

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
GÜZEL SANATLAR ENSTİTÜSÜ
MÜZİK TEKNOLOJİSİ BİLİM DALI
Yüksek Lisans Tezi

**BİLGİSAYAR TABANLI MÜZİK ANALİZİNDE
MIDI TOOLBOX KULLANIMI**

Hazırlayan
Orkun Murat YILDIRIM

Danışman
Prof. Dr. Cihan IŞIKHAN

İZMİR / 2020

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Bilgisayar Tabanlı Müzik Analizinde MIDI Toolbox Kullanımı” adlı çalışmanın, tarafımdan bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın yazıldığını, yararlandığım eserlerin bibliyografyada gösterilenden oluştuğunu ve bunlara atıf yapılarak yararlanılmış olduğunu belirtir, bunu onurumla doğrularım.

19/ 06/ 2020

Orkun Murat YILDIRIM

TUTANAK

Dokuz Eylül Üniversitesi Güzel Sanatlar Enstitüsü'nün/...../...../ tarih ve.....sayılı toplantısında oluşturulan jüri, Lisansüstü Öğretim Yönetmeliği'nin.....maddesine göre Müzik Teknolojisi Bilim Dalı, Yüksek Lisans öğrencisi Orkun Murat YILDIRIM'ın “Bilgisayar Tabanlı Müzik Analizinde MIDI Toolbox Kullanımı” konulu tezini incelenmiş ve aday/...../...../ tarihinde, saat’da jüri önünde tez savunmasına alınmıştır.

Adayın kişisel çalışmaya dayanan tezini savunmasından sonra dakikalık süre içinde gerek tez konusu, gerekse tezin dayanağı olan bilim dallarından jüri üyeleri tarafından sorulan sorulara verdiği cevaplar değerlendirilerek tezin olduğuna oy ile karar verilmiştir.

BAŞKAN**ÜYE****ÜYE**

ÖZET

Bilgisayar tabanlı müzik analizinde tonallık, perde, aralık, ezgi çizgisi vb. gibi müziğin birçok farklı özelliği/özniteliği analiz edilebilmektedir. Müzik analizi, doğrudan ses dosyaları (*audio*) üzerinden yapılabildiği gibi müziği temsil eden sembolik veriler kullanılarak da yapılabilmektedir. Bilgisayar ortamında müziğin sembolik temsilinde günümüzde en yaygın kullanılan araçlardan biri MIDI'dir. MIDI verileri üzerinden müzik analizi yapmaya olanak sağlayan birçok farklı araç geliştirilmiştir. Günümüzde yaygın olarak kullanılan araçların başında, Matlab yazılımında kullanılmak üzere geliştirilen MIDI ToolBox gelmektedir.

Müzik analizinde MIDI ToolBox kullanımı, ses dosyalarını MIDI'ye dönüştürme ihtiyacını doğurmaktadır. Yapılacak analiz işleminin sağlıklı olması için oluşturulan MIDI dosyasının orijinal sesle birebir uyumlu olması gerekmektedir. Ancak genellikle MIDI dosyalar notasyon yazılımları aracılığıyla ve elle hazırlanmaktadır. Günümüzde manuel yöntemlerle MIDI dosyaları hazırlamanın yanında, sesten MIDI'ye dönüşüm yapan dönüştürücü yazılımlar geliştirilmiştir.

Bu çalışmada, öncelikle MIDI ToolBox'ın özellikleri ve kullanımı açıklanırken; ses dosyasından MIDI'ye dönüşüm yapan yazılımlar tespit edilerek belirlenen örnek müzikler bu yazılımlar tarafından MIDI'ye dönüştürülmüş ve yapılan karşılaştırmalar doğrultusunda yazılımların başarı sıralaması oluşturulmuştur.

ABSTRACT

In computer-based music analysis has been created a music database in order to analysis music features such as key, pitch, interval distribution, melodic contour, etc. While generally music analysis operations are applied into raw data/audio files, symbolically representing of musical data are often used for such operations. One of the most widely used to demonstrate of music as symbolically is MIDI and many different tools have been developed to use MIDI in music analysis. In terms of prevalence, the foremost one is MIDI ToolBox in MatLab.

The use of MIDI ToolBox in music analysis raises the need to convert audio files to MIDI and MIDI file created must be exactly compatible with the original/raw audio for the analysis process to be performed. However, MIDI data/files are usually prepared manually and through notation software. Today, besides preparing MIDI with manual methods, some of software that converts from audio to MIDI has been developed.

In this study, while explaining the features and usage of MIDI ToolBox in MatLab, some of unique software that converts from audio to MIDI has been detected. Then, some music files in a pool has been converted to MIDI by this software. As a result of this study, the software that converts audio to MIDI has been compared with success.

ÖNSÖZ

Çalışmamın başından son anına kadar büyük bir sabırla ilgilenen, değerli bilgi ve birikimlerini paylaşarak çalışmamı yönlendiren, mesai saati gözetmeksizin ihtiyacım olan her an vaktini ayıran değerli hocam ve tez danışmanım Prof. Dr. Cihan IŞIKHAN'a göstermiş olduğu özveri ve ilgisi için teşekkür ederim.

Lisans ve yüksek lisans eğitim hayatım boyunca çalışma fırsatı bulduğum, bilgi ve birikimlerini paylaşarak mesleki gelişimime katkı sağlayan ve farklı bakış açıları kazanmama vesile olan, çalışma fırsatı bulduğum için kendimi her zaman şanslı hissettiğim bütün hocalarıma saygılarımı sunar ve teşekkür ederim.

Hayatımın her anında maddi ve manevi desteğini hiçbir zaman esirgemeyen, onca zorluğa rağmen ilgi duyduğum alanda eğitim görmem için bütün imkanlarını seferber eden, bu çalışmam boyunca destekleriyle beni motive eden, beni ben yapan bütün değerlerin kaynağı annem Filiz YUSUF'a ve babam Şener YILDIRIM'a teşekkür ederim.

Son olarak 12 yaşındayken tanıştığım ve üniversite eğitimime başlayana kadar uzun bir süre çalışma şansı elde ettiğim, hayatıma yön veren değerli hocam Ali POLAT'a saygı, sevgi ve minnetlerimi sunuyorum.

İÇİNDEKİLER

YEMİN METNİ.....	ii
TUTANAK	iii
ÖZET.....	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
ŞEKİLLER LİSTESİ	ix
TABLolar LİSTESİ	xi
KISALTMALAR	xii
GİRİŞ	1
1.BÖLÜM MÜZİK ANALİZİNDE MIDI TOOLBOX KULLANIMI.....	8
1.1 MIDI ToolBox	9
1.2 MTB Genel Özellikler.....	12
1.3 MTB Fonksiyonlar	13
1.3.1 Dönüştürme	13
1.3.2 Üretim	15
1.3.3 Filtre	17
1.3.4 Meta.....	18
1.3.5 Çizim	18
1.3.6 İstatistiksel.....	21
1.3.7 Anahtar Bulma	22
1.3.8 Kontur	28
1.3.9 Melodik	28
1.3.10 Bölümleme	30

2. BÖLÜM: AUDIO-to-MIDI YAZILIMLARIN BAŞARI KARŞILAŞTIRMASI.....	34
2.1 Karşılaştırmada Kullanılan Yazılımlar	35
2.1.1 Ableton Live.....	35
2.1.2 Amazing MIDI	36
2.1.3 FL Studio.....	37
2.1.4 Intelli Score	38
2.1.5 Logic Pro	38
2.1.6 Melodyne.....	39
2.1.7 WIDI	40
2.2 Müzik Havuzu.....	41
2.2.1 Tek Çalgı Teksesli.....	42
2.2.2 Tek Çalgı Çoksesli	42
2.2.3 Çok Çalgı Teksesli	43
2.2.4 Çok Çalgı Çoksesli:.....	44
2.3 Karşılaştırmalar	45
2.3.1 Kategori 1: TÇT	47
2.3.2 Kategori 2: TÇÇ	51
2.3.3 Kategori 3: ÇÇT	55
2.3.4 Kategori 4: ÇÇÇ.....	59
2.4 Ölçümleme ve Sonuçlar	63
2.5. Sonuçların Bazı MIDI Toolbox Fonksiyonlarında Kullanımı	67
SONUÇ	71
KAYNAKÇA	73
ÖZGEÇMİŞ	

ŞEKİLLER LİSTESİ

Şekil 1- <i>nmat</i> 'deki <i>readmidi</i> fonksiyonu çıktısı	11
Şekil 2- <i>readmidi</i> fonksiyonu ile elde edilen <i>nmat</i>	15
Şekil 3- <i>nmat</i> 'den elde edilen dalga biçimi	16
Şekil 4- <i>Pianoroll</i> fonksiyonu kullanılarak elde edilen görsel	19
Şekil 5- <i>Plotmelcontour</i> fonksiyonu ile elde edilen görsel.....	20
Şekil 6- Nota dağılımının hesaplanmasıyla elde edilen görsel	22
Şekil 7- <i>keymode</i> fonksiyonu kullanımı.....	23
Şekil 8- Nota dağılım haritası	24
Şekil 9- <i>kkkey</i> fonksiyonu kullanımı.....	24
Şekil 10- <i>meldistance</i> fonksiyonunu kullanmak için <i>nmat</i> yapısına dönüştürme işlemi	29
Şekil 11- <i>meldistance</i> fonksiyonunun kullanımı.....	30
Şekil 12- Aynı ezgiler üzerinde <i>meldistance</i> fonksiyonunun kullanımı.....	30
Şekil 13- <i>segmentgestalt</i> fonksiyonu ile elde edilen ve bölümlleme noktalarını gösteren görsel	31
Şekil 15- Dönüştürme işlemleri esnasında Ableton Live yazılımından bir görüntü.....	36
Şekil 16- Dönüştürme işlemleri esnasında Amazing MIDI yazılımından bir görüntü	37
Şekil 17- Dönüştürme işlemleri esnasında FL Studio yazılımından bir görüntü.....	37
Şekil 18- Dönüştürme işlemleri esnasında Intelli Score yazılımından bir görüntü	38
Şekil 19- Dönüştürme işlemleri esnasında Logic Pro yazılımından bir görüntü	39
Şekil 20- Dönüştürme işlemleri esnasında Melodyne yazılımından bir görüntü.....	39
Şekil 21- Dönüştürme işlemleri esnasında Widi yazılımından bir görüntü.....	40
Şekil 22- Dönüştürme işlemi için kullanılacak yaratının ilk iki ölçüsü.....	42
Şekil 23- Seçilen yaratının dönüşüm işlemi için kullanılacak kısmı.	43
Şekil 24- Seçilen yaratının dönüşüm işleminde kullanılacak olan kısmı	44
Şekil 25- Yaratının dönüşüm işlemi esnasında kullanılacak bölümünden bir görüntü ..	44
Şekil 26- MIDI'ye dönüştürülecek ses dosyası üzerindeki düzenleme işlemleri esnasında Ableton Live yazılımından bir görüntü.....	45
Şekil 27- Yazılımların karşılaştırılmasında kullanılan iş akışı	46
Şekil 28- Karşılaştırma işlemi esnasında Ableton Live yazılımından bir görüntü	47

Şekil 29- TÇT orijinal MIDI.....	47
Şekil 30- TÇT Ableton Live	48
Şekil 31- TÇT Amazing MIDI.....	48
Şekil 32- TÇT Intelli Score.....	49
Şekil 33- TÇT Logic Pro	50
Şekil 34- TÇT Melodyne	50
Şekil 35- TÇT Widi	51
Şekil 36- TÇÇ orijinal MIDI	51
Şekil 37- TÇÇ Ableton Live.....	52
Şekil 38- TÇÇ Amazing MIDI	52
Şekil 39- TÇÇ FL Studio	53
Şekil 40- TÇÇ Intelli Score	53
Şekil 41- TÇÇ Logic Pro	54
Şekil 42- TÇÇ Melodyne.....	54
Şekil 43- TÇÇ Widi	55
Şekil 44- ÇÇT orijinal MIDI	55
Şekil 45- ÇÇT Ableton Live.....	56
Şekil 46- ÇÇT Amazing MIDI	56
Şekil 47- ÇÇT FL Studio	57
Şekil 48- ÇÇT Intelli Score	57
Şekil 49- ÇÇT Logic Pro	57
Şekil 50- ÇÇT Melodyne.....	58
Şekil 51- ÇÇT Widi	58
Şekil 52- ÇÇÇ orijinal MIDI	59
Şekil 53- ÇÇÇ Ableton Live.....	59
Şekil 54- ÇÇÇ Amazing MIDI	60
Şekil 55- ÇÇÇ FL Studio.....	60
Şekil 56- ÇÇÇ Intelli Score	61
Şekil 57- ÇÇÇ Logic Pro	61
Şekil 58- ÇÇÇ Melodyne.....	62
Şekil 59- ÇÇÇ Widi	62

TABLOLAR LİSTESİ

Tablo 1- <i>nmat</i> içindeki sütunların ifade ettiği değerler	11
Tablo 2- Dönüştürme fonksiyonları ve görevleri	14
Tablo 3- Üretim fonksiyonları ve görevleri.....	16
Tablo 4- Filtre fonksiyonları ve görevleri	17
Tablo 5- Meta fonksiyonları ve görevleri.....	18
Tablo 6- Çizim fonksiyonları ve görevleri	19
Tablo 7- İstatistiksel fonksiyonlar ve görevleri.....	21
Tablo 8- Anahtar bulma fonksiyonları ve görevleri	23
Tablo 9- Tonalitelerin rakam olarak karşılıkları	25
Tablo 10- <i>Kkkey</i> fonksiyonu uygulanacak yaratılar	25
Tablo 11- <i>Kkkey</i> fonksiyonu sonuçları	27
Tablo 12- Kontur fonksiyonları ve görevleri.....	28
Tablo 13- Melodik fonksiyonlar ve görevleri	28
Tablo 14- Bölümleme fonksiyonları ve görevleri	31
Tablo 15- Tespit edilen ve karşılaştırmada kullanılan yazılımlar	35
Tablo 16- Dönüşüm işlemi için oluşturulan müzik havuzundaki kategoriler	41
Tablo 17- Karşılaştırma sonuçlarına göre F-Measure parametre değerleri.....	64
Tablo 18- Tüm kategorilerde yazılımların F-Measure, Precision ve Recall sonuçları...	65
Tablo 19- Tüm kategorilerde F-Measure sonuçları.....	66
Tablo 20- Fonksiyonların TÇT kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar	67
Tablo 21- Fonksiyonların TÇÇ kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar	68
Tablo 22- Fonksiyonların ÇÇT kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar	69
Tablo 23- Fonksiyonların ÇÇÇ kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar	69

KISALTMALAR

ÇÇÇ: Çok çalgı Çoksesli

ÇÇT: Çok Çalgı Teksesli

DAW (Digital Audio Workstation): Dijital Ses İşleme İstasyonu

F: F-Measure

FN (False Negative): Yanlış Saptanmamış

FP (False Positive): Yanlış Saptanmış

Hz: Hertz

MIDI: Musical Instrument Digital Interface

MTB: MIDI ToolBox

nmat: Notematrix

P (Precision): Kesinlik

R (Recall): Hassasiyet

T&P: Tenney & Polansky

TÇÇ: Tek Çalgı Çoksesli

TÇT: Tek Çalgı Teksesli

TP (True Positive): Doğru Saptanmış

GİRİŞ

Müzik teknolojisi, ses kayıt teknolojisinin gelişmesiyle birlikte özgün bir çalışma alanı olarak ortaya çıkmıştır. Müziğin üretim aşamasından kayıt aşamasına, buradan da dinleyiciye ulaştığı son ana kadar gerçekleşen bütün aşamalar, müzik teknolojisinin uğraşı alanına girmektedir. Müzik bir taraftan stüdyo ortamında kaydedilerek dinleyiciye ulaştırılırken öte yandan geniş kitleler karşısında canlı olarak icra edilmektedir. Müziğin canlı olarak icra edildiği etkinliklerde kullanılan seslendirme sistemleri de müzik teknolojisinin ana alanlarından biridir.

Günümüzde seslendirme sistemleri ve stüdyo ortamındaki kayıt işlemleri, kendi başına farklı sektörler haline gelmiştir. Müzik teknolojisi her iki sektör ile doğrudan bağlantılı bir çalışma alanıdır. Her ne kadar müzik temelli bir çalışma alanı olsa da sesin doğası ve hayatımızdaki yeri, müzik teknolojisinin çalışma alanını genişletmektedir. Günümüzde sesin, dolayısıyla müziğin doğrudan etkileşimde olduğu akustik bilimi de müzik teknolojisinin temel uğraşı alanlarından biridir.

Müzik teknolojisi ses kayıt, seslendirme sistemleri, akustik, yayıncılık, ses tasarımı gibi birçok farklı alanı kapsayan disiplinler arası bir çalışma alanıdır. Yapısı gereği müzik, sinema, oyun, eğlence vb. gibi birçok farklı sektörle doğrudan bağlantılıdır. Bütün bunların yanında, müzik üzerinden veri toplama, müziğin sembolik verilerle temsili ve müzik analizi gibi çalışma alanları da müzik teknolojisinin uğraşları arasında yer almaktadır.

Müzik analizi bilgisayar bilimleri, istatistik, müzikoloji, müzik teknolojisi ve bilgi teknolojisi gibi birçok farklı uzmanlık alanının bir arada bulunduğu, disiplinler arası bir çalışma alanıdır. Bilgisayar tabanlı müzik analizinde kullanılmak üzere birçok farklı teknik ve yazılım geliştirilmiştir¹.Yazılımların geliştirilmesi, bilgisayar bilimleri ve veri

¹ “SOMeJB Musipedia, Themefinder, Orpheus,PROMS,QBH, Meldex, C-Brahms CubyHum, SoundCompass, SuperMBox...” bu yazılımların yalnızca birkaçı. Daha detaylı bilgi için Bkz: Typke, R., Wiering, F., & Veltkamp, R. C. (2005, September). A Survey of Music Information Retrieval Systems. In *Proc. 6th International Conference on Music Information Retrieval* (pp. 153-160). Queen Mary, University of London.

analizi tekniklerinde yaşanan teknolojik gelişmelerle doğru orantılıdır.

Bilgisayar tabanlı sistemlerde müzik analizi, müziklerden toplanan veriler aracılığıyla oluşturulan veri tabanı üzerinden gerçekleştirilmektedir. Veri bankası oluşturacak düzeyde müzikten veri toplamak, müziğe ait frekans, genlik, süre vs. gibi öznitelikleri tespit edip, ayrıştırıp onları sanal ortamda salt birer veri olarak depolamaktır (Işıkhani, 2018: 3). Toplanan veriler üzerinden, kullanıcıların ihtiyaçları doğrultusunda müziğin birçok farklı özelliği analiz edilebilmektedir. Yapılan işlemlerle elde edilen sonuçlar görselleştirilerek, müziği görsel olarak analiz edebilme imkânı oluşmaktadır. Örneğin, analizi yapılan müziğin hangi tonallığa ait olduğu ya da zaman içerisindeki tonal değişiklikleri saptanabilmektedir. Müziğin içinde bulunan nota sayısı hızlı bir şekilde hesaplanabilmekte, ardışık nota grupları tespit edilebilmektedir. Melodinin sahip olduğu çizgi (*contour*) hesaplanabilmekte ve ezgi çizgisinin görsel analizi yapılabilmektedir.

Mevcut müzik analizi yazılımları, temel olarak teorisyenlerin ve müzikologların elle gerçekleştirdiklerini otomatikleştirmede yardımcı olan araçlar olmuştur (McKay, Fujinaga, 2006: 65). Bilgisayar tabanlı müzik analizi sistemlerinde işlemler, el ile yapılan çalışmalara göre çok daha hızlı olmakla beraber analiz yapan araştırmacıların uzmanlık alanlarından bağımsız olarak farklı müzik türlerinde aynı tutarlılıkla sonuç alma imkânı yaratmaktadır. Müziklerin teknik olarak birbirleriyle olan benzerlikleri ve etkileşimleri üzerine yapılan araştırmalarda bilgisayar tabanlı analiz yöntemleri kullanılmaktadır. Müzik analizi, müzik bilimi alanında çalışma yapan araştırmacılar tarafından kullanılan bir yöntem olmakla beraber, müzisyenlere ve sıradan kullanıcılara da farklı imkânlar sağlamaktadır.

Bilgisayar tabanlı sistemlerde müzik analizi, kendi içinde oldukça karmaşık bir yapıya sahip olsa da günümüzde yaygın olarak kullanılan dizüstü ya da masaüstü bilgisayarlar çoğu analiz işlemi için yeterli olmaktadır. Bu sebepten bilgisayar ortamında müzik analizi, teknik şartlar açısından daha geniş kullanım potansiyeline sahiptir. Örneğin, bir besteci bilgisayar ortamında müzik analizi yaparak diğer yaratıları daha hızlı bir şekilde inceleyebilmekte ve kendi üretimini

geliştirebilmektedir².

Diğer taraftan, bilgisayar tabanlı müzik analiz yazılımları sayesinde benzer müzikler karşılaştırılabilmekte ve iki müzik arasındaki benzerlik ölçülebilmektedir. Bu özellik sayesinde bilgisayar tabanlı müzik analizi, telif hakkı konusunda yapılan araştırmalarda kullanılabilir hale gelmektedir. Analiz yöntemleri ve hesaplamalarda kullanılan algoritmaları bilim insanları tarafından onaylanmış yazılımlar, telif hakkı ile ilgili yapılan çalışmalarda geleneksel yöntemlere kıyasla daha hızlı ve tutarlı sonuç alma imkânı sağlamaktadır³.

Müziği rastgele ses kaynaklarından ayıran özelliklerden biri, müziğin düzenlendiği hiyerarşik yapıdır (Muller, 2015: 167). Müziğin bu yapısı müzik analizi için oldukça önemli, hatta temel sayılabilecek bir noktadır. Müziğin içinde bulunan seslerin perde yüksekliği, süreleri gibi özellikleri, zamana bağlı olarak değişkenlik göstermektedir. Bu değişkenlikler ses veya nota olayları (*note events*) olarak adlandırılmaktadır. Müziğin içerdiği nota olayları bir araya gelerek, çeşitli motifler ve ifadeleri oluşturmaktadır. Bu motifler ise birbirleriyle birleşerek müziğin genelini oluşturmaktadır. Farklı teknikler olmakla beraber, müziğin içindeki nota olayları incelenerek analiz işlemleri yapılmakta ve bütünü hakkında bilgi edinmeye yardımcı olmaktadır. Aynı zamanda tespit edilen motifler farklı motiflerle karşılaştırılarak, karşılaştırmalı analiz işlemleri yapılmaktadır

Bilgisayar tabanlı müzik analiz işlemleri iki ana guruba ayrılmaktadır (Typke, 2015). Bunlardan ilki direkt olarak ses dosyaları üzerinden yapılan analiz işlemleridir. Bilgisayarda kayıtlı olan ses dosyaları üzerinden müzik analizi yapmak üzere geliştirilmiş sistemler olduğu gibi, mikrofon tarafından algılanan sesler üzerinden analiz yapma yeteneği olan sistemler de bulunmaktadır.

² Ayrıntılı bilgi için Bkz.: Smith, C. H., & Georges, P. (2014). Composer Similarities Through “The Classical Music Navigator”: Similarity Inference from Composer Influences. *Empirical Studies of the Arts*, 32(2), 205-229.

³ Ayrıntılı bilgi için Bkz.: Marshall, L., & Frith, S. (2013). *Music and Copyright*. Routledge.

Veri tabanına dayalı müzik analizinde kullanılan diğer bir yöntem ise müziğin sembolik verilerle temsil edilerek, analizin müziği temsil eden sembolik veriler üzerinden gerçekleştirilmesidir (Velardo vd. 2016). Müziğin içerdiği perde yüksekliği, vuruş duyarlılığı, nota süresi gibi özelliklerini sembolik verilerle temsil eden farklı formatlar bulunmaktadır. Analizin başarılı olması, müziği temsil eden sembolik verilerin temsil gücünün yüksek olmasıyla doğru orantılıdır.

Sembolik veriler üzerinde yapılan analiz işlemlerinde, müzikten veri toplama işlemi için farklı metotlar geliştirilmiştir. Teksesli müziklerin analizinde dizi tabanlı (*string-based*) metotlar kullanılmaktadır. Teksesli müzikler, tek boyutlu karakter dizileriyle temsil edilebilmekte; burada her karakter bir notayı veya bir çift ardışık notayı tanımlamaktadır. Diziler ezginin konturunu, nota dizilerini, aralık dizileri gibi özelliklerini temsil etmektedir. Bu metotta müziğin analizi, tespit edilen diziler üzerinden yapılmaktadır. Çok sesli müziklerin analizindeyse küme tabanlı (*set-based*) yöntemler kullanılmaktadır. Bu yöntemde müziğin içinde bulunan notaların süresi, başlangıç zamanı ve perde yükseklikleri tespit edilmektedir. Tespit edilen özniteliklerin dağılımı hesaplanarak bu dağılımlar üzerinden analiz işlemleri gerçekleştirilmektedir (Typke, 2015).

Müzik birçok farklı şekilde temsil edilebilmektedir. Manuel ortamda müzik temsili notalar aracılığıyla yapılmaktadır. Notalar, seslerin zaman içindeki uzunluğunu, perde yüksekliğini ifade etmek için kullanılan yazılı işaretlerdir. Bir besteci tarafından yazılı olarak notaya dökülen bir müzik, nota okumayı bilen bütün müzisyenler tarafından anlaşılabilen, icracılar tarafından müziğe dönüştürülebilmektedir.

Bilgisayar ortamındaysa durum farklıdır. Müziğin içindeki perde yüksekliği, vuruş duyarlılığı, notaların süresi gibi öznitelikleri, sembolik veriler ile temsil eden farklı protokoller geliştirilmiştir. Bu protokoller, dijital ortamda müziğin özniteliklerini dijital bir veri haline getirmektedir. Müziğin temsili gerçekleştirilen dijital sembolik veriler sanal ortamda depolanabilmekte ve karşılıklı veri iletimi yapılabilmektedir (Velardo vd. 2016).

Müziğin sembolik temsilinde günümüzde en yaygın kullanılan formatlarından biri MIDI'dir (Musical Instrument Digital Interface). MIDI, müzik donanımları arasında ortak/standart bir diyalogu sağlamak amacıyla müziğin temel bileşenlerini sembolik verilere dönüştüren güçlü bir protokoldür (Durmaz, 2000:23). Bu nedenle MIDI müziğin temel tüm özniteliklerini içerir. 1980'li yılların başında geliştirilen MIDI günümüzde hala yaygın bir şekilde kullanılmaktadır. Başlangıçta MIDI'nin tasarlanma amacı, birbirinden farklı firmaların ürettiği farklı elektronik müzik aletlerinin beraber kullanımına imkân sağlamaktır. Ancak ilerleyen yıllarda MIDI verilerinin bilgisayar ortamında saklanması ve karşılıklı değiş tokuş yapması mümkün hale geldi. Bu sayede MIDI verileri bilgisayar ortamında .mid uzantılı dosya olarak kaydedilebilmekte ve birçok farklı amaç için kullanılabilir.

Daha önce bahsedildiği üzere, müziğin sembolik verilerle temsilinde birçok farklı protokol geliştirilmiştir. Geliştirilen her bir protokolün, müziği temsil gücü birbirlerinden farklıdır. Ancak geliştirilmiş olan hiçbir protokol, müziği tam olarak temsil etmemektedir. MIDI gibi temsil gücü yüksek diller, müziği her yönüyle eksiksiz olarak temsil edemese de birçok işleme olanak sağlayacak temsil gücüne sahiptir.

MIDI verileri üzerinden müzik analizi yapmaya olanak sağlayan birçok farklı araç geliştirilmiştir. Günümüzde yaygın olarak kullanılan araçların başında, Matlab yazılımında kullanılmak üzere geliştirilen **MIDI ToolBox (MTB)** gelmektedir.

MTB, Finlandiyalı iki bilim insanı Eerola ve Toiviainen tarafından 2004 yılında geliştirilen ve tamamen MIDI tabanlı sembolik veriler üzerinden detaylı müzik analizine imkân sağlayan MatLab fonksiyonlu bir araçtır. MTB, müzik teknolojisinin tüm bu çalışma boyunca anlatılan diğer aktif alanlarıyla birlikte farklı bir alt alanı olan müzikte analizde günümüzde en yaygın kullanılan araçlarından biri haline gelmiştir. İçeriğinde kullandığı onlarca fonksiyon, müzikte herhangi bir nedenle analiz yapılacak olanlar için geniş olanaklar içermektedir ve bu yapısıyla MTB, sembolik müzik analizinde güçlü bir araç olarak karşımıza çıkmaktadır.

Bu çalışmanın birinci bölümünde MTB ele alınmıştır. MTB'nin içerdiği fonksiyonlar kategoriler halinde incelenerek görevleri açıklanmıştır. Fonksiyonların

görevlerinin açıklanmasının yanı sıra, bazı fonksiyonlarla örnek uygulamalar yapılmış ve MTB'nin kullanımı hakkında bilgi verilmiştir.

MTB'de müzik analizinin MIDI verileri üzerinden yapılıyor olması, analizi yapılacak müzik ses dosyalarını MIDI dosyasına dönüştürme ihtiyacını doğurmaktadır. Müzik analizinin gerçekçi olması için oluşturulan MIDI dosyasının, analizi yapılacak müzikle birebir uyumlu olması gerekmektedir. Analizi yapılacak müzikler ile bire bir uyumlu MIDI dosyası oluşturmanın en kesin yolu manuel olarak elle dönüştürmektir. Günümüzde bu işlemi otomatik olarak gerçekleştiren, başka bir deyişle, ses dosyalarını MIDI formatına dönüştüren yazılımlar bulunmaktadır. Ancak bu yazılımların dönüştürme işlemini ne kadar doğru yaptığı bilinmemektedir.

İkinci bölümde, ses dosyasından MIDI'ye dönüşüm yapan yazılımlar tespit edilerek belirlenen örnek müzikler bu yazılımlar tarafından MIDI'ye dönüştürülmüştür. Elde edilen MIDI dosyaları incelenerek yazılımların performans karşılaştırması yapılmış ve yapılan karşılaştırmalar doğrultusunda yazılımların başarı sıralaması yapılmıştır.

Sesten MIDI'ye dönüşüm yapan yazılımlar, müzik analizinde ihtiyaç duyulan MIDI dosyası oluşturmak için yeterli beceriye sahip midir? MIDI dosyasını manuel hazırlamak yerine bu yazılımları kullanmak sağlıklı bir yöntem midir? Dolayısıyla bu çalışmada, yazılımların performansları ve elde ettiği sonuçlar incelenerek bu soruların cevabı aranacaktır.



1.BÖLÜM

MÜZİK ANALİZİNDE MIDI TOOLBOX KULLANIMI

1.BÖLÜM MÜZİK ANALİZİNDE MIDI TOOLBOX KULLANIMI

Ses dosyalarında müzik özniteliklerini birbirlerinden bağımsız şekilde bilgisayarla ayrıştırarak her birini birer veri haline getirmek, oldukça karmaşık bir işlemdir. Günümüzde bunu yapan bazı yazılımların varlığı konuşulsa da neredeyse hiçbiri tek başına aralarında Melodyne, Ableton, Cubase gibi yazılımların da bulunduğu etkili bir eklenti (*plug-in*) veya bir DAW (*Digital Audio Workstation*) kadar sonuç vermemektedir.

Bunun sebebi elbette, teknoloji ne kadar ilerlerse ilerlesin henüz sanıldığı kadar doğal olanın özüne ulaşamamasıdır. Ses doğal bir oluşumdur. Müzik ise doğal olanı daha da karmaşıktırmaktadır. Böyle karmaşık bir zinciri ancak en az onlar kadar karmaşık bir alıcının çözmesi mümkündür. Eldeki olanakların, gelişen teknoloji sayesinde geçmişle kıyaslanamayacak derecede fazla olduğu bir gerçektir. Ancak yine de sistemin bütününe bakıldığında, bir ses dosyasının doğal yapısı içinden öznitelik çıkarıp analiz yoluna gitmek için kusursuz bir sistem olmadığı bir gerçektir.

Böyle bir eylem için güçlü yazılımların varlığından bahsetmiştik. Aralarında MatLab gibi profesyonel veya MusicScope, E-Analysis, Sonic Visualiser gibi popüler olanların yanında; arama motoru özelliğiyle müzik sorgulama amacıyla yola çıkılmış ancak giderek müzik analizi ve işleme özelliğine bürünmüş Themefinder ve MelDex gibi olanları mevcuttur. Ancak genelde tüm bu yazılımların ortak noktası, ses dosyasına Fourier Kuramı⁴ başta olmak üzere yüzlerce filtreleme fonksiyonuyla yaklaşarak elde edilen verilerle işlem yapmaktır.

Yapılan işlemler sonucu ortaya çıkan veriler birçok farklı alanda kullanıma uygun olsa da her zaman beklenen sonucu verememektedir. Böyle bir durumda müzik teknolojisinde çareler tükenmeyerek ikinci bir seçenek devreye sokulmaktadır: Ses

⁴ Joseph Fourier'in kuramına göre: "Herhangi bir periyodik fonksiyon, sinus ve cosinus dalgalarının toplam veya fark serisi (sonsuzluğu) olarak ifade edilebilir. Böylece aslında tüm ses dalgaları, sinüs ve cosinus ses dalgalarının toplamından oluşmaktadır. Eldeki toplam dalga (*waveform*) bir sinüs veya cosinus fonksiyonuyla katsayılar sonsuzluğunda ayrıştırılabilirse, ses dalgasını oluşturan en küçük dalgaya (özniteliklere) ulaşılabilir (Christensen 2019: 117).

dosyalarını sembolik verilere dönüştürmek.

Tıpkı doğal ortamda müziğin notalarla sembolleştirilmesi gibi bilgisayar ortamında MIDI ile sembolleştirmek, bugün müzik analizinde en yaygın kullanılan yöntemlerden biri haline gelmiştir. Böyle bir sembolleştirmeye ses dosyaları doğal yapılarından kopararak sayısal verilere dönüştürülmekte ve sonraki işlemler için bu dönüşümden elde edilen veriler kullanılmaktadır. İşte bu veriler, 1982 yılında standart hale getirilen ve günümüzde bir-iki eklemeyle halen aynı standardı koruyan güçlü MIDI verileridir.

MIDI aslında, müzik donanımları arasında diyalogu sağlayan bir protokol olarak hayatına başlamıştır (Durmaz, 2000:23). Bu nedenle MIDI, müziğin tüm özniteliklerini içermektedir. 1980’li yılların başında geliştirilen MIDI, günümüzde hala yaygın bir şekilde kullanılmaktadır. Başlangıçta MIDI’nin tasarlanma amacı, birbirinden farklı firmaların ürettiği farklı elektronik müzik aletlerinin beraber kullanımına imkân sağlamaktır. Ancak ilerleyen yıllarda MIDI verilerinin bilgisayar ortamında saklanması ve karşılıklı değiş tokuş yapması mümkün hale gelmiştir. Bu sayede MIDI verileri bilgisayar ortamında .mid uzantılı dosya olarak kaydedilebilmekte ve birçok farklı amaç için kullanılabilir. İşte bu amaçlardan biri, bu çalışmanın temel konusunu oluşturan müziğin analiz amacıyla özniteliklerini çıkarabilmektir.

1.1 MIDI ToolBox

MIDI verileriyle müzik analizi yapmakta kullanılan ve günümüzde müzik sorgulama sistemleri sayesinde çok daha yaygın hale gelen araçlardan biri **MIDI ToolBox (MTB)**’dır. MTB, Finlandiyalı iki bilim insanı Eerola ve Toiviainen tarafından 2004 yılında geliştirilen ve tamamen MIDI tabanlı sembolik veriler üzerinden detaylı müzik analizine imkân sağlayan MatLab fonksiyonlu bir araçtır⁵. MTB, müzik teknolojisinin tüm bu çalışma boyunca anlatılan diğer aktif alanlarıyla birlikte farklı bir

⁵ MTB en ilk 2004 yılında ISMIR’de sunulmuş (Eerola, T., & Toiviainen, P. (2004), MIR in Matlab: The MIDI Toolbox, *International Symposium of Music Information Retrieval (ISMIR)*), ardından geliştirilerek tüm kullanıcıya açılmıştır. (<https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/miditoolbox>).

alt alanı olan müzikte analizde günümüzde en yaygın kullanılan araçlarından biri haline gelmiştir. İçeriğinde kullandığı onlarca fonksiyon ve müzikte herhangi bir nedenle analiz yapacak olanlar için geniş olanaklar içermektedir. Bu yapıyla MTB, sembolik müzik analizinde güçlü bir araç olarak karşımıza çıkmaktadır.

MTB, MatLab'de kullanılmak üzere tasarlanmış fonksiyonlara sahiptir ve toplam 94 ana fonksiyon içerir. Bu fonksiyonlar, MIDI dosyaları içindeki perde isimleri, vuruş duyarlılığı, nota süreleri ve ezgisel çizginin karakteri gibi farklı alanlarda analiz yapma imkânı sağlamaktadır (Işıkhân, 2018:4).

MTB'nin MIDI analizi için sağladığı Matlab fonksiyonları, tıpkı Matlab gibi modüler bir yapıya sahiptir. Bu özellik, kullanıcılara spesifik analiz ihtiyaçları doğrultusunda, fonksiyonlar üzerinde çeşitli geliştirmeler ve uyarlamalar yapmaya olanak sağlamaktadır. Sahip olduğu modüler yapı, MTB'yi müzik analizinde daha güçlü bir araç haline getirmektedir.

MTB' de MIDI dosyaları içerdiği her bir notanın başlangıç zamanı, MIDI kanalı, perde yüksekliği ve vuruş duyarlılığı hakkında bilgi içeren bir matris ile temsil edilmektedir. Bu matris sistemi MTB içinde *notematrix* (*nmat*) olarak adlandırılmaktadır. MIDI dosyalarını temsil eden *nmat*'ler bir dönüştürme fonksiyonu olan *readmidi* aracılığıyla oluşturulmaktadır. Bu fonksiyonun detayları ve analizi yapılacak MIDI dosyalarının *nmat*'e dönüştürülme işlemi ilerleyen konularda ele alınacaktır.

Ableton Live yazılımında el ile hazırlanan ve do majör dizisinin seslerini içeren MIDI dosyası, MatLab içinde *nmat*'e dönüştürüldüğünde Şekil 1'de sunulduğu gibi görünmektedir.

```

Command Window
>> readmidi ('example.mid')

ans =

    0    0.1354    0 60.0000 100.0000    0    0.0677
  0.5000    0.1354    0 62.0000  70.0000    0.2500    0.0677
  1.0000    0.1875    0 64.0000  67.0000    0.5000    0.0938
  1.5000    0.2500    0 65.0000  37.0000    0.7500    0.1250
  2.0000    0.2500    0 67.0000  48.0000    1.0000    0.1250
  2.5000    0.2500    0 69.0000  67.0000    1.2500    0.1250
  3.0000    0.2500    0 71.0000  52.0000    1.5000    0.1250
  3.5000    0.2500    0 72.0000  70.0000    1.7500    0.1250

```

Şekil 1- *nmat*'deki *readmidi* fonksiyonu çıktısı

Şekil 1'de sunulan *nmat* sekiz satır, yedi sütundan oluşmaktadır. Matris içinde, *nmat*'e çevrilen MIDI dosyasının içerdiği perde sayısı kadar satır bulunmaktadır. Dolayısıyla Şekil 1'deki *nmat*'in 8 perde içerdiği tespit edilmektedir. İlk sütunda verilen değerler perdelerin ölçü içindeki başlangıçlarını temsil etmektedir. Bu temsil yapılırken onaltılık, yani çeyrek nota ölçü birimi olarak kullanılmaktadır. İkinci sütundaki değerler perdelerin uzunluğunu belirtmektedir. İkinci sütundaki temsilde de çeyrek nota ölçü birimi olarak kullanılmaktadır. Üçüncü sütun MIDI kanalı bilgilerini içerirken, dördüncü sütun perde isimlerinin, MIDI protokolündeki numara karşılığını vermektedir⁶. Beşinci sütunda bulunan değerler vuruş duyarlılığı hakkında bilgi vermektedir. Altıncı sütun notaların başlangıcını, yedinci sütun ise her nota vuruşunun süresini ifade etmektedir. Son iki sütun ölçü birimi olarak saniyeyi kullanmaktadır. Burada dikkat edilmesi gereken nokta, ilk iki sütun ile son iki sütunun aynı özellikleri ifade etmesine rağmen, ilk iki sütunun ölçü olarak çeyrek notayı kullanırken son iki sütunun saniyeyi kullanıyor olmasıdır. MIDI verilerini temsil eden *nmat*'lerin içerdiği sütunların özellikleri Tablo 1'de sunulmuştur.

Tablo 1- *nmat* içindeki sütunların ifade ettiği değerler

Başlangıç (Ölçü)	Uzunluk (Ölçü)	MIDI Kanalı	MIDI Notası	Vuruş Duyarlılığı	Başlangıç (Saniye)	Uzunluk (Saniye)
---------------------	-------------------	----------------	----------------	----------------------	-----------------------	---------------------

⁶ MIDI protokolünde perde isimleri, 0-127 arası değerlerle ifade edilmektedir. Örneğin C4 perdesinin karşılığı 60'dır. Ayrıntılı bilgi için Bkz: (Durmaz, 2000:205)

1.2 MTB Genel Özellikler

MTB tonalite analizi, iki ezgi arasındaki melodik benzerliğin ölçümü, ezgi içindeki aralıkların tespit edilmesi, nota dağılımları gibi müzik analizinde ihtiyaç duyulan birçok işlemi otomatik olarak bilgisayar ortamında gerçekleştirmeye olanak sağlamaktadır. Bunun yanında, filtre fonksiyonlarıyla MIDI dosyaları üzerinde çeşitli değişiklikler yaparak el ile düzenleme gereksinimini minimuma indirmektedir.

MTB’de analiz işlemlerinin MIDI verileri üzerinden yapılıyor olması, analizi yapılacak olan müziklerin MIDI verilerini oluşturma gereksinimini ortaya çıkartmaktadır. Yapılan analizin sağlıklı olması için oluşturulan MIDI karşılıklar, tartımsal ve notasyon olarak analizi yapılacak olan müziğe bire bir uyumlu olmak zorundadır. Aksi takdirde yapılan analiz işlemleri ve elde edilen sonuçlar gerçeği yansıtmayacaktır.

Bir müziğe birebir uyumlu MIDI verisi hazırlamanın en kesin yolu manuel olarak el ile hazırlamaktır. Bu işlem için Muse Score benzeri müzik notasyon yazılımları⁷ kullanılmaktadır. Notalar, müziğin orijinal notasından faydalanarak el ile yazılıma girilmekte ve oluşturulan notasyonun MIDI versiyonu yazılım dışına aktarılmaktadır. Bu yöntemde notalar el ile girildiğinden dolayı bire bir uyum sağlanabilmektedir.

Müziğin orijinal notasına sadık kalmak suretiyle manuel olarak MIDI versiyonu oluşturmak, elde edilen sonucun doğruluğu açısından sağlıklı bir yöntem olsa da ciddi bir emek ve zaman kaybı anlamına gelmektedir. Müziğin sadece ilk 4 ölçüsü gibi ufak bir bölümünün analiz edileceği durumlarda büyük bir sorun gibi gözükme de tamamını el ile girmek uzun zaman almaktadır. Özellikle birden fazla müziğin analizinin yapılacağı durumlarda bu sorun daha da büyümektedir.

Bazı durumlarda kullanıcılar, birçok müziğin MIDI verilerini oluşturarak dijital ortamda MIDI arşivi hazırlamaya ihtiyaç duymaktadır. Müzik analizinde kullanmak için

⁷ Günümüzde müzik notasyon yazılımlarıyla bilgisayar ortamında nota yazmak ve yazılan notanın .mid uzantılı dosya olarak çıktısını almak mümkündür. Yaygın olarak kullanılan müzik notasyon yazılımlardan bazıları Finale, Muse Score Noteflight, Notion, Sibelius’dur.

oluşturulan MIDI arşivlerini manuel olarak hazırlamak, analiz haricinde başlı başına bir iş haline gelmektedir.

Analizi yapılacak müziklerin MIDI versiyonlarını oluşturmak için bir başka yöntem de ses dosyalarını otomatik olarak MIDI'ye dönüştüren yazılımlardır. El ile hazırlamanın aksine çok kısa süre içerisinde dönüştürme işlemini yapan bu yazılımlar kullanıcılara zaman açısından tasarruf sağlamaktadır.

Ses dosyalarından MIDI'ye dönüştürme işlemine olanak sağlayan yazılımların kazandırdığı zaman elbette çok önemlidir. Ancak daha önemli olan nokta ise yapılan analiz işlemlerinin sağlıklı ve gerçekçi olmasıdır. Bu da daha önce bahsedildiği üzere oluşturulan MIDI versiyonunun müzikle birebir uyumlu olmasıyla doğru orantılıdır.

1.3 MTB Fonksiyonlar

MTB'nin sağladığı MatLab fonksiyonları, işlevlerine göre çeşitli kategorilere ayrılmıştır. MTB'nin içerdiği ana fonksiyonlar aşağıdaki bölümlerde tablolar yardımıyla tanıtılmıştır. Fonksiyonlar tablolar aracılığıyla tanıtılmakla beraber, bazı fonksiyonlar uygulama aşaması hakkında bilgi vermesi amacıyla detaylandırılarak örneklerle açıklanmıştır.

Fonksiyonlar, MatLab yazılımının komut penceresi (*command window*) olarak adlandırılan kısmına girilen kodlar aracılığıyla kullanılmaktadır. Fonksiyonların kullanılmasıyla elde edilen *nmat*'ler, komut penceresinin yanında bulunan çalışma alanında (*work space*) hazır hale gelmektedir.

1.3.1 Dönüştürme

Dönüştürme (*conversion*) fonksiyonları analizi yapılacak MIDI dosyalarını *nmat*'e dönüştürerek, MatLab yazılımı içine aktarmak ve aynı zamanda MatLab içindeki *nmat*'leri MIDI'ye dönüştürerek dışarı aktarmak için kullanılmaktadır. Bunun dışında Tablo 2'de görüldüğü gibi kodlar aracılığıyla girilen frekanslar ve MIDI nota numaraları arasında karşılıklı dönüştürme işlemi yapmaya olanak sağlamaktadır. Bu kategori altında bulunan fonksiyonlar ve görevleri Tablo 2'de sunulmuştur.

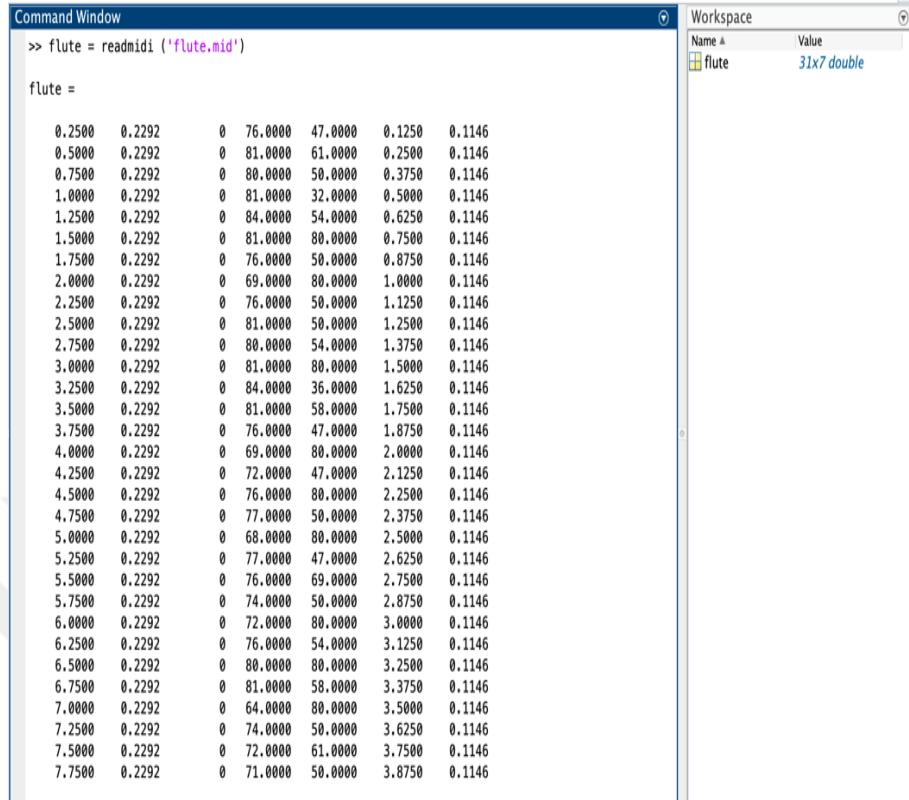
Tablo 2- Dönüştürme fonksiyonları ve görevleri

Fonksiyon	Görev
<i>dir2coll</i>	Belirtilen klasördeki MIDI dosyalarını <i>cellmatrix</i> ⁸ yapısına dönüştürür.
<i>hz2midi</i>	Hertz (Hz) cinsinden girilen frekans değerinin MIDI nota numara karşılığını verir.
<i>keyname</i>	Tonallığı harflerle ifade eder.
<i>midi2hz</i>	MIDI nota numaralarının Hz cinsinden frekans karşılığını verir.
<i>notename</i>	MIDI nota numaralarını harflere dönüştürür.
<i>readmidi</i>	MIDI dosyalarını <i>nmat</i> 'e dönüştürür.
<i>writemidi</i>	<i>nmat</i> 'leri MIDI dosyasına dönüştürür.

MTB'de analizi yapılacak MIDI dosyaları, daha önce belirtildiği gibi *nmat* yapısına dönüştürülerek temsil edilmektedir. MIDI dosyalarını *nmat* yapısına dönüştürmek için Tablo 2'de görülen *readmidi* fonksiyonu kullanılmaktadır. Analiz işlemlerinde ilk olarak analizi yapılacak MIDI dosyalarının MatLab içine aktarılması ve *nmat* yapısına çevrilmesi gerektiğinden bu fonksiyon analiz işlemlerinde önemli bir yere sahiptir.

MIDI dosyasının *nmat*'e dönüştürülmesi için “*flute = readmidi ('flute.mid')*” komutu, MatLab komut penceresine girilmiştir. Girilen komutun başında bulunan *flute*, oluşturulacak *nmat*'in ismini ifade etmekle beraber kullanıcının tercihiyle bağlı olarak değişiklik gösterebilmektedir. Komutun son kısmındaki parantez içinde, bilgisayarda kayıtlı bulunan MIDI dosyasının ismi belirtilmektedir. İşlem sonucunda oluşturulan *nmat*'in görünümü Şekil 2'de sunulmuştur.

⁸ *Cellmatrix*, geniş MIDI arşivleri üzerinde eş zamanlı işlem yapmaya olanak sağlayan, içinde birden fazla *nmat*'i barındırabilen MatLab hücresidir.



Şekil 2- `readmidi` fonksiyonu ile elde edilen *nmat*

Dönüşüm işlemi sonucunda oluşturulan *nmat*, komutun başına yazılan isimle adlandırılmış ve MatLab yazılımının çalışma alanında otomatik olarak hazır hale gelmiştir.

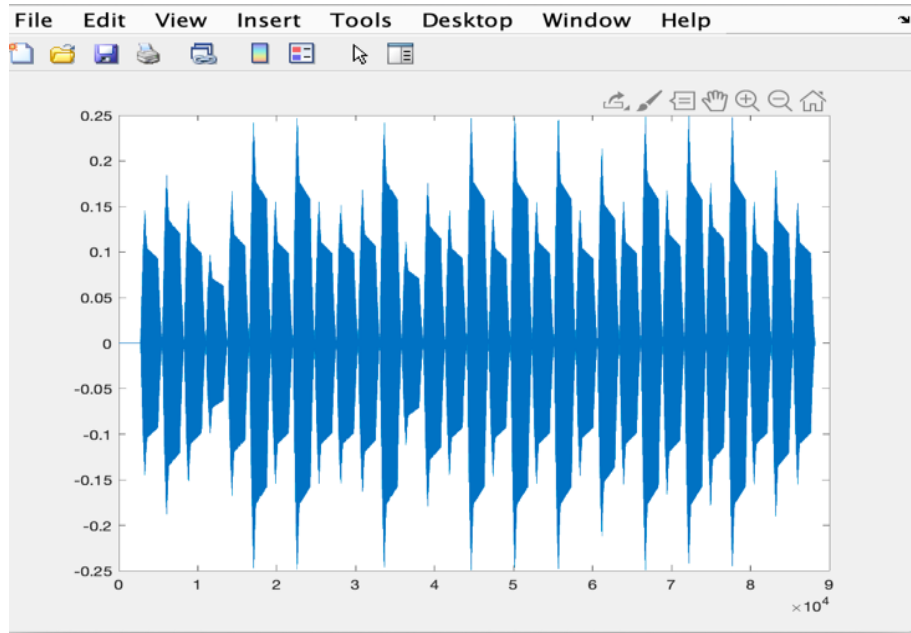
1.3.2 Üretim

Üretim (*generation*) fonksiyonlarıyla kodlar aracılığıyla girilen özelliklere göre *nmat*'ler oluşturulabilmektedir. Girilen kodlarla oluşturulan ya da MIDI dosyasından dönüştürülerek elde edilen *nmat*'ler, sentezlenerek dalga biçimleri oluşturulabilmekte ve sentezlenen *nmat*'ler harici bir MIDI oynatıcı tanımlanarak oynatılabilmektedir. Elde edilen dalga biçimleri istendiği takdirde çizim fonksiyonları kullanılarak görselleştirilmektedir. Bu kategori altında bulunan fonksiyonlar ve görevleri Tablo 3'de sunulmuştur.

Tablo 3- Üretim fonksiyonları ve görevleri

Fonksiyon	Görev
<i>createnmat</i>	Kod içinde girilen nota, süre ve vuruş duyarlılığı verilerine bağlı olarak yeni bir <i>nmat</i> oluşturur.
<i>setmidiplayer</i>	Harici MIDI oynatıcıyı tanımlar.
<i>nmat2snd</i>	<i>nmat</i> 'leri dalga biçimine dönüştürür.
<i>playmidi</i>	Harici MIDI oynatıcı kullanarak <i>nmat</i> 'leri oynatır.
<i>playsound</i>	<i>nmat</i> 'leri dalga biçimine çevirerek oynatır.
<i>reftune</i>	Bir melodiyi referans ya da örnek olarak alır.

MatLab içinde bulunan MIDI dosyasının dalga biçimini oluşturmak ve elde edilen sonucu görselleştirmek için ilk olarak “*a = readmidi('flute.mid')*” komutu kullanılarak *flute* isimli MIDI dosyası *nmat*'e dönüştürülmüş ve *a* harfiyle adlandırılmıştır. Elde edilen *nmat*'in dalga biçimi “*wf = nmat2snd(a)*” komutu ile oluşturulmuştur. Daha sonra “*plot(wf)*” komutu girilerek elde edilen ve *wf* olarak adlandırılan dalga biçimi görselleştirilmiştir. Yapılan işlemler sonucunda elde edilen görsel Şekil 3'de sunulmuştur.

**Şekil 3-** *nmat*'den elde edilen dalga biçimi

1.3.3 Filtre

Filtre (*filter*) fonksiyonları, MatLab içinde bulunan *nmat*'ler üzerinde çeşitli değişiklikler yapmak üzere kullanılmaktadır. Örneğin, *nmat* içindeki istenilen MIDI kanal üzerinde bulunan tüm notalar, *dropmidich* fonksiyonu kullanarak silinebilmektedir. Bu fonksiyonların uygulamasında, kullanılan komutun başına kullanıcının tercihinine bağlı olarak bir isim girilmektedir. MatLab, filtre fonksiyonları aracılığıyla üzerinde işlem yapılan *nmat*'in orijinal halini koruyarak komutun başına girilen isimle adlandırılmış yeni bir *nmat* oluşturmaktadır. İşlem sonucunda oluşturulan yeni *nmat* çalışma alanında hazır hale gelmektedir.

Analiz işlemleri esnasında *nmat*'in bütünü yerine sadece belirli bölgeler üzerinde işlem yapmak istendiğinde, filtre fonksiyonları kullanılarak istenilen bölümü içeren yeni *nmat*'ler oluşturulabilmektedir. Bu kategoriye ait fonksiyonlar ve işlevleri Tablo 4'de sunulmuştur.

Tablo 4- Filtre fonksiyonları ve görevleri

Fonksiyon	Görev
<i>dropmidich</i>	Belirlenen MIDI kanalındaki notaları siler.
<i>dropshortnotes</i>	Belirlenen eşikten kısa olan notaları siler.
<i>elim</i>	Kısa MIDI kanallarını ortadan kaldırır.
<i>extreme</i>	Çoksesli <i>nmat</i> 'lerin üst veya alt perdelerini filtreler
<i>getmidich</i>	Girilen MIDI kanalındaki notalardan oluşan yeni bir <i>nmat</i> oluşturur.
<i>smonophonic</i>	<i>nmat</i> 'in tek sesli olup olmadığını kontrol eder.
<i>mchannels</i>	<i>nmat</i> içinde kullanılan MIDI kanallarını gösterir.
<i>movewindow</i>	Kullanılacak fonksiyonu tanımlı bir zaman aralığında çalıştırır.
<i>onsetwindow</i>	Ölçü ya da saniye cinsinden girilen minimum ve maksimum değer arasında kalan notalar ile yeni bir <i>nmat</i> oluşturur.
<i>perchannel</i>	Bir fonksiyonun, <i>nmat</i> içinde bulunan her midi kanalının üzerinde ayrı ayrı uygulanmasını sağlar.
<i>quantize</i>	Notaları girilen değerlere göre ölçeklendirir.

<i>scale</i>	<i>nmat</i> içinde istenilen herhangi bir değeri ölçeklendirir
<i>setvalues</i>	<i>nmat</i> içindeki vuruş duyarlılığı, nota süresi gibi parametrelerden belirleneni istenilen değere çeker.
<i>shift</i>	Notaların başlangıç zamanı, süresi veya perde yüksekliği üzerinde istenilen değerde değişiklik yapar.
<i>transpose2c</i>	Belirlenen <i>nmat</i> 'i C majör ya da minöre transpoze eder.
<i>trim</i>	<i>nmat</i> 'in başlangıcında bulunan boşluğu siler.

1.3.4 Meta

MatLab'de yapılan analiz işlemleri, *nmat* yapıları üzerinde yapılmaktadır. Ancak bazı durumlarda birden fazla MIDI dosyası üzerinde eş zamanlı işlem yapma ihtiyacı ortaya çıkmaktadır. MatLab içinde birden fazla *nmat* veya MIDI dosyası grup haline getirilebilmekte ve bu gruplar *nmat* koleksiyonu olarak adlandırılmaktadır. Oluşturulan *nmat* koleksiyonu üzerinde toplu analiz yapabilmek için meta fonksiyonları kullanılmaktadır. Meta fonksiyonları ve görevleri Tablo 5'de sunulmuştur.

Tablo 5- Meta fonksiyonları ve görevleri

Fonksiyon	Görev
<i>analyzecoll</i>	Fonksiyonları koleksiyon içinde bulunan bütün <i>nmat</i> 'lere uygular.
<i>analyzedir</i>	Hedefteki bütün MIDI dosyalarını analiz eder.
<i>filtercoll</i>	Girilen filtre fonksiyonuna göre, koleksiyon içinde bulunan tüm <i>nmat</i> 'leri filtreler.

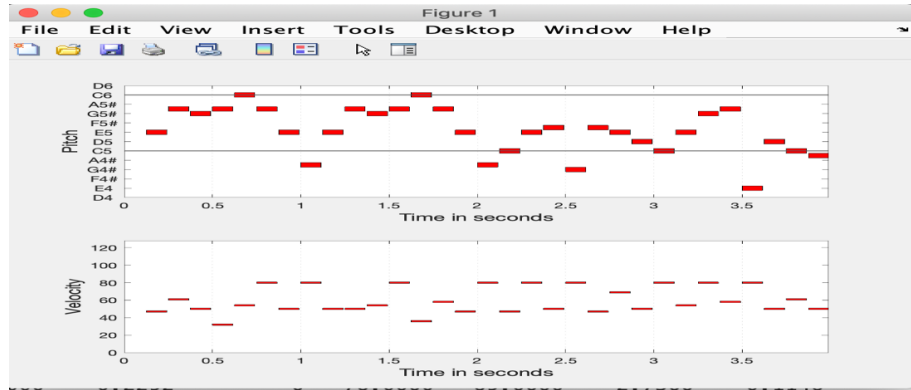
1.3.5 Çizim

Çizim (*plotting*) fonksiyonları, MIDI verileri üzerinde yapılan analizleri görselleştirme işlemleri için kullanılan fonksiyonlardan oluşmaktadır. Bu fonksiyonlar genellikle MTB'nin sağladığı diğer fonksiyonlarla beraber kullanılmaktadır. Örneğin, bir sonraki bölümde incelenecek olan istatistiksel fonksiyonlarla istenilen özellikler hakkında orantısal hesaplamalar yapılarak çizim fonksiyonlarıyla elde edilen sonuçlar görselleştirilmektedir. Bu kategorideki fonksiyonlar ve görevleri Tablo 6'da sunulmuştur.

Tablo 6- Çizim fonksiyonları ve görevleri

Fonksiyon	Görev
<i>pianoroll</i>	Analizi yapılan <i>nmat</i> 'i istenilen bilgiler doğrultusunda <i>piano-roll</i> notasyonu ⁹ olarak görselleştirir.
<i>plotdist</i>	Perde yüksekliği, aralık veya süre dağılımlarını hesaplayan fonksiyonların elde ettiği sonuçlara göre grafik oluşturur.
<i>plothierarchy</i>	<i>nmat</i> 'in tonal yapısını dikey olarak görselleştirir.
<i>plotmelcontour</i>	<i>nmat</i> 'in melodik konturunu çizer.

Daha önce MIDI dosyasından dönüştürülerek *flute* olarak adlandırılan *nmat* üzerinde *pianoroll* fonksiyonu kullanılmıştır. Komut penceresine “*pianoroll (flute,'sec','vel')*” komutu girilerek ilgili *nmat*'in *piano-roll* notasyonunun çizimi gerçekleştirilmiştir. Girilen komutta parantez içinde bulunan ilk komut, analizi yapılacak *nmat*'in ismidir. Yanında bulunan diğer iki komut ise kullanıcının tercihinin bağlı olarak değişkenlik gösterebilmekte ve analizi yapılmak istenen özellikleri ifade etmektedir. Yapılan işlem aracılığı ile elde edilen sonuç Şekil 4' de sunulmuştur.

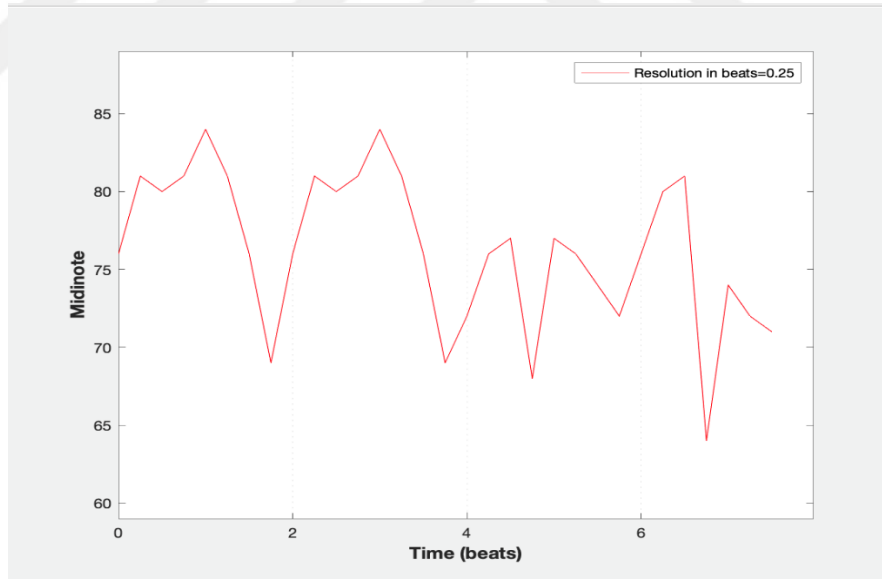
**Şekil 4-** *Pianoroll* fonksiyonu kullanılarak elde edilen görsel

Şekil 4'te sunulan sonuçta, y ekseninde nota isimleri ve vuruş duyarlılığı değerleri gösterilirken, x ekseninde saniye cinsinden zaman bilgisi verilmektedir.

⁹ *Piano-roll* notasyonun tarihi 20. Yüzyılın başlarına dayanmaktadır. İlk başlarda özel üretilen piyanolarda, müziğin icrası esnasında perde yükseklikleri ve perde süreleri boş bir kâğıt ruloya delinerek kaydedilmekteydi (Muller, 2015). Günümüzde *piano-roll* temsili, 2 boyutlu geometrik bir görselleştirme olarak kullanılmaktadır.

Bir sonraki örnekte, komut penceresine girilen “*plotmelcontour (flute,0.25,'abs','r')*” komutu ile daha önce oluşturulan *flute* adlı *nmat*’in melodik kontur analizi yapılmış ve analiz sonucu elde edilen veriler görsel olarak oluşturulmuştur. Girilen komutta parantez içinde yer alan *flute*, ilgili *nmat*’in ismini ifade etmektedir. Melodik kontur hesaplamaları kullanıcı tarafından belirlenen çözünürlük değerine göre yapılmakta ve bu değer 0 ile 1 arasında belirlenebilmektedir. Belirlenen çözünürlük değeri büyüdükçe detay azalmaktadır. Girilen komutta, *nmat* isminin yanında bulunan 0.25, yapılan örnekteki çözünürlük derecesini ifade etmektedir. Çözünürlük derecesi saniye veya ölçü içindeki vuruşlara göre düzenlenebilmektedir. Analizin çözünürlük derecesinin, ölçü vuruşlarına göre gösterilmesi için ‘*abs*’ komutu eklenmiştir. Girilen komutun en sonunda bulunan ‘*r*’ ise grafik çizgisinin renk kodu olup kırmızıyı temsil etmektedir.

Yapılan işlemle elde edilen sonuç Şekil 5’de sunulmuştur



Şekil 5- *Plotmelcontour* fonksiyonu ile elde edilen görsel.

Yukarıdaki örnekte 0.25 çözünürlük değerine göre analiz yapılmış ve melodinin konturu çizilmiştir. Ancak yapılan analizin ihtiyaçları doğrultusunda aynı anda farklı çözünürlük derecelerinde analiz yapılabilen ve sonuçlar tek grafik üzerinde görselleştirilebilmektedir. Renk kodları sayesinde, çözünürlük derecelerinin renk seçimi yapılarak sonuçlar birbirinden ayırt edilebilmektedir.

Analizi yapılacak MIDI verilerinin ezgisel çizgisini, grafiksel olarak analiz etmeye yarayan bu fonksiyon, müzik analizinde özel öneme sahip fonksiyonlardan biridir.

1.3.6 İstatistiksel

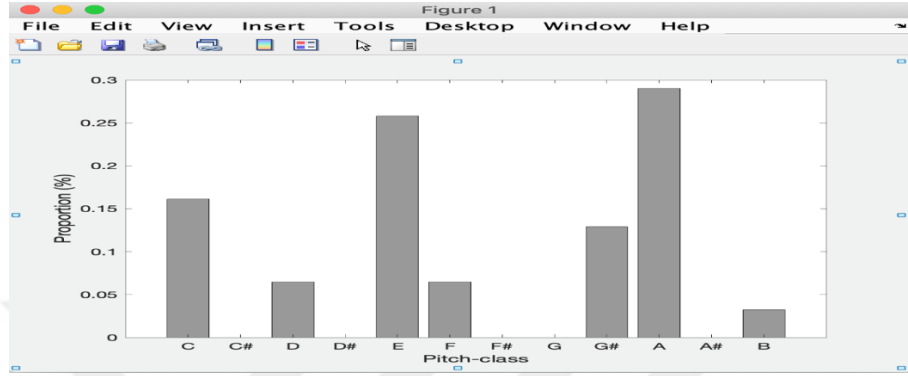
İstatistiksel fonksiyonlar, yapılan analiz işlemleri esnasında *nmat*'lerin içindeki istenilen bilgilerin dağılımını istatistiksel olarak hesaplamaktadır. İstatistiksel fonksiyonlar kullanılarak yapılan hesaplamalar sonucunda elde edilen verilerin görselleştirilebilmesi, çizim fonksiyonlarıyla beraber kullanılmasıyla mümkün hale gelmektedir.

Tablo 7- İstatistiksel fonksiyonlar ve görevleri

Fonksiyon	Görev
<i>durdist1</i>	Nota sürelerinin dağılımını verir.
<i>durdist2</i>	Notalar arasındaki geçiş sürelerinin dağılımını verir.
<i>entropy</i>	Girdi argümanı olarak verilen bir dağılımın entropisini verir.
<i>ivdirdist1</i>	Aralıkların yönsel (inici veya çıkıcı) olarak dağılımını verir.
<i>ivdist1</i>	Aralıkların zamana göre dağılımını verir.
<i>ivdist2</i>	Aralıkların nota sürelerine göre dağılımını verir.
<i>ivsizedist1</i>	Aralık boyutlarının dağılımını verir.
<i>nnotes</i>	<i>nmat</i> içindeki nota sayısını verir.
<i>pcdist1</i>	Bir aşıt içindeki 12 perdenin süre/sayı olarak kaç defa geldiğini gösterir.
<i>pcdist2</i>	Bir aşıt içinde notalar arasındaki geçişleri süre/sayı olarak gösterir.
<i>refstat</i>	Elde edilen bir dağılımı, referans olarak seçilen başka bir dağılım ile karşılaştırmak için kullanılır.

Aşağıdaki örnekte, daha önce oluşturulan *flute* isimli *nmat* üzerinde *pcdist1* fonksiyonu kullanılmıştır. Bu fonksiyon *nmat* içindeki notaların dağılımını belirtmektedir. Yapılacak işlem için MatLab'de komut penceresine “*plotdist (pcdist1(flute));*” komutu girilmiştir. Bu komutta parantez içinde bulunan iç fonksiyon

pcdist1 nota dağılım oranını hesaplarken, parantez dışında bulunan *plotdist* komutuysa hesaplanan oranların grafiğini oluşturmaktadır. Yapılan işlem sonucunda ortaya çıkan sonuç Şekil 6’da sunulmuştur.



Şekil 6- Nota dağılımının hesaplanmasıyla elde edilen görsel

Elde edilen sonuç incelendiğinde, analizi yapılan *nmat*’in içinde sırasıyla en çok A, E ve C notalarının bulunduğu görülmektedir. Bu veriler doğrultusunda melodinin tonu hakkında öngörü yapılabilmektedir.

1.3.7 Anahtar Bulma

MTB’de ‘anahtar’ kelimesi tonalite ile eş anlamlı olarak kullanılmaktadır. Anahtar bulma (*key finding*) fonksiyonları, tonalite üzerinde analiz yapmak için tasarlanan fonksiyonlardan oluşmaktadır. Bu fonksiyonlarla ezginin geneli üzerinde tonaliteyle alakalı analiz yapılabildiği gibi zaman içindeki tonalite değişimleri de analiz edilebilmektedir. MTB bu analiz işlemlerini klasik Krumhansl ve Schmuckler tonallık bulma algoritmasını¹⁰ kullanarak yapmaktadır.

¹⁰ Krumhansl & Schmuckler tonallık bulma algoritması (Krumhansl&Schmuckler, 1990), Krumhansl ve Kessler’in daha önceki bilişsel çalışmalarından elde edilen tonalite profillerine dayanmaktadır (Krumhansl& Kessler, 1982). Bu profiller bugün batı müziğinde kullanılan 12 majör ve 12 minör tonallığa karşılık gelir.

Tablo 8- Anahtar bulma fonksiyonları ve görevleri

Fonksiyon	Görev
<i>keymode</i>	Tonallığın majör mü yoksa minör mü olduğunu analiz eder.
<i>keysom</i>	Nota dağılımının haritasını oluşturur.
<i>keysomanim</i>	Nota dağılımının animasyonunu oluşturur.
<i>kkcc</i>	Krumhansl-Kessler profillerine göre korelasyonu gösterir.
<i>kkkey</i>	Tonallığı analiz ederek rakamlarla ifade eder. ¹¹
<i>maxkkcc</i>	Krumhansl-Kessler profillerine göre en yüksek korelasyonu gösterir.
<i>tonality</i>	Ezgi içindeki notalara genel tonaliteye göre kararlılık değeri verir.

Analizi yapılacak olan müziğin majör ya da minör tonallığa sahip olduğunu öğrenmek için *keymode* fonksiyonu kullanılmaktadır. Bu fonksiyonu kullanabilmek için öncelikle komut penceresine “*a = readmidi('flute.mid')*” komutu girilerek, *flute* isimli MIDI dosyası *nmat* yapısına dönüştürülmüş ve *a* harfi ile adlandırılmıştır. Oluşturulan *nmat* üzerinde *keymode* fonksiyonunun kullanımı ve kullanımla elde edilen sonuç Şekil 7’de sunulmuştur.

```
>> keymode(a)
```

```
ans =
```

```
1
```

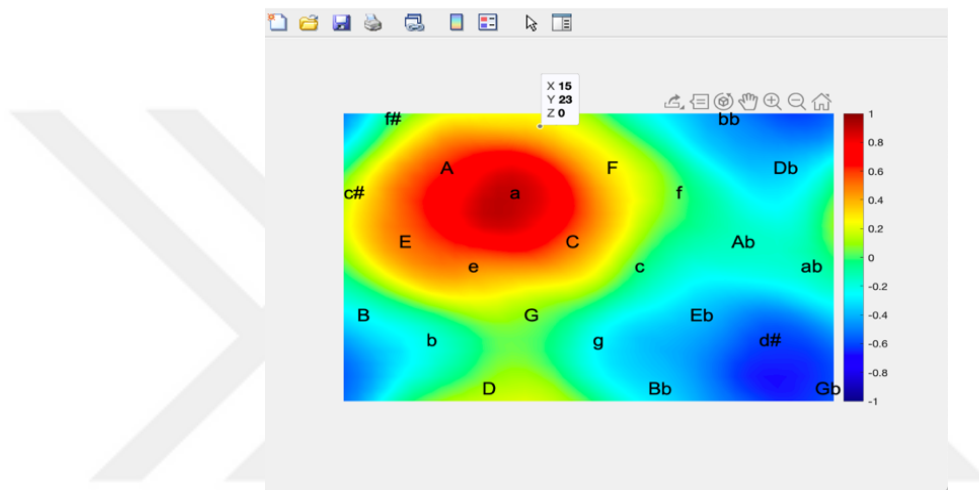
Şekil 7- *keymode* fonksiyonu kullanımı

Keymode fonksiyonuyla yapılan işlemlerin sonucunda elde edilen rakam, 1 ya da 2 olmaktadır. 1 rakamı majör tonallığı ifade ederken 2 rakamı ise minör tonallığı ifade etmektedir. Yapılan örnekte işlem sonucunun 1 olduğu göz önünde bulundurulduğunda, *a* harfi ile adlandırılan *nmat*’in tonallığının majör olduğu anlaşılmaktadır. *Tonality*

¹¹ *Kkkey* fonksiyonuyla yapılan analiz işlemlerinde sonuçlar, 1 ile 24 arasında çıkmaktadır. 1 ile 12 arasındaki rakamlar majör, 13 ile 24 arasındakiler ise minör tonaliteleri ifade etmektedir. Rakamlar ve temsil ettiği tonaliteler Tablo 9’da sunulmuştur.

fonksiyonunu kullanabilmek için öncelikle *keymode* fonksiyonu ile analiz yapılması gerekmektedir.

Nota dağılım haritasını oluşturmak için Tablo 8’de görüldüğü üzere *keysom* fonksiyonu kullanılmaktadır. A harfiyle adlandırılmış olan *nmat*’in, nota dağılım haritasını oluşturmak için komut penceresine “*keysom(a)*” komutu girilmiştir. Elde edilen sonuç aşağıda Şekil 8’ de sunulmuştur.



Şekil 8- Nota dağılım haritası

Anahtar bulma fonksiyonlarına son örnek olarak, aynı *nmat*'in hangi tonallığa ait olduğu incelenmiştir. Bu işlem *kkkey* fonksiyonu kullanılarak yapılmıştır. Yapılan işlem Şekil 9'da sunulmuştur.

```
>> kkkey(a)
```

ans =

22

Şekil 9- *kkkey* fonksiyonu kullanımı.

Tablo 9- Tonalitelerin rakam olarak karşılıkları

Majör Tonaliteler		Minör Tonaliteler	
1	C	13	c
2	C #	14	c #
3	D	15	d
4	D #	16	d #
5	E	17	e
6	F	18	f
7	F #	19	f #
8	G	20	g
9	G #	21	g #
10	A	22	a
11	A #	23	a #
12	B	24	b

MTB’de anahtar bulma kategorisinin içinde yer alan *kkkey* fonksiyonu, müziğin tonalitesini rakamlarla ifade etmektedir. Bu fonksiyonun ne kadar doğru ve verimli çalıştığını saptamak için bu aşamada bir uygulama yapmakta fayda var.

MTB’de yapılacak örnek uygulama için klasik döneme ait 4 yaratı seçilmiş ve *kkkey* fonksiyonuyla yaratıların tonalite analizi yapılmıştır. Analiz işlemlerinde kullanılacak yaratılar aşağıda Tablo 10’da sunulmuştur.

Tablo 10- *Kkkey* fonksiyonu uygulanacak yaratılar

Besteci	Yaratı	Dosya ismi	nmat ismi
W. A. Mozart	Rondo Alla Turco	mz.mid	A
W. A. Mozart	Piano Sonata No.9	mopsn9.mid	B
W. A. Mozart	Piano Sonata No.1	mopsn1a.mid	C
L. V. Beethoven	Piano Sonata No.16	bepsn16.mid	D

İlk olarak, Mozart'ın Rondo Alla Turco (A majör) adlı yaratısına ait MIDI dosyası, MatLab içinde *nmat* yapısına dönüştürülmüştür. Bu işlem için “*a = readmidi('mz.mid')*” komutu girilmiş ve *a* harfiyle adlandırılan yeni bir *nmat* oluşturulmuştur. Elde edilen *nmat* üzerinde *kkkey* fonksiyonunun kullanımı ve fonksiyonun kullanılmasıyla elde edilen sonuç şöyledir:

```
>> kkkey(a)
ans =
    10
```

Sonuçta yer alan 10 rakamının A majöre denk geldiği göz önünde bulundurulduğunda, MTB'nin tonaliteyi doğru tahmin ettiği görülmektedir.

İkinci sırada yer alan, Mozart'ın 9 numaralı Piyano Sonatının (D majör) MIDI versiyonu; “*b = readmidi('mopsn9.mid')*” komutuyla *nmat* yapısına dönüştürülmüştür. Bu komut ile *b* harfiyle adlandırılmış yeni bir *nmat* oluşturulmuştur. Elde edilen *nmat* üzerinde *kkkey* fonksiyonunun kullanımı ve fonksiyonun kullanılmasıyla elde edilen sonuç şudur:

```
>> kkkey(b)
ans =
     3
```

Yapılan işlem sonucu elde edilen 3 rakamının D majöre denk geldiği göz önünde bulundurulduğunda, MTB'nin tonaliteyi doğru tespit ettiği görülmektedir.

Üçüncü olarak yine Mozart'a ait 1 numaralı piyano sonatının (C majör), Allegro bölümü kullanılmıştır. Yaratıya ait MIDI versiyonu, “*c = readmidi('mopsn1a.mid')*” komutu kullanarak *nmat* yapısına dönüştürülmüştür. Girilen komut aracılığıyla *c* harfiyle adlandırılmış yeni bir *nmat* oluşturulmuştur. Elde edilen *nmat* üzerinde *kkkey* fonksiyonunun kullanımı ve fonksiyonun kullanılmasıyla elde edilen sonuç:

```
>> kkkey(c)

ans =

    1
```

Yapılan işlem sonucu elde edilen 1 rakamının C majöre denk geldiği göz önünde bulundurulduğunda, MTB'nin tonaliteyi doğru tahmin ettiği görülmektedir.

Son olarak, Beethoven'ın 16 numaralı Piyoano Sonatının (G majör) MIDI versiyonu, “*d = readmidi('bepsn16.mid')*” komutu kullanılarak *nmat* yapısına dönüştürülmüştür. Girilen komutla beraber d harfiyle adlandırılmış yeni bir *nmat* oluşturulmuştur. Elde edilen *nmat* üzerinde *kkkey* fonksiyonunun kullanımı ve fonksiyonun kullanılmasıyla elde edilen sonuç aşağıdaki gibidir:

```
>> kkkey(d)

ans =

    8
```

Yapılan işlem sonucu elde edilen 8 rakamının G majöre denk geldiği göz önünde bulundurulduğunda, MTB'nin tonaliteyi doğru tahmin ettiği görülmektedir.

4 ayrı müzik üzerinde yapılan işlemler ve elde edilen sonuçlar Tablo11'de sunulmuştur.

Tablo 11- Kkkey fonksiyonu sonuçları

Besteci	Yaratı	Fonksiyon	Sonuç
W. A. Mozart	Rondo Alla Turco	kkkey	10 (A)
W. A. Mozart	Piano Sonata No.9	kkkey	3 (D)
W. A. Mozart	Piano Sonata No.1	kkkey	1 (C)
L. V. Beethoven	Piano Sonata No.16	kkkey	8 (G)

1.3.8 Kontur

Kontur (*contour*) fonksiyonları, MatLab içinde oluşturulan *nmat*'lerin içerdiği ezgilerin melodik konturu üzerinde analiz yapmak için tasarlanmış fonksiyonlardan oluşmaktadır. Bu kategorideki fonksiyonlar ve görevleri Tablo 12'de sunulmuştur.

Tablo 12- Kontur fonksiyonları ve görevleri

Fonksiyon	Görev
<i>melcontour</i>	İncelenen <i>nmat</i> 'in melodik konturunu hesaplayarak yeni bir <i>nmat</i> oluşturur
<i>combcontour</i>	Melodik konturun matris temsilini oluşturur.

1.3.9 Melodik

Analizi yapılan ezginin karmaşıklığı, özgünlüğü gibi özelliklerini analiz ederek ölçen ve sonuçları rakamlarla ifade eden fonksiyonlardan oluşmaktadır. Bu ölçümler, çeşitli araştırmacıların ilgili analiz konusu üzerinde yaptığı araştırmalar sonucunda elde ettiği çıkarımlara dayandırılarak yapılmaktadır. Bahsedilen fonksiyonların yanı sıra iki ezgi arasındaki benzerliği ölçen ve müzik analizinde özel öneme sahip olan *meldistance* fonksiyonu da bu kategoride yer almaktadır. Kategoriyeye ait bütün fonksiyonlar Tablo 13'de sunulmuştur.

Tablo 13- Melodik fonksiyonlar ve görevleri

Fonksiyon	Görev
<i>ambitus</i>	Ezginin aralık genişliğini gösterir.
<i>complebm</i>	Ezginin karmaşıklığını gösterir.
<i>compltrans</i>	Ezginin özgünlüğünü gösterir.
<i>gradus</i>	Ezginin melodikliğini gösterir.
<i>melaccent</i>	Ezginin içindeki melodik aksanları analiz eder.
<i>melattraction</i>	Melodik çekiciliği gösterir.
<i>meteraccent</i>	Vurgu senkronizasyonunu ölçer.
<i>mobility</i>	Ezginin içindeki melodik yön değişimlerini analiz eder.

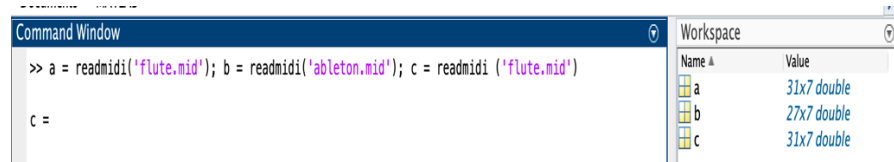
<i>narmour</i>	Ezginin melodik beklentisini tahmin eder.
<i>meldistance</i>	İki ezgi arasındaki benzerliği ölçer.

Melodik fonksiyonlar kategorisinde bulunan *meldistance* fonksiyonu, müzik analizinde özel öneme sahip fonksiyonlardan biridir. Bu fonksiyon aracılığıyla, analizi yapılacak iki ezginin arasındaki benzerlik ölçümü yapılabilmektedir. *meldistance* komutu ile yapılan benzerlik ölçümü sonucunda 0 ile bir arasında bir değer çıkmaktadır. Çıkan değer ne kadar ufaksa karşılaştırılan ezgiler birbirine o kadar benzerdir.

Meldistance fonksiyonuna ilişkin yapılan örnekte, ezginin orijinal MIDI dosyası ile aynı ezgiye ait ses dosyasının, Ableton Live yazılımında MIDI'ye dönüştürülmesi sonucu elde edilen MIDI dosyası kullanılmıştır.

İlk olarak komut penceresine “*a = readmidi ('flute.mid')*” komutu girilerek orijinal MIDI dosyası *nmat*'e dönüştürülmüştür. Ardından “*b = readmidi('ableton.mid')*” komutuyla Ableton Live yazılımında oluşturulan MIDI dosyası, *nmat*'e dönüştürülmüştür. Son olarak, “*c = readmidi('flute.mid')*” komutu girilerek orijinal MIDI dosyası farklı bir isimle adlandırılarak üçüncü bir *nmat* oluşturulmuştur. Üçüncü *nmat*'in oluşturulmasındaki neden, *meldistance* fonksiyonunun birebir aynı olan iki ezgi üzerinde uygulanmasıyla elde edilecek sonucu görmektir.

Komut penceresine girilen komutlar ve sonucunda çalışma alanında hazır hale gelen *nmat*'lerin görünümü aşağıda Şekil 10'da sunulmuştur.



Şekil 10- *meldistance* fonksiyonunu kullanmak için *nmat* yapısına dönüştürme işlemi

MIDI dosyalarından oluşturulan a ve b *nmat*'leri üzerinde *meldistance* fonksiyonunun kullanımı Şekil 11'de sunulmuştur.

```
>> meldistance(a,b,'pcdist1','taxi')
ans =
    0.4491
```

Şekil 11- *meldistance* fonksiyonunun kullanımı

Girilen kodda bulunan a ve b karşılaştırılacak *nmat*'lerin isimleridir. Diğer iç fonksiyonlardan *pcdist1*, iki *nmat*'in karşılaştırmasının nota dağılımlarına göre yapılmasını belirlerken, diğer iç fonksiyon olan *taxi* ise karşılaştırmada kullanılacak mesafe ölçümü ile ilgilidir. Çıkan sonucun 0.4491 olduğu görülmektedir. Sonuçların 0 ile 1 arasında olduğu ve değer küçüldükçe benzerliğin arttığı göz önünde bulundurulduğunda %50' den daha fazla benzerlik olduğu söylenebilmektedir.

Son olarak aynı MIDI dosyalarından farklı isimlerle oluşturulmuş olan iki *nmat*'in benzerlik ölçümü yapılmıştır. İşlem aşaması ve sonucu Şekil 12'de sunulmuştur.

```
>> meldistance(a,c,'pcdist1','taxi')
ans =
    0
```

Şekil 12- Aynı ezgiler üzerinde *meldistance* fonksiyonunun kullanımı

İki aynı ezginin benzerlik ölçümü yapıldığında Şekil 12'de görüldüğü gibi sonuç 0 olmaktadır.

1.3.10 Bölümleme

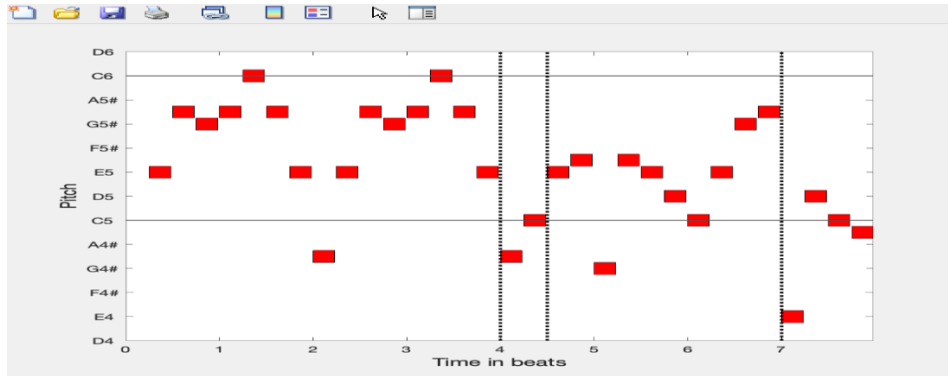
Ezginin bütünü üzerinde yapılan analiz işlemlerinin yanında melodik cümleler, motifler vb. belirlenerek daha küçük bölümler üzerinde de analiz işlemleri yapılmaktadır. Bölümleme (*segmentation*) fonksiyonları, analiz işlemi için ezginin bölümlere ayrılacağı noktaları saptayan fonksiyonlardan oluşmaktadır. MTB'de analizi yapılan ezginin bölümlenmesi iki ayrı algoritmaya dayandırılarak yapılmaktadır.

Bunlardan ilki bölümlleme işlemini Tenney ve Polansky'nin (T&P) bölümlleme algoritmasına¹² göre gerçekleştirmektedir. Diğer teknik ise, bir ezginin analizi yapılarak elde edilen çeşitli olasılıklara göre bölümlleme işlemini yapmaktadır. Bu teknikte bölümlleme noktalarını belirleyen olasılıklar, Essen halk müziği koleksiyonundaki nota, aralık ve nota sürelerinin dağılımlarından türetilmiştir (Eerola, Toiviainen, 2004: 32).

Tablo 14- Bölümlleme fonksiyonları ve görevleri

Fonksiyon	Görev
<i>segmentgestalt</i>	Ezgiyi T&P bölümlleme algoritmasına göre bölümlere ayırır.
<i>segmentprob</i>	Ezgiyi Essen koleksiyonundan türetilen olasılıklara göre bölümlere ayırır.
<i>boundary</i>	Bölgesel sınır tespit modeli.

segmentgestalt fonksiyonunu kullanarak ezginin hangi noktalardan bölüneceğini tespit etmek için ilk olarak, “*a = readmidi('flute.mid')*” komutu girilerek *nmat* oluşturulmuştur. Oluşturan *nmat* üzerinde fonksiyonu kullanmak için komut penceresine “*segmentgestalt(a,'fig')*” komutu girilmiştir. Elde edilen sonuç Şekil 13’de sunulmuştur. Bu komutta bulunan iç fonksiyon ‘*fig*’ sonucun görselleştirilmesi ile alakalıdır.



Şekil 13- *segmentgestalt* fonksiyonu ile elde edilen ve bölümlleme noktalarını gösteren görsel

¹² T&P bölümlleme algoritması (Tenney&Polansky, 1980), ezginin içindeki değişim noktalarını tespit etmektedir. Değişim noktaları büyük aralık başlangıçlarına denk gelmektedir.

Çıkan görselde, yatay düzlemdeki rakamlar ölçüleri ifade ederken dikey düzlemdeki harfler ise nota isimlerini ifade etmektedir. Kesik dikey çizgiler ise tespit edilen bölme noktalarını ifade etmektedir.

Daha önce bahsedildiği üzere MTB’de müzik analizi işlemleri MIDI verileri üzerinden yapılmaktadır. Dolayısıyla analizi yapılan müziklerin MIDI verilerini oluşturmak MTB kullanımı için temel bir ihtiyaçtır. Çalışmanın ilerleyen kısımlarında sesten MIDI’ye dönüşüm yapan yazılımlar incelenerek dönüştürme işlemini hangi yazılımların daha başarılı yaptığı araştırılmıştır ve MTB kullanımı için ihtiyaç duyulan MIDI versiyonunu oluşturmak için yeterli bir yazılım olup olmadığı sorusuna cevap aranmıştır.

2. BÖLÜM:

AUDIO-to-MIDI YAZILIMLARIN BAŞARI KARŞILAŞTIRMASI

2. BÖLÜM: AUDIO-to-MIDI YAZILIMLARIN BAŞARI KARŞILAŞTIRMASI

Müzik analizinde sembolik verilerle hareket etmek, bu verileri kullanarak detaya inmek ve araştırma yapmak önemlidir. Bunu MIDI ToolBox anlatımında gördük. Ancak sembolik verilerle hareket etmek için müziği, yani sesi/ses dosyasını sembolik verilere çevirmek çok daha önemlidir. Çünkü müziğin doğal akışında (yaratma/dinleme) önemli olan sestir, semboller değil. Dolayısıyla, iyi bir MIDI ToolBox kullanımı için ses dosyalarını MIDI sembolik verilerine çevirmek oldukça önemlidir.

Günümüzde ses dosyalarını MIDI'ye çeviren ve son yıllarda sayıca artarak rekabetçi gelişimini sürdürmeye devam eden yazılımlara sahibiz. Bu yazılımların bir kısmının tek işi ses dosyasını MIDI'ye çevirmek olmakla birlikte, geriye kalan büyük bir kısmı çok güçlü ses işlemleri yapabilen DAW (Digital Audio Workstation) kategorisindeki yazılımlardır. Çalışmanın bu bölümünde, iyi bir MIDI ToolBox kullanımı için şart olan sesi MIDI'ye dönüştürme işi yapan yazılımların başarısı kıyaslanacaktır. Bunun için öncelik başarı karşılaştırması yapılacak yazılımların tespiti olmuştur. Ardından, karşılaştırmada kullanılacak müzikler sınıflandırılarak seçilmiş ve son olarak bu müzikler yazılıma gönderilerek sonuçları F-Measure tekniğiyle elde edilen istatistiklerle incelenmiştir.

Ses dosyalarını MIDI'ye dönüştüren yazılımlar hakkında web aracılığıyla yapılan araştırmalar sonucunda, yaygın olarak kullanılan 12 yazılım tespit edilmiştir. Tespit edilen yazılımlardan bir kısmı sadece dönüşüm işlemi için tasarlanmış olsa da içlerinde çok amaçlı olup, sestten MIDI'ye dönüşüm işlemine olanak sağlayan yazılımlar da bulunmaktadır. Örneğin bir DAW olan Ableton Live, detaylı ses işlemlerinin yanı sıra sestten MIDI'ye dönüştürme işlemine de olanak sağlarken; Widi yazılımının tek işi, ses dosyalarını MIDI formatına dönüştürmektir. Dolayısıyla Ableton Live ve Widi karşılaştırmada ele alınan yazılımlar arasındadır.

Ancak web araştırmasında, çeviri işi yaptığını iddia etse de uygulamada hiç sonuç vermeyen ve bu nedenle karşılaştırmaya almadığımız yazılımlar da olmuştur. Örneğin Akoff işlem sırasında sık sık veri yenileyerek sonuç vermezken; Digital Ear henüz

kurulum aşamasında kendini sıfırlamıştır. Dolayısıyla, tespit edilen yazılımlardan 4 tanesi kurulum aşamasında veya sestten MIDI'ye dönüşüm esnasında çözilemeyen hatalar vermesi, çeşitli aşamalarda yazılımın kendini kapatması gibi sebeplerden dolayı karşılaştırmada kullanılamamıştır. Diğer taraftan Cubase yazılımı, yaşanan lisans problemi nedeniyle çalışmaya dahil edilmemiştir. Sonuç olarak tespit edilen 12 yazılım olmasına rağmen 5 tanesi elenerek karşılaştırmaya 7 yazılımla gidilmiştir. Tablo 15'de tespit edilen ve karşılaştırmada kullanılan yazılımlar sunulmuştur.

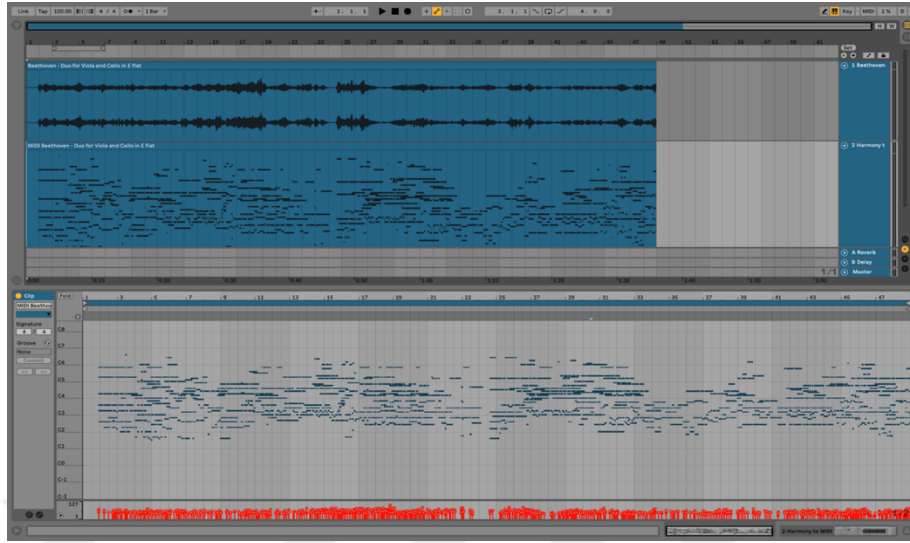
Tablo 15- Tespit edilen ve karşılaştırmada kullanılan yazılımlar

	Tespit edilen	Kullanılan
1	Ableton Live	Ableton Live
2	Akoff	Amazing MIDI
3	Amazing MIDI	FL Studio
4	Awave Studio	Intelli Score
5	Cubase	Logic Pro
6	Digital ear	Melodyne
7	FL Studio	Widi
8	Intelli Score	
9	Logic Pro	
10	Melodyne	
11	Neuratron	
12	Widi	

2.1 Karşılaştırmada Kullanılan Yazılımlar

2.1.1 Ableton Live

Ableton Live, günümüzde oldukça yaygın kullanılan bir DAW'dır. İlk sürümü Gerhard Behles, Robert Henke ve Bernard Roggendorf tarafından 2001 yılında Berlin'de geliştirilmiştir. Yazılım müzik üretimi, ses kayıt, düzenleme ve karışım gibi birçok işleme olanak sağlamakla beraber, ses dosyalarını MIDI formatına dönüştürmek için de kullanılır.



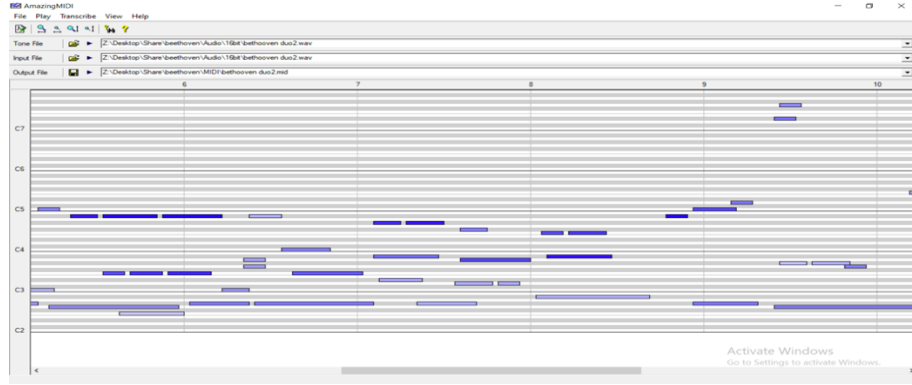
Şekil 14- Dönüştürme işlemleri esnasında Ableton Live yazılımından bir görüntü

Diğer taraftan Ableton Live, aynı zamanda karşılaştırma sonuçlarının hesaplanacağı ve karşılaştırma boyunca ihtiyaç duyulan çeşitli ses ve MIDI düzenleme işlemlerinin yapılacağı ana yazılım olarak da kullanılacaktır.

2.1.2 Amazing MIDI

Amazing MIDI, 2003 yılında Araki Software tarafından geliştirilen bir yazılımdır. Ableton Live gibi çok amaçlı değil, sadece ses dosyalarını MIDI formatına dönüştürmek için tasarlanmıştır. Yalnızca .wav formatlı ses dosyaları üzerinde işlem yapmaktadır.

Dönüşüm işlemi sonucu elde edilen MIDI versiyonu, yazılımın içinde bulunan *piano-roll* üzerinde gösterilmektedir. Elde edilen sonuçlar üzerinde düzenleme işlemleri yapmaya olanak sağlamaktadır. Dönüşüm sonucu oluşturulan MIDI, .mid uzantılı dosya olarak bilgisayara kaydedilebilmektedir.

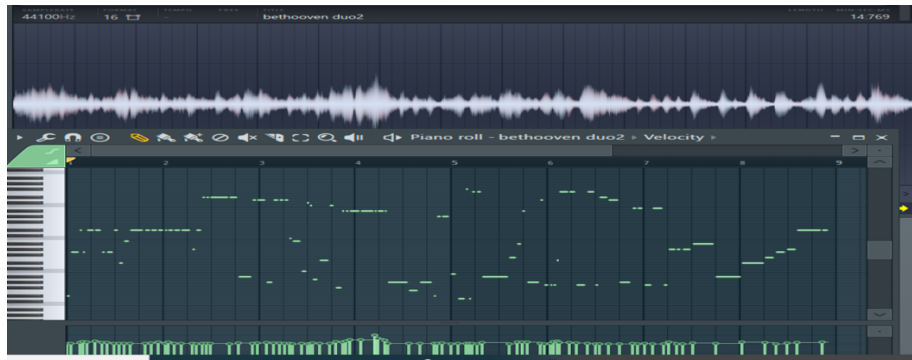


Şekil 15- Dönüştürme işlemleri esnasında Amazing MIDI yazılımından bir görüntü

2.1.3 FL Studio

FL Studio, günümüzde özellikle elektronik müzik üreticilerinin yaygın olarak kullandığı dijital ses işleme istasyonudur. Ses ve MIDI üzerinde kayıt, düzenleme gibi temel görevlerinin yanında, sesten MIDI'ye dönüştürme işlemlerine de olanak sağlamaktadır.

Belçika firması olan Image-Line tarafından geliştirilen yazılımın ilk sürümü, 1997 yılında piyasaya sürülmüştür. İlk versiyonu 1.0 olarak piyasaya sürülen yazılımın en güncel versiyonu 20.6.2'dir.



Şekil 16- Dönüştürme işlemleri esnasında FL Studio yazılımından bir görüntü

2.1.4 Intelli Score

Intelli Score, Innovative Music Systems adlı firma tarafından ses dosyalarını MIDI'ye dönüştürme işlemi için tasarlanmış bir yazılımdır. Yazılım aracılığıyla .wav .mp3 gibi ses dosyaları MIDI'ye dönüştürülebilmekte ve MIDI üzerinde düzenleme yapılabilmektedir. Yazılım bilgisayarda kayıtlı bulunan ses dosyaları haricinde, canlı olarak mikrofon aracılığıyla kaydedilen müzikleri de MIDI'ye dönüştürme imkânı sağlamaktadır.

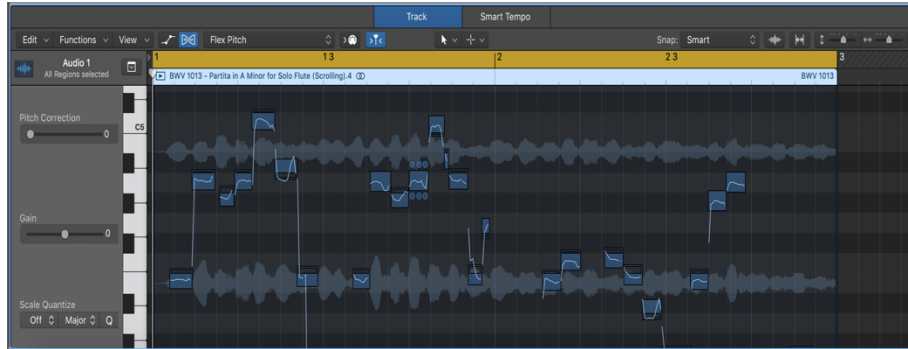


Şekil 17- Dönüştürme işlemleri esnasında Intelli Score yazılımından bir görüntü

2.1.5 Logic Pro

Logic, 1990'lı yılların başında Alman yazılım firması *Emagic* tarafından geliştirilmiştir. 2002 yılında Apple firması Emagic'i satın alarak Logic yazılımını bünyesine katmış ve ismi şu an kullanılan Logic Pro olarak değiştirilmiştir.

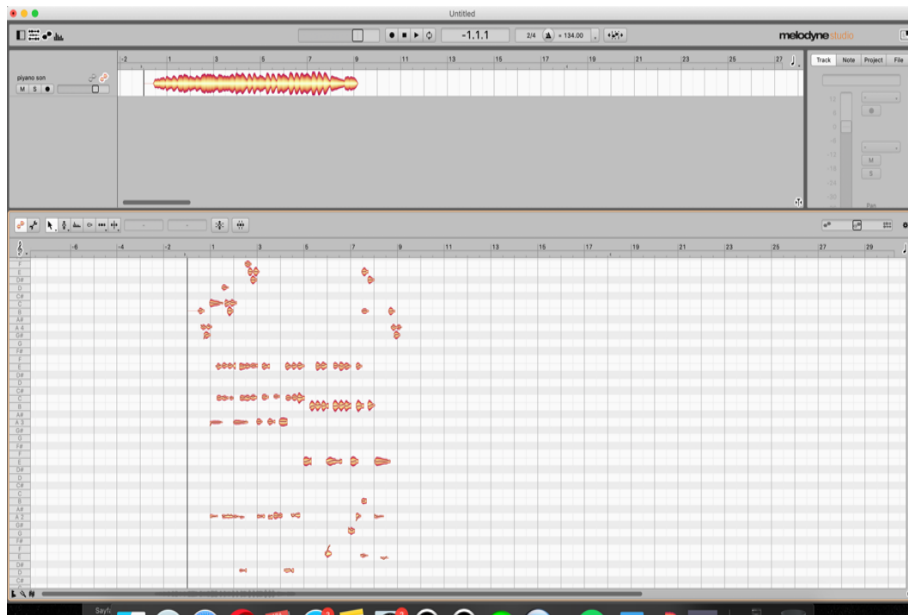
Logic Pro ses kayıt, düzenleme, miks gibi birçok farklı imkân kullanıcılara sunmaktadır. Logic Pro bu özelliklerinin yanı sıra, kayıtlı ses dosyalarını MIDI formatına dönüştürme işlemine de olanak sağlamaktadır.



Şekil 18- Dönüştürme işlemleri esnasında Logic Pro yazılımından bir görüntü

2.1.6 Melodyne

Melodyne, Celemony firması tarafından üretilmiş ve ilk versiyonu 2001 yılında piyasaya sürülmüştür. Melodyne kayıtlı ses dosyaları üzerinde zamansal ve ezgisel düzenleme işleri yapmaya olanak sağlamaktadır. Asıl amacı kayıt anında yapılan müzisyen kaynaklı ritmik veya ton dışı kalma gibi hataları düzeltmek olan Melodyne, aynı zamanda yazılımın içine aktarılan ses dosyasını MIDI'ye dönüştürmekte ve dönüştürme işlemi sonucunda elde edilen MIDI dosyasını dışarıya aktarmaya olanak sağlamaktadır.

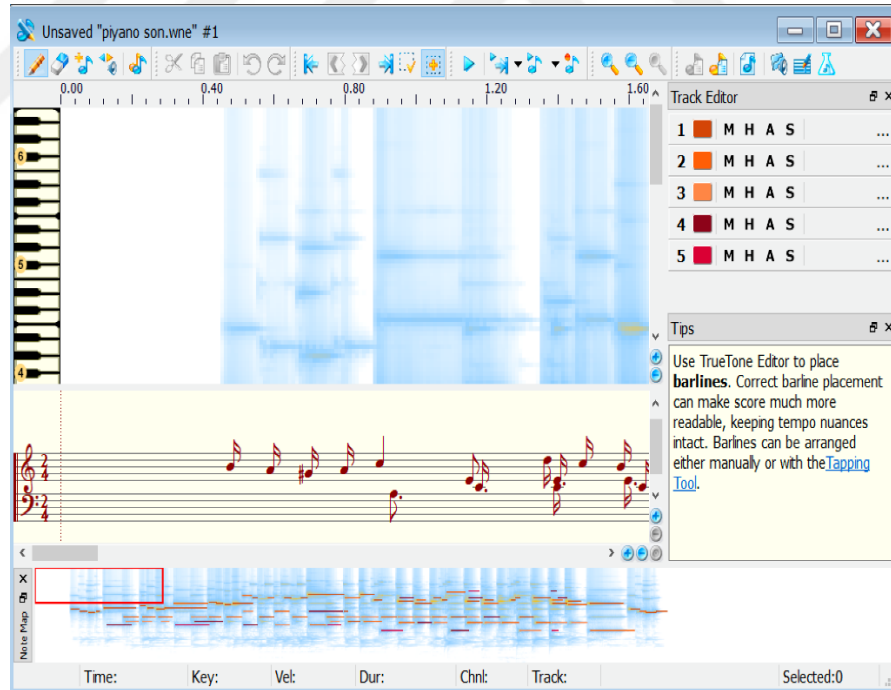


Şekil 19- Dönüştürme işlemleri esnasında Melodyne yazılımından bir görüntü

2.1.7 WIDI

Widi, Widisoft firması tarafından geliştirilen, ses dosyalarını MIDI'ye dönüştürmek amacıyla tasarlanmış bir yazılımdır. İlk olarak 1998 yılında Music Recogniton ismiyle piyasaya sürülmüştür. 2002 yılında yeni bir ara yüz ve yeni ismiyle aynı firma tarafından Widi adıyla piyasaya sunulmuştur.

Widi, .mp3 ve .wav formatlı ses dosyalarını MIDI'ye dönüştürerek elde edilen MIDI verileriyle dosya üzerinde düzenleme olanağı sunmaktadır. Günümüzde sesten MIDI'ye dönüşüm işleminde yaygın olarak kullanılmaktadır. Bilgisayar işletim sistemlerinde tek başına bir uygulama olarak çalışan versiyonu olmakla beraber, DAW için eklenti (plug-in) olarak kullanılan bir versiyonu daha bulunmaktadır. Bu çalışmada bilgisayar işletim sistemi üzerinden kendi başına çalışan (stand-alone) versiyonu kullanılmıştır.



Şekil 20- Dönüştürme işlemleri esnasında Widi yazılımından bir görüntü

2.2 Müzik Havuzu

Günümüzde üretilen ve dinlenmekte olan müzikler, yapı bakımından birbirinden oldukça farklı özelliklere sahiptir. Çok sesli ve karmaşık armonik yapılara sahip müziklerden tek sesli müziklere kadar uzanan bir çeşitlilik söz konusudur. Müziklerin armonik yapıları haricinde, kullanılan çalgıların çeşitliliği de farklılıklar göstermektedir.

Ses dosyalarını MIDI formatına dönüştüren yazılımların performans karşılaştırmasını yaparken yalnızca bir müzik örneği ya da yapı bakımından benzer örnekler üzerinden yola çıkmak yeterli olmayacaktır. Dönüştürücü yazılımların sesten MIDI'ye dönüştürme yeteneklerinin, tek sesli müzikler ile daha karmaşık yapıya sahip çoksesli müzikler üzerinde aynı olmayacağı öngörülmüştür. Bu öngörü doğrultusunda karşılaştırma işleminde kullanılacak müzikleri kategorize etmek, sınıflandırmak ihtiyacı ortaya çıkmıştır.

Ortaya çıkan ihtiyaç doğrultusunda Tek Çalgı Teksesli (TÇT), Tek Çalgı Çoksesli (TÇÇ), Çok Çalgı Teksesli (ÇÇT) ve Çok Çalgı Çoksesli (ÇÇÇ) olmak üzere 4 kategori oluşturulmuştur. Oluşturulan kategoriler sayesinde, yazılımların farklı yapılara sahip müzikler üzerindeki analiz yeteneklerinin değerlendirilmesi hedeflenmektedir. Dolayısıyla yazılımların başarı sıralamaları da her kategoriye göre farklı bir şekilde düzenlenmiştir. Oluşturulan kategoriler Tablo 17'de sunulmuştur.

Tablo 16- Dönüşüm işlemi için oluşturulan müzik havuzundaki kategoriler

Kategoriler	
1	Tek Çalgı Teksesli (TÇT)
2	Tek Çalgı Çoksesli (TÇÇ)
3	Çok Çalgı Teksesli (ÇÇT)
4	Çok Çalgı Çoksesli (ÇÇÇ)

2.2.1 Tek Çalgı Teksesli

Ses dosyalarını MIDI formatına dönüştüren yazılımların, Tek Çalgı Teksesli (TÇT) müzikler üzerindeki performans karşılaştırmasını yapmak için barok dönem besteci *J. S. Bach*'ın *Solo Flüt için La Minör Süit (BWV 1013 Partita in A Minor)* yaratısı seçilmiştir. Seçilen yaratı flüt için bestelenmiş olup çalışmada kullanılan yaratıya ait ses kaydı da flüt çalgısıyla icra edilmiştir.

Çalgıların fiziksel özelliklerine göre birbirinden farklı karakterde ses dalgaları meydana getirdiği bilinen bir gerçektir. Bazı çalgılar daha karmaşık ses dalgaları üretirken flüt, sinüs dalga formuna nispeten daha yakın bir ses ürettiği için analiz esnasında yazılımların daha başarılı tepki vereceği öngörülmüştür. Bu nedenle bu kategoride flüt ile icra edilen bir kayıt tercih edilmiştir.

Dönüştürme işlemi için yaratının Şekil 22'de sunulan ilk 2 ölçüsü yeterli olacağından, çalışmada sadece bu iki ölçünün kullanılması kararlaştırılmıştır. Yaratının teste alınacak bölümünde 31 perde bulunmaktadır.



Şekil 21- Dönüştürme işlemi için kullanılacak yaratının ilk iki ölçüsü
(*J. S. Bach, La Minör Süit*)

2.2.2 Tek Çalgı Çoksesli

Dönüştürücü yazılımların, Tek Çalgı Çoksesli (TÇÇ) müzikler üzerindeki performans karşılaştırması için çok sesli bir çalgı olan piyano tercih edilmiştir. Bu kategoride klasik dönem bestecisi W.A. Mozart'ın *Türk Marşı (The Piano Sonata No. 11 in A major – Rondo)* seçilmiştir. Klasik dönem yaratılarının, ezgisel ve armonik yapısının

yazılımların dönüştürme işlemini daha kolay yapmasına olanak sağlayacağı öngörüldüğünden, bu yaratı uygun görülmüştür.

Yazılımların karşılaştırma işlemi için yeterli olacağından yaratının aşağıdaki Şekil 23’de sunulan ilk 9 ölçüsü kullanılmıştır. Yaratının teste alınan bölümünde 104 perde bulunmaktadır.



Şekil 22- Seçilen yaratının dönüşüm işlemi için kullanılacak kısmı.
W.A. Mozart, Türk Marşı (The Piano Sonata No. 11 in A majör – Rondo)

2.2.3 Çok Çalgı Teksesli

Kayıtlı ses dosyalarını MIDI formatına dönüştüren yazılımların, Çok Çalgı Tek Sesli (ÇÇT) müzikler üzerindeki performans karşılaştırması için Nikolaki *Buselik Peşrev* kullanılmıştır. Özellikle Türk Müziğinde karşımıza çıkan bu yapı üzerinde yapılan araştırmalar, yazılımların Türk Müziği üzerindeki performansı hakkında fikir sahibi olmamıza olanak sağlamaktadır.

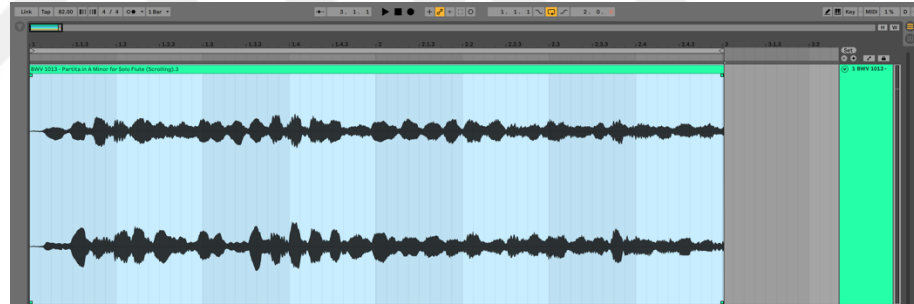
Buselik makam, klasik batı müziğindeki la minör dizisine denk gelmekte ve dolayısıyla içinde herhangi bir koma ses bulundurmamaktadır. Bu sebepten ötürü, ÇÇT kategorisinde buselik makamı tercih edilmiştir.

Dönüştürme işleminde yeterli olacağı için yaratının yalnızca birinci hanesinin, ilk 4 ölçüsü kullanılmıştır. Teste alınan bölüm 24 perdeden oluşmaktadır.

2.3 Karşılaştırmalar

Sesten MIDI'ye dönüşüm işlemi esnasında kullanmak amacıyla seçilen müziklerin, .wav formatlı ses dosyaları bilgisayara kaydedilmiştir. Ableton Live yazılımı kullanılarak daha önce belirtilen müziklerin analiz için yeterli olan kısımları kesilip, yeni ses dosyası haline getirilmiştir. Elde edilen yeni ses dosyaları dönüşüm işlemi esnasında kullanılmak üzere, .wav formatlı ses dosyası olarak bilgisayarda oluşturulan müzikler klasörüne kaydedilmiştir.

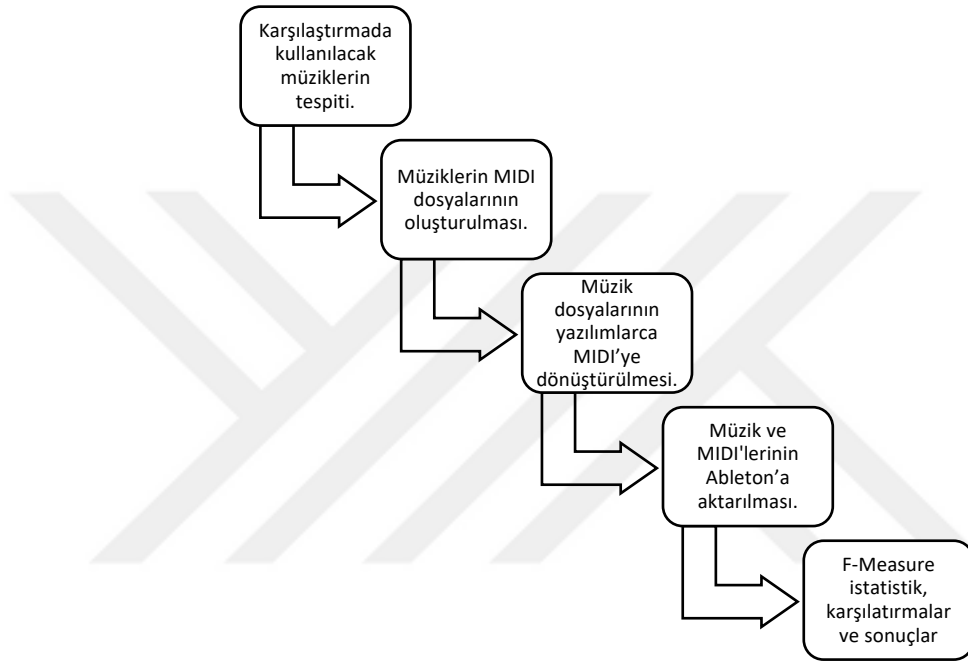
Çalışmamız için gerekli olan bölümler kesilip yeni ses dosyaları oluşturulduktan sonra, yine Ableton Live yazılımında dönüşüm işleminde kullanılacak müziklerin metronom bilgisi tespit edilmiştir. Bir sonraki adımda müziklerin başlangıç ve bitiş noktaları detaylıca incelenmiş ve MIDI'ye dönüşüm işleminin daha sağlıklı ve tutarlı olması için gerekli görülen düzenleme işlemleri yapılmıştır.



Şekil 25- MIDI'ye dönüştürülecek ses dosyası üzerindeki düzenleme işlemleri esnasında Ableton Live yazılımından bir görüntü

Uygulama esnasında kullanılmak için oluşturulan yeni ses dosyaları, karşılaştırması yapılacak yazılımların içine *File-Import* komutu kullanılarak aktarılmıştır. Daha sonra, yazılımların içinde dönüşümü yapılacak müziklerin metronom, ölçü gibi bilgileri el ile girilerek, dönüşüme hazır hale getirilmiştir. Bu aşamalardan sonra dönüşüm işlemleri sırasıyla gerçekleştirilmiştir. Yapılan işlemler sonucunda elde edilen MIDI dosyaları, karşılaştırma esnasında kullanılmak üzere, *Midi-Export* komutu kullanılarak bilgisayarda oluşturulan sonuçlar klasörüne, .mid uzantılı dosya olarak kaydedilmiştir.

Dönüşüm işlemi sonucunda elde edilen MIDI dosyalarıyla karşılaştırma yapabilmek için uygulamada kullanılan müziklerin MIDI versiyonuna ihtiyaç duyulmaktadır. Seçilen müziklerin orijinal notası esas alınarak Muse Score yazılımında el ile yazmak suretiyle yeni MIDI dosyaları oluşturulmuş, oluşturulan yeni MIDI'ler bilgisayara .mid uzantılı dosya olarak kaydedilmiştir.

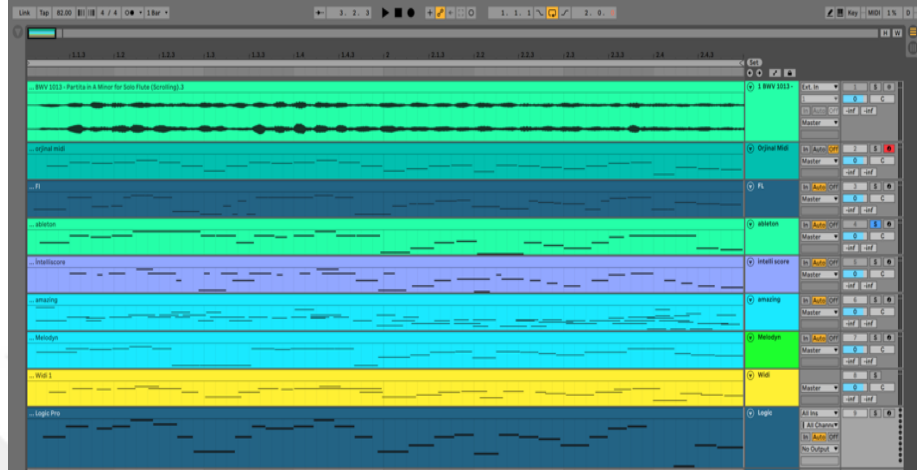


Şekil 26- Yazılımların karşılaştırılmasında kullanılan iş akışı

Yazılımların performans karşılaştırmasının yapıldığı her bir kategori için Ableton Live yazılımı içinde ilgili kategorinin isimleriyle adlandırılmış 4 ayrı proje dosyası oluşturulmuştur. İlk olarak, her bir projede birer ses kanalı açılmıştır. Dönüşüm işleminde kullanılmak üzere belirlenen ilgili kategoriye ait .wav formatlı ses dosyası yazılım içine aktarılmış ve oluşturulan ses kanalına yerleştirilmiştir.

MIDI'ye dönüştürülen müziğin bulunduğu ses kanalının altına yeni bir MIDI kanal açılmış, Muse Score yazılımında oluşturulan orijinal MIDI dosyası bu kanala aktarılmıştır. Bu aşamadan sonra karşılaştırması yapılacak 7 yazılım için birer MIDI kanal daha oluşturulmuş ve her bir kanal, ilgili yazılımın ismiyle adlandırılmıştır. Sesten

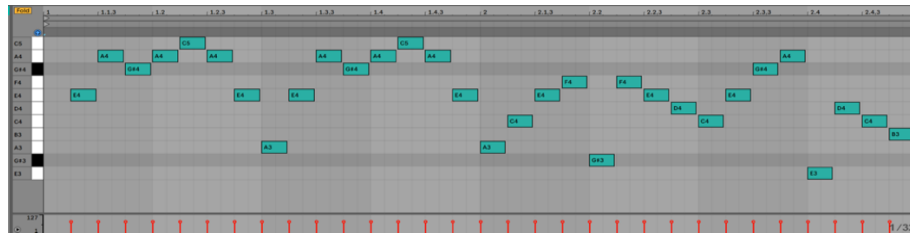
MIDI'ye dönüşüm sonucunda elde edilen MIDI dosyaları, isimlerine göre oluşturulan MIDI kanallarının içine aktarılmıştır.



Şekil 27- Karşılaştırma işlemi esnasında Ableton Live yazılımından bir görüntü

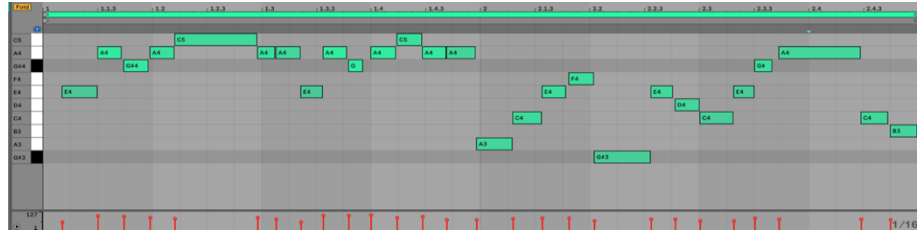
Yapılan uygulama sonucunda elde edilen MIDI'ler ile orijinal MIDI, oluşturulan proje dosyası içerisinde detaylıca incelenmiş ve karşılaştırması yapılmıştır. Karşılaştırma işlemleri manuel olarak el ile yapılmıştır. Aşağıdaki bölümde, her kategori için ayrı alt başlıklar altında dönüşümü yapılan müziğin orijinal MIDI'si ve yazılımlar tarafından yapılan işlem ile elde edilen sonuçların *piano-roll* üzerindeki görünümü sunulmuştur.

2.3.1 Kategori 1: TÇT



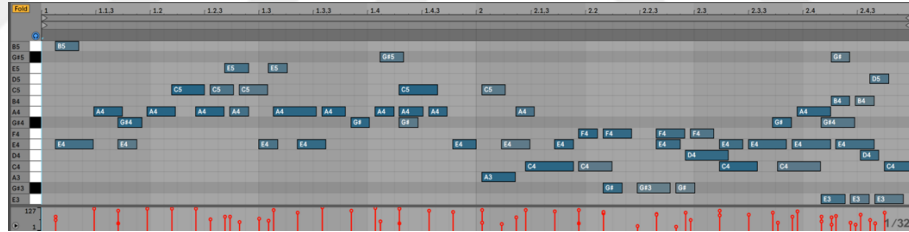
Şekil 28- TÇT orijinal MIDI

Havuzdaki TÇT kategorisindeki Bach BWV 1013 Partita La Minör, ilk 31 perdeyle teste alınmıştır. Perde uzunlukları eşit (16'lık) ve ezgi çizgisi inişli çıkışlıdır. Buna göre, kıyaslamaya tabi tutulan yazılımlar TÇT kategorisinde aşağıdaki sonuçları döndürmüşlerdir.



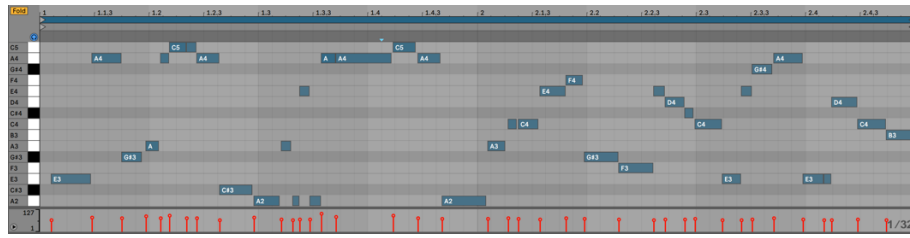
Şekil 29- TÇT Ableton Live

Orijinale kıyasla Ableton Live çıktılarında ezgi seyir çizgisinin nispeten doğru saptandığı gözlemlenirken, doğru tespit edilen perdelerin bazılarında süre sıkıntısı yaşandığı görülmektedir. Örneğin ilk beş perde doğrudur ancak birinci ve beşinci perdelerin süreleri orijinale göre daha uzundur. Bununla birlikte, eğer sürece uzun yansıyan perde oluştuğunda bir sonraki perdelerin yanlış saptandığı; tam tersi, doğru perdelerin süreleri de doğruysa bir sonraki perdelerin de doğru tespit edildiği gözlemlenmiştir. Buna göre TÇT kıyaslamada Ableton Live 25 perdeyi doğru, 6 perdeyi hatalı saptamış veya hiç tespit edememiştir.



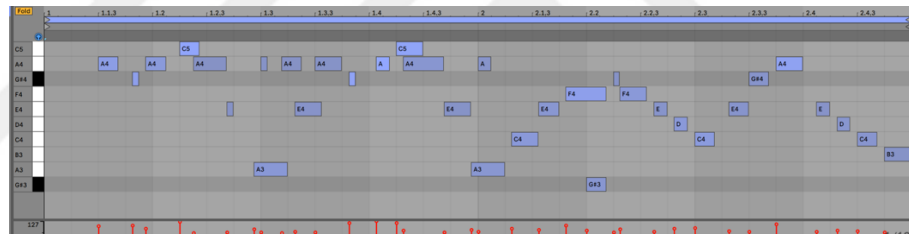
Şekil 30- TÇT Amazing MIDI

Amazing MIDI'nin elde ettiği sonuçlar incelendiğinde yazılımın doğru saptadığı perde sayısının oldukça fazla olduğu gözlemlenmektedir. Ancak müziğin içinde bulunan perdeleri doğru saptamasına rağmen aslında hiç olmayan birçok perdeyi hatalı olarak saptadığı görülmektedir. Yazılımın saptadığı ve müziğin içermediği bu perdeler elde edilen sonucu karmaşık ve gerçeklikten uzak bir yapıya dönüştürmektedir. Buna göre TÇT kıyaslamada Amazing MIDI 28 perdeyi doğru, 34 perdeyi hatalı saptamış veya hiç tespit edememiştir. Görüldüğü üzere doğru tespitlerinin yanında daha fazla hatalı tespit bulunmaktadır.



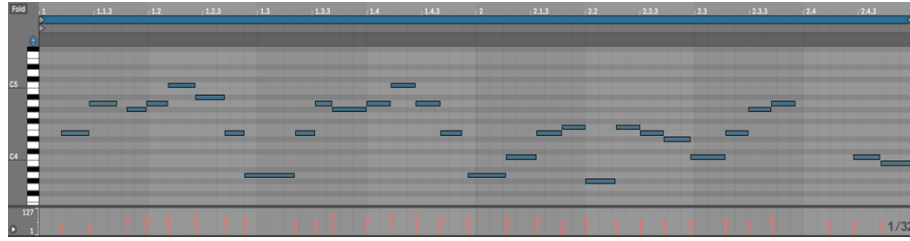
Şekil 34- TÇT FL Studio

FL Studio yazılımının elde ettiği sonuçlar incelendiğinde, doğru saptanan perdelerin yanında, müziğin içermediği perdeleri hatalı olarak saptadığı görülmektedir. Bu hatalı saptamanın elde edilen sonucun ezgisel seyrini orijinalinden uzaklaştıracak düzeyde olduğu gözlemlenmektedir. Hatalı saptamaların yanında, nota sürelerinde de çeşitli hataların olduğu görülmektedir. Buna göre TÇT kıyaslamada FL Studio 24 perdeyi doğru, 23 perdeyi hatalı saptamış veya hiç tespit edememiştir.



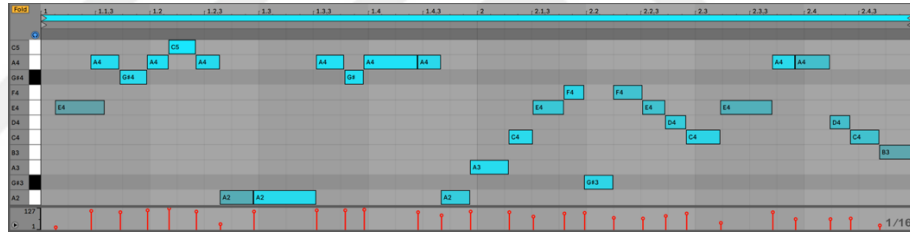
Şekil 31- TÇT Intelli Score

Intelli Score yazılımının elde ettiği sonuçlar incelendiğinde, ezgisel seyrin orijinale kıyasla özellikle son kısma doğru benzerlik gösterdiği görülmektedir. Hataların genelde perde sürelerinin saptanmasında yapıldığı ve sürelerin uzamasıyla aslında tek sesli olan bu ezgide, yer yer aynı anda iki perdenin üst üste geldiği gözlemlenmektedir. Buna göre TÇT kıyaslamada Intelli Score 27 perdeyi doğru, 7 perdeyi hatalı saptamış veya hiç tespit edememiştir.



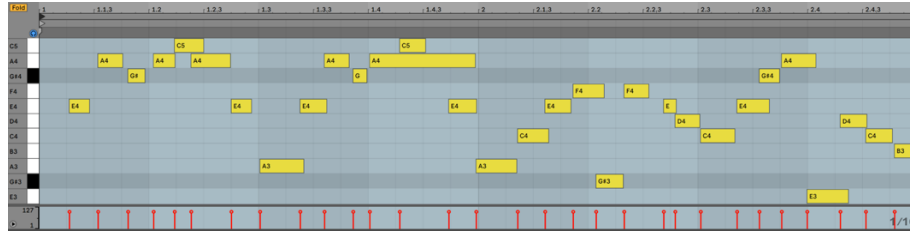
Şekil 32- TÇT Logic Pro

Logic Pro yazılımının elde ettiği sonuç incelendiğinde, ezgisel seyrin orijinale kıyasla yakın olduğu gözlemlenmektedir. Diğer yazılımlarda olduğu gibi bazı perdelerin sürelerinde hatalar gözlemlenmektedir. Ancak müziğin akışını bozacak düzeyde, müziğin içinde olmayan perdelerin hatalı tespit edilme durumu, Logic Pro yazılımında oldukça düşük gözükmemektedir. Buna göre TÇT kıyaslamada Logic Pro 27 perdeyi doğru, 4 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 33- TÇT Melodyne

Melodyne'in elde ettiği sonuç incelendiğinde, ezgi seyr çizgisinin orijinale kıyasla kısmen uyumlu olduğu görülse de bazı perdelerin saptanamadığı bazı perdelerin ise sürelerinin hatalı saptandığı görülmektedir. Örneğin sekizinci perdenin süresinin orijinale kıyasla hatalı saptandığı görülmektedir. Buna göre TÇT kıyaslamada Melodyne 25 perdeyi doğru, 6 perdeyi hatalı saptamış veya hiç tespit edememiştir.

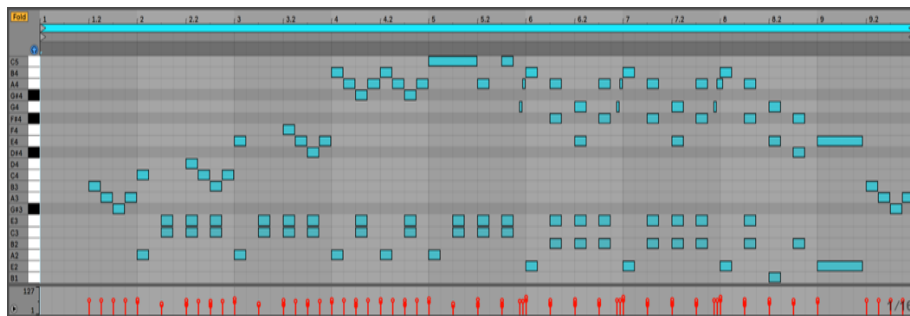


Şekil 34- TÇT Widi

Widi'nin elde ettiği sonuç incelendiğinde, ezgi seyir çizgisinin orijinale kıyasla ciddi bir benzerlik içinde olduğu görülmektedir. Nota sürelerinde, orijinal MIDI versiyonuna kıyasla farklılıklar görülmektedir. Orijinal MIDI versiyonu elle hazırlandığından perde süreleri müziğin notasıyla birebir uyumludur. Ancak müzik icra edilirken perde süreleri müzisyen kaynaklı farklılıklar göstermektedir. Bu farklılıklar müziğin akışını bozmamakla beraber müziğin insan tarafından icra edilmesinin kaçınılmaz sonucudur.

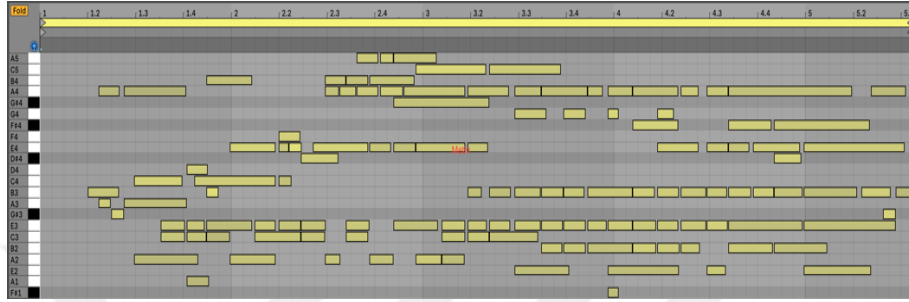
Dönüştürücü yazılımlar doğrudan müziği MIDI'ye dönüştürdüğünden sonuçlarda görülen bu ufak farklılıklar hata olarak değerlendirilmemektedir. Ancak on ikinci perde incelendiğinde, perde süresinin müziğin akışını bozacak düzeyde hatalı saptandığı görülmektedir. Buna göre TÇT kıyaslamada Widi 30 perdeyi doğru, 1 perdeyi yanlış tespit etmiştir.

2.3.2 Kategori 2: TÇÇ



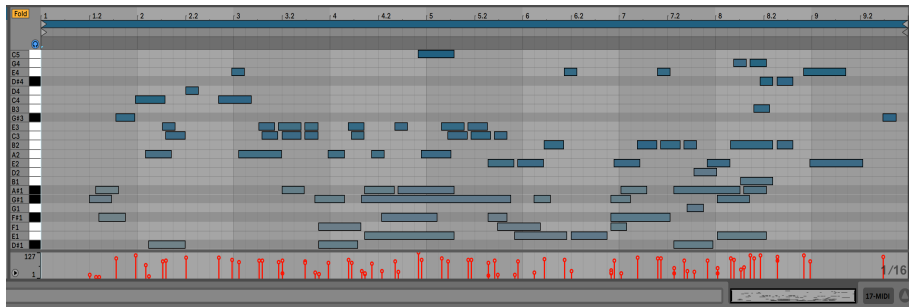
Şekil 35- TÇÇ orijinal MIDI

Havuzdaki TÇÇ kategorisinde bulunan W.A. Mozart'ın Rondo Alla Turca adlı yaratısı, ilk 9 ölçüsüyle teste alınmıştır. Yaratının teste alınan 9 ölçülük bölümü 104 perde içermektedir. Buna göre, kıyaslamaya tabi tutulan yazılımlar TÇÇ kategorisinde aşağıdaki sonuçları döndürmüşlerdir.



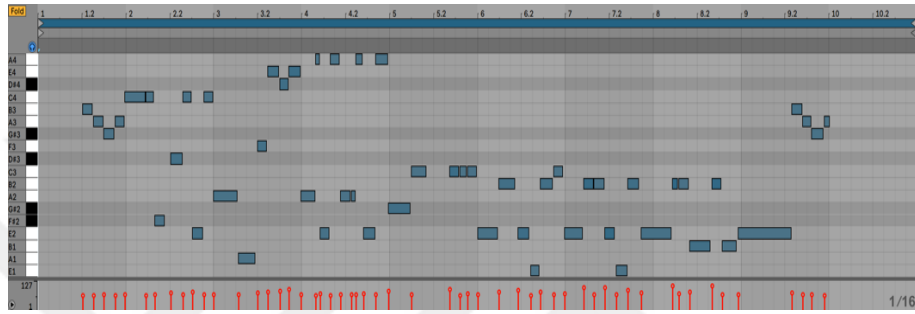
Şekil 36- TÇÇ Ableton Live

Ableton Live yazılımının elde ettiği sonuçlar incelendiğinde, birçok yazılıma göre daha başarılı olduğu gözlemlenmektedir. Perde sürelerinde hatalı tespitler göze çarpmakla beraber doğru perde tespitinin göz ardı edilemeyecek düzeyde olduğu anlaşılmaktadır. Doğru tespit edilen perdelerin sayısının yanında hatalı tespit edilen perdelerin de olduğu gözlemlenmektedir. Orijinale kıyasla çok yakın bir sonuç olmasa da kaba hatlarıyla andıran bir sonuç alındığı görülmektedir. Buna göre TÇÇ kıyaslamada Ableton Live 88 perdeyi doğru, 36 perdeyi hatalı saptamış veya hiç tespit edememiştir.



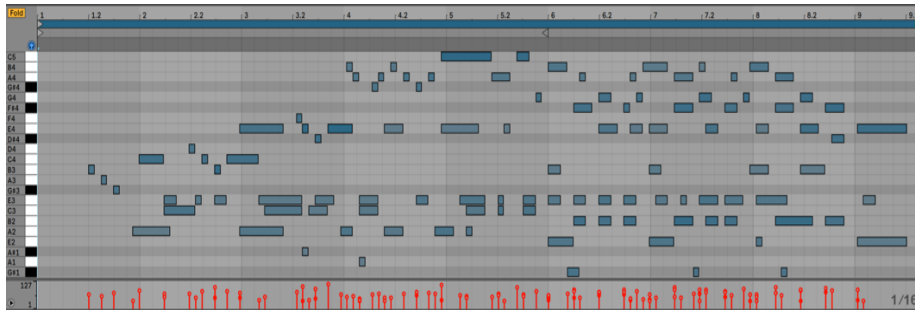
Şekil 37- TÇÇ Amazing MIDI

Amazing MIDI yazılımının elde ettiği sonuçlar incelendiğinde, gerek perdelerin tespitinde gerekse perde sürelerinde ciddi hatalar olduğu gözlemlenmektedir. Elde edilen sonucun orijinale kıyasla oldukça farklı olduğu anlaşılmaktadır. Perde süreleri ve perdelerin hatalı tespitinin yanında, orijinalinde bulunmayan perdelerin hatalı tespit oranının da oldukça yüksek olduğu gözlemlenmektedir. Buna göre TÇÇ kıyaslamada Amazing MIDI 43 perdeyi doğru, 81 perdeyi hatalı saptamış veya hiç tespit edememiştir.



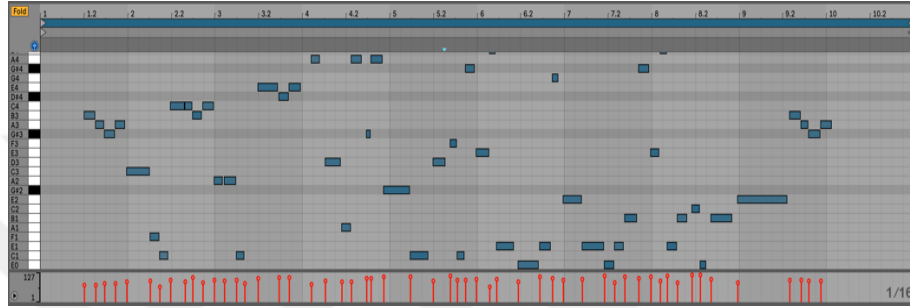
Şekil 38- TÇÇ FL Studio

FL Studio yazılımının elde ettiği sonuçlar incelendiğinde, müziğin çok sesli olmasına rağmen yazılımın tek sesli gibi algıladığı ve sonuçları bu şekilde oluşturduğu görülmektedir. Doğru perde tespitleri bulunsada gerek süre hataları gerek tespit edilemeyen perdeler göz önünde bulundurulduğunda, orijinale kıyasla gerçekten uzak bir sonuç elde edildiği gözlemlenmektedir. Buna göre TÇÇ kıyaslamada Fl Studio 31 perdeyi doğru, 73 perdeyi hatalı saptamış veya hiç tespit edememiştir.



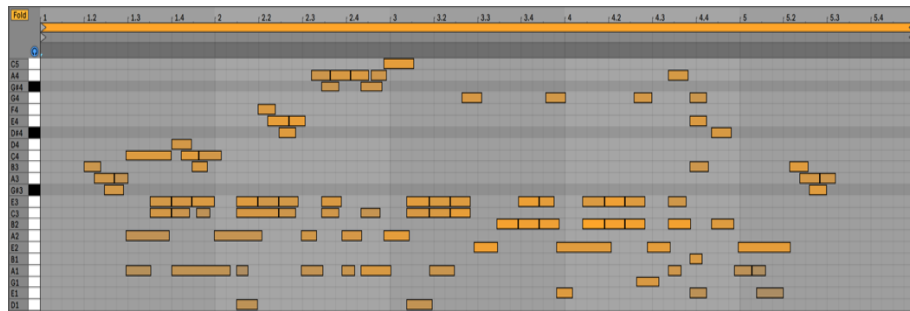
Şekil 39- TÇÇ Intelli Score

Intelli Score yazılımının elde ettiği sonuçlar incelendiğinde, perde sürelerinin Melodyne'a göre daha doğru tespit edildiği görülmektedir. Orijinalinde bulunmayan perdelerin hatalı tespiti de Melodyne'e göre daha düşük gözükmetedir. Orijinale kıyasla birebir uyumlu olmasa da yakın olarak nitelendirilebilecek bir sonuç elde edildiği gözlemlenmektedir. Buna göre TÇÇ kıyaslamada Intelli Score 86 perdeyi doğru, 34 perdeyi hatalı saptamış veya hiç tespit edememiştir.



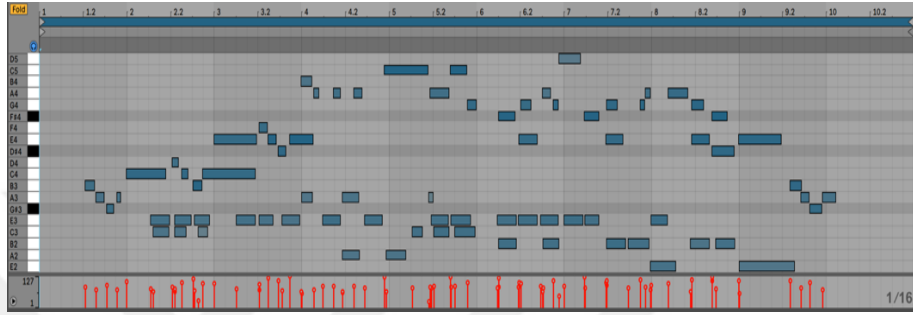
Şekil 40- TÇÇ Logic Pro

Logic Pro yazılımının elde ettiği sonuçlar incelendiğinde, tıpkı FL Studio gibi çok sesli dönüştürme işlemi yapamadığı anlaşılmaktadır. Elde edilen sonuçta perdelerin doğru tespitleri bulunsa da orijinaline kıyaslandığında bu başarılı tespitler anlam ifade etmemektedir. Çok sesli dönüştürme işlemi yapamaması nedeniyle birçok perde tespit edilemezken, orijinalinde bulunmayan perdelerin hatalı tespitleri de sonuç içinde görülmektedir. Buna göre TÇÇ kıyaslamada Logic Pro 23 perdeyi doğru, 85 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 41- TÇÇ Melodyne

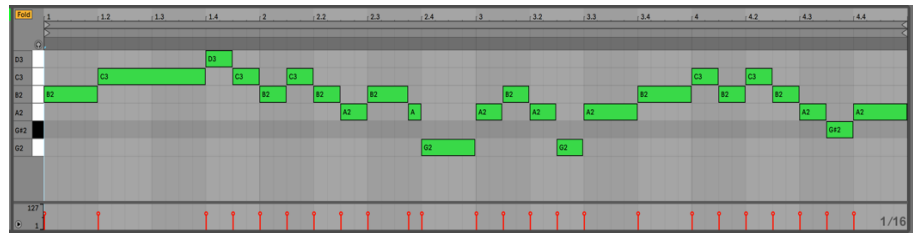
Melodyne yazılımının elde ettiği sonuçlar incelendiğinde, ilk başta perde sürelerindeki hatalı tespitler göze çarpmaktadır. Bunun yanında özellikle akor yürüyüşlerinin altına orijinalinde olmayan perdelerin eklendiği görülmektedir. Buna göre TÇÇ kıyaslamada Melodyne 75 perdeyi doğru, 42 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 42- TÇÇ Widi

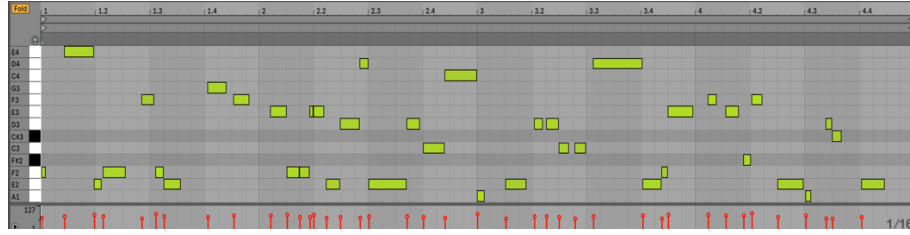
Widi yazılımının elde ettiği sonuçlar incelendiğinde, perdelerin kısmen doğru tespit edildiği görülmekle beraber, perde sürelerinin tespiti noktasında hataların olduğu gözlemlenmektedir. Müziğin akor yürüyüşlerinin kısmen doğru saptandığı bölgelerde üstte bulunan ezgisel yürüyüşün hatalı tespit edildiği, üstteki ezginin kısmen doğru tespit edildiği noktalarda ise akor yürüyüşlerinde hatalı tespitlerin olduğu gözlemlenmektedir. Buna göre TÇÇ kıyaslamada Widi 99 perdeyi doğru, 23 perdeyi hatalı saptamış veya hiç tespit edememiştir.

2.3.3 Kategori 3: ÇÇT



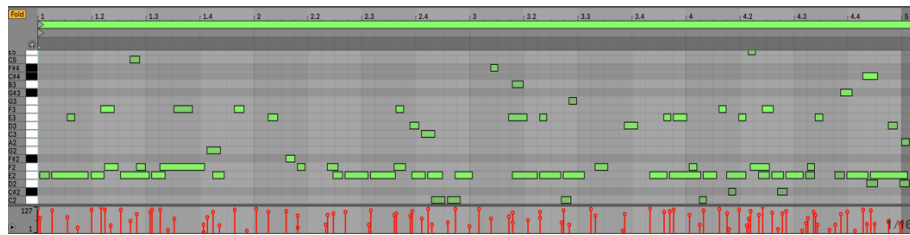
Şekil 43- ÇÇT orijinal MIDI

Havuzdaki ÇÇT kategorisinde bulunan besteci Nikolaki'nin buselik makamındaki peşrevi ilk 4 ölçüsüyle teste alınmıştır. Yaratının teste alınan ilk 4 ölçüsünde toplam 24 perde bulunmaktadır. Buna göre, kıyaslamaya tabi tutulan yazılımlar TÇÇ kategorisinde aşağıdaki sonuçları döndürmüşlerdir.



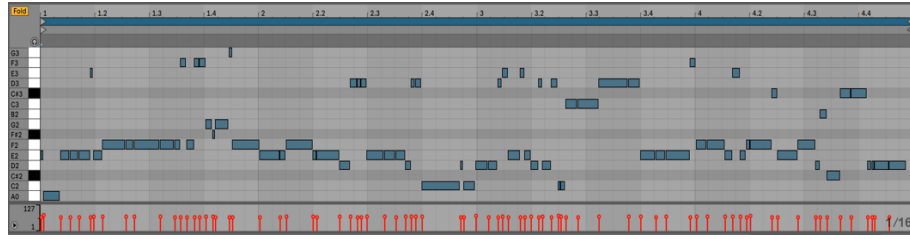
Şekil 44- ÇÇT Ableton Live

Ableton Live yazılımının elde ettiği sonuçlar incelendiğinde, orijinalinde bulunmayan perdelerin hatalı olarak tespit edildiği ve perde sürelerinde hatalı tespitler olduğu gözlemlenmektedir. Her ne kadar doğru tespit edilen perdeler bulunsu da hatalı tespit edilen perdeler, doğru tespit edilenlerin etkisini düşürmektedir. Buna göre ÇÇT kıyaslamada Ableton Live 17 perdeyi doğru, 23 perdeyi hatalı saptamış veya hiç tespit edememiştir.



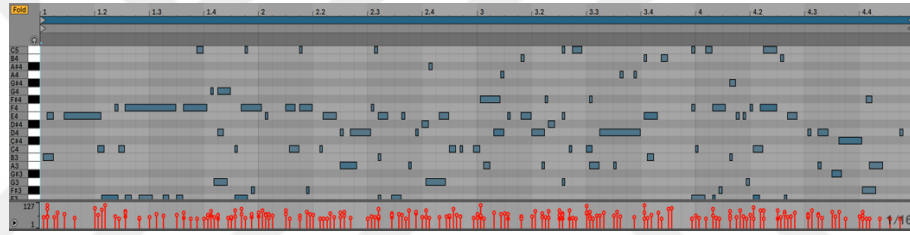
Şekil 45- ÇÇT Amazing MIDI

Amazing MIDI yazılımının elde ettiği sonuçlar incelendiğinde perde tespitinde ve perde sürelerinde oldukça fazla hata gözlemlenmiştir. Öte yandan orijinalinin içermediği birçok perdeyi hatalı olarak tespit ettiği görülmektedir. Buna göre ÇÇT kıyaslamada Amazing MIDI 13 perdeyi doğru, 47 perdeyi hatalı saptamış veya hiç tespit edememiştir.



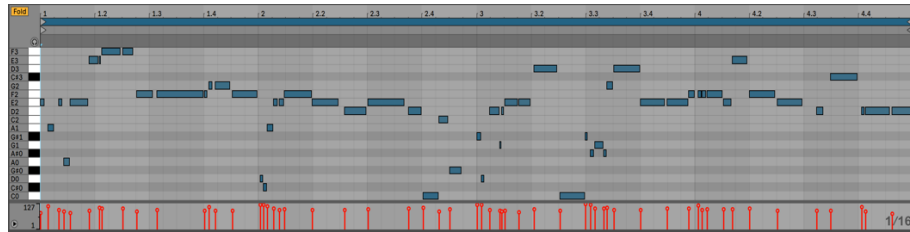
Şekil 46- ÇÇT FL Studio

FL Studio yazılımının elde ettiği sonuçlar incelendiğinde, ÇÇT kategorisindeki birçok yazılıma göre daha az hatalı perde tespiti olduğu gözlemlenmektedir. Buna göre ÇÇT kıyaslamada FL Studio 23 perdeyi doğru, 20 perdeyi hatalı saptamış veya hiç tespit edememiştir.



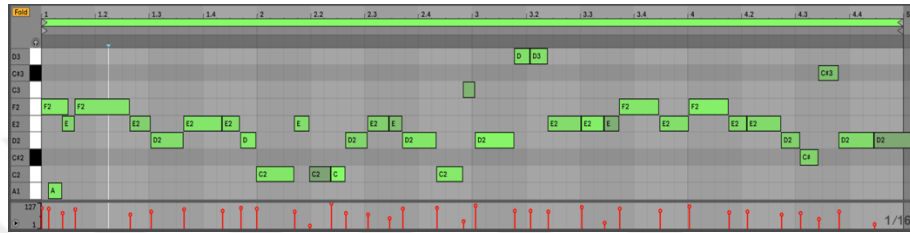
Şekil 47- ÇÇT Intelli Score

Intelli Score yazılımının elde ettiği sonuç incelendiğinde, orijinale kıyasla uzak bir sonuç karşımıza çıkmaktadır. Gerek hatalı perde tespiti gerek perde sürelerindeki hatalı tespitlerin oldukça fazla olduğu gözlemlenmektedir. Buna göre ÇÇT kıyaslamada Intelli Score 20 perdeyi doğru, 104 perdeyi hatalı saptamış veya hiç tespit edememiştir.



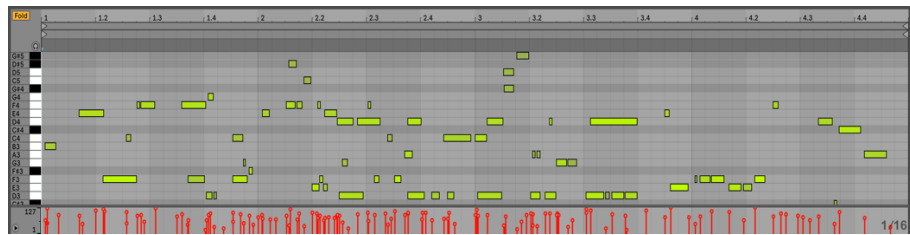
Şekil 48- ÇÇT Logic Pro

Birçok yazılımın, orijinalinde bulunmayan perdeleri hatalı tespit ederek oluşturduğu MIDI versiyonu içinde gösterdiği göz önünde bulundurulduğunda, Logic Pro yazılımının hatalı tespit oranının düşük olduğu gözlemlenmektedir. Her ne kadar orijinale kıyasla birebir sonuç elde edemese de doğru tespit oranının küçümsenemeyecek seviyede olduğu gözlemlenmektedir. Buna göre ÇÇT kıyaslamada Logic Pro 18 perdeyi doğru, 23 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 49- ÇÇT Melodyne

Melodyne yazılımının elde ettiği sonuçlar incelendiğinde, perde tespitinde zayıf olduğu gözlemlenmektedir. Perde sürelerinde de çeşitli hatalar görülmektedir. Müziğin içermediği perdeleri, hatalı tespit etme noktasında birçok yazılıma göre daha az hata yapmış olduğu gözlemlense de sonucun orijinale kıyasla oldukça farklı olduğu anlaşılmaktadır. Buna göre ÇÇT kıyaslamada Melodyne 10 perdeyi doğru, 18 perdeyi hatalı saptamış veya hiç tespit edememiştir.

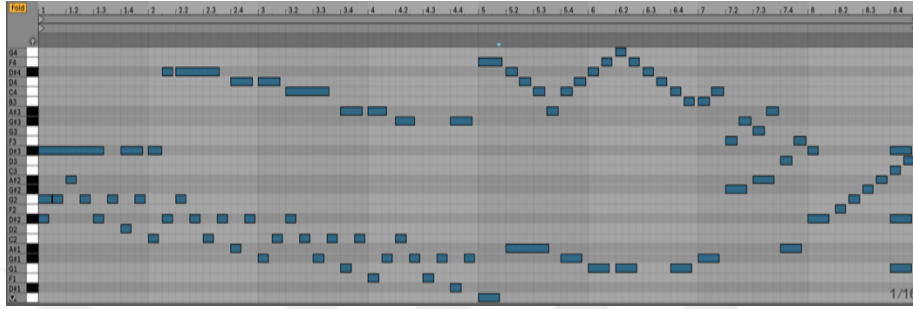


Şekil 50- ÇÇT Widi

Widi yazılımının elde ettiği sonuçlar incelendiğinde doğru tespit edilen perdelerin sayısının az olmadığı görülmektedir. Ancak orijinalinde bulunmayan perdeleri hatalı olarak tespit etme oranı oldukça fazladır. Aslında müziğin içermediği ama varmış gibi hatalı tespit edilen bu perdelerin sonucu doğrudan etkilediği ve gerçeklikten

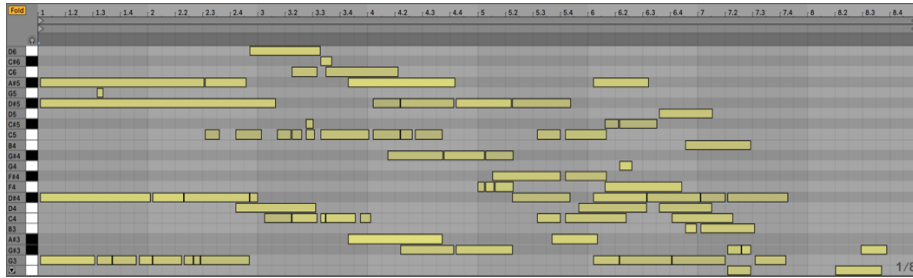
uzaklaştırdığı gözlemlenmektedir. Buna göre ÇÇT kıyaslamada Widi 20 perdeyi doğru, 53 perdeyi hatalı saptamış veya hiç tespit edememiştir.

2.3.4 Kategori 4: ÇÇÇ



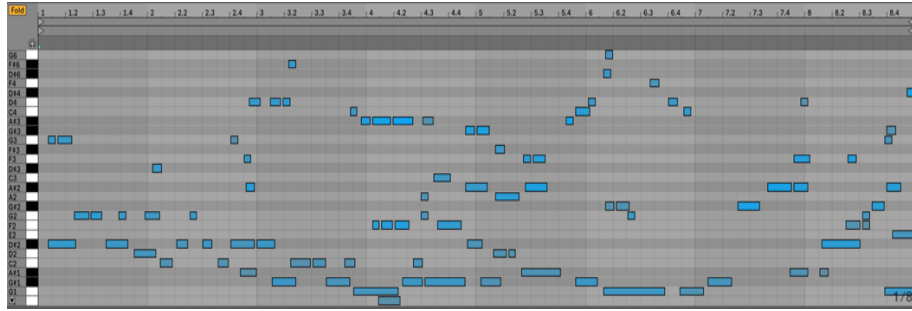
Şekil 51- ÇÇÇ orijinal MIDI

Havuzdaki ÇÇÇ kategorisinde bulunan, klasik dönem bestecisi Beethoven'ın Duo for Viola and Cello adlı yaratisı, 9 ölçülük bölümüyle teste alınmıştır. Yaratının teste alınan 9 ölçülük bölümünde 89 adet perde bulunmaktadır. Buna göre, kıyaslamaya tabi tutulan yazılımlar ÇÇÇ kategorisinde aşağıdaki sonuçları döndürmüşlerdir.



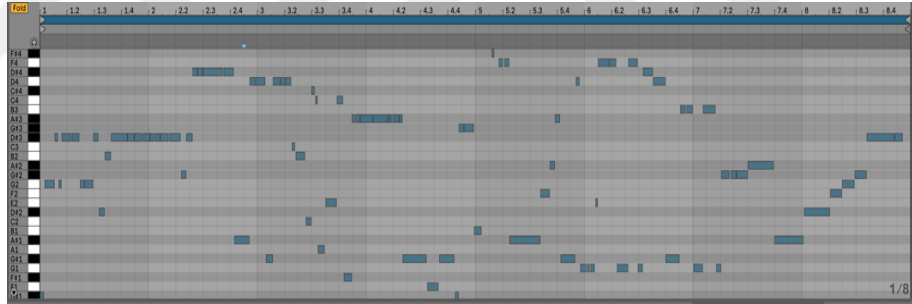
Şekil 52- ÇÇÇ Ableton Live

Ableton Live yazılımının elde ettiği sonuç incelendiğinde, perde tespitinde belirli bir başarının yakalandığı görülse de perde sürelerindeki hataların fazlalığı dikkat çekmektedir. Her ne kadar doğru tespitler olsa da hatalı tespit sayısı da oldukça fazladır. Sonuca genel olarak bakıldığında, dönüştürüldüğü müziği temsil etmekten uzak olduğu gözlemlenmektedir. Buna göre ÇÇÇ kıyaslamada Ableton Live 72 perdeyi doğru, 93 perdeyi hatalı saptamış veya hiç tespit edememiştir.



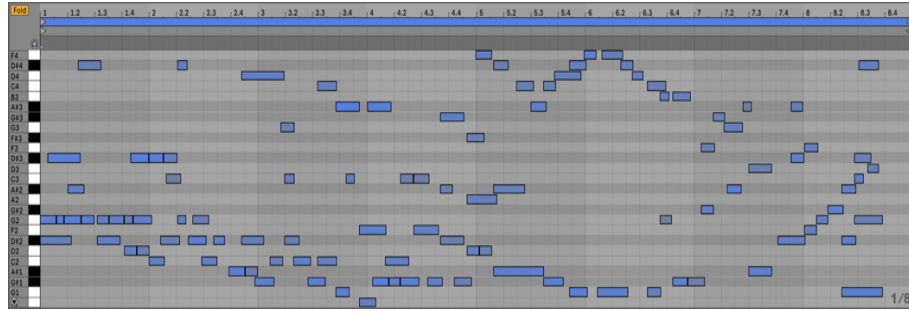
Şekil 53- ÇÇÇ Amazing MIDI

Amazing MIDI yazılımının elde ettiği sonuçlar incelendiğinde doğru tespit oranının oldukça düşük olduğu, bunun yanında hatalı perde tespitlerinin de dikkat çekici düzeyde olduğu gözlemlenmektedir. Bu yazılımın elde ettiği sonuç genel olarak incelendiğinde orijinal MIDI versiyonuyla uyumlu olmadığı görülmektedir. Buna göre ÇÇÇ kıyaslamada Amazing MIDI 42 perdeyi doğru, 69 perdeyi hatalı saptamış veya hiç tespit edememiştir.



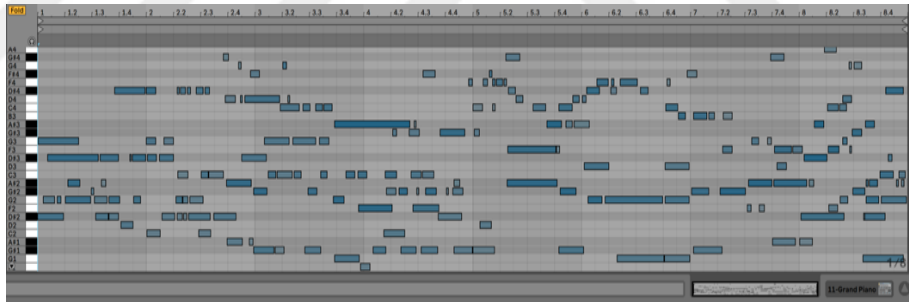
Şekil 54- ÇÇÇ FL Studio

FL Studionun elde ettiği sonuçlar incelendiğinde her ne kadar doğru perde tespitleri olduğu görülse de çok sesli olan bu müziği tek sesliymiş gibi dönüştürdüğü gözlemlenmektedir. Orijinale kıyasla çok ciddi bir fark olduğu ortadadır. Buna göre ÇÇÇ kıyaslamada FL Studio 28 perdeyi doğru, 74 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 57- ÇÇÇ Melodyne

Melodyne yazılımının elde ettiği sonuç incelendiğinde, doğru tespit sayısının birçok yazılıma göre daha yüksek olduğu gözlemlenmektedir. Ancak melodyne'ın da hatalı perde tespiti bulunmaktadır. Orijinale kıyaslandığında yer yer benzerlikler gözlemlense de sonuç tatmin edici, bir başka deyişle dönüştürüldüğü müziği temsil edebilecek düzeyde başarılı değildir. Buna göre ÇÇÇ kıyaslamada Melodyne 78 perdeyi doğru, 28 perdeyi hatalı saptamış veya hiç tespit edememiştir.



Şekil 58- ÇÇÇ Widi

Widi yazılımının elde ettiği sonuç incelendiğinde, doğru tespit edilen perdelerin varlığı görülse de hatalı perde tespitinin oldukça yüksek olduğu gözlemlenmektedir. Özellikle TÇT sesli kategorisindeki başarısı göz önünde bulundurulduğunda ÇÇÇ kategorisinde elde ettiği sonuç şaşırtıcıdır. Buna göre ÇÇÇ kıyaslamada Widi 65 perdeyi doğru, 115 perdeyi hatalı saptamış veya hiç tespit edememiştir.

2.4 Ölçümleme ve Sonuçlar

Sesten MIDI'ye dönüşüm yapan yazılımlar tarafından döndürülen sonuçlar incelenerek yazılımların doğru saptadığı, yanlış saptadığı ve hiç saptayamadığı perde sayıları tespit edilmiş ve hesaplamalarda kullanılmak üzere not edilmiştir. Karşılaştırma sonuçlarının bilimsel bir temele oturtulması için *F-Measure* ölçüm tekniği kullanılmıştır.

F-Measure, karşılaştırılan iki verinin doğru saptanma oranını belirten bir ölçümdür. Bu çalışmada verinin karşılığı, orijinal ve dönüşüm sonucu elde edilen MIDI verileridir (notalar, perdeler). *F-Measure* ölçümünün sonucu 0 ile 1 arasında çıkmakta ve elde edilen değer ne kadar yüksek olursa sonuç o kadar başarılı olmaktadır. *F-Measure* hesaplamasının formülü aşağıda sunulduğu gibidir.

$$F = \frac{2PR}{P + R}$$

F-Measure (F) ölçümlemede P (Precision), kesinlik olarak adlandırılmaktadır. P değeri, yazılımlar tarafından elde edilen sonuçlardaki doğru saptanan verilerin, bütün saptanan verilere oranını ifade etmektedir. Saptanan veriler içindeki doğru saptanmış veriler TP, (True Positive) yanlış saptanmış verilerse FP (False Positive) olarak adlandırılmaktadır. Bu çalışmada yazılımların doğru tespit ettiği notalar TP, yanlış tespit ettiği notalar ise FP olarak tanımlanmıştır. P değerinin hesaplanmasında aşağıdaki formül kullanılmaktadır.

$$P = \frac{TP}{TP + FP}$$

R (Recall) değeriye hassasiyet olarak adlandırılmakta ve doğru saptanmış verilerin, saptanması gereken bütün verilere oranını ifade etmektedir. R değeri hesaplamalarında kullanılan FN'nin (False Negative) karşılığı, olması gereken kısımda doğru ya da yanlış hiçbir notanın saptanamaması olarak tanımlanmaktadır. R değerine ulaşmak için aşağıda sunulan formül kullanılmaktadır.

$$R = \frac{TP}{TP + FN}$$

Örneğin test edilen yazılımların içinde bulunan Ableton Live'ın, TÇT kategorisinde döndürdüğü sonuç incelendiğinde, TP 25, FP 2, FN 4 olarak tespit edilmiştir. İlk olarak P ve R değeri aşağıdaki gibi hesaplanmıştır.

$$P = \frac{25}{25+2} \quad P = 0,92 \quad R = \frac{25}{25+4} \quad R = 0,86$$

Elde edilen değerler ile F-Measure aşağıdaki gibi hesaplanmıştır.

$$F = \frac{2 \cdot 0,92 \cdot 0,86}{0,92 + 0,86} \quad F = 0,88$$

Tüm karşılaştırmaların F-Measure parametre değerleriyse Tablo 18'de belirtilmiştir.

Tablo 17- Karşılaştırma sonuçlarına göre F-Measure parametre değerleri

	TÇT			TÇÇ			ÇÇT			ÇÇÇ		
	TP	FP	FN	TP	FP	FN	TP	FP	FN	TP	FP	FN
Ableton Live	25	2	4	88	33	3	17	21	1	72	86	7
Amazing MIDI	28	33	1	43	34	47	13	45	2	42	46	23
FL Studio	24	12	1	31	15	54	23	19	1	28	41	33
Intelli Score	27	5	2	86	25	9	20	103	1	41	35	33
Logic Pro	27	3	1	23	29	56	18	21	1	13	58	27
Melodyne	25	4	2	75	17	25	10	17	1	78	22	6
WIDI	30	1	0	99	20	3	20	52	1	65	112	3

Tablo 18- Tüm kategorilerde yazılımların F-Measure, Precision ve Recall sonuçları

TÇT	F-Measure	Precision	Recall
Widi	97	96	99.9
Logic Pro	92	90	96
Ableton Live	88	92	86
Intelli Score	88	84	93
Melodyne	88	86	92
FL Studio	66	66	66
Amazing MIDI	60	45	96
TÇÇ	F-Measure	Precision	Recall
Widi	89	83	97
Ableton Live	82	72	96
Intelli Score	82	77	90
Melodyne	77	81	75
Amazing MIDI	50	55	47
FL Studio	46	67	36
Logic Pro	34	44	29
ÇÇT	F-Measure	Precision	Recall
FL Studio	68	54	95
Logic Pro	61	46	94
Ableton Live	59	44	94
Melodyne	51	37	90
Widi	41	27	95
Amazing MIDI	34	22	86
Intelli Score	27	16	95
ÇÇÇ	F-Measure	Precision	Recall
Melodyne	59	45	91
Ableton Live	54	47	64
Amazing MIDI	42	40	45
Intelli Score	53	53	55
Widi	22	18	32
FL Studio	84	78	92
Logic Pro	51	36	95

TÇT kategorisi altında yapılan dönüştürme işlemleriyle elde edilen sonuçlar incelendiğinde, en başarılı yazılımın Widi olduğu görülmektedir. Başarı sıralamasında Widi'yi Logic Pro takip etmektedir. Bu kategorideki sonuçlar incelendiğinde özellikle Widi'nin, müziğin orijinal MIDI'sine büyük oranda yakın sonuçlar elde ettiği görülmektedir.

TÇÇ kategorisinde de ölçümler sonucu Widi'nin en başarılı yazılım olduğu görülmektedir. Widi'nin ardından Ableton Live yazılımı gelmektedir. Yazılımların dönüştürme işleminde, genel olarak TÇT müziklere göre daha başarısız sonuçlar elde ettiği gözlenmektedir.

Widi, dönüşümü yapılan müziğin orijinal MIDI versiyonuna diğer yazılımlara göre daha yakın sonuç elde etse de elde edilen sonucun müzik analizinde kullanılabilecek doğrulukta olmadığı anlaşılmaktadır.

ÇÇT kategorisinde ise en başarılı yazılımın FL Studio olduğu gözükmemektedir. FL Studio'yu, Logic pro yazılımı takip etmektedir. Ancak genel olarak diğer kategorilere göre başarı oranında ciddi bir düşüş gözükmemektedir. Sonuçlar incelendiğinde müzik analizinde kullanılmak üzere ihtiyaç duyulan MIDI dosyalarının, ÇÇT müzikler üzerinde kullanılmasının, sağlıklı bir yöntem olmadığı anlaşılmaktadır.

Sesten MIDI'ye dönüştürücü yazılımların başarı oranlarının en düşük olduğu kategorinin, ÇÇÇ kategorisi olduğu anlaşılmaktadır. Yapılan dönüştürme işleminde en başarılı yazılımın Melodyne, ardından Ableton Live olduğu anlaşılmaktadır. Melodyne her ne kadar diğer yazılımlara göre daha başarılı olsa da müzik analizinde kullanılmak üzere ihtiyaç duyulan MIDI'leri hazırlamak için yeterli değildir

Tablo 19- Tüm kategorilerde F-Measure sonuçları

	TÇT	TÇÇ	ÇÇT	ÇÇÇ
Ableton Live	%88	%82	%59	%54
Amazing MIDI	%66	%50	%34	%42
FL Studio	%66	%46	%68	%84
Intelli Score	%88	%82	%27	%53
Logic Pro	%92	%34	%61	%51
Melodyne	%88	%77	%51	%59
WIDI	%97	%89	%41	%22

Sonuçların genel analizi yapıldığında, yazılımların TÇT müzikler üzerinde diğer kategorilere göre daha başarılı sonuçlar elde ettiği anlaşılmaktadır. İkinci olarak TÇÇ gelirken ÇÇT üçüncü, son olarak ise ÇÇÇ kategorisi gelmektedir.

Yazılımlar her ne kadar tek sesli müzikler üzerinde daha başarılı olarak değerlendirilebilse de günümüzde müziklerin genellikle çoksesli olduğu bir gerçekliktir. Buradan yola çıkılarak, günümüzde geliştirilmiş olan sesten MIDI'ye dönüştürücü yazılımların, müzik analizinde ihtiyaç duyulan MIDI versiyonlarını oluşturmak için kullanılmasının, sağlıklı bir yöntem olmadığı anlaşılmaktadır.

Yapılan deney ve sonuçların incelenmesiyle görülmektedir ki bu yazılımlar, dönüştürülen müziğe bire bir uyumlu MIDI versiyonu hazırlayamamaktadır. Ancak özellikle tek sesli müzikler üzerinde göz ardı edilemeyecek bir başarı olduğu gözlemlenmektedir. Her ne kadar müzik analizi için uygun olmasa da Widi'nin farklı amaçlar için tek sesli müzikler üzerinde kullanılabilir bir yazılım olduğu anlaşılmaktadır.

2.5. Sonuçların Bazı MIDI Toolbox Fonksiyonlarında Kullanımı

Müzik seslerini MIDI verilerine çeviren yazılımların karşılaştırmasında açıkça görülmektedir ki, TÇT (Tek Çalgı Teksesli), TÇÇ (Tek Çalgı Çoksesli) kategorilerinde diğerlerine göre en başarılı yazılım olarak WIDI ön plana çıkmıştır. ÇÇT (Çok Çalgı Teksesli) kategorisinde FL Studio en başarılı yazılım olmakla beraber Son kategori olan ÇÇÇ (Çok Çalgı Çoksesli) özelliğindeki müziklerdeyse Melodyne ilk sırayı almaktadır.

Çalışmanın bu son aşamadaysa, elde edilen karşılaştırma sonuçlarına göre ortaya çıkan MIDI verilerinin tekrar MIDI Toolbox içindeki kullanımlarını incelemek yerinde olacaktır. Bu sayede, bu çalışmaya adını veren MIDI Toolbox'ın kullanımı konusunda ortaya çıkan tüm sonuçların bir tür testi yapılmış olacaktır.

Bunun için öncelikli olarak MIDI Toolbox içinden aşağıda açıklanacak olan ve en yaygın kullanılan fonksiyonlar seçilmiştir. Bu fonksiyonlarda önce havuzdaki yaratıların orijinal MIDI verileri kullanılmış; ardından, karşılaştırma yapılan yazılımlardan elde edilen MIDI verileri fonksiyonda test edilmiştir. Böylece, karşılaştırması yapılan dönüşüm yazılımlarının ve hatta bir MIDI veri bütününün MIDI Toolbox ile olan başarılı analiz ilişkisi ortaya çıkarılmıştır.

Buna göre, müzik havuzunda bulunan müziklerin orijinal MIDI verileri ile dönüştürücü yazılımlar tarafından elde edilen MIDI verileri, *readmidi* fonksiyonu kullanılarak *nmat* yapısına dönüştürülmüştür. *nmat* yapısına dönüştürülen MIDI verileri üzerinde sırasıyla *kkkey*, *keymode* ve *nnotes* fonksiyonları kullanılmıştır. Fonksiyonların uygulanmasıyla elde edilen sonuçlar aşağıda tablolar aracılığıyla sunulmuştur.

Tablo 20- Fonksiyonların TÇT kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar

TÇT	<i>Keymode</i>	<i>Kkkey</i>	<i>nnotes</i>
Orijinal	1	22	31
Widi	1	22	30
Logic Pro	1	22	30
Ableton Live	1	22	27
Intelli Score	1	22	34
Melodyne	1	22	28
FL Studio	1	22	27
Amazing MIDI	1	22	61

Tablo 20'deki sonuçlar incelendiğinde, *keymode* ve *kkkey* fonksiyonlarının TÇT kategorisinde bulunan bütün MIDI verileri üzerinde aynı sonucu verdiği görülmektedir. Perde sayısını analiz eden *nnotes* fonksiyonunun ise yazılımlar tarafından dönüştürülen MIDI verileri üzerinde farklı sonuçlar verdiği ve bu sonuçların dönüştürücü yazılımların başarı sıralamasıyla uyumlu olduğu anlaşılmaktadır.

Tablo 21- Fonksiyonların TÇÇ kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar

TÇÇ	<i>Keymode</i>	<i>Kkkey</i>	<i>nnotes</i>
Orijinal	1	17	104
Widi	1	17	120
Ableton Live	1	5	127
Intelli Score	1	17	114
Melodyne	1	22	92
Amazing MIDI	1	5	77
FL Studio	1	5	54
Logic Pro	1	5	54

Tablo 21'deki sonuçlar incelendiğinde, *keymode* fonksiyonunun TÇÇ kategorisinde bulunan bütün MIDI verileri üzerinde aynı sonuçları verdiği görülmektedir. *Kkkey* ve *nnotes* fonksiyonlarıyla yapılan işlemler sonucunda ise TÇT kategorisinde olduğu gibi genel olarak yazılımların başarı sıralamasıyla uyumlu sonuçlar alındığı gözlemlenmektedir.

Tablo 22- Fonksiyonların ÇÇT kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar

ÇÇT	<i>Keymode</i>	<i>Kkkey</i>	<i>Nnotes</i>
Orijinal	1	15	24
FL Studio	1	6	79
Logic Pro	1	15	58
Ableton Live	1	17	40
Melodyne	1	15	34
Widi	1	15	123
Amazing MIDI	1	10	98
Intelli Score	1	15	213

Tablo 22'deki sonuçlar incelendiğinde, *keymode* fonksiyonun ÇÇT kategorisinde bulunan bütün MIDI üzerinde aynı sonucu verdiği görülmektedir. Önceki incelenen iki kategorinin aksine *kkkey* ve *nnotes* fonksiyonuyla elde edilen sonuçların, yazılımların başarı sıralamasına büyük oranda uyumlu olmadığı anlaşılmaktadır.

Tablo 23- Fonksiyonların ÇÇÇ kategorisinde bulunan MIDI dosyaları üzerinde uygulanmasıyla elde edilen sonuçlar

ÇÇÇ	<i>Keymode</i>	<i>Kkkey</i>	<i>Nnotes</i>
Orijinal	2	4	89
Melodyne	2	4	108
Ableton Live	2	13	175
Amazing MIDI	2	4	94
Intelli Score	2	4	78
Widi	2	4	210
FL Studio	2	4	94
Logic Pro	2	4	94

Tablo 23'deki sonuçlar incelendiğinde, *keymode* fonksiyonunun ÇÇÇ kategorisinde bulunan bütün MIDI verilerinde aynı sonucu verdiği, *kkkey* fonksiyonunun ise sadece Ableton Live tarafından elde edilen MIDI dosyası üzerinde farklı sonuç verdiği

görülmektedir. *Nnotes* fonksiyonuyla elde edilen sonuçların ise yazılımların başarı sıralamasıyla tam olarak uyumlu olmadığı gözlemlenmektedir.

Bu sonuçlara göre WIDI, MIDI Toolbox fonksiyonlarının kullanımı öncesinde tek sesli bir ses dosyasını MIDI verisine çevirmek için tavsiye edilebilir. Bilimsel olarak WIDI'yle elde edilen dönüşüm sonucu verileri, MIDI Toolbox aracılığıyla analiz için kullanılabilir. Aynı şekilde henüz kusursuz bir çevrim işlemi yapamasa da çok sesli müziklerin MIDI çevriminde Melodyne kullanılabilir. Diğer taraftan son olarak şunu belirtmek gerekir ki, çevrim olsun veya olmasın MIDI verileriyle müzik analizinde kullanılacak MIDI Toolbox'ın oldukça etkili bir araç olduğu görülmektedir. *Nnotes* fonksiyonuyla elde edilen sonuçların ise yazılımların başarı sıralamasıyla tam olarak uyumlu olmadığı gözlemlenmektedir.

SONUÇ

MTB incelendiğinde, müzik analizi esnasında kullanıcıların ihtiyaç duyabileceği birçok işlemi bilgisayar ortamında otomatik olarak gerçekleştirmeye olanak sağladığı görülmektedir. MTB fonksiyonlarının sahip olduğu geliştirilebilir modüler yapı, kullanıcılara müzik analizinde özel ihtiyaçlar karşısında farklı çözümler üretme imkânı tanımaktadır.

MTB'nin sahip olduğu özellikler ile müzik analizini manuel yöntemlere göre çok daha hızlı yapabildiği görülmektedir. Kazandırdığı zamanın yanı sıra, sonuçların tutarlı olması yapılan çalışmalar için oldukça önemli bir yere sahiptir. MIDI verileri üzerinden aralıkların ve notaların dağılımı, ezginin yönsel hareketi gibi matematiksel hesaplama yöntemleriyle yapılan analiz işlemleri, analizi yapılacak olan müziğin türden bağımsız aynı tutarlılıkla sonuç vermektedir. MTB'nin müzik analizinde kullanılan güçlü bir araç olduğu anlaşılmaktadır. Manuel yöntemlerin aksine, kullanıcılara ciddi bir zaman kazandırarak enerjilerini çalışmalarının daha farklı bölümlerine ayırmaları için olanak sağladığı görülmektedir. Bu bakımdan MTB'nin müzik analizi alanında yapılan çalışmalara ciddi bir katkı sağlayacağını söylemek yanlış olmayacaktır.

MTB'de müzik analizinin MIDI verileri üzerinden gerçekleştirdiği göz önünde bulundurulduğunda, analizin başarılı olmasının müziği temsil eden MIDI verilerinin analizi yapılacak olan müziği birebir temsil etmesiyle doğru orantılı olduğu anlaşılmaktadır. Aksi taktirde yapılan analiz işlemleri ve elde edilen sonuçlar gerçeği yansıtmayacaktır.

Bu çalışmada, sestten MIDI'ye dönüşüm yapan yazılımların, analiz esnasında kullanılacak olan MIDI versiyonlarını hazırlamak için kullanılmasının sağlıklı bir yöntem olup olmadığı sorusuna da cevap aranmıştır. Tespit edilen yazılımlar, oluşturulan 4 ayrı kategoride test edilmiştir. Çıkan sonuçlar, F-Measure tekniği kullanılarak ölçülmüş ve yazılımların başarı sıralaması yapılmıştır.

Elde edilen sonuçlar incelendiğinde, yazılımların özellikle çoksesli müzikler üzerindeki performansının yeterli olmadığı görülmüştür. Teksesli müzikler üzerinde

kısmen daha doğru sonuçlar elde edildiği gözlemlense de ortaya çıkan MIDI dosyaları, müziği analiz işlemlerinde kullanılabilecek seviyede temsil etmediği anlaşılmaktadır.

Her ne kadar tek sesli müzikler üzerinde kısmen daha başarılı sonuçlar elde edilse de günümüzdeki müziklerin genellikle çok sesli olduğu gerçeği göz ardı edilmemelidir. Öte yandan, daha başarılı olmasına rağmen tek sesli müzikler üzerinde yapılan dönüştürme işlemi sonucunda elde edilen MIDI dosyalar da dönüştürülen müzikle bire bir uyumlu olmadığı anlaşılmaktadır. Yazılımların genel olarak performansları göz önünde bulundurulduğunda, analizi yapılacak müziklerin MIDI versiyonlarının hazırlanmasında dönüştürücü yazılımların kullanılmasının sağlıklı bir yöntem olmadığı anlaşılmaktadır.

Ancak diğer taraftan elde edilen sonuçlara göre tam bir MIDI dönüşüm olmamakla birlikte, günümüz şartlarında bu işi işe yarar bir şekilde başarabilen yazılımları da bu çalışmada ortaya çıkarmış olduk. Karşılaştırma sonuçlarına göre aslında yazılımlar içinde en başarılı sayılabilecek yazılımın WIDI olduğu ortaya çıkmış; bunu sınıflanan müzik havuzu yaratılarına göre sırasıyla Ableton Live ve Melodyne izlemiştir. Ancak yukarıda da belirtildiği gibi tam bir analiz için ses dosyasını MIDI verilerine çevirebilen sorunsuz bir yazılım henüz üretilmemiştir.

KAYNAKÇA

CHRISTENSEN, M. G. (2019). Introduction to Audio Processing, DOI: 10.1007/978-3-030-11781-8, SpringerLink Publ.

DURMAZ, Serhat (2000). MIDI, İzmir: Dokuz Eylül Yayınları.

DURMAZ, Serhat (2009). Müzik Teknolojisi ve Audio Terimleri Sözlüğü, İstanbul: Cinius Yayınları.

EEROLA, T., North, A. C. (2000). “Expectancy-based model of melodic complexity” (ed. Woods, C., Luck, G.B., Brochard, R., O'Neill, S. A., and Sloboda, J. A.), *Proceedings of the Sixth International Conference on Music Perception and Cognition*, Keele, Staffordshire, UK: Department of Psychology.

EEROLA, T., Toiviainen, P. (2004). “MIDI Toolbox: MATLAB Tools for Music Research”, University of Jyväskylä: Kopijyvä, Jyväskylä, Finland.

EEROLA, T., Toiviainen, P. (2004). “MIR in Matlab: The MIDI Toolbox”, *International Symposium of Music Information Retrieval (ISMIR)*,

İŞIKHAN, Cihan (2018). “Midi Dönüştürücü Yazılımların Başarı Karşılaştırması ve Matlab’de Müzik Analizi”, *Afyon Kocatepe Üniversitesi Akademik Müzik Araştırmaları Dergisi*, Yıl 4, sayı 8.1, Haziran 2018, ss. 1-10.

KRUMHANS, C. L., & Kessler, E. J. (1982). Tracing The Dynamic Changes in Perceived Tonal Organization in A Spatial Representation of Musical Keys. *Psychological Review*, V.89.4, pp.334- 368.

KRUMHANS, C. L., & Schmuckler, M. (1990). A Key-Finding Algorithm Based on Tonal Hierarchies, In: *Cognitive Foundations of Musical Pitch*, New York: Oxford University Press, pp.77-110.

MARSHALL, L., Frith, S. (2013). *Music and Copyright*, Routledge

McKay, C., Fujinaga, I. (2007). "Style-Independent Computer-Assisted Exploratory Analysis of Large Music Collections", *journal of interdisciplinary music studies*, vol:1.1, ss.63-66.

MULLER, M. (2015). "Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications", Switzerland: Springer International Publishing, ss. 1-36, 167-178.

SMITH, C. H., Georges, P. (2014). "Composer Similarities Through 'The Classical Music Navigator': Similarity Inference from Composer Influences", *Empirical Studies of the Arts*, vol: 32.2., ss.205-229.

SMITH, C. H., Georges, P. (2014). "Composer Similarities Through "The Classical Music Navigator": Similarity Inference from Composer Influences.", *Empirical Studies of the Arts*, vol: 32.2, ss. 205-229.

TENNEY, J., Polansky, L. (1980). "Temporal gestalt perception in music", *Journal of Music Theory*, vol:24.2, ss.205-41.

TYPKE, R., Wiering, F., Veltkamp, R. C. (2005, September). "A Survey of Music Information Retrieval Systems", In *Proc., 6th International Conference on Music Information Retrieval*, Queen Mary, University of London, ss.153-160.

VELARDO, V., Vallati, M., Jan, S. (2016). "Symbolic melodic similarity: State of the art and future challenges", *Computer Music Journal*, vol: 40.2, ss.70-83.

ÖZGEÇMİŞ

Orkun Murat YILDIRIM, 29/07/1989 tarihinde İstanbul'da doğmuştur. 2009 yılında Antalya Devlet Konservatuvarı Gitar Bölümüne başlamıştır. Müzik teknolojisine olan ilgisinden dolayı konservatuvar eğitimini yarıda bırakarak 2011 yılında Dokuz Eylül Üniversitesi Güzel Sanatlar Fakültesi Müzik Teknolojisi Bölümüne başlamıştır. 2017 yılında Dokuz Eylül Üniversitesi Güzel Sanatlar Enstitüsü Müzik Teknolojisi Bilim Dalı'nda yüksek lisans eğitimine başlamıştır.

2013 yılından itibaren çeşitli albüm çalışmalarında yer almış olup albüm ve performans alanında çalışmalarına devam etmektedir. Aktif müzisyenlik hayatının yanında *mastering* ve ses tasarımı alanında çalışmalar yapmaktadır.