştir.



```
Command Window
fx >> audioread ('CLAP.wav')
```

Şekil 9 “audioread” komutu Matlab çıktısı

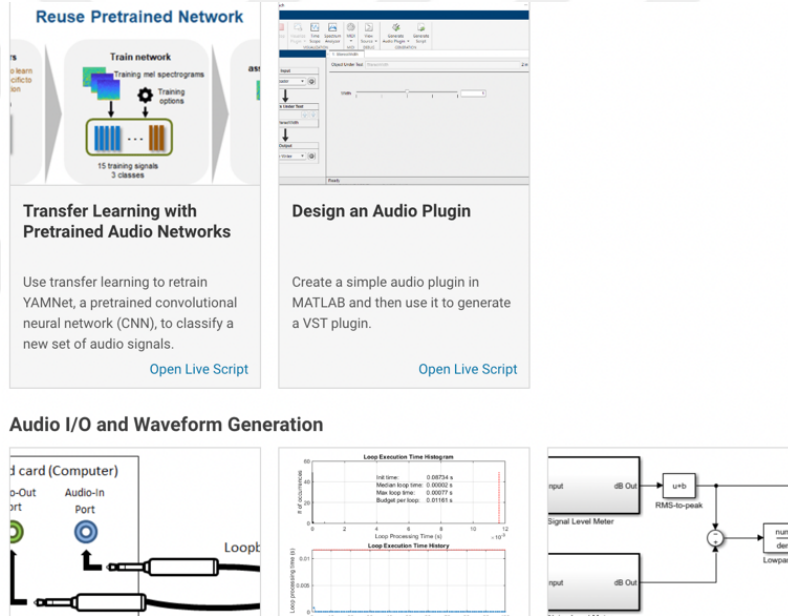


Variables - ans					
ans					
22050x2 double					
	1	2	3	4	
1	9.1553e-...	-1.2207e...			
2	-1.2207e...	9.1553e-...			
3	1.2207e-...	-6.1035e...			
4	-1.2207e...	3.0518e-...			
5	1.5259e-...	0			
6	-1.5259e...	0			
7	1.2207e-...	0			
8	-1.2207e...	0			
9	1.2207e-...	-3.0518e...			
10	-1.2207e...	6.1035e-...			
11	1.2207e-...	-9.1553e...			
12	-1.2207e...	1.5259e-...			
13	1.2207e-...	-2.1362e...			

Şekil 10 Ses dosyasının Workspace üzerindeki çıktısı

4.1.2 Matlab Simulink Yazılımı ile Model Oluşturma

Matlab Simulink yazılımı, modelleme ve simülasyon araçları sağlayan bir yazılımdır. Bu yazılım, gerçek zamanlı simülasyon sağlamak için diğer modelleme araçları ile birlikte çalışır. Modelleme, blok diyagram araçları kullanılarak yapılmıştır. Öncelikle blok diyagramda kullanılacak bloklar seçilmiş ve bu blokların parametreleri belirlenmiştir. Ardından bir model oluşturmak için bu bloklar birleştirilmiştir. Model oluşturma işlemi tamamlandıktan sonra simülasyon yapılabilir hale gelir ve sonuçlar analiz edilebilir. Bu işlem sıralamaları Şekil 11 ve Şekil 12 üzerinde gösterilmiştir.



Şekil 11 Simulink modelleme örnekleri

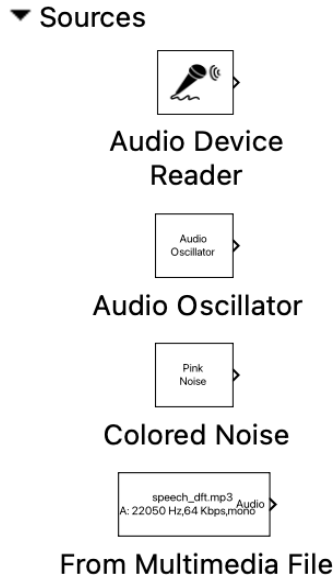


Şekil 12 Örnek bir ses dosyası okuyucu modeli

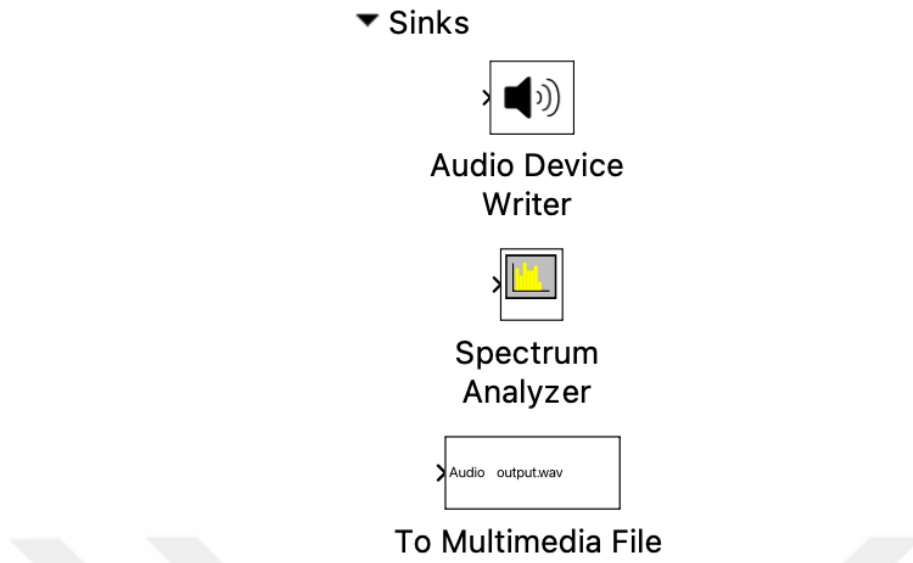
4.1.3 Matlab Simulink ile Ses Dosyalarının Analizi ve Simülasyonu

Simulink yazılımında bir model oluşturabilmek için boş bir model sayfası açılmıştır. Ardından ‘From Multimedia File’ bloğu modelin ilk kısmına yerleştirilmiştir. Blok, parametrelerinin belirlenmesi sonucunda ‘CLAP.wav’ isimli ses dosyasını bloğa aktarmıştır. Daha sonra ‘Audio Device Writer’ bloğu modelin devamına eklenmiş ve iki blok birbirine bağlanmıştır. ‘Audio Device Writer’ bloğu içerisinde yer alan parametreler ayarlanarak ses çıkış cihazı seçilmiştir. Son olarak, Simulink yazılımının simülasyon başlatma aracı kullanılarak ‘CLAP.wav’ isimli ses dosyasının çalınması ve modelin simülasyonunun çalışması gerçekleştirilmiştir.

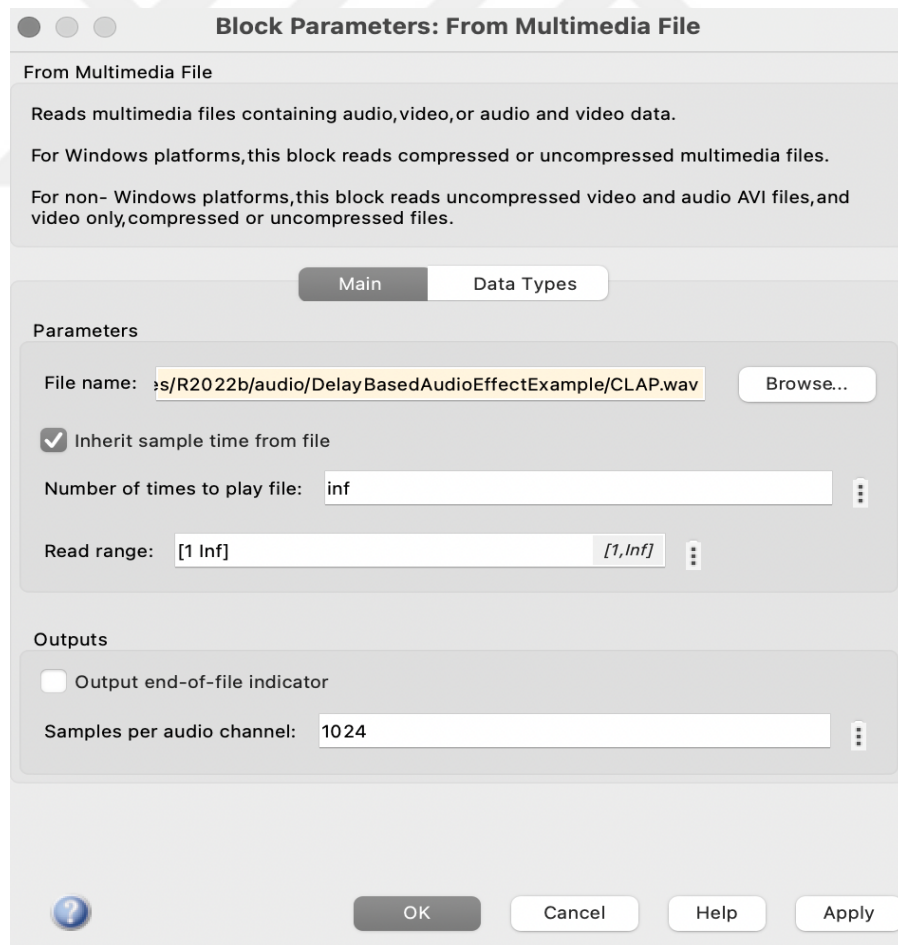
Bu adımlar sayesinde, Simulink yazılımı ile ses dosyalarının işlenmesi ve çalınması için bir model oluşturulmuştur. Modelin oluşturulma sürecinde, blok diyagram aracı kullanarak farklı blokların birbiriyle bağlanması ile oluşturulan bu model, simülasyon aracı ile gerçek zamanlı olarak çalıştırılabilir. Bu sayede modelin doğruluğu ve performansı kolayca analiz edilebilir ve iyileştirmeler yapılabilir. Bu işlem sıralamaları Şekil 13, Şekil 14, Şekil 15 ve Şekil 16 üzerinde gösterilmiştir.



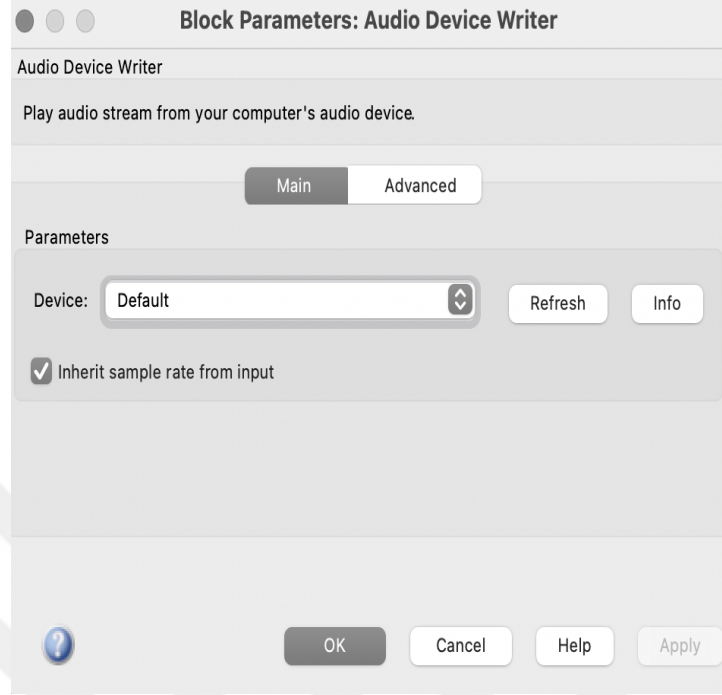
Şekil 13 ‘From Multimedia File’ bloğu



Şekil 14 'Audio Device Writer' bloğu



Şekil 15 'From Multimedia File' blok parametreleri



Şekil 16 ‘Audio Device Writer’ blok parametreleri

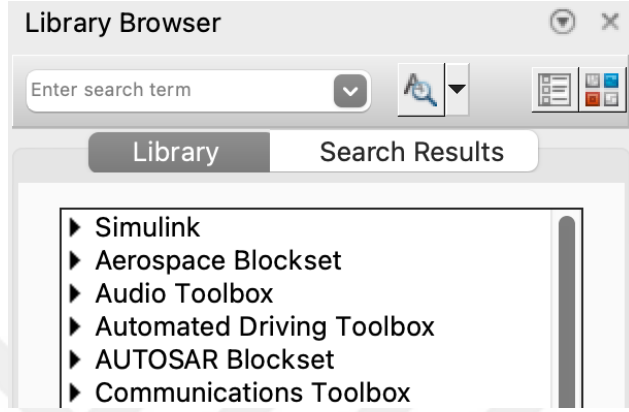
4.1.4 Modelleme Süreçlerine Efekt Bloklarının Eklenmesi

Simulink yazılımı, farklı işlevler için kullanılabilen bir blok kütüphanesi sunar. Bu bloklar, kullanıcıların modelleme işlemini hızlandırmalarına ve özelleştirebilmelerine olanak sağlar. Bu modelde “Library Browser” sekmesi kullanılarak modelleme iş akışı içerisinde “Reverberator” ve “Graphic EQ” blokları kullanılmıştır.

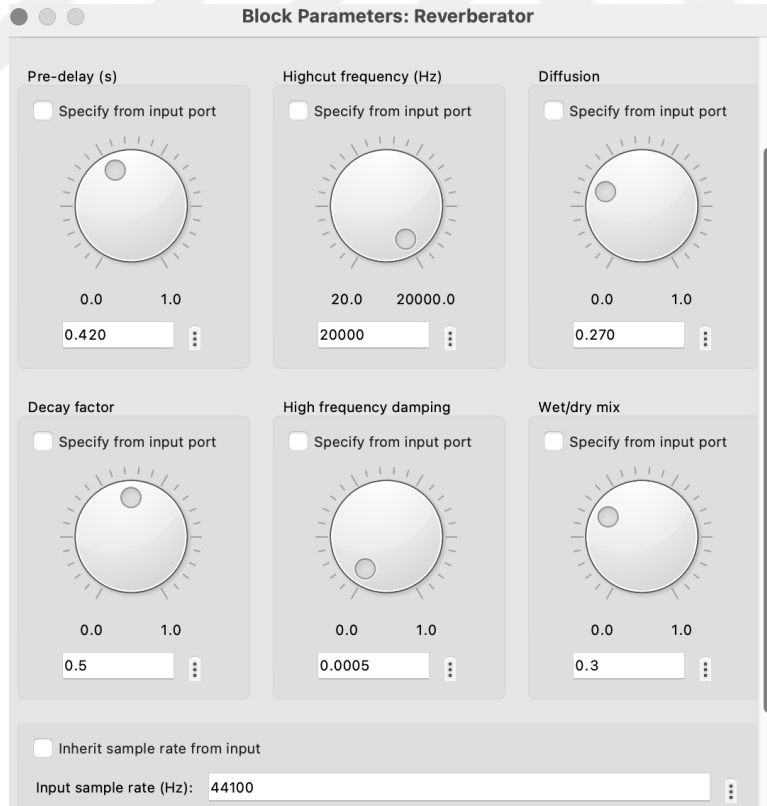
Bu blokların parametreleri incelenmiş ve üzerinde oynamalar yapılmıştır. Örneğin, modelin efekt işleme performansı, “Reverbator” bloğunun “Diffusion” ve “Decay” gibi parametreleri değiştirilerek iyileştirilmiştir. Benzer şekilde, “Graphic EQ” bloğunun “Low Frequencies” ve “High Frequencies” gibi parametreleri de değiştirilmiştir.

Bu analizleri görselleştirebilmek için “Library Browser” sekmesindeki “Spectrum Analyzer” bloğu da modele eklenmiştir. Bu blok, simülasyon sonuçlarının

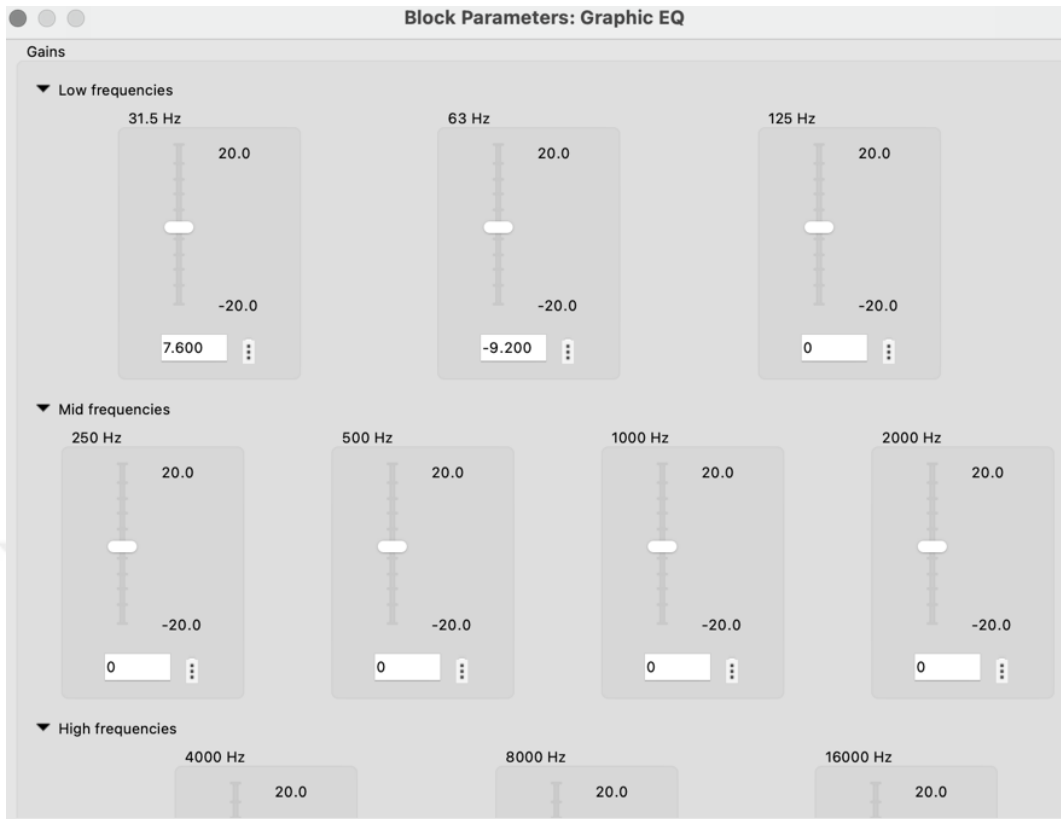
spektral analizini gerçekleştirir ve sonuçları grafiksel olarak sunmamıza imkan tanır. Bu nedenle, modelin performansının daha eksiksiz bir analizi yapılabilir. Bu işlem sıralamaları Şekil 17, Şekil 18, Şekil 19, Şekil 20, Şekil 21 üzerinde gösterilmiştir.



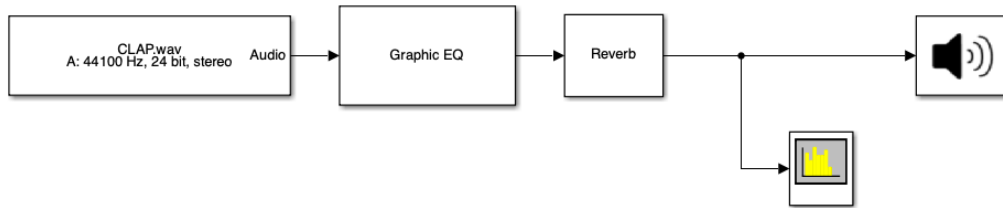
Şekil 17 Simulink “Library Browser” sekmesi



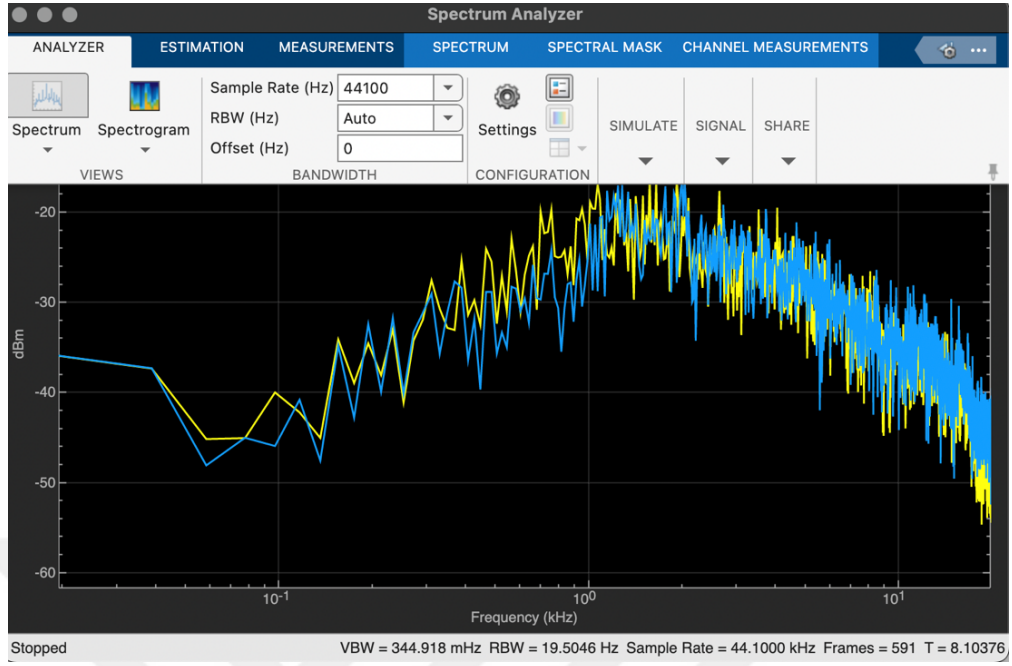
Şekil 18 Reverberator blok parametreleri



Şekil 19 Graphic EQ blok parametreleri



Şekil 20 Model blok dizilimi

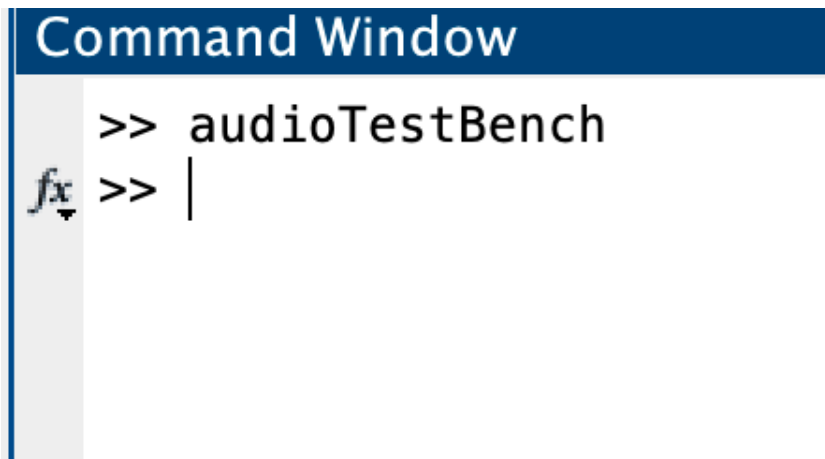


Şekil 21 Modelin spectrum analyzer çıktısı

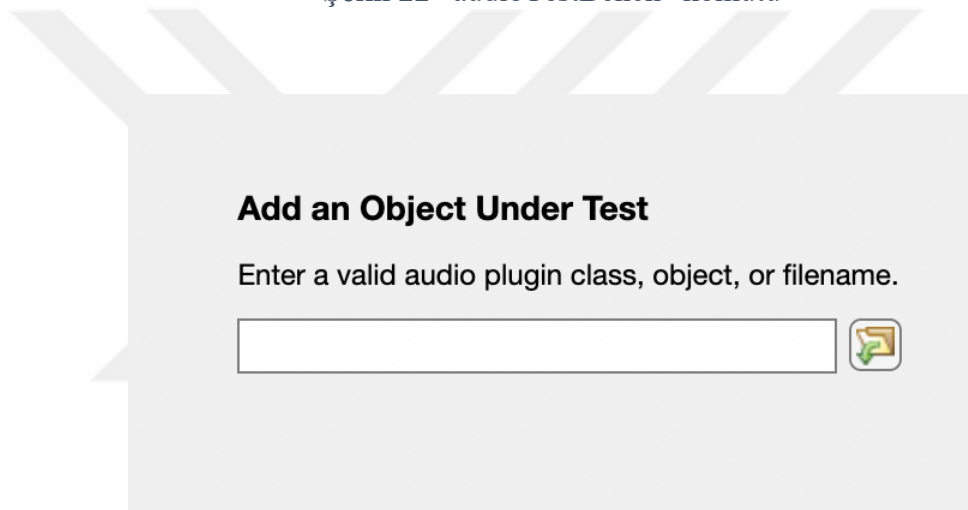
4.1.5 Matlab Audio Test Bench ile Tını Efektleri Analizi ve Uygulaması

Matlab yazılımı, tını efektleri testlerini çalıştırmak için kullanılan bir “Audio Test Bench” sayfası sunar. Audio Test Bench sayfasına “Command Window” içerisinde, “audioTestBench” komutu kullanılarak ulaşılmıştır. Bu sayfada efektleri test edebilmek için “Add an Object Under Test” sekmesi kullanılarak “audiopluginexample.Chorus” plugini eklenmiştir. Daha sonra “Input” sekmesine giriş sinyali olarak “CLAP.wav” adlı bir ses dosyası yerleştirilmiştir ve çıkış sinyali olarak da “Audio Device Writer” komutu seçilmiştir.

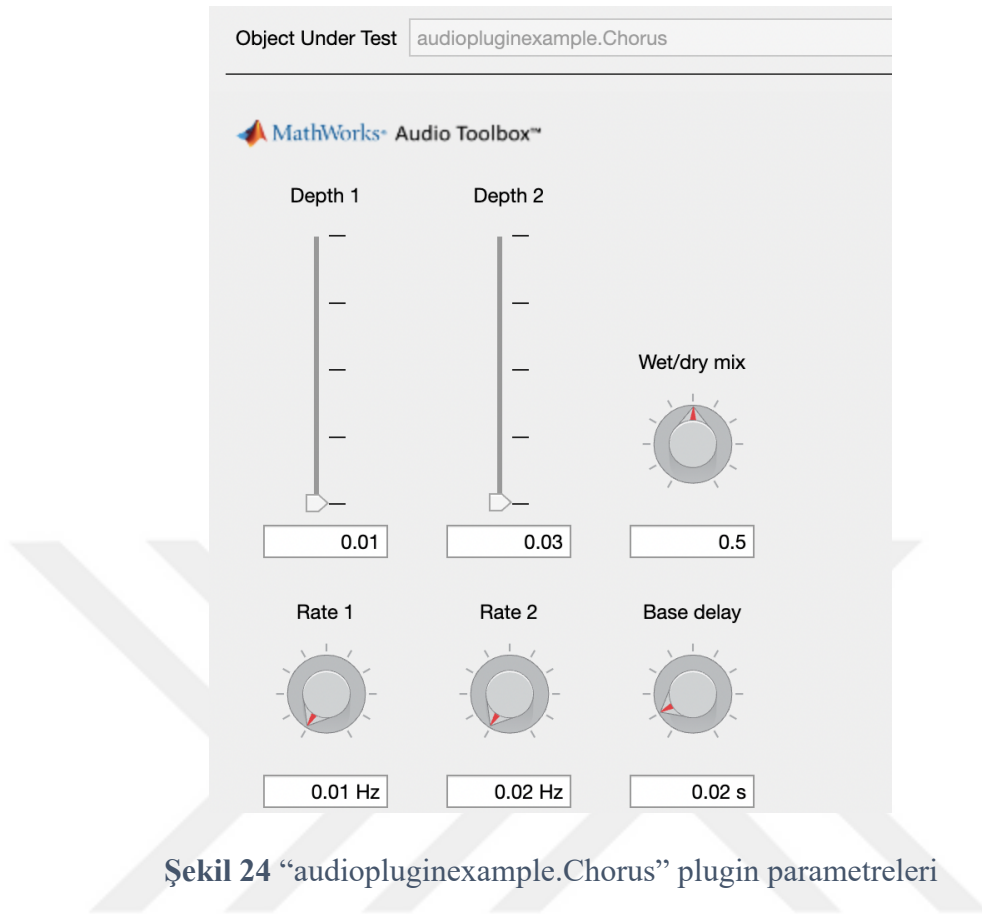
Grafik çıktısı için, sinyalin frekans bileşenleri Audio Test Bench sayfasındaki “Spectrum Analyzer” aracı kullanılarak analiz edilmiştir. Bu işlem sıralamaları Şekil 22, Şekil 23, Şekil 24, Şekil 25, Şekil 26, Şekil 27 üzerinde gösterilmiştir.



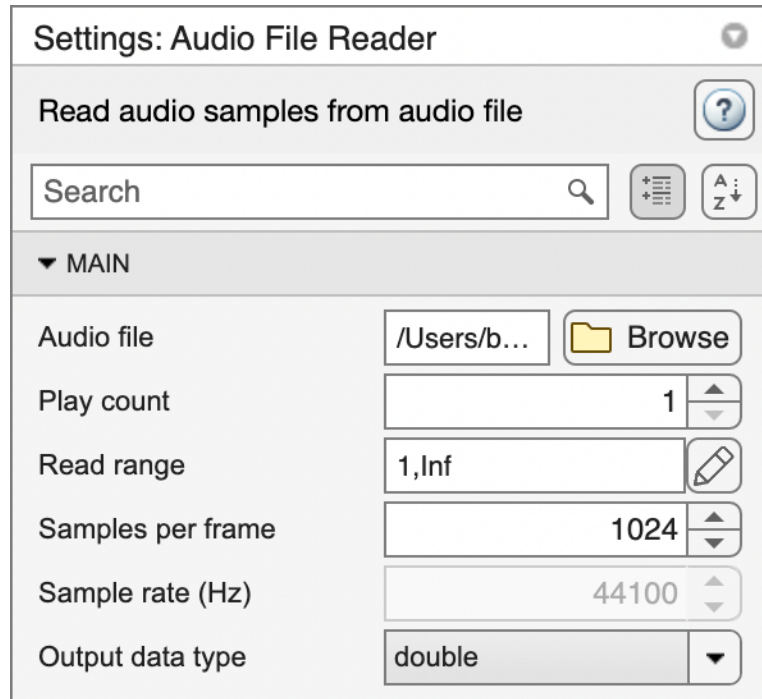
Şekil 22 “audioTestBench” komutu



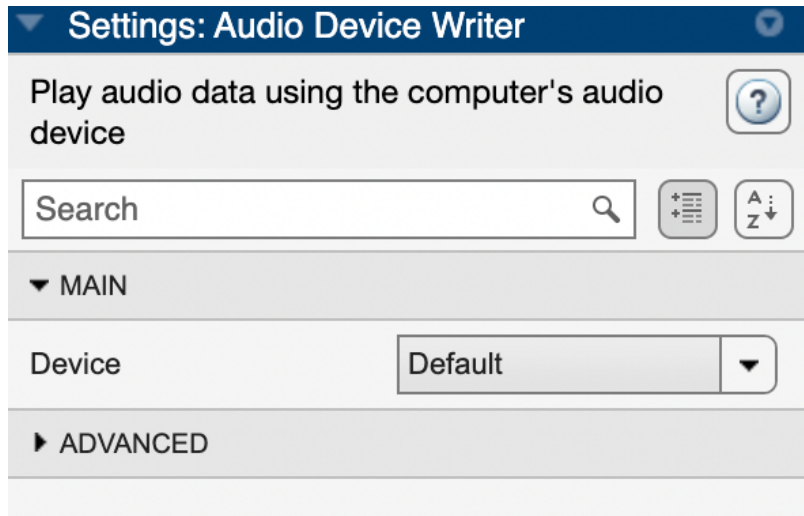
Şekil 23 “Add an Object Under Test” sekmesi



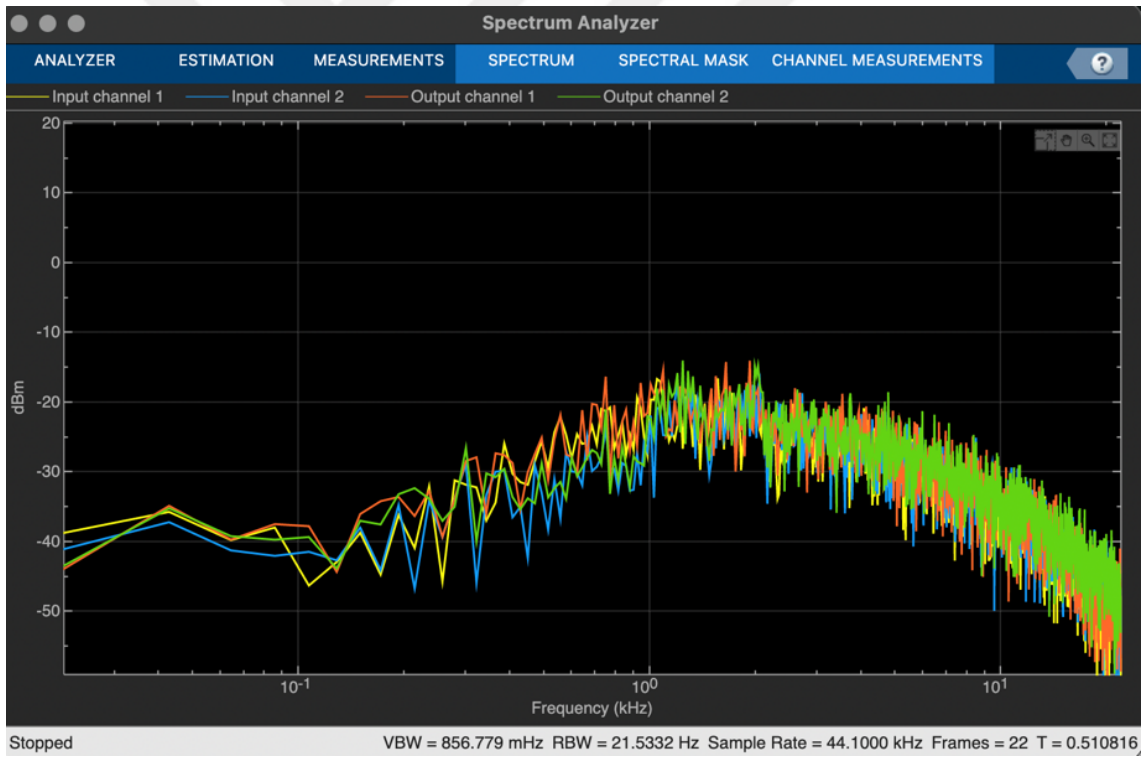
Şekil 24 “audiopluginexample.Chorus” plugin parametreleri



Şekil 25 “Input” sekmesi giriş sinyal parametreleri



Şekil 26 “Output” sekmesi çıkış sinyal parametreleri



Şekil 27 Audio Test Bench üzerinden “Spectrum Analyzer” grafik çıktısı

4.2 İkinci Alt Probleme İlişkin Bulgu ve Yorumlar

“Efekt parametreleri üzerinde durularak, farklı parametre değerlerinin etkileri nasıl analiz edilir?”

Efekt parametreleri, bir tını efektinin çıkış sinyalindeki özelliklerini değiştirerek, istenilen tını efektinin oluşturulmasını sağlar. Farklı parametre değerlerinin etkilerinin analiz edilmesi, efektin ne şekilde çalıştığını daha iyi anlamamıza yardım eder.

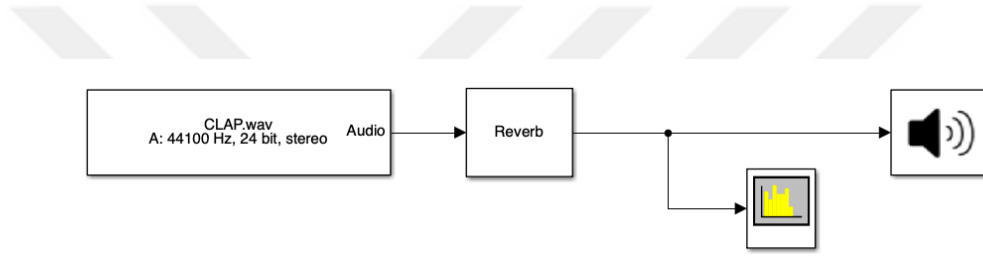
Bu analiz üzerinde, gerçek zamanlı efektlerin Simulink üzerinden “CLAP.wav” isimli ses dosyasına uygulanması ve oluşturulan bu modelin simülasyonu gerçekleştirilmiştir. Farklı parametre değerlerinin ses dosyası üzerindeki etkisi “Spectrum Analyzer” ile grafik çıktıları alınarak analiz edilmiş ve karşılaştırmalar yapılmıştır. Simulink üzerinden kullanılan ve grafik analizleri yapılan efektler; Reverberator ve Graphic EQ’dur.

Daha sonra Audio Test Bench üzerinde örnek efekt pluginleri oluşturulmuş ve bu pluginlerin parametre değerleri değiştirilerek “Spectrum Analyzer” üzerinden grafik çıktıları alınmış ve bu grafikler analiz edilmiştir. Bu analizlerin sonuçları incelenmiş ve karşılaştırılmıştır. Audio Test Bench üzerinden kullanılan ve grafik analizleri yapılan efekt pluginleri; Chorus, Echo, Flanger, Phaser ve Parametric Equalizer’dır.

Bu analizlerin sonuçları, efekt parametrelerinin sinyal özellikleri üzerindeki etkisini daha iyi anlayabilmek için önemlidir. Ayrıca, oluşturulan bu pluginlerin “classdef” üzerindeki kodları ve parametre değerleri de incelenerek, efektlerin nasıl çalıştığı daha iyi anlaşılmıştır.

4.2.1 Matlab Simulink ile Gerçek Zamanlı Tını Efektı Uygulaması ve Analizi

Simulink üzerinden bir ses dosyası okuyucu modeli oluşturulmuştur. Oluşturulan bu modelde, Reverberator ve Spectrum Analyzer blokları kullanılmıştır. “CLAP.wav” isimli ses dosyası da modele eklenerek, simülasyona hazır hale getirilmiştir. Bu model ses işleme işlemlerinin görselleştirilmesine imkan sağlamış ve ses işleme algoritmalarının performansını analiz etmek için kullanılmıştır. Bu model Şekil 28 üzerinde gösterilmiştir.



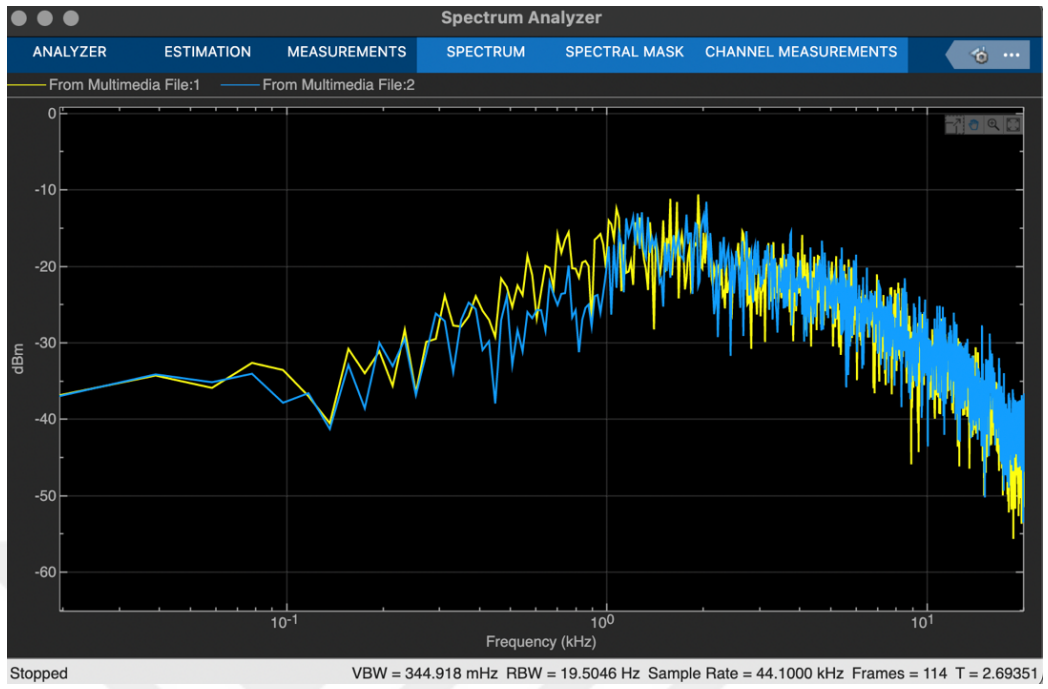
Şekil 28 Reverberator ve Spectrum Analyzer blokları eklenmiş bir ses dosyası okuyucu modeli

4.2.1.1 Reverberator Bloğunun Ses Dosyası Üzerindeki Etkilerinin Grafıksel Analizi

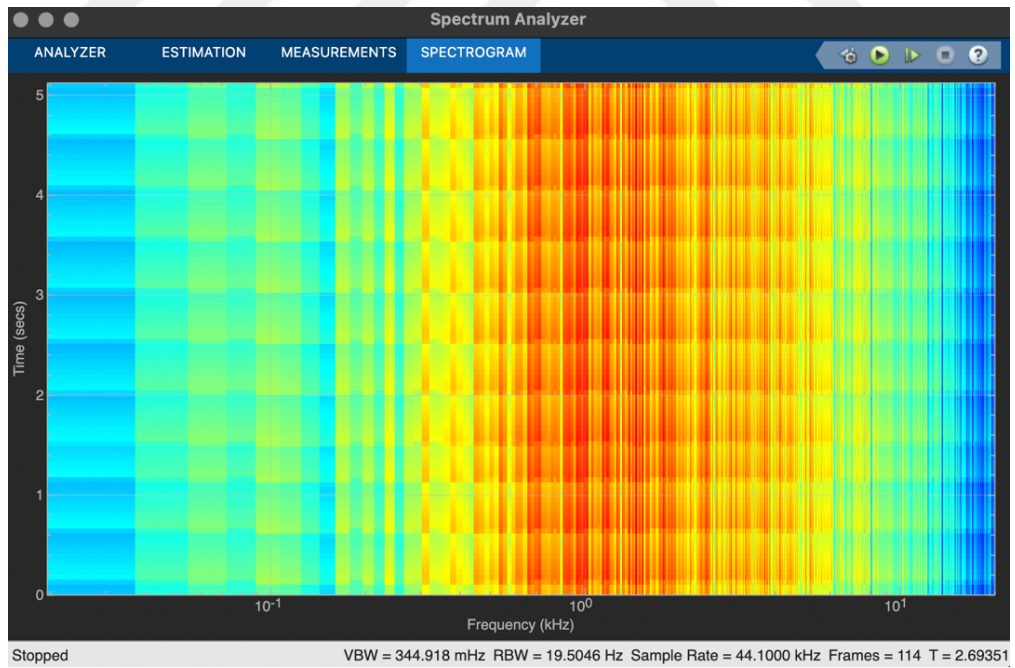
Oluşturulan model üzerindeki “CLAP.wav” isimli ses dosyasının, herhangi bir efekt bloğu eklenmeden önceki grafik analizi aşağıdaki gibidir;

1. “CLAP.wav” isimli ses dosyasının 20 Hz ve 20.000 Hz arasında yaklaşık -60 ve -10 dB aralığında bir sinyale sahip olduğu analiz edilmiştir.
2. Tepe noktasının 1650 Hz bölgesinde yaklaşık -10 dB şiddetinde olduğu, dip noktasının ise 20.000 Hz bölgesinde yaklaşık -60 dB şiddetinde olduğu analiz edilmiştir.
3. Spectrogram üzerinden ses sinyalinin en yoğun olduğu bölge 900 Hz ve 3000 Hz arası olduğu analiz edilmiştir.

Bu analizler Şekil 29, Şekil 30 üzerinde gösterilmiştir.



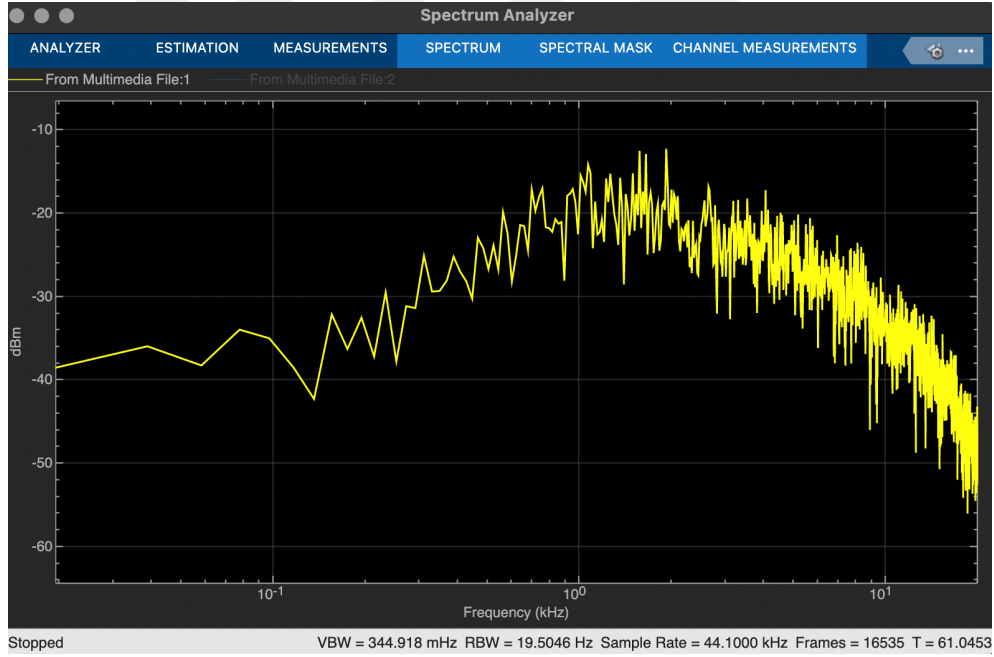
Şekil 29 “CLAP.wav” isimli ses dosyasının Spectrum çıktısı



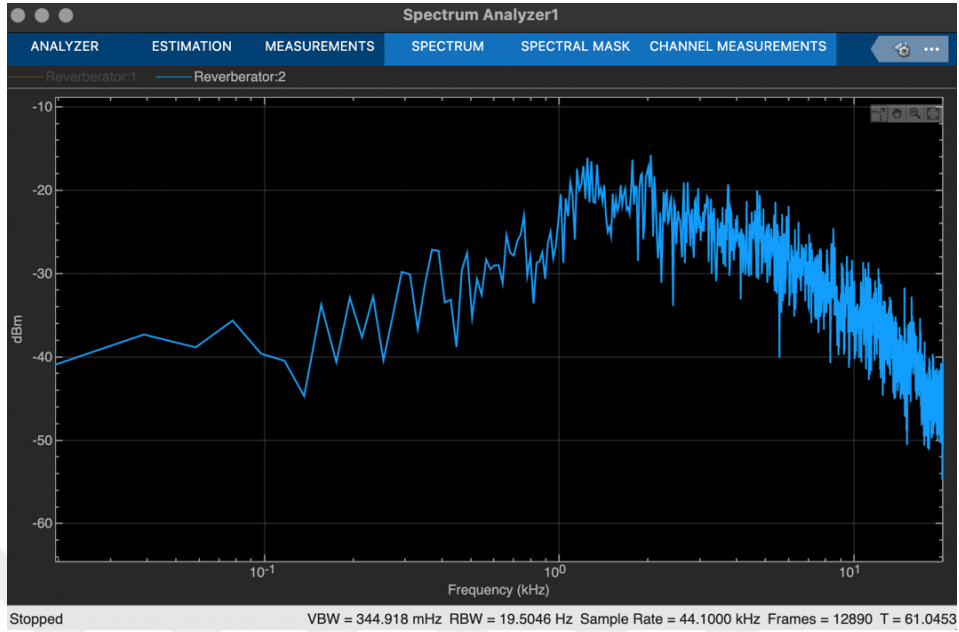
Şekil 30 “CLAP.wav” isimli ses dosyasının Spectrogram çıktısı

Oluşturulan modele Reverberator efekt bloğu eklendikten sonra farklı parametre değerleri üzerindeki grafik analizleri ve karşılaştırmaları aşağıdaki gibidir;

1. Reverberator bloğunun parametre değerleri, Pre-delay:0.25, Highcut Freaquency:20.000, Diffusion:0.5, Decay Factor:0.5, High Frequency Damping:0.0005, Wet/Dry Mix:0.25, Input Sample Rate(Hz):44100 olacak şekilde ayarlanmıştır. Bu parametre değerleri ile Spectrum Analyzer üzerinden analiz yapılmış, 100 Hz bölgesinde ve 20.000 Hz bölgesinde dB farklılıkları gözlenmiştir. Tepe noktası 1930 Hz olarak değişmiştir. Bu parametre değerleri üzerinden Spectrum Analyzer grafiği Şekil 31 ve Şekil 32 üzerinde karşılaştırmalı olarak gösterilmiştir. Sarı sarı ham sinyal grafiğini, mavi renk ise işlenmiş sinyal grafiğini göstermektedir.

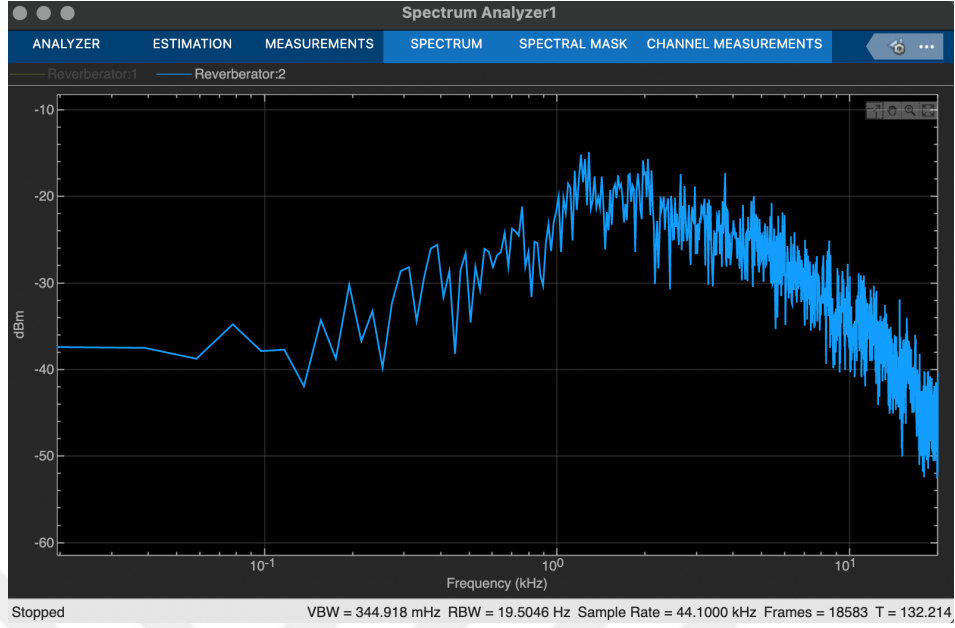


Şekil 31 Ham sinyal Spectrum Analyzer grafik çıktısı



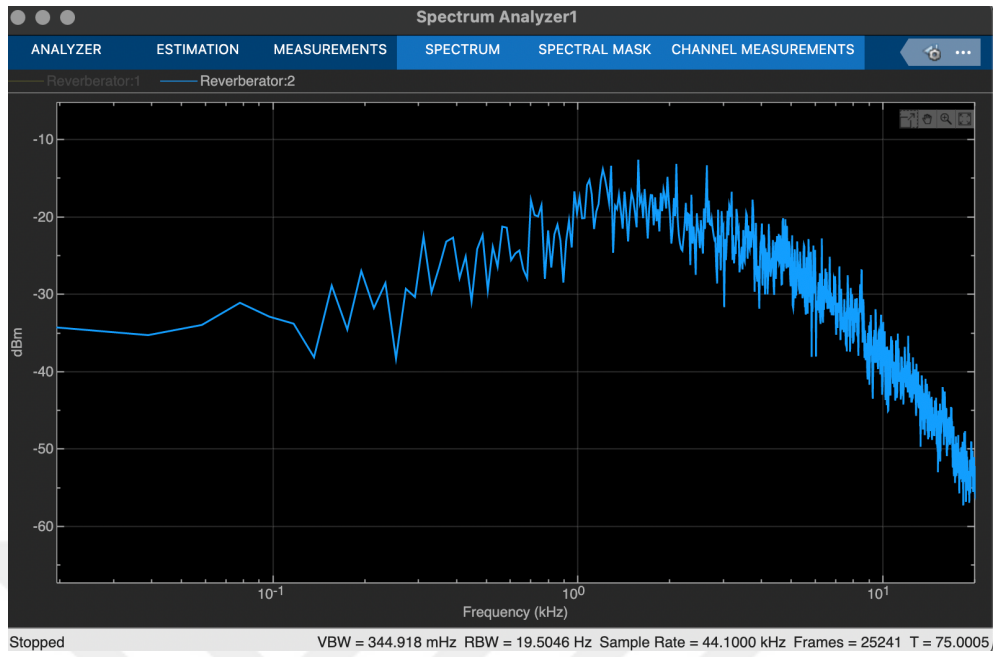
Şekil 32 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

2. Reverberator bloğunun parametre değerleri, Pre-delay:0.5, Highcut Freaquency:10.000, Diffusion:0.5, Decay Factor:0.5, High Frequency Damping:0.0005, Wet/Dry Mix:0.5, Input Sample Rate(Hz):44100 olacak şekilde ayarlanmıştır. Bu parametre değerleri ile Spectrum Analyzer üzerinden analiz yapılmış, 100 Hz ve 10.000 Hz bölgesinde dB farklılıkları gözlenmiştir. Bu parametre değerleri üzerinden Spectrum Analyzer grafiği Şekil 33 üzerinde gösterilmiştir.

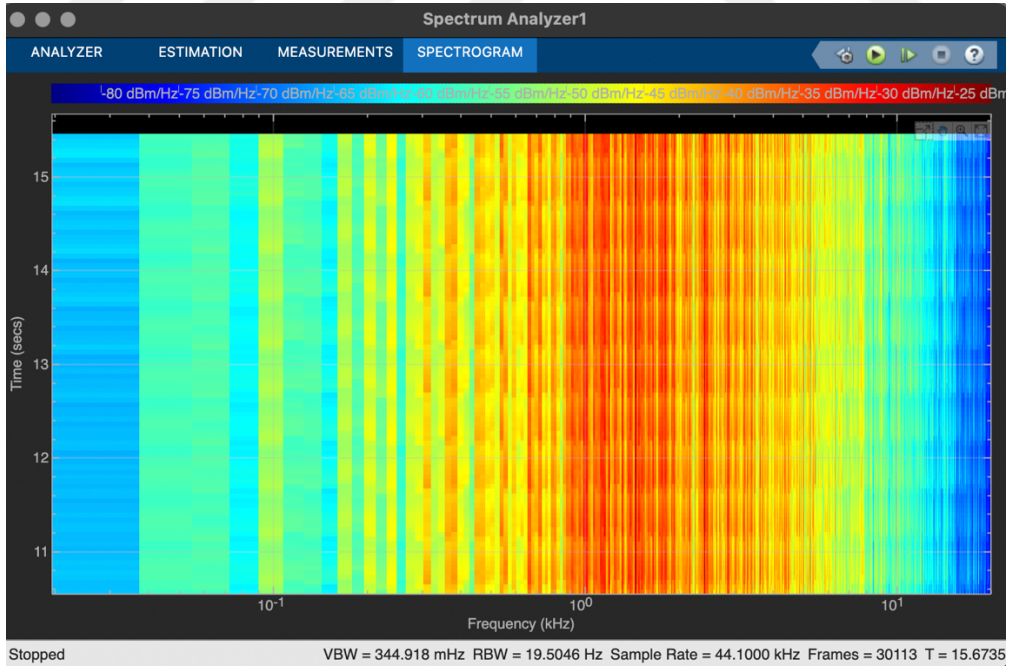


Şekil 33 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

3. Reverberator bloğunun parametre değerleri, Pre-delay:0.75, Highcut Freaquency:5.000, Diffusion:0.5, Decay Factor:0.5, High Frequency Damping:0.0005, Wet/Dry Mix:1, Input Sample Rate(Hz):44100 olacak şekilde ayarlanmıştır. Bu parametre değerleri ile Spectrum Analyzer üzerinden analiz yapılmış, bu parametre değerleri de baz aldığında sonuç olarak 100 Hz bölgesi ve 10.000 Hz bölgesi üzerinde 3-4 dB değerinde azalmalar olduğu gözlenmiştir. Bu parametre değerleri üzerinde Spectrum Analyzer ve Spectrogram grafikleri Şekil 34 ve Şekil 35 üzerinde gösterilmiştir.



Şekil 34 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

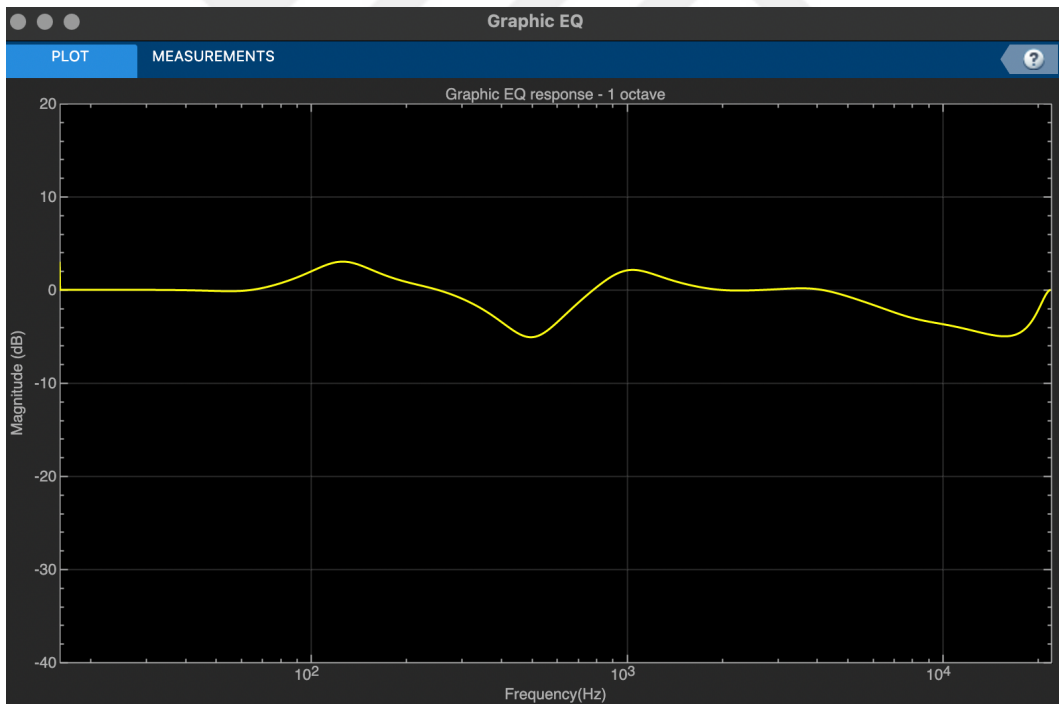


Şekil 35 İşlenmiş sinyal Spectrogram grafik çıktısı

4.2.1.2 Graphic EQ Bloğunun Ses Dosyası Üzerindeki Etkilerinin Grafikselsel Analizi

Oluşturulan modele Graphic EQ bloğu eklendikten sonra farklı parametre değerleri üzerindeki grafik analizi aşağıdaki gibidir;

1. Graphic EQ bloğunun parametre değerleri, 31.5 Hz: 0 dB, 64 Hz: 0 dB, 125 Hz: 3 dB, 250 Hz: 0 dB, 500 Hz: -5 dB, 1000 Hz: 2 dB, 2000 Hz: 0 dB, 4000 Hz: 0 dB, 8000 Hz: -3 dB, 16.000 Hz: -5 dB, Input Sample Rate(Hz): 44100 olarak ayarlanmıştır. Bu parametre değerleri üzerinden Graphic EQ grafik çıktısı Şekil 36 üzerinde gösterilmiştir.



Şekil 36 İşlenmiş sinyal Graphic EQ grafik çıktısı

4.2.2 Matlab Audio Test Bench ile Tını Efekt Pluginlerinin Grafiksel Analizleri

Matlab içerisinde bulunan “Command Window” bölümünden “audioTestBench” komutu kullanılarak Audio Test Bench arayüzü oluşturulmuştur. Bu arayüze “CLAP.wav” isimli ses dosyası yüklenmiş ve bu ses dosyası üzerinden çeşitli efekt Pluginlerinin testleri yapıp, grafiksel analiz sonuçları “Time Scope” ve “Spectrum Analyzer” üzerinden incelenmiştir.

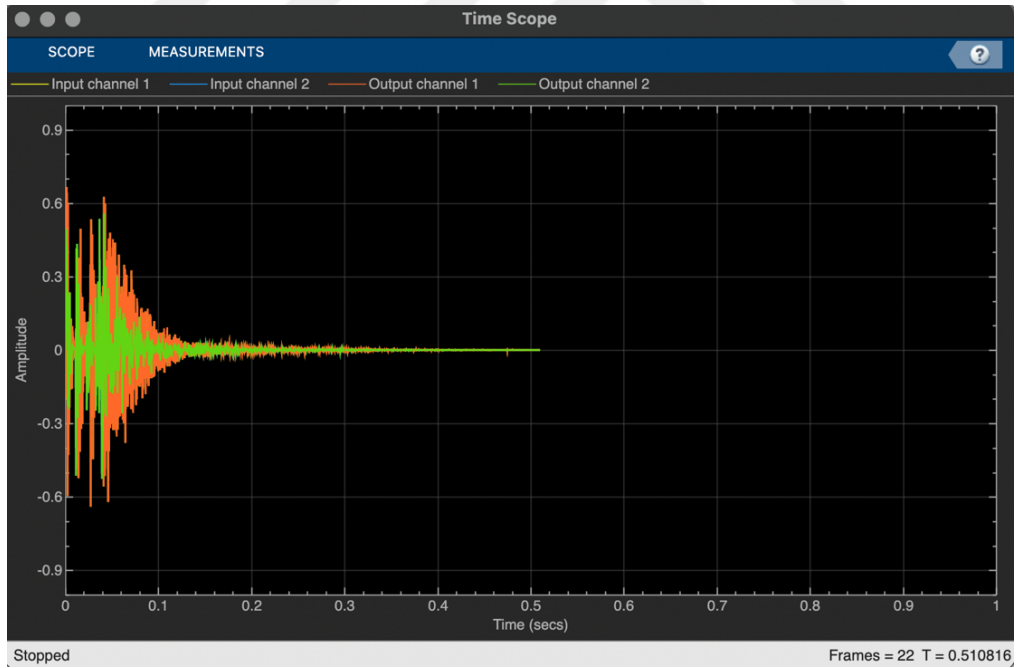
4.2.2.1 Chorus Efekt Plugininin Test Süreci ve Grafiksel Analizi

Audio Test Bench içerisindeki “Object Test” sekmesine örnek bir Chorus efekt plugini eklenmiş ve teste hazır hale getirilmiştir. Farklı parametre değerleri üzerinden Test Scope ve Spectrum Analyzer çıktıları analiz edilmiştir. Bu değerlerin analizleri ve karşılaştırmaları aşağıdaki gibidir;

1. Chorus efektinin “Wet/Dry Mix” parametresinin değeri 0 olarak ayarlanmış ardından ham sinyalin Time Scope ve Spectrum Analyzer grafikleri analiz edilmiştir. Ham sinyalin grafik çıktıları Şekil 37 ve Şekil 38 üzerinde gösterilmiştir.

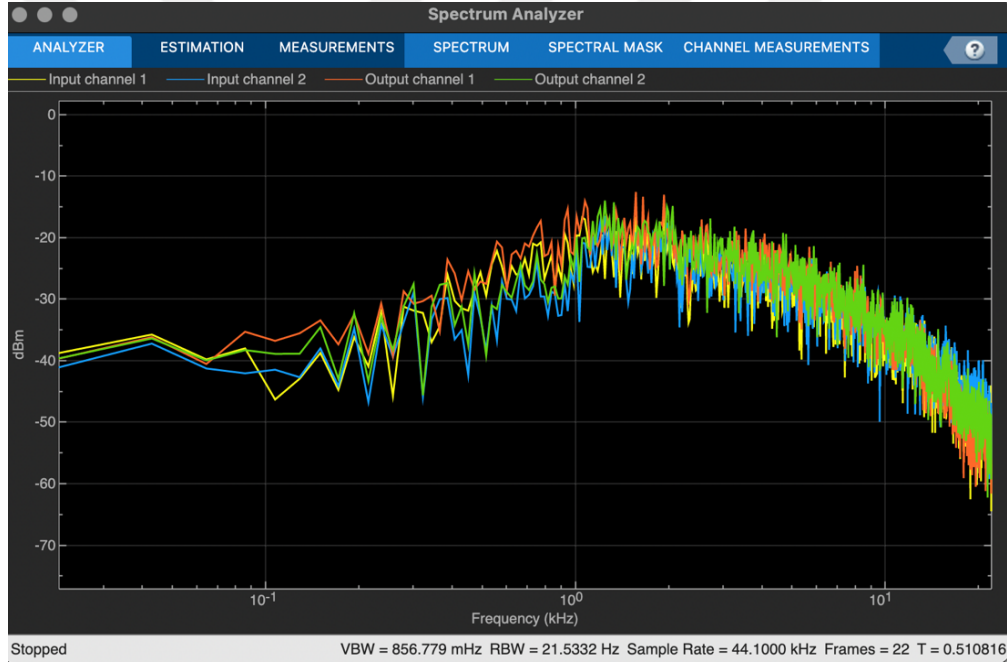


Şekil 37 Ham sinyal Spectrum Analyzer grafik çıktısı

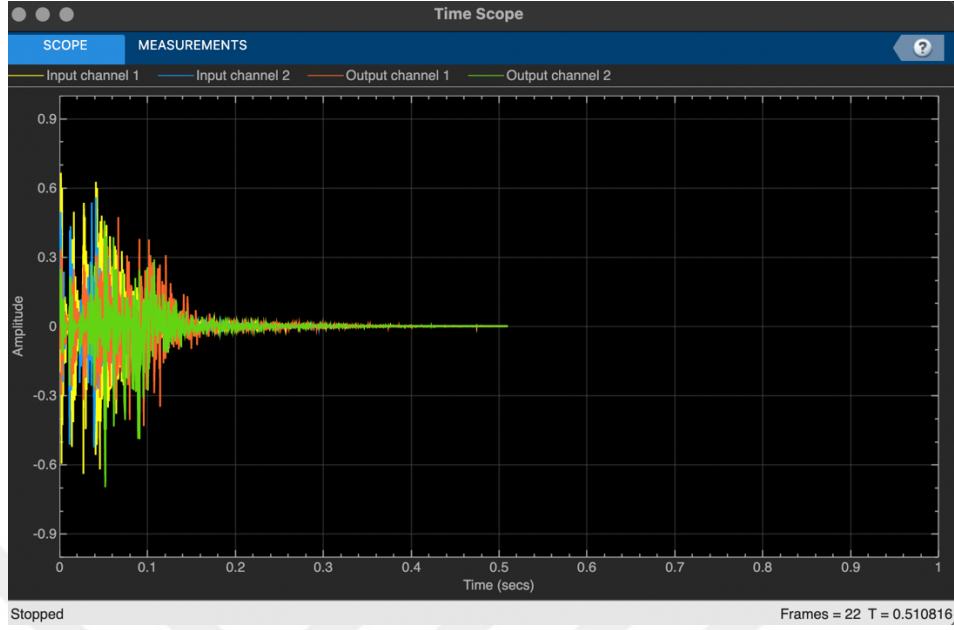


Şekil 38 Ham sinyal Time Scope çıktısı

2. Chorus efektinin parametre değerleri, Depth 1:2.5, Depth 2:2.5, Rate 1:2.5 Hz, Rate 2:2.5 Hz, Base Delay:0.05 s, Wet/Dry Mix:0.5 olarak ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer grafiği incelendiğinde, yeşil ve turuncu sinyallere ek olarak sarı ve mavi sinyallerin eklendiği ve ham sinyale benzer frekans aralıklarında iki farklı sinyalin daha grafiğe eklendiği gözlenmiştir. Time Scope grafiği incelendiğinde ise yine iki sinyalin daha grafiğe eklendiği ve 0 ila 0.1. saniyeler arasındaki Amplitude değerinde, yani ses dosyasının şiddetinde bir artış meydana geldiği gözlenmiştir. Bu grafik çıktıları Şekil 39 ve Şekil 40 üzerinde gösterilmiştir.

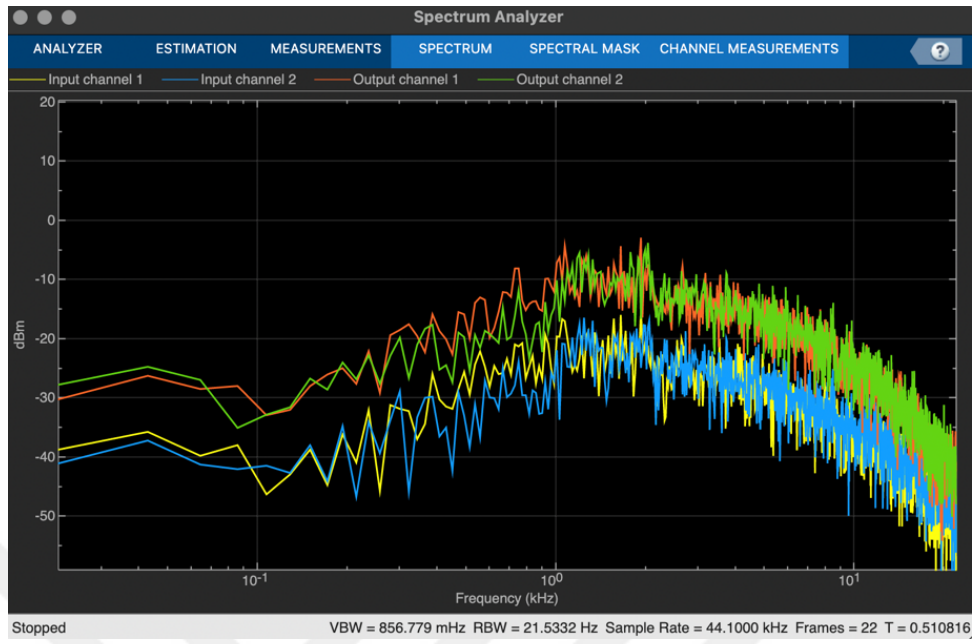


Şekil 39 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

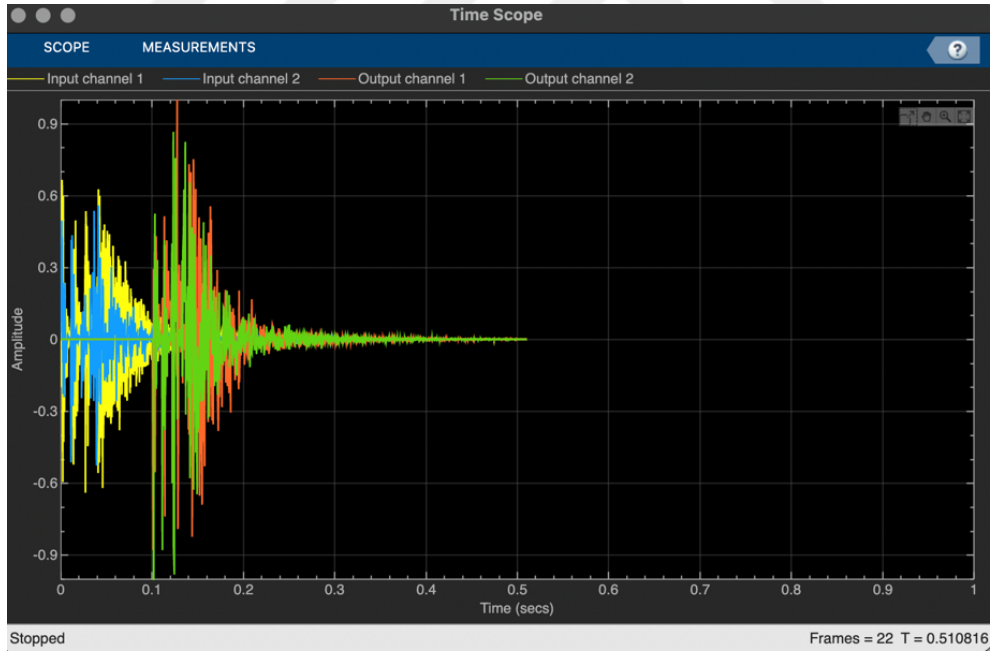


Şekil 40 İşlenmiş sinyal Time Scope grafik çıktısı

3. Chorus efektinin parametre değerleri, Depth 1:5, Depth 2:5, Rate 1:5 Hz, Rate 2:5 Hz, Base Delay:0.1 s, Wet/Dry Mix:1 olarak ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer grafiği incelendiğinde, “Base Delay” parametresinin arttırılması sonucunda oluşan daha düşük dB şiddetine sahip efekt sinyalleri gözlenmiştir. Time Scope grafiği incelendiğinde ise oluşan efekt sinyallerinin 0 ila 0.1 saniye aralıklarında ham sinyale benzer şiddette bir gecikme yaşattığı gözlenmiştir. Bu grafik çıktıları Şekil 41 ve Şekil 42 üzerinde gösterilmiştir.



Şekil 41 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

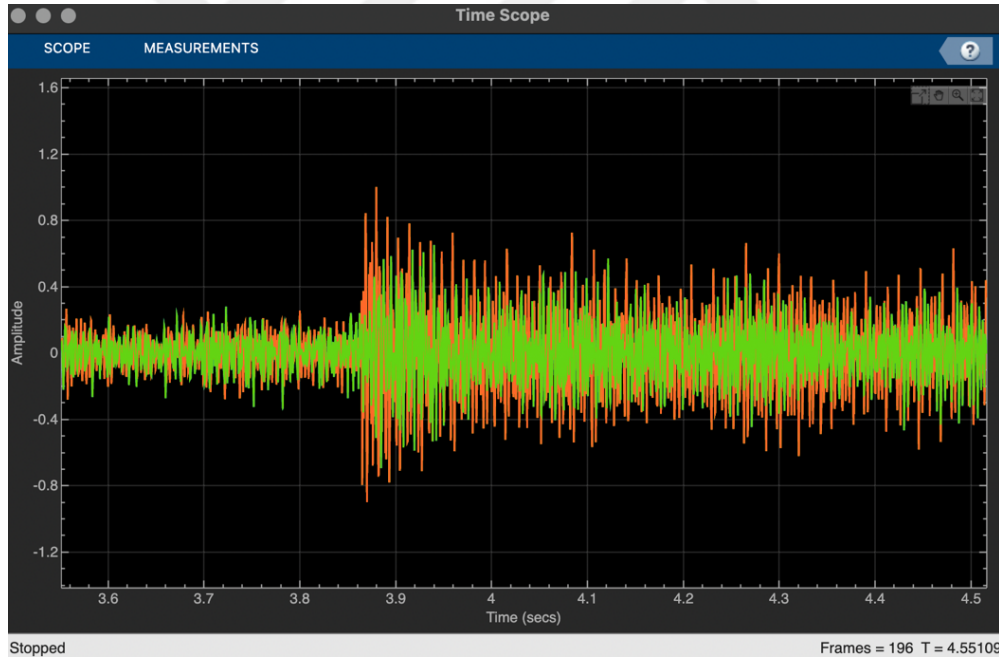


Şekil 42 İşlenmiş sinyal Time Scope grafik çıktısı

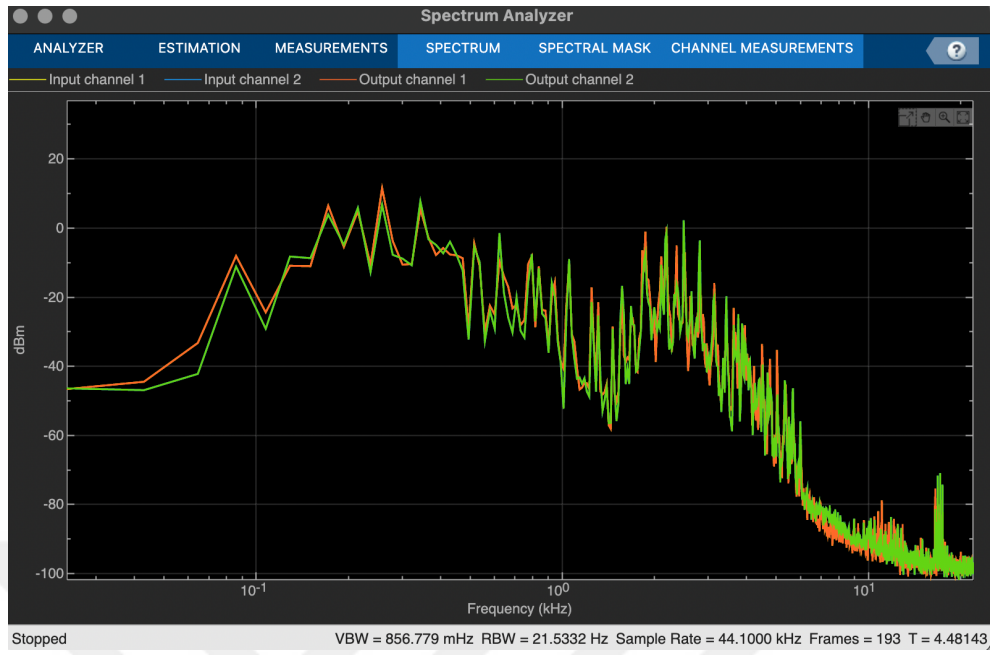
4.2.2.2 Echo Efekt Plugininin Test Süreci ve Grafiksel Analizi

Audio Test Bench içerisindeki “Audio File Reader” sekmesine “GUITAR.wav” isimli ses dosyası yüklenmiş ardından “Object Test” sekmesine örnek bir Echo efekt plugini eklenmiş ve teste hazır hale getirilmiştir. Farklı parametre değerleri üzerinden Test Scope ve Spectrum Analyzer çıktıları analiz edilmiştir. Bu değerlerin analizleri ve karşılaştırılmaları aşağıdaki gibidir;

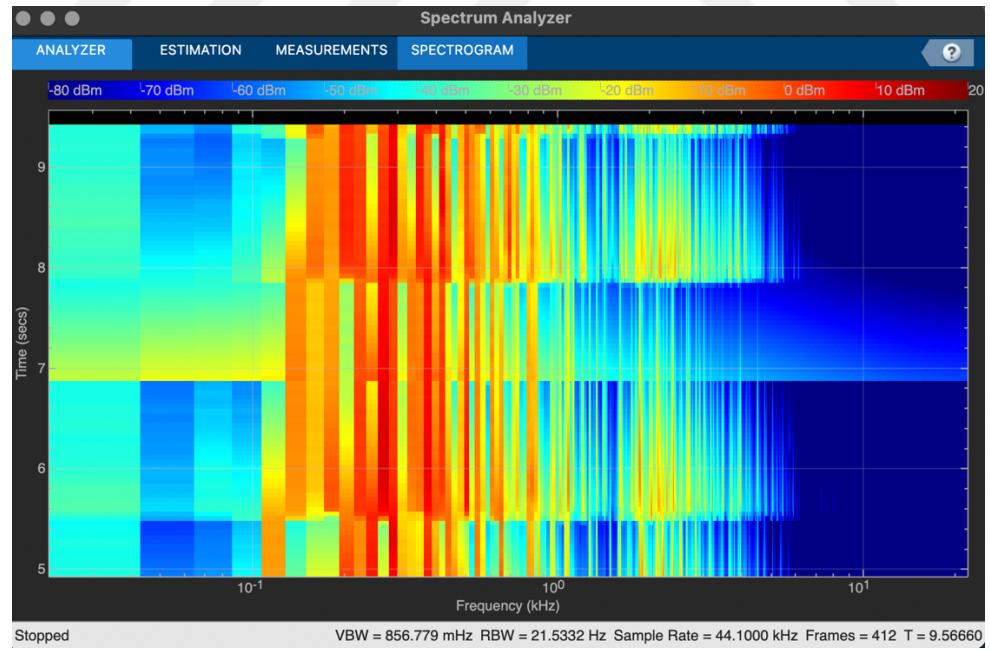
1. Echo efektinin “Wet/Dry Mix” parametresinin değeri 0 olarak ayarlanmış ardından ham sinyalin Time Scope, Spectrum Analyzer ve Spectrogram grafikleri analiz edilmiştir. Ham sinyalin grafik çıktıları Şekil 43, Şekil 44 ve Şekil 45 üzerinde gösterilmiştir.



Şekil 43 Ham sinyal Time Scope grafik çıktısı

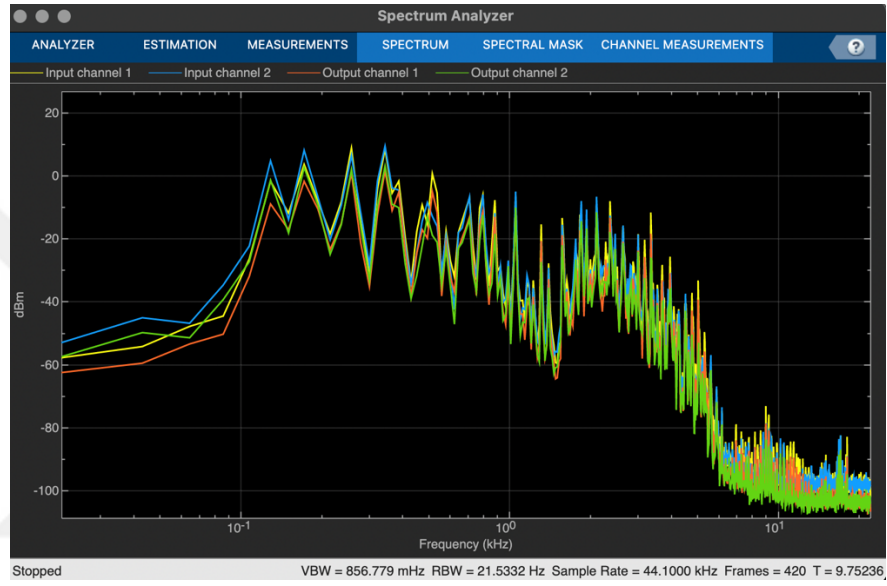


Şekil 44 Ham sinyal Spectrum Analyzer grafik çıktısı



Şekil 45 Ham sinyal Spectrogram grafik çıktısı

2. Echo efektinin parametre değerleri, Gain:0.3, Feedback:0.25, Base Delay:0.25 s, Wet/Dry Mix:0.5 olacak şekilde ayarlanmıştır ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde ses dosyasının dB olarak azaldığını ve sinyalin sıkıştırıldığını gözlenmiştir. Bu grafik çıktıları Şekil 46 ve Şekil 47 üzerinde gösterilmiştir.

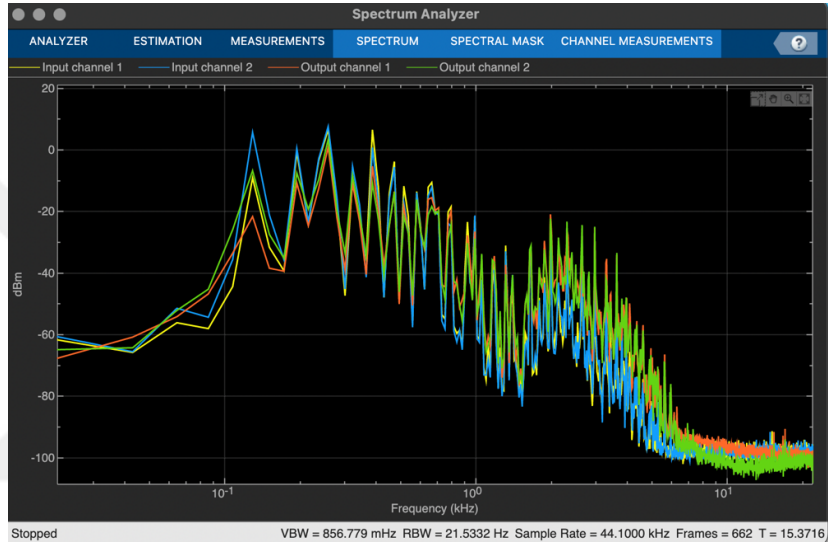


Şekil 46 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

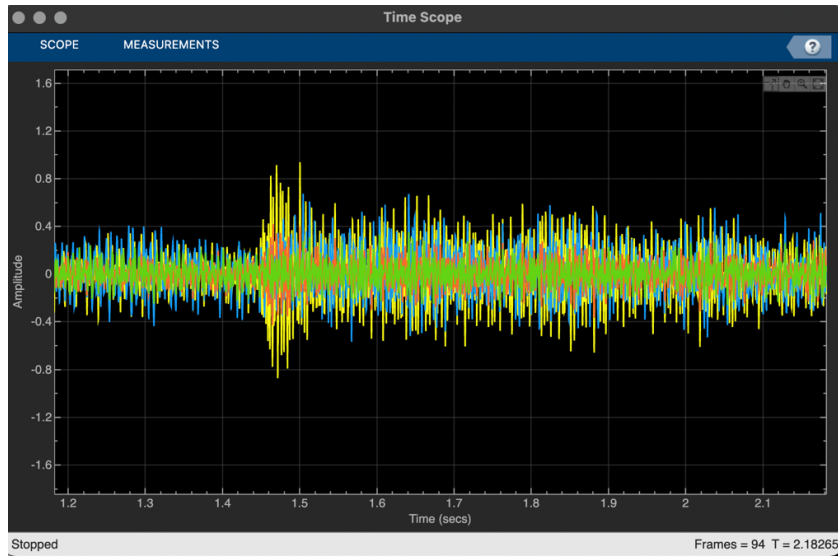


Şekil 47 İşlenmiş sinyal Time Scope grafik çıktısı

3. Echo efektinin parametre değerleri, Gain:0.5, Feedback:0.4, Base Delay:0.7 s, Wet/Dry Mix:0.6 olacak şekilde ayarlanmıştır ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde 1000 Hz bölgesinin dB olarak ciddi bir kayıp yaşadığı ve sinyalin iyice sıkıştığı gözlenmiştir. Bu grafik çıktıları Şekil 48 ve Şekil 49 üzerinde gösterilmiştir.



Şekil 48 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

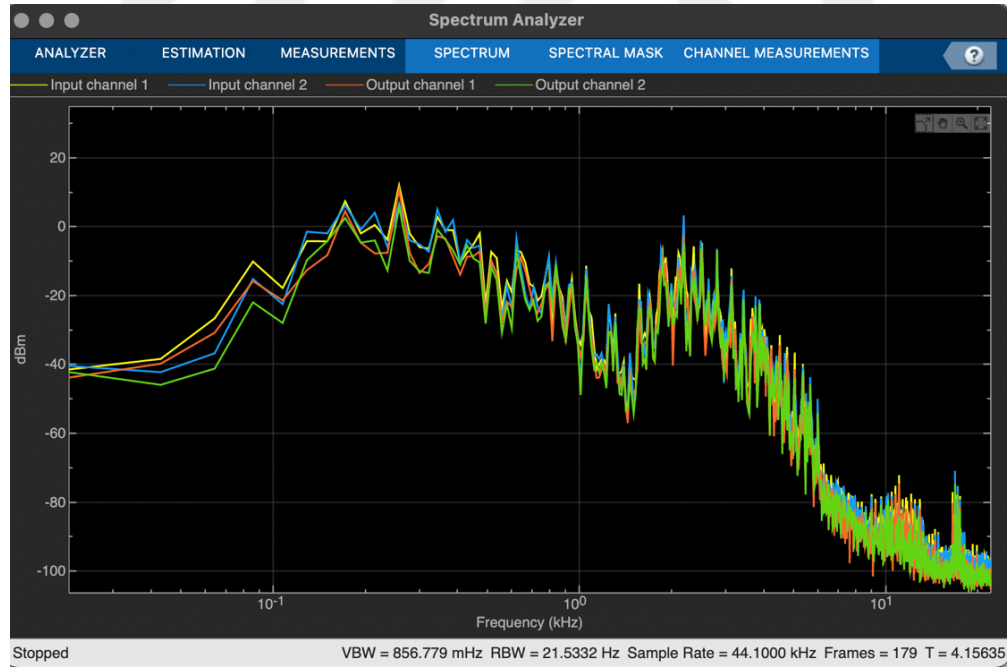


Şekil 49 İşlenmiş sinyal Time Scope grafik çıktısı

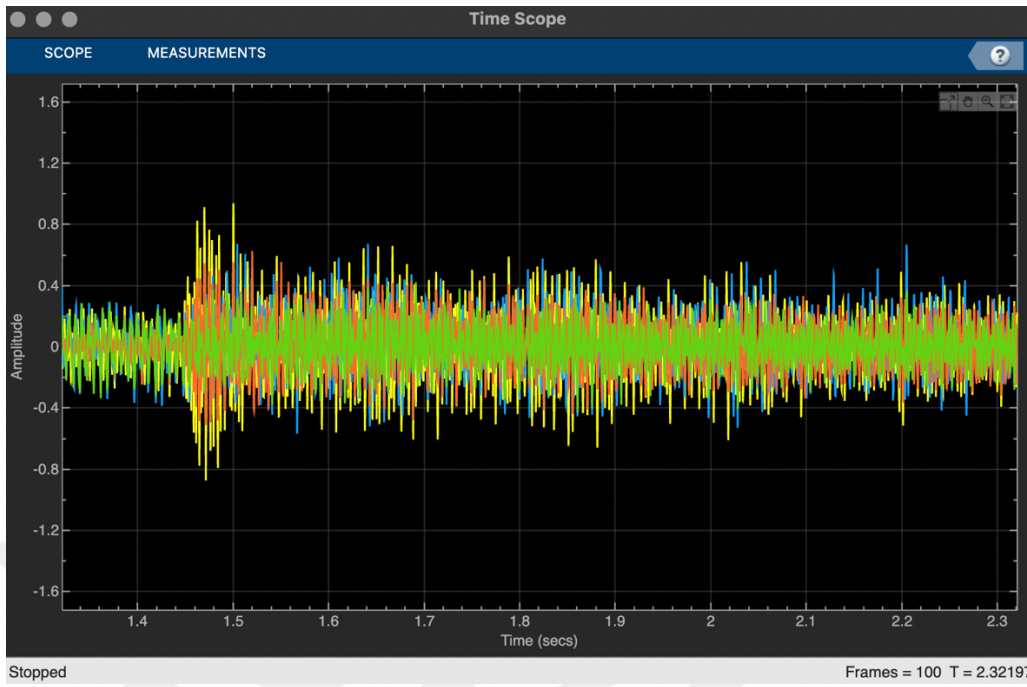
4.2.2.3 Flanger Efekt Plugininin Test Süreci ve Grafiksel Analizi

Audio Test Bench içerisindeki “Object Test” sekmesine örnek bir Flanger efekt plugini eklenmiş ve teste hazır hale getirilmiştir. Farklı parametre değerleri üzerinden Test Scope ve Spectrum Analyzer çıktıları analiz edilmiştir. Bu değerlerin analizleri ve karşılaştırmaları aşağıdaki gibidir;

1. Flanger efektinin parametre değerleri, Delay:0.05 s, Depth:10, Rate:0.25, Feedback:0.25, Wet/Dry Mix:0.4 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde Flanger efektinin frekans spektrumu üzerinde belirli bir düşüş sağlandığı gözlenmiştir. Time Scope grafiğinin analizinde Flanger efektinin ses dosyası üzerindeki atak süresini ve darbe süresini azalttığı görülmüştür. Bu grafik çıktıları Şekil 50 ve Şekil 51 üzerinde gösterilmiştir.

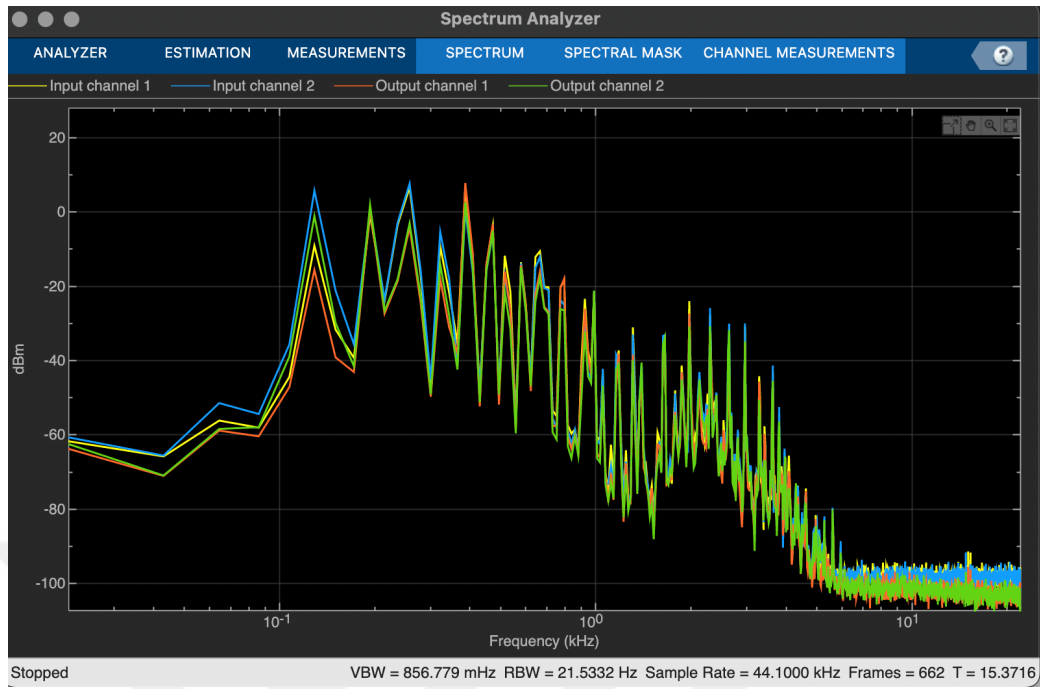


Şekil 50 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

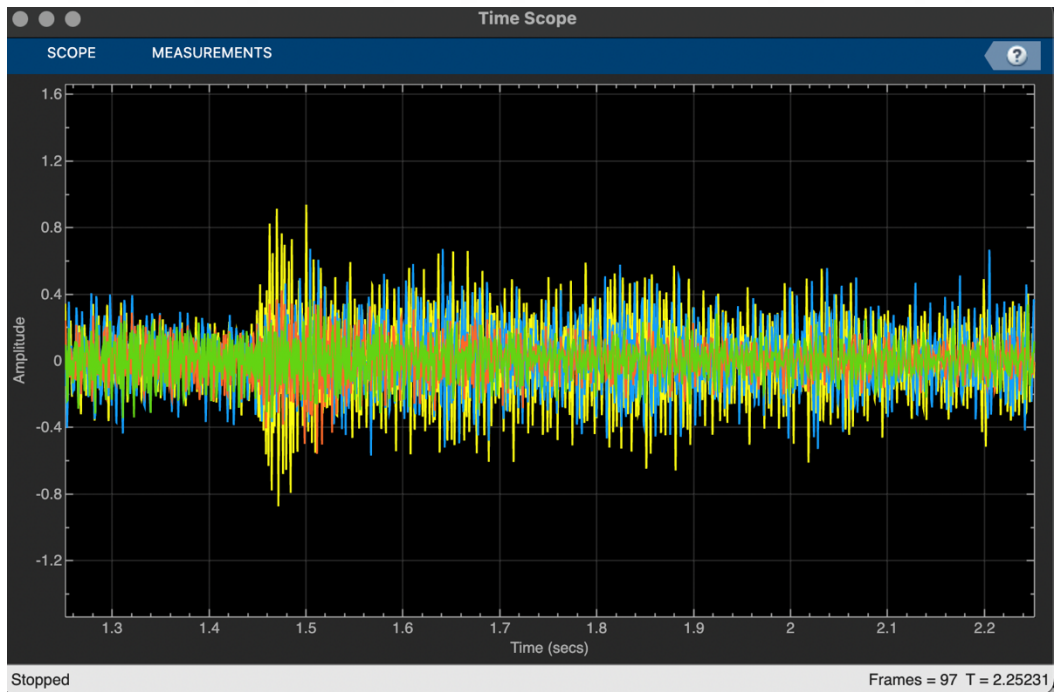


Şekil 51 İşlenmiş sinyal Time Scope grafik çıktısı

2. Flanger efektinin parametre değerleri, Delay:0.04 s, Depth:25, Rate:0.35, Feedback:0.3, Wet/Dry Mix:0.6 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde ham sinyal ile işlenmiş sinyal arasındaki desibel farklarının daha fazla açıldığı gözlenmiştir. İki grafik de incelendiğinde, oluşan efekt sinyallerinin belirli modülasyonlar ve genlik farklılıkları oluşturduğu gözlenmiştir. Bu grafik çıktıları Şekil 52 ve Şekil 53 üzerinde gösterilmiştir.

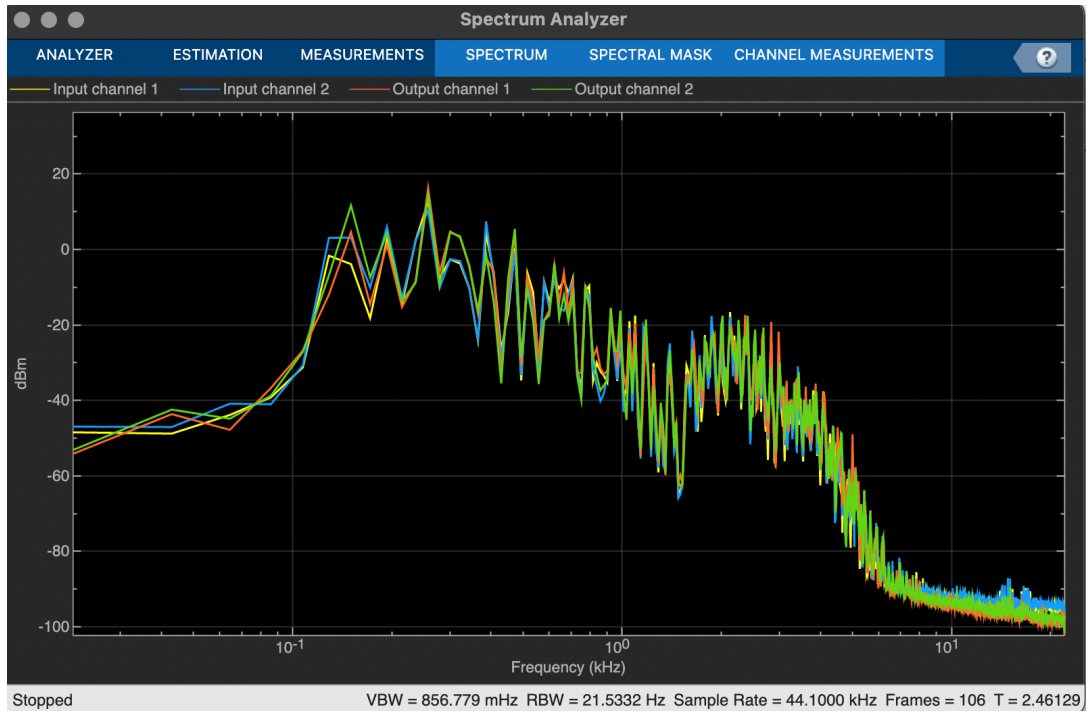


Şekil 52 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

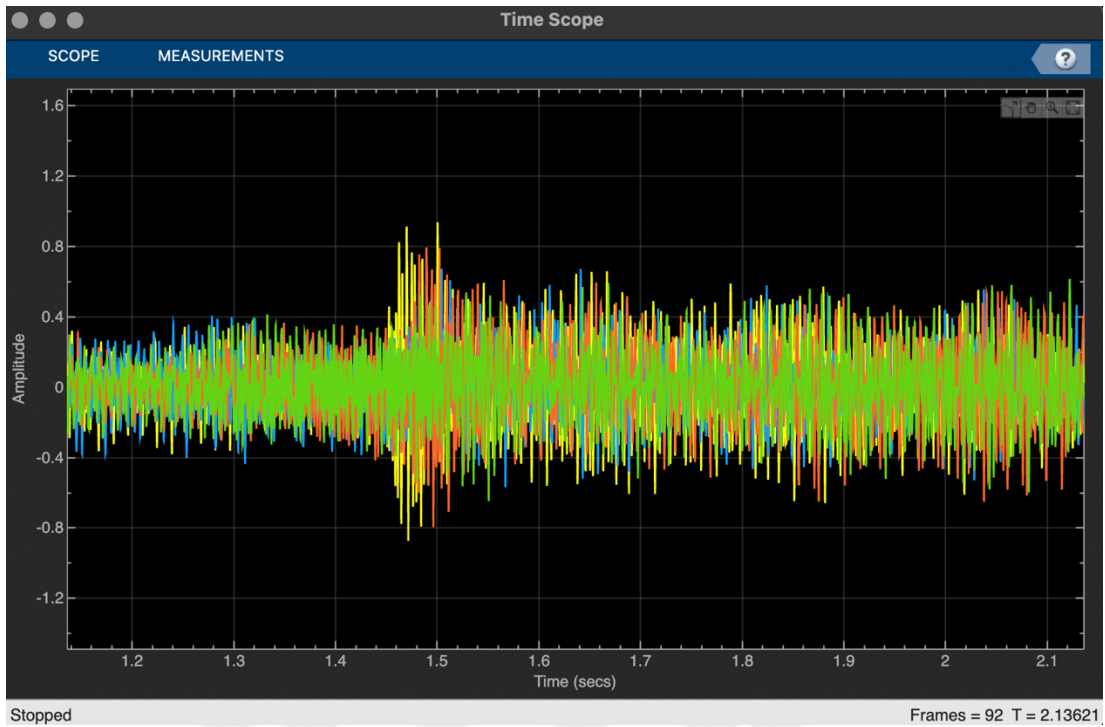


Şekil 53 İşlenmiş sinyal Time Scope grafik çıktısı

3. Flanger efektinin parametre deęerleri, Delay:0.02 s, Depth:30, Rate:0.35, Feedback:0.7, Wet/Dry Mix:0.8 olacak řekilde ayarlanmıř ve grafiksel sonuları analiz edilmiřtir. Spectrum Analyzer ve Time Scope zerindeki grafik sonuları incelendięinde belli frekans blgelerinde geniřlemeler ve baskın karakterli harmoniklerin oluřtuęu gzlenmiřtir. Bu grafik ıktıları řekil 54 ve řekil 55 zerinde gsterilmiřtir.



řekil 54 İřlenmiř sinyal Spectrum Analyzer grafik ıktısı

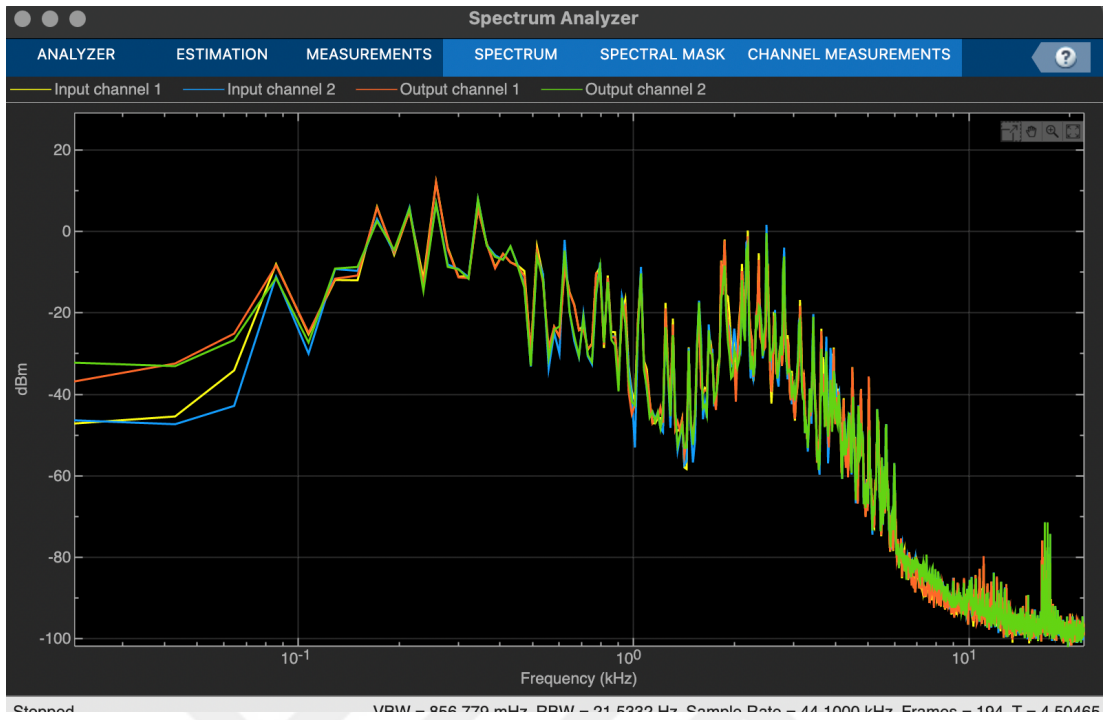


Şekil 55 İşlenmiş sinyal Time Scope grafik çıktısı

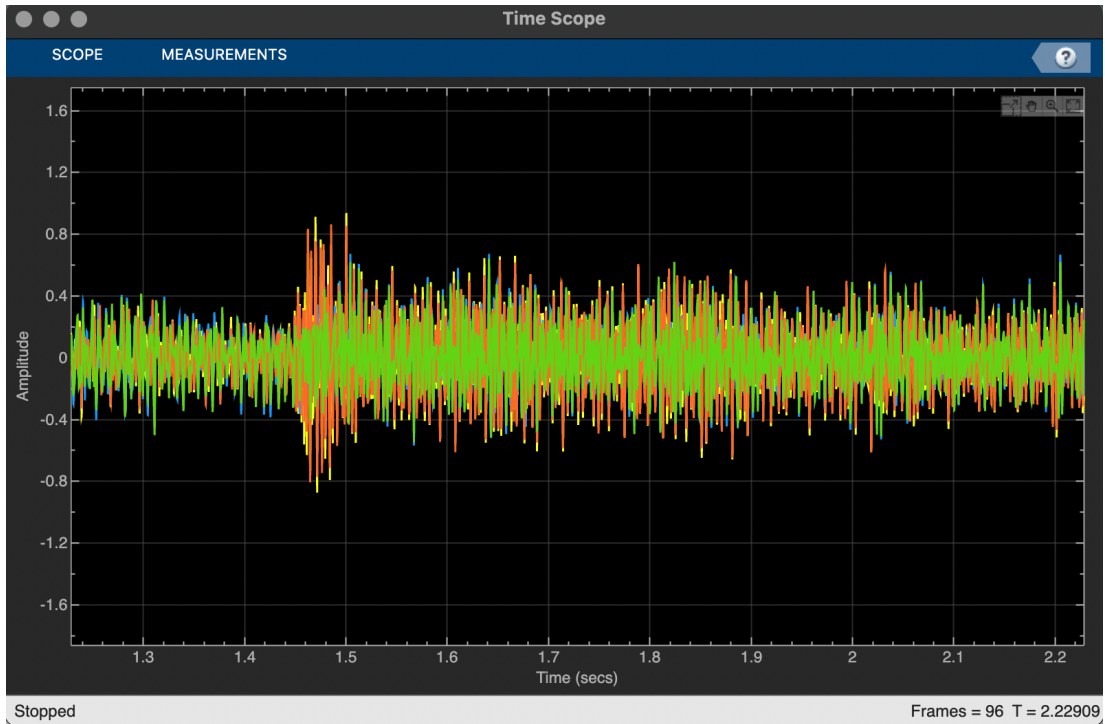
4.2.2.4 Phaser Efekt Plugininin Test Süreci ve Grafiksel Analizi

Audio Test Bench içerisindeki “Object Test” sekmesine örnek bir Flanger efekt plugini eklenmiş ve teste hazır hale getirilmiştir. Farklı parametre değerleri üzerinden Test Scope ve Spectrum Analyzer çıktıları analiz edilmiştir. Bu değerlerin analizleri ve karşılaştırmaları aşağıdaki gibidir;

1. Phaser efektinin parametre değerleri, Rate:400 Hz, Center Frequency:250 Hz, Depth:1, Quality Factor:10, Modulation Source:Sin, Notches:4 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde Phaser efektinin Flanger efektine benzer etkiler yarattığı gözlenmiştir. Spectrum Analyzer grafiği üzerinde belirli frekans bölgelerinde genlik düşmesi ve faz kayması olduğu analiz edilmiştir. Bu grafik çıktıları Şekil 56 ve Şekil 57 üzerinde gösterilmiştir.

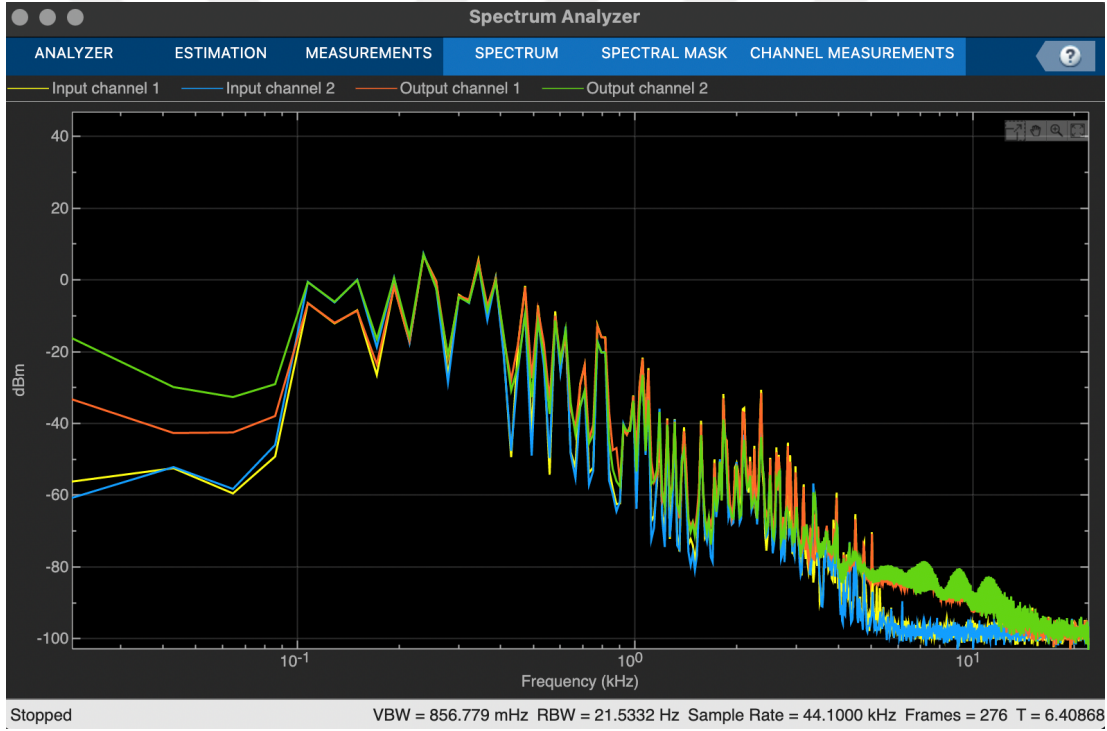


Şekil 56 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

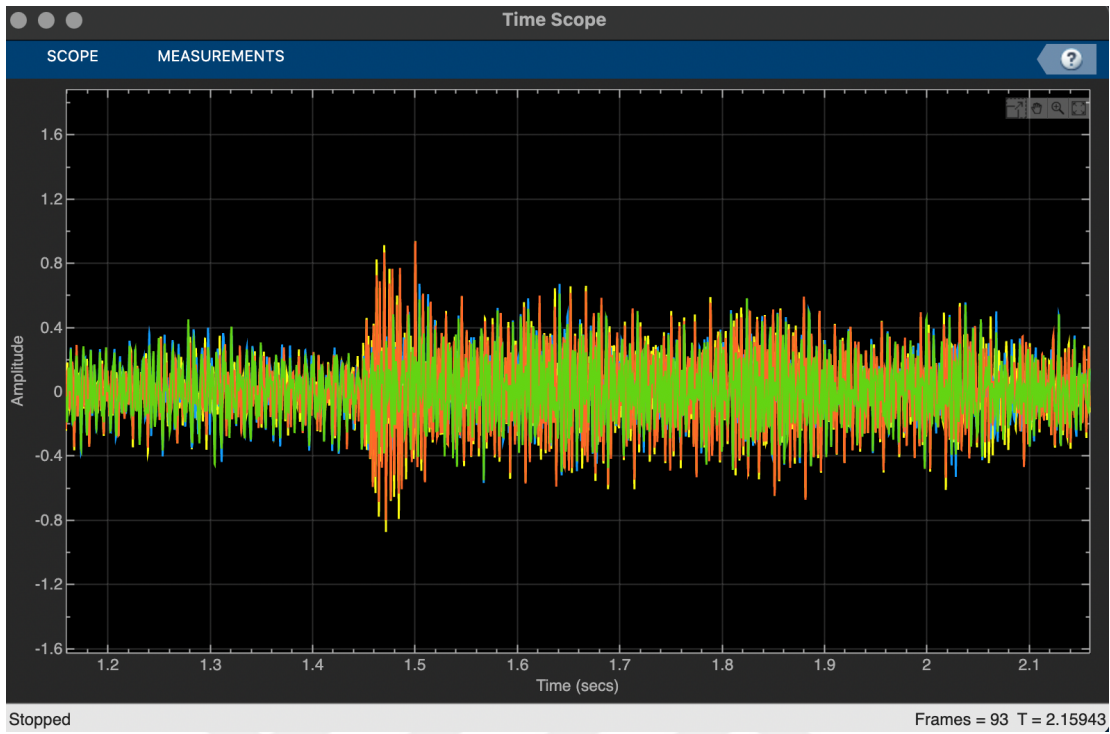


Şekil 57 İşlenmiş sinyal Time Scope grafik çıktısı

2. Phaser efektinin parametre değerleri, Rate:600 Hz, Center Frequency:350 Hz, Depth:1.3, Quality Factor:12, Modulation Source:Triangle, Notches:6 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde Rate parametresinin artırılmasıyla birlikte frekans bölgelerinde daha hızlı ve sık değişen modülasyon dalgaları gözlenmiştir. “Modulation Source” parametresinin Triangle olarak ayarlanması belirli frekans bölgelerinde üçgen bir dalga formu oluşturarak faz etkisi yaratmaktadır. Bu grafik çıktıları Şekil 58 ve Şekil 59 üzerinde gösterilmiştir.

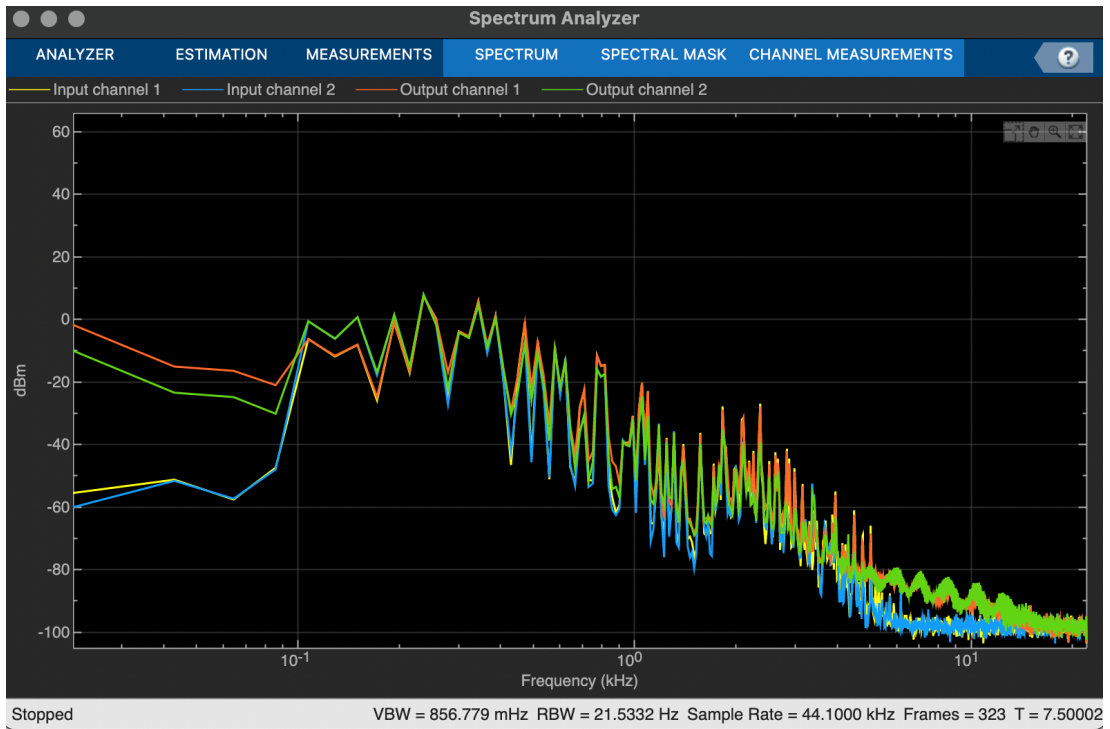


Şekil 58 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

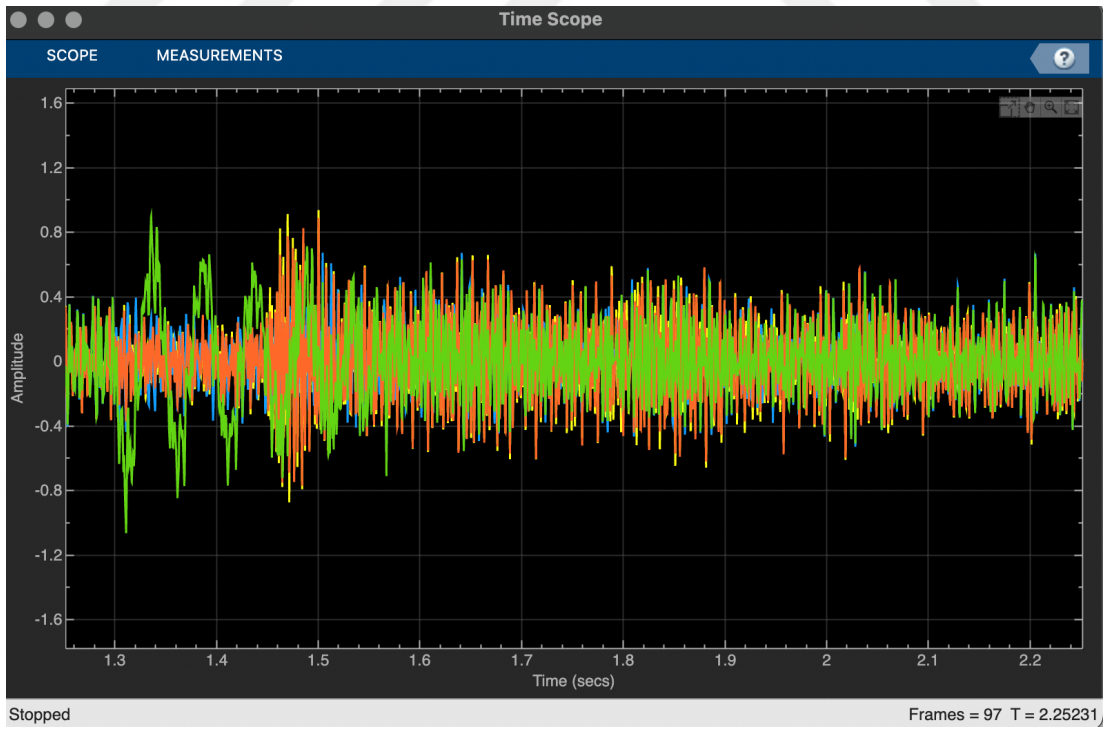


Şekil 59 İşlenmiş sinyal Time Scope grafik çıktısı

3. Phaser efektinin parametre değerleri, Rate:800 Hz, Center Frequency:400 Hz, Depth:1.5, Quality Factor:20, Modulation Source:Ramp, Notches:8 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Time Scope üzerindeki grafik sonuçları incelendiğinde 'Modulation Source' parametresinin Ramp olarak ayarlanması, lineer bir rampa dalga formunun oluştuğu ve Spectrum Analyzer üzerinde keskin geçişler yaşandığı gözlenmiştir. Bu grafik çıktıları Şekil 60 ve Şekil 61 üzerinde gösterilmiştir.



Şekil 60 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

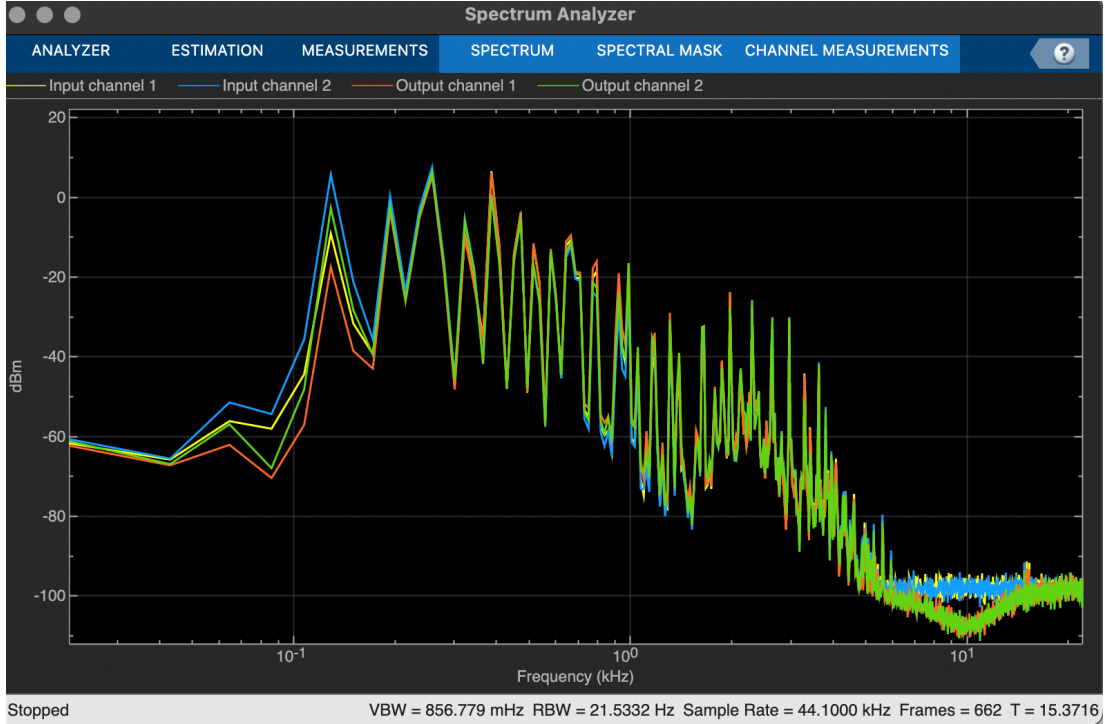


Şekil 61 İşlenmiş sinyal Time Scope grafik çıktısı

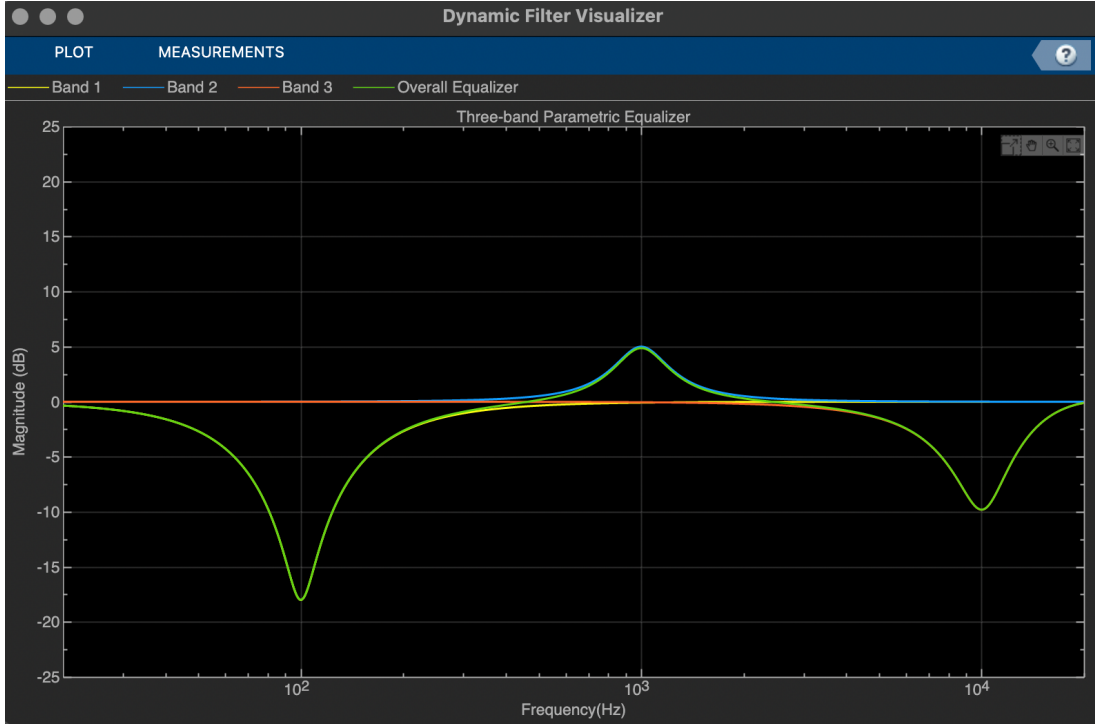
4.2.2.5 Parametric Equalizer Plugininin Test Süreci ve Grafiksel Analizi

Audio Test Bench içerisindeki “Object Test” sekmesine örnek bir Flanger efekt plugini eklenmiş ve teste hazır hale getirilmiştir. Farklı parametre değerleri üzerinden Test Scope ve Spectrum Analyzer çıktıları analiz edilmiştir. Bu değerlerin analizleri ve karşılaştırmaları aşağıdaki gibidir;

1. Parametric Equalizer plugininin parametre değerleri, Low Gain:-18 dB, Low Frequency:100 Hz, Low Q:2, Mid Gain: 5 dB, Mid Frequency:1000 Hz, Mid Q:2, High Gain:-10, High Frequency:10.000 Hz, High Q:2 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Dynamic Filter Visualizer üzerindeki grafik sonuçları incelendiğinde dinamik bir filtreleme etkileri gözlenmiştir. Bu ses dosyasının belirli frekans aralıklarında daha net ve hassas bir kontrol sağlar. Bu parametre değerleri üzerinden gitar sesine daha karakteristik bir ton kazandırılmaya çalışılmıştır. Bu grafik çıktıları Şekil 62 ve Şekil 63 üzerinde gösterilmiştir.

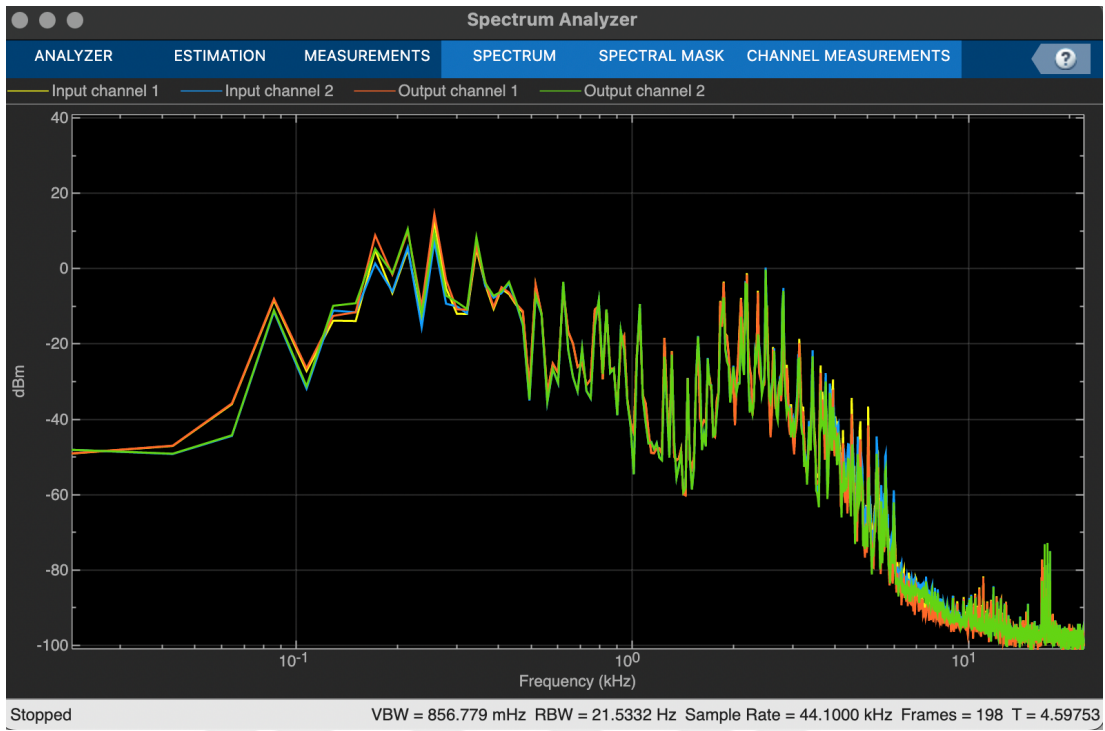


Şekil 62 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

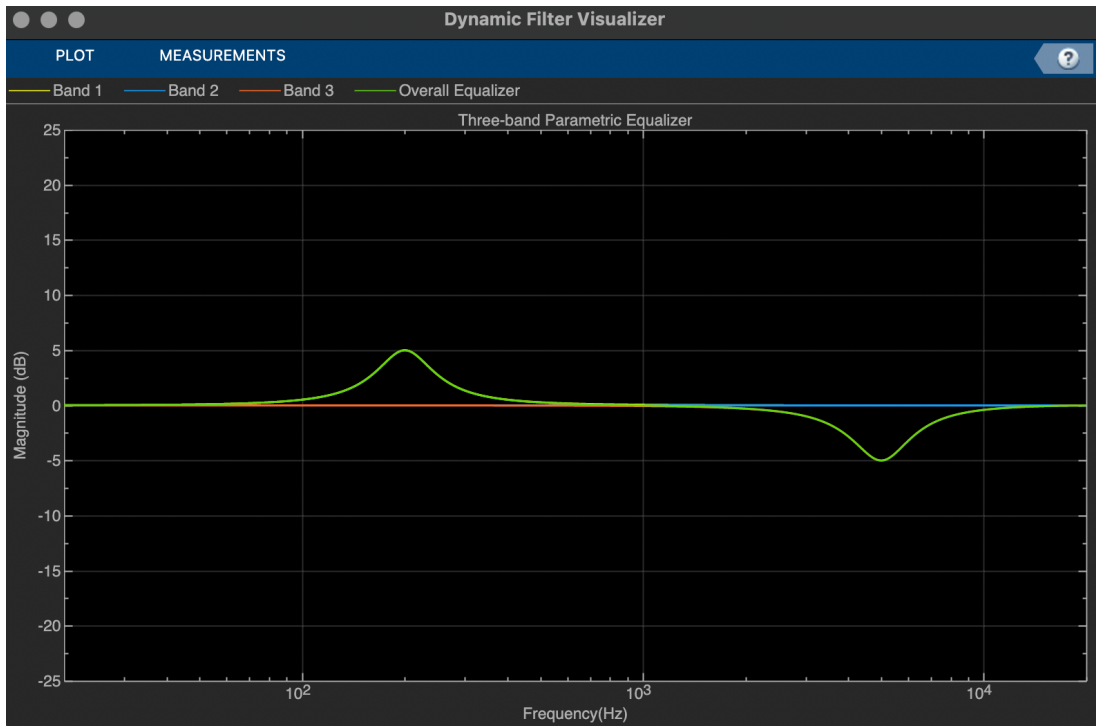


Şekil 63 İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı

2. Parametric Equalizer plugininin parametre değerleri, Low Gain:5 dB, Low Frequency:200 Hz, Low Q:2, Mid Gain: 0 dB, Mid Frequency:1000 Hz, Mid Q:2, High Gain:-5, High Frequency:5000 Hz, High Q:2 olacak şekilde ayarlanmış ve grafiksel sonuçları analiz edilmiştir. Spectrum Analyzer ve Dynamic Filter Visualizer üzerindeki grafik sonuçları incelendiğinde “Low Gain” parametresinin artırılmasıyla birlikte, frekans spektrumundaki düşük frekans bölgelerinde belirli bir artış olduğu gözlenmiştir. Bu artış sesin alt frekanslarında daha vurgulu ve güçlü bir etki yaratmasını sağlayabilir. Bu grafik çıktıları Şekil 64 ve Şekil 65 üzerinde gösterilmiştir.



Şekil 64 İşlenmiş sinyal Spectrum Analyzer grafik çıktısı

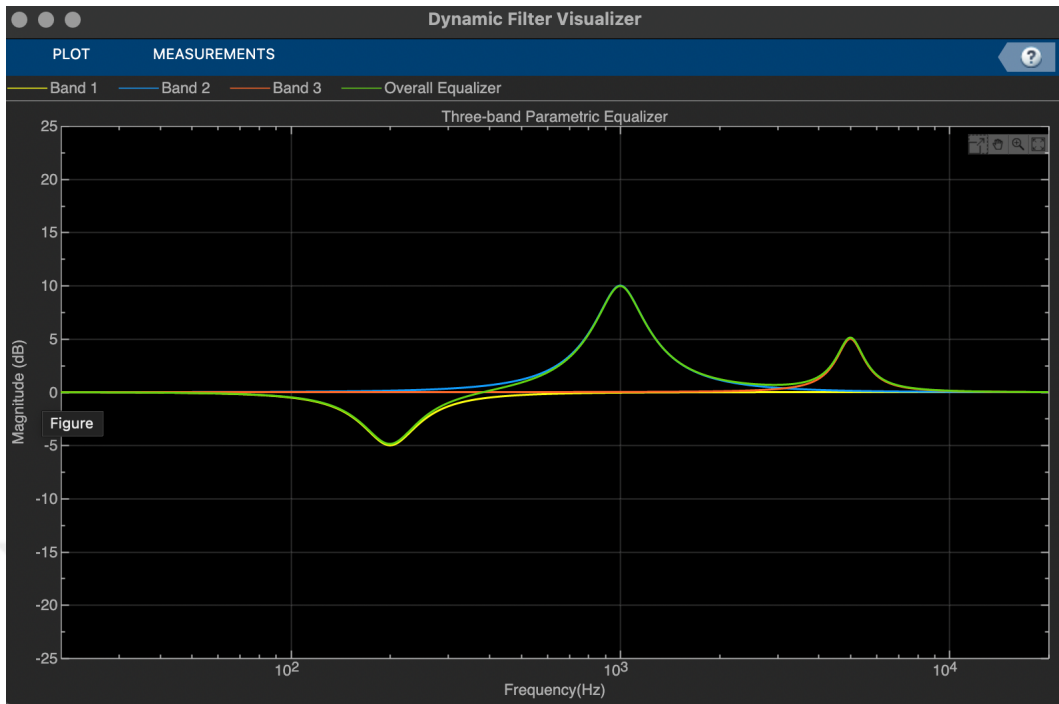


Şekil 65 İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı

3. Parametric Equalizer plugininin parametre deęerleri, Low Gain:-5 dB, Low Frequency:200 Hz, Low Q:2, Mid Gain:10 dB, Mid Frequency:1000 Hz, Mid Q:2, High Gain:5, High Frequency:5000 Hz, High Q:4 olacak řekilde ayarlanmıř ve grafiksel sonuları analiz edilmiřtir. Spectrum Analyzer ve Dynamic Filter Visualizer zerindeki grafik sonuları incelendięinde “Mid Gain” parametresinin 10 dB arttırılması sonucunda, orta frekans blgelerinin vurgulandıęı ve sesin karakteristik zelliklerinin arttıęı gzlenmiřtir. “High Gain” parametresinin 5 dB arttırılması ve “Q” parametresinin arttırılması sonucunda, st frekans blgelerinde daha gl ve aık bir etki yarattıęı analiz edilmiřtir. Bu grafik ıktıları řekil 66 ve řekil 67 zerinde gsterilmiřtir.



řekil 66 İřlenmiř sinyal Spectrum Analyzer grafik ıktısı



Şekil 67 İşlenmiş sinyal Dynamic Filter Visualizer grafik çıktısı

4.3 Üçüncü Alt Probleme İlişkin Bulgu ve Yorumlar

“Üretilen efektler müzik prodüksiyonu içerisinde nasıl kullanılabilir?”

Bu problemin amacı, üretilen efektlerin müzik prodüksiyonu içerisindeki rolünü ve kullanımı araştırmaktır. Efektler, ses kaynaklarının karakterini, genişliğini, derinliğini ve duygusal etkilerini arttırmak veya azaltmak için kullanılabilir. Hem dinamik hem de harmonik etkileri olan farklı efekt çeşitleri, üretilen müziğin ihtiyaçlarına göre birçok farklı yerde kullanılabilirler.

Bu bölümde, üretilen efekt pluginlerinin Matlab yazılımı içerisinde hangi formatlarla ve nasıl dışarı çıkartılabileceği konusu araştırılmıştır. Araştırma, PreSonus firmasının geliştirmiş olduğu Studio One programı kullanılarak, bu efektlerin DAW içerisine nasıl entegre edilebileceği ve nasıl kullanılabileceği üzerine odaklanmıştır. Bu doğrultuda, Studio One programı kullanılarak efektlerin nasıl uygulandığı ve

müzikal anlamda nasıl etkiler yarattığı örneklerle incelenmiş ve bu örneklerin grafiksel sonuçları analiz edilmiştir.

4.3.1 Matlab Audio Test Bench ile Tını Efekt Pluginlerinin Harici Çıktılara Aktarılması

Matlab Audio Test Bench, oluşturulan pluginlerin harici çıktılara aktarılabilmesi için gerekli işlevler sunan bir araçtır. Audio Test Bench, Matlab yazılım ortamında ses işleme uygulamalarının geliştirilebilmesi ve test edilebilmesi için oluşturulmuştur. “Generate Audio Plugin” sekmesi, oluşturulan pluginin hangi formatta dışarı aktarmak istediğimizi belirlediğimiz bölümdür. Pluginlerin harici çıktılara aktarılma süreci, örnek bir Chorus efekt plugini üzerinden anlatılmıştır. Audio Test Bench aracına aktarılan Chorus plugini, “Audio Plugin Generator” üzerinden parametreleri ayarlanarak istenilen formatta dışarı aktarılmıştır. “Audio Plugin Generator” parametreleri ve dosya formatları Şekil 68 ve Şekil 69 üzerinde gösterilmiştir.

Audio Plugin Generator

Audio plugin class name:

▼ **Validation options**

☒ Run a MEX version of the test bench

☐ Save test benches to output folder

▼ **Generation options**

Output folder:

Output file name:

Format: ☐ Generate a macOS universal plugin

▼ **Coder configuration**

Deep learning library:

Code replacement library:

Şekil 68 Audio Plugin Generator parametreleri

Format:

▼ **Coder configurat**

Deep learning libr

Code replacemen

- VST
- VST
- VST3
- AU
- AUv3
- Standalone executable
- JUCE project

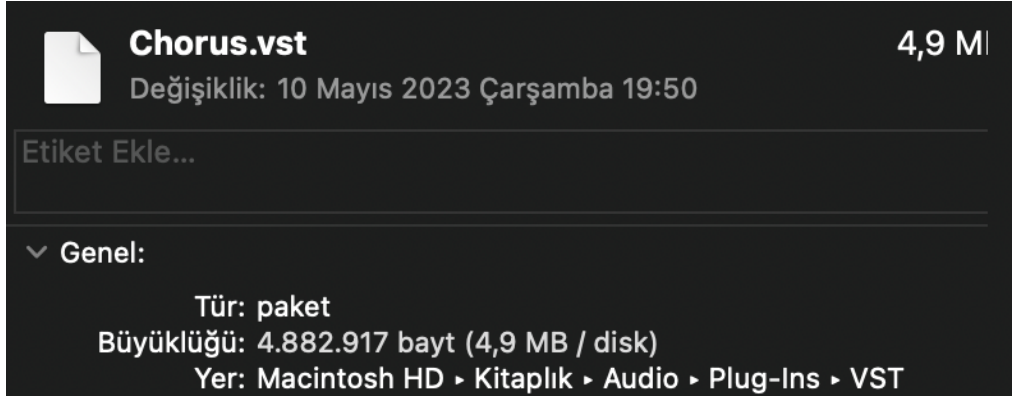
Şekil 69 Audio Plugin Generator dosya formatları

Farklı DAW programları ve ses işleme platformları için geliştirilen pluginler, VST, VST3, AU, Standalone gibi farklı dosya formatları olarak işlenebilir. VST,

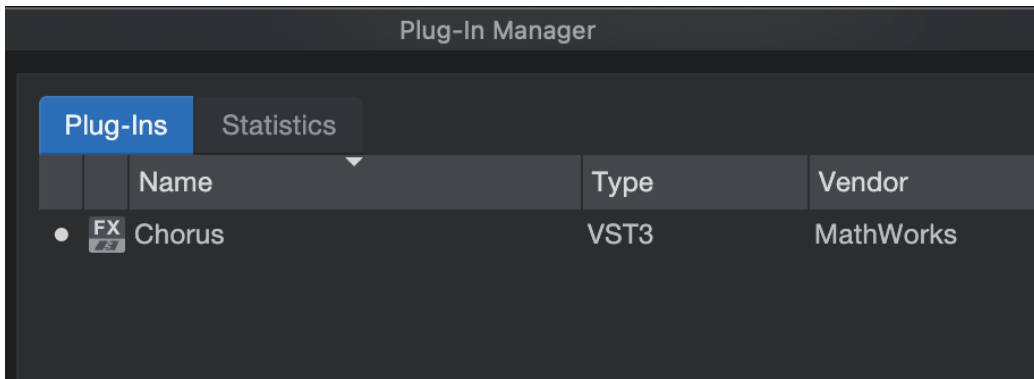
Steinberg isimli firmanın geliřtirdiđi bir formattır ve birok ses iřleme platformu ile entegre alıřabilir. VST formatı, kullanıcı arayüzü ve parametre kontrolleri ile birlikte gelir. VST3 formatı ise daha geliřtirilmiř bir versiyon sunar. AU, Apple firmasının geliřtirdiđi bir formattır ve yalnızca macOS iřletim sistemleriyle uyumludur. Standalone formatı ise bađımsız olarak alıřabilen pluginleri ifade eder. Chorus plugininin rneklendirilmesi ve harici bir ıktıya aktarılması, yaygın olarak kullanılan VST formatı ile gerekleřtirilmiřtir.

4.3.2 Oluřturulan Pluginin DAW ierisine Entegre Edilerek Kullanılması

Matlab üzerinden oluřturulan ve VST formatında iřlenen Chorus efekt plugini, Studio One yazılımında kullanılabilmesi iin bilgisayarın VST klasörüne tařınmıř ve Studio One ierisinden bulunabilir bir hale getirilmiřtir. Kullanılan Chorus efekt plugininin dosya yolu ve Studio One ierisindeki grüntüsü řekil 70 ve řekil 71 üzerinde gsterilmiřtir.



řekil 70 Chorus.vst dosya yolu



Şekil 71 Chorus pluginin Studio One içerisindeki gösterimi

Studio One içerisine entegre edilen Chorus, Echo, Flanger, Phaser ve Parametric Equalizer pluginleri, davul, bas, elektro gitar ve piyano sesleri kullanılarak oluşturulan kısa bir aranjenin miks sürecinde kullanılmıştır ve bu aranjenin master kanalı üzerine Spectrum Analyzer eklenerek ham sinyal ve işlenmiş sinyal grafikleri analiz edilmiştir. Studio One içerisinde kullanılan pluginlerin “Console” sekmesindeki görüntüsü Şekil 72 üzerinde gösterilmiştir.



Şekil 72 Studio One “Console” sekmesi

Oluşturulan aranjenin miks sürecindeki plugin parametre değerleri aşağıdaki gibidir;

1. “Bass” kanalı üzerine Parametric Equalizer ve Phaser pluginleri eklenmiştir. Parametric Equalizer pluginin parametre değerleri, Low Gain:4.800 dB, Low Frequency:50.500 Hz, Low Q:2, Mid Gain:0 dB, High Gain:-20 dB, High Frequency:5000 Hz, High Q:1 olacak şekilde ayarlanmıştır. “Bass” kanalı üzerindeki Phaser pluginin parametre değerleri ise, Rate:170 Hz, Center Frequency:201 Hz, Depth:0.880, Quality Factor:11.012, Modulation Source:Triangle, Notches:4 olacak şekilde ayarlanmıştır.
2. “Guitar Chords” kanalı üzerine Chorus plugini eklenmiştir. Chorus pluginin parametre değerleri, Depth 1:3.264, Rate 1:3.470 Hz, Depth 2:7.793, Rate 2:1.480 Hz, Base Delay:0.021 s, Wet/Dry Mix:0.587 olacak şekilde ayarlanmıştır.
3. “Guitar Muted” kanalı üzerine Parametric Equalizer ve Echo pluginleri eklenmiştir. Parametric Equalizer pluginin parametre değerleri, Low Gain:0 dB, Mid Gain:-5 dB, Mid Frequency:1000 Hz, Mid Q:2, High Gain:-3 dB, High Frequency:5000 Hz, High Q:2 olacak şekilde ayarlanmıştır. “Guitar Muted” kanalı üzerine eklenen Echo pluginin parametre değerleri ise, Gain:0.215, Feedback:0.174, Base Delay:0.200 s, Wet/Dry Mix:0.310 olacak şekilde ayarlanmıştır.
4. “Piano” kanalı üzerine Parametric Equalizer ve Flanger pluginleri eklenmiştir. Parametric Equalizer pluginin parametre değerleri, Low Gain:2.566 dB, Low Frequency:201Hz, Low Q:2, Mid Gain:0 dB, High Gain:-4 dB, High Frequency:10.000 Hz, High Q:2 olacak şekilde ayarlanmıştır. “Piano” kanalı üzerindeki Flanger pluginin parametre değerleri ise, Delay:0.001 s, Depth:25.623, Rate:0.135, Feedback:0.115, Wet/Dry Mix:0.651 olacak şekilde ayarlanmıştır.
5. “Drum” kanalı üzerine Parametric Equalizer plugini eklenmiştir. Parametric Equalizer pluginin parametre değerleri, Low Gain:3 dB, Low Frequency:100 Hz, Low Q:2, Mid Gain:-15 dB, Mid Frequency:1000 Hz, Mid Q:2, High

Gain:-5 dB, High Frequency:10.000 Hz, High Q:2 olacak şekilde ayarlanmıştır.

Bu parametre değerleri baz alınarak, misk sürecinin etkili ve etkisiz grafik görselleri Span aracılığı ile Şekil 73 ve Şekil 74 üzerinde gösterilmiştir.



Şekil 73 Ham sinyallerin master kanalı üzerinden Span grafik çıktısı



Şekil 74 İşlenmiş sinyallerin master kanalı üzerinden Span grafik çıktısı

Span grafik çıktıları incelendiğinde, pluginlerin devreye girdikten sonraki işlenmiş sinyal grafiğinde, 100 Hz ve 1000 Hz bölgesinde belirgin dB düşüşlerinin meydana geldiği gözlenmiştir. Ayrıca, 20.000 Hz bölgesindeki frekansların kesilmesi, grafik üzerinde de görsel olarak analiz edilmiştir. Bu gözlemler, pluginlerin sinyal işleme ve miks sürecinde spektrum grafikleri üzerinde belirli bölgelerde değişikliklere neden olduğunu göstermektedir.

BÖLÜM V

aSONUÇ, TARTIŞMA VE ÖNERİLER

5.1 Sonuç

Bu çalışma, yazılım programları kullanarak dijital tını efektlerinin nasıl üretilebileceğini incelemekte ve Matlab yazılımının bu süreçteki etkisini göstermektedir. Matlab, işlevsel ses işleme kütüphanesi ve kolay kullanım arayüzü sayesinde yazılımcılar ve müzik yapımcıları için dijital tını efektleri oluşturmaya ve özelleştirmeye imkan sağlamıştır. Matlab'ın ses işleme fonksiyonları, ses sinyallerinin analizi, işlenmesi ve filtrelenmesi için önemli araçlar sunar. Ses işleme araçları, Chorus, Echo, Flanger, Phaser gibi popüler tını efektlerinin oluşturulmasında önemli bir rol üstlenmektedir. Kullanıcılar, Matlab'ın geniş ses işleme araçları sayesinde istedikleri efektleri özelleştirebilir ve bu deneyimlerini kendi istedikleri şekilde şekillendirebilirler.

Özetle, bu araştırma Matlab yazılımının dijital tını efektleri oluşturma ve kişiselleştirme süreci üzerindeki teknik etkisini göstermiştir. Matlab'ın işlevsel ses işleme kütüphanesi ve kolay arayüzü, ses sinyallerini analiz etmek, işlemek ve filtrelemek için güçlü araçlar sağlar. Matlab, kullanıcıların istedikleri efektleri yaratmalarına ve özelleştirmelerine ve kendi yaratıcı vizyonlarını hayata geçirmelerine olanak tanır. Araştırmanın bulguları, dijital tını efektlerinin yazılım programları kullanarak nasıl oluşturulabileceği konusunda bir bakış açısı sunmaktadır ve bu alanda çalışan kişilerin çalışmalarını daha çok geliştirmelerine yardımcı olabilir.

Gelecekteki araştırmalar, bu sektördeki teknolojik gelişmeleri takip ederek ve Matlab'ın ses işleme yeteneklerini daha da geliştirerek, dijital tını efektlerinin ileri düzeyde ve karmaşık biçimlerde üretilmesine katkı sağlayabilir.

5.2 Tartışma

Bu araştırma, dijital tını efektlerinin üretilmesi ve özelleştirilebilmesi için Matlab yazılımının bu sürece olan teknik etkisini incelemektedir. Matlab'ın ses işleme araçları ve kolay arayüzü bu süreci kolaylaştırmaktadır. Ancak bu araştırmanın sonuçları baz alınarak bazı tartışma noktaları ortaya çıkmaktadır.

Matlab'ın gelişmiş ses işleme fonksiyonları ve kolay arayüzü, kullanıcıların oluşturduğu efektlerin parametrelerini ayarlayabilmelerini ve istedikleri efektleri oluşturabilmelerini sağlamaktadır. Bazı kullanıcılar için bu araçların yetersiz kalabileceği düşünülebilir. Belirli bir efektin istenilen sonucu verebilmesi için daha teknik ve spesifik araçlar kullanılması gerekebilir. Bu nedenle, Matlab'ın ses işleme fonksiyonlarının sınırları ortaya çıkabilir ve kullanıcıların farklı yazılım ve farklı kütüphanelere yönelmeleri gerekebilir.

Bu araştırma, Matlab yazılımının dijital tını efekti üretme konusunda etkili bir yazılım olduğunu gösterse de diğer yazılımların da tını efekti üretme konusunda kullanılabileceği unutulmamalıdır. Farklı ses işleme kütüphaneleri ve farklı programlama dilleri, özgün dijital tını efektleri oluşturabilmek için daha işlevsel araçlar sunabilir. Bu sebeple kullanıcıların oluşturmak istediği efekt türü ve ihtiyaçları doğrultusunda kullanabilecekleri yazılımlar farklılık gösterebilir.

Matlab'ın kolay arayüzü ve hazır ses işleme kütüphanesi hızlı sonuçlar elde etme imkanı sunsa da bu durum bazı kullanıcılar için daha az teknik bir deneyim olarak görülebilir. Daha ileri seviye ses işleme tekniklerinin kullanılabilmesi için daha geniş bir bilgi ve beceri gerekebilir. Bu nedenle kullanıcıların gelişmiş ses işleme tekniklerini öğrenmeleri ve uygulamaları için farklı kaynaklara ve eğitimlere ihtiyaç duyabilecekleri unutulmamalıdır.

5.3 Öneriler

Bu araştırmanın sonuçları göz önüne alındığında, dijital tını efektlerinin üretimi sürecinde Matlab yazılımının teknik etkisi ile ilgili bazı öneriler sunulabilir. Bu öneriler, kullanıcıların daha etkili ve verimli bir şekilde tını efekti üretme sürecini desteklemek amacıyla aşağıda verilmiştir;

1. Farklı ses işleme kütüphanelerini araştırın: Birçok farklı ses işleme kütüphanesi bulunmaktadır. Bu kütüphaneler ile ilgili araştırma yapıp bilgi edinmek ve farklı yazılımların sunduğu özellikleri keşfetmek, kullanıcıların dijital tını efekti üretme sürecinde alternatifleri değerlendirmelerine katkı sağlayacağı düşünülmektedir.
2. İleri düzey teknik ses işleme tekniklerini araştırın: Dijital tını efekti üretim sürecinde ileri düzey ses işleme tekniklerini kullanmak, daha karmaşık ve özgün efektlerin oluşturulabilmesine yardımcı olur. Filtreleme teknikleri, modülasyon, gecikme gibi konuları araştırmak, kullanıcıların daha özgün efektler oluşturabilmesine katkı sağlayacağı düşünülmektedir.
3. Forumlar ve kaynaklardan yararlanın: Matlab kullanıcıların oluşturduğu forumlar ve topluluklar, dijital tını efekti üretimi sürecinde farklı bakış açılarını görebilmek ve deneyim edebilmek için kullanışlı platformlar sağlar. Bu platformlara katılmak ve diğer kullanıcılar ile iletişim kurmak, bilgi ve birikim kazanma açısından katkı sağlayacağı düşünülmektedir. Bu forumlar ve topluluklar üzerinden farklı kullanıcıların ve MathWorks firmasının paylaştığı olduğu açık kaynak kodlarına ve örnek çalışmalara ulaşılabilir.

Bu öneriler, kullanıcıların dijital tını efekti üretme ve özelleştirme sürecinde daha fazla teknik bilgi ve kaynaklara erişmelerini sağlayarak, daha özgün çalışmalar yapmalarına fayda sağlayabilir.

KAYNAKÇA

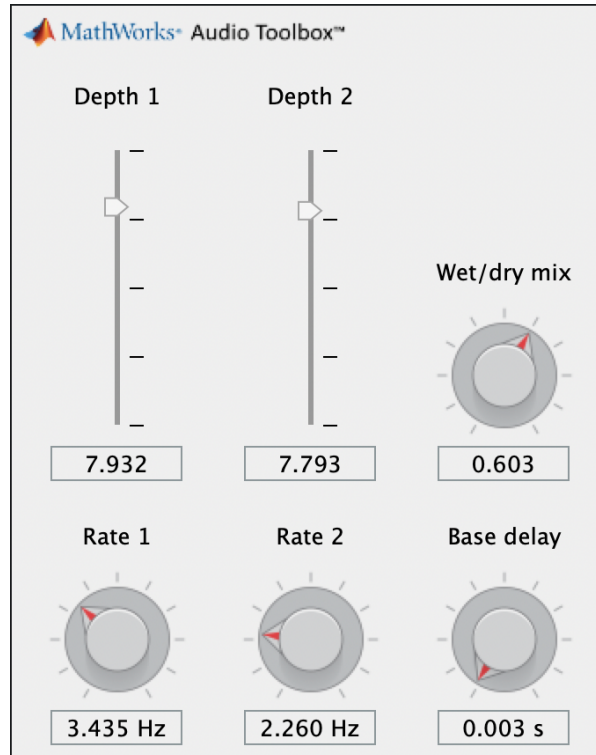
- Önen, U. (2007). *Ses Kayıt ve Müzik Teknolojileri*. Çitlembik Yayınları.
- Önen, U. (2016). *Miks Üzerine*. Çitlembik Yayınları.
- Altın, M. (2013, Mart 28). Sayısal Sinyal İşleme. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, s. 581-590.
- Buza, O., Todorean, G., & Nica, A. (2006). Voice Signal Processing for Speech Synthesis.
- Earl, D. (2006, January). The Ultimate Guide to Audio Effects: Delay. *Sound on Sound Magazine*.
- Karthick, R., & Kumar, P. G. (2019). Real-time Audio Signal Processing: Fundamentals, Algorithms and applications. *Journal of Ambient Intelligence and Humanized Computing*, s. 1805-1819.
- Little, J., & Moler, C. (1984). *MathWorks*. MathWorks: www.mathworks.com adresinden alındı
- Lyons, R. G. (2010). Understanding Digital Signal Processing.
- Ozerov, A., Vincent, E., & Bimbot, F. (2012). A General Flexible Framework for the Handling of Prior Information in Audio Source Separation.
- Price, A. (2019, July). An Introduction to Delay Effects. *Audio Technology Magazine*.
- Proakis, J. G. (2006). Principles, Algorithms and Applications.
- Smith, J. R., & Rodet, X. (2014, September). Python and C in Musical Signal Processing: A Comparative Study. *Proceedings of the International Computer Music Conference*, s. 423-428.
- van der Hulst, A. J. (2001). Audio Effects: Theory, Implementation and Application. *Journal of the Audio Engineering Society*, pp. 247-264.
- Varghese, M., Yadav, Y., & Bhatia, G. N. (2021). Comparative study and detailed Graphical Analysis of Equalizer, Chorus, Panning and Reverb Digital Audio Effects using MATLAB. *International Conference on Circuits, Controls and Communications*.
- Yılmaz, E. (2017). Ses İşleme Yöntemleri ve Uygulamaları. *Ege Üniversitesi Basımevi*.

EKLER

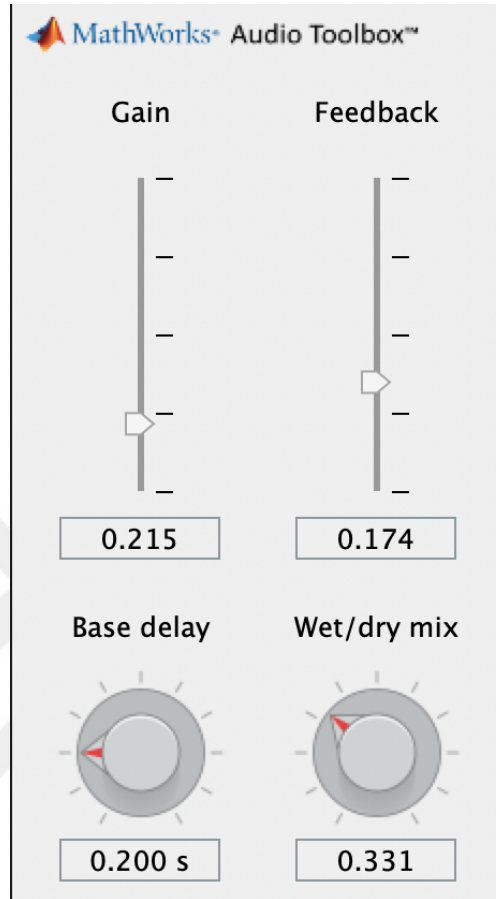
EK-1 Tını Efektleri Kullanılarak Oluşturulan ve Miksaj Yapılan Ses Dosyalarının QR Kodu



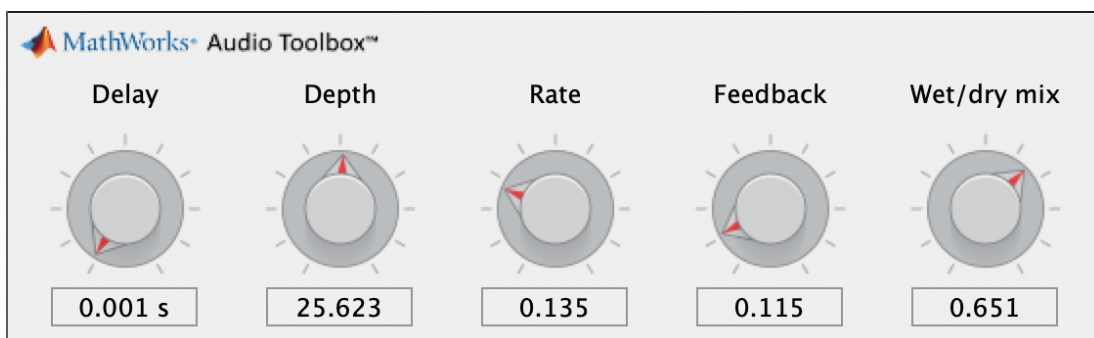
EK-2 Üretilen Chorus Plugininin Arayüz Görself

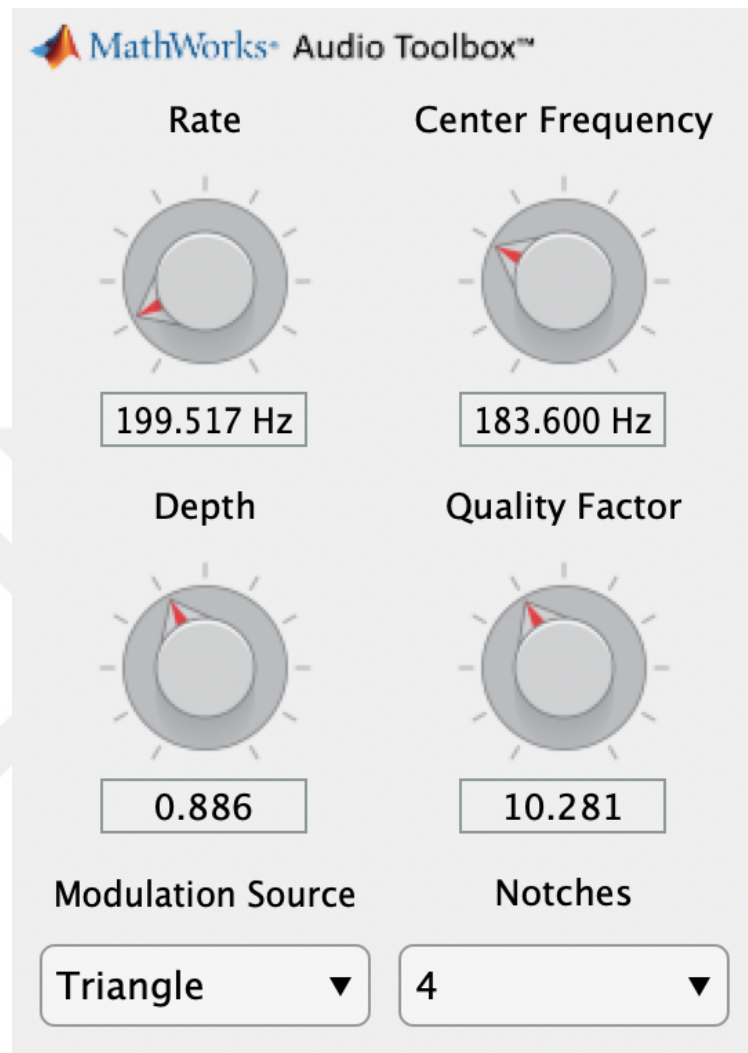


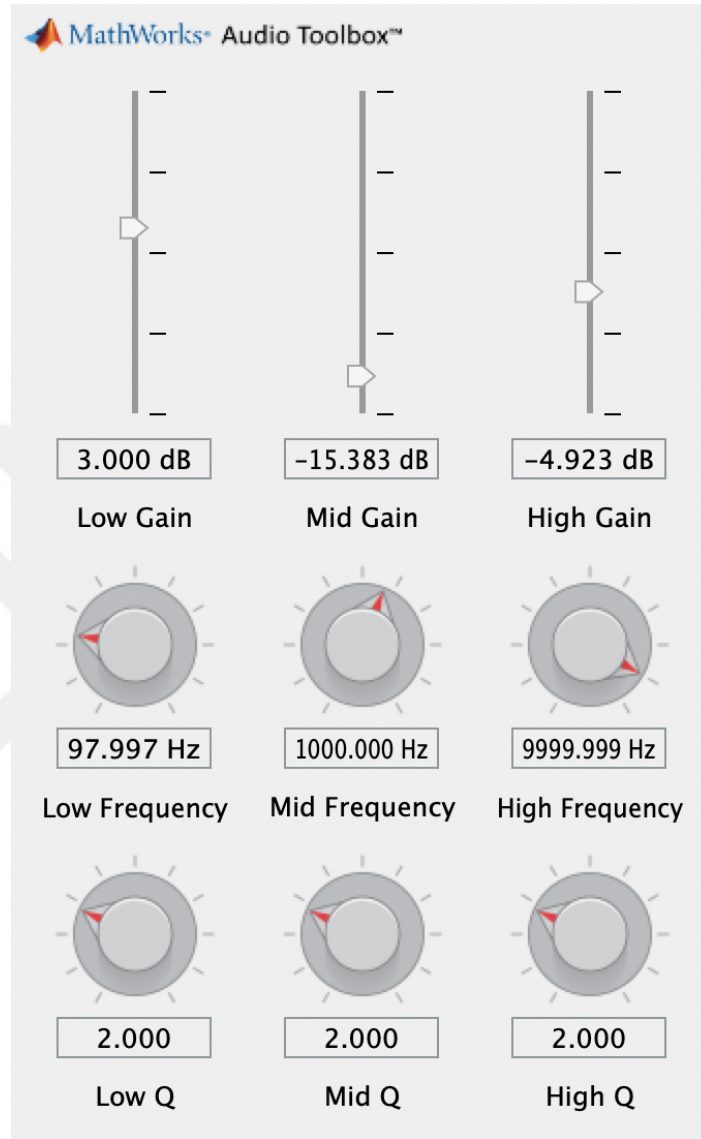
EK-3 Üretilen Echo Plugininin Arayüz Görseli



EK-4 Üretilen Flanger Plugininin Arayüz Görseli



EK-5 Üretilen Phaser Plugininin Arayüz Görseli

EK-6 Üretilen Parametric Equalizer Pluginin Arayüz Görseli

EK-7: YAYIMLAMA VE FİKRÎ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin tamamını veya herhangi bir kısmını, basılı (kâğıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Akdeniz Üniversitesi'ne verdiğimi bildiririm. Bu izinle Üniversite'ye verilen kullanım hakları dışındaki tüm fikrî mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalara (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan, telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinleri yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversite'ye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan **Lisansüstü Tezlerin Elektronik Ortamda Toplanması Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge*** kapsamında tezim aşağıda belirtilen haricinde YÖK Ulusal Tez Merkezi Erişim Sisteminde erişime açılır.

- ☐ Enstitü Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir. (1)
- ☐ Enstitü Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ... ay ertelenmiştir. (2)
- ☐ Tezimle ilgili gizlilik kararı verilmiştir. (3)

...../...../.....

Ekin Baran YILDIZ

*Lisansüstü Tezlerin Elektronik Ortamda Toplanması
Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge

- (1) Madde 6.1. Lisansüstü tezle ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine Enstitü Yönetim Kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

- (2) Madde 6.2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmasını ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkânı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve Enstitü Anabilim Dalının uygun görüşü üzerine Enstitü Yönetim Kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.
- (3) Madde 7.1. Ulusal çıkarılan veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan iş birliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile Enstitünün uygun görüşü üzerine Üniversite Yönetim Kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

Madde 7.2. Gizlilik kararı verilen tezler gizlilik süresince Enstitü tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

Tez Danışmanının önerisi ve Enstitü Anabilim/Anasanat Dalının uygun görüşü üzerine Enstitü Yönetim Kurulu tarafından karar verilir.

EK-8: ETİK BEYANI

Akdeniz Üniversitesi Güzel Sanatlar Enstitüsü Tez Yazım Kılavuzu'na uygun olarak hazırladığım bu Tezde

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı bütün bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin bütününe kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- bu Tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir Tez çalışması olarak sunmadığımı,

beyan ederim.

...../...../.....

Ekin Baran YILDIZ

EK-9: TEZ ORJİNALLİK RAPORU
AKDENİZ ÜNİVERSİTESİ
Güzel Sanatlar Enstitüsü

Tez Başlığı: Yazılım Tabanlı Tını Efektü Üretimi Üzerine Örnek Bir Araştırma

Yukarıda başlığı verilen Tezimin tamamı Turnitin adlı intihal programı aracılığı ile Tez Danışmanım tarafından kontrol edilmiştir. (Tez 106, Tez 206, Tez 010 Formları) Kontrol sonucunda (kaynakça hariç) aşağıdaki veriler elde edilmiştir:

Raporlama Tarihi	Sayfa Sayısı	Karakter Sayısı	Savunma Tarihi	Benzerlik Oranı (%)	Gönderim Numarası
22.06.2023	110	103078	22.06.2023	6	2120777409

Akdeniz Üniversitesi Güzel Sanatlar Enstitüsü Tez Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim. 22/06/2023

Ekin Baran YILDIZ

Öğrenci No: 202053011003

Anasanat/Anabilim Dalı: Müzik Anasanat Dalı

Program:

Yüksek Lisans	Sanatta Yeterlik	Doktora	Bütünleşik Doktora
X			

DANIŞMAN ONAYI

UYGUNDUR.

Prof. Dr. Hasan ARAPGİRLİOĞLU

EK-10: Master's/Proficiency in Art/PhD Thesis/ Art Work Report**Originality Report****AKDENİZ UNIVERSITY****Institute of FineArts**

Title: A Sample Research on Software Based Timbre Effect Generation

The whole thesis/art work report is checked by my supervisor, using Turnitin plagiarism detection software taking into consideration the below mentioned filtering options. (Tez 106, Tez 206, Tez 010 Forms) According to the originality report, obtained data are as follows.(Bibliography excluded)

Date Submitted	Page Count	Character Count	Date of Thesis Defence	Similarity Index (%)	Submission ID
22.06.2023	110	103078	22.06.2023	6	2120777409

I declare that I have carefully read the Akdeniz University Institute of Fine Arts Guidelines for Obtaining and Using Thesis Originality Reports; that my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations, I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge. I respectfully submit this for approval.

22/06/2023

Student No: 202053011003

Ekin Baran YILDIZ

Department: Music

Program/Degree:

Master's	Proficiency in Art	PhD	Integrated Phd
X			

SUPERVISOR APPROVAL

APPROVED.

Prof. Dr. Hasan ARAPGİRLİOĞLU