ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

İNSAN HAREKETLERİNİN İNTERPOLASYON YÖNTEMLERİ İLE MODELLENMESİ

EGEMEN HALICI

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2019

Her hakkı saklıdır

TEZ ONAYI

Egemen HALICI tarafından hazırlanan "İnsan Hareketlerinin İnterpolasyon Yöntemleri İle Modellenmesi" adlı tez çalışması 27/ 05/ 2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Doç. Dr. Gazi Erkan BOSTANCI
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri:

Başkan: Doç. Dr. Mehmet Serdar GÜZEL
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Gazi Erkan BOSTANCI
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Dr. Öğr. Üyesi Mehmet DİKMEN
Başkent Üniversitesi Bilgisayar Mühendisliği Bölümü

Yukarıdaki sonucu onaylarım.

Prof. Dr. Özlem YILDIRIM
Enstitü Müdür Vekili

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

10.06.2019



Egemen HALICI

ÖZET

Yüksek Lisans Tezi

İNSAN HAREKETLERİNİN İNTERPOLASYON YÖNTEMLERİ İLE MODELLENMESİ

Egemen HALICI

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Gazi Erkan BOSTANCI

İnsan hareketleri ile eklemlerde meydana gelen konum değişikliklerinin interpolasyon yöntemleri ile modellenerek, sonrasında elde edilen modellerden geri elde edilmesi üzerine çalışılmıştır. Modellemeler esnasında Lagrange, Kübik Spline ve Spline interpolasyonlarından yararlanılmıştır. Modelleme esnasında tüm veri kümesinin kullanılması yerine yalnızca seçilen verimli veriler ile hesaplamalar yapılmıştır. Modellemeler yapıldıktan sonra interpolasyon hesaplamaları sırasında kullanılmayan verilerin geri elde edilmesine çalışılmıştır. İnsan hareketlerinin modellenmesi için gerekli veri kümeleri Microsoft Kinect sensörlerinden ve önceden kamera kaydı yapılması sonucu elde edilen görüntüler üzerinde OpenPose algoritmasının çalışmasıyla elde edilmiştir. İnsan hareketlerinin modellemesi üzerine yapılan çalışmalar bulunmakta olup, bu çalışmalardan farklı bir yöntem kullanılarak yenilikçi bir yaklaşım denenmiştir. Bu yöntemde düşük maliyet ve sistem gücü ile insan hareketlerinin öğrenilmesine çalışılmıştır. İnsan hareketlerinin modellenmesi sonucunda eklem konumlarının geri elde edilebilmesi için, her bir eklem konum değeri için bir bilinmeyenli polinom oluşturulmuştur. Modelleme sırasında, kullanılmayacak verilerinin belirlenmesi için aralık değerine ihtiyaç duyulmuştur. Aralık değeri arttıkça kullanılan veri sayısı azaldığı için aralık değerinin mümkün olan en yüksek değerde olması gerekmektedir. Ayrıca kullanılmayan veri kümelerinin geri elde edilebilmesi için hesaplanan polinomların derecesinin yüksek olması hesaplama süresini artırmaktadır. Hesaplama süresini en aza indirgemek için de grup değerine ihtiyaç duyulmuştur. Aralık ve grup değerlerinin belirlenebilmesi için, farklı aralık ve grup değerleri seçilerek meydana gelen hatalar hata kareler ortalaması ve hata mutlak ortalaması ile hesaplandıktan sonra en ideal değerlerin bulunmasına çalışılmıştır. İnterpolasyon yöntemleri ile iki farklı kişinin yaptığı dans hareketleri modellenmiş ve meydana gelen modeller birbirleri ile karşılaştırılarak ortaya çıkan

farklar hesaplanmıştır. Bu farklar sonucu bir kişinin diğer kişiye dans koreografisinin öğretilme düzeyi tahmin edilmiştir.

Mayıs 2019, 82 sayfa

Anahtar Kelimeler: İnterpolasyon yöntemleri, insan hareketlerinin modellenmesi, dinamik hareket ilkelleri, Microsoft Kinect, OpenPose

ABSTRACT

Ph.D Thesis

THE MODELING OF HUMAN BODY MOVEMENTS WITH INTERPOLATION METHODS

Egemen HALICI

Ankara University
Graduate School of Natural and Applied Science
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr Gazi Erkan BOSTANCI

In this thesis, interpolation methods are used for modelling of human movements unlike casual learning algorithms. There is a significant difference between Lagrange,Spline and Cubic Spline interpolations and it had found by Mc-Nemar statistical test.Results show that the Cubic Spline interpolation method is more efficient then the others.This modelling algorithm is memory-efficient. Instead of using whole dataset, used selected data which is the parts of dataset. OpenPose Algorithm and Microsoft Kinect device are used to create dataset for movements. For OpenPose algorithm, firstly recorded human movements and run that algorithm on the video. There are different methods for modelling human body movements but in this work tried novel method for this problem. This solution need low resource and run time to learn and model human body movements. To model human body movements and retrieve the joints' value, the algorithms developed here to create unique polynomial equations for each joint position values. In order to select data for modelling the movements, a step variable is needed. The step variable is important for dataset size so that the step variable should be as large as possible. Furthermore, the polynomial degree is important to retrieve unused data because of calculation time. To reduce polynomial degree, we used to group variable and calculating mean square error and mean abstract error to decide to value of step and group variables. Finally, the difference from compared datasets that modelling two different people is dance movements was calculated. That difference represent to learning rate of second dancer's learning success.

May 2019, 82 pages.

Key Words: Interpolations, human motion modelling, dynamic movement primitives, Microsoft Kinect, OpenPose

TEŞEKKÜR

Çalışmalarım süresince, engin bilgi ve tecrübelerini esirgemeden paylaşan, sabırla hatalarımı düzelten, akademik çalışmalarımın yanında sosyal ortamda da gelişimime katkıda bulunan danışman hocam sayın Doc. Dr. Gazi Erkan BOSTANCI'ya en içten duygularımla teşekkür ederim.

Tanıştığımız günden itibaren hiçbir zaman desteğini eksik etmeyen, tez çalışmam süresince sabırla, anlayışla yardımcı olan ve geri kalan hayatımızda çok daha başarılı çalışmalar yapacağımıza inandığım değerli eşim Dilara HALICI 'ya, beni bu günlere getiren, maddi manevi her konuda destekleyici olan annem Nevin HALICI 'ya, babam Fethi HALICI 'ya ve abim Ahmet Emre HALICI 'ya sonsuz teşekkürlerimi sunarım.

Egemen HALICI

Ankara, Mayıs 2019

İçindekiler

ETİK ..... **İ**

ÖZET ..... **İİ**

ABSTRACT ..... **İV**

SİMGELER VE KISALTMALAR DİZİNİ ..... **İX**

ÇİZELGELER DİZİNİ ..... **XV**

1. GİRİŞ **1**

2. KURAMSAL TEMELLER **5**

2.1 Dinamik Hareket İlkelleri (Dynamic Movement Primitives-DMP) **6**

2.2 İnterpolasyon **8**

2.2.1 Lagrange Interpolasyonu **9**

2.2.2 Doğrusal Spline İnterpolasyonu **11**

2.2.3 Kübik Spline İnterpolasyonu **12**

2.2.4 Hata kareler ortalaması hesaplamaları **13**

2.2.5 Mutlak hatalar ortalaması hesaplamaları **14**

2.3 Makine Öğrenmesi **14**

2.3.1 Destek vektör makinesi (Support Vector Machine-SVM) **14**

2.3.2 Policy learning by weighting exploration with the return (POWER) **15**

2.3.3 Denetimsiz öğrenme (Unsupervised learning) **15**

3. MATERYAL VE YÖNTEMLER **19**

3.1 Materyal **19**

3.1.1 JSON veri formatı **22**

3.1.2 DLL (Dynamic Link Library) dosya formatı **23**

3.2 Yöntemler **24**

3.2.1 Veri Kümesi **25**

3.2.1.1 OpenPose ile veri kümesinin elde edilişi **25**

3.2.1.2. Kinect sensörü ile veri kümesinin elde edilişi **29**

3.2 Veri Kümesinin Hazırlanması **33**

3.4 Önerilen Yöntem **34**

3.5 MATLAB ile İnterpolasyon Hesaplamalarının Yapılması **36**

3.6 Veri Kümelerinin Normalize Edilmesi **36**

3.7 Veri Kümelerinin Karşılaştırılması 37
4. ARAŞTIRMA BULGULARI 39
5.SONUÇ VE TARTIŞMA 73
5.1 Tezin Amacı 73
5.2 Tezin Sonucu 73
5.3 McNemar Testi ile anlamlılık testi 74
5.4 Gelecek Çalışmalar 78
KAYNAKLAR 80
ÖZGEÇMİŞ 83

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

DMP	Dinamik Hareket İlkelleri
HKO	Hata Kareler Ortalaması
MHO	Mutlak Hatalar Ortalaması
SVM	Destekçi Vektör Makinası
POWER	Policy Learning by Weighting Exploration with the Return
JSON	JavaScript Object Notation
DLL	Dynamic Link Library
SDK	Software Development Kit
GPU	Graphics Processing Unit
RGB Kamera	Red Green Blue Kamera
FPS	Frame Per Second

Simgeler

q	Sınıflandırma Katsayısı
k(.,.)	Kernel Fonksiyonu
V1	Kinect for Windows User(Versiyon 1)
VK1	Veri Kümesi 1
VK2	Veri Kümesi 2
VK3	Veri Kümesi 3
VK4	Veri Kümesi 4
VK5	Veri Kümesi 5
Q_r	Serbest Zamandaki Eklem Derecesi
Q_O	Kasılı Zamandaki Eklem Derecesi
GB	Gigabyte
RAM	Random Access Memory (Rastgele Erişimli Bellek)
GHz	Giga Hertz
Cm	Santimetre
MP	Megapixel

ŞEKİLLER DİZİNİ

Şekil 1.1 Geleneksel Sistemler İle Microsoft Kinect Sensörünün Karşılaştırması 3
Şekil 2.1 Bantların Alkışlanması Dansı. 6
Şekil 2.2 Dinamik Hareket İlkellerinin Kontrol Mekanizma Şeması. 7
Şekil 2.3 Lagrange İnterpolasyonu 10
Şekil 2.4 Lineer Spline İnterpolasyonu 11
Şekil 2.5 Kübik Spline İnterpolasyonu 13
Şekil 2.6 İnsan Eklemlerinin Üçgenler İle Modellenmesi 16
Şekil 2.7 Modelleme Örnekleri 17
Şekil 3.1 OpenPose Yüz ve El Takibi 20
Şekil 3.2 OpenPose İskelet Takibi 20
Şekil 3.3 Kinect Cihazını Üzerinde Bulunan Kamera Sensörleri 21
Şekil 3.4 JSON Key-Value Nesne Yapısı 22
Şekil 3.5 JSON Liste Nesne Yapısı 22
Şekil 3.6 Openpose İle İnsan Hareketlerinin Takibi 26
Şekil 3.7 Openpose İle Takip Yapılan İskelet Sistemi Noktasal Eklemleri 28
Şekil 3.8 Kinect Sensörleri İle Takip Edilen Eklemlerin İskelet Sistem Konumları 30
Şekil 3.9 İskelet Takibinde Kinect Sensörü Üzerindeki Kamera Görüntüleri 32
Şekil 3.10 Kinect Depth Kamera İle İnsan İskelet Takibi 32
Şekil 4.1 Openpose Veri İşleme Aşamasındaki Görüntüleri 40
Şekil 4.2 Kinect Derinlik Kamerası - VK1 veri kümesi için Lagrange interpolasyon ile yöntemi modelleme sonucu 42
Şekil 4.3 Openpose - VK1 veri kümesi için Lagrange interpolasyon yöntemi ile modelleme sonucu 42
Şekil 4.4 Kinect Derinlik Kamerası - VK1 veri kümesi için Spline interpolasyon yöntemi ile modelleme sonucu 43
Şekil 4.5 Openpose - VK1 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu 43
Şekil 4.6 Kinect Derinlik Kamerası - Vk1 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu 44

Şekil 4.7 Openpose - VK1 Veri Kümesi İçin Kübik Spline İnterpolasyon Yöntemi İle Modellenmesi Sonucu 44
Şekil 4.8 Kinect Derinlik Kamerası - VK1 Veri Kümesi İçin Grup Sayısı 4 Aralık Değeri 6 İken Tüm İnterpolasyon Yöntemleri İle Elde Edilen Modeller 45
Şekil 4.9 OpenPose - VK1 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modeller 45
Şekil 4.10 Openpose – VK1 veri seti için grup sayısı 4 aralık değeri 2 iken tüm interpolasyon yöntemleri ile elde edilen modeller 48
Şekil 4.11 Kinect Derinlik Kamerası – VK1 veri seti için grup sayısı 4 aralık değeri 2 iken tüm interpolasyon yöntemleri ile elde edilen modeller 48
Şekil 4.12 Kinect Derinlik Kamerası - VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modeller 49
Şekil 4.13 Openpose - VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modeller 51
Şekil 4.14 Kinect Derinlik Kamerası - VK2 veri seti için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 52
Şekil 4.15 Openpose - VK2 veri kümesi için lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 52
Şekil 4.16 Kinect Derinlik Kamerası - VK2 veri seti için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 53
Şekil 4.17 Openpose - VK2 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 53
Şekil 4.18 Kinect Derinlik Kamerası - VK2 veri seti için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 54
Şekil 4.19 Kinect Derinlik Kamerası - VK2 veri seti için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri seti ile karşılaştırılması. 55
Şekil 4.20 Openpose - VK2 veri seti için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri seti ile karşılaştırılması. 56
Şekil 4.21 Kinect Derinlik Kamerası - VK3 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları 58

Şekil 4.22 Openpose - VK3 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa Ekleminin Hata Sonuçları 58
Şekil 4.23 Kinect Derinlik Kamerası - VK3 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları..... 59
Şekil 4.24 Openpose - VK3 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları..... 59
Şekil 4.25 Kinect Derinlik Kamerası - VK3 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları..... 60
Şekil 4.26 Openpose -VK3 veri kümesi için Kübik Spline İnterpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları..... 60
Şekil 4.27 Kinect Derinlik Kamerası - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri kümesi ile karşılaştırılması. 62
Şekil 4.28 Openpose - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklem verilerinin orijinal veri kümesi ile karşılaştırılması 63
Şekil 4.29 Kinect Derinlik Kamerası - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sol-omuz eklem verilerinin orijinal veri kümesi ile karşılaştırılması..... 63
Şekil 4.30 Openpose - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sol-omuz eklem verilerinin orijinal veri kümesi ile karşılaştırılması. 64
Şekil 4.31 Kinect ile elde edilen VK4 veri kümelesinin Sağ-El eklem değerleri..... 65
Şekil 4.32 OpenPose ile elde edilen VK4 veri kümesinin Sağ-El eklem değerleri 66
Şekil 4.33 Kinect ile elde edilen VK4 veri kümesinin Sol-El eklem değerleri..... 66
Şekil 4.34 OpenPose ile elde edilen VK4 veri kümesinin Sol El eklem değerleri 67
Şekil 4.35 Kinect - VK5 veri kümesi için normalizasyon uygulanmadan önce Kalça-Merkez eklem verilerinin karşılaştrılması..... 69
Şekil 4.36 Kinect - VK5 veri kümesine normalizasyon uygulandıktan sonra Kalça-Merkez eklem değerlerinin gösterimi. 69
Şekil 4.37 OpenPose - VK5 veri kümesine normalizasyon uygulanmadan önceki 1. ve 2. kişinin Kalça Merkez eklem değerlerinin gösterimi. 70

Şekil 4.38 OpenPose - VK5 veri kümesine normalizasyon uygulandıktan sonra 1. ve 2. kişinin Kalça Merkez eklem değerlerinin gösterimi. 70

ÇİZELGELER DİZİNİ

Çizelge 3.1 Openpose İle Elde Edilen Eklem Konum Değerleri Veri Kümesi 27
Çizelge 3.2 Microsoft Kinect İle Elde Edilen Eklem Konum Değerleri Veri Kümesi ...30
Çizelge 3.3 DMP ve İnterpolasyon Yöntemlerinde Kullanılan Veri Kümesi Boyutları.35
Çizelge 3.3 DMP ve İnterpolasyon Yöntemlerinde Kullanılan Veri Kümesi Boyutları (Devamı) 34
Çizelge 4.1 Kinect sensörleri ile aralık değeri 6 grup değeri 4 iken modellenen VK1...46
Çizelge 4.2 OpenPose ile aralık değeri 6 grup değeri 4 iken modellenen VK1.....47
Çizelge 4.3 Kinect sensörleri ile aralık değeri 6 grup değeri 4 iken modellenen VK2 için 40 ile 47 kare numaraları arasındaki geri elde edilen Kafa eklem değerleri.....56
Çizelge 5.1 İnterpolasyon yöntemleri karşılaştırması için yapılan varyans analiz tablosu 74
Çizelge 5.2 Kinect - VK3 veri kümesinin modellenmesinde interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları.....75
Çizelge 5.3 Kinect - VK2 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları 76
Çizelge 5.4 Kinect - VK1 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları 76
Çizelge 5.5 OpenPose - VK3 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları 77
Çizelge 5.6 OpenPose - VK2 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları 77
Çizelge 5.7 OpenPose - VK1 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları 78

1. GİRİŞ

İnsan gücünün yetersiz kaldığı operasyonlarda, sağlığa zararlı ortamlarda ve zaman tasarrufu gerektiren çalışmalarda robotik sistemler sık sık kullanılmaktadır. Ayrıca durmaksızın çalışabilen ve belli bir iş koluna özel geliştirilen sistemler endüstride oldukça fayda sağlamaktadır. Otonom sistemler 18.yy'dan itibaren ilk olarak tekstil endüstrisinde kullanılmaya başlanmış ve devamında diğer sektörlerde yerlerini almışlardır.

Robot kelimesi ilk olarak Çekoslovak yazar Karel Čapek tarafından yazılan "Universal Robots" adlı tiyatro oyununda kullanılmıştır. Genel anlamda robot kelimesi otonom ya da önceden programlanmış elektronik cihazlar olarak tanımlanmıştır.

Günümüzde gelişen teknoloji ile insansı robot geliştirmeleri başlamıştır (Inoue et al., 2000). İnsansı hareket kabiliyetlerini robotlara önceden tanımlamak oldukça masraflı, hataya meyilli ve ciddi miktarda zaman gerektiren bir işlemdi. Ancak insan hareketlerinin modellenerek insansı robotlara aktarılabilmesi Dinamik Hareket İlkelleri (DMP) ile mümkün hale gelmiştir.

İnsan hareketleri, eklem konumlarının değişmesi sonucu meydana gelir. Hareket sırasında uzuvların meydana getirdiği değişimler ile eklemlerin hızı, pozisyonu ve ivmesi hesaplanabilir. Bu değişimler takip edilerek veri kümeleri elde edilir.

Dinamik Hareket İlkelleri (DMP) insan iskeletinin meydana getirdiği şekillerin matematiksel olarak ifade edilme şeklidir (Prada, Remazeilles, Koene, & Endo, 2013). DMP ile insan hareketleri takip edilerek elde edilen veri kümeleri ile iskelet sisteminin modellemesi yapılabilmektedir (Ben Amor, Neumann, Kamthe, Kroemer, & Peters, 2014). Bu modellerin robotlara aktarılması ile herhangi bir ön hazırlık gerekmeden insan hareketleri insansı robotlar tarafından yapılabilmektedir (Inoue et al., 2000).

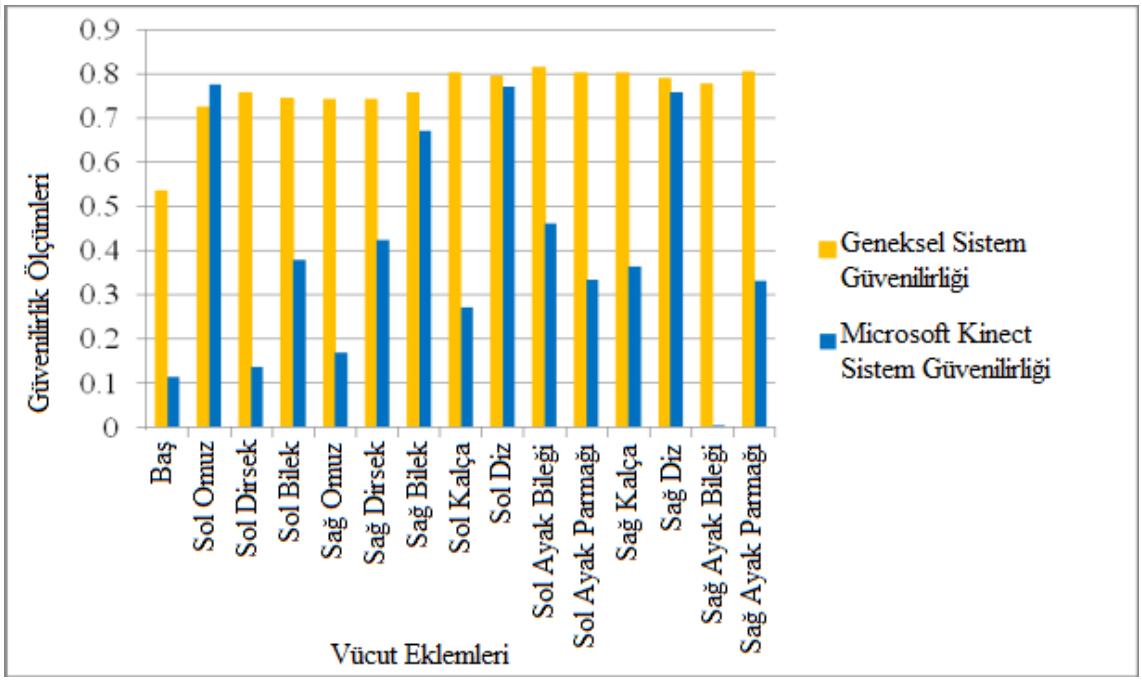
DMP ile insan hareketlerinin sayısallaştırılması esnasında elde edilmiş olan veri kümelerinin tamamı kullanılır. Tüm veri kümesinin DMP algoritmaları kullanımı esnasında yüksek bilgisayar gücüne ihtiyaç duyulur. Bugüne kadar yapılan insan hareketleri modellenmesi için birçok yöntem bulunmaktadır. Bu çalışmalarda tüm veri kümesinin tamamı kullanılarak modellemelerin yapıldığı görülmektedir (Joo, Simon, & Sheikh, 2018).

POWER, Destekçi Vektör Makinesi, Denetimsiz Makine Öğrenmesi algoritmaları kullanılan algoritmalardan birkaçıdır. Ayrıca Destekçi Vektör Makinesi, Denetimsiz Makine Öğrenmesi algoritmaları ile hareketlerin sınıflandırılması da yapılabilmektedir.
OpenPose ve Microsoft Kinect cihazı belli mesafedeki insan iskeletini tahmin ederek veri kümeleri oluşturmaktadır (Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, 2017; Kar, 2010). Geleneksel insan iskelet sistemi takip sistemlerinde ise, eklemler üzerine yansımalı işaretleyici takılarak en az iki farklı kamera ile veri kümeleri oluşturulmaktadır. Ortaya çıkan maliyetleri ve sistem gereksinimlerini azaltabilmek için Microsoft Kinect cihazı ve OpenPose algoritması tercih edilmiştir. Microsoft Kinect cihazı ile OpenPose algoritmaları karşılaştırılarak farklılıkları araştırılmıştır (Miguel Arduengo Sven Jens Jorgensen Supervisors Kimberly Hambuchen Luis Sentis Francesc Moreno Guillem Alenyà, 2017).

Bu tez çalışmasında, DMP algoritmalarından farklı olarak tüm veri setinin kullanılması yerine seçili veriler kullanılarak insan hareketlerinin modellenmesi hedeflenmiştir. Bu sayede daha düşük bilgisayar gücüyle hareketlerin modellenmesi sağlanmıştır.
OpenPose algoritması ve Microsoft Kinect cihazı ile insan hareketleri takip edilerek veri kümeleri oluşturulmuştur. Elde edilen veri kümelerini modellemek için üç farklı interpolasyon yöntemi kullanılmıştır. Elde edilen modellemeler gerçek veriler ile karşılaştırılarak en etkili interpolasyon yöntemine karar verilmiştir.
OpenPose algoritmasının ve Kinect cihazının çalışma prensipleri benzer olsa da modellenen hareketin aynı olmasına rağmen farklı veri kümeleri oluşturabilmektedirler. Yapılan karşılaştırmalarda bu farklar görülmektedir. Bu farkların oluşma sebebi genellikle eklemlerin üst üste gelmesidir. Veri kümeleri hazırlanırken meydana gelen bu

eksik veriler bu çalışmada geliştirilen algoritma ile de tamamlanabileceği düşünülmektedir.

Şekil 1.1 de geleneksel insan iskelet takip sistemleri ile Microsoft Kinect takip sistemleri baş, omuz, dirsek, bilek, kalça ve diz eklemleri üzerinden elde edilen verilerin karşılaştırılması yapılmıştır. Yapılan bu çalışmada geleneksel sistemlerin daha başarılı olduğu gösterilmektedir. Ancak Microsoft Kinect sisteminin tahmin sonuçları da geleneksel sistemlere çok yakın sonuçlar verdiği ve başarılı sonuçlar elde edildiği bildirilmektedir (Zerpa, Lees, Patel, & Pryzsucha, 2015).



Şekil 1.1 Geleneksel Sistemler İle Microsoft Kinect Sensörünün Karşılaştırması

Veri Güvenirliliği arasındaki farklar gösterilmektedir (Zerpa, Lees, Patel, & Pryzsucha, 2015).

Bu tez çalışmasında insan hareketlerinin interpolasyon yöntemleri ile modellenmesi hedeflenmektedir. İnterpolasyon yöntemleri ile yapılan modellemelerde tüm veri kümesinin kullanılması yerine daha etkili ve uygun veriler seçilmekte ve seçilmiş olan veriler ile modellemeler yapılabilmektedir. Böylece daha az sistem gücü

kullanılacağından daha etkili bir modelleme yönteminin geliştirilebileceği düşünülmektedir.

Bu yöntem ile iki farklı kişinin yaptığı aynı hareketler modellenerek karşılaştırılmıştır. Karşılaştırma sonucu yapılan hareketler arasında meydana gelen farklılıklar görülebilmektedir. Yapılan hareketler arasındaki farklılıklar ile skor değeri hesaplanmıştır.

Bunun yanı sıra, tez çalışmasındaki veri kümeleri insan hayatının bir parçası olan, tarihte oldukça eskilere dayanan dans figürlerinden elde edilmiştir. Yaşamın her döneminde var olan dansın modellenerek, gelecek nesillere aktarılması mümkün hale getirilmesi de hedeflenmektedir. Bunlara ek olarak modellenen dansların robotlara aktarılması ve robotların insan hareketleri meydana getirmesi sağlanabilecektir. Ayrıca dans eğitimlerinde, eğitici ve öğrencinin ayrı zamanlarda yaptıkları dans hareketleri karşılaştırılarak eğitim sonucunda öğrencinin başarısı hakkında yorum yapılabilecektir.
Polinomların oluşturulması esnasında Lagrange, Spline ve Kübik Spline interpolasyonları kullanılmıştır ve 2.bölümde kullanılan interpolasyonlar anlatılmıştır. Ayrıca tez çalışmasının akışı içerisindeki tüm konular sıralı bir şekilde 2. bölümde tanımlanmıştır. 3. bölümde kullanılan yöntemlerden bahsedilmiştir ve 4. bölümde ise modellerden geri elde edilen hareketlerin başarısı incelenerek hata oranları hesaplanmıştır. Meydana gelen hatalar Hata Kareler Ortalaması (HKO) ve Mutlak Hatalar Ortalaması (MHO) ile hesaplanmıştır. Yapılan deneylerde meydana gelen hata değerlerini düşürülerek daha başarılı modellemeler yapılması hedeflenmiştir. Hesaplanan hata değerleri de ANOVA ve McNemar istatistik testi ile analiz edilmiştir. 5. Bölümde ise elde edilen sonuçlar incelenerek modellemelerin başarısı tartışılmıştır.

2. KURAMSAL TEMELLER

Hareket, tüm canlıların temel gereksinimlerindendir. Bir insanın gerçekleştirdiği iş, yaptığı hareketlerin bütününden oluşmaktadır. Yaşam içerisinde insan kabiliyetlerinin yetmediği ya da vücut yapısının uygun olmadığı iş tanımları için tarih boyunca makineler geliştirilerek çözümler üretilmiştir.

Günümüzde gelişen teknoloji ile birlikte üretilen makine ve robotlar hayatın birçok alanında yaşamımızı kolaylaştırmayı amaçlamaktadır. Bu amaç doğrultusunda da insanların hareketlerini taklit ederek ya da modelleyerek makinaların işlevselliğini artırmak son yılların en popüler teknolojileri arasında yer almaktadır.

İnsan hareketlerinin modellenerek gelecek nesillere aktarılmasını hedefleyen ve kültürlerin devam etmesini amaçlayan birçok çalışmada dans figürleri modelleyen çalışmalar yapılmıştır (Alexiadis et al., 2011a). Danslar tarihi oldukça eski çağlara dayanmaktadır. İlk dans figürlerinin izlerine şekil 2.1 de görüldüğü üzere milattan önce 3300'lerde rastlanmıştır. Mısırlıların duvarlara çizdikleri resimlerden dansın Mısırlılara dayandığını ve insanlık tarihinin tümünde bulunduğu bilinmektedir.



Şekil 2.1 Bantların Alkışlanması Dansı.

Mısırlılar, Ur-ari-en-Ptah tapınağı, 6. Hanedan, Yaklaşık olarak M.Ö. 3300 (British Museum.)

Bu tez çalışmasında insan tarihinde oldukça önemli olan dans hareketlerinin modellenebilmesi ve gelecek nesillere aktarılması mümkün hale getirilebileceği düşünülmektedir.Buna ek olarak modellenen dansların robotlara aktarılarak robotların dans figürlerini meydana getirmesi hedeflenmektedir. Dans hareketlerinin doğru bir şekilde modellenebilmesi için yapılan harekete bağlı olarak çeşitli algılama teknolojileri mevcuttur. (Alexiadis et al., 2011b)

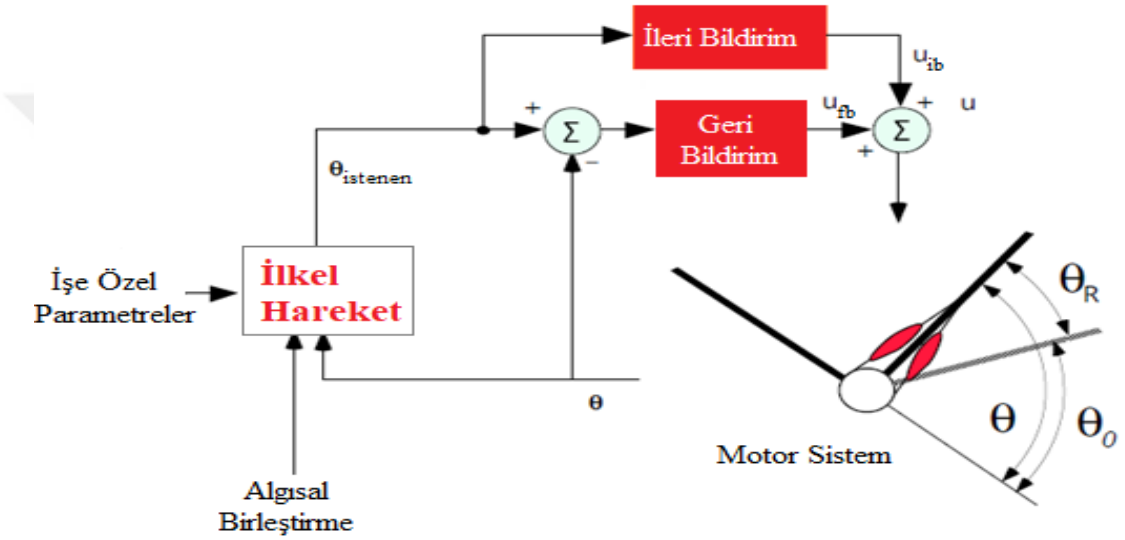
2.1 Dinamik Hareket İlkelleri (Dynamic Movement Primitives-DMP)

Dinamik hareket ilkelleri genellikle robotik sistemlerin fiziksel hareketleri yapabilmesi için kullanılır. Bu yöntem ile yapılabilirliği zor olan karmaşık eklem motor hareketlerinin herhangi bir parametreye bağlı olmadan modellenmesi hedeflenmektedir (Li, Zhang, & Liu, 2010)

DMP'nin kontrol mekanizması şekil 2.2 de gösterilmektedir. DMP ile insan iskelet sisteminde bulunan eklem motor hareketleri matematiksel işlemler ile

modellenmektedir. Θ_R eklemlerin serbest zamandaki derecesini, Θ_O ise eklemlerin kasılma anındaki derecesini belirtir. Yapılan hesaplamalar sonucunda sistem tarafından yapılan geri bildirimler ile optimal değerlere ulaşılır (Prada et al., 2013b).

Eklem motor hareketleri esnasında ivme, yön ve hız parametreleri değişimler gösterir ve bu değişimler hareketlerin meydana gelmesini sağlar.



Şekil 2.2 Dinamik Hareket İlkellerinin Kontrol Mekanizma Şeması.

(Prada et al., 2013)

DMP modellemeleri esnasında zaman değişkeninden bağımsız olarak, ivme yön ve hız parametreleri kullanılarak doğal insansı hareketleri robotik sistemlere aktarılabilir.Parça parça modellenen insan hareketleri birleştirilip, bütün bir iskelet sistemi hareketi elde edilerek robotik uygulamalara aktarılabilir (Ben Amor, Neumann, Kamthe, Kroemer, & Peters, 2014; Prada, Remazeilles, Koene, & Endo, 2013a; Vuga et al., 2013).

2.2 İnterpolasyon

Yapılan deneysel çalışmalar sonucu elde edilen veriler kullanılarak, değeri bilinmeyen ve elde edilen veri kümesi arasında farklı konumlu noktaların değerini tahmin etmeyi amaçlayan yöntemlere interpolasyon denir (Newman & Davis, 2006).

İnterpolasyonlar genel olarak, elde edilmiş verilerin bir fonksiyon eğrisine uyarlanması amacıyla kullanılmaktadır. Değeri bilinmeyen noktalar, fonksiyon eğrisinden yararlanılarak tahmin edilebilir. Bir fonksiyonun sonlu sayıdaki $x_0, x_1 \mathrm{K}\, , x_n \in \mathrm{R}$ noktalarında aldığı $f(x_0), f(x_1), \ldots, f(x_n)$ değerleri bilindiğinde (fonksiyonun kendisi bilinmiyor), bu noktalardan geçen *n.* dereceden bir tek $P_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ polinomu vardır ($i = 0,1,2\ldots,n$ için $P_n(x_i) = f(x_i)$). $P_n(x)$ polinomu elde edilip bir x noktasındaki $f(x)$ değerinin yerine $P_n(x)$ alınırsa, bilinmeyen $f(x)$ değeri yaklaşık olarak hesaplanmış olur (Berrut & Trefethen, 2004).

İnterpolasyon yöntemleri kullanılırken büyük veri kümeleri yerine veri kümelerini parçalayarak elde edilen küçük parçalardan elde edilen polinomlar daha az hataya sahip tahminlerde bulunabilmektedir. Bu sebeple veri kümeleri parçalara ayrılırken grup ve adım değişkenleri kullanılmıştır. Grup değişkeni bir veri parçasında kaç adet verinin bulunacağını, adım değişkeni ise veri kümesi içerisinde bulunan karelerin hangilerinin kullanılacağını belirtir. Adım değişkeni ile tüm veri kümesinin kullanımı yerine en az miktarda veri kullanımı ile modelleme yapılabilir.

Modelleme yapılırken, adım değişkeninden dolayı modelleme sırasında kullanılmayan veriler interpolasyon yöntemleri ile geri tahmin edilebilir. Geri elde edilen veriler orijinal veriler ile karşılaştırıldığında, interpolasyon yöntemleri ile tahmin edilen verilerde sapmalar görülmektedir. Bu sapmaların HKO ve MHO değerleri hesaplanarak, oluşturulan modellerin doğrulukları hesaplanmaktadır.

Elde edilen HKO ve MHO değerlerini minimize eden grup ve aralık değeri hesaplanarak veri kümesinin modellenmesi minimum sapma ile gerçekleştirilebilmektedir.

Tez çalışmasında Langrange interpolasyonu, Lineer Spline interpolasyonu ve Kübik Spline interpolasyonu kullanılmıştır.

2.2.1 Lagrange Interpolasyonu

Lagrange interpolasyonu, x_i *(i=0,1,...N)* noktalarında bilinen f_i *(i=0,1,...N)* veri noktaları için uygun eğri oluşturulması işlemidir. Eğer x_i'ler çerçeve numarası ve x_j'ler de çerçeve numarasına göre vücut noktalarının konumu olarak düşünülürse, konumların geri elde edilebilmesi için gerekli olan fonksiyonlar bu iki değişken ile oluşturulabilir. İnterpolasyon için kullanılan ifade $i=0,1,\ldots,n$ olmak üzere:

$$P_{n(x)} = \sum_{i=0}^{n} \left(\prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{(x - xj)}{xi - xj} \right) f(x)$$

polinomu elde edilir. Elde edilen polinomun derecesi kullanılan veri kümesinin uzunluğunun bir eksiği (n-1) olur (Berrut & Trefethen, 2004).

Şekil 2.3 de Lagrange interpolasyonu ile türetilmiş olan eğri ile orijinal eğri arasında meydana gelen farklılıklar görülmektedir.

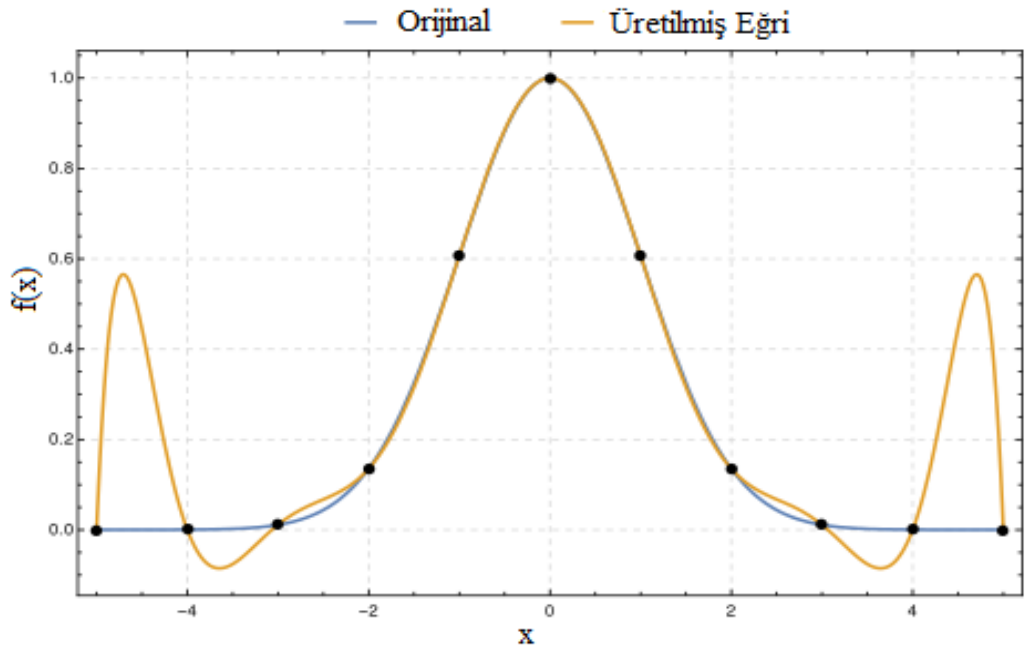
Büyük veri kümelerinde Lagrange interpolasyonu ile elde edilen polinomlarda meydana gelen sapmalar oldukça büyük miktarlarda olmaktadır. Ayrıca veri kümesinde bulunan noktalar arasındaki mesafeler büyüdükçe meydana gelen sapmalar da büyümektedir.

Bu sapmaları meydana getiren sebepler,

a) Her bir p(x) denkleminin çözümü için $O(n^2)$ süresinde toplama ve çarpma işleminin gerekli olması.

b) Sonradan eklenen her bir veri kümesi (x_{n+1}, f_{n+1}) için sıfırdan hesaplamaya ihtiyaç duyulması.

c) Hesaplamaların sayısal olarak değişken olması şeklinde listelenebilir.



Şekil 2.3 Lagrange İnterpolasyonu

Belirli noktalardan geçen, Lagrange interpolasyonu ile üretilmiş eğri örneği bulunmaktadır.

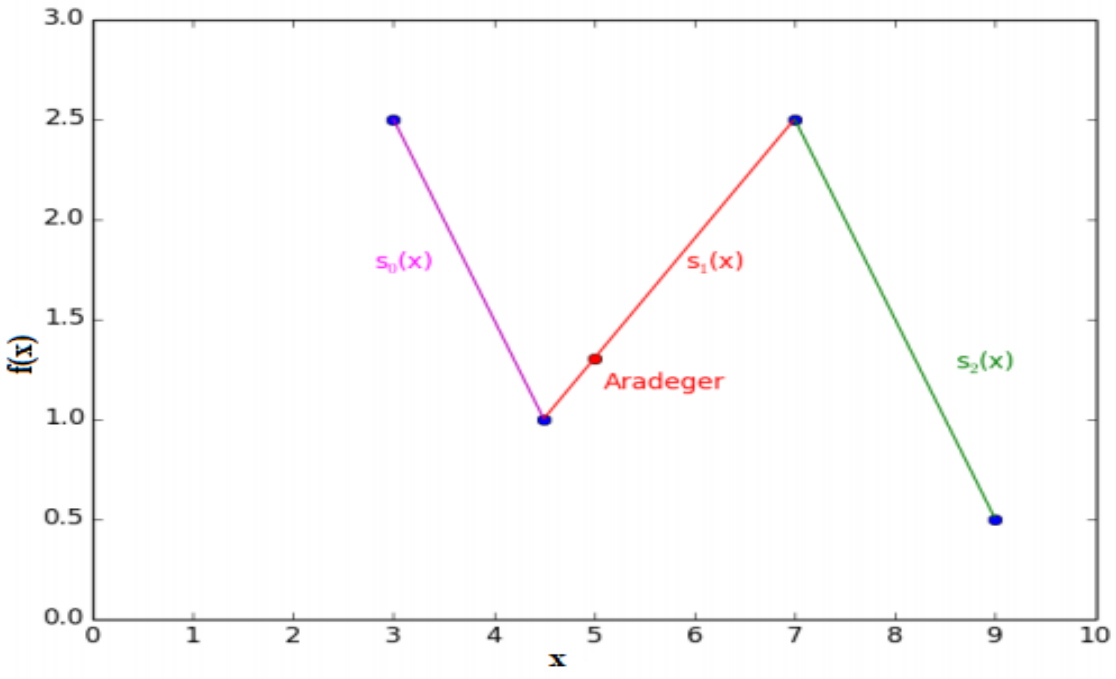
2.2.2 Doğrusal Spline İnterpolasyonu

Spline interpolasyonu istenilen verinin değerini bulabilmek için, bu değerin sağındaki ve solundaki noktalardan geçen eğriler oluşturur ve oluşan bu eğrilere göre istenen x değeri için y değerini verir. Veri kümesinde buluna ikili veri kümelerini kullanarak farklı bir denklem belirlenmelidir. Herhangi (x_i, f_i), (x_{i+1}, f_{i+1}) ikili veri kümesi için

$$s_1(x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

denklemi elde edilir.(Lee, 1983)

Şekil 2.4 de verilen noktalar arasındaki ara değerler Lineer Spline interpolasyon ile hesaplanarak, tahmini ara değer oluşturulan grafik üzerinde belirtilmektedir.



Şekil 2.4 Lineer Spline İnterpolasyonu

Lineer Spline interpolasyonu ile verilen noktalarden (mavi noktalar) ara değeri (kırmızı) hesaplanan örnek eğri mevcuttur.

Spline interpolasyonları ile kullanılmak istenen veri kümesindeki tüm noktalardan geçen ve küçük dereceli polinomlara uygun olarak düzgün değişen fonksiyonlarla, sürekli bir benzetme elde etmek ve bu şekilde veri kümesinde bulunmayan ara değer hesabı yapmak mümkündür.

2.2.3 Kübik Spline İnterpolasyonu

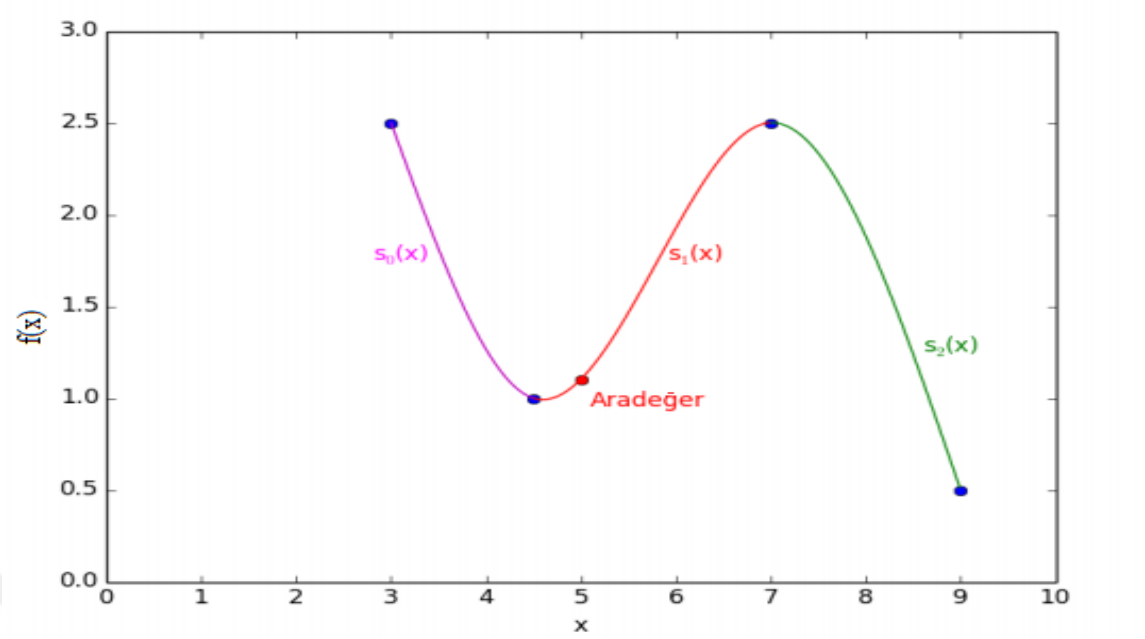
Kübik Spline interpolasyonu, eldeki noktalar için her aralığa üçüncü dereceden bir polinom uyarlamasıyla elde edilir (Mckinley & Levine, 1998).
Verilen iki farklı noktadan sonsuz sayıda polinom geçebileceği için elde edilen polinomları tek bir polinoma dönüştürebilmek için bazı kurallar gereklidir.
Kübik polinomların yapısı;

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

olmalıdır.

Her n tane nokta için (n-1) tane çizgi olacak, her bir nokta aralığında bir tane Kübik polinomu olacağından (n-1) tane $s_i(x)$ adet polinom olacaktır. Bu polinomları elde etmek için, her bir polinom 4 farklı katsayı (a_i, b_i, c_i, d_i) içerdiğinden 4(n-1) tane katsayı belirlenmelidir. Bu işlem için de 4(n-1) adet denkleme ihtiyaç vardır ki bu denklemler ile katsayılar hesaplanarak, her aralık için üçüncü dereceden bir polinom bulunmuş olacaktır.

Kübik polinom yapısı üzerinden hesaplanan polinomlardan elde edilen grafik Şekil 2.5' de gösterilmektedir. Elde edilen polinom ile verilen noktalar arasındaki ara değer tahmin sonucu belirtilmektedir.



Şekil 2.5 Kübik Spline İnterpolasyonu

Kübik Spline interpolasyonu ile verilen noktalardan (mavi noktalar) ara değeri (kırmızı) hesaplanan örnek eğri mevcuttur.

Oluşturulan Lagrange, Lineer Spline ve Kübik Spline polinomları arasındaki sapmaları tespit edebilmek ve hata oranlarının hesaplanması amacıyla HKO ve MHO testleri yapılmıştır.

2.2.4 Hata kareler ortalaması hesaplamaları

Hata kareler ortalaması (HKO), regresyon çizgisinin veri noktalarına olan uzaklıklarını gösterir. Burada bahsedilen uzaklıklar meydana gelen hesaplama hatalarıdır. Bu hesaplama sırasında uzaklıkların karelerini hesaplayarak negatif işaretler ortadan kaldırılır ve hataların ortalaması bulunur (Schluchter, 2014).

$$\text{HKO}= \frac{1}{n}\sum_{j=1}^{n}\left(y_j - \acute{y}_j\right)^2$$

2.2.5 Mutlak hatalar ortalaması hesaplamaları

Mutlak Hatalar Ortalaması (MHO), HKO'sına benzer şekilde hataları hesaplamak için kullanılan farklı bir yöntemdir. Bu yöntemde negatif değerleri ortadan kaldırmak için veri noktalarına olan uzaklıklar mutlak değer fonksiyonu kullanılır. Uzaklık değerlerinin ortalaması alınarak hata sonuçları hesaplanmış olur (Chai & Draxler, 2014).

$$MHO = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \acute{y}_j\right|$$

2.3 Makine Öğrenmesi

İnsan hareketlerinin modellenmesi için çeşitli algoritmalar bulunmaktadır. Support Vector Machine, Policy Learning by Weighting Exploration with the Return, Unsupervised Learning bunlardan birkaçıdır.

Bu öğrenme algoritmaları, hareketlerin modellenmesi sırasında oluşan tüm veri kümesini kullanmaktadırlar. Bu yüzden bu algoritmalar fazla kaynak kullanımına sebep olmaktadır. Ayrıca bu algoritmalarda optimal modelleme için çeşitli sayılarda denemelerin yapılması gerekmektedir ve bu durum her bir hareket çeşidinin sınıflandırılması için gerekli olduğundan zaman maliyeti getirmektedir. (Ben Amor et al., 2014; Williams, Blake, & Cipolla, 2003)

2.3.1 Destek vektör makinesi (Support Vector Machine-SVM)

İnsan hareketlerinin modellenmesinde Destekçi Vektör Makinesi sık sık tercih edilen algoritmalardan biridir. Özellikle SVM araçları denetimli öğrenme sınıflandırma

sorunlarının çözümlenmesi için tasarlanmıştır. Denetimli öğrenme sınıflandırma ve regresyon problemleri olarak gruplandırılır. (Bhavsar & Panchal, 2012; Cristianini & Shawe-Taylor, 2000; Hare et al., 2016; P. W. Wang & Lin, 2014)

f(q) $=\sum_{i=1}^{N} yi\alpha ik(q, xi) + b$ $x_{i,}$ veri kümesi i ∈ [1, N] aralığındaki vektörlerdir. q sınıflandırma katsayısını tanımlamaktadır. $k(.,.)$ ise kernel fonksiyonudur. Bu sınıflandırma ile meydana gelen hareketlerin birbirlerine olan komşulukları hakkında karar verilebilmektedir. Bu şekilde hareketler sınıflandırılarak öğrenme işlemi gerçekleştirilmektedir. SVM'nin etkili bir algoritma olmasının yanı sıra uygulanabilirliği de oldukça basittir. Fakat algoritma çalıştırılırken tüm veri kümesi kullanımı bir dezavantajdır.

2.3.2 Policy learning by weighting exploration with the return (POWER)

POWER öğrenme algoritması fonksiyon bazlı bir algoritmadır. Bu algoritmanın hesaplanmasında iki farklı (w, x) değişken kullanılmaktadır. w hareket faz değerini, x ise olası dış bağlantılarını belirtmektedir (Kober & Peter, 2014). Belli sayılardaki tekrar sonrasında meydana gelen dönme hareketlerinin hız değerini öğrenebilmek için fonksiyonlar elde edilmektedir. Bu algoritma yaklaşık 20 tekrar sonucunda ideal sonuçlar elde edebilmektedir ve elde ettikleri her konum değişikliklerini kullanarak fonksiyon halini almaktadır (Daniel, Neumann, & Peters, 2012).

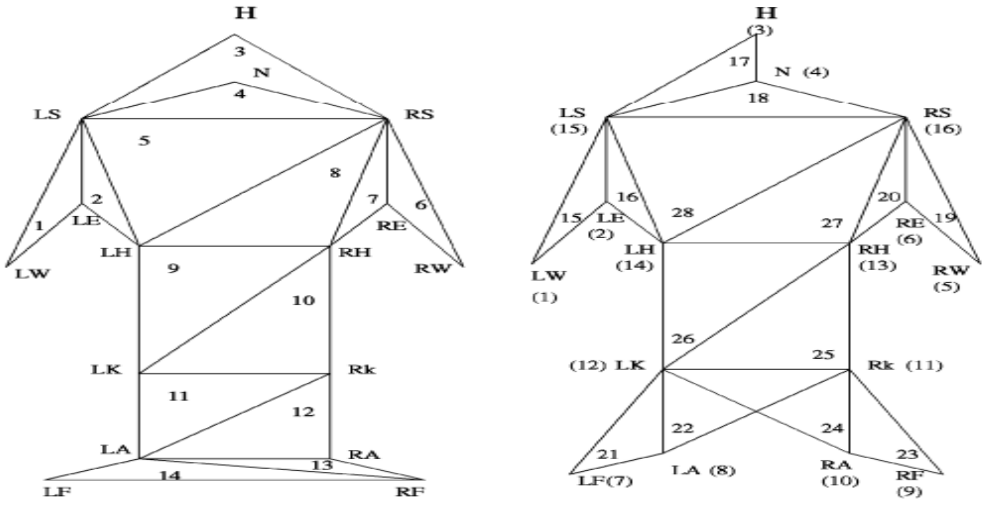
2.3.3 Denetimsiz öğrenme (Unsupervised learning)

İnsan hareketlerinin modellenmesi için kullanılan algoritmalardan biri de Denetimsiz Öğrenme'dir. Kümeleme ve ilişkilendirme problemleri olarak sınıflandırılır. İnsan vücudunun modellenmesi için bu yöntem ile elde edilen veri kümesinde konumlara göre

kümele işlemi yapılır ve her bir küme vücut üzerinde bir uzuva denk gelir. Yapılan kümeler herhangi bir test verisi olmadan oluşturulduğu için herhangi bir gözetim yoktur. Bu sebeple oluşturulan uzuvlar için herhangi bir geri bildirimde bulunulmaz (Le, 2013).

İnsan hareketlerinin modellenebilmesi için yapılan bir çalışmada Denetimsiz Öğrenme yönteminden yararlanılmıştır. Şekil 2.6 üzerinde görüleceği üzere, yapılan çalışmada iskelet sistemi üçgenler ile sembolize edilerek belirli köşe silme işlemi ile vücut organlarının modellenmesine çalışılmıştır.

Denetimsiz öğrenme algoritması ile yapılan ilk modellemelerde gözetim olmadığından dolayı, yapılan ile modellerin doğruluğunda bahsedilmez. Tekrar sayısı arttıkça modellemelerin de başarısı artmaktadır (Van Den Berg et al., 2010).

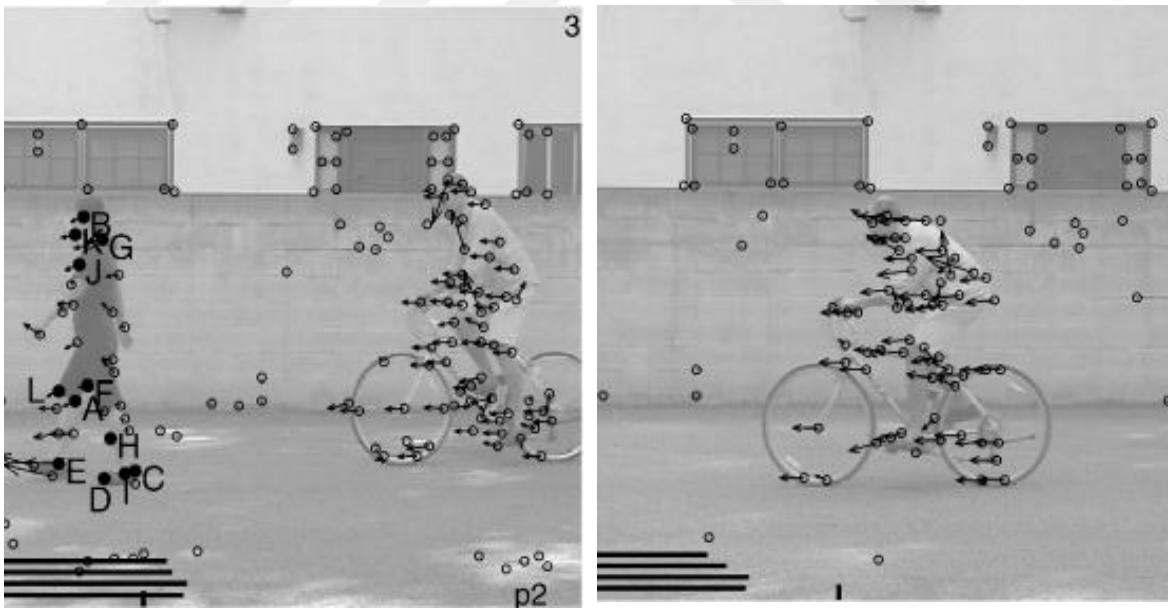


Şekil 2.6 İnsan Eklemlerinin Üçgenler İle Modellenmesi

‘L’ ve ‘R’ harfleri sol ve sağ bölümleri belirtmektedir. H: Kafa, N: Boyun, S: Omuz, E: Dirsek, W: Bilek, H: Kalça, K: Diz, A: Ayak bileği, F: Ayak.(Y. Song, Goncalves, Di Bernardo, & Perona, 2008)

Bu çalışmada yapılan tekrarlardan sonra incelenen hareketin optimal başarı ile sınıflandırıldığı görülmektedir. Ancak bazı durumlarda yapılan aktivitelerin sınıflandırılamadığı görülmüştür. Yapılan deneyler sonucunda sınıflandırılamayan durumlar başarısız modelleme olarak etiketlenmiştir.

Şekil 2.7 de iki farklı resmin modellenmesi sonucu 1. Karede yürüyen insanın, denetimsiz öğrenme algoritmaları ile elde edilen modelin insan kümesinde sınıflandırıldığı ancak 2.karede bisiklet kullanan kişinin modellemesinin insan kümesine konulmadığı görülmüştür. Bu sebeple yapılan modelleme başarısız modelleme olarak sınıflandırılmıştır (Yang Song, Goncalves, & Perona, 2003).



Şekil 2.7 Modelleme Örnekleri

(A) Başarılı Modelleme, (B) Başarısız Modelleme (Yang Song et al., 2003)

3. MATERYAL VE YÖNTEMLER

3.1 Materyal

İnsan hareketlerini modellemek amacıyla Microsoft Kinect cihazı ve OpenPose algoritması kullanıldı. Microsoft Kinect üzerinde bulunan sensörler ile insan iskelet sistemini modelleyebilirken OpenPose cihaz bağımsız bir algoritma olup, herhangi bir kameraya veya video kaydına entegre edilerek kullanılabilir. (Cao, Simon, Wei, & Sheikh, 2017; Nakai, Tsunoda, Hayashi, & Murakoshi, 2017; Zeng, 2012)

Bu çalışmada Kinect for Windows(V1) kullanılmıştır (Zeng, 2012). Kinect cihazının kullanılabilmesi için ürünün resmi sayfasında 2 GB RAM, 2.6 GHz ve üzeri hızda işlemcilere sahip bilgisayarlarda kullanılabileceği belirtilmiştir. Kinect cihazı için Microsoft tarafından yayımlanan SDK, C# diline uygun olarak yazılmıştır.

Tez çalışmasında kullanılan OpenPose algoritması, açık kaynaklı olup C++ dili ile yazılmıştır. OpenPose sistem gereksinimi olarak ekran kartında en az 1,6 GB'lık alan olması gerektiği ve 2 GB üzerinde boş RAM alanının olması gerektiği belirtilmiştir. Eğer kullanılan bilgisayar üzerinde yeterli GPU alanı yok ise bu durumda yalnızca CPU kullanılabilen bir versiyonda algoritma çalışabilmekte fakat en az 8 GB'lik boş RAM alanı ve kullanılan işlemcinin en az 8 çekirdekli olması gerekmektedir.

OpenPose gerçek zamanlı olarak video veya resim üzerinden birden fazla kişiyi eklemlerinden takip edebilmektedir. Ek olarak şekil 3.1 ve şekil 3.2 görüleceği üzere iskelet, el, yüz takibi yapabilmektedir ve toplamda 130 farklı noktanın verisi hesaplanabilmektedir.



Şekil 3.1 OpenPose Yüz ve El Takibi

(Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, 2017)



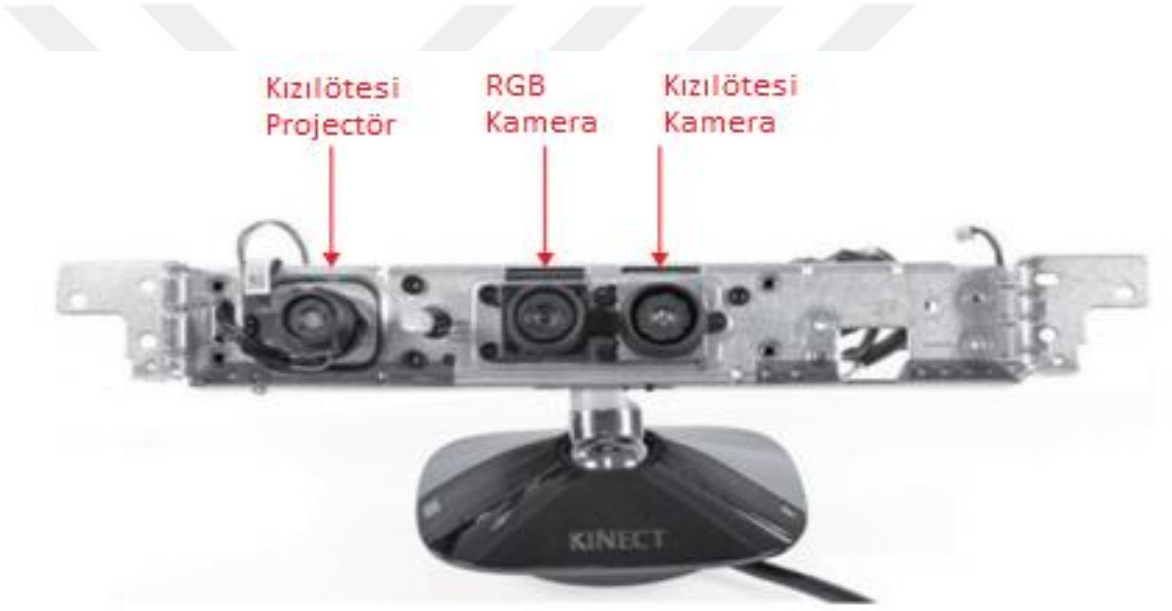
Şekil 3.2 OpenPose İskelet Takibi

(Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, 2017)

OpenPose başlangıç olarak yalnızca Linux işletim sistemli bilgisayarlarda çalışmaktaydı. Ancak yapılan güncellemeler ile Windows işletim sistemi bilgisayarlarda da desteklenmesi sebebi ile oldukça yaygınlaşmıştır. Açık kaynaklı olması sebebi ile de sürekli gelişerek daha verimli tahminler yapabilmektedir. Herhangi başka araca ihtiyaç duymadan günlük kayıtlar üzerinde çalışabilen bir algoritmadır (Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, 2017).

Kinect, Microsoft firması tarafından geliştirilen insan vücut hareketlerinin takibini sağlayan gelişmiş bir kameradır (Ganea, Mereuta, & Mereuta, 2014).

Şekil 3.3'de görülmekte olan cihaz üzerinde 3 farklı kamera sensörü bulunmaktadır. Bu sensörler ile 3 boyutlu insan vücut takibi yapılabilmektedir. Kinect cihazının temel amacı oyuncuları oyunlara interaktif olarak dahil etmesi olmasına rağmen Microsoft yayınlamış olduğu software development kit ile yazılımcıların kendi projelerinde Kinect cihazını kullanmasının önünü açmıştır.



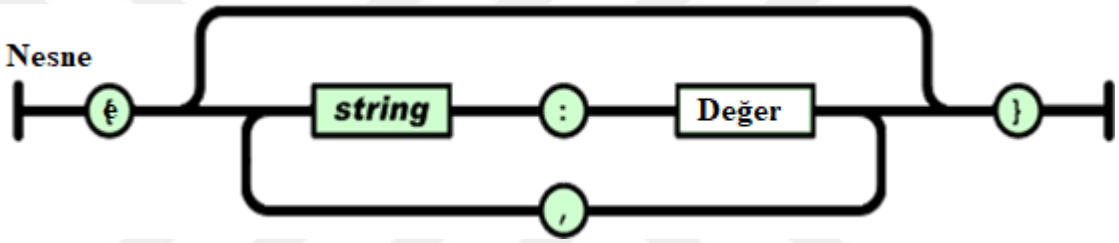
Şekil 3.3 Kinect Cihazını Üzerinde Bulunan Kamera Sensörleri

(Zeng, 2012)

Kinect kamera sensör insan iskelet sistemi takibine ek olarak, benzer şekilde el, yüz hareketlerinin takibi de yapabilmektedir. Ayrıca üzerinde bulunan kameralar yazılımsal olarak hareket ettirilebilir yapıdadır. Bu özelliği ile de daha geniş bir çerçeveden hareket takibi yapılabilmektedir.

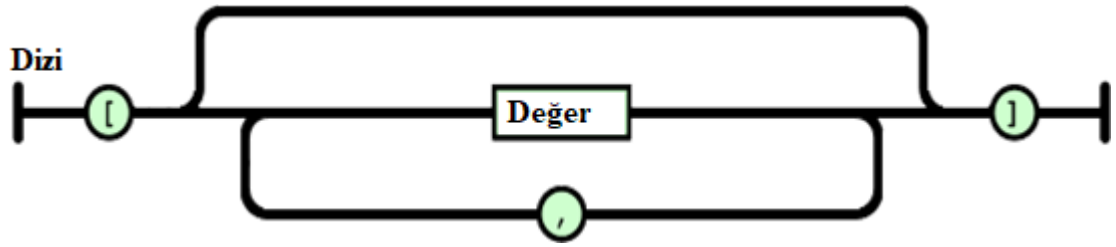
3.1.1 JSON veri formatı

JSON (JavaScript Object Notation) 1999 yılında standartlaştırılmış veri topluluğudur. Rahat okunabilir yapısı, kolaylıkla düzenlenebilir ve veri eklenebilir yapısı ile oldukça popülerdir. Dil bağımsız özelliği ile de birçok programlama dilinde rahatça kullanılabilen yapıdadır. (Shin, n.d.) Şekil 3.4 ve şekil 3.5'de JSON veri yapıları verilmektedir.



Şekil 3.4 JSON Key-Value Nesne Yapısı

("Introducing JSON," 2554)



Şekil 3.5 JSON Liste Nesne Yapısı

("Introducing JSON," 2554)

JSON veri tipi iki farklı yapıda olabilir.

- Birinci yapı bir nesne şeklindedir ve her nesne benzersiz anahtar kelimeleri ve her bir kelimenin değerini barındırır.
- İkincisi ise, her nesnenin liste halinde değerleri vardır. Dizi veya vektör olarak da nitelendirilebilirler.

OpenPose ile yapılan insan hareket modellemelerinde, elde edilen veri kümesi JSON formatındadır. Elde edilen JSON formatlı veriler üzerinden eklem değerleri alınarak liste değişkenlerine çevrilmiştir ve hesaplamalarda kullanılmıştır.

3.1.2 DLL (Dynamic Link Library) dosya formatı

DLL bir dosya formatıdır. Bu dosya formatı Windows işletim sistemine özel bir formattır. Kaynak kodlarını başka projeye aktarmadan yalnızca geliştirilen fonksiyonları başka uygulamalar tarafından kullanılmasını sağlamak amacıyla DLL dosya formatı kullanılmaktadır.

DLL'lerin tercih edilmesinin en büyük sebeplerinden biri değiştirilemez yapıda olmasıdır. Geliştirilen herhangi bir projede DLL dosyaları kolaylıkla kullanılabilir fakat DLL içinde bulunan fonksiyonlar özelleştirilemezler. Bu özelliği ile de güvenilir ve dağıtılabilir fonksiyonlar elde edilmektedir.

DLL dosyaları yazılım dillerinden bağımsız geliştirilmektedir. Farklı yazılım dilleri kullanılarak DLL yardımı ile tek bir uygulama gibi geliştirmeler yapılmaktadır.
Bu tez çalışmasında MATLAB dili ile yazılan interpolasyonlar ile polinom oluşturma kodları, DLL formatına dönüştürülerek C# dili tarafından kullanılabilir hale getirilmiştir.

3.2 Yöntemler

Bu tez çalışmasında madde madde uygulanan adımlar sırasıyla listelenmiştir.

- Microsoft Kinect ve OpenPose algoritması ile veriler elde edildi.
- OpenPose algoritmasından elde edilen veriler Microsoft Kinect sensörünün veri formatına dönüştürüldü.
- En uygun aralık ve grup değerini bulabilmek için farklı değerlerle modellemeler yapıldı. Modelleme sırasında her bir grup farklı bir dosya olarak oluşturuldu. İsimlendirme formatı olarak eklem adı ve grup adı kullanıldı. Belirlenen grup değeri kadar eklem değeri kullanılarak hesaplanan polinomun katsayıları kendi dosyasına yazıldı. Her bir interpolasyon yöntemi için ayrı klasörlerde dosyalar oluşturuldu. Dosyaların ayrı olmasının sebebi aranılan karenin, hangi gruba ait olduğu hesaplandıktan sonra ait dosya bir bilinmeyenli polinomu minimum işlem ile oluşturulabilmesidir. Bu polinomda üzerinde bilinmeyen değer aranan kare değeri ile hesaplandı.
- Orijinal veriler ile modellenen veriler karşılaştırılarak hesaplanan HKO ve MHO sonuçlarına göre en uygun olan aralık ve grup değerleri belirlendi. Bu değerler seçilirken mümkün olan en büyük değerlerin seçilmesi sağlandı. Aralık değerinin büyümesi kullanılan veri kümesinin küçülmesini sağladığı için kullanılan değerin büyüklüğü oldukça önemlidir.
- Birinci ve ikinci kişinin modellenen verilerinin karşılaştırılması yapılarak, başarı oranı hesaplanmıştır.

Veri kümesinin elde edilmesi esnasında bir saniyede her bir eklem için 30 kare bulunduğunu varsayalım. Örnek olarak incelenen 7 saniyelik bir videoda 210 kare elde edilecektir. Böyle bir videonun modellenmesi için tüm veri kümesinin kullanılması yerine belirli aralıklarda seçilen verilerin interpolasyon yöntemleri ile polinomu oluşturulacak ve bu polinomlar ile tüm veri kümesi geri elde edilebilecektir. Polinomlar tek değişkenli olarak her bir eklem için ayrı oluşturulacaktır. Bu sayede hangi eklemin

hangi karede nerede bulunduğu verisi çok hızlı bir şekilde hesaplanabilecektir. Polinomlardan geri elde edilen veri kümeleri, orijinal veri kümeleri ile karşılaştırıldığında küçük farklılıklar meydana gelmektedir. Bu farklılık interpolasyonların tüm veri kümesini kullanmadan modellemeler yaptığı için meydana gelmektedir.

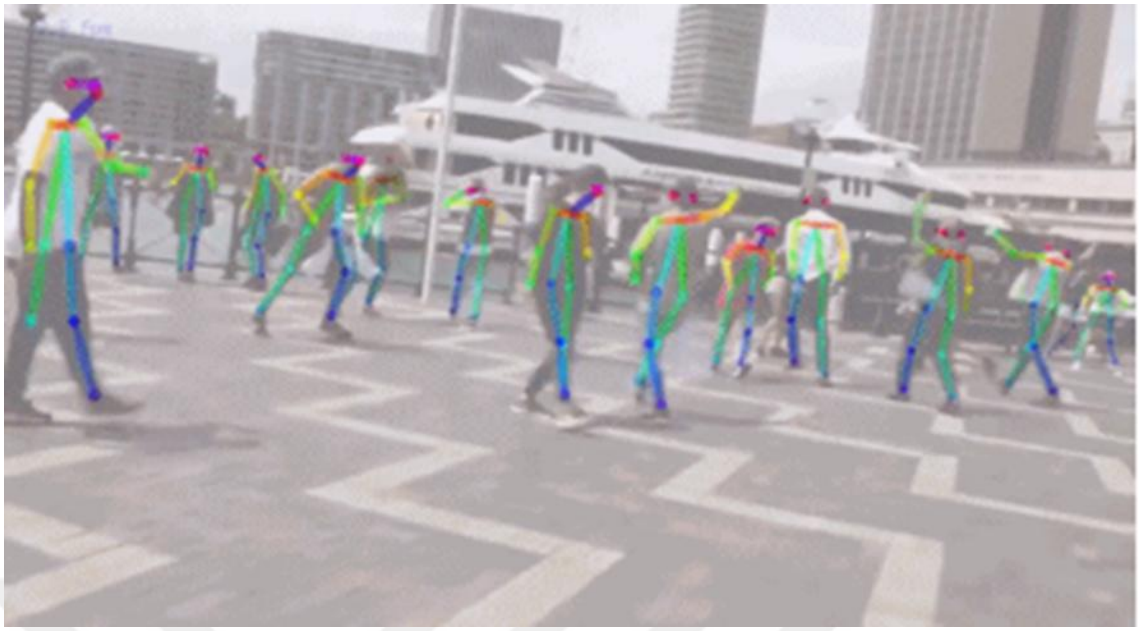
3.2.1 Veri Kümesi

Kullanılan veri kümeleri OpenPose uygulaması ve Kinect sensörleri ile elde edilmiştir. Kinect sensörü ile kaydedilen görüntüler işlenerek her bir kareden insan iskelet sistemi üzerinde bulunan eklemlerin konum değerleri elde edilir. Elde edilen eklem konum değerleri dosyalara kayıt edilerek modelleme esnasında kullanılabilir hale getirilir.

3.2.1.1 OpenPose ile veri kümesinin elde edilişi

OpenPose açık kaynaklı insan hareketlerini takip eden video, resim ve anlık kayıt üzerinde çalışarak modelleme yapabilen bir yazılımdır. İnsan vücut eklemlerinin, yüz ve el hareketlerinin anlık olarak kare takibi yapılabilmektedir. Bu takip esnasında kamera ve modellemesi yapılan insan arasında bulunan mesafe sabit tutulmalıdır. Özellikle videolar üzerinden modellemeler yapılırken, yakınlaştırma yapılan kayıtlarda iskeletin tamamı takip edilmediğinde sapmalar meydana gelmektedir.

Şekil 3.6'da OpenPose ile insan hareket verilerinin alınma görüntüsü verilmektedir.



Şekil 3.6 Openpose İle İnsan Hareketlerinin Takibi

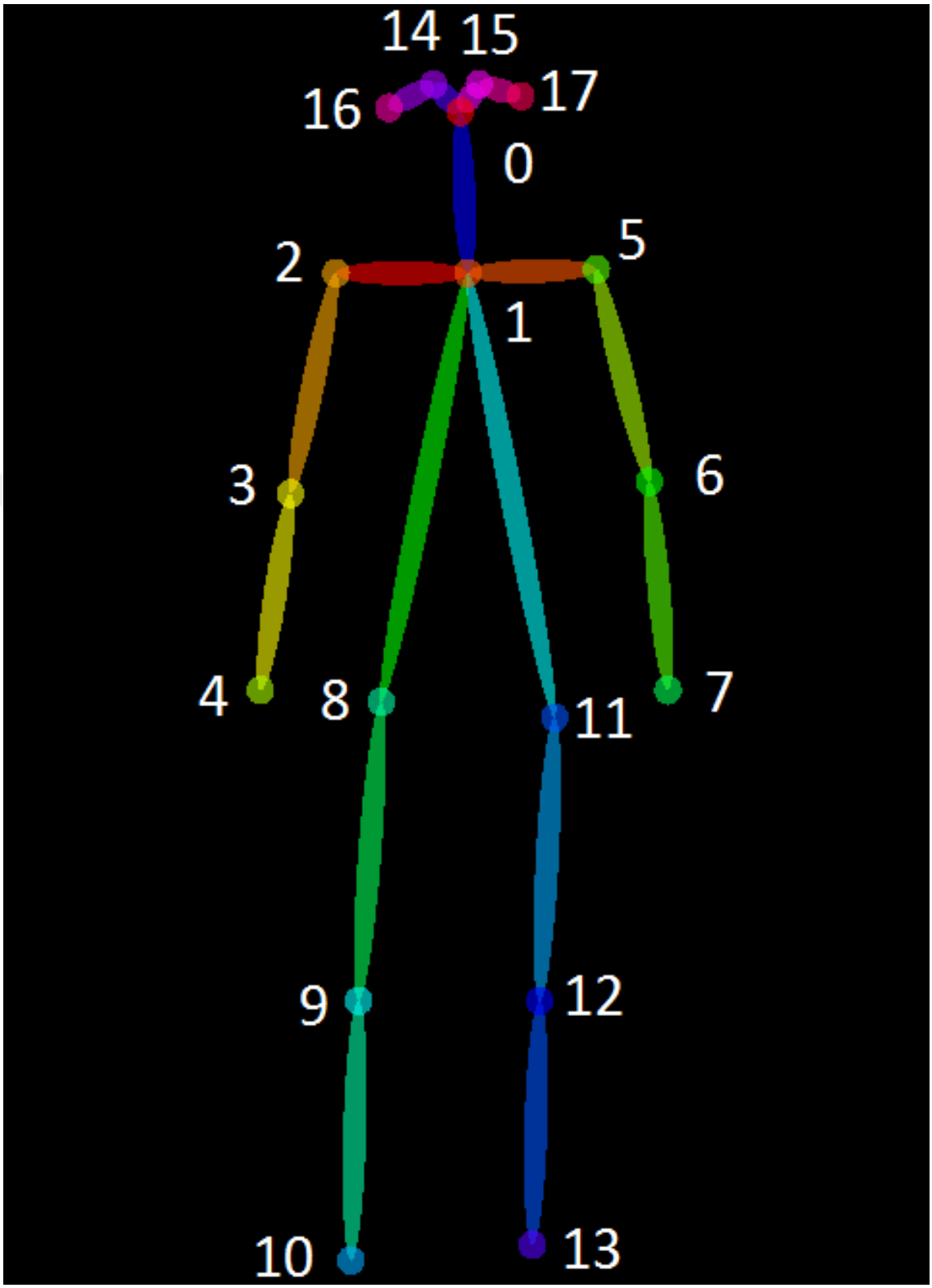
(Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, 2017)

Şekil 3.7 de görüldüğü üzere OpenPose ile 18 farklı eklem konum bilgisi takip edilebilmektedir. Eklem bilgileri kartezyen koordinat sistemi üzerinde gösterilebilir. Bu modelleme esnasında birden fazla insan takibi yapılabilmektedir. Ancak kullanılan bilgisayar gücüne göre bir saniyede incelenen kare sayısı videoda bulunan insan sayısına göre değişiklik göstermektedir. Hesaplamalar sırasında kareler üzerinden takip yapıldığı için, bu durum bir sorun meydana getirmemektedir.

Çizelge 3.1 de görüleceği üzere, OpenPose ile elde edilen eklem veri kümesi JSON formatındadır. Veri kümesinin ilk elemanı olan sayılar şekil 3.7 üzerinde işaretlenmiş olan bölgelere denk gelmektedir. OpenPose ile elde veri kümelerinde bazı eklem konum değerleri x1,y1,c1,x2,y2,c2, bazı eklem konum değerleri ise x,y,c şeklindedir. Burada c değişkenleri verinin güven skorunu belirtmektedir. Eklem konum bilgisinde birden fazla x, y, c değişkeninin bulunması farklı güven skorlarında aynı karede aynı eklem için yapılan tahminlerden kaynaklanmaktadır. Burada c değeri yüksek olan eklem konum bilgisinin alınması daha iyi modellemeler yapılmasını sağlamaktadır (Cao et al., 2017).

Çizelge 3.1 Openpose İle Elde Edilen Eklem Konum Değerleri Veri Kümesi

"0":[310.011, 75.0179, 0.983612, 286.022, 84.0426, 0.757338],	**"1"**:[335.012, 96.0035, 0.74601, 268.035, 109.003, 0.78616],	**"2"**:[327.001, 98.9962, 0.663911, 260.005, 109.007, 0.895391],	**"3"**:[304.985, 134.005, 0.628113, 265.955, 138.968, 0.860017],
"4":[276.048, 107.96, 0.858432, 298.984, 130.007, 0.445422],	**"5"**:[343.003, 92.0392, 0.809188, 278.003, 108.03, 0.59922],	**"6"**:[327.004, 116.044, 0.869507, 275.994, 140.019, 0.0512275],	**"7"**:[289.017, 101.024, 0.992053],
"8":[328.992, 173.021, 0.609511, 269.004, 174.008, 0.647082],	**"9"**:[268.002, 235.976, 0.616006, 335.986, 236.031, 0.70483],	**"10"**:[261.008, 288.001, 0.377169, 341.008, 288.001, 0.373216],	**"11"**:[336.013, 173.02, 0.639251, 280.978, 173.996, 0.540456],
"12":[270.019, 234.997, 0.47316, 324.998, 236.996, 0.721656],	**"13"**:[260.987, 284, 0.322648, 331.004, 291.014, 0.472995],	**"14"**:[283.964, 82.9872, 0.913366],	**"15"**:[313.979, 68.0419, 0.954664, 284.054, 83.9762, 0.132955],
"16":[269.025, 87.0142, 0.756887],	**"17"**:[326.002, 66.9947, 0.892902]		



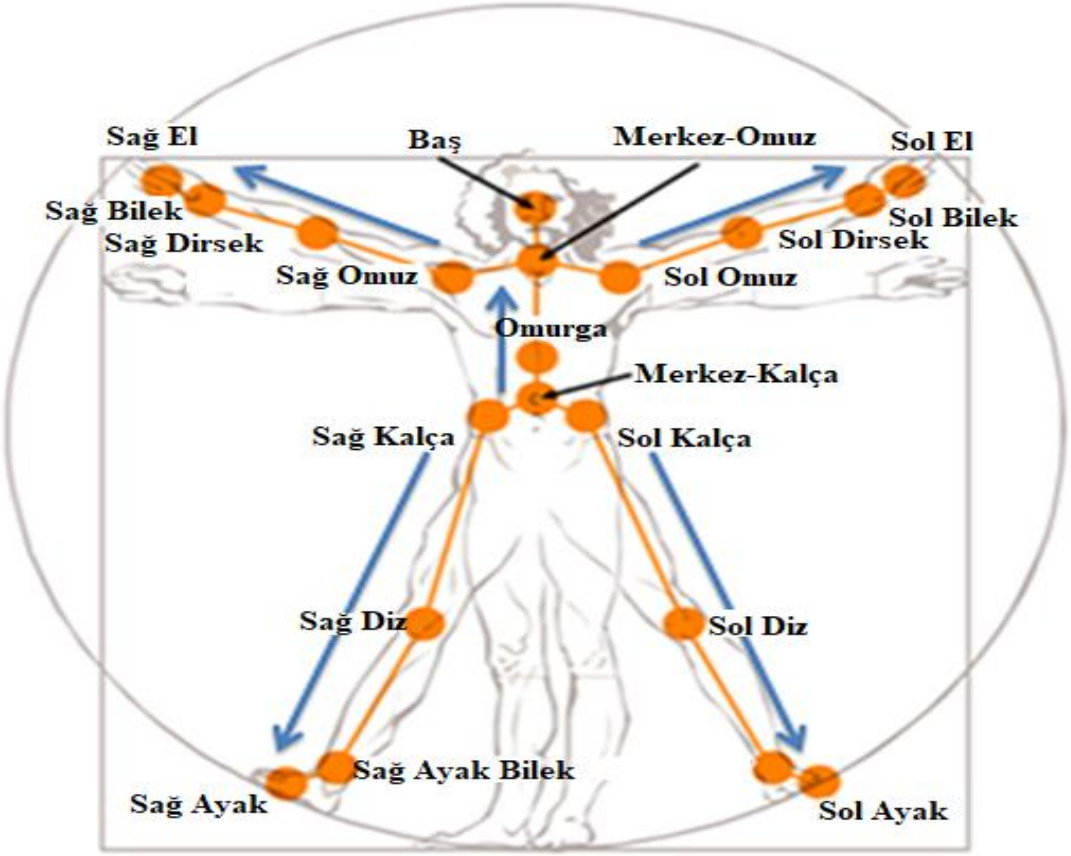
Şekil 3.7 Openpose İle Takip Yapılan İskelet Sistemi Noktasal Eklemleri

(Miguel Aarduengo Sven Jens Jorgensen Supervisors Kimberly Hambuchen Luis Sentis

Francesc Moreno Guillem Alenya, 2017)

3.2.1.2. Kinect sensörü ile veri kümesinin elde edilişi

Kinect Microsoft firmasının geliştirdiği, insan iskelet sistemini takip eden bir üründür. Bu ürünün geliştirilme amacı yine Microsoft firmasının geliştirmiş olduğu XBox oyun konsolu ile oyuncuları oyunlara interaktif olarak dahil edilmesini sağlamaktır. Ancak yayınlanan Windows SDK ve veri aktarım kablosu ile yazılım geliştiriciler tarafından kullanılabilmesi sağlanmıştır (Zerpa et al., 2015). Kinect sensörü ile insan hareketlerinin takibi için her bir saniyede en fazla 30 kare işlenmektedir ve iskelet sistemi üzerinde ki 20 farklı eklem verisi elde edilmektedir. Takip edilen eklemler Şekil 3.8 üzerinde gösterilmiştir. Sistemsel olarak en fazla 30 olarak belirlenen FPS değeri düşürülebilmektedir. İstendiği takdirde her bir saniyede daha az kare takibi yapabilmek mümkündür.



Şekil 3.8 Kinect Sensörleri İle Takip Edilen Eklemlerin İskelet Sistem Konumları (Game, 2016; Ren, Yuan, Meng, & Zhang, 2013)

Kinect sensörleri ile elde edilen eklem konumlarını OpenPose'dan farklı olarak amc formatındadır ve işlenen kare sayısı sabit kalmaktadır.

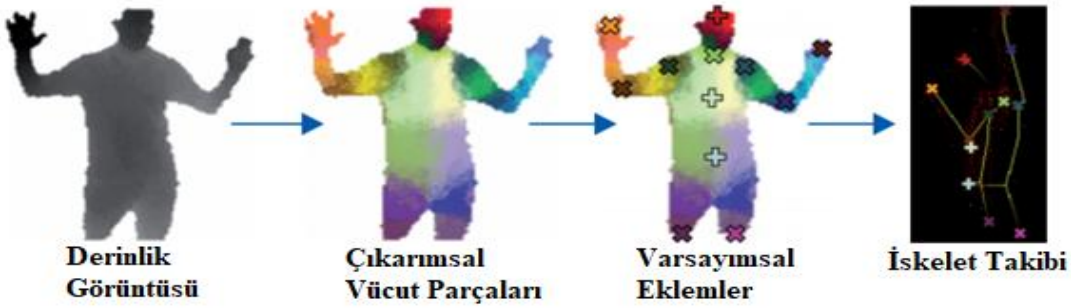
Çizelge 3.2 de Microsoft Kinect Sensörleri ile takip edilen insan hareketlerinin 270. Karedeki iskelet sistemi üzerinde bulunan eklem konum değerlerinin veri kümesi bulunmaktadır. Veri kümesi ilk satırda ait olduğu karenin numarası ile başlar ve devamında boşluklar ile ayrılarak sırasıyla eklem adı, X konum bilgisi, Y konum bilgisi ve rotasyon bilgisi bulunmaktadır.

Çizelge 3.2 Microsoft Kinect İle Elde Edilen Eklem Konum Değerleri Veri Kümesi

270

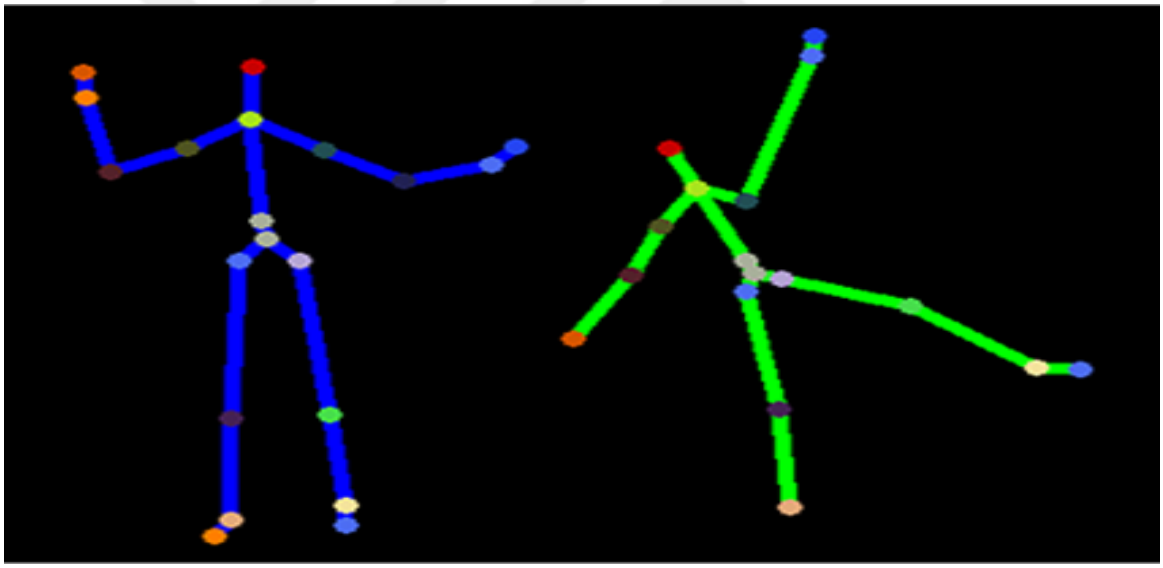
HipCenter	1.517299	-0.2663365	3.172178
Spine	1.527393	-0.2054516	3.178674
ShoulderCenter	1.577862	0.09897247	3.225849
Head	1.502417	0.2761515	3.186341
ShoulderLeft	1.478438	0.02340087	3.370244
ElbowLeft	1.429889	-0.1756384	3.362745
WristLeft	1.290652	-0.1238605	3.208721
HandLeft	1.257245	-0.1065912	3.16587
ShoulderRight	1.633906	0.02130159	3.099792
ElbowRight	1.625241	-0.2493066	3.012941
WristRight	1.443323	-0.1537324	2.998887
HandRight	1.386791	-0.1517075	2.991777
HipLeft	1.471184	-0.3330276	3.21974
KneeLeft	1.470582	-0.7633861	3.124828
AnkleLeft	1.465996	-1.067358	3.100488
FootLeft	1.430762	-1.151783	3.070029
HipRight	1.541144	-0.3339723	3.10046
KneeRight	1.54665	-0.6347807	2.968357
AnkleRight	1.579809	-0.8553371	2.917449
FootRight	1.536089	-0.9351836	2.949226

Kinect sensörleri ile yapılan iskelet takibi şekil 3.9'da verilmektedir. Takip sırasında, hareketlerin çok hızlı olması veri kayıplarının yaşanmasına neden olabilmektedir. Bu tip sorunlar geliştirilen algoritma ile çözülebilmektedir. İnsan hareketlerinin takibi cihaz üzerinde bulunan derinlik kamerası ile yapılmaktadır. Bu kamera ile şekil 3.10'da gösterildiği şekilde eklemler üzerinden tahminlerde bulunarak konum değerleri hesaplanmaktadır. Benzer şekilde hareket esnasında eklem noktalarının üst üste gelmesi sonucu veri kayıpları olabilmektedir.



Şekil 3.9 İskelet Takibinde Kinect Sensörü Üzerindeki Kamera Görüntüleri

(Zhang, 2012)



Şekil 3.10 Kinect Depth Kamera İle İnsan İskelet Takibi

(J. Wang, Liu, Wu, & Yuan, 2012)

Tek bir Microsoft Kinect Sensörü (V1) ile aynı anda en fazla 2 farklı iskelet takibi yapılabilmektedir. Burada iskelet sayısının 1 ya da 2 olmasının sistemin çalışma süresi üzerinde hiçbir etkisi yoktur. Birden fazla iskelet takibi yapıldığı sırada, farklı iskeletlere ait eklemlerin de kamera açısına göre üst üste gelmesi sonucu veri kümesinde kayıpların oluşmasına sebep olmaktadır (Alexiadis et al., 2011a; Kar, 2010; Shotton, J Fitzgibbon, A Cook, M Sharp, T Finocchio, M Moore, 2011).

3.2 Veri Kümesinin Hazırlanması

Veri kümelerinin elde edilmesinden sonra seçilen grup ve aralık değerlerine göre verilerin düzenlenmesi sağlanmıştır. Kinect sensörleri için örnek olarak tam 5 saniye uzunluğundaki bir videodan 150 kare elde edilecektir. Grup değerinin 5, aralık değerinin 8 seçildiği durumda 18 farklı kare kullanılacaktır. Grup değerinin 5 olmasından dolayı 4 farklı grup olacaktır. Bu çalışmada her bir eklem ayrı bir dosyada tutulmuştur. Aynı zamanda eklemin içinde kendi aralarında ayrılan gruplar farklı dosyalarda tutulmuştur. Veri hazırlama işlemi tamamlandıktan sonra 5 saniyelik bir video için toplamda 80 farklı dosya oluşacaktır.

OpenPose ile çekilmiş olan görüntülerden iskelet takibi yapılmıştır. Kinect sensörlerinin aksine OpenPose için sabit bir FPS değeri yoktur ve kullanılan sistem özelliklerine göre değişim göstermektedir. Bu çalışmada OpenPose kullanılan bilgisayarın özellikleri Intel i7 4.00 GHz işlemci 32GB bellek ve 64 bit Windows 10 işletim sistemidir. Ekran kartı ise NVIDIA GeForce GTX790 4GB 256Bittir. OpenPose grafik kartı üzerinde çalıştığı için FPS değeri için ekran kartlarının gücü oldukça önemlidir.

Bu özelliklere sahip bir bilgisayarda OpenPose ile bir kişinin iskelet takibi sırasında 12-13 FPS ortalamasına sahip olduğu görülmüştür. Bu sebeple kayıt süresi Kinect ve OpenPose için aynı olmasına rağmen elde edilen kare sayıları arasında ciddi farklar bulunmaktadır. Benzer bir örnekle bu çalışmada kullanılan bilgisayar ile 5 saniyelik bir videoda OpenPose ile 60 kare iskelet takibi yapılabilmektedir. Yine Grup değerinin 5, aralık değerinin 8 seçildiği durumda 7 farklı kare kullanılacaktır. Grup değerinin 5 olmasından dolayı 2 farklı grup olacaktır.

OpenPose ile işlenen verilerin dışarı aktarılma biçimi oldukça değiştirilebilir yapıdadır ve bu çalışmada JSON formatında veriler elde edilmiştir. Geliştirdiğimiz sisteme uygun olması açısından OpenPose'dan elde edilen JSON verileri ve Kinect sensörlerinden elde edilen veriler ortak bir veri formatına dönüştürülmüştür. Bu sayede veri kaynağı farklı olsa bile dönüştürülen veriler aynı algoritmalar ile modellenmiştir.

Kinect sensörlerinden gelen veriler satırlar sırasıyla ilk olarak kare numarası ile başlar ve hep aynı sırada olacak şekilde eklem isimleri ile devam eder. Eklem isimlerinin yanlarında ise ilk satırda yazılı olan kare için her bir ekleme ait konum değerleri yazar. Konum değerleri arasında boşluk karakteri bulunmaktadır. C# ile eklem konum değerlerinin liste haline getirilebilmesi verilerin bulunduğu satırlar okunarak her bir eklem bir liste olacak şekilde oluşturulur ve her listeye sırayla kendi ekleminin değerleri eklenir. Verilerin ayrıştırılması işlemi sonrasında takip edilen eklem sayısı kadar liste meydana gelir. Her bir listenin veri boyutu eşit ve hareketlerin takibi sırasında elde edilen kare sayısı kadar olmaktadır.

Benzer şekilde OpenPose algoritmasının çalıştırılması sonucunda, elde edilen veriler C# ile ayrıştırılmaktadır. Kinect sensörlerinden elde edilen verilerin aksine OpenPose algoritmasından elde edilen veriler JSON veri formatında olduğundan dolayı burada satırların tek tek okunmasına gerek yoktur. İstenilen eklem verisi JSON veri formatı içerisinden seçilebilmektedir. Bu sebeple yalnızca kullanılacak olan eklemin liste haline getirilmesi yeterli olacaktır.

3.4 Önerilen Yöntem

İnsan hareket modellemesi için kullanılan DMP matematiksel formüllerinde, meydana gelen hareketten elde edilen tüm veri setinin kullanılması gerekmektedir. Veri kümesinin büyümesi ile birlikte yapılan matematiksel işlemlerin süreleri artmaktadır.

Meydana gelen hareketlerde, daha az veri kullanılarak daha hızlı modellemeler yapılabilmesi için bu tez çalışmasında modellemelerin interpolasyon yöntemleri ile yapılması önerilmektedir.

Veri kümelerinin interpolasyon yöntemleri ile modellenebilmesi için aralık ve grup değerlerine ihtiyaç duyulmaktadır. Burada aralık değerinin büyümesi kullanılan veri küme boyutunun küçülmesini sağlarken, grup değerinin küçük olması da polinomlardan eklem değerlerinin geri elde edilmesi için yapılan hesaplamaların basitleştirilmesini sağlamaktadır. Grup değeri ne kadar büyük olursa, hareketlerin modellenmesi sonucu elde edilen polinomların derecesi kullanılan grup değerinin bir eksiği olacağından dolayı hesaplama süreside o kadar uzun olmaktadır. İnsan Hareketlerinin Modellemesi için DMP ve İnterpolasyon yöntemleri arasından kullanılan veri kümesi boyut farklılıklarının gösterimi çizelge 3.3 de gösterilmiştir.

Çizelge 3.3 DMP ve İnterpolasyon Yöntemlerinde Kullanılan Veri Kümesi Boyutları

Yöntemlere Göre Kullanılan Veri Kümesi Uzunluğu / **Veri Kümesi Uzunluğu**	**Aralık Değeri**	**DMP**	**İnterpolasyon**	**İnterpolasyon İle Kullanılan Veri Yüzdesi**
100	8	100	12	%12
100	5	100	20	%20
100	2	100	50	%50

Çizelge 3.3 DMP ve İnterpolasyon Yöntemlerinde Kullanılan Veri Kümesi Boyutları (Devamı)

400.000	8	400.000	50.000	%12
400.000	5	400.000	100.000	%20
400.000	2	400.000	200.000	%50

Hata oranlarına bakılarak seçilen aralık değerinin olabilen en büyük ve grup değerinin olabilen en küçük olması sonucu kullanılan bilgisayar kaynağını azaltmaktadır. Bu durumda önerilen yöntemin DMP formüllerinden daha verimli modellemelerin yapılacağı düşünülmektedir.

3.5 MATLAB ile İnterpolasyon Hesaplamalarının Yapılması

MATLAB dili ile matematiksel işlemler hızlı bir şekilde yapılabilir. İçerisinde çok miktarda hazır matematiksel fonksiyonlar bulunmaktadır. Bu sayede birçok işlem kolaylıkla yapılabilmektedir.

MATLAB üzerinde çeşitli matematiksel işlemleri yapan fonksiyonlar oluşturulabilir. Bu fonksiyonlar DLL haline getirilerek programlama dilleri tarafından kullanılabilir. DLL haline getirilen fonksiyonlar ile MATLAB dili ve diğer programlama dillerinin birlikte çalışması mümkün hale gelir.

Bu çalışmada kullanılan Lagrange, Spline ve Kübik Spline interpolasyon fonksiyonları MATLAB dilinde fonksiyon olarak DLL haline getirilerek C# dili ile kullanılmıştır (Halıcı & Bostanci, 2019).

MATLAB fonksiyonlarının çalışma sırasında beklediği bazı parametreler bulunmaktadır. C# dili ile MATLAB fonksiyonlarını kullanırken bu parametrelerin verilmesi gerekmektedir. MATLAB dilinin kendi veri tipleri bulunmaktadır. Bu veri tipleri C# dilinde tanımlanabilir ve önceden hazırlanan liste verileri bu veri tiplerine dönüştürülebilir (“Numerical computing with MATLAB,” 2013).

3.6 Veri Kümelerinin Normalize Edilmesi

Microsoft Kinect sensörü ve OpenPose algoritması ile modellenen insanların hareketlerinin karşılaştırılması istendiğinde, kişilerden elde edilen verilerin normalize edilmesi gerektiği düşünülmektedir. Birinci kişinin yapmış olduğu dans hareketlerini ikinci kişinin taklit etmesi sonucu oluşan modellerin karşılaştırılabilmesi için iki veri kümesinin benzer değer aralıklarında olması gerekmektedir.

Normalize işlemi sırasında elde edilen eklem verileri, verilerin sahibi olan kişinin boy uzunluğuna oranlanarak her bir eklem için bir katsayı elde edilmiştir ve sonrasında ortak bir uzunluk ile bu katsayının çarpılması ile yeni bir veri kümesi oluşturulmuştur. Bu çalışmada ortak uzunluk değeri için 200 seçilmiştir.Bir başka deyişle, veriler 200 cm uzunluğundaki bir vücuda göre normalize edilmiştir.

Normalize işlemi sonrasında, hareketleri modellenen kişilerin boy uzunluklarından bağımsız olarak hazırlanmış veri kümeleri ile karşılaştırma yapılabilir hale getirilmesi beklenmektedir.

Boy uzunluğu b, eklem konum değeri e olan verinin normalizasyonu yapıldıktan sonra yeni eklem değeri x olsun ve aşağıdaki formül ile hesaplanabilmektedir.

$$x = \frac{200 * e}{b}$$

3.7 Veri Kümelerinin Karşılaştırılması

İki farklı kişi tarafından dans hareketlerinin yapılması ve elde edilen verilerin modellenmesinden sonra, elde edilen modeller üzerinde karşılaştırmalar yapılmıştır. Karşılaştırma yapılmadan önce modellerin başlangıç noktaları eşitlenmiştir. Bu işlem ile aynı karelerde aynı hareketlerin yapılıyor olması beklenmiştir.

İki farklı kişi tarafından hareketlerin farklı zamanlarda yapılması sonucu, elde edilen iki veri kümesinde farklı miktarlarda kare sayıları meydana gelmektedir. Aynı sayıda kare karşılaştırmasının yapılabilmesi için veri kümelerinden küçük olanın kare sayısı ele alınarak karşılaştırmalar yapılmıştır. Eksik olan kareler hata sonuçlarına eklenmiştir.

Karşılaştırmalar yapılmadan önce veriler üzerinde normalize adımları da ayrıca uygulanmıştır. Normalize edilen veriler de kendi aralarında ayrıca incelenmiş olup test sonuçlarına eklenmiştir.

4. ARAŞTIRMA BULGULARI

VK1: Yalnızca sağ el sallaması sonucu meydana gelen hareket modeli

VK2: Eğilip kalkma sonucu meydana gelen hareket modeli

VK3: Eşli Bachata dansı sonucu meydana gelen hareket modeli

VK4: Havada sağ el ile 8 şekli çizilmesi sonucu meydana gelen hareket modeli

VK5: 2. Kişinin 1.Kişinin boks hareket figürlerinden oluşan dans koreografisinin tekrarı sonucu meydana gelen hareket modeli

Veri kümeleri bir ev ortamında elde edilmiştir. Kinect cihazı üzerinde bulunan sensörler ve günlük kullanıma uygun bir cep telefonu ile elde edilen görüntüler işlenerek modellenebilir veri kümeleri oluşturulmuştur. Kullanılan cep telefonunun kamerası 13 MP'dir.

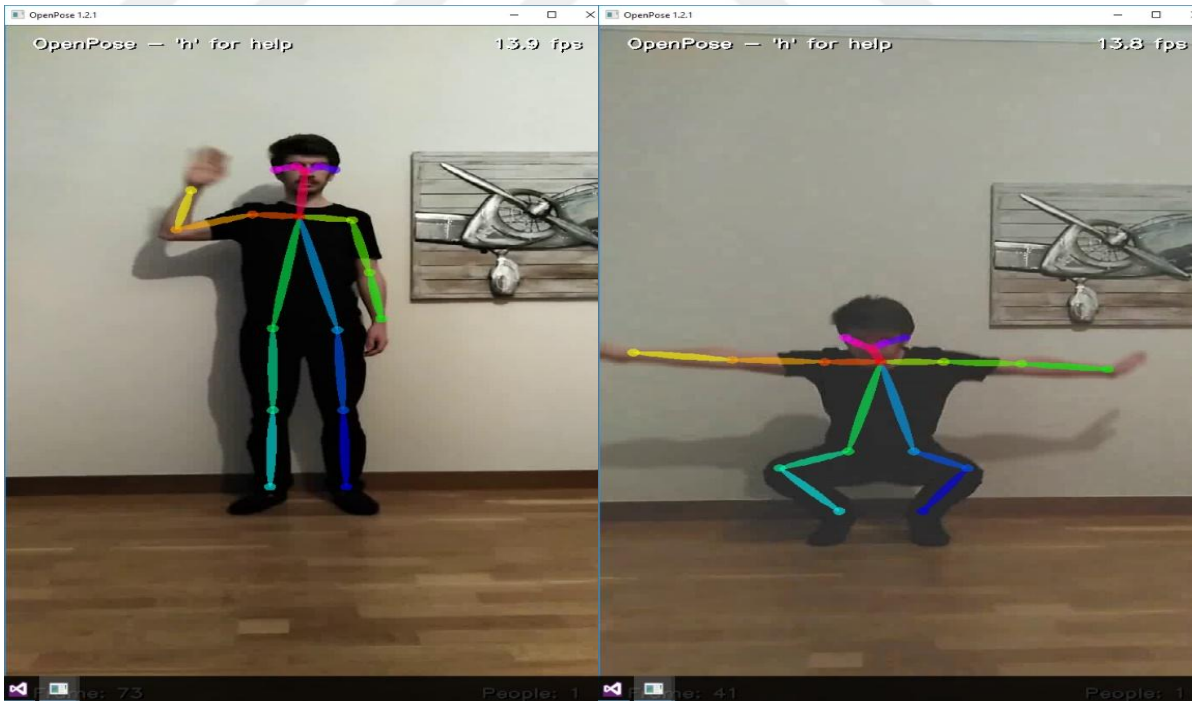
Kayıt sırasında Microsoft Kinect ve cep telefonu kamerası birbirlerine oldukça yakın tutulmuştur ve hareket esnasında iki cihazda aynı anda başlatılıp aynı anda durdurulmuştur. Kinect sensörleri ile eş zamanlı veriler elde edilmiştir fakat cep telefonu kamerası ile elde edilen görüntüler bilgisayar hafızasına aktarılarak OpenPose ile modellenebilir veri kümeleri elde edilmiştir.

OpenPose ve Microsoft Kinect üzerinden elde edilen veri kümeleri öncelikle kendi aralarında incelenerek başarı grafikleri oluşturulmuştur. Daha sonrasında elde edilen veri kümeleri üzerinde interpolasyon yöntemleri uygulanarak modellemeler yapılmış ve sonuçlar üzerinden çıkarımlar yapılmıştır.

Ayrıca OpenPose ve Microsoft Kinect ile elde edilen veri kümeleri karşılıklı olarak incelenmiş ve aralarındaki başarı oranları hesaplanarak grafikler üzerinde görselleştirilmiştir.

Aynı dans figürlerinin iki farklı kişi tarafından yapılması sonucu modellemeler yapılarak, veri kümeleri arasında meydana gelen farklılık hesaplamaları yapılmıştır. Bu farklılıklar üzerinden koreografi aktarım skorları hesaplanarak başarı değeri hesaplanmıştır.

İlk olarak yalnızca sağ el sallaması sonucu meydana gelen eklem şekil 4.1'de gösterildiği gibi konum değişikleri 3 farklı interpolasyon ile modellendi ve elde edilen modellerin hata oranları sağ eklemi için hesaplandı.



Şekil 4.1 Openpose Veri İşleme Aşamasındaki Görüntüleri

Kamera ile elde edilen görüntülerin bilgisayar ortamında OpenPose ile işlenebilir veri kümeleri oluşturulması.

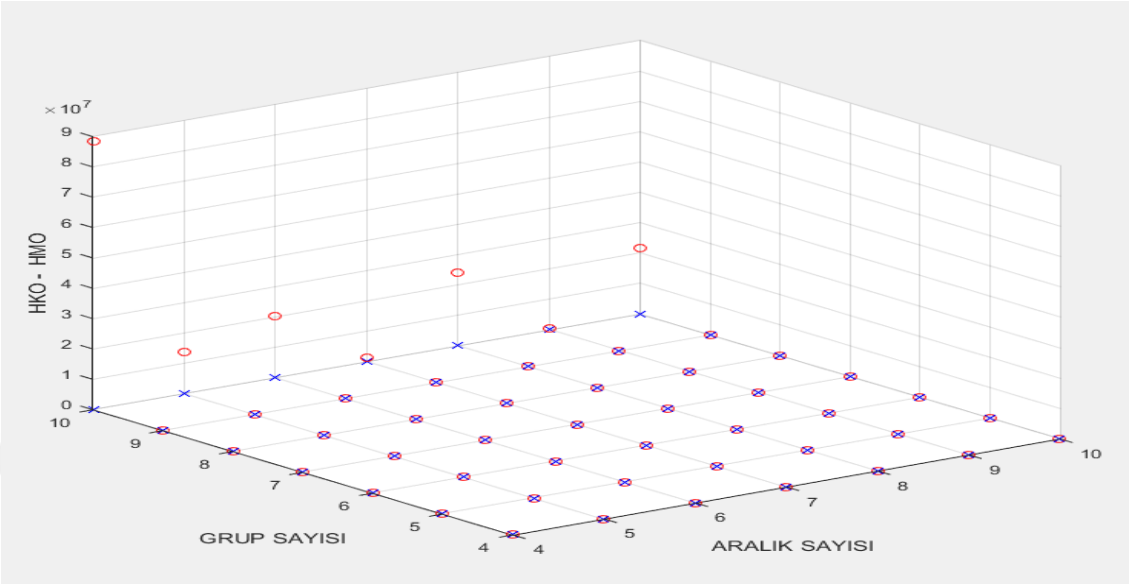
Her bir interpolasyon türünden elde edilen modeller ile orijinal veri kümeleri karşılaştırılarak ortaya çıkan hata değerleri grafikler üzerinde gösterildi. Grafik üzerinde aralık ve grup değişkenlerinin değerinin artmasının meydana gelen hata değerlerini de ciddi olarak etkilediği görülmektir. İnterpolasyon yöntemleri ile modelleme sırasında ortaya çıkan sıçramaların da benzer şekilde aralık ve grup sayılarına bağlı olarak arttığı görülmektedir. Bu şekilde yapılan aralık ve grup değerlerine, bu iki değerin farklı kombinasyonları ile modellemeler yapılarak ortaya çıkan hata değerlerine göre karar verilmesi sağlanmıştır. Farklı interpolasyon yöntemleri için farklı aralık ve grup değeri seçilerek hata oranları düşürülebilmektedir.

Yine bu grafiklere bakarak, modellenen kare sayısının artması hesaplanan HKO ve MHO değerlerinin de artmasına sebep olmaktadır. OpenPose algoritmasının eklem değer aralıkları ile Microsoft Kinect sensöründen elde edilen eklem değer aralıklarının farklı olması, grafikler üzerinde gösterilen HKO ve MHO değerlerini etkilemektedir.

Grafik üzerinde orijinal veri kümesinin herhangi bir uç noktası olmamasına rağmen Kübik Spline, Spline ve Lagrange interpolasyonları ile elde edilen verilerde bazı uç nokta sapmaları meydana gelmektedir. Bu sapmaların en temel sebebi y eksenin üzerindeki salınımlardır. Aralık değerinin rastgele belirleniyor olması hareketlerin y eksenin uç değerlerini kaçırma ihtimalini oluşturuyor ve kaçırmalar sonucu Şekil 4.8 üzerinde rahatça görülebileceği üzere sapmalar meydana getirmektedir. HKO ve MHO değerlerinin büyümesinin sebebi bu sapmalardan kaynaklanmaktadır.

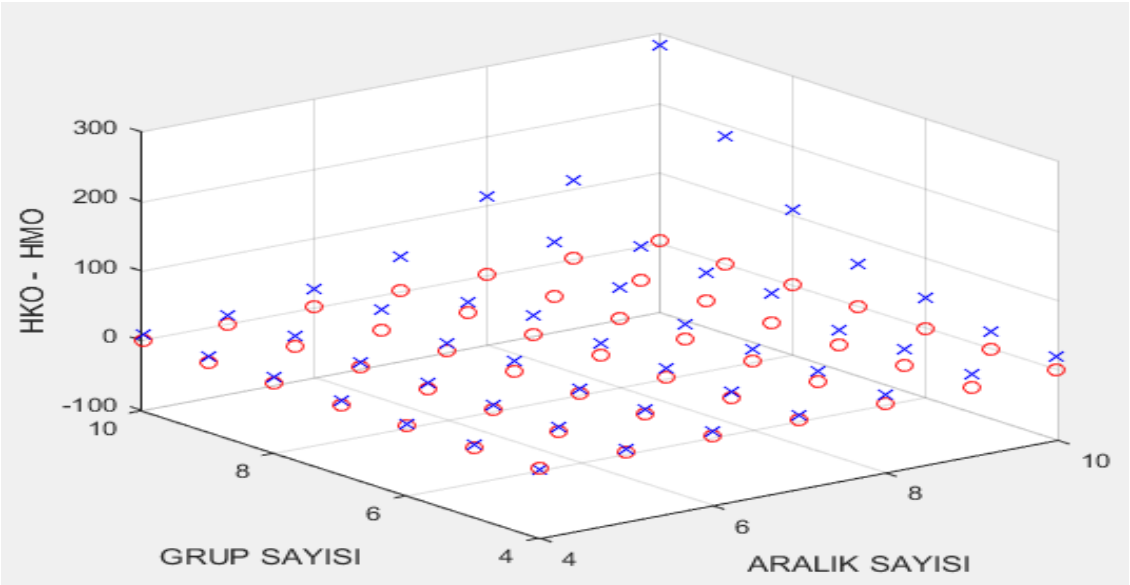
Şekil 4.2 ile 4.7 arasındaki grafiklerde çeşitli eklemler için HKO ve MHO değerleri hesaplanmıştır. Bu grafikler de farklı aralık ve grup değerlerinin birbirleri ile farklı kombinasyonları sonucu meydana gelen hata değerleri gösterilmektedir. Bu grafikler üzerinden modellemesi yapılan hareket ve eklem için uygun olan aralık ve grup değer

seçimleri yapılabilir. Bu seçimler yapılırken aralık ve grup değerlerinin mümkün olan en büyük sayılardan seçilmesi modellemelerin daha verimli olmasını sağlamaktadır.



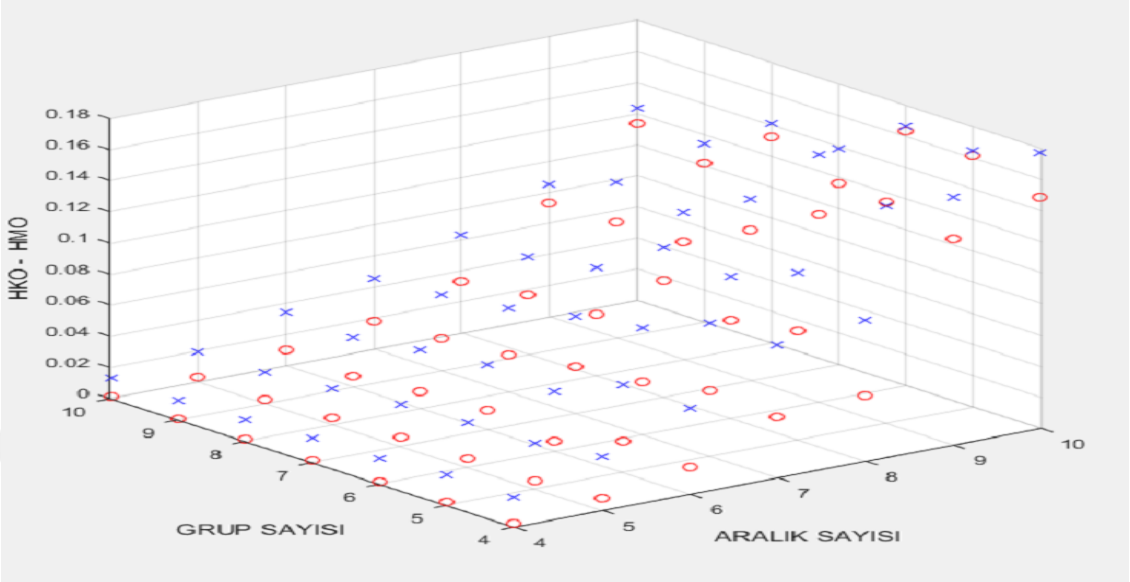
Şekil 4.2 Kinect Derinlik Kamerası - VK1 veri kümesi için Lagrange interpolasyon ile yöntemi modelleme sonucu

Kinect Derinlik Kamerası ile elde edilen VK1 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Sağ-El ekleminin hata sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



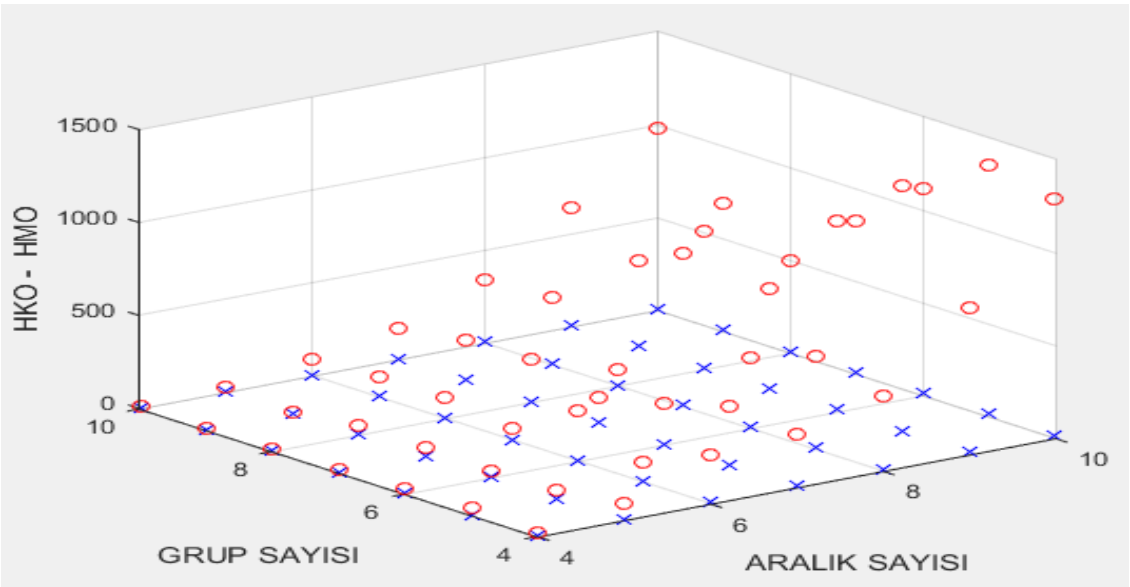
Şekil 4.3 Openpose - VK1 veri kümesi için Lagrange interpolasyon yöntemi ile modelleme sonucu

OpenPose ile elde edilen VK1 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Sağ-El ekleminin hata sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



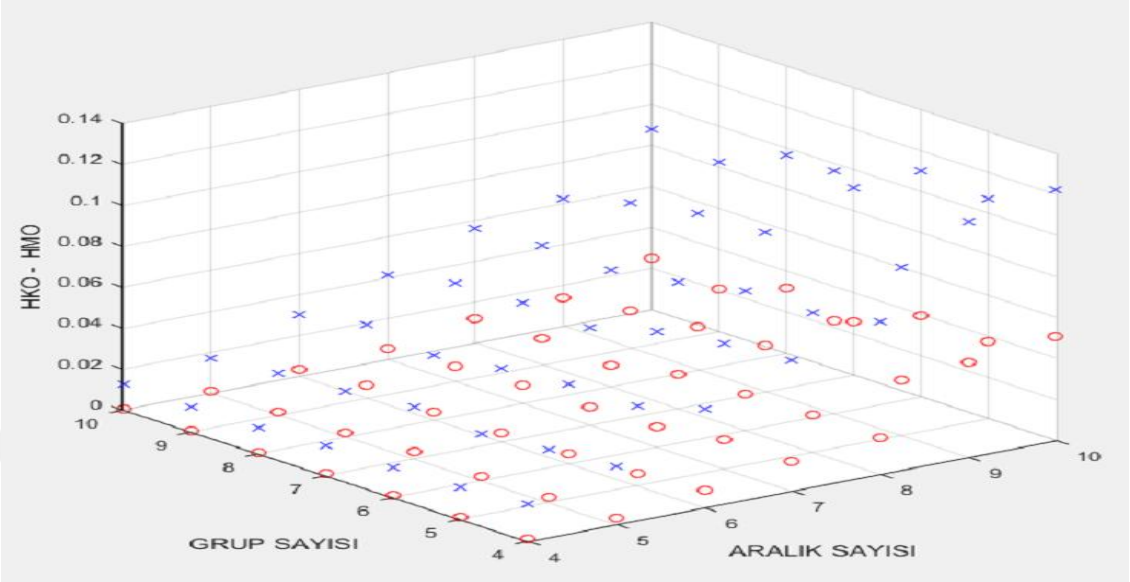
Şekil 4.4 Kinect Derinlik Kamerası - VK1 veri kümesi için Spline interpolasyon yöntemi ile modelleme sonucu

Kinect Derinlik Kamerası ile elde edilen VK1 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Sağ-El ekleminin sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



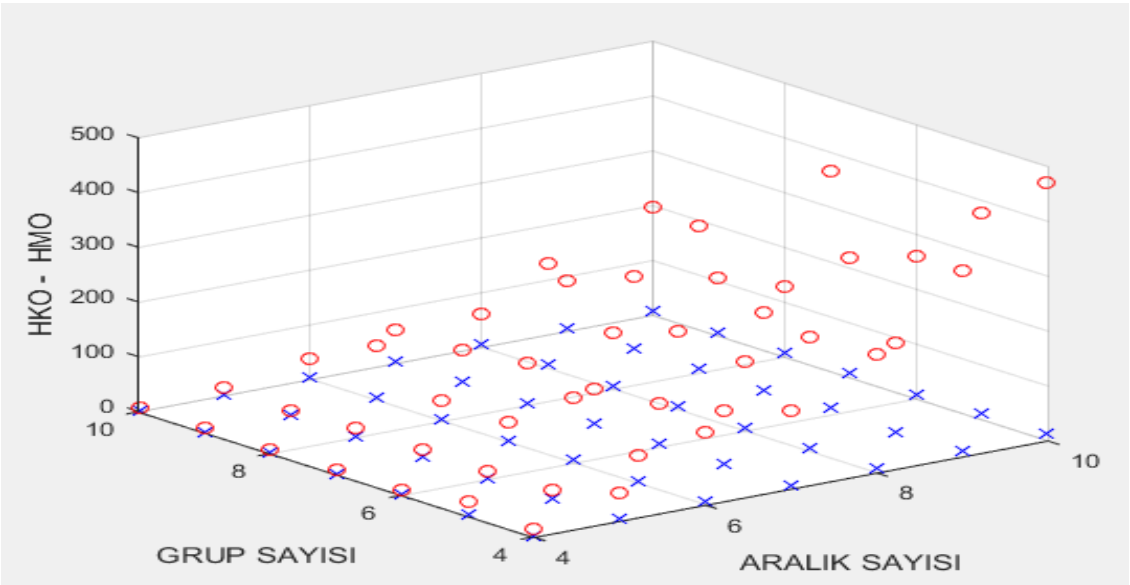
Şekil 4.5 Openpose - VK1 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu

OpenPose ile elde edilen VK1 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Sağ-El ekleminin sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



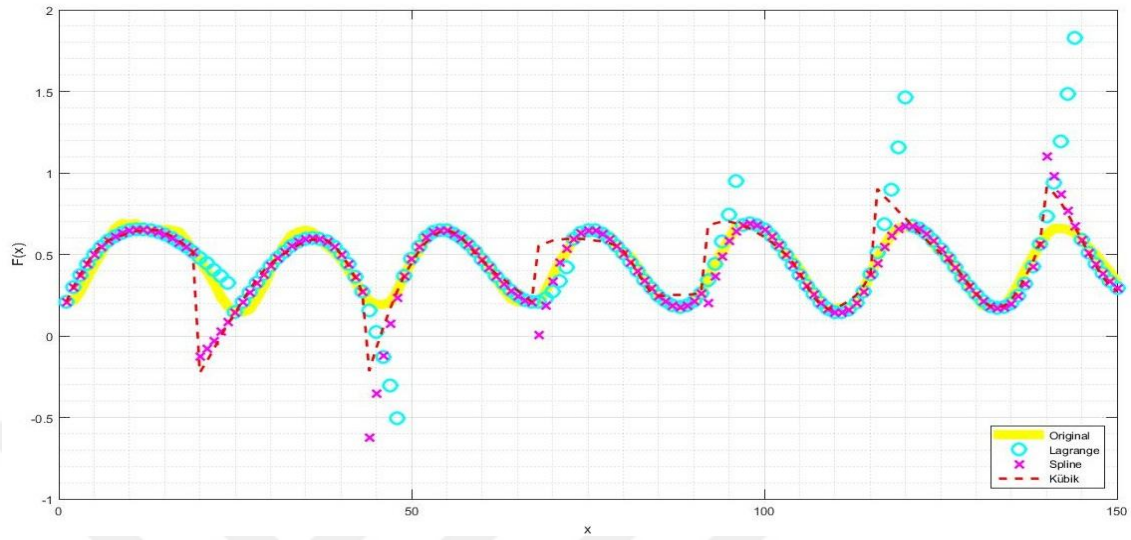
Şekil 4.6 Kinect Derinlik Kamerası - Vk1 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu

Kinect Derinlik Kamerası - VK1 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Sağ-El ekleminin hata sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



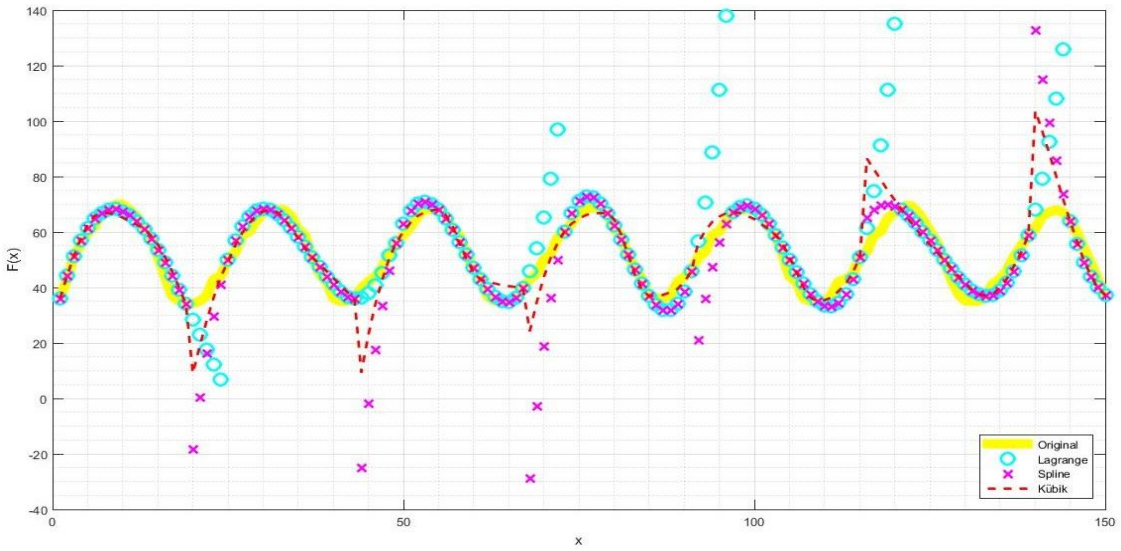
Şekil 4.7 Openpose - VK1 Veri Kümesi İçin Kübik Spline İnterpolasyon Yöntemi İle Modellenmesi Sonucu

OpenPose - VK1 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Sağ-El ekleminin hata sonuçlarının grafiksel gösterimi. o HKO, x MHO değerlerini göstermektedir.



Şekil 4.8 Kinect Derinlik Kamerası - VK1 Veri Kümesi İçin Grup Sayısı 4 Aralık Değeri 6 İken Tüm İnterpolasyon Yöntemleri İle Elde Edilen Modeller

Kinect Derinlik Kamerası ile elde edilen VK1 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sağ-El eklem verilerinin orijinal veri kümesi ile karşılaştırılması. Orijinal veri seti (sarı kalın çizgi), Kübik Spline (kesikli çizgi), Spline (mor çarpı) ve Lagrange (mavi yuvarlak) renkli eğrilerdir.



Şekil 4.9 OpenPose - VK1 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modeller

OpenPose ile elde edilen VK1 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sağ-El eklemi verilerinin orijinal veri kümesi ile karşılaştırılması. Orijinal veri seti (sarı kalın çizgi), Kübik Spline (kesikli çizgi), Spline (mor çarpı) ve Lagrange (mavi yuvarlak) renkli eğrilerdir.

Çizelge 4.1 de Kinect sensörleri ile aralık değeri 6 grup değeri 4 iken modellenen VK1 için 350 ile 357 kare numaraları arasındaki geri elde edilen Sağ-El eklem değerleri verilmiştir.

Çizelge 4.1 Kinect sensörleri ile aralık değeri 6 grup değeri 4 iken modellenen VK1

İnterpolasyon Adı / Kare No	**Orijinal**	**Lagrange**	**Spline**	**Kübik Spline**
350	0.26357	0,21706	0,21706	0,22075
351	0.34112	0,24026	0,24026	0,25194
352	0,38656	0,28164	0,28164	0,30129
353	0,44578	0,34339	0,34339	0,36663
354	0,51193	0,42767	0,42767	0,44581
355	0,53667	0,53667	0,53667	0,53667
356	0,55477	0,67255	1,1267	0,86993
357	0,55816	0,83749	0,96142	0,79313

Çizelge 4.2 de şekil 4.9 üzerinde görülen sıçramalardan bir tanesinin tahmin edilmiş verileri ve orijinal verileri yer almaktadır. Benzer şekilde tüm interpolasyon hesaplamaları ve orijinal veri kümesinin değeri 121. ve 127. karelerde eşittir. Bu karelerin orijinal değerinin hesaplamalarda kullanıldığını, tahmin edilmediğini göstermektedir. Orijinal değerlerin bulunduğu kare numaraları arasında fark 6'dır ve bu fark da tüm orijinal veri kare numaraları arasında sabit olup aralık değerini göstermektedir.

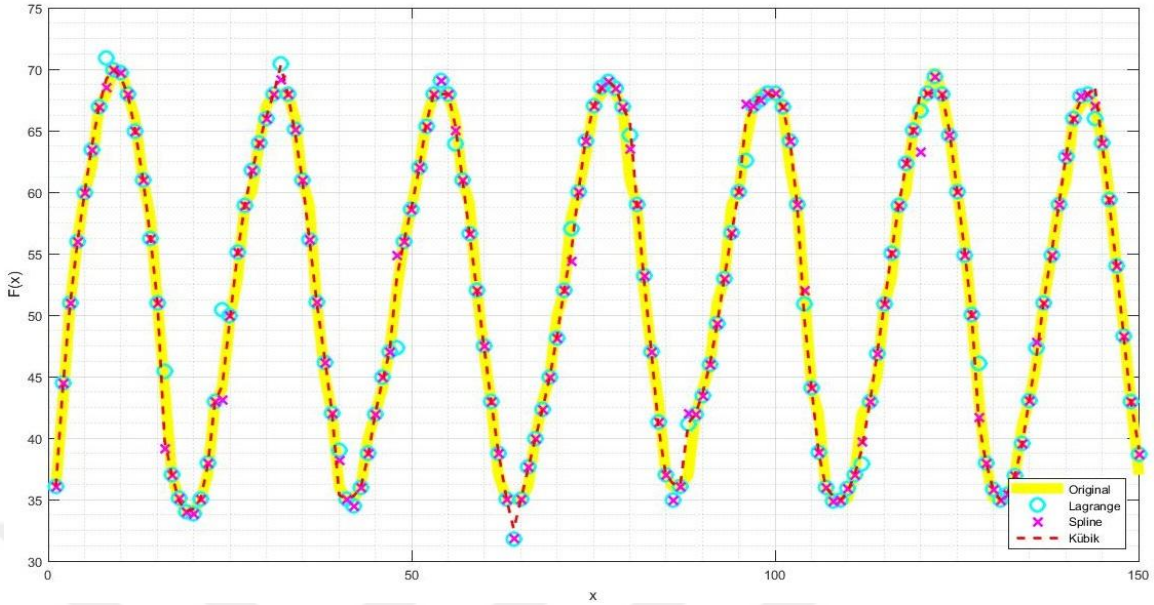
Çizelge 4.2 de OpenPose ile aralık değeri 6 grup değeri 4 iken modellenen VK1 için 120 ile 127 kare numaraları arasındaki geri elde edilen Sağ-El eklem değerleri verilmiştir.

Çizelge 4.2 OpenPose ile aralık değeri 6 grup değeri 4 iken modellenen VK1

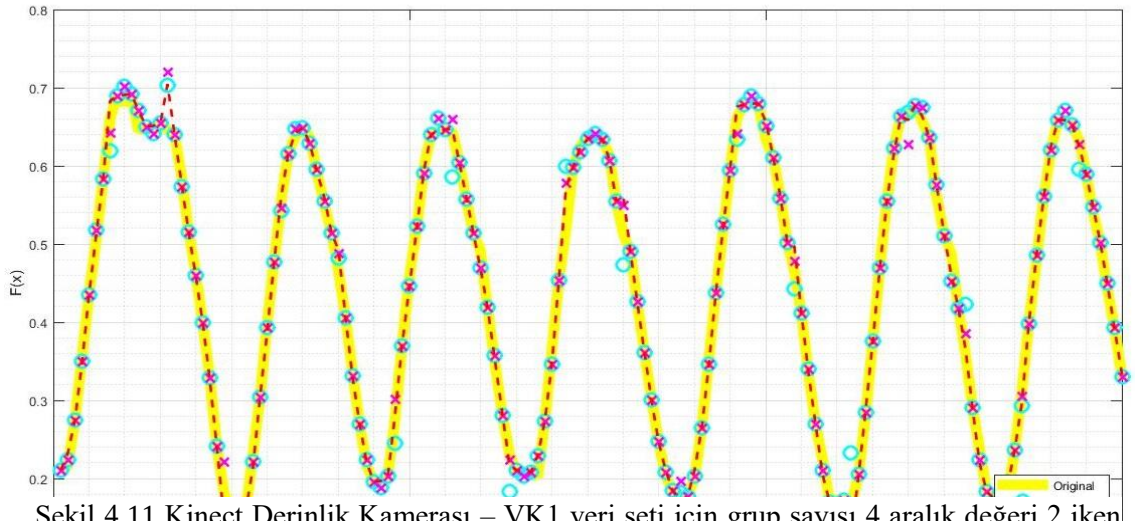
İnterpolasyon Adı / Kare No	**Orijinal**	**Lagrange**	**Spline**	**Kübik Spline**
120	67,0400	135,1103	69,4415	71,5368
121	68,0523	68,0523	68,0523	68,0523
122	69,9835	65,9686	65,9687	64,6849
123	67,9839	63,3297	63,3297	61,4459
124	65,0067	60,2744	60,2744	58,3467
125	60,0113	56,9418	56,9418	55,3985
126	55,9930	53,4710	53,4710	52,61288
127	50,0009	50,0009	50,0009	50,0009

Şekil 4.10 ve 4.11 üzerinde görüleceği üzere aralık değeri düşürüldüğünde kullanılmayan ve tahmin edilen veri sayısı azaldığı için meydana gelen hata oranları çok düşüktür ve sıçramalar neredeyse kalmamıştır.

Şekil 4.10 da OpenPose ve şekil 4.11 de Kinect Derinlik Kamerası ile elde edilen VK1 veri setleri için grup sayısı 4 aralık değeri 2 iken tüm interpolasyon yöntemleriyle elde edilen modellerden geri elde edilen Sağ-El eklemi verilerinin orjinal veri seti ile karşılaştırılması verilmiştir. Orjinal veri seti sarı kalın çizgiyle, Kübik Spline kesikli çizgiyle, Spline mor çarpıyla ve Lagrange mavi yuvarlaklar ile gösterilmektedir.



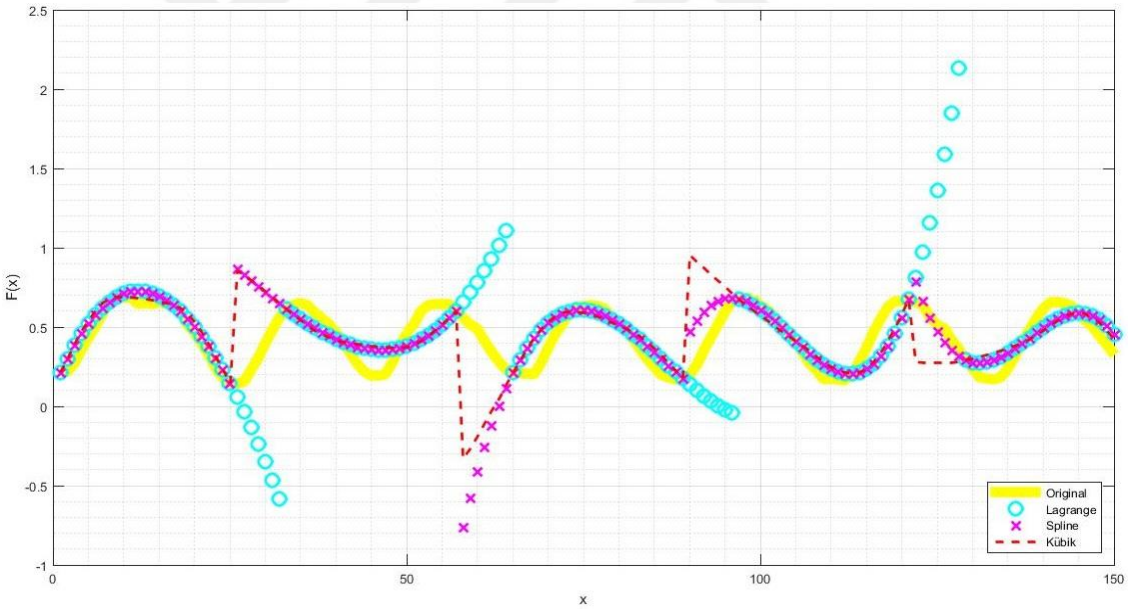
Şekil 4.10 Openpose – VK1 veri seti için grup sayısı 4 aralık değeri 2 iken tüm interpolasyon yöntemleri ile elde edilen modeller



Şekil 4.11 Kinect Derinlik Kamerası – VK1 veri seti için grup sayısı 4 aralık değeri 2 iken tüm interpolasyon yöntemleri ile elde edilen modeller

Hesaplamalar sonucu Kinect sensörleri ile elde edilen veri kümesinde Lagrange interpolasyonu için aralık değeri 2 grup sayısı 4 iken meydana gelen MHO değeri 0,00627 ve HKO değeri 0,00025 olmaktadır bu değerler Kübik Spline interpolasyonu için sırasıyla 0,00531 ve 0,00023 Spline interpolasyonu için 0,00040 ve 0,00675 olarak hesaplanmıştır.

Aralık sayısının azaltılması hata oranlarının düşmesini sağlayarak yapılan modellemenin doğruluğunu artırsa da bazı zorlukları meydana getirmektedir. Bu zorlukların başında işlem süresinin uzaması gelmektedir. Büyük veri kümeleri ile çalışıldığında işlem süresinin artmasının yanı sıra kullanılan sistem ihtiyaçlarını da artırmaktadır.



Şekil 4.12 Kinect Derinlik Kamerası - VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modeller

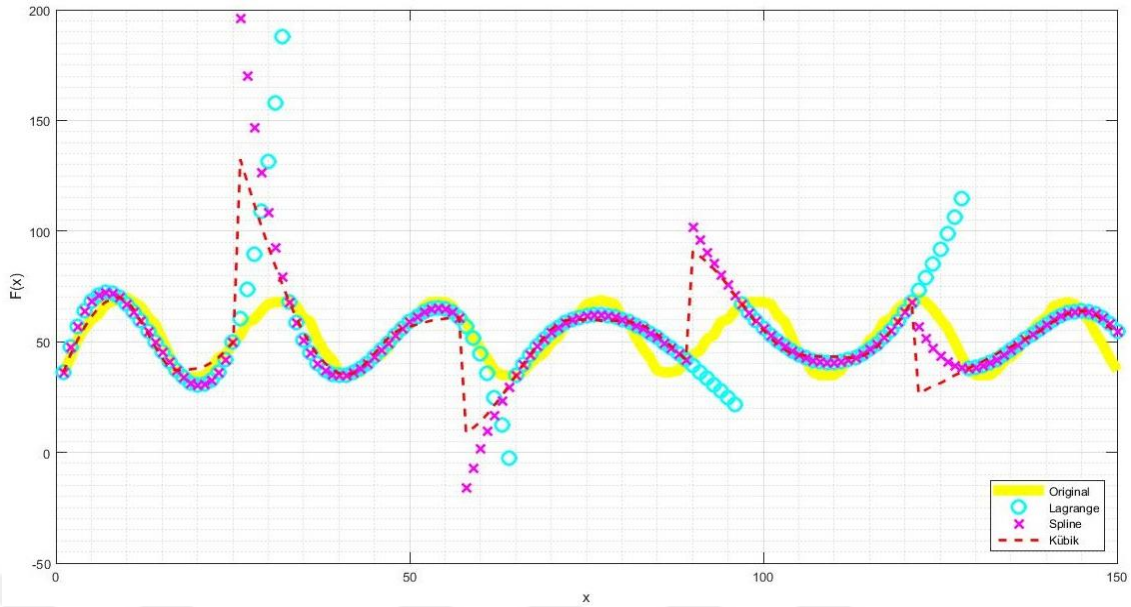
Kinect Derinlik Kamerası ile elde edilen VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Right-hand eklemi verilerinin orijinal veri kümesi ile karşılaştırılması. Orijinal veri seti (sarı kalın çizgi), Kübik Spline (kesikli çizgi), Spline (mor çarpı) ve Lagrange (mavi yuvarlak) renkli eğrilerdir.

Grup değişkeninin tek başına büyümesi ya da küçülmesinin hata oranları üzerinde herhangi bir etkisi olmamasına rağmen hesaplama süresi ve sistem ihtiyaçları üzerinde ciddi etki göstermektedir.

Şekil 4.12 üzerinde görülen hata miktarları aralık değerinin yüksek olması sebebi ile daha fazladır. Yapılan hesaplamalar sonucu HKO ve MHO değerleri sırasıyla Lagrange interpolasyonu için 0.10344 ve 0.14502, Spline interpolasyonu için 0.04195 ve 0,09049 ve Kübik Spline için 0.01777 ve 0,07449 olarak hesaplanmıştır.

Şekil 4.13 üzerinde yapılan hesaplamalar sonucu HKO ve MHO değerleri sırasıyla Lagrange interpolasyonu için 794.3384 ve 7.9438, Spline interpolasyonu için 9.9777 ve 1.2328 ve Kübik Spline için 6.568 ve 1.403 olarak hesaplanmıştır.

VK2 normal hızda eğilip kalkarak elde edilen veri kümesidir. Yukarı aşağı yönlü bir hareket meydana geldiği için hesaplamalar kafa noktası için elde edilen değerden yapılmıştır.

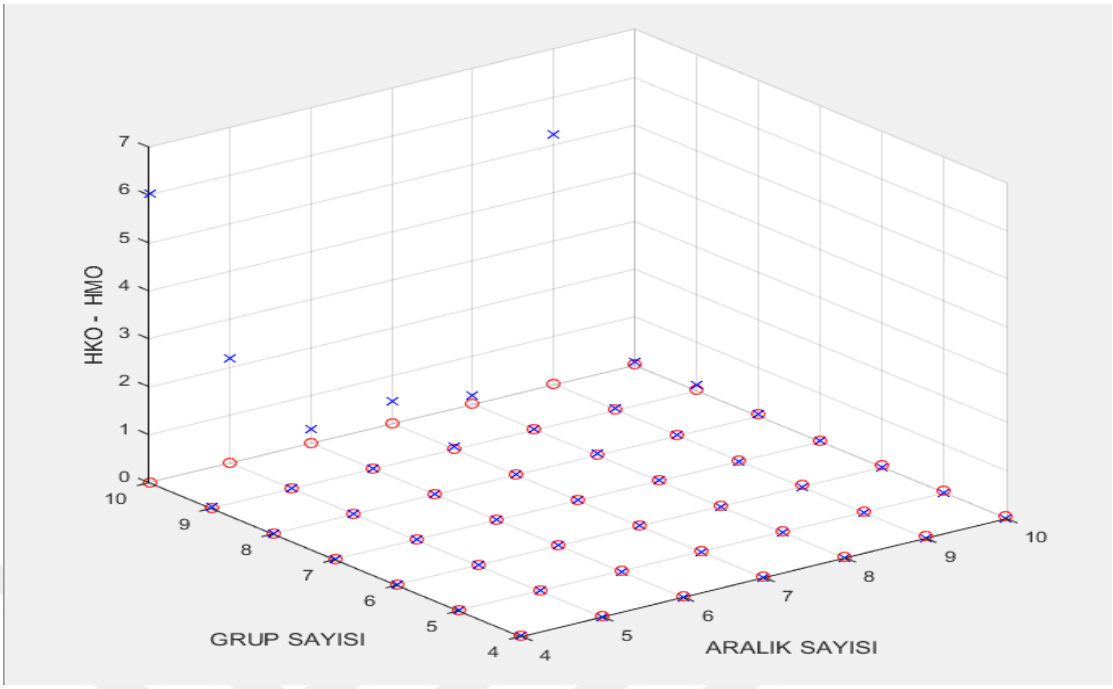


Şekil 4.13 Openpose - VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modeller

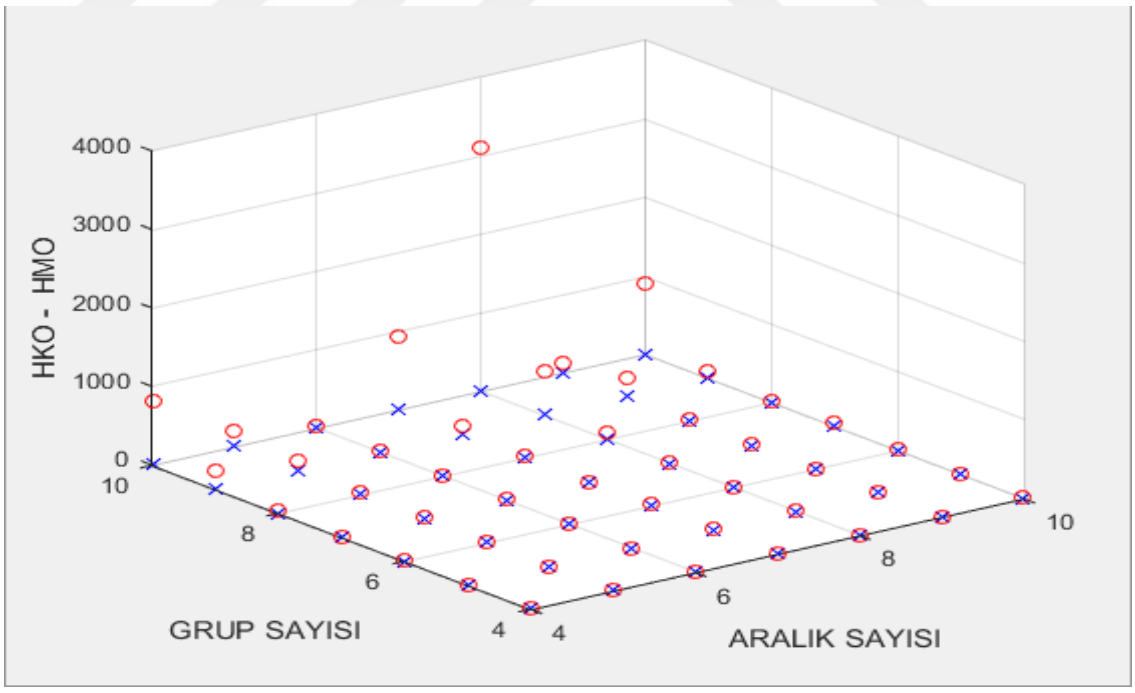
OpenPose ile elde edilen VK1 veri kümesi için grup sayısı 4 aralık değeri 8 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sağ-El eklemi verilerinin orijinal veri kümesi ile karşılaştırılması. Orijinal veri seti (sarı kalın çizgi), Kübik Spline (kesikli çizgi), Spline (mor çarpı) ve Lagrange (mavi yuvarlak) renkli eğrilerdir.

Şekil 4.14 de Kinect Derinlik Kamerası ile elde edilen ve şekil 4.15 de OpenPose ile elde edilen VK2 veri setleri için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir. o HKO, x HMO değerlerini belirtmektedir.

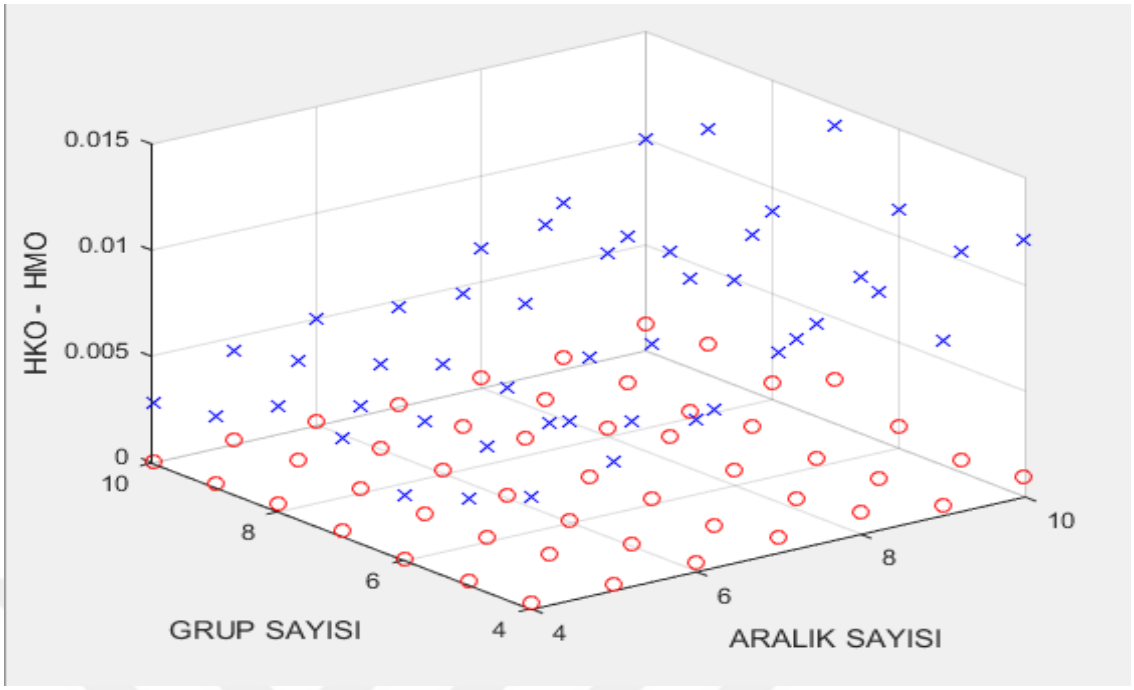
Şekil 4.16 da Kinect Derinlik Kamerası ile elde edilen ve şekil 4.17 de OpenPose ile elde edilen VK2 veri setleri için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir. o HKO, x HMO değerlerini belirtmektedir.



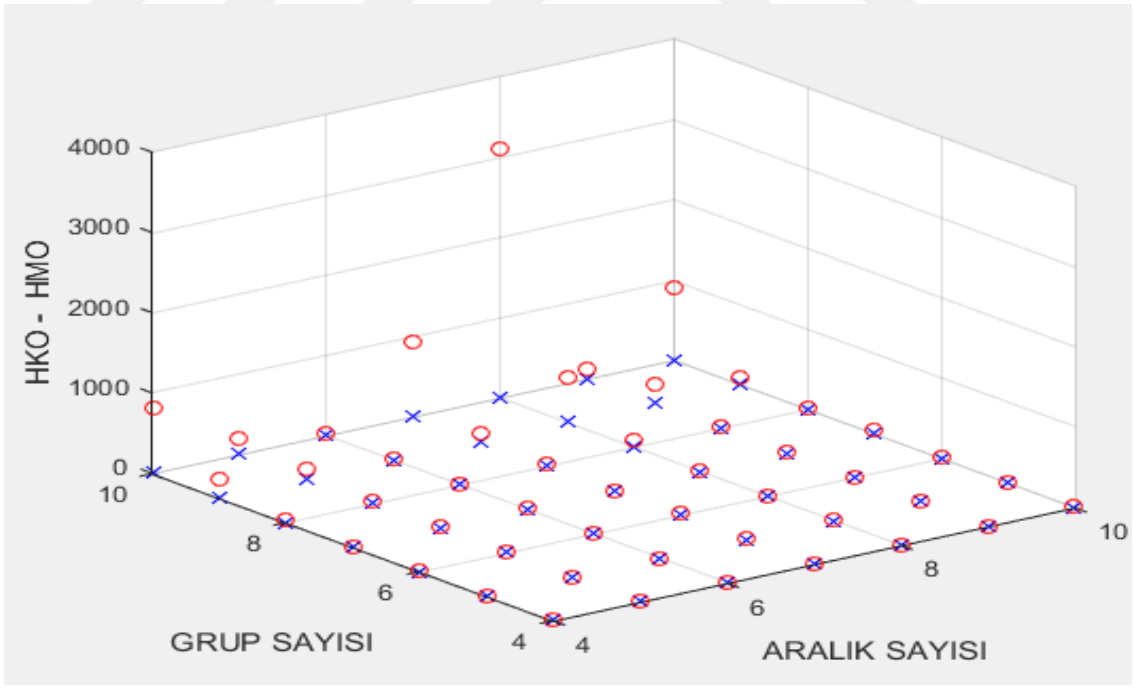
Şekil 4.14 Kinect Derinlik Kamerası - VK2 veri seti için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları



Şekil 4.15 Openpose - VK2 veri kümesi için lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları

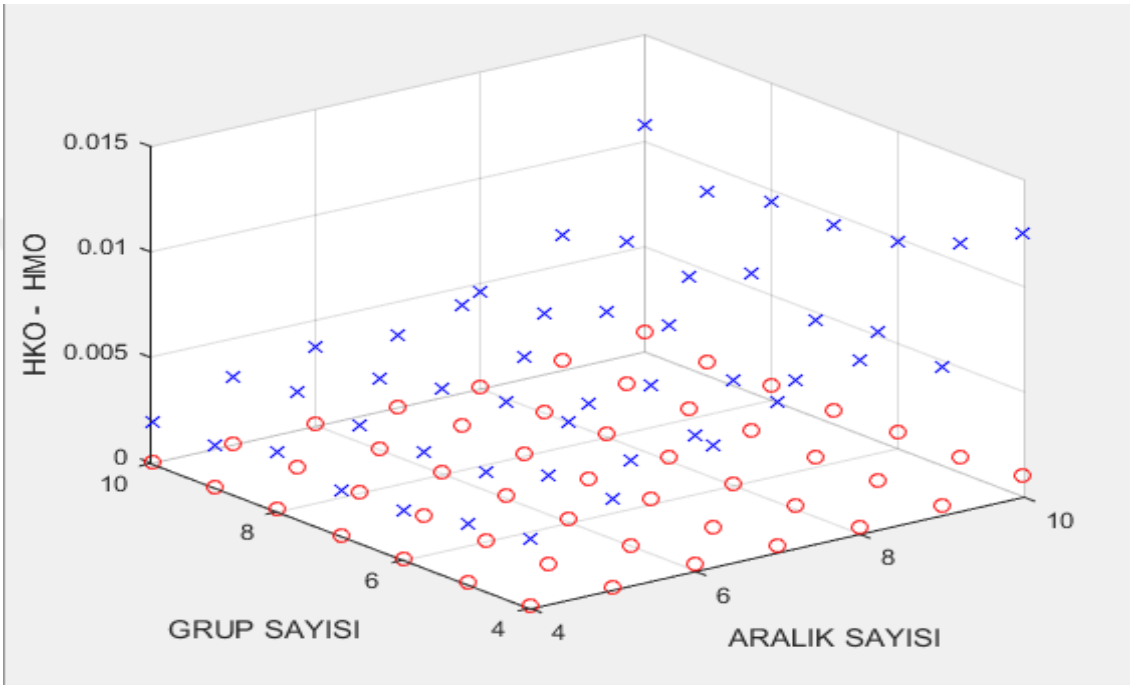


Şekil 4.16 Kinect Derinlik Kamerası - VK2 veri seti için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları



Şekil 4.17 Openpose - VK2 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları

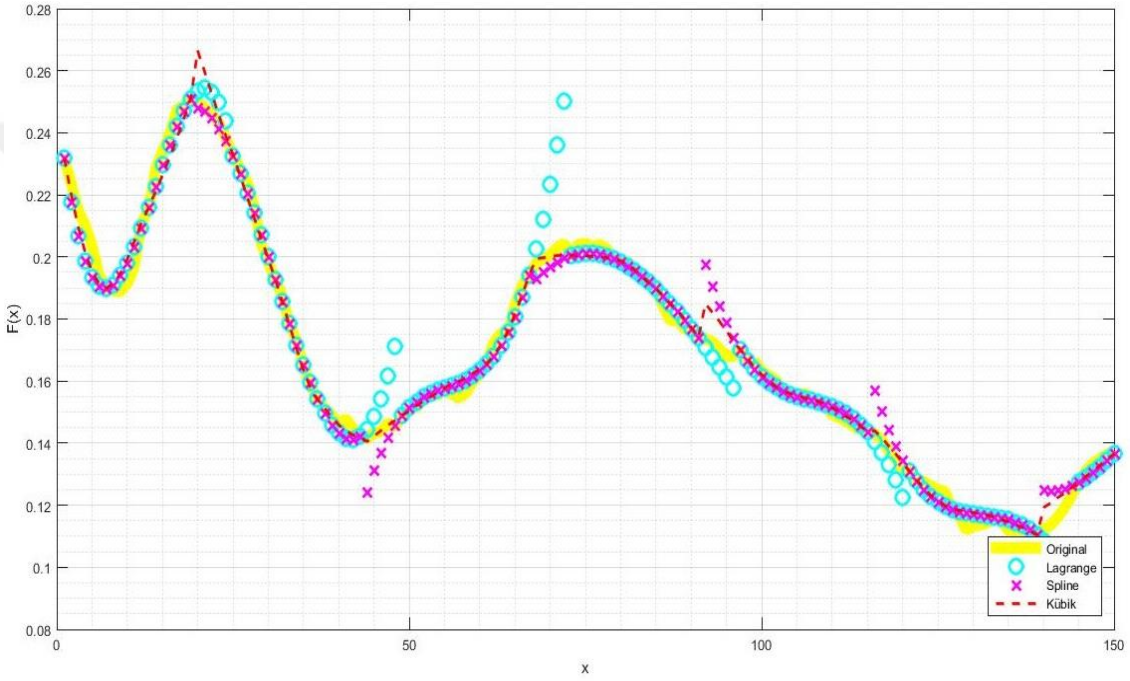
Şekil 4.18 de Kinect Derinlik Kamerası ile elde edilen VK2 veri seti için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir.. o HKO, x HMO değerlerini belirtmektedir



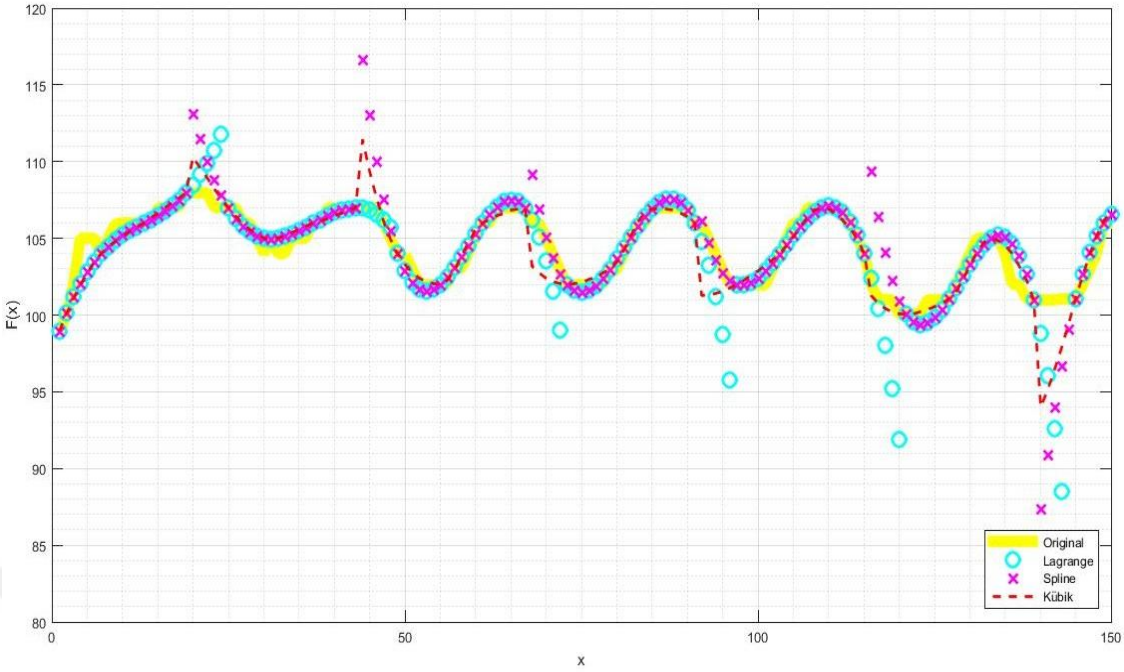
Şekil 4.18 Kinect Derinlik Kamerası - VK2 veri seti için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları

Çizelge 4.3 ve şekil 4.19 üzerinde görülen sıçramalardan bir tanesinin tahmin edilmiş verileri ve orijinal verileri yer almaktadır. Benzer şekilde Lagrange ve Spline interpolasyon yöntemlerinin elde ettiği veriler birbirlerine daha yakın olmasına rağmen Kübik Spline interpolasyonu orijinal veri değerlere daha yakınlaşmıştır. Aralık değeri 6 olduğu için 43. karenin orijinal değerinin hesaplamalarda kullanıldığını, tahmin edilmediğini göstermektedir.

Şekil 4.19 da Kinect Derinlik Kamerası ve şekil 4.20 de OpenPose ile elde edilen VK2 veri setleri için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklem verilerinin orijinal veri seti ile karşılaştırılması gösterilmektedir. Orijinal veri seti sarı kalın çizgiyle, Kübik Spline kesikli çizgiyle, Spline mor çarpıyla ve Lagrange mavi yuvarlak renkli eğriler ile ifade edilmektedir.



Şekil 4.19 Kinect Derinlik Kamerası - VK2 veri seti için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri seti ile karşılaştırılması.



Şekil 4.20 Openpose - VK2 veri seti için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri seti ile karşılaştırılması.

Çizelge 4.3 Kinect sensörleri ile aralık değeri 6 grup değeri 4 iken modellenen VK2 için 40 ile 47 kare numaraları arasındaki geri elde edilen Kafa eklem değerleri

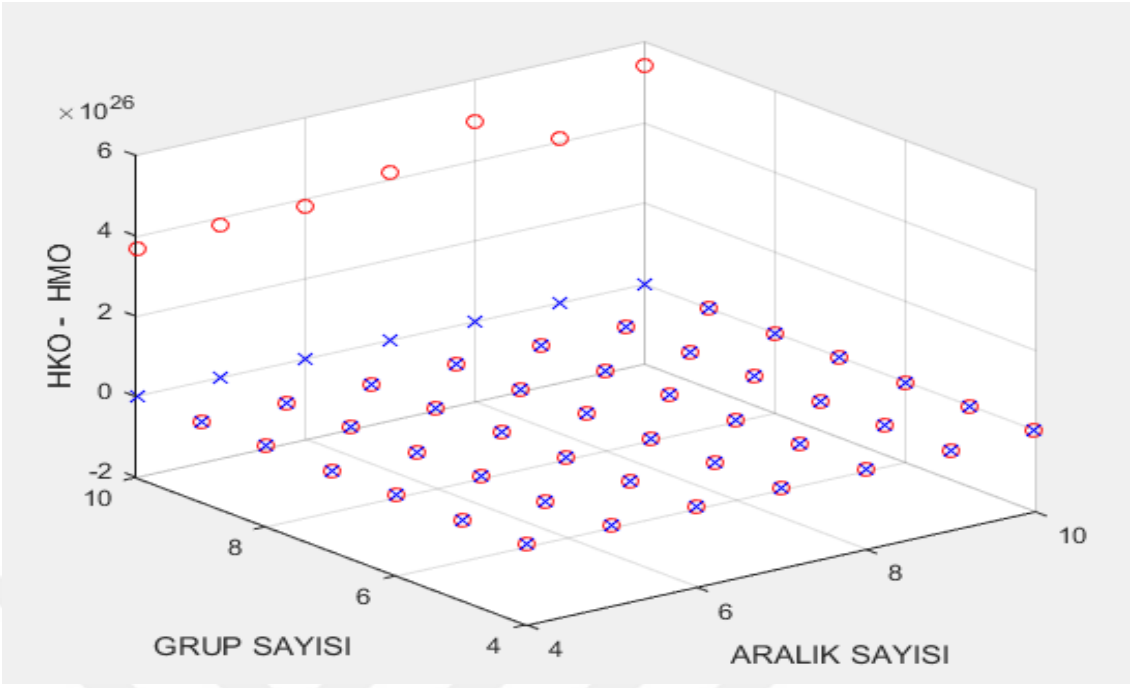
İnterpolasyon Adı / Kare No	**Lagrange**	**Spline**	**Kübik Spline**	**Orijinal**
40	0,14319	0,14319	0,14584	0.14575
41	0,14159	0,14159	0,14389	0.14773
42	0,14120	0,14120	0,14261	0,14479
43	0,14215	0,14215	0,14215	0,14215
44	0,14455	0,12434	0,14053	0,14164
45	0,14852	0,13118	0,14394	0,14317
46	0,15418	0,14174	0,14561	0,14374
47	0,16165	0,14568	0,14725	0,14635

VK3 eşli Bachata dansı yapılarak elde edilen veri kümesidir. Bir dans performansının baştan sona modellenmesi ile elde edilmiştir. İki iskelet takibi yapılmış olup hesaplamalar bir iskelet verisi üzerinden yapılmıştır. Bu veri kümesi şimdiye kadar testleri yapılmış olan diğer veri kümelerinden daha büyük bir hacme sahiptir. Veri kümeleri büyüklüğünün geliştirilen algoritma üzerinde herhangi bir sorun çıkarmamıştır. Yalnızca hesaplama süresi üzerinde farklılıklar meydana gelmiştir.

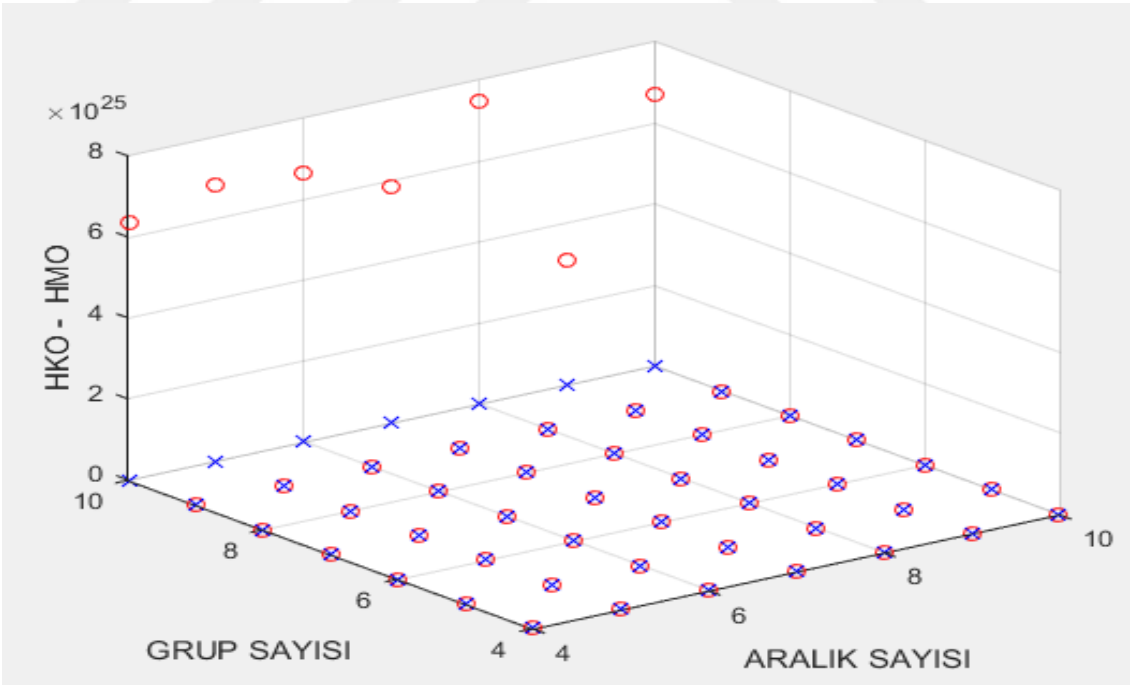
Şekil 4.21 da Kinect Derinlik Kamerası ve şekil 4.22 de OpenPose ile elde edilen VK3 veri kümeleri için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir. o HKO, x MHO değerlerini ifade etmektedir.

Şekil 4.23 de Kinect Derinlik Kamerası ve şekil 4.24 de OpenPose ile elde edilen VK3 veri kümeleri için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir. o HKO, x MHO değerlerini ifade etmektedir.

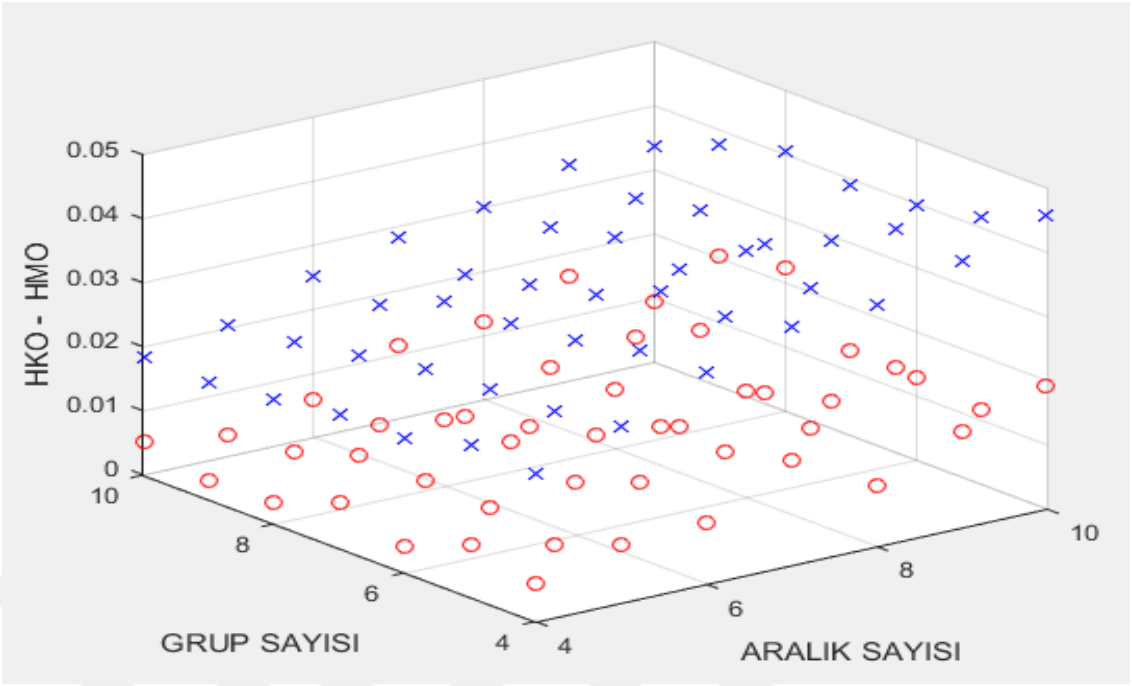
Şekil 4.25 de Kinect Derinlik Kamerası ve şekil 4.26 da OpenPose ile elde edilen VK3 veri kümeleri için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları grafiksel olarak gösterilmektedir. o HKO, x MHO değerlerini ifade etmektedir.



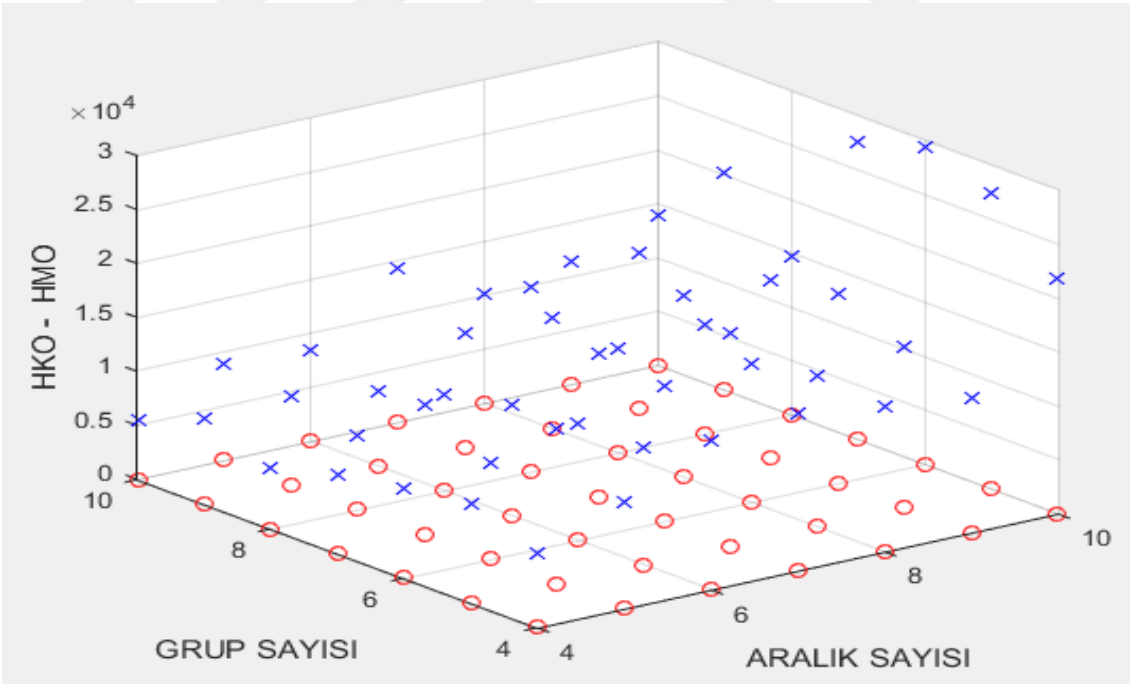
Şekil 4.21 Kinect Derinlik Kamerası - VK3 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa ekleminin hata sonuçları



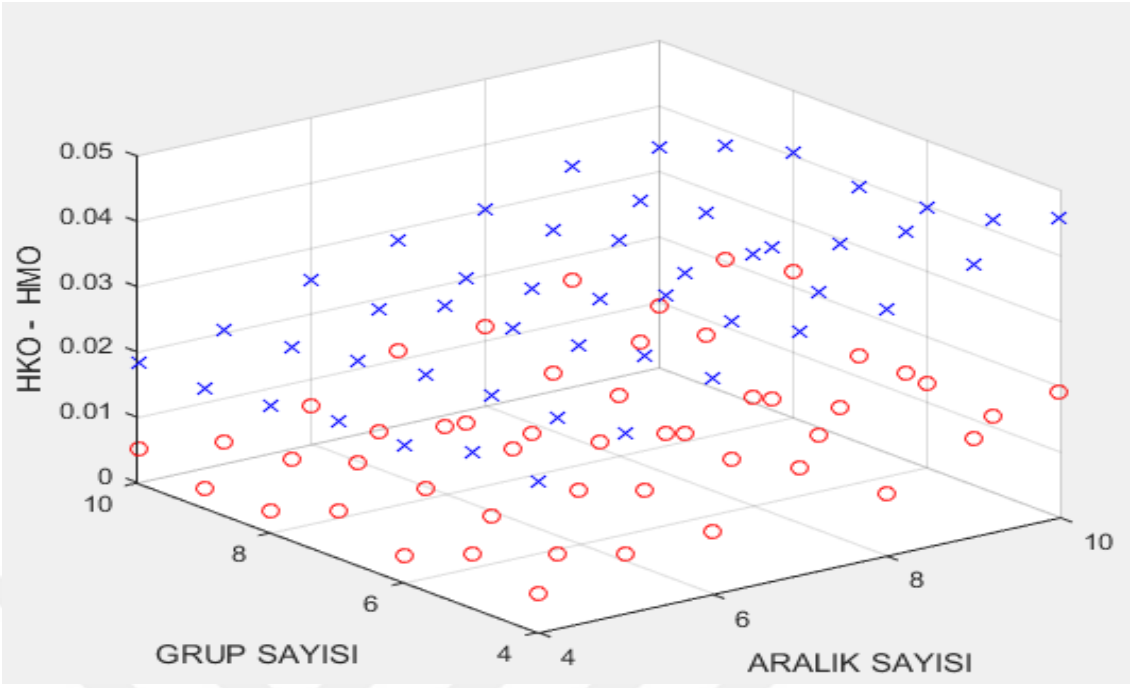
Şekil 4.22 Openpose - VK3 veri kümesi için Lagrange interpolasyon yöntemi ile modellenmesi sonucu meydana gelen Kafa Ekleminin Hata Sonuçları



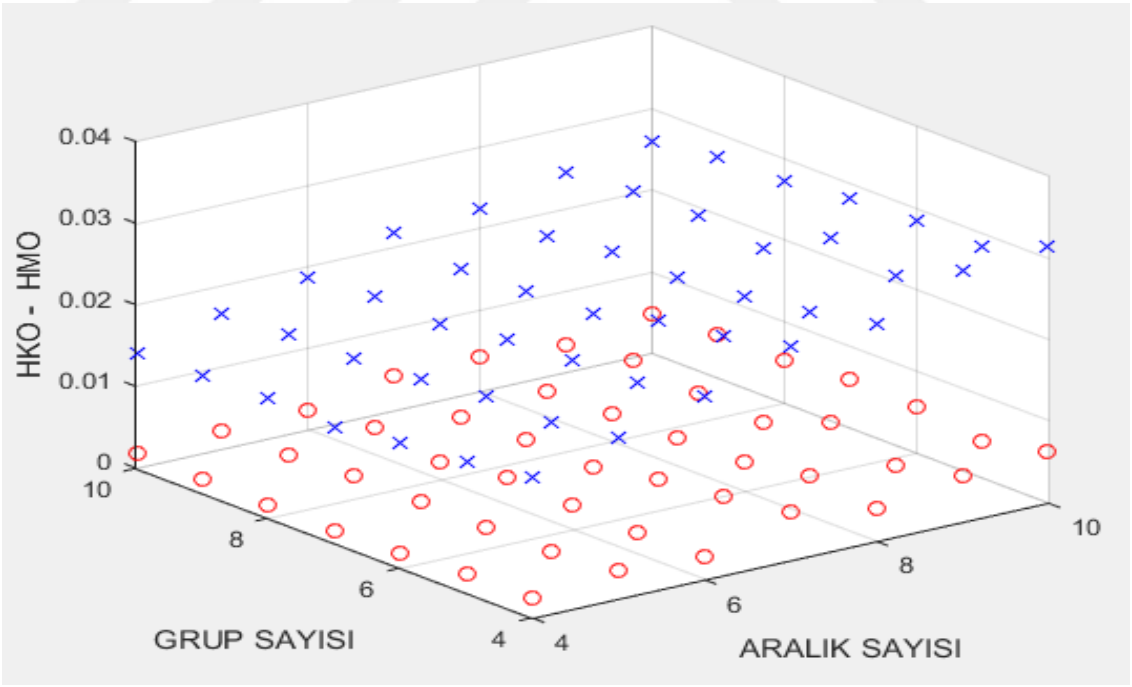
Şekil 4.23 Kinect Derinlik Kamerası - VK3 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları



Şekil 4.24 Openpose - VK3 veri kümesi için Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları



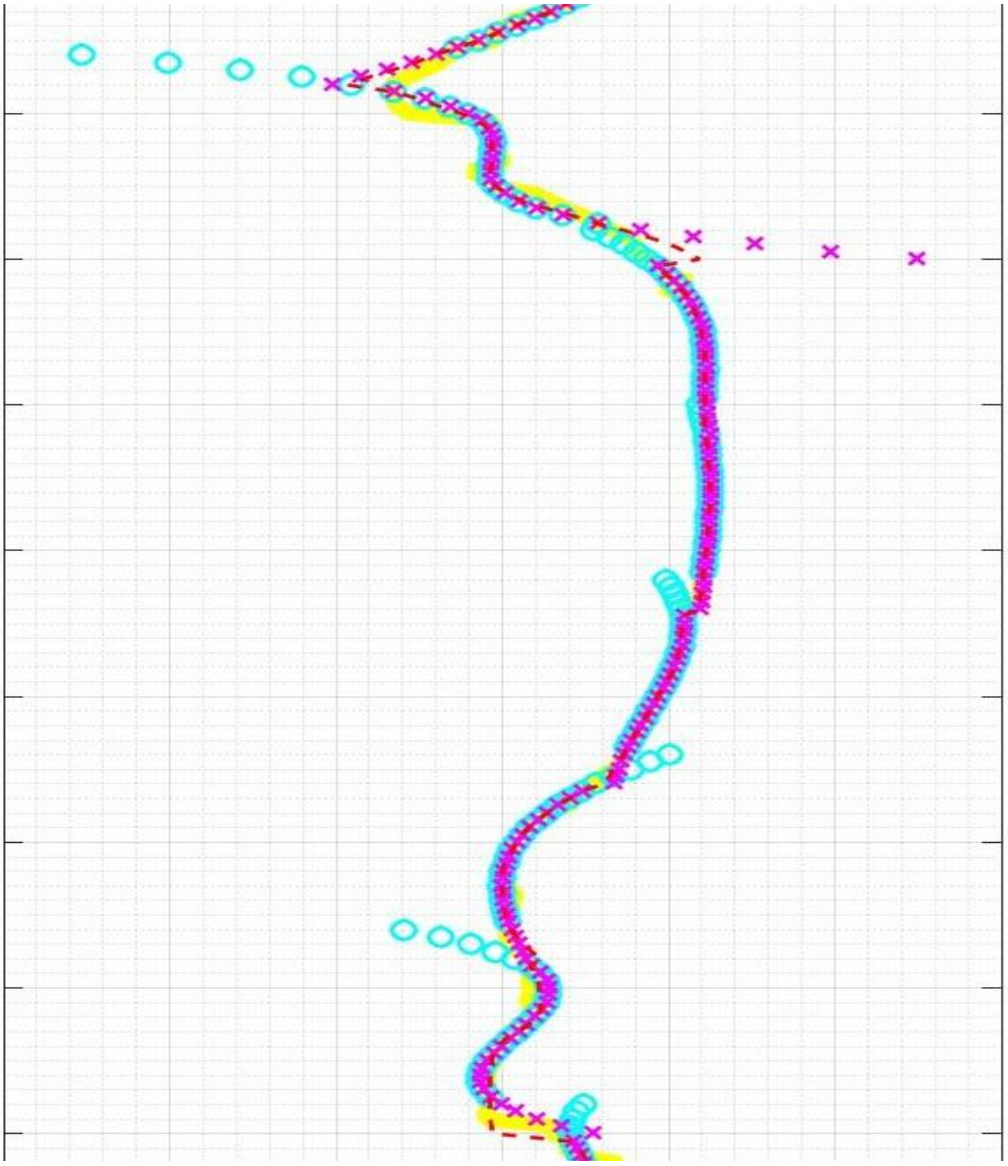
Şekil 4.25 Kinect Derinlik Kamerası - VK3 veri kümesi için Kübik Spline interpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları



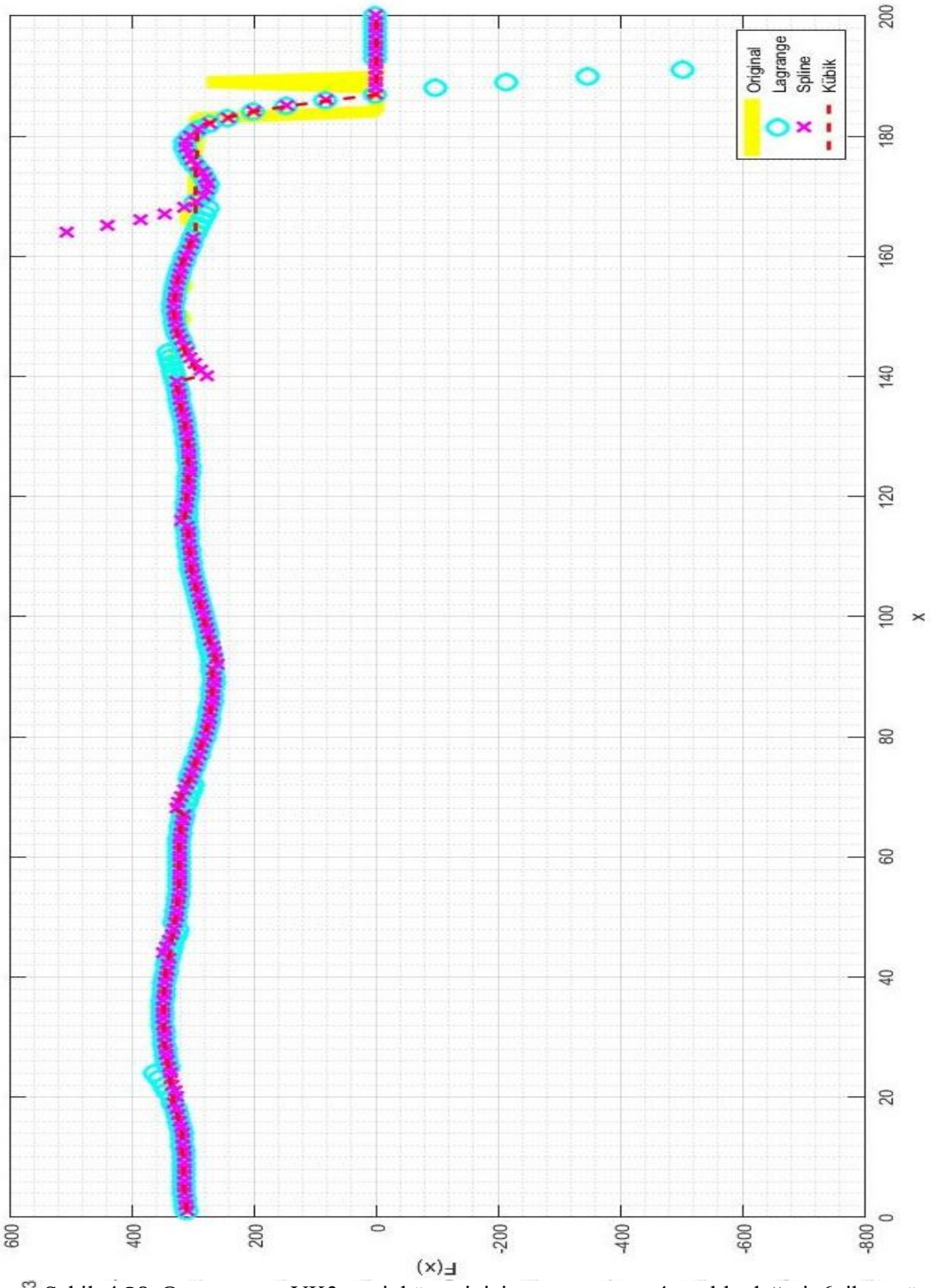
Şekil 4.26 Openpose -VK3 veri kümesi için Kübik Spline İnterpolasyon yöntemi ile modellenmesi sonucu meydana gelen hata Kafa ekleminin sonuçları

Şekil 4.27 de Kinect Derinlik Kamerası ve şekil 4.28 de OpenPose ile elde edilen VK3 veri kümeleri için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklem verilerinin orijinal veri kümesi ile karşılaştırılması gösterilmektedir. Orijinal veri seti sarı kalın çizgiyle, Kübik Spline kesikli çizgiyle, Spline mor çarpıyla ve Lagrange mavi yuvarlak ile ifade edilmektedir.

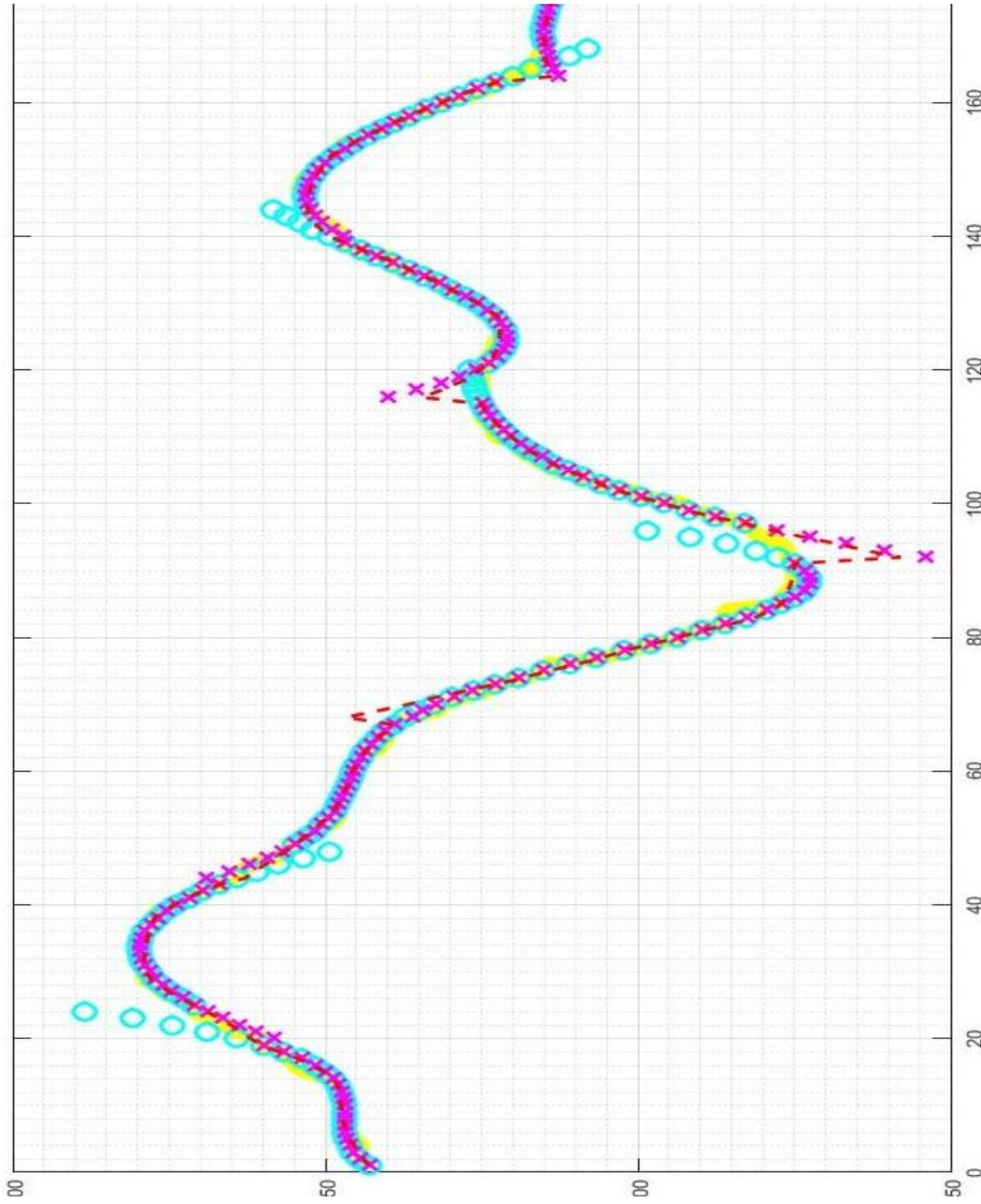
Şekil 4.29 de Kinect Derinlik Kamerası ve şekil 4.30 da OpenPose ile elde edilen VK3 veri kümeleri için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sol-omuz eklem verilerinin orijinal veri kümesi ile karşılaştırılması gösterilmektedir. Orijinal veri seti sarı kalın çizgiyle, Kübik Spline kesikli çizgiyle, Spline mor çarpıyla ve Lagrange mavi yuvarlak ile ifade edilmektedir.



Şekil 4.27 Kinect Derinlik Kamerası - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklemi verilerinin orijinal veri kümesi ile karşılaştırılması.

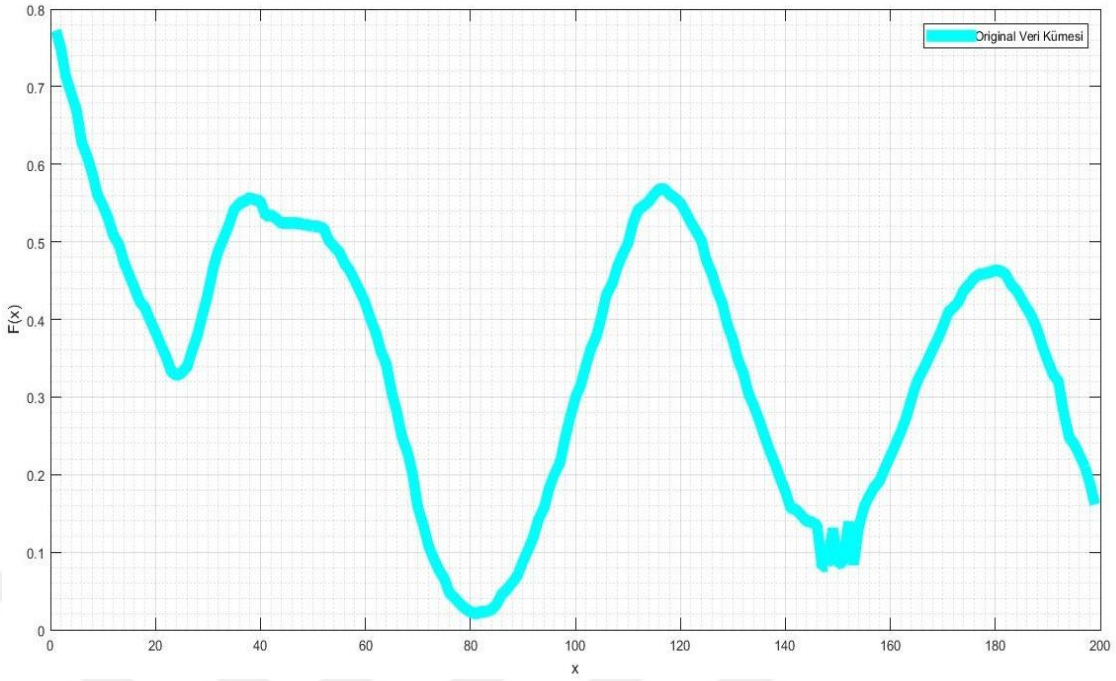


Şekil 4.28 Openpose - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Kafa eklem verilerinin oriŞekil 4.29 Kinect Derinlik Kamerası - VK3 veri kümesi için grup sayısı 4 aralık değeri 6 iken tüm interpolasyon yöntemleri ile elde edilen modellerden geri elde edilen Sol-omuz eklem verilerinin orijinal veri kümesi ile karşılaştırılması



Şekil 4.30 Openpo
yöntemleri ile eld
kümesi ile karşıla

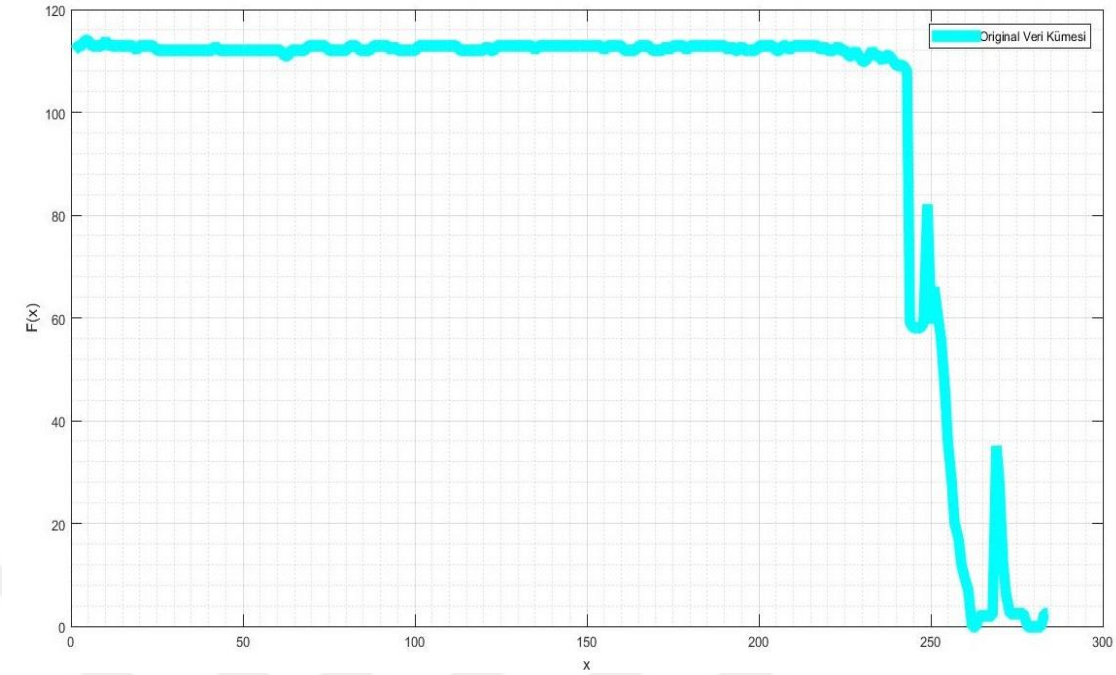
VK4 Havada sağ el ile 8 şekli oluşturulması esnasında modellenen veri kümesidir. 8 şeklinin seçilmesindeki amaç hareketin başlangıç ve bitiş noktasının aynı olmasıdır. Bu hareketin OpenPose algoritması ve Microsoft Kinect sensörleri ile modellemesi sonucunda elde edilen verilerin grafik üzerinde dalgalanması beklenmektedir. Ayrıca bu hareketin, iki yöntemin modellenmesi arasında herhangi bir farklılık olup olmadığını anlamak için uygun olduğu düşünülmüştür.



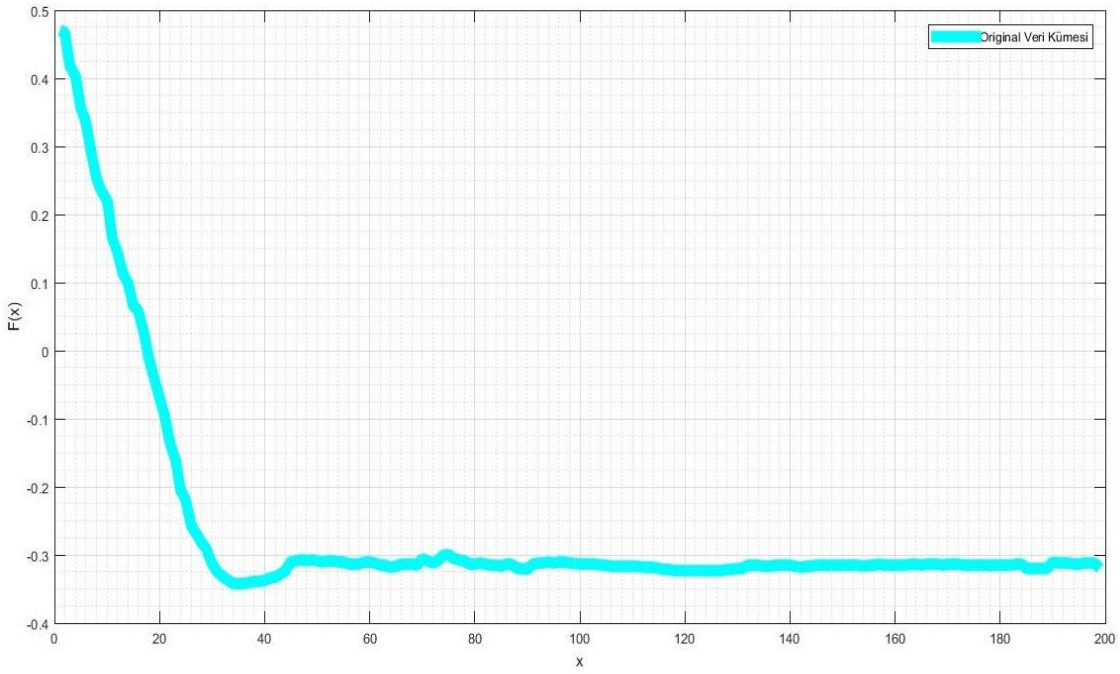
Şekil 4.31 Kinect ile elde edilen VK4 veri kümelesinin Sağ-El eklem değerleri

OpenPose algoritması ve Microsoft Kinect sensörlerinin eklem konum değer tahminleri sonucu hesapladığı değerlerin aralıkları aynı değildir. Microsoft Kinect sensörlerinin hesapladığı eklem değerleri -1 ile 1 aralığında olmasına rağmen, OpenPose algoritması ile bu değerler 100'den büyük olabilmektedir.

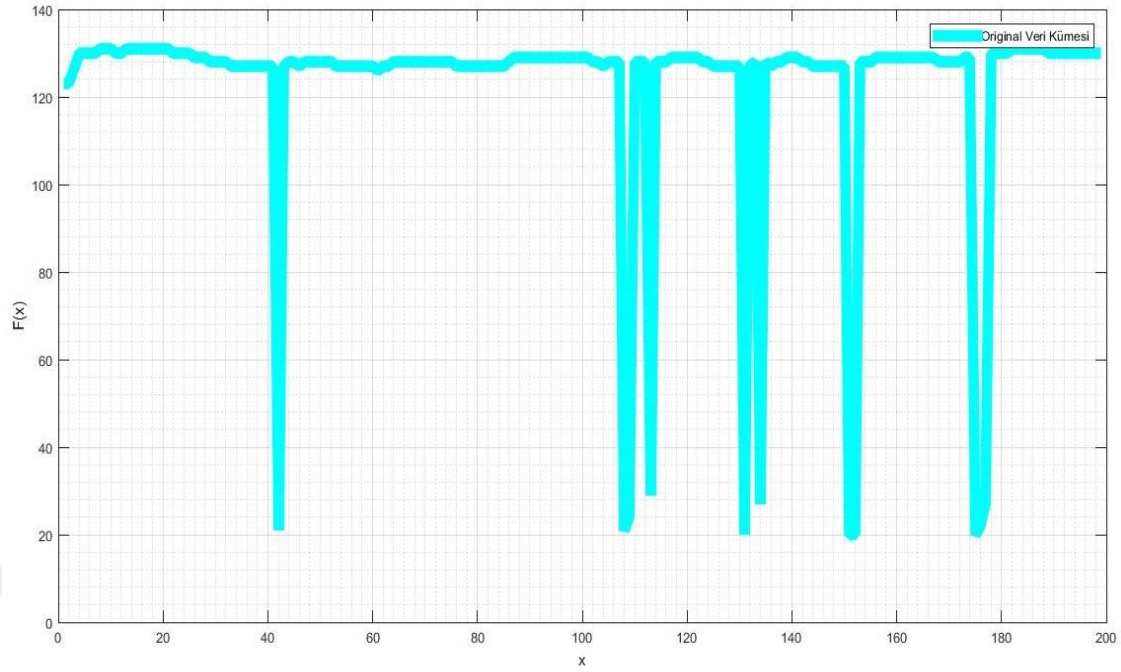
Bu durum grafikler üzerinde yanıltıcı olsada, elde edilen örüntülerin benzer yapıda olması beklenmektedir. Ancak OpenPose algoritması ve Microsoft Kinect sensörleri ile takip edilen ve elde edilen veri kümelerinin başarısı farklı eklemlerde farklı sonuçlar meydana getirmektedir.



Şekil 4.32 OpenPose ile elde edilen VK4 veri kümesinin Sağ-El eklem değerleri



Şekil 4.33 Kinect ile elde edilen VK4 veri kümesinin Sol-El eklem değerleri



Şekil 4.34 OpenPose ile elde edilen VK4 veri kümesinin Sol El eklem değerleri

Şekil 4.30, 4.31, 4.32 ve 4.33 eklemlerin değerlerinin düzlemsel değişimini göstermektedir. Yapılan hareketlerde takip edilen eklemlerin, her bir karede bulundukları konumu grafikler üzerinde görselleştirilmiştir. Sağ el eklem değişikliklerinin başlangıç ve bitiş noktalarının birbirlerine yakın oldukları bu grafiklerde görülmektedir. Hareket sırasında sol el eklem değişikliklerinin daha az olduğu da ayrıca görülmektedir.

OpenPose algoritmasının saniye başına işlediği kare sayısı ile Microsoft Kinect sensörünün saniye başına işlediği kare sayısı farklı olduğundan iki yöntemin toplamda işlediği kare sayısı birbirlerinden farklıdır. Ancak meydana gelen desenler birbirlerine oldukça benzemektedir. Bu yüzden OpenPose algoritması ile Microsoft Kinect sensörlerinin hareket modellemesinin sonuçlarını birbirlerine yakın olduğu düşünülebilir.

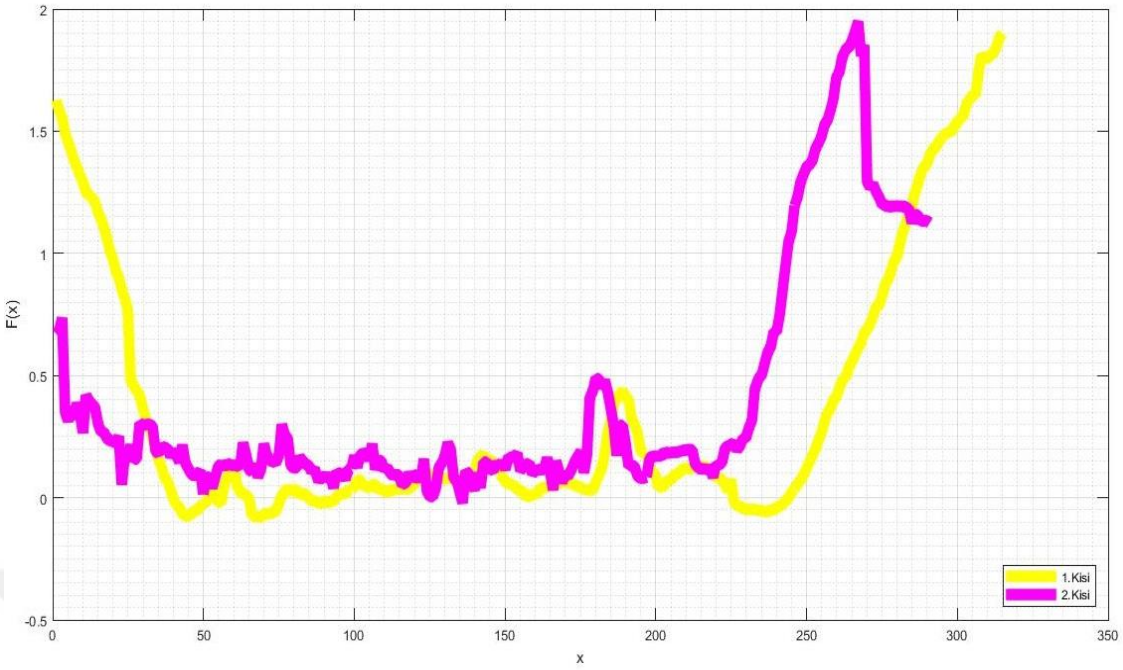
VK5 Boks hareketleri ile yapılan dans sonucu elde edilen veri kümesidir. Bu dans koreografisi 4 adım ileri geri zıplama, sol-ileri yöne 1 sağ yumruk devamında 2 adım ileri geri zıplama ve son olarak sağ-ileri yöne 1 sol yumruk atılması olarak belirlenmiştir.

Modellemelerin yapıldığı alanda, öncelikle bir nokta belirlenmiş ve işaretlenmiştir. 2 farklı kişide aynı nokta üzerinde danslarını gerçekleştirmiştir. Bu noktanın belirlenmesinin nedeni Microsoft Kinect sensörü ve OpenPose algoritması için kayıt yapan cep telefonun bulunduğu noktaya, modellemesi yapılacak olan 2 kişinin aynı uzaklıkta bulunmasını sağlamaktır.

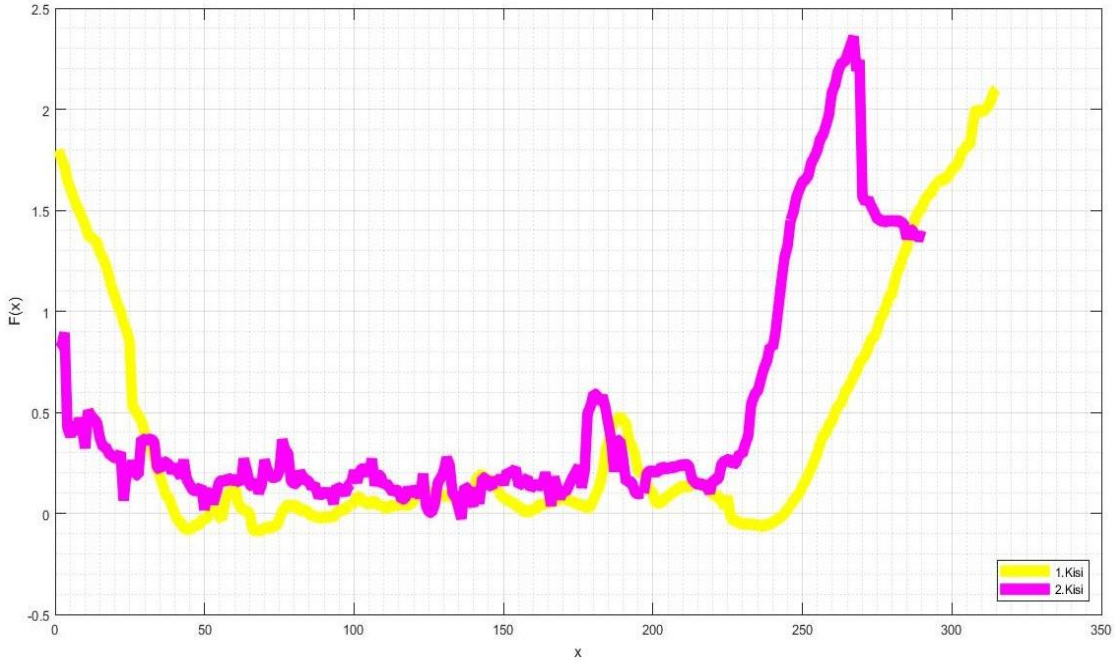
Modellemesi yapılan 1. Kişinin boy uzunluğu 181 cm, 2. Kişinin ise 165 cm olarak ölçülmüştür. Boy uzunluklarının modellemelere etkisinin olmaması için 200 cm uzunluğuna göre veriler normalize edilmiştir.

Kinect ile elde edilen VK5 veri kümesine normalizasyon işlemi uygulanmadan önceki şekil 4.35 de ve normalizasyon işlemi uygulandıktan sonraki şekil 4.36 da, 1. ve 2. Kişinin Kalça-Merkez eklem değerleri gösterilmiştir. Sarı eğri 1. kişinin ve pembe eğri ise 2. kişinin meydana getirdikleri eğriyi göstermektedir.

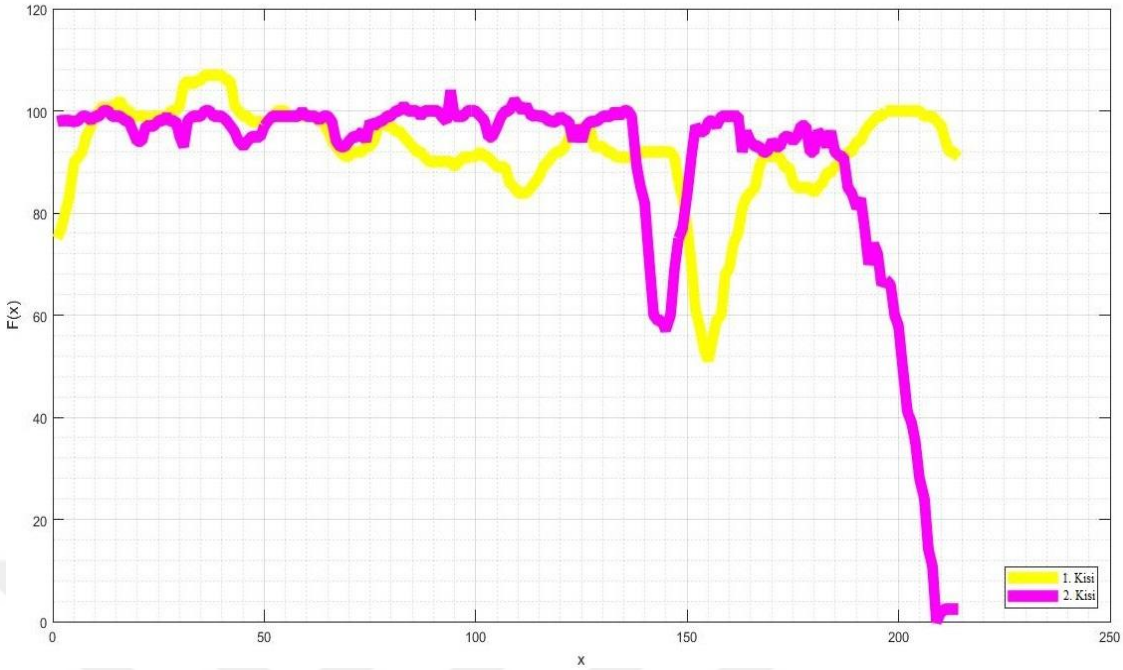
OpenPose ile elde edilen VK5 veri kümesine normalizasyon işlemi uygulanmadan önceki şekil 4.37 de ve normalizasyon işlemi uygulandıktan sonraki şekil 4.38 de, 1. ve 2. Kişinin Kalça-Merkez eklem değerleri gösterilmiştir. Sarı eğri 1. kişinin ve pembe eğri ise 2. kişinin meydana getirdikleri eğriyi göstermektedir.



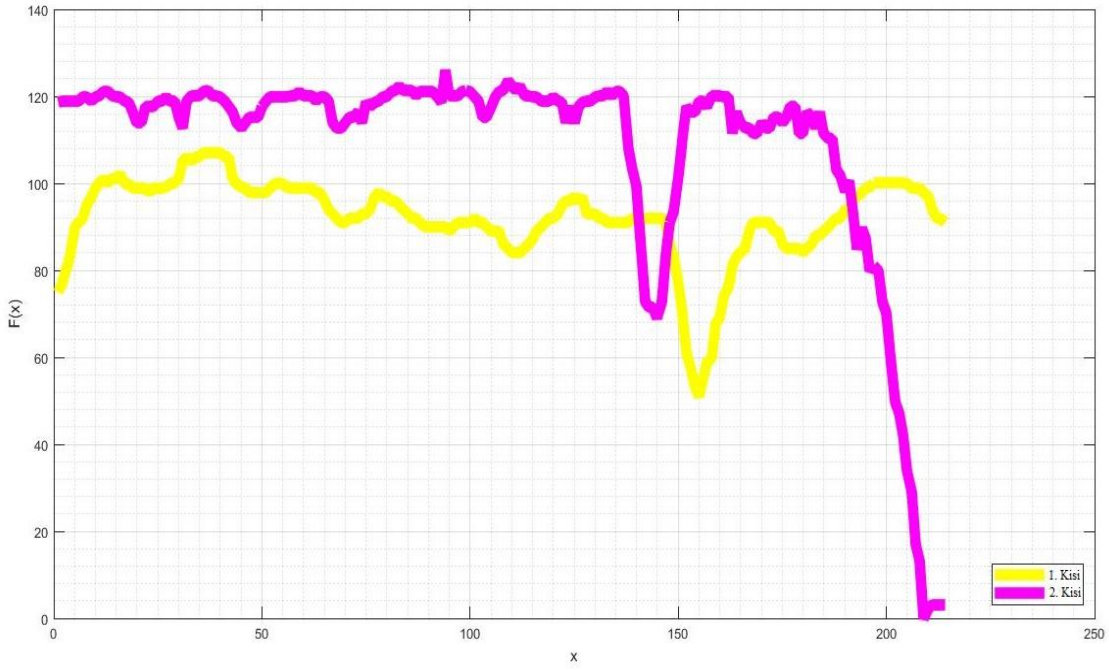
Şekil 4.35 Kinect - VK5 veri kümesi için normalizasyon uygulanmadan önce Kalça-Merkez eklem verilerinin karşılaştırılması



Şekil 4.36 Kinect - VK5 veri kümesine normalizasyon uygulandıktan sonra Kalça-Merkez eklem değerlerinin gösterimi.



Şekil 4.37 OpenPose - VK5 veri kümesine normalizasyon uygulanmadan önceki 1. ve 2. kişinin Kalça Merkez eklem değerlerinin gösterimi.



Şekil 4.38 OpenPose - VK5 veri kümesine normalizasyon uygulandıktan sonra 1. ve 2. kişinin Kalça Merkez eklem değerlerinin gösterimi.

Şekil 4.35, 4.36, 4.37 ve 4.38 de görüleceği üzere normalizasyon işlemi uygulanan veri kümelerinin orijinal veri kümelerinden beklendiği üzere uzaklaştığı görülmektedir.

Meydana gelen bu uzaklaşmalara, normalizasyon işleminin 200cm uzunluğuna göre yapılması sebep olmaktadır.

Yapılan deneyler sonucu insan hareketlerinin modellemesi esnasında vücut boy uzunluğunun kullanılmadığı görülmüştür. Normalizasyon sonrasında meydana gelen verilerin hatalı sonuçlara sebep olacaktır. Karşılaştırma sonuçlarının hesaplanması için iki farklı insandan modellenen hareketlerin orijinal değerlerinin kullanılması gerekmektedir. Çizelge 4.4 de normalizasyon işlemi uygulandıktan sonra elde edilen başarı oranları, çizelge 4.5 de ise normalizasyon işlemi uygulanmadan önce elde edilen başarı oranları gösterilmiştir. Normalizasyon işlemi sonrasında başarı oranının ciddi miktarda düştüğü görülmektedir.

Bu durum verilerin normalize edilmesinin sonuçlar üzerinde hatalı yorumlar yapıldığını göstermektedir. Bu yüzden elde edilen veri kümeleri üzerinde normalizasyon işleminin yapılmasının yanlış olduğunu göstermektedir.

Çizelge 4. 4 de Kinect Sensor – OpenPose algoritması ile modellenen iki farklı kişiye ait olan hareketlerin normalize edildikten sonra karşılaştırılması sonucu hesaplanan başarı oranları tablo halinde verilmektedir.

Çizelge 4.4 Kinect Sensor – OpenPose algoritması ile modellenen hareketlerin normalize edildikten sonra hesaplanan başarı oranları.

Eklem Adı	Aralık ve Grup Değeri	Lagrange Öğrenme Başarı Yüzdesi	Spline Öğrenme Başarı Yüzdesi	Cubic Spline Öğrenme Başarı Yüzdesi
Kafa	6 – 4	10 – 19	11 – 18	8 – 18
Kafa	2 – 4	17 – 18	13 – 18	15 – 18
Sağ Bilek	6 – 4	31 – 25	40 – 29	46 – 27
Sağ Bilek	2 – 4	52 – 26	52 – 27	53 – 27
Kalça Merkez	6 – 4	30 – 10	26 – 17	8 – 17
Kalça Merkez	2 – 4	8 – 16	8 – 17	8 – 17

Çizelge 4.5 de Kinect Sensor – OpenPose algoritması ile modellenen iki farklı kişinin oluşturduğu hareketlerin normalize edilmeden önceki (orijinal) verilerin karşılaştırılması sonucu hesaplanan başarı oranları tablo halinde verilmiştir.

Çizelge 4.5 Kinect Sensor – OpenPose algoritması ile modellenen hareketlerin normalize edilmeden önceki (orijinal) verilerin hesaplanan başarı oranları

Eklem Adı	Aralık ve Grup Değeri	Lagrange Öğrenme Başarı Yüzdesi	Spline Öğrenme Başarı Yüzdesi	Cubic Spline Öğrenme Başarı Yüzdesi
Kafa	6 – 4	31 – 86	25 – 88	29 – 88
Kafa	2 – 4	33 – 87	38 – 87	33 – 87
Sağ Bilek	6 – 4	86 – 82	56 – 79	60 – 81
Sağ bilek	2 – 4	64 – 81	63 – 82	65 – 82
Kalça Merkez	6 – 4	67 – 88	61 – 89	39 – 90
Kalça Merkez	2 – 4	36 – 89	37 – 89	37 – 37

Polinomlar elde edilirken kullanılan interpolasyon yöntemleri MATLAB üzerinde çalıştırılmaktadır. İşlem süreleri oldukça fazla olduğu için büyük verilerde insan hareketlerinin modellenmesi fazla miktarda zaman almaktadır.

5.SONUÇ VE TARTIŞMA

Tez çalışmasının bu bölümünde amaç ve sonuç tartışılmış devamında bölümler özetlenmiş son olarak da algoritmanın eksik yanları ve geleceğe yönelik geliştirilebilir alanlar üzerine tavsiyelerde bulunulmuştur.

5.1 Tezin Amacı

İnsan hareketlerinin modellenmesi için iskelet sistemi üzerinde bulunan eklem konumları sayısallaştırılmaktadır. Sayısallaştırılan bu veriler ciddi miktarlarda yer kaplamaktadır. Bu çalışmada sayısallaştırılan verilerin polinomlar haline getirilmesi sağlanmış ve üretilen polinomlardan verilerin geri elde edilebileceği gösterilmiştir. Geri elde edilen veri kümeleri ile orijinal veri kümeleri arasında meydana gelen hataların minumum düzeye indirilebilmesi için hata kareler ortalaması ve hata mutlak ortalaması değerleri hesaplanarak modellenen harekete uygun aralık ve grup parametre değerleri seçilmiştir.

5.2 Tezin Sonucu

Grup sayıları arttıkça Lagrange interpolasyonu ile elde edilen polinomlarda hem MSE hem de MAE için aynı oranda arttığı görülmektedir. Yapılan analizlere göre MSE ve MAE hata değerleri grup ve aralık sayısı arttıkça benzer oranlarda artmaktadır. Ancak grup ve aralık sayıları 10 değerini aştıklarında MAE'nin aksine MSE de daha büyük oranda artış göstermiştir. Bu artışın sebebi de Lagrange interpolasyonunun büyük veri kümelerinde sapmalar meydana getirmesidir.

Microsoft Kinect cihazı ve OpenPose algoritması ile elde edilen veriler farklı grup ve aralık sayılarına göre her bir eklem için polinomlar elde edilmiştir. Polinomlardan geri elde edilen veriler orijinal veriler ile karşılaştırılmış meydana gelen hata değerleri grafikler üzerinde gösterilmiştir. Çoklu insan iskeletinin sayısallaştırılması sırasında tek cihaz ile işlem yapıldığından dolayı eksik veriler ortaya çıkabilmektedir. Bir iskeletin diğer iskeletinin önüne geçerek kameralar tarafından tek bir iskelet olarak algılanması oldukça doğaldır. Bu tür durumlarda modellenen polinomlarda yüksek hata değerleri ortaya çıkmıştır. Ancak bu hatalar tek bir insan hareket modellenmesi sırasında gözlenmemiştir.

5.3 McNemar Testi ile anlamlılık testi

Verilerin modellenmesi sırasında meydana gelen hataların interpolasyon yöntemleri arasındaki farklılıklarını anlamak için modellemeleri yapılan veri kümeleri kullanılarak çizelge 5.1 üzerinde ANOVA testi ile Varyans Analizi Tablosu oluşturulmuştur.

Çizelge 5.1 İnterpolasyon yöntemleri karşılaştırması için yapılan varyans analiz tablosu

İnterpolasyon Yöntemi	F	P
Lagrange	15.06	0.002
Spline	4.3	0.001
Kübik Spline	4.45	0

İnterpolasyon yöntemleri arasında anlamlı bir fark bulunmaktadır.

$F_{0,5,\text{ Lagrange İnterpolasyonu}} = 15.06$, $p = 0.002 < 0,005$

$F_{0,5,\text{ Spline İnterpolasyonu}} = 4.3$, $p = 0.001 < 0,005$

$F_{0,5,\text{ Cubic İnterpolasyonu}} = 4.45$, $p = 0 < 0,005$

İnterpolasyon yöntemleri aynı veriler üzerinde modelleme yaptıkları için veriler bağımlı grup olarak nitelendirilmektedir. Bağımlı gruplar üzerinde farklılık araştırılması için de McNemar testi oldukça uygundur (Bostanci & Bostanci, 2013; Dietterich, 1998).

McNemar testi ile interpolasyon yöntemleri arasında ki farkı anlamak Z skor değerine bakılmalıdır. Eğer %95 güven aralığında yapılan analizler sonucu hesaplanan Z skor değeri 1.645'ten büyükse karşılaştırılan yöntemler arasında fark bulunmaktadır sonucuna varılır.

Çizelge 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 karşılaştırılan interpolasyon yöntemleri arasında fark bulunduğu durumda karşılaştırılmanın yapıldığı satıra yukarı ok, fark olmadığı durumlarda sola bakan ok şekli eklenmiştir.

Çizelge 5.2 de Kinect ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK3 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.2 Kinect - VK3 veri kümesinin modellenmesinde interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**1,8195**	**0.1455**
Spline			**14.4836**
Kübik Spline			

Çizelge 5.3 de Kinect ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK2 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.3 Kinect - VK2 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**4.3817**	**2.6777**
Spline			**4.9906**
Kübik Spline			

Çizelge 5.4 de Kinect ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK1 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.4 Kinect - VK1 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**2.6558**	**0.8580**
Spline			**6.4148**
Kübik Spline			

Çizelge 5.5 de OpenPose ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK3 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.5 OpenPose - VK3 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**0.5801**	**9.8884**
Spline			**18.8183**
Kübik Spline			

Çizelge 5.6 da OpenPose ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK2 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.6 OpenPose - VK2 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**0.6803**	**0.3779**
Spline			**2.1921**
Kübik Spline			

Çizelge 5.7 de OpenPose ile elde edilen aralık değeri 6 grup sayısı 4 olarak belirlenen VK1 veri kümesi için Kafa ekleminin modellenmesinde interpolasyon yöntemleri arasında meydana gelen ve McNemar Testi ile hesaplanan hata sonuçları verilmektedir.

Çizelge 5.7 OpenPose - VK1 veri kümesinin modellenmesinde kullanılan interpolasyon yöntemleri arasında meydana gelen hata değerlerinin McNemar Test sonuçları

İnterpolasyon Tipi	**Lagrange**	**Spline**	**Kübik**
Lagrange		**2.2819**	**0.6711**
Spline			**2.2819**
Kübik Spline			

McNemar istatistiksel test analizine göre, yapılan hesaplamalar da kullanılan interpolasyon tipleri değiştikçe meydana gelen sonuçlar arasında belirgin bir fark meydana gelmektedir. Sonuçlara göre Kübik Spline interpolasyonunun en etkili modellemeleri yaptığı daha sonrasında Spline interpolasyonunun Lagrange interpolasyonuna göre daha az hatalı polinomlar ürettiği görülmektedir.

5.4 Gelecek Çalışmalar

Çoklu iskelet modellenmesi sırasında veri kaybının engellenmesi için birden fazla cihaz ile farklı açılarda kayıt yapılması sonrasında da elde edilen verilerin birleştirilerek tekrar bu çalışmada geliştirilen algoritma çalıştırılarak modellenmesi ile daha doğru polinomlar elde edilerek hata değerlerini azalacağı düşünülmektedir. Çoklu kamera kayıtlarının sonrasında üst üste gelen eklem noktasının farklı açılardaki kameralar ile

giderilebileceği düşünülmektedir. Özellikle eşli dans sonrasında yapılan modellemelerde hata oranlarının küçülmesi beklenmektedir.

Ayrıca dans eğitim sonrasında başarı hesaplaması yapılırken, tüm koreografinin bir anda karşılaştırılmasının yanı sıra parça parça karşılaştırılması sonucu figür bazlı skorların hesaplanabileceği düşünülmektir.

KAYNAKLAR

Alexiadis, D. S., Kelly, P., Daras, P., O'Connor, N. E., Boubekeur, T., & Moussa, M. Ben. (2011a). Evaluating a dancer's performance using kinect-based skeleton tracking. *Proceedings of the 19th ACM International Conference on Multimedia - MM '11*. https://doi.org/10.1016/j.mcm.2012.06.008

Alexiadis, D. S., Kelly, P., Daras, P., O'Connor, N. E., Boubekeur, T., & Moussa, M. Ben. (2011b). *Evaluating a dancer's performance using kinect-based skeleton tracking*. https://doi.org/10.1145/2072298.2072412

Ben Amor, H., Neumann, G., Kamthe, S., Kroemer, O., & Peters, J. (2014). Interaction primitives for human-robot cooperation tasks. *Proceedings - IEEE International Conference on Robotics and Automation*, 2831–2837. https://doi.org/10.1109/ICRA.2014.6907265

Berrut, J.-P., & Trefethen, L. N. (2004). Barycentric Lagrange Interpolation. *SIAM Review*. https://doi.org/10.1137/S0036144502417715

Bhavsar, H., & Panchal, M. H. (2012). A Review on Support Vector Machine for Data Classification. *International Journal of Advanced Research in Computer Engineering & Technology*.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. https://doi.org/10.1109/CVPR.2017.143

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*. https://doi.org/10.5194/gmd-7-1247-2014

Cristianini, N., & Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods-Cambridge University Press. *Book*.

Daniel, C., Neumann, G., & Peters, J. (2012). Learning concurrent motor skills in versatile solution spaces. *IEEE International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/IROS.2012.6386047

Game, R. (2016). *Hand Gesture Recognition with Microsoft Kinect – A Computer Player for the Hand Gesture Recognition with Microsoft Kinect – A Computer Player for the Rock-paper-scissors Game*. (September 2014).

Ganea, D., Mereuta, E., & Mereuta, C. (2014). Human Body Kinematics and the Kinect Sensor. *Applied Mechanics and Materials*, *555*(June 2016), 707–712. https://doi.org/10.4028/www.scientific.net/amm.555.707

Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, Y. S. (2017). *CMU-Perceptual-Computing-Lab/openpose*.

Halici, E., & Bostanci, E. (2019). Evaluating the Use of Interpolation Methods for Human Body Motion Modelling. *International Journal of Computer Theory and Engineering*, *10*(1), 25–29. https://doi.org/10.7763/ijcte.2018.v10.1194

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., & Torr, P. H. S. (2016). Struck: Structured Output Tracking with Kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2015.2509974

Inoue, H., Tachi, S., Tanie, K., Yokoi, K., Hirai, S., Hirukawa, H., … Sudo, M. (2000). HRP: Humanoid Robotics Project of MITI. *International Conference on Humanoid Robots*.

Introducing JSON. (2554). Retrieved from https://www.json.org/

Joo, H., Simon, T., & Sheikh, Y. (2018). Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2018.00868

Kar, A. (2010). Skeletal Tracking using Microsoft Kinect. *Iitk.Ac.In*.

Kober, J., & Peter, J. (2014). Policy search for motor primitives in robotics. *Springer Tracts in Advanced Robotics*. https://doi.org/10.1007/978-3-319-03194-1_4

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP.2013.6639343

Lee, C. (1983). *Spline Interpolation of*. *MI*(3), 142–149.

Mckinley, S., & Levine, M. (1998). Cubic spline interpolation. *College of the Redwoods*.

Miguel Arduengo Sven Jens Jorgensen Supervisors Kimberly Hambuchen Luis Sentis Francesc Moreno Guillem Alenyà, A. (2017). *ROS Wrapper for Real-Time Multi-Person Pose Estimation with a Single Camera*. (July). Retrieved from http://www.iri.upc.edu/files/scidoc/1912-ROS-wrapper-for-real-time-multi-person-pose-estimation-with-a-single-camera.pdf

Nakai, M., Tsunoda, Y., Hayashi, H., & Murakoshi, H. (2017). *Prediction of Basketball Free Throw Shooting by*.

Newman, M., & Davis, P. J. (2006). Interpolation and Approximation. *The American Mathematical Monthly*. https://doi.org/10.2307/2315076

Numerical computing with MATLAB. (2013). *Choice Reviews Online*. https://doi.org/10.5860/choice.42-3475

Prada, M., Remazeilles, A., Koene, A., & Endo, S. (2013). Dynamic Movement Primitives for Human-Robot interaction: Comparison with human behavioral observation. *IEEE International Conference on Intelligent Robots and Systems*, 1168–1175. https://doi.org/10.1109/IROS.2013.6696498

Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. *IEEE Transactions on Multimedia*. https://doi.org/10.1109/TMM.2013.2246148

Schluchter, M. D. (2014). Mean Square Error. In *Wiley StatsRef: Statistics Reference Online*. https://doi.org/10.1002/9781118445112.stat05906

Schwarz, L. A., Mkhitaryan, A., Mateus, D., & Navab, N. (2011). Estimating human 3D pose from Time-of-Flight images based on geodesic distances and optical flow. *2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011*. https://doi.org/10.1109/FG.2011.5771333

Shin, S. (n.d.). Introduction to JSON (JavaScript Object Notation). In *Sun Microsystems*. Retrieved from http://cse.iitd.ernet.in/~cs5090250/JSON.pdf

Shotton, J Fitzgibbon, A Cook, M Sharp, T Finocchio, M Moore, R. (2011). Real- Time Human Pose Recognition in Parts from Single Depth Images. In: Computer Vision and Pattern Recognition. *(CVPR), 2011 IEEE Conference; 20–25 June*.

Song, Y., Goncalves, L., Di Bernardo, E., & Perona, P. (2008). *Monocular perception of biological motion-detection and labeling*. https://doi.org/10.1109/iccv.1999.790304

Song, Yang, Goncalves, L., & Perona, P. (2003). Unsupervised learning of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2003.1206511

Van Den Berg, J., Miller, S., Duckworth, D., Hu, H., Wan, A., Fu, X. Y., … Abbeel, P. (2010). Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. *Proceedings - IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/ROBOT.2010.5509621

Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2012.6247813

Wang, P. W., & Lin, C. J. (2014). Support vector machines. In *Data Classification: Algorithms and Applications*. https://doi.org/10.1201/b17320

Williams, Blake, & Cipolla. (2003). A sparse probabilistic learning algorithm for real-time tracking. *Proceedings Ninth IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2003.1238366

Zeng, W. (2012). Microsoft kinect sensor and its effect. *IEEE Multimedia*. https://doi.org/10.1109/MMUL.2012.24

Zerpa, C., Lees, C., Patel, P., & Pryzsucha, E. (2015). The Use of Microsoft Kinect for Human Movement Analysis. *International Journal of Sports Science*. https://doi.org/10.5923/j.sports.20150504.02

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE Multimedia*. https://doi.org/10.1109/MMUL.2012.24

ÖZGEÇMİŞ

Adı Soyadı : Egemen HALICI
Doğum Yeri : Ankara
Doğum Tarihi : 15.07.1991
Medeni Hali : Evli
Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Kazan Mustafa Hakan Güvençer Anadolu Lisesi (2009)
Lisans : Kırıkkale Üniversitesi, Fen Edebiyat Fakültesi, İstatistik Bölümü (2013)
Yüksek Lisans : Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı (Ocak 2015 - Halen)

Çalıştığı Kurumlar ve Yıl

Eşkom Bilgisayar (2013-2015)
OPLOG (2016- Halen)

Hakemli Dergiler

Halıcı, E. And Bostancı E. 2018. Evaluating the use of Interpolation Methods for Human Body Motion Modelling. International Journal of Computer Theory and Engineering. Volume 10 Number 1

Ulusal Kongre Sunum

Halıcı, E. And Bostancı E. 2016. Modelling human body motion using lagrange interpolation. IEEE 24th Signal Processing and Communication Application Conference SIU2016

Uluslararası Kongre Sunum

Halıcı, E. And Bostancı E. 2016. Analysis Of Training Performances Of Interpolation Methods For Modelling Human Body Motion. 2. International Conference on Analysis and Its Applications ICAA'16