

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ TEZLİ YÜKSEK
LİSANS PROGRAMI**

**ATALETSEL NAVİGASYON SİSTEMLERİNDE KESTİRİM İÇİN
FARKLI YÖNTEMLERİN PERFORMANSLARININ
KARŞILAŞTIRILMASI**

HAZIRLAYAN

BEKİR GÖĞÜŞ

YÜKSEK LİSANS TEZİ

ANKARA – 2022

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ TEZLİ YÜKSEK
LİSANS PROGRAMI**

**ATALETSEL NAVİGASYON SİSTEMLERİNDE KESTİRİM İÇİN
FARKLI YÖNTEMLERİN PERFORMANSLARININ
KARŞILAŞTIRILMASI**

HAZIRLAYAN

BEKİR GÖĞÜŞ

YÜKSEK LİSANS TEZİ

TEZ DANIŞMANI

DR. ÖĞR. ÜYESİ MURAT ÜÇÜNCÜ

ANKARA – 2022

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı Elektrik Elektronik Mühendisliği Tezli Yüksek Lisans Programı çerçevesinde Bekir GÖĞÜŞ tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 23 / 08 / 2022

Tez Adı: Ataletsel Navigasyon Sistemlerinde Kestirim için Farklı Yöntemlerin Karşılaştırılması

Tez Jüri Üyeleri (Unvanı, Adı – Soyadı, Kurumu)

İmza

Prof. Dr. Sedat NAZLIBİLEK, Başkent Üniversitesi

.....

Prof. Dr. Sezai ELAGÖZ, ASELSAN ARGE Yönetimi GMY

.....

Dr. Öğr. Üyesi Murat ÜÇÜNCÜ, Başkent Üniversitesi

.....

ONAY

Prof. Dr. Ömer Faruk ELALDI

Fen Bilimleri Enstitüsü Müdürü

Tarih: ... / ... / 2022

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: .../.../2022

Öğrencinin Adı, Soyadı : Bekir GÖĞÜŞ
Öğrencinin Numarası : 21810229
Anabilim Dalı : Elektrik-Elektronik Mühendisliği Anabilim Dalı
Programı : Elektrik-Elektronik Mühendisliği Tezli Yüksek Lisans Programı
Danışmanın Adı, Soyadı : Dr. Öğr. Üyesi Murat ÜÇÜNCÜ
Tez Başlığı : Ataletsel Navigasyon Sistemlerinde Kestirim için Farklı Yöntemlerin Karşılaştırılması

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan toplam 94 sayfalık kısmına ilişkin, 25/08/2022 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %3'tür.

Uygulanan filtrelemeler:

- 1.Kaynakça hariç
- 2.Alıntılar hariç
- 3.Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

26/08/2022

Öğrenci Danışmanı

Dr. Öğr. Üyesi Murat ÜÇÜNCÜ

TEŞEKKÜR

Tez çalışmamın olgun hale gelmesinde gösterdiği kararlı tutum, yönetim ve yönlendirmelerinin yanı sıra bana tecrübeleri doğrultusunda yol gösteren ve üzerimde emeği büyük olan değerli hocam ve tez danışmanım Dr. Öğr. Üyesi Murat ÜÇÜNCÜ'ye çok teşekkür ederim.

Yüksek lisans eğitimim boyunca aklımdaki her soruya bıkmadan usanmadan sabır ve hoşgörüyle cevap veren çok değerli hocam Dr. Gülnur Begüm CANGÖZ'e teşekkürlerimi sunarım.

Yüksek lisans eğitimimde Tez yazım aşamasında aklıma takılan her türlü soruya cevap veren. Fikir ve öneriyle bana yol gösteren sayın hocalarım Prof. Dr. Hamit ERDEM'e ve Dr. Öğr. Üyesi Selda GÜNEY'e teşekkür ederim

Yüksek lisans eğitimimde, karamsarlığa kapıldığım her anımda moral ve motivasyonlarıyla yanımda olan, tecrübelerinden faydalandığım ve her geçen gün çalışmam için beni daha çok yüreklendiren iş arkadaşlarım Arş. Gör. Berkan TEZCAN'a, Arş. Gör. Büşra AÇIŞ'a, Arş. Gör. Büşra Kübra KARACA'ya ve Arş. Gör. Burcu OLTU'ya teşekkür ederim.

Her zaman fikirlerini sorduğum, zor anımda yanımda olan her zaman beni destekleyen ve motive eden dostlarım Özkan AKBULUT'a, Erdem KEMAL'e ve Canberk AYDEMİR'e teşekkür ederim.

Bugün bu başarıları elde etmemde çok fazla emeği olan, bu bilgi birikiminin temelini oluşturan ilkokul öğretmenim Kemal YALÇINKAYA'ya teşekkür ederim.

Tüm eğitim hayatım boyunca her zaman desteklerini hissettiğim, emeklerini bir an olsun esirgemeyen, beni her zaman motive eden ve başarılarımla her zaman gurur duyan babam Süleyman GÖĞÜŞ'e, annem Nurcan GÖĞÜŞ'e ve abim Ali İhsan GÖĞÜŞ'e sonsuz teşekkürlerimi sunarım.

ÖZET

Bekir GÖĞÜŞ

ATALETSEL NAVİGASYON SİSTEMLERİNDE KESTİRİM İÇİN FARKLI YÖNTEMLERİN KARŞILAŞTIRILMASI

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

2022

Navigasyon sistemleri, belirli bir referans noktası yardımıyla bir nesnenin konum, hız ve yönelim bilgisini bulan sistemlerdir. Bu sistemlerden en önde gelenlerden birisi de Ataletsel Navigasyon Sistemidir. Bu sistem, dış uyaranlardan bağımsız kendi dahili sistemiyle çalıştığından dolayı tercih edilme oranı gittikçe artmaktadır. Ataletsel Navigasyon Sistemi, bünyesinde barındırdığı ivmeölçer ve dönüölçer sensörlerinden gelen verileri kullanarak bir aracın, konum, yönelim veya hız bilgileri kestirim algoritmaları ile tahmin edilebilir.

Bu çalışmada veri seti, bir otomobilin hızlanma verilerinden oluşmaktadır. Bu veriler, şehirler arası bir yolda 30,4 km mesafede değişken hızlar altında oluşturulmuştur. Ölçümler, bir akıllı telefonda MATLAB uygulaması çalıştırılarak elde edilmiştir. Bu zamana kadar yapılan çalışmalarda kestirim için ağırlıklı olarak Kalman Filtre algoritmaları kullanılmıştır. Bu tez çalışmasında ise kestirim algoritmaları olarak geleneksel kestirim yöntemi olan Kalman Filtresinin yanı sıra kendini ispat etmiş derin öğrenme algoritmalarından Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM), Çift Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-Term Memory, BLSTM) ve Geçitli Tekrarlayan Birimler (Gated Recurrent Unit, GRU) kullanılmıştır. Bu derin öğrenme algoritmaları farklı optimize edicilerle eğitilerek optimize edicilerin kestirim sonuçlarına etkileri de incelenmiştir.

Tez kapsamında detayları verilen benzetim çalışmalarında elde edilen bulgular dikkate alındığında; en başarılı sonuç 2,5414 RMSE değeri ile GRU derin öğrenme ağına aittir. LSTM ve BLSTM derin öğrenme ağlarında ise sırasıyla 2,5547 ve 2,7592 RMSE değerleri elde edilmiştir. Kalman Filtresi ile yapılan çalışmalarda ise RMSE 2,9322 bulunmuştur.

ANAHTAR KELİMELELER: Ataletsel Ölçüm Birimi, Ataletsel Navigasyon Sistemi, Derin Öğrenme, Kalman Filtresi, LSTM

ABSTRACT

Bekir GÖĞÜŞ

COMPARASION OF DIFFERENT METHODS FOR ESTIMATING IN INERTIAL NAVIGATION SYSTEMS

Baskent University Institute of Science

Department of Electrical and Electronics Engineering

2022

Navigation systems are systems that find the position, velocity and orientation information of an object with the help of a specific reference point. One of the most prominent of these systems is the Inertial Navigation System. Since this system works with its own internal system independent of external stimuli, its preference rate is increasing. Inertial Navigation System can estimate the position, orientation or velocity information of a vehicle with estimation algorithms by using data from accelerometer and rotometer sensors.

In this study, the dataset consists of acceleration data of an automobile. These data were generated under variable speeds at a distance of 30.4 km on an intercity road. The measurements were obtained by running a MATLAB application on a smartphone. In previous studies, Kalman Filter algorithms have been predominantly used for estimation. In this thesis, in addition to the traditional Kalman Filter, the proven deep learning algorithms Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BLSTM) and Gated Recurrent Units (GRU) are used as prediction algorithms. These deep learning algorithms were trained with different optimizers and the effects of the optimizers on the prediction results were also examined.

Considering the findings obtained in the simulation studies detailed in the thesis, the most successful result belongs to the GRU deep learning network with an RMSE value of 2.5414. In LSTM and BLSTM deep learning networks, RMSE values of 2.5547 and 2.7592 were obtained respectively. In the studies conducted with the Kalman Filter, the RMSE was found to be 2.9322.

KEYWORDS: Inertial Measurement Unit, Inertial Navigation System, Deep Learning, Kalman Filter, LSTM

İÇİNDEKİLER

Sayfa

TEŞEKKÜR.....	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ	vi
ŞEKİLLER LİSTESİ	vii
SİMGELER VE KISALTMALAR LİSTESİ	x
1. GİRİŞ	1
1.1. Çalışmanın Genel Konusu	1
1.2. Çalışmanın Amacı ve Önemi	2
1.3. Literatür Taraması.....	3
2. ATALETSEL NAVİGASYON SİSTEMİ	10
2.1. Sistemin Tanımı ve Mimari Yapı.....	10
2.2. Ataletsel Ölçüm Ünitesi.....	13
2.3. Ataletsel Sensörler	14
2.4. İvmeölçerler	16
2.4.1. MEMS ivmeölçerler	18
2.5. Dönüölçerler.....	21
2.5.1. Optik dönüölçerler	23
2.5.2. MEMS dönüölçerler	25
2.5.3. Dönüölçer teknolojilerinin karşılaştırılması	27
3. MATERYAL VE METOTLAR	29
3.1. Kullanılan Veri Seti.....	29
3.2. Kullanılan Kestirim Algoritmaları	31
3.2.1. Rotasyon matrisi.....	32

3.2.2. Luenberger gözlemcisi	34
3.2.2.1 Gözlenebilirlik	35
3.2.2.2 Kontrol edilebilirlik	35
3.2.2.3 Luenberger gözlemci algoritması	36
3.2.3. Kalman filtresi	38
3.2.3.1 Kalman filtresi algoritması	38
3.2.4. Uzun-kısa vadeli bellek (Long-short term memory, LSTM).....	42
3.2.5. Çift yönlü uzun kısa süreli bellek (Bidirectional long short-term memory, BLSTM).....	55
3.2.6. Geçitli tekrarlayan birim (Gated recurrent unit, GRU).....	56
4. BENZETİM ÇALIŞMALARI	61
4.1. Geleneksel Kestirim Yöntemleri İçin Kinematik Modelin Oluşturulması.....	61
4.2. Geleneksel Kestirim Yöntemleri İçin Benzetim Sonuçları	64
4.2.1. Kalman filtresi benzetim sonuçları.....	65
4.2.2. Luenberger gözlemcisi benzetim çalışmaları.....	73
4.3. Derin Öğrenme Tabanlı Yöntemler ile Kestirim	74
4.3.1. LSTM algoritması ile alınan benzetim sonuçları	75
4.3.2. BLSTM algoritması ile alınan benzetim sonuçları.....	82
4.3.3. GRU algoritması ile alınan benzetim sonuçları.....	88
5. SONUÇLAR VE ÖNERİLER	93
KAYNAKLAR	95

TABLÖLAR LİSTESİ

	Sayfa
Tablo 2.1. ANS ve KKS özelliklerinin karşılaştırma tablosu [24]	11
Tablo 2.2. Donanımsal Olarak Sistemlerin Karşılaştırma Tablosu [24].....	13
Tablo 2.3. Dönüölçer teknolojisi karşılaştırma tablosu [54].....	28
Tablo 3.1. Kullanılan veri setine ait detaylar	30
Tablo 3.2. Çalışmalarda kullanılan veri seti bilgileri	31
Tablo 4.1 LSTM derin öğrenme ağı hız kestirim sonuçları	78
Tablo 4.2. BLSTM derin öğrenme ağı hız kestirim sonuçları	83
Tablo 4.3. GRU derin öğrenme ağı hız kestirim sonuçları	88
Tablo 4.4. Derin öğrenme ağlarına ait en başarılı kestirim sonuçları	92

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1. Ataletsel Navigasyon Sistemi Yapı Taşları [23].....	10
Şekil 2.2. Halkalı Bir Sistem Örneği [25]	12
Şekil 2.3. Direkt Gövdeye Bağlı Bir Sistemin Temel Görünümü [27]	13
Şekil 2.4. Tipik Ataletsel Ölçüm Birimi Bileşenleri [24]	14
Şekil 2.5. Halkalı atalet navigasyon sistemi algoritması [25].....	15
Şekil 2.6. Direkt gövdeye bağlı atalet navigasyon sistemi [25]	16
Şekil 2.7. Farklı koşullarda kütle ve yay ivmeölçeri [30]	16
Şekil 2.8. İvmeölçer Çeşitleri hakkında karşılaştırmalı bilgi [32]	17
Şekil 2.9. (a) Piezodirenç ivmeölçer kesit görünümü. (b) Piezoivmeölçer üstten görünümü [37].....	19
Şekil 2.10. Piezoelektrik sensör genel yapısı [39]	19
Şekil 2.11. Tipik bir kapasitif ivmeölçer [48]	21
Şekil 2.12. Tipik bir dönüölçer yapısı [52]	22
Şekil 2.13. Güncel dönüölçer teknolojisi uygulamaları [55]	22
Şekil 2.14. Sagnac etkisine dayalı bir optik dönüölçer genel şeması [58]	23
Şekil 2.15. RLG Örneği [60].....	24
Şekil 2.16. IFÖG Örneği [60]	25
Şekil 2.17. Coriolis etkisinin gösterimi [64]	25
Şekil 2.18. Akort çatalı dönüölçer genel görünümü [65].....	26
Şekil 2.19. Bir akort çatal dönüölçerin üstten görünümü [67].....	26
Şekil 2.20. Titreşimli tekerlek dönüölçerin şematik üstten görünümü [69].....	27
Şekil 3.1. Akıllı telefonun sensör konfigürasyonu.....	30
Şekil 3.2. Derin sinir ağı genel modeli [73]	32
Şekil 3.3. Otomobil üzerinde açılarının gösterimi [75].....	33
Şekil 3.4. Luenberger gözlemci blok şeması [82].....	37
Şekil 3.5. Gauss olasılık yoğunluk grafiği	39
Şekil 3.6. Kalman filtre blok şeması [88]	40
Şekil 3.7. Kalman filtresi algoritma şeması	42
Şekil 3.8. $\tanh$ Fonksiyonu	43
Şekil 3.9. Standart bir RNN bloğu [93]	43

Şekil 3.10. Standart bir LSTM bloğu [93]	44
Şekil 3.11. LSTM bloğundaki doğrusal işlem hattı [94].....	44
Şekil 3.12. Sigmoid fonksiyon grafiği	45
Şekil 3.13. LSTM bloğunun ilk katmanı [94]	45
Şekil 3.14. LSTM bloğunun ikinci katmanın ilk hali [94]	46
Şekil 3.15. LSTM bloğunun ikinci katmanının sonraki hali [94]	46
Şekil 3.16. LSTM bloğunun son katmanı [94].....	47
Şekil 3.17. Öğrenme oranı seçimine göre sonuç karşılaştırması şekil [96]	48
Şekil 3.18. Momentumun optimize edicilerdeki etkisi	49
Şekil 3.19. Stokastik gradyan iniş ile gradyan iniş algoritmasının karşılaştırması [99]	50
Şekil 3.20. BLSTM yapısı [108]	55
Şekil 3.21. LSTM ve BLSTM mimarisi karşılaştırması [109]	56
Şekil 3.22. Standart bir GRU yapısı [110]	57
Şekil 3.23. Kapı güncelleme işlemi [110].....	58
Şekil 3.24. Sıfırlama kapısı [110]	58
Şekil 3.25. GRU yapısının üçüncü adımı [110]	59
Şekil 3.26. GRU yapısının son işlem basamağı [110]	60
Şekil 4.1. Sistemin temsili gösterimi.....	61
Şekil 4.2. Geleneksel hız kestirim yöntemleri için ivme verilerini hazırlama aşamaları	64
Şekil 4.3. Yapay verilerin ivme-zaman grafiği	65
Şekil 4.4. Oluşturulan yapay ivme veri setinden elde edilen hız kestirim sonucu.....	66
Şekil 4.5. Akıllı telefon ivmeölçer algılayıcılarından elde edilen y eksenli ivme verileri	66
Şekil 4.6. Kalman filtresi hız kestirim grafiği.....	67
Şekil 4.7. Tüm veri setinin ortalama karekök sapma hatası.....	68
Şekil 4.8. Kalman Filtresi hız kestirim grafiği (138 saniye).....	68
Şekil 4.9. Ortalama karekök sapma hatası (138 saniye)	69
Şekil 4.10. Sabit kaymadan arındırılmış ivme verileri.....	70
Şekil 4.11. Arındırılmış ivme verileri ile hız kestirim grafiği.....	71
Şekil 4.12. Kestirim sonucu elde edilen RMSE değerleri.....	71
Şekil 4.13. Arındırılmış ivme verileri ile Kalman Filtresi hız kestirimi (138 saniye).....	72
Şekil 4.14. Kestirim sonucu elde edilen RMSE değerleri (138 saniye).....	72

Şekil 4.15. Derin öğrenme ağları hız kestirim akış diyagramı.....	75
Şekil 4.16. LSTM ağında kullanılan ham veri seti.....	75
Şekil 4.17. Adam optimize edici eğitim süreci grafiği.....	79
Şekil 4.18. RMSprop optimize edici eğitim süreci grafiği	80
Şekil 4.19. SGDM optimize edici eğitim süreci grafiği.....	81
Şekil 4.20. LSTM hız tahmin grafiği	81
Şekil 4.21. LSTM hız kestirimi RMSE değerleri grafiği	82
Şekil 4.22. Adam optimize edici eğitim süreci grafiği.....	84
Şekil 4.23. RMSprop optimize edici eğitim süreci grafiği	85
Şekil 4.24. SGDM optimize edici eğitim süreci grafiği.....	86
Şekil 4.25. BLSTM hız tahmin grafiği	87
Şekil 4.26. BLSTM hız kestirimi RMSE değerleri grafiği	87
Şekil 4.27. Adam optimize edici eğitim süreci grafiği.....	89
Şekil 4.28. RMSprop optimize edici eğitim süreci grafiği	90
Şekil 4.29. SGDM optimize edici eğitim süreci grafiği.....	91
Şekil 4.30. GRU hız tahmin grafiği	91
Şekil 4.31. GRU derin öğrenme ağı RMSE grafiği	92

SİMGELER VE KISALTMALAR LİSTESİ

AKF	Adaptive Kalman Filter (Adaptif Kalman Filtresi)
ADAM	Adaptive Moment Estimation (Uyarlanabilir Moment Tahmini)
ADC	Analog Digital Converter (Analog Dijital Dönüştürücü)
ANS	Ataletsel Navigasyon Sistemi
AÖB	Ataletsel Ölçüm Birimi
AHRS	Attitude and Heading Reference System (Tutum ve Yön Referans Sistemi)
BLSTM	Bidirectional Long Short-Term Memory (Çift Yönlü Uzun Kısa Süreli Bellek)
BCoM	Body Center of Mass (Vücut Kütle Merkezi)
CW	Clock Wise (Saat Yönü)
CF	Complementary Filter (Tamamlayıcı Filtre)
CNN	Convolutional Neural Network (Evrişimsel Sinir Ağı)
CCW	Counter Clock Wise (Saat Yönünün Tersi)
DTG	Dry Tuned Gyro (Kuru Ayarlı Gyro)
EKF	Extended Kalman Filter (Genişletilmiş Kalman Filtresi)
FOG	Fibre-Optic Gyroscope (Fiber Optik Jiroskop)
GBT	Gradient Boosted Trees (Gradyan Destekli Ağaçlar)
GLM	Generalized Linear Model (Genelleştirilmiş Doğrusal Model)
GRU	Gated Recurrent Unit (Geçitli Tekrarlayan Birimler)
GPR	Gaussian Process Regression (Gauss Süreç Regresyonu)
GNSS	Global Navigation Satellite Systems (Küresel Uydu Seyrüsefer Sistemi)
GPS	Global Positioning System (Küresel Konumlandırma Sistemi)
GSVM	Gaussian Support Vector Machine (Gauss Destek Vektör Makinesi)
IMU	Inertial Measurement Unit (Ataletsel Ölçüm Birimi)
IONet	Inertial Odometry Neural Network (Ataletsel Odometri Sinir Ağı)
IFOG	Interferometric Fibre-Optic Gyroscope (İnterferometrik Fiber Optik Jiroskop)
IPM	Inverted Pendulum Model (Ters Sarkaç Modeli)
İHA	İnsansız Hava Aracı
KF	Kalman Filtresi
KKS	Küresel Konumla Sistemi
LS	Least Squares (En Küçük Kareler)
LKF	Linearized Kalman Filter (Doğrusallaştırılmış Kalman Filtresi)
LSTM	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MAÖB	Manyeto-Atalet Ölçüm Birimi
MS	Mean Shift (Ortalama Kayma)
MEMS	Mikro Elektro-Mekanik Sistemler
NRMSE	Normalized Root Mean Square Error (Normalleştirilmiş Kök Ortalama Kare Hatası)
RNN	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
RLG	Ring Laser Gyroscope (Halka Lazer Jiroskop)

RMSE	Root-Mean-Square Error (Kök-Ortalama-Kare Hatası)
SGD	Stochastic Gradient Descent (Stokastik Gradyan İniş)
SGDM	Stochastic Gradient Descent with Momentum (Momentumlu Stokastik Gradyan İniş)
SLD	Super Luminescent Diode (Süper Lüminesan Diyet)
SVM	Support Vector Machine (Destek Vektör Makinesi)
UKF	Unscented Kalman Filter (Kokusuz Kalman Filtresi)
VRU	Vertical Reference Units (Dikey Referans Birimleri)
ZVU	Zero Velocity Update (Sıfır Hız Güncellemesi)



1. GİRİŞ

1.1. Çalışmanın Genel Konusu

Temel olarak açıklanmaya çalışıldığında navigasyon, belirli bir referans noktası yardımıyla bir nesnenin konumunu, hızını ve yönelimini bulma işlemidir. İlk insanlar eski zamanlarda avcılık yaptıklarında tekrar yaşadıkları yerlere dönebilmek amacıyla iç güdüsel olarak belirli referans noktalarını kullanarak en basit haliyle navigasyonu kullanmışlardır. Günümüzde ise teknolojinin de ilerlemesiyle beraber navigasyonun birçok alanda kullanım olanağı oluşmuştur. Navigasyon; askeri alanda, keşif, hedef izleme, arama kurtarma ve güdüm sistemlerinde, havacılık ulaşımında rota tayinine ve aynı şekilde diğer ulaşım ağlarında da aynı amaçla kullanılmaktadır.

Günümüzde gelişen teknolojiyle beraber, Amerika Birleşik Devletleri'ne ait GPS, Rusya'ya ait GLONASS, Avrupa Birliğine ait olan Galileo gibi sistemler yardımıyla navigasyon hizmetlerinden yararlanılır. Bu hizmetler Küresel Konumlama Sistemleri (KKS) olarak bilinir ve uydular yardımıyla hizmet sağlarlar. Ancak bu tür sistemlerde sinyal bağlantısının kesilmesi durumunda navigasyon teknolojisi çalışmamaktadır. Bundan dolayı bu tür senaryolara karşı alternatif yöntemler geliştirilmektedir.

Teknolojik gelişmelere paralel olarak ataletsel sensörlerin üretimi ve geliştirilmesi kolaylaşmıştır. Bunun sonucunda dışarıdan herhangi bir bilgiye ihtiyaç duymadan kendi üzerinde bulunan sensörler yardımıyla elde ettiği verileri işleyerek bir nesnenin hız, konum ve yönelim bilgisini belirleyen Ataletsel Navigasyon Sistemleri (ANS) ortaya çıkmıştır.

ANS temelde bir kör seyir sistemidir. Tamamen bağımsızdır ve herhangi bir elektromanyetik uyarana ihtiyaç duymaz. Bu sistem entegre edildiği nesnenin hızını ve yönelimini kendi üzerindeki sensörler yardımıyla hesaplar. Bu sensörler birbirleriyle 90°lik açı yapacak şekilde üç eksene yerleştirilmiş ivmeölçer ve dönüölçerlerdir. İvme ölçerler yardımıyla her bir eksene etki eden kuvvetler bulunur. Aynı eksenlere yerleştirilmiş dönüölçerler yardımıyla da her bir eksen etrafındaki dönüş hızı elde edilir. Bu sensör verileri işleme sokularak navigasyon bilgileri elde edilir.

ANS sensör verilerinden navigasyon bilgilerini elde ederken farklı uygulamalardan yararlanmaktadır. Bunlardan birisi olan Kalman Filtresi, durum uzay modeli oluşturulmuş sistemin sensörlerden alınan ölçümler yardımıyla bir sonraki durumunu kestiren filtre yapısıdır.

Luenberger Gözlemci ise, sensör ölçümleri yardımıyla sistemin durum değişkenlerini kestirebilen diğer bir gözlemcidir. Çalışma mantığı birçok yönden Kalman Filtresiyle aynıdır. İki kestirim algoritması arasındaki temel fark, Kalman Filtresi kazancını her adımda güncellerken Luenberger Gözlemci'nin kazancında bir değişiklik yapılmamasıdır. Bu özelliklerle Kalman Filtresi değişken sistemlere uyum sağlarken Luenberger izleyici doğrusal sistemlerde başarılı sonuçlar vermektedir. Bu kestirim algoritmalarına ek olarak navigasyon bilgileri elde etmek için derin öğrenme ağı algoritmalarından da faydalanılmıştır. Derin öğrenme ağı algoritmalarında ise sensörlerden sağlanan verilerin belirli bir kısmı eğitim verisi olarak kullanılıp veri kestirimi yapılmıştır.

1.2. Çalışmanın Amacı ve Önemi

Bu çalışmada, veri seti oluşturulurken bir otomobilin ivme ve dönü verileri elde edilmiş ve bu sensör verileri kestirim algoritmaları yardımıyla otomobilin hız kestirimi yapılmıştır. Hız kestirimi yapılırken, sistem modeli ve kestirim algoritmaları bir otomobilin hızının kestirilebilmesi amacıyla düzenlenmiştir. Ancak bu çalışmanın amacı sadece otomobil hız kestirimi olarak sınırlandırılmamalıdır. Zira bu çalışmalarda kullanılan yöntemler gerekli uyarlamalarla bir yayanın yürüyüş hızı kestirimi veya herhangi bir hava aracının hava hızı kestirimi için de kullanılabilir. Başka bir deyişle bu tez kapsamında yapılan çalışmanın amacına daha geniş bir bakış açısıyla değinilecek olursa, herhangi bir araca ait bir parametrenin kestirimi için yöntem araştırması yapıldığında kullanılabilecek yöntemler için yol göstermesi ve hangi yöntemin daha başarılı sonuçlar sağlayabileceği yönünden bir rehber oluşturması için yapılmıştır.

Bu tez çalışmasında, bir otomobilin hız kestirimi yapılması amacıyla Kalman Filtresi, Luenberger gözlemcisi, Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM), Çift Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short-Term Memory BLSTM) ve Geçitli Tekrarlayan Birim (Gated Recurrent Unit, GRU) algoritmaları kullanılmıştır. Bu algoritmalar yardımıyla, sensörlerden alınan ivme verileri işlenerek durum kestirimi yapılmıştır. Hız kestirimi sonucunda algoritmaların durum kestirim başarımları karşılaştırılmış, Kalman filtresinin Q (işlem gürültü kovaryansı) ve R (ölçüm gürültü kovaryansı) parametreleri ile kestirim sonuçları en başarılı hale getirilmiştir. Ek olarak derin öğrenme ağı algoritmalarında bulunan parametrelerin kestirim başarısına etkileri incelenerek ve bütün kestirim yöntemlerinden elde edilen sonuçlar RMSE türünden tek bir tabloda toplanarak kıyaslama yapılmıştır.

1.3. Literatür Taraması

Navigasyon alanında son yıllarda gittikçe artan miktarda çalışmalar yapılmıştır. Bu çalışmalardan bir kısmı detaylı olarak incelenmiş ve yapılan çalışmalar takip eden paragraflarda özetlenmiştir.

Hassen Fourati tarafından yapılan çalışmada [1] yaya takibi için ayağın konumu, pozisyonu, yürüyüş aşamaları, hız ve hızlanma gibi kinematik parametreleri, ayak hareketleri filtreleme algoritmaları yönelim tahmini, yürüyüş tespiti ve konum tahmini için yöntemler ele alınmaktadır. Bu çalışmada, ayağa takılı bir Ataletsel Ölçüm Birimi (Inertial Measurement Unit, IMU) tarafından ölçümler yapılmış, ayağın yöneliminin tahmini bir kuaterniyona dayalı Tamamlayıcı Filtre (Complementary Filter, CF) ile gerçekleştirilmiştir. Ölçümler sırasında sensör gürültüsü sebebiyle ivmeölçer verilerinde kayma gözlemlenmiştir. Söz konusu kayma Sıfır Hız Güncellemesi (Zero Velocity Update, ZVU) kullanılarak düzeltilmişlerdir. Çalışma kapsamında yapılan deneylerde bitiş ve başlangıç noktası arasındaki farkın ölçüm sırasındaki yürüme mesafesinin neredeyse %0,4'üne tekabül ettiğini göstermişlerdir.

Benzer bir diğer çalışmada [2], bir yayanın hızını ve yöneliminin tahmininden yola çıkarak yayanın konumu belirlenmeye çalışılmıştır. Uzayan süreçlerde gürültülü ivme ve açısal hız verilerinin entegrasyonu kaymalara sebep olduğundan hız tahmini yapmak zorlaşmıştır. Bu nedenle hız tahmini yaparken çeşitli parametre ayarlamaları yapmak gerekmiştir. Bu nedenle hibrit bir model oluşturulmuş, bu hibrit modelde evrimsel bir sinir ağı (CNN: Convolutional Neural Network) ve çift yönlü tekrarlayan bir sinir ağı (BLSTM) hız tahminlerinde iyileştirmeler sağlamak amacıyla doğrusal bir Kalman Filtresi (KF) ile birleştirilmiştir. Yeni mimari; geleneksel, makine ve derin öğrenme yöntemleriyle kıyaslanmıştır. Yeni mimarinin hızı 0,16 m/s 'den küçük değerler ile en son teknolojilerden daha iyi sonuçlar vermektedir.

Mats Jonasson ve arkadaşları tarafından yapılan çalışmada [3] ise otonom araçlar için konumun ve konum belirsizliğinin tahmini problemine çözüm geliştirilmeye çalışılmıştır. Bu çalışmada kullanılan ölçüm birimleri filtreleme işlemlerinde kullanılabilmesi için karakterize edilmiştir. Genişletilmiş Kalman Filtresi (Extended Kalman Filter, EKF) araç dinamiği modelleriyle birleştirilmiştir. Konum tahmini için iki farklı EKF konfigürasyonu geliştirilmiştir. Bunlardan ilki yumuşak manevralarda kullanılan daha basit bir filtredir. İkincisi ise daha sert manevralarda kullanılan daha karmaşık bir filtredir. Belirli bir süre boyunca denenen ve elde edilen verilere göre aracın ilk hızının 70 km/s ve üzeri durumlarda konum tahmin hatalarının önemli ölçüde arttığını gözlemlemişlerdir. Kabul edilebilir konum

hataları boylamasına 3 m ve yanal olarak 0.75 m'dir. Yapılan deneyler sonucunda 70 km/s başlangıç hızının altındaki hızlarda konum hataları kabul edilebilir sınırları aşılmamıştır. Ataletsel Ölçüm Birimi (AÖB)'nin düşük ofset sapmasının konum tahmini doğruluğu için önemli olduğu belirtilmiştir. Bunun yanı sıra araç güvenli durma pozisyonundayken AÖB'nin algılayıcı ön yargısının önemli ölçüde değişiklik göstermeyeceği ve bu süreçte hata telafisinin gereksiz olduğunu göstermişlerdir.

İncelenen bir başka çalışmada [4] bileğe takılı altı eksenli ataletsel ölçüm birimi verilerinden hız tahmini yapılmıştır. Hız tahmini için TreeBagger modeliyle önceden belirlenmiş yedi farklı algılayıcı taşıma biçimine uyarlanan Gauss Süreç Regresyonu (GPR: Gaussian Process Regression) modelleri kullanılmıştır. TreeBagger modelinde elde edilen olasılıklar, çoklu GPR modellerinin ağırlıklı toplamıyla hesaplanarak hız tahmini yapılmıştır. Kullanılan yöntemin başarı oranını ölçmek amacıyla yürüme hızı tahmin hatasını, ticari bileğe takılan cihazlarla ve mevcut yapılan çalışmaları karşılaştırmışlardır. Test ettikleri çalışmalar ve ticari ürünler normal koşullarda %10 civarında hata gösterirken kendi önerdikleri yöntem ile tüm koşullarda %5 civarında bir hata aldıklarını göstermişlerdir.

Yapılan başka bir çalışmada [5], motor hız verileri ve AÖB verileri doğrultusunda dört rotorlu bir İnsansız Hava Aracının (İHA) hız ve konum tahminini ele alınmıştır. Yüksek hızlı uçuş esnasında bir İHA'nın hızını ve tutumunu tahmin etmek için yeni bir Unscented Kalman Filtresi (UKF) geliştirilmiştir. UKF, bu sistemde hız ve irtifa tahminine ek olarak ivmeölçer sapmasını, sürüklenme ve itme katsayılarını da tahmin etmektedir. Bunun yanı sıra dönüölçerlere bağlı ölü hesaplama ile elde edilen sapmayı ciddi oranlarda azaltarak İHA'nın yalpa oranını da tahmin etmek için yeni bir yaklaşım önerilmiştir. Geliştirilen yöntem ile geleneksel RANSAC algoritması atlanarak, kamera görüntüsünün kareleri arasındaki gövde hareketini tahmin etmek için UKF çıktısı kullanılmıştır.

Havacılıkla ilgili yapılan başka bir çalışmada [6] ataletsel ölçüm birimi kullanan bir helikopterin hız destekli konum tahmini için bir algoritma sunulmaktadır. Bu çalışmada, yer çekimi dışındaki bozucu faktörlerden etkilenmemek için hava hızı kullanılmıştır. Helikopter ortamının koşullarını temel alarak, hataya yol açabilecek bozucu ivme denklemleri oluşturulup, bu denklemler sayesinde bozucu etkenleri ortadan kaldırmışlardır. Bozulmayan ivme verileri dönüölçer verileri ile kullanacakları tamamlayıcı bir filtre tasarlayarak, dinamik koşulların olduğu şartlarda bile etkin bir tahmin doğruluğu elde etmişlerdir. Bozucu ivmeden arındırılmış tamamlayıcı filtre ile yalpalama hareketinde RMS cinsinden hata

1.6498'den 0.4136'ya ve yuvarlanma hareketinde yine RMS cinsinden 1.8789'dan 0.3371 değerleri elde edilmiştir.

Zrenner ve arkadaşlarınca yapılan çalışmada [7], koşu ayakkabısının tabanının ortasına AÖB yerleştirilmiştir. AÖB'den alınan verileri kullanarak koşu sırasında adım hızını ve adım uzunluğunu hesaplayan dört algoritmanın doğruluğu değerlendirilmiştir. Dört algoritma sırasıyla adım atma süresi, ayak ivmesi, ayak yörünge tahmini ve derin öğrenmeye dayanmaktadır. Referans olarak 3D hareket izleme ile 27 denekten 2377 adımdan oluşan laboratuvar temelli bu çalışmada 3,2 km'lik bir koşu gerçekleştiren 12 denekten oluşan bir saha çalışması yapılmıştır. Sonuçlar incelendiğinde, ayak yörüngesi tahmin algoritmasının en iyi performansı gösterdiği ve ortalama $0,032 \pm 0,274$ m hata elde edildiği gösterilmiştir. Bu çalışmada ayrıca yörünge algoritmasının 5 m/s'ye kadar hızlar için en iyi performansı ortaya koyduğu gösterilmiştir.

Derin öğrenme yöntemi kullanılan bir çalışmada [8], koşu bandında egzersiz yaparken giyilebilir bir cihaz tarafından elde edilen ivme verilerinden hız tahmini yapılmıştır. Hareket hızı tahmini için derin Evrişimli Sinir Ağı (Convolutional Neural Network, CNN) modeline uygulanan bir regresyon algoritması kullanılmıştır. Hız kestirimi ortalama hatasının, gerçek koşu ve yürüme testi hızlarının sırasıyla %7 ve %18'inde doğru olduğu gösterilmiştir.

İncelenen başka bir çalışmada [9], Heng Deng ve çalışma arkadaşları bir multirotor uçuş kararlılığı ve manevra becerisini iyileştirmek için hız tahmini yapmışlardır. GPS (Global Positioning System) sinyalinin olmadığı durumlarda görsel ve atalet bilgilerine dayalı multirotor için yeni bir hız tahmin yöntemi önerilmiştir. Bu yöntemde, herhangi bir harita ya da çevresel yapay bir işarete gerek yoktur. En uygun eşleşme noktalarını seçmek için Ortalama Kayma (Mean Shift, MS) algoritması kullanılmıştır. Ayrıca görüş kısıtlamasında metrik hız Doğrusal Kalman Filtresi kullanarak tahmin yapılmıştır. Gerçek veriler ve simülasyon verileri ile önerilen yöntem test edilmiştir. GPS'nin olmadığı durumlarda multirotor hızının doğru bir şekilde tahmin edildiği gösterilmiştir.

Bir hastanın yürüyüşünü değerlendirmek için tıp alanında hareket yakalama sistemleri kullanılır. Hori ve çalışma arkadaşları [10], ayak yörüngesini ve geçici yürüyüş parametrelerini tahmin etmek için ayak bileğine takılan AÖB'leri kullanan bir yürüyüş analiz yöntemini test etmişlerdir. Önerdikleri yöntem iki aşamadan oluşmaktadır. Bu aşamalar; sürekli yürüyüş verilerinin aşamalı olarak çoklu adımlara ayrılması ve ivmeölçer ile jiroskoplardan elde edilen verilerden üç boyutlu yörünge tahminidir. Yöntemi, 19 sağlıklı katılımcının yürüyüş verilerini analiz ederek yorumlamışlardır. Ortalama doğruluk olarak, adım uzunluğu için $0,054 \pm 0,031$ m, yürüme hızı için $0,034 \pm 0,039$ m/s, adım süresi için

0,002 ± 0,020 s, duruş süresi için 0,000 ± 0,017 s ve sallanma süresi için 0,002 ± 0,024 s değerleri elde edilmiştir.

Kısa mesafe koşucularının adım uzunluğu tahmini üzerine bir çalışma yapılmıştır [11]. Bu çalışmada; AÖB'ler yalnızca düşük hızlara (<%50 maks) sınırlıdır. Burada her iki ayağa yerleştirilen AÖB'ler iki zamanlama geçidi kullanarak, azami sürat ivmesi sırasında adım uzunluğunu anlamak için basit bir seçenek önermişlerdir. Tek üstel hız-zaman fonksiyonları, AÖB'den sağlanan adım süreleri ile birleşimi halinde adım uzunluğu kestirimi veren 30m ve 60m koşu zamanlarına uyarlamışlardır. Bu yaklaşımı kanıtlamak için 16 iyi eğitilmiş sporcunun 60 metrelik kısa mesafe koşu verileri alınmıştır. Karşılaştırmak için video görüntülerinden de adım uzunluğu belirlemişlerdir. Adım uzunluklarının kestirimini referans ölçümlerle karşılaştırılması sonucu anlamlı bir farklılık ($p > 0.05$) ve kabul edilebilir bir Kök Ortalama Kare Hata (Root Mean SquareError, RMSE) göstermemiştir, RMSE = 8,0 cm, sapma ± Uyum Sınırları = $-0,15 \pm 16$ cm olarak bulunmuştur. Adım uzunluğu kestirimleri polinom fonksiyonuyla ($R^2 = 0.94 \pm 0.04$) yumuşatıldıktan sonra iyileştirilmiştir. Azami koşu ivmesi sırasında, AÖB'den çıkarılan adım sürelerinden ve 30 m ve 60 m koşu sürelerinden kabul edilebilir adım ve adım uzunluğu kestirimleri elde edilmiştir.

Ariante ve çalışma arkadaşları [12] yaptıkları bu çalışmada, Hız ve konum kestirimi amacıyla pitot tüpü sensörü ve AÖB kullanarak yörünge ölçümü ve kontrolüne odaklanmışlardır. Çalışmada, İHA hızı ve tutum değerlerinin kestirimini sağlamak ve bunları sensör verileriyle kıyaslamak amacıyla eski bir otomatik pilot ve küçük bir dörtlüden alınan deneysel veriler ile yöntemlerini sınamışlardır. Ölçüm gürültüsünü gidermek ve veri füzyonu için Kalman filtresi kullanmışlardır. Bir takım uçuş testinden gelen sonuçlar ile yörünge tespiti yaklaşımının geçerliliği doğrulanmıştır.

İncelenen başka bir çalışmada [13], çoklu sensörlerden gelen bilgileri birleştirerek bir iç mekân navigasyonu üzerinde çalışılmıştır. Önerdikleri yöntemde temel amaç Kalman filtresini sıfır hız güncelleme algoritması ile kullanmaktır. Yaya yörünge tahmininin doğruluğunu artırmak için ayağa takılmış AÖB sistemi kullanılmıştır. İlâveten, hız ve konum kestirimini daha iyi hale getirebilmek amacıyla birtakım ayarlanabilir parametreler kullanılmıştır. Önerilen yöntemin yeterliliği, kapalı mekân deneylerinde doğrulanmıştır. Ayrıca yöntemin uzun yol performansı da bir parkurda denenmiştir.

Celal GENÇOĞLU ve Hikmet GÜMÜŞ tarafından yapılan çalışmada [14] hentbolda top atış hızı kestirimi için düşük maliyetli AÖB sensörlerinin etkinliği araştırılmıştır. Çalışmalar sporcunun atış için kullandığı koluna AÖB yerleştirildikten sonra, oyunculara

1'er dakikalık dinlenme süresi verilerek atışlar yaptırılmıştır. Radar hız tabancası ve ivmeölçer sensör verilerinin kestirim sonuçları karşılaştırılmıştır. Genelleştirilmiş Doğrusal Model (Generalized Linear Model, GLM), Gradyan Destekli Ağaçlar (Gradient Boosted Trees, GBT) ve Destek Vektör Makinesi (Support Vector Machine, SVM) performans ölçümleri; kök ortalama kare hataları, mutlak hata ve korelasyon katsayıları cinsinden hesaplamışlardır. Sonuç olarak ivmeölçerlerden edinilen verilerin makine öğrenmesi modelleri yardımıyla top atış hızını kesin olarak tahmin ettiği gösterilmiştir.

Robert H. Rogne ve çalışma arkadaşları [15] yaptıkları çalışmada dinamik konumlandırma sistemine sahip bir gemiye ataletsel navigasyon sisteminin ilaveten fayda sağlayabileceğini ortaya koymuştur. Tutum, yükselme, hız ve konum kestirimi için doğrusal olmayan gözlemciler kullanılmıştır. Çalışmada, iki farklı MEMS (Mikro elektro-mekanik sistem) AÖB kullanılmıştır. Deneysel verilerinin büyük kısmı Norveç kıyılarının dışında çalışan açık deniz gemisinden derlenmiştir. Neticede, gözlemci sonuçlarını endüstri standardı Dikey Referans Birimi (Vertical Reference Units, VRU)'larla ve Genişletilmiş Kalman filtresi (EKF)'ye dayalı navigasyon yazılımıyla kıyaslanması, iyi performans elde edilmesiyle tutum tahmini için yöntemin AÖB'nin kendisinden daha önemli olduğu gösterilmiştir.

Yapılan başka bir çalışmada [16], Mars'a iniş aşamasında Ataletsel Navigasyon Sistemi (ANS)'yi güncellemek için bir radar altimetresi ve hız ölçerin kullanımı araştırılmıştır. Başlangıç durumu hakkında bir bilgiye sahip olmadan halihazırdaki sensörlerden yararlanan yeni bir çevrimiçi yönlenme algoritması önerilmiştir. Çevrimiçi yönlenme algoritmasının başlangıç adımı, yeni bir eylemsizlik çerçevesi belirlemek ve nadir vektörün hesaplanmasının akabinde navigasyon; konum, hız ve tutumunun başlatılabileceği açıklanmıştır. Başlatma işleminden sonra AÖB'den çıkan tahminler radar verileriyle birleştirilmiştir. Sayısal simülasyonla EKF destekli radar güncelleme, ataletsel navigasyonun uygulanabileceği gösterilmiştir.

Won-Yeol Kim ve çalışma arkadaşları [17], bir kullanıcının güzergahını izlemek üzerine bir çalışma yapmışlardır. Güzergâh takibi için kullanıcının cihazında mevcut mikro elektro-mekanik sistem tabanlı ataletsel ölçüm biriminden alınan ivme ve açısal hız verilerinin birleştirilmesi gereklidir. AÖB'de sürekli kaymalar meydana geldiğinden dolayı hız tahmini için 6 Eksenli AÖB tabanlı Ataletsel Odometri Sinir Ağı (Inertial Odometry Neural Network, IONet) kullanılmıştır. Bunun sonucunda sürüklenme problemiyle karşılaşmış ve 9 Eksenli IMU tabanlı IONet ve Pose-TuningNet'i birleştiren Genişletilmiş IONet'i önerilmektedir. Oxford Ataletsel Odometri Veri Kümesi ile yaptıkları deneylerle

önerdikleri sinir ağının performansını doğrulamışlardır. Genişletilmiş IONet algoritmasını 6 Eksenli IONet ile karşılaştırdıklarında yedi farklı senaryonun beşinde üstün performans elde etmişlerdir. Çalışmanın genelinde %39,8 RMSE iyileştirmesiyle olumlu sonuçlar almışlardır.

Ampütasyonlu bireyler gibi yürüme bozukluğu olan insanlarda Body Center of Mass (Vücut Kütle Merkezi) (BCoM) 3 boyutlu kinematik analizi bireyin hareketi hakkında önemli bilgiler barındırmaktadır [18]. Buradan yola çıkarak Emeline Simonetti ve çalışma arkadaşları [18] Vücut Kütle Merkezi ivmesini ve hızını elde etmek için giyilebilir, Manyeto-Atalet Ölçüm Birimi (MAÖB) kullanımına dayalı bir yöntem önermişlerdir. Önerdikleri yöntemi, kuvvet plakalarından edindikleri ivme ve optoelektronik bir sistemden aldıkları hızlanma ve hız verileriyle doğrulamışlardır. Bireyin vücuduna yerleştirdikleri 5 adet MAÖB ile yüksek doğrulukta hareket saptama (korelasyon katsayısı $\geq 0,94$) ve 3B hızlanma için Normalleştirilmiş Ortalama Karekök Hatası ($\text{NRMSE} \leq \%14$ ve 3B hız için $\leq \%15$) elde etmişlerdir. Bu çalışmalarla ataletsel sensörlerin, kinematik analiz için laboratuvar ekipmanlarına bir alternatif olabileceklerini göstermişlerdir.

Ankit Jain ve çalışma arkadaşları [19], multiGNSS Doppler verileriyle kentsel yerleşim alanlarında hız tahmini için iki farklı varyans modelinin başarı durumlarını kıyaslamaktadır. Bir araçla iki saat boyunca kentsel bir alanda, Küresel Uydu Seyrüsefer Sistemi (Global Navigation Satellite Systems, GNSS) ve AÖB verileri toplanmıştır. Farklı varyans modellerinin tahmin başarılarını bulmak için; Doppler verileri, En Küçük Kareler (Least Squares, LS) ve Doğrusallaştırılmış Kalman Filtresi (Linearized Kalman Filter, LKF) ile hız kestirimi yapılmıştır. Edinilen hız kestirimleri bir referans yörünge ile karşılaştırılıp varyans modellerinin sapmaları değerlendirilmiştir. SIGMA- ϵ modeli, yüksekliğe bağlı varyans modeline karşılık kestirim hız ortalama kare hatası (RMSE) asgari %16'dan az, azami yaklaşık %41 olarak bulunmuştur. İlaveten SIGMA- ϵ modeli ile maksimum sapmalar büyük oranda azaltılmaktadır.

Yürüyüş analizi üzerine yapılan bir başka çalışmada [20], alt bacak bölgesine yerleştirilen AÖB ile Ters Çevrilmiş Sarkaç Modeli (Inverted Pendulum Model, IPM) uzamsal yürüyüş değişkenlerini uygun bir şekilde tahmin edebilmiştir. Optik hareket yakalama mekanizması ile AÖB temelli yürüme analizi doğrulanmaktadır. Bu süreç de adım hızını, uzunluğunu ve bacağın dikey yer değiştirme mesafesini tahmin eden yürüyüş analizi için IPM kullanma ihtimali araştırılmıştır. IPM yöntemiyle kestirim yapılan her değişkenin doğruluğu, sıfır hız güncellemesine dayalı başka bir yöntem kullanılarak karşılaştırılmıştır. Önerilen yöntemle 0.006m'lik adım uzunluğu ortalama hatası elde edilmiştir. Gelecekteki

alıřmalar da klinik teřhisler iin elde edilen sonuların alternatif bir yntem sunacađı belirtilmektedir.

Ameen M. Bassam ve arkadařları tarafından yapılan alıřmada [21], gemi performanslarının optimizasyonu zerine nemli etkisi olan gemi hızının tahminin yksek dođrulukta gerekleřtirilmesi zerine alıřmalar yapmıřlardır. Eriřim kolaylıđı ve veri setinin kaliteli olmasından dolayı "M/S Smyril" feribotunun operasyonel veri setini kullanarak eřitli makine đrenme yntemleri ile hız tahmini yapılmıřtır. Yntem olarak da gemi hızı tahmininde yksek dođruluk sađlayan regresyon algoritmalarının performans ıktıları karřılařtırılmıřtır. Ayrıca alıřma yapılan modellerin daha iyi performans sađlamaları iin nerilerde bulunulmuřtur. Bunun yanı sıra eřitli veri lekleme yntemlerinin, hız tahmini yapılırken regresyon modellerinin bařarılarına etkisi hakkında iře yarar bilgiler sađlanmıřtır. alıřmalar sonucunda diđer yntemlere gre daha fazla eđitim sresi ile en bařarılı sonucu GPR yntemi vermiřtir. Duyarlılık analizinde SVM yntemlerinin, apraz dođrulama katsayıları 5'ten 50'ye kadar ykseltilebilmektedir. Bunun sonucunda kbik Gauss Destek Vektr Makinesi (Gaussian Support Vector Machine, GSVM) modelinin RMSE deđeri %11 ve ince Gaussian SVM modelinde de %3'lk bir dođruluk artıřı elde edilmiřtir.

Hakan Aıkgz ve Deniz Korkmaz [22], asenkron motorun rotor hız tahmin modeli zerine alıřma yapmıřlardır. Bu alıřmada, hız tahmini iin LSTM derin đrenme ađı kullanılmıřtır. alıřmada kullanılan veri seti MATLAB/Simulink yazılım paketi ile elde edilmiřtir. Test kısmında kullanılması amacıyla farklı veri setleri de oluřturulmuřtur. Derin đrenme ađındaki nron sayılarının artıřına karřılık performans verilerini inceleyip en verimli sonu verecek nron sayısını bulmuřlardır. Tasarladıkları modeli eđitim srecinden geirmıřlerdir ve test verileri ile modelin dođruluđunu sınamıřlardır. Yaptıkları alıřmalar sonucun da en iyi performansı RMSE cinsinden 0,0123 olarak elde etmiřlerdir. Tasarladıkları modelin deđiřken hızlar altında daha dođru sonuları vereceđi ve gml sistemlerde rahata uygulanabileceđi sonucuna varmıřlardır.

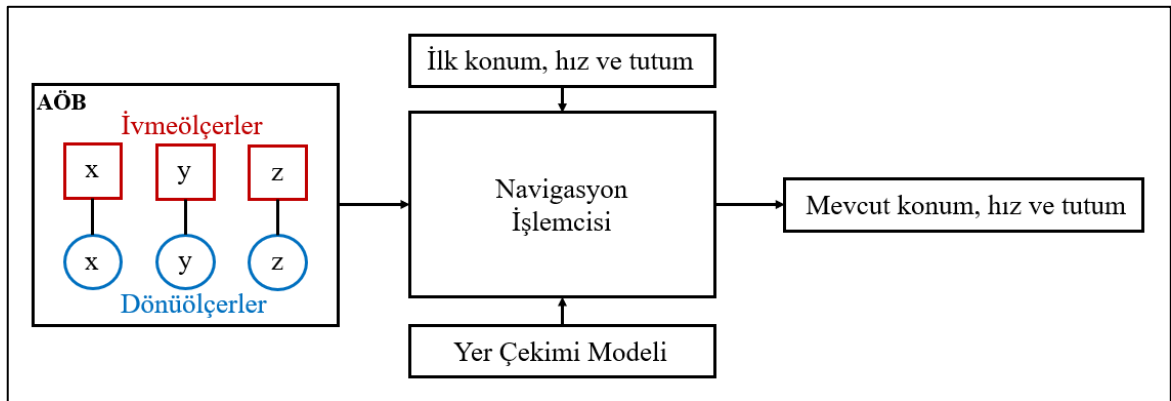
2. ATALETSEL NAVİGASYON SİSTEMİ

2.1. Sistemin Tanımı ve Mimari Yapı

Herhangi bir kara, deniz, hava aracı veya bir yayanın konumunu belirlemek için kullanılan sistemler navigasyon sistemleri olarak adlandırılmaktadır. Küresel Konumlama Sistemi (KKS) ve Ataletsel Navigasyon Sistemi (ANS) halihazırda kullanılan önemli navigasyon sistemlerindendir. Kullanım açısından ANS bazı durumlarda tek başına kullanılırken bazı dezavantajları gidermek için KKS ile entegre bir şekilde kullanım alanları da bulunmaktadır.

Ataletsel Navigasyon herhangi bir harici algılayıcıya ihtiyaç duymadan kendi dâhili sensörleri yardımıyla ölçtüğü ivme veya dönü verilerini navigasyon bilgisayarında tümlev alma işleminden geçirdikten sonra hız, konum ve pozisyon verilerini elde eden sistemdir. Bu sistem tamamen bağımsız çalışır ve herhangi bir dış elektromanyetik uyarana ihtiyaç duymaz. Bu çalışma şekliyle Ataletsel Navigasyon Sistemi kendisine çeşitli avantajlar sağlar.

Ataletsel Navigasyon Sistemi kendi aralarında birbirlerine dik olacak şekilde konumlandırılmış ivmeölçer ve dönüölçer sensörlerinden oluşmaktadır. ANS, bu sensörlerden aldığı verilere tümlev alma işlemi uygular ve kendisine gelen konum bilgisinden de yola çıkarak monte edildiği aracın mevcut konumuna, hızına, yönelim bilgisine ulaşır. Ataletsel navigasyon sisteminin mimari yapısını oluşturan temel yapı taşları Şekil 2.1’de verilmiştir.



Şekil 2.1. Ataletsel Navigasyon Sistemi Yapı Taşları [23]

Ataletsel Navigasyon Sistemleri, sensörler yardımıyla birikimli olarak sonuç ürettiği için zamanla bu sensörlerden kaynaklı hatalar birikmeye başlar. Hataların birikmesi

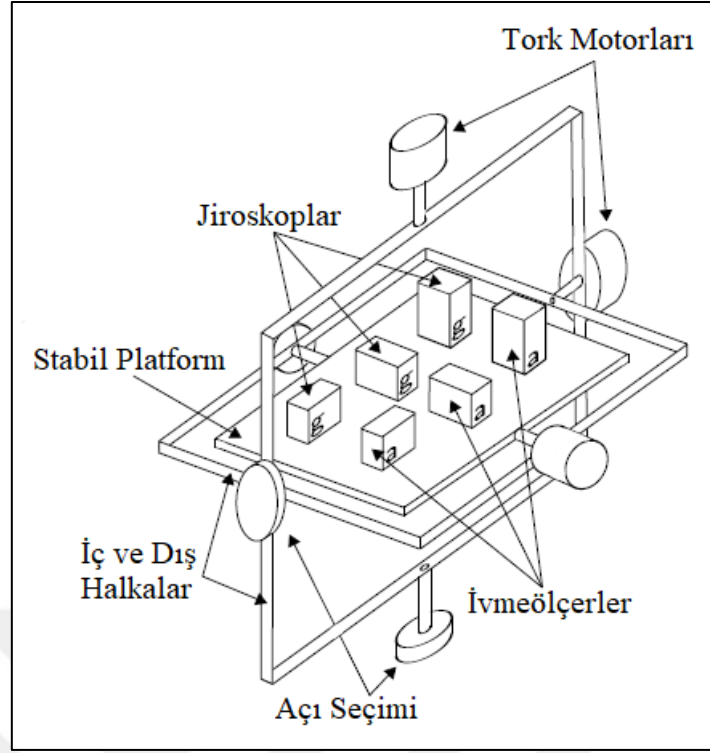
ANS'nin konum hassasiyetini azaltmaktadır. Bundan dolayı genel kullanımda ANS ve KKS sistemlerinin beraber kullanılması tercih edilmektedir. Bu kullanımda temel amaç birbirlerinin eksik taraflarını kapatmaktır. ANS'nin gösterdiği veriler, ölçüm alınan sensörlerden kaynaklı zamanla biriken bir hata payına sahiptir. Ayrıca, ANS çalışmaya başlamak için ilk konum bilgisine ihtiyaç duyar. KKS ise uydular sayesinde işleyen bir navigasyon sistemi olduğundan başlangıç bilgisine ihtiyaç duymamakta, birikimli bir hata oluşmamakta, ancak herhangi bir sinyal kaybında ise tamamen çalışmaz duruma gelebilmektedir. Bu nedenle, bu iki sistemin beraber kullanımı bu sıkıntıları ortadan kaldırmaktadır. ANS ve KKS özelliklerinin karşılaştırılması Tablo 2.1'de verilmiştir.

Tablo 2.1. ANS ve KKS özelliklerinin karşılaştırma tablosu [24]

Özellikler	ANS	KKS
Navigasyon Doğruluğu	Zamanla kötüleşen kısa vadede yüksek doğruluk	Uzun vadede yüksek doğruluk
Bağımsız	Evet	Hayır
Başlangıç Konum Bilgisi	İhtiyacı var	İhtiyaç yok
Yerçekimi Hassasiyeti	Evet	Hayır
Parazite Karşı Dayanıklılık	Evet	Hayır
Yönelim Bilgisi	Mevcut	Mevcut
Veri Hızı	Yüksek hız	Düşük hız

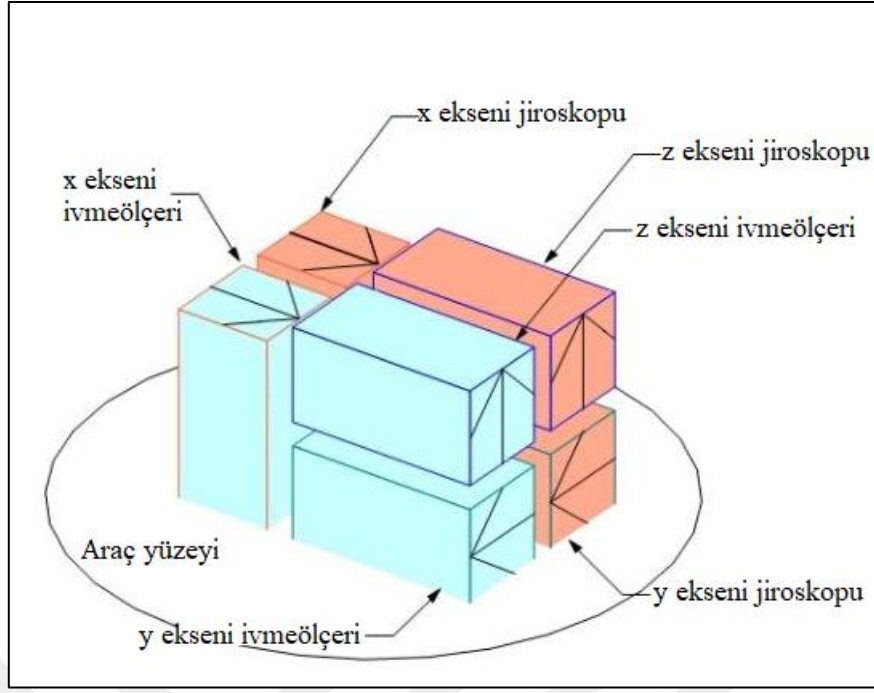
ANS donanımsal olarak dengeleme halkalı ve direk gövdeye bağlı sistemler olarak iki grupta incelenmektedir. Bu sistemlerin kendi içlerinde farklı avantajları ve dezavantajları vardır.

Halkalı sistemlerde, Ataletsel Ölçüm Birimleri navigasyonun kullanılacağı araca yalpalama özelliğine sahip bir halka ile monte edilir. Ataletsel Ölçüm Birimi içerisinde birbiriyle 90° açı yapacak ve her eksene bir tane olacak şekilde yerleştirilmiş üçer tane ivmeölçer ve dönüölçer bulunmaktadır. Halkalı sistemlerde amaç sensörlerin aracın navigasyon çerçevesi ile aynı doğrultuda kalmasını sağlamaktır. Bunu gerçekleştirmek için dönüölçerlerden gelen verilere karşılık olarak bir tork motoru yardımıyla halkalı sistem platformu tekrar aynı doğrultuya gelecek şekilde döndürür. Bu işlem sonucunda AÖB'nin diğer sensörü olan ivmeölçerler navigasyon çerçevesindeki konum ile hız için düzeltilmiş olur. Böylece sistem stabil hale getirilmiş olur. Bu sistemler mekanik olarak karmaşık ve maliyetleri yüksek sistemlerdir. Aşağıda Şekil 2.2'de halkalı bir sistem örneği verilmiştir.



Şekil 2.2. Halkalı Bir Sistem Örneği [25]

Direk gövdeye bağlı sistemler, adından da anlaşılacağı üzere navigasyon sistemini kullanan aracın üzerine sabit bir şekilde monte edilir. Bu sistem de halkalı sistemlerde olan yalpalama yapan halkanın yerine bir bilgisayar vardır. Bu bilgisayarın görevi Ataletsel Ölçüm Birimi içerisindeki dönüölçerlerden aldığı bilgileri bir algoritma yardımıyla işleyerek aracın gövdesinin doğrultusu ile navigasyon çerçevesi arasındaki değişimi devamlı olarak güncellemektir. İvmeölçerlerden elde edilen veriler navigasyon çerçevesindeki ivmeleri elde etmek için bilgisayarın algoritmasından geçirilir. Daha sonra bilgisayarda işlenen ivme verileri de hıza entegre edilir. Direk gövdeye bağlı sistemlerde açısal durumun hesaplanması ve ivmenin dönüştürülmesi karmaşık işlemlerdir. Bu sistemler bilgisayar teknolojisinin gelişmesiyle ve optik dönüölçerlerin ortaya çıkmasıyla kullanılabilir olmuştur. Direk gövdeye bağlı sistemlerin; basit mekanik, geniş bant olanağı, çalışma sıcaklık aralığının yüksek oluşu ve artırılmış güvenilirlik gibi avantajları mevcuttur [26]. Söz konusu sistemlerin donanımsal olarak karşılaştırması Tablo 2.2’de, direk gövdeye bağlı bir sistemin temel görünümü ise Şekil 2.3’te verilmiştir.



Şekil 2.3. Direk Gövdeye Bağlı Bir Sistemin Temel Görünümü [27]

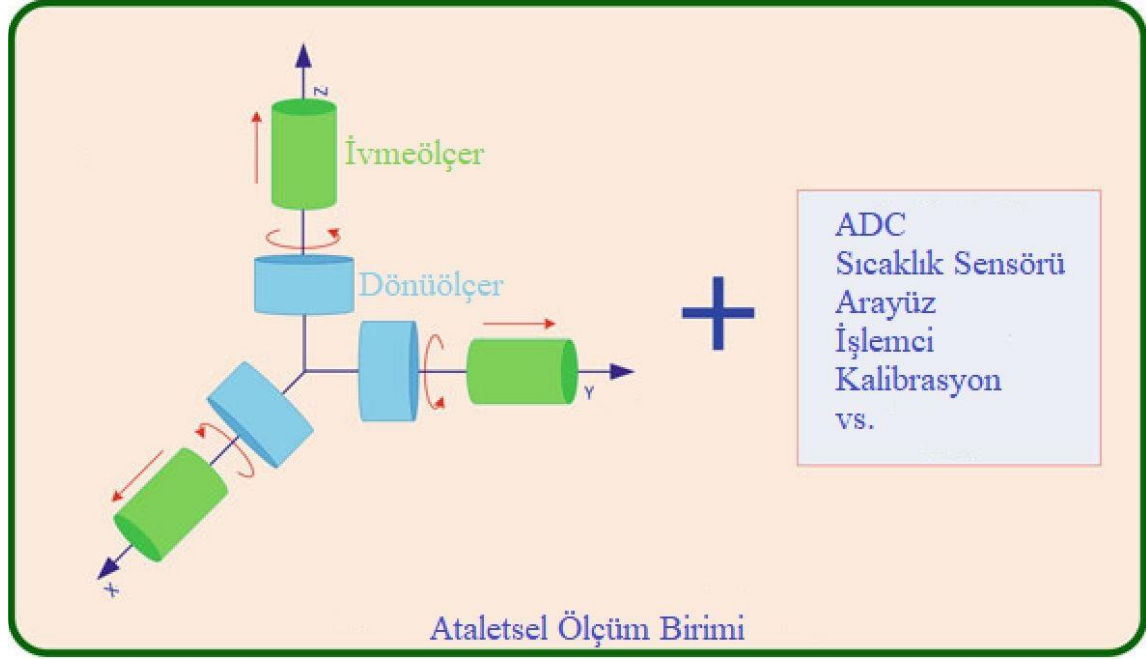
Tablo 2.2. Donanımsal Olarak Sistemlerin Karşılaştırma Tablosu [24]

Özellikler	Dengeleme Halkalı Sistem	Direk Gövdeye Bağlı Sistem
Performans	Daha Yüksek	Yüksek
Yük	Ağır	Hafif
Boyut	Büyük	Küçük
Sağlamlık	Yüksek	Daha Yüksek

2.2. Ataletsel Ölçüm Ünitesi

Bir cismin yerinin tam olarak belirlenebilmesi için o cisme ait hız ve yönelim verilerinin bilinmesi gerekmektedir. Bu bilgiler ataletsel ölçüm ünitesinden elde edilmektedir. Ataletsel ölçüm ünitesi, entegre bir sistemdir. Bu sistem ivmeölçer ve dönüölçerden oluşmaktadır. Sistemdeki ivmeölçerler bağlı olduğu cismin doğrusal ivmesini ölçmede yardımcı olmaktadır. Bu sistemin diğer bileşeni olan dönüölçerler ise bağlı olduğu cismin açısal hızını ölçmektedir. İvmeölçerden gelen sensör verisinin integrali alınarak sensörün bağlı olduğu cisme ait hız verisi elde edilmektedir. Tek başına hızın bilinmesi

yetersiz olacağından dolayı dönüölçer sensörlerinden gelen açısal hız verisi yardımı ile cismin yönelim açısı doğrultusunda tam yeri hesaplanmaktadır. Bu tür sistemlerde hatalar birikerek artmaktadır. Bundan dolayı kullanılan sensörlerin hassasiyeti ve kalitesi önem arz etmektedir. Tipik bir ataletsel ölçüm birimi bileşenleri Şekil 2.4'te gösterilmektedir.



Şekil 2.4. Tipik Ataletsel Ölçüm Birimi Bileşenleri [24]

2.3. Ataletsel Sensörler

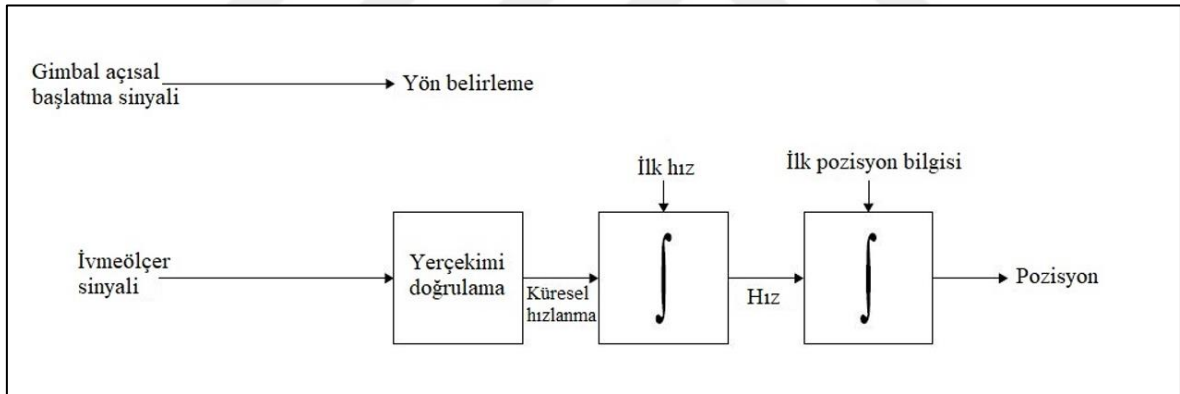
İvmeölçer ve dönüölçer sensörleri ataletsel sensörlerdir. İvmeölçerler bağlı olduğu cismin tek eksenindeki etkin kuvvetini ölçerken dönüölçerler tek eksenindeki açısal hızı ölçmektedir. Bu sensörler bir cisme yerleştirileceği zaman genellikle her ekseninde ölçüm elde edebilmek için üç tane birbirlerine dik olacak şekilde yerleştirilirler. Bu üçlü sensör gruplarının birleşimine de ataletsel sensör birimi denir.

Ataletsel sensörler için yüksek kalite gereken noktalarda ise bu sensörlerin mekanik bileşenlerinin hava geçirmez bir şekilde yalıtılması önemlidir. Hermetik sızdırmazlık sensörün çalışmasına etki edecek kirleticilere, neme ve parçacıklara (partiküllere) karşı koruma sağlar [28].

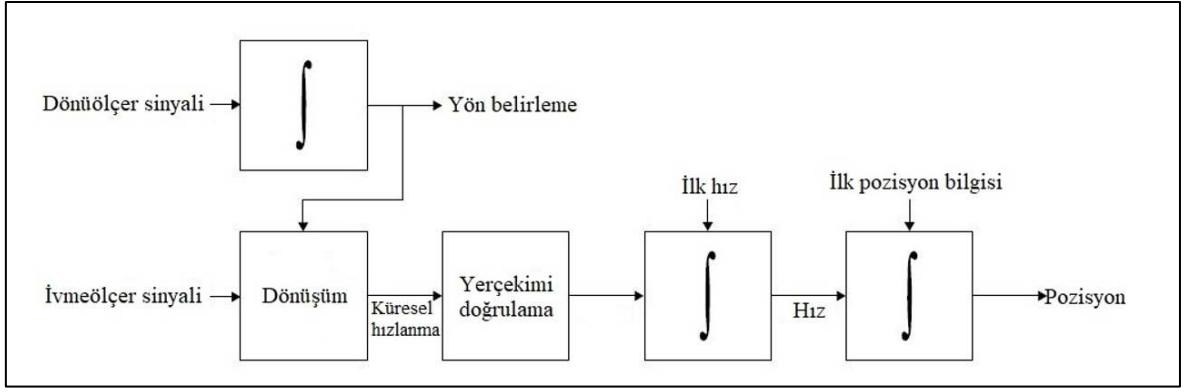
İvmeölçer ve jiroskoplardan oluşan atalet sensörleri silikon tabanlı sensörlerin en önemli çeşitlerinden biridir. Sensör alanında ivmeölçerler, basınç sensörlerinden sonra en büyük satış hacmine sahip ikinci sensör grubudur. Dönüölçerlerin de zamanla satış hacminde ivmeölçerlere yetişeceği düşünülmektedir. İvmeölçerlerin talep gördüğü alanların en büyükleri, hava yastıklarının da içinde bulunan güvenlik sistemlerini aktif etmek, araç

kararlılık sistemlerinde ve elektronik süspansiyonu uygulamak için kullanılan otomotiv alanlarıdır. Bununla birlikte ivmeölçerleri düşük maliyetleri ve küçük boyutları sayesinde uygulama alanları oldukça geniştir. Yüksek duyarlılığa sahip ivmeölçerler, rehberlik sistemlerinde, deprem tahmininde, bağımsız navigasyonda, petrol aramada, uzayda mikro yerçekimi ölçümlerinde ve platform stabilizasyon sistemlerinde çok önemli bileşenlerdir [29].

Cihazın konumunu hesaplamak için platforma monte ivmeölçerlerden gelen sinyallerin çift olarak entegre edildiği bir düzenek Şekil 2.5 ile verilmektedir. Hesaplamalar yapılmadan önce dikey kanaldan yerçekiminden kaynaklanan ivmeyi çıkarmak gerekmektedir. Halkalı sistem platform atalet navigasyon algoritması Şekil 2.5'te gösterilmektedir. Şekil 2.6'da ise direk gövdeye bağlı atalet navigasyon sistemi algoritması blok diyagramı verilmektedir. Verilen blok diyagram incelendiğinde dönüölçerlerden gelen açı bilgisi ile ivmeölçerlerden gelen ivme verileri dünya koordinat sistemine döndürülür. Dünya koordinat düzlemine çevrilen ivme verilerinden yerçekimi çıkarılarak ivme verileri temizlenmiş olur. Temiz ivme verilerinden hız verileri hesaplanır. Sonraki adımda bu hız verilerine tümlev alma (integral alma) uygulanarak pozisyon bilgisine ulaşılır.



Şekil 2.5. Halkalı atalet navigasyon sistemi algoritması [25]

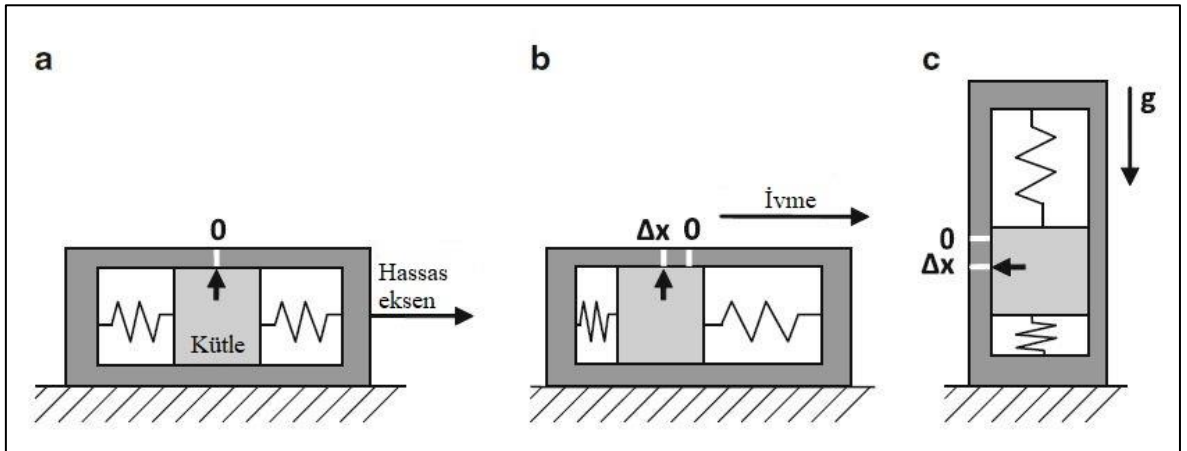


Şekil 2.6. Direk gövdeye bağlı atalet navigasyon sistemi [25]

2.4. İvmeölçerler

Bir ivmeölçer, üzerine tesir eden kuvvetlerden kaynaklı öteleme ivmesini ölçen bir cihazdır. Bu ivme, ivmeölçer sensöründe referans çerçevesinde mevcut bir kütlenin deneyimlediği ağırlık ile ilgilidir. Bu durum Newton'un ikinci hareket kanunu ile tanımlanmaktadır. "M kütleli bir cisme etkiyen bir F kuvveti, cismin eylemsiz uzaya göre hızlanmasına neden olur."

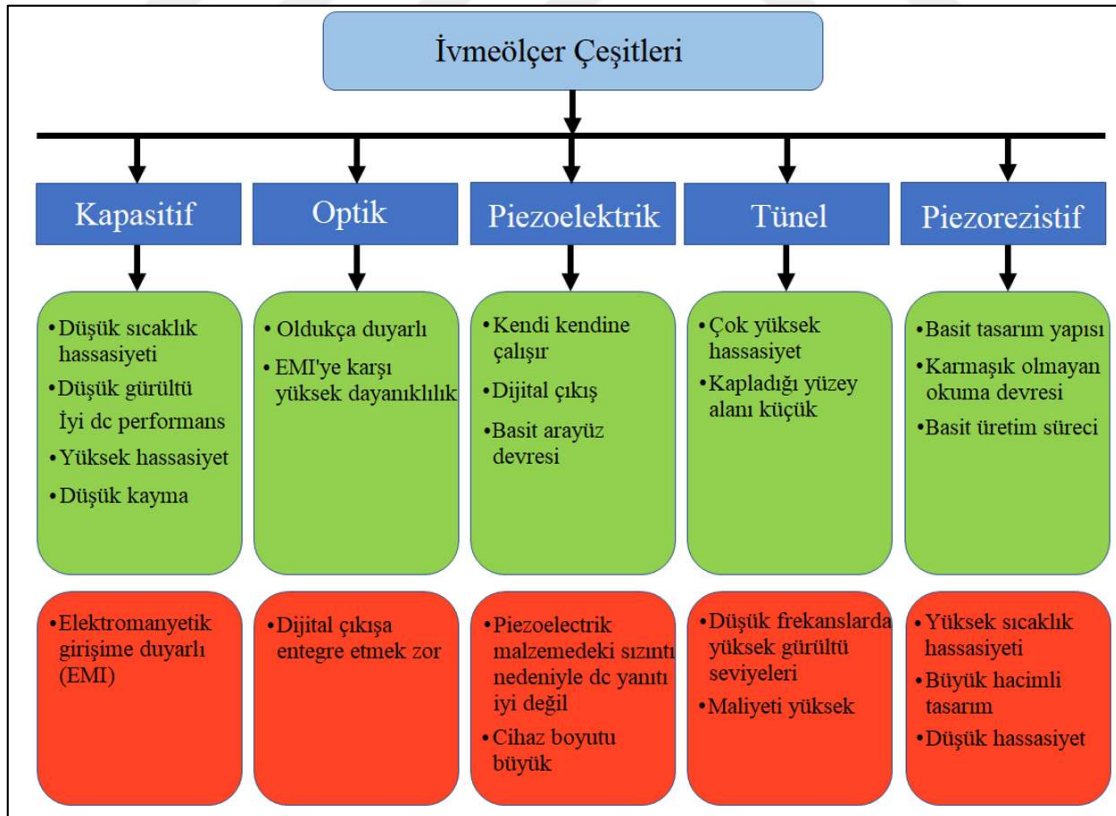
İvmeölçer sensörünün yerleştirildiği cismin hassas ekseni boyunca ivme yokken, kütle Şekil 2.7 a'daki gibi gözüktür. Hassas ekseni boyunca ivme etkisi altında olduğunda ise kütle bulunduğu konuma göre yer değiştirir (Şekil 2.7 b). Herhangi bir ivmelenme olmadığında, kütle üzerindeki kuvvet, yayın gerilimiyle dengede kalır. Yayın sıkışması veya uzaması, kütlenin yer değiştirmesi ile orantılı bir kuvvet yaratır. Kütlenin hareketine karşı herhangi bir sürtünme kuvveti olmadığında, kütlenin yer değiştirmesi ivme ile doğru orantılıdır. Cisme uygulanan ivme yaya bağlı kütlenin yer değiştirmesi ölçülerek bulunabilir [30].



Şekil 2.7. Farklı koşullarda kütle ve yay ivmeölçeri [30]

İvmeölçerler, sensörün içindeki kütleyi ve kütlenin içinde bulunduğu kasayı aynı ölçüde etkileyen ivmeleri algılayamaz [31]. Örnek olarak bir ivmeölçerin serbest düşüşünü inceleyelim. İvmeölçer sensörünün kütlesi, sensör kasasının içerisinde olduğundan dolayı kütle ile kasa birlikte düşeceklerdir. Bu sebeple kasa içerisindeki kütlenin konumu kasaya göre herhangi bir değişiklik göstermeyecektir. Başka bir deyişle, ataletsel olarak hassas eksene göre cismin ivmesi a , yerçekimi ivmesi olan g 'ye eşittir. Bununla birlikte özgül kuvvette sıfırdır. İvmeölçerin sabit durması, $a = 0$ olması, Şekil 2.7 c'deki durumunda olduğu gibi düşmeyi durdurmak için karşıt etkileyen kuvvet $\mathbf{f} = -\mathbf{mg}$ olarak ölçülecektir. Bu kuvvet yerçekiminin etkisini dengelemek için gereklidir ve ivmeölçer tarafından gelen ölçümler, sensör ünitesinin atalet alanına göre ivmesini belirlemek için yerçekimi alanı bilgisi ile birleştirilmelidir [30].

Birçok ivmeölçer tasarımı mevcuttur. İvme ölçerlerin büyük çoğunluğu basit yay ve kütle sistemine benzer bir mantıkta çalışır. Üçboyutlu hızlanma vektörünün gerekli olduğu birçok uygulama vardır. Genellikle bu uygulamalarda üç adet bir eksenli ivmeölçerler kullanılmaktadır. Üç eksenli düşük maliyetli ivmeölçerler son yıllarda çok rağbet görmektedir [30]. İvmeölçer türlerinin avantajları ve dezavantajları Şekil 2.8'de özetlenmektedir.



Şekil 2.8. İvmeölçer Çeşitleri hakkında karşılaştırmalı bilgi [32]

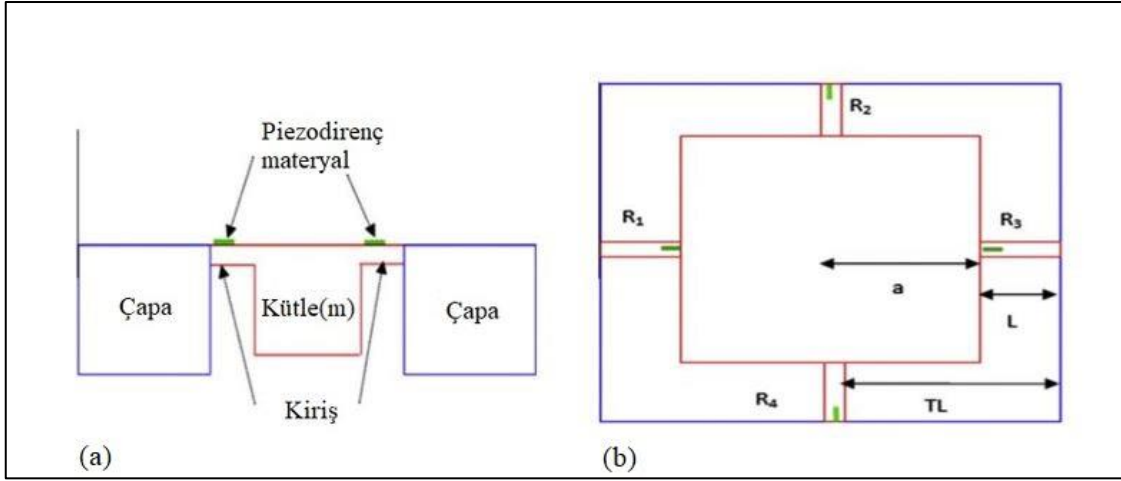
Mikro elektromekanik sistemlerin (MEMS) bileşenleri; aktüatörler, mikro mekanik sensörler ve mikro elektronik devrelerdir. Mikroelektronik teknolojinin gelişmesi ile birlikte MEMS'in gelişimi; mikro mekanik algılayıcıların, aktüatörlerin ve mikro mekanik dönüştürücülerin araştırılmasına ve gelişimine yoğunlaşmıştır [33]. Bununla birlikte, mikro ivmeölçerlerin araştırma konularının kapsamı genişledi ve maliyete duyarlı ticari ürünler ortaya çıkmaya başladı [34].

2.4.1. MEMS ivmeölçerler

MEMS ivmeölçerler algılayıcı yapılarına göre sekiz gruba ayrılabilir [35]:

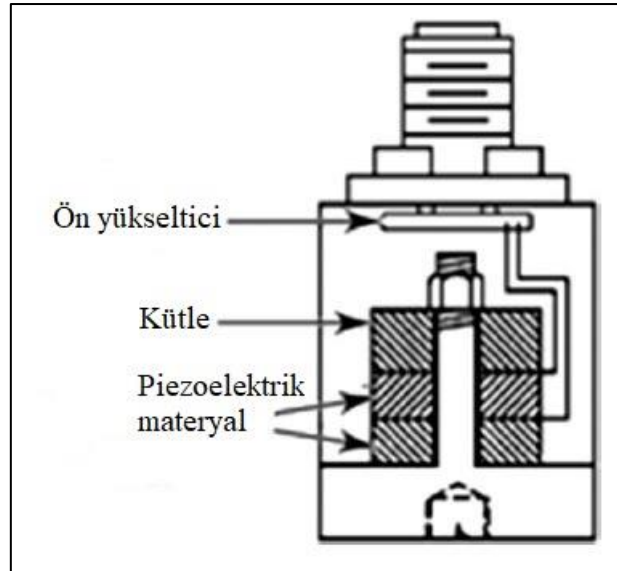
- Piezodirençli
- Piezoelektrik
- Termal
- Tünelleme Akımlı
- Manyetik
- Rezonans
- Kapasitif
- Optik

Piezodirençli malzemeler, üzerine düşen basınca orantılı olarak direnci değişen malzemelerdir. Bu malzemelerin özellikleri sayesinde basınç ve ivme sensörlerinde kullanılma imkânı olmaktadır. Basınç sonucu değişen direnç elektriksel sinyale çevrilerek ivme ölçümü yapmak mümkündür. Bu sensörler, oldukça yüksek doğrusallık ve küçük çıkış empedansı ile karakterize edilir. Sensördeki dirençler normalde, çıkış voltajında ivmelenme ile orantılı bir değişiklik sağlayan Wheatstone köprüsü devresiyle yapılandırılır. Wheatstone köprüsü elektriksel dirençleri ölçmek için ya da dirençleri karşılaştırmak için kullanılan bir devredir [36]. Şekil 2.9'da bir piezodirenç ivmeölçerin kesit görünümü ve üstten görünümüne yer verilmiştir [37].



Şekil 2.9. (a) Piezodirenç ivmeölçer kesit görünümü. (b) Piezoivmeölçer üstten görünümü [37]

Piezoelektrik, birtakım materyallerin (bazı kristaller ve seramikler) uygulanan basınca karşı tepki olarak bir elektrik potansiyeli oluşturma yeteneğidir. Genel bir piezoelektrik sensör, ilgili yüzeye monte edilen piezoelektrik özellikte bir kristalden meydana gelir. Elektrotlar, piezoelektrik materyalde oluşan elektrik potansiyelini(yük) algılamak için kristalin her iki ucuna da bağlanır. Bu daha sonra materyal üstündeki basınçla ilişkilendirilir [38]. Şekil 2.10’da piezoelektrik sensörün genel yapısı gösterilmiştir.



Şekil 2.10. Piezoelektrik sensör genel yapısı [39]

Termal ivmeölçerlerde, bir ısıtıcı direnci ve ısıtıcıdan eşit uzaklıkta, boşlukta asılı duran sıcaklık algılayıcısı bulunur. Geleneksel olan silikonlu modellerde boşluk silikon

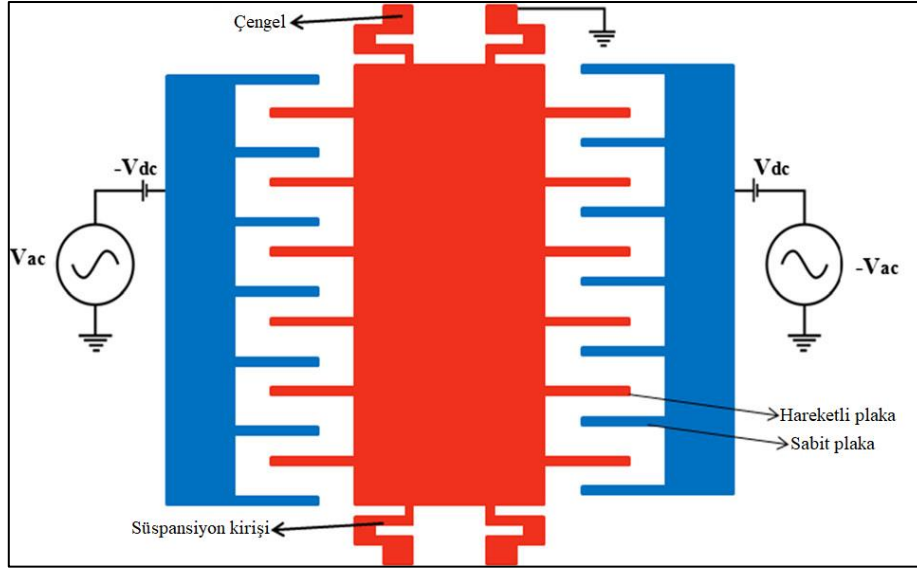
üzerine kazınır ve içerisindeki gaz kabarcığı ile hava geçirmez şekilde yalıtılır. Isıtıcı dirence elektrik akımı verilirse, ortam ısı iletimle ısıtılır. Herhangi bir ivme yokken ortamdaki ısı simetriktr. Sıcaklık algılayıcılarından alınan değerler aynıdır. Bir ivme uygulandığında ortamdaki ısı simetrisi bozulacaktır ve bu sıcaklık farkı ivmeye dönüştürölerek ölçüm alınacaktır [40].

Tünel tipi ivmeölçer, bir alt elektrot ayrılmış sivri uçlu bir kanıt kütesinden oluşur. Uç karşı tarafa yeterince yaklaştığında tünel akımı oluşur [41]. Bu iletken tabakalar arasındaki tünel akımı sayesinde ivme ölçümü yapılır.

Elektromanyetik ivmeölçerin çalışma prensibi, elektromanyetik transformatör ilkesini temel almaktadır [42]. Bu ivme ölçerlerde biri kanıt kütesinde ve diğeri de alt tabakada mevcut olmak üzere iki bobin bulunur. Primer sargıya bir kare dalga sinyali uygulanır ve sekonder sargıdaki kanıt kütesiyle alt katman arasındaki uzaklıkla orantılı olan gerilim indüklenir. İndüklenen gerilim kullanılarak ivme ölçölür.

Rezonans ivmeölçerler, bir kanıt kütesinin etkilendiğı kuvvetleri rezonans algılayıcılar vasıtasıyla frekans kayması olarak ölçer [43]. Genellikle ivmenin değışim hızının düşük olduğı alanlarda kullanılmaktadır [44]. Frekans çıkışlı bu sensörlerin hassasiyeti yüksek olup kararlılıklarıyla yüksek doğruluk ve çözünürlük sağlarlar [45].

Kapasitif ivmeölçerlerin, uygun üretim maliyetleri, küçük boyutlu olmaları, yüksek sıcaklık performansı ve hassasiyetlerinin yanı sıra CMOS ile kolay entegrasyonları sebebiyle otomotiv sanayi ve tüketici elektroniğinde kullanımı yaygındır [46]. Algılama elemanları, iki paralel kapasitör plakasından oluşur. Bu plakaların tarak şeklinde iç içe geçen parmakları vardır. Bir köprü konfigürasyonunda çalışır. Hızlanma ile orantılı çıktı üretmek için demodülasyon işlemi yapan bir devreye bağlıdır [47]. Şekil 2.11’de bir kapasitif ivmeölçer görölmemektedir. Burada ivmelenme yönüne göre hareketli plaka ile sabit plaka arasında mesafe değışerek kapasitif bir değışiklik elde edilir ve bu değışiklik elektriksel sinyale dönüştürölerek ivme ölçölür.



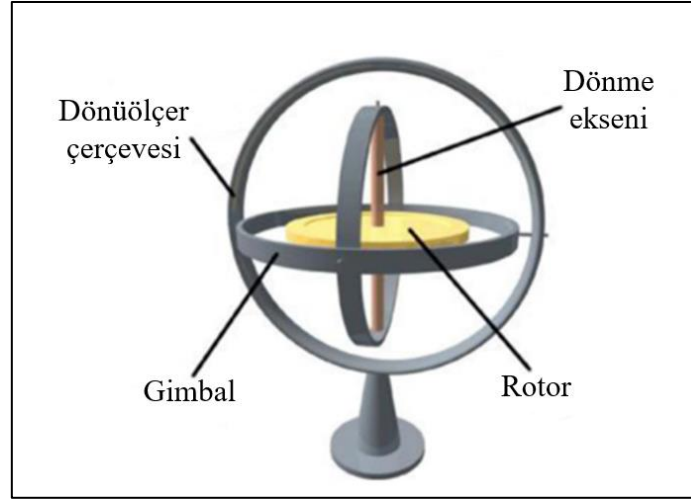
Şekil 2.11. Tipik bir kapasitif ivmeölçer [48]

Optik ivmeölçerlerde, bir ışık kaynağından gelen ışınları hareketli bir zar üstüne düşüren bir ışık yoğunluğundan istifade edilir. Bu ışık yoğunluğu ivmelenmeyle değişir. Bu yoğunluk değişimi ivmeyi ölçmek için kullanılır [41]. Optik ilkelere dayalı ivmeölçer tasarımı, geleneksel tasarımlara göre daha karışıktır. Ayrıca, optik ivmeölçer maliyeti diğer ivmeölçerlerden yüksektir. Bu yüzden, optik ivmeölçerlerin kullanım alanı sınırlıdır [49].

2.5. Dönüölçerler

Dönüölçerler, bir platformun sabit bir eksene göre açısal hızını bulan algılayıcılarıdır. Dönüölçer teknolojisi, açısal momentumun muhafaza edilmesi, Sagnac ve Coriolis etkilerine dayalı olarak farklı türde dönüölçerler geliştirmek, kullanımını uygun hale getirmek ve üretmek için araştırmalar yoğunlaşmıştır. Son zamanlarda dönüölçerlerin gelişimi, mikro fotonik ve mikro elektro-mekanik sistemlere doğru kaymaya başlamıştır [50].

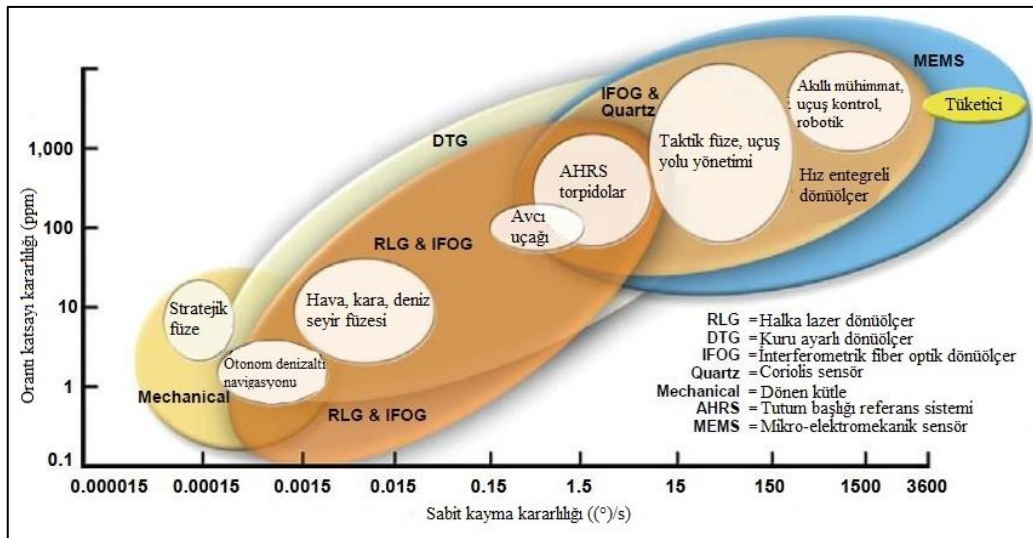
Dönüölçerlerin temel yapısı, yüksek hızda dönen bir rotorun eylemsizlik özelliklerine dayanır. Diğer bir deyişle, bir cismin momentumunun yönünde herhangi bir değişime karşı gelmesi sebebiyle dönme ekseninin yönünü tutma eğiliminde olması kuralıdır [50]. Ağır dönen bir diske sahip bir çocuk oyuncağını canlandıralım. Bu ağır disk hızla döndürüldüğünde, diskin eksen, gimballerin döndürülmesine rağmen aynı istikameti göstermeye devam eder. Bu, bir dönüölçerinin genel çalışma prensibidir [51]. Şekil 2.12’de tipik bir dönüölçer yapısı gösterilmektedir.



Şekil 2.12. Tipik bir dönüölçer yapısı [52]

Dönüölçerler, navigasyon, stabilizasyon, otopilot geri bildirimi, uçuş yolu sensörü veya platform stabilizasyonu gibi çeşitli uygulamalarda kullanılır. Modern dönüölçerlerle, tek bir algılayıcının bu gibi uygulamaların her birini yerine getirmesi olağandır, ancak genelde iki veya daha fazla algılayıcı kümesi kullanılır [53].

Çalışma prensibine ve teknolojik alt yapısına göre birçok dönüölçer mevcuttur. Dönüölçerler; yaygın olarak kullanılan mekanik dönüölçerler, Halka Lazer Dönüölçerleri (Ring Laser Gyroscope, RLG), Optik Dönüölçerler, Fiber Optik Dönüölçerler (Fibre-Optic Gyroscope, FOG) ve Mikro-elektromekanik (MEMS) dönüölçerler olarak sınıflandırılırlar [54]. Şekil 2.13'te çeşitli görevler için orantı katsayı kararlılığı ve sabit kayma kararlılığı gereksinimlerinin hangi uygulamalarda karşılandığı ve kullanıldığına ilişkin bir şema verilmektedir [55].



Şekil 2.13. Güncel dönüölçer teknolojisi uygulamaları [55]

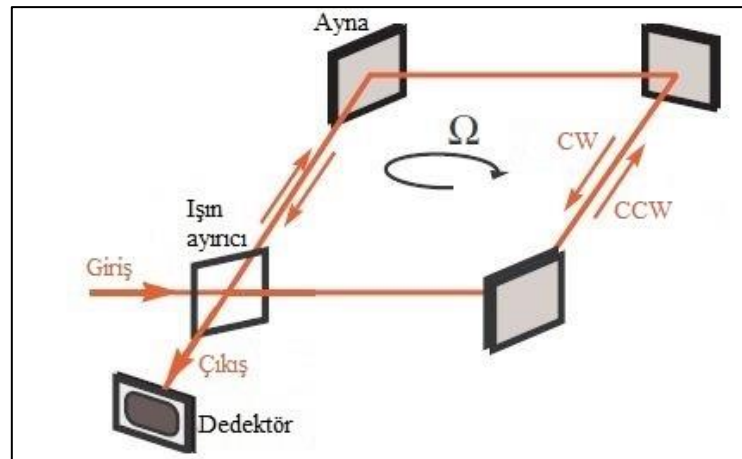
Mekanik dönüölçerler, 19. yüzyıldan beri kendi ekseninde dönen bir rotordan oluşan tarihsel algılayıcıdır [54]. Mekanik dönüölçerlerin sabit kayma kararlılığı ve orantı katsayı kararlılığı dikkate alındığında diğer dönüölçerlere kıyasla düşük olduğu görülmektedir (Şekil 2.13). Gelişen teknolojiyle beraber RLG ve FOG dönüölçerler geliştirilmiş ve daha kararlı oldukları görülmüştür.

MEMS dönüölçerler boyutlarının küçük olması, üretim maliyetlerinin düşmesi nedeniyle tercih edilmeye başlamıştır. Otomobillerde devrilmeyi önleme, hava yastığı sistemleri, görüntü sabitleme sistemleri MEMS dönüölçerlerin kullanıldığı bazı alanlardır.

2.5.1. Optik dönüölçerler

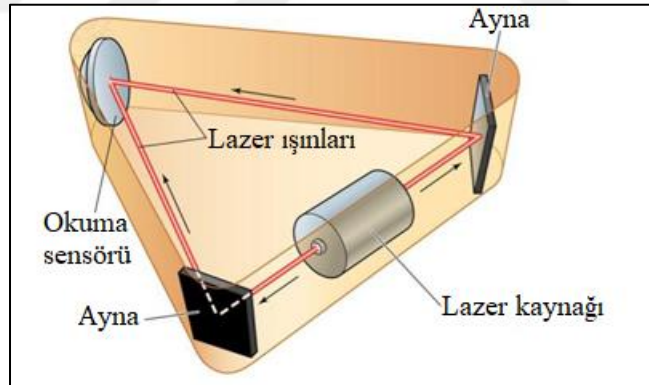
Optik dönüölçerler Sagnac etkisinden yararlanarak ölçüm yapan dönüölçerlerdir. Bu dönüölçerlerin ana tipi FOG (Fiber optik dönüölçer)'dur. Bir FOG, ışık kaynağından çıkan optik sinyalin izlediği yolu gösteren bir fiber optik bobin içerir. Akabinde, sinyali algılamak için bir algılayıcı kullanır. Dönüyü ölçmek için fiber bobine karşıt istikametlerde giren iki ışın kullanılır. Algılayıcı, bir ışık kaynağı demetiyle eş yönde dönüyorsa, ışık demeti için yol diğer yoldan daha uzun olacaktır. Bu sürece Sagnac etkisi denir [56].

Sagnac etkisi, iki ışın demeti dönen bir halka boyunca karşıt yönlerde hareket ettiğinde meydana gelir [57]. Şekil 2.14'te Sagnac etkisi daha ayrıntılı gösterilmektedir. Kapalı bir sistemde bir ışık kaynağından çıkan ışın demetleri Şekil 2.14'te görüldüğü gibi ışın ayırıcıdan geçerek aynalardan yansırken sistem Ω açısal hızıyla saat yönünde döndürülürse ışın demetleri arasında optik bir yol farkı oluşur. Bu farktan yararlanılarak açısal hızın değeri tespit edilir.

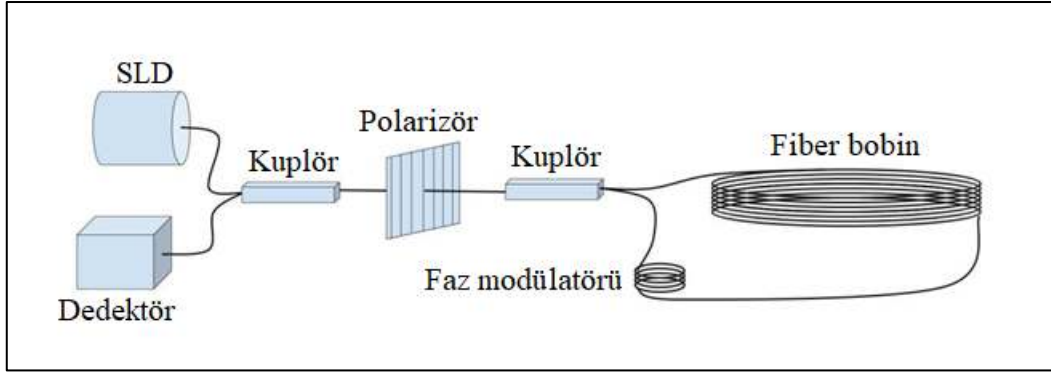


Şekil 2.14. Sagnac etkisine dayalı bir optik dönüölçer genel şeması [58]

Bir diğerk optik dönüölçer olan RLG (Halka Lazer Dönüölçer), içerişinde iki lazer ışının karşıit istikametlerde yani saat yönünde (Clock Wise, CW) ve saat yönünün tersine (Counter Clock Wise, CCW) yayıldığı çokgen kapalı bir sistemin içindeki gaz boşluğundan oluşur. İki ışın tüm boşluk boyunca dolaştıktan sonra dönüş noktasında buluşup bir girişim deseni oluşur. Bitişik girişim saçakları arasındaki yer değıştirme sensörün dönüş hızından etkilenir. Bu Sagnac etkisinin bir sonucudur. RLG bir açısal hız etkisi altında kaldığında iki ışın tarafından oluşank hareket yolu uzunluğundaki farktır. Böylece sensörün açısal hızının ölçümü iki ışının faz kayması fotodedektör yardımıyla ölçülerek sağlanır [59]. Şekil 2.15'te RLG dönüölçerlerin genel gösterimi mevcuttur ve tipik bir İnterferometrik Fiber Optik Jiroskop (Interferometric Fibre-Optic Gyroscope, IFOG) temel bir gösterimi Şekil 2.16'da görölmektedir. Işık demetleri Süper Işıklı Diyot'ta (Super Luminescent Diode, SLD) üretilir. İlk kuplörün görevi (fiber optik kabloları bağlamak için kullanılır), fiber bobinden dönen ışığın dedektöre girmesine izin vermektir. SLD'den gelen ışık polarize edilir ve ikinci bir kuplörle fiber bobinin iki ucu arasında bölünür. Faz modülatörü, fiber optik teli salınan bir çıkış yoğunluğu ile sonuçlanan bir frekansla gererek dönüş yönünü ölçmek için kullanılan bir piezoelektrik kristalden oluşur. Bu salınım için rotasyon eklenmesi, yukarı veya aşağı kaydırılan bir dalga ile sonuçlanır [60].



Şekil 2.15. RLG Örneğı [60]



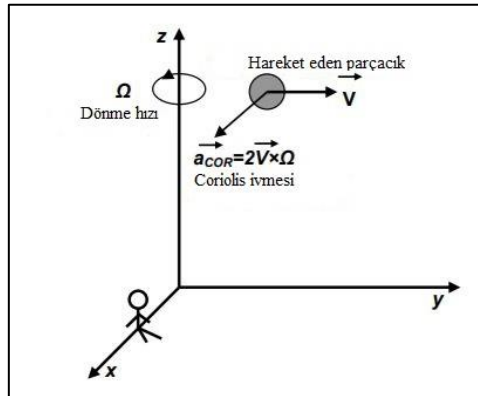
Şekil 2.16. IFOG Örneği [60]

2.5.2. MEMS dönüölçerler

MEMS dönüölçerlerin tamamına yakını dönüşü algılamak için titreşen mekanik elemanlar kullanır. Yapılarında dönen parçaları bulunmadığından rulman gereksinimleri de yoktur. Bu sebeple kolayca boyutları küçültülebilir. Silikon ya da piezoelektrik malzemelerde mikro işleme yöntemleri kullanılarak üretilirler [61]. Bu titreşimli dönüölçerler, Coriolis kuvvetinin neden olduğu enerji transferine dayanır [62].

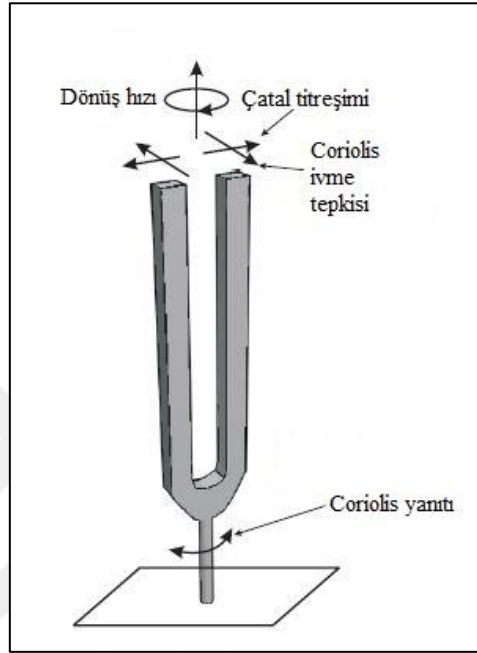
Coriolis kuvveti, kuvvet olarak adlandırılrsa da Coriolis etkisi kullanımı yaygındır. Zira, direkt olarak nesneye etki eden bir kuvvetin sebebinden ziyade dönen bir düzlem üzerinde düz bir çizgide hareket eden bir cismin algılanan hareketidir [63].

Z eksenini etrafında Ω hızıyla dönen bir koordinat sisteminin x ekseninde oturan bir gözlemcinin uzayda V hızıyla hareket halindeki parçacığı izlediğini düşünelim. Dönen koordinat sistemi sebebiyle, gözlemciye göre parçacık $2V \times \Omega$ ivme ile x eksenine doğru bir yörüngeye sahiptir. Ω hızıyla dönen bir koordinat sisteminde parçacığın hareket hızıyla (V) orantılı oluşan bu etkiye Coriolis ivmesi denir ve Şekil 2.17’de gösterilmiştir [64].



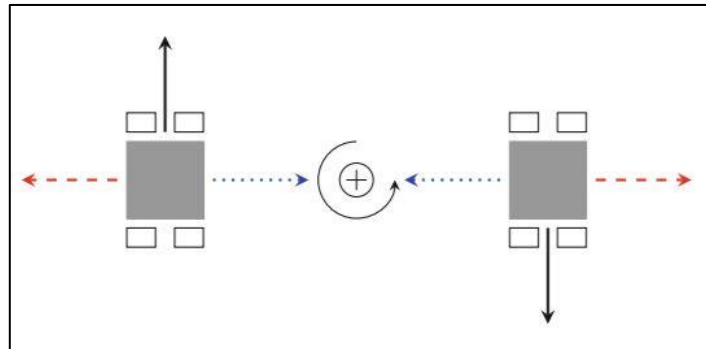
Şekil 2.17. Coriolis etkisinin gösterimi [64]

Titreşimli dönüölçerlerin hepsi, dönüşü ölçmek için Coriolis etkisini kullanırlar. Algılama elemanın rezonansı ve sistemin dönüşünden oluşan Coriolis etkisi, dönüölçerin iki titreşim çatalı arasında enerji transferine sebep olur [65]. Bu tür dönüölçerler ayar çatalı dönüölçer olarak tanımlanır. Döndürüldüğünde, Coriolis kuvveti, çeşitli mekanizmalar tarafından algılanabilen ortogonal bir titreşim yaratır [66]. Şekil 2.18’de akort çatalı dönüölçerin genel görünümü yer almaktadır.



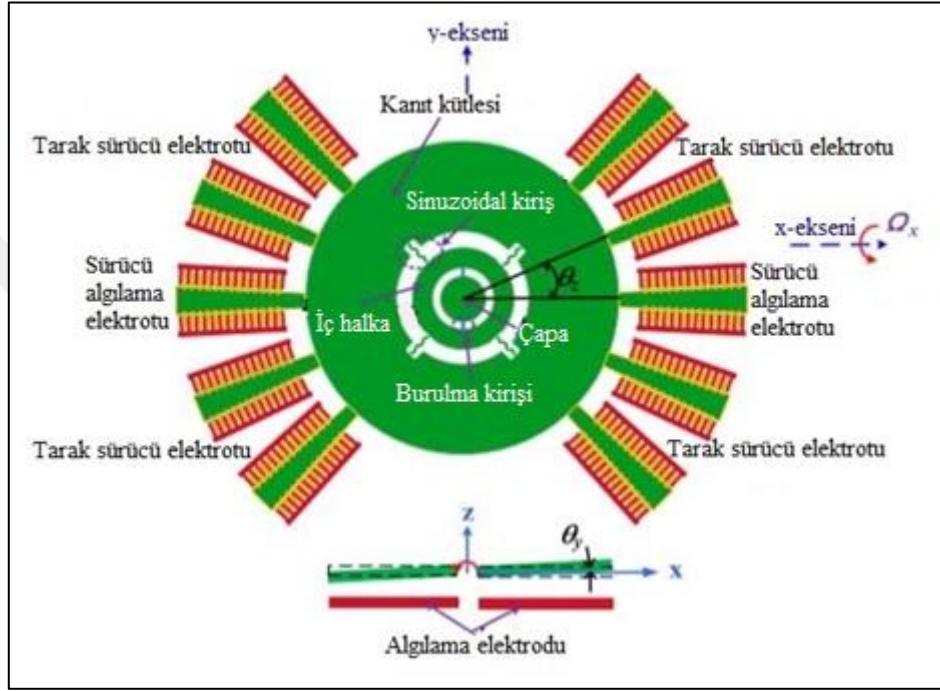
Şekil 2.18. Akort çatalı dönüölçer genel görünümü [65]

Şekil 2.19’da bir akort çatal dönüölçerin üstten görünümü sunulmaktadır. Kırmızı kesikli oklar ve mavi noktalı oklar titreşimin yönünü, düz siyah oklar Coriolis kuvvetinin yönünü göstermektedir. Kütleler (gri kareler), bir diyapazonun iki ucu gibi aynı frekansta titreşmeye zorlanır. Farklı yönlerde titreşirler. Başka bir deyişle ya birbirlerine yaklaşırlar (mavi noktalı oklar) ya da birbirlerinden uzaklaşırlar (kesikli kırmızı oklar) [67].



Şekil 2.19. Bir akort çatal dönüölçerin üstten görünümü [67]

Bir diğer dönüölçer çeşidi olan titreşimli tekerlek dönüölçerler, tekerleğin simetri eksenini çevresinde titreşmesi üzerine sürülür. Herhangi bir eksenin etrafındaki dönüş, tekerlekte eğilmeye sebep olur; bu, tekerleğin altında bulunan kapasitif elektrotlarla ölçülen bir değişimdir. Titreşimli çark sayısı bir iken iki dönme eksenini algılamak mümkündür. Mekanik rezonatörünü ve elektronik sürücüsü tek bir çipte birleştirilebilmesinden dolayı boyutlarının minyatürleştirilmesine olanak sağlar [68]. Şekil 2.20’de titreşimli tekerlek dönüölçerin üstten görünümü verilmiştir.



Şekil 2.20. Titreşimli tekerlek dönüölçerin şematik üstten görünümü [69]

Şekil 2.20’de dönüölçer, titreşimli bir tekerlek, iç halka, merkezi ankraj, bir grup tarak sürücü ve algılama elektrotundan oluşur. Dört sinüzoidal kiriş, iç halkaya ve ispat kütlesine bağlanmıştır. Büyük bir sürüş genliği θ_z elde etmek amacıyla beş tane açısız tarak kullanılır. X eksenindeki Ω_x açısal hızı için, yapı katmanının altında bulunan paralel plakalı kapasitörler, Coriolis kuvvetinin bir ölçümünü sağlayan θ_y sapma açısını ölçmek için kullanılmaktadır. Kanıt kütlesi, algılama modu için daha çok hassasiyet sağlamak amacıyla sürücünün hareketini birleştirir [69].

2.5.3. Dönüölçer teknolojilerinin karşılaştırılması

MEMS, bugüne kadar geliştirilen ilk dönüölçerlere göre ciddi gelişimler elde etti. Özellikle endüstriyel alanda ve tüketici elektroniğinde pazar talebini karşılayarak yüksek sağlamlık ve etkili performans seviyeleri sağlamaktadır [54].

Endüstriyel uygulamalarda, geri besleme kontrolünün zaruri olduğu düzeneklerin çoğu, (elleçleme robotları ve fabrika taşıma sistemleri gibi endüstriyel otomasyon üniteleri) bir MEMS dönüölçer ile donatılmıştır. FOG dereceli MEMS dönüölçerler günümüzde geliştirme aşamasında olup, gelecekte optik dönüölçerlerin yerini alması beklenmektedir [54].

Tablo 2.3'te "ön yargı kararlılığını" ana performans değişkeni olarak kabul edip dönüölçer teknolojileri karşılaştırılmıştır.

Tablo 2.3. Dönüölçer teknolojisi karşılaştırma tablosu [54]

Kullanım Alanı	Kullanım Amacı	Önyargı Kararlılığı	Dönüölçer Teknolojisi
Üst düzey taktiksel	Füze navigasyonu	0.1-1°/h	RLG/FOG
Stratejik	Denizaltı navigasyonu	0.0001-0.01°/h	RLG/FOG
Navigasyon	Havacılık navigasyonu	0.01-0.1°/h	RLG/FOG
Taktiksel	Platform stabilizasyonu	0.1-30°/h	RLG/FOG
Tüketici	Hareket arayüzü	30-1000°/h	MEMS
Endüstriyel	Mühimmat ve roket güdüm	1-30°/h	MEMS

3. MATERYAL VE METOTLAR

3.1. Kullanılan Veri Seti

Bu tez kapsamında yapılan çalışmalarda kullanılan veri seti akıllı telefonun ivmeölçer ve jiroskop sensörleri yardımıyla elde edilmiştir. Akıllı telefonun sensörlerinden verileri okumak ve dışa aktarmak için MATLAB programının mobil uygulaması kullanılmıştır.

Veri seti oluşturulma esnasında bir otomobilin hızlanma verileri akıllı telefonun sensörleri yardımıyla elde edilmiştir. Veri seti oluşturulmaya başlamadan, uygun bir ölçüm ortamı oluşturulmuştur. Akıllı telefon, ölçüm için otomobilin sürücü koltuğu ile yan koltuk arasında bulunan konsola yerleştirilirken yuvarlanma(roll), yunuslama(pitch) ve dönme(yaw) açıları 0° olacak şekilde akıllı telefonun su terazisi uygulaması yardımıyla ayarlanmış ve akıllı telefon sabitlenerek uygun bir ölçüm ortamı oluşturulmuştur. Veri seti oluşturulurken eğimin çok değişken olmadığı şehirler arası bir yol kullanılmış ve yol uzunluğu 30,4km olarak ölçülmüş ve bunun yanı sıra virajlı ve eğimli yollarda da ölçümler alınarak farklı yol ve sürüş dinamikleri altında ölçümler gerçekleştirilmiştir.

Veri seti oluşturulurken dört farklı ölçüm senaryosu gerçekleştirilmiştir. Bu ölçüm senaryoları aşağıda açıklanmıştır.

Birinci ölçüm senaryosunda, sensörlerin sabit kayma hatasının tespiti için herhangi bir ivmelenme olmadan ölçümler alınmış ve sensörün aktif olduğu durumlarda var olan bu sabit hatanın etkilerini gidermek için veriler kaydedilmiştir.

İkinci ölçüm senaryosunda, otomobilin hız sabitleyici fonksiyonu kullanılarak sabit hız altındaki verileri GPS ile alınan hız verileri ile karşılaştırılarak sensör gürültüsü elde edilebilmesi açısından veriler kaydedilmiştir.

Üçüncü ölçüm senaryosunda, bu tez kapsamında hız tahmini yaparken kullanılmak için ivme verileri, değişken hızlar altında ölçümler yapılarak veriler kaydedilmiştir.

Son ölçüm durumu olan dördüncü ölçüm senaryosunda, bu tez kapsamında tasarlanan hız kestirim algoritmalarının üçüncü ölçüm senaryosundan daha sert değişen ivmelere nasıl tepki vereceğinin gözlenebilmesi açısından ivmede ani artışların ve azalışların olduğu bir sürüş ortamı oluşturulmuştur.

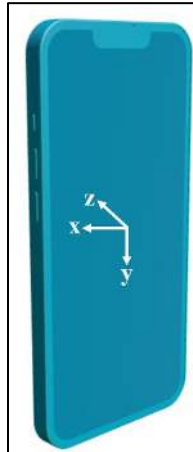
Veri seti, telefonun ivmeölçer ve dönüölçer sensörlerinden alınan zaman serisi sensör verileri içerir. Ayrıca, bu tez çalışmasında yapılan kestirim sonuçlarının karşılaştırılabilmesi için ölçüm esnasında 1 Hz (her 1 s’de bir) frekansında GPS verileri de kaydedilmiştir. Ölçümler esnasında, akıllı telefon olarak IOS 15.0 çalıştıran Apple Iphone 13 Pro

kullanılmıştır. Sensörlerden alınan veriler 20 Hz (her 50 ms’de bir) frekansında örneklenmiştir. Veri setine ait detaylar Tablo 3.1’de sunulmuştur.

Tablo 3.1. Kullanılan veri setine ait detaylar

Ölçümler sırasındaki aktivite sayısı	4
Ölçüm aşamasında kullanılan uygulama	MATLAB mobil
Ölçümlerin alındığı yol uzunluğu	30,4 km
Sensör örnekleme frekansı	20 Hz
Kullanılan akıllı telefon	Apple Iphone 13 Pro
Toplam ham veri ölçüm sayısı	176 274

Şekil 3.1’de sensör okumalarının yapıldığı akıllı telefonun temsili bir görünüşü verilmiş ve telefon üzerindeki sensör konfigürasyonu Şekil 3.1.’deki gibidir. Bu tez kapsamında doğrusal hız kestirimi yapıldığından dolayı veri seti oluşturma aşamasında, akıllı telefon otomobile ivme verileri -y ekseninden okunacak şekilde yerleştirilmiştir. Ölçümler sırasında düz, eğimsiz ve viraj olmayan bir yol tercih edildiğinden x ekseninde gürültü haricinde herhangi bir hız görülmemiş ve z ekseninde de gürültüyle karışık bir şekilde yer çekimi ivmesi gözlemlenmiştir.



Şekil 3.1. Akıllı telefonun sensör konfigürasyonu

Bu tez çalışmasında, oluşturulan veri setinden sadece akıllı telefonun ivmeölçer sensör verileri kullanılmış, otomobilin -y eksenlerindeki ölçümleri işleme alınmıştır. İvmeölçer sensörü için birim m/s^2 olarak MATLAB mobil program çıktısında belirtilmiştir. Çalışmalarda kullanılan veri seti bilgileri Tablo 3.2’de verilmiştir.

Tablo 3.2. Çalışmalarda kullanılan veri seti bilgileri

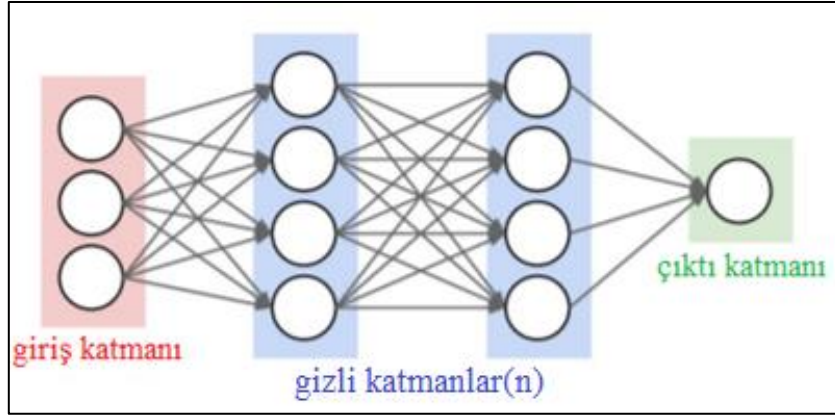
Sürüş Şekli	Eksen bilgisi	Ölçüm sayısı
1.Ölçüm senaryosu	-y	56 580
2.Ölçüm senaryosu	-y	6 534
3.Ölçüm senaryosu	-y	56 580
4.Ölçüm senaryosu	-y	56 580

3.2. Kullanılan Kestirim Algoritmaları

Literatürde benzer çalışmalara [70], [71] bakıldığında Kalman filtresinin kestirim amacıyla fazlaca kullanıldığı ve başarılı çıktılar verdiği gözlemlenmiştir. Ayrıca son yıllarda oldukça popüler hale gelen derin öğrenme algoritmalarının da Kalman filtresi gibi geleneksel yöntemlere bir alternatif olarak sunulduğu dikkat çekmektedir. Bu sebeplerle yapılan çalışmalarda geleneksel ve derin öğrenme tabanlı yöntemler hız kestirimi amacıyla karşılaştırmalı olarak irdelenmiştir.

Makine öğrenimi, insan beyninin çalışma mantığının bilgisayar bilimine uyarlanmış halidir. Makine öğrenimi, tam anlamıyla bir kodlama mantığına dayanmadan, veri kümelerinden öğrendiği bilgilerden çıkarım yaparak sınıflandırma ve tahmin işlemlerini yapabilmektedir. Örneğin, yüksek teknoloji içeren araçlarda otomatik park yardımı, şerit takip sistemleri gibi kullanım alanlarının yanı sıra günümüz akıllı telefonlarında sesli dikte özelliği sayesinde sesin yazıya çevrilmesi gibi farklı çalışma alanları da mevcuttur ve bu örneklerin çoğaltılması mümkündür.

Derin öğrenme, birden fazla işlem katmanı içeren ve bu katmanlarındaki verileri önceki katmanlardaki verilerden hesaplamak amacıyla yararlanılan bir çalışma alanıdır. Veri boyutlarının giderek büyümesi sonucu oluşan karmaşık yapıyı keşfetmek amacıyla geri yayılım algoritması kullanarak derin öğrenme ağının içerdiği parametrelerin nasıl değişmesi gerektiğini çözümler [72]. Klasik bir makine öğrenmesinden farkı çok fazla ara katman içermesidir. Şekil 3.2’de örnek teşkil etmesi açısından bir derin sinir ağının genel gösterimi verilmiştir. Kullanılan mimarilere göre gizli katmanların sayılarında ve sıralamalarında değişiklikler olur.



Şekil 3.2. Derin sinir ağı genel modeli [73]

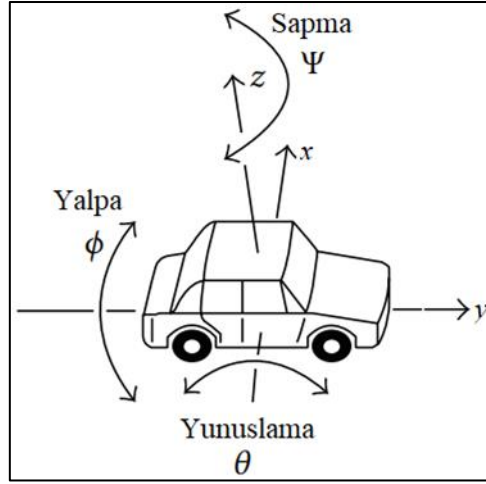
Eğitim verilerinin miktarının artması sonucunda derin öğrenme modelleri daha da kullanışlı hale gelmiş ve gelişen bilgisayar altyapısı sayesinde derin öğrenme modelleri zamanla büyüme göstermiştir. Böylece doğruluk derecesini artırarak zamanla karmaşıklaşan problemleri de çözüme ulaştırmıştır [74].

Kullanım alanı olarak da oldukça geniş bir çalışma alanına sahiptir. Bunlar; savunma teknolojileri, sağlık, görüntü işleme, sesli asistan, araçlardaki oto pilot uygulamalarında ve daha birçok alanda kullanımı mevcuttur.

3.2.1. Rotasyon matrisi

Akıllı telefonlar, sensörleri aracılığıyla elde ettiği verileri 3 eksenli bir koordinat eksenini kullanarak ifade eder. Akıllı telefon varsayılan duruş pozisyonundayken, cihazın ekranına göre eksenler tanımlanır. Bu tez kapsamında kullanılmak için oluşturulan veri setinde, ölçümlerin alındığı akıllı telefonun sensör eksenleri Bölüm 3.1’de kullanılan veri seti başlığı altında anlatılmıştır.

Veri seti oluşturulurken, ölçüm alınan yol her ne kadar düz ve eğimsiz olmasına dikkat edilse de doğrusal bir ivme ölçümü almak zordur. Veri seti için ölçümler alınırken, ölçüm alınan yoldaki bozukluklardan kaynaklı otomobil roll angle (yalpa açısı), pitch angle (yunuslama açısı) ve yaw angle (sapma açısı) etkisi altında kalmaktadır. Bu açılara maruz kalan otomobildeki akıllı telefon sensörleri, ivme verilerini ölçtüğünde yer çekimi ivmesinin ölçüldüğü z eksenini dışındaki eksenlerde de yer çekimi etkisi görülecektir. Bu durumda edinilen ivme verileri kullanılmadan önce diğer eksenlerdeki ölçümlerin yer çekimi etkisinden arındırılması gerekir. Bu işlem yapılmadığı durumda ivme verilerinden hız kestirimi hatalı sonuçlar verecektir. Örnek teşkil etmesi açısından Şekil 3.3’te açılarının araç üstündeki gösterimi verilmiştir.



Şekil 3.3. Otomobil üzerinde açıların gösterimi [75]

Akıllı telefon sensörleri yardımıyla elde edilen veri setinde, yerçekiminin ivmesinin etkisiyle bozulan x ve y eksenlerindeki ivme verileri rotasyon matrisleri yardımıyla düzeltilmiştir. Rotasyon matrisi eşitlik (3.1)'de R harfi ile gösterilmektedir [76].

$$R = \begin{bmatrix} E_x & E_y & E_z \\ N_x & N_y & N_z \\ G_x & G_y & G_z \end{bmatrix} \quad (3.1)$$

R matrisindeki x, y ve z akıllı telefonun eksenini göstermektedir. E ve N sırasıyla doğuyu ve kuzeyi gösteren birim vektörlerdir. G ise yerçekimi vektörüdür.

Rotasyon matrisinde kullanılan Euler açıları ψ , θ ve ϕ harfleri ile ifade edilir. Azimut açısı ψ harfi ile gösterilir ve G vektörü etrafında döndürmeyi ifade eder. Yunuslama açısı θ harfi ile gösterilir ve E vektörü etrafında döndürmeyi ifade eder. Son olarak yalpa açısı ϕ harfi ile gösterilir ve N vektörü etrafında döndürmeyi ifade eder.

Akıllı telefon sensörleri, ölçüm esnasında herhangi bir açıya maruz kaldığında ve yerçekiminden etkilendiğinde rotasyon matrisi yardımıyla ölçümler yerçekimi etkisinden arındırılır. Bunun için Azimut, yunuslama ve yalpa açıları için birer rotasyon matrisi mevcuttur. Bu matrisler sırasıyla eşitlik (3.2), (3.3) ve (3.4)'de verilmiştir.

$$R_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$R_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \quad (3.3)$$

$$R_\varphi = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (3.4)$$

Dönme açılarının Euler açıları ile arasındaki bağıntıdan eşitlik (3.5) ve (3.6) elde edilir.

$$R(\psi, \theta, \varphi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (3.5)$$

$$R(\psi, \theta, \varphi) = \begin{bmatrix} \cos \varphi \cos \psi - \sin \varphi \cos \theta \sin \psi & \cos \varphi \sin \psi + \sin \varphi \cos \theta \cos \psi & \sin \varphi \sin \theta \\ -\sin \varphi \cos \psi - \cos \varphi \cos \theta \sin \psi & -\sin \varphi \sin \psi + \cos \varphi \cos \theta \cos \psi & \cos \varphi \sin \theta \\ \sin \theta \sin \psi & -\sin \theta \cos \psi & \cos \theta \end{bmatrix} \quad (3.6)$$

Akıllı telefon sensörleriyle ölçülen verilerin telefon koordinat sisteminden dünya koordinat sistemine dönüşümü eşitlik (3.7)'de verilmiştir.

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = R(\psi, \theta, \varphi) [a_x \ a_y \ a_z]^T \quad (3.7)$$

Uygulanan son adımla beraber ivme verileri telefon koordinat sisteminden dünya koordinat sistemine dönüştürülmüş ve bu dönüştürülen ivme verisinin sadece z ekseninde yerçekimi ivmesi mevcuttur. Eşitlik (3.8)'e uygulanarak z eksen bileşeninden yerçekimi etkisi çıkartılır. Eşitlikte bulunan g ifadesi yerçekimini göstermektedir ve değeri 9,81m/s² dir.

$$A_{doğrusal} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} - g[0 \ 0 \ 1]^T \quad (3.8)$$

Son adımda elde edilen A doğrusal matrisi yerçekimi etkisinden arındırılmış ivme verisinden oluşmaktadır ve işlem yapabilmek için uygun hale getirilmiştir.

3.2.2. Luenberger gözlemcisi

Gözlemci; sensör çıktıları ve sistem uyarı sinyallerini, sistem ve sensör modelleriyle birleştiren matematiksel bir yapıdır. Ek olarak gözlemci tek başına sensör çıktısından daha çok önem arz eden geri besleme sinyalleri de sağlar [77]. Luenberger gözlemcisi ise ek ölçümler kullanarak gürültü içeren durum değişkenlerinin gerçek değerlerini tahmin edebilen bir yöntemdir. Çalışma şekli açısından gözlemci kazancının değişmemesi dışında

Kalman filtresiyle büyük benzerlikler içerir [78]. Luenberger gözlemcisi durum tahmini yapılabilmesi için sistemin gözlenebilir ve kontrol edilebilir olması gerekmektedir.

3.2.2.1 Gözlenebilirlik

Dinamik bir sistem eşitlik (3.9) ve (3.10) gibi gösterilmektedir.

$$\dot{x} = Ax + Bu \quad (3.9)$$

$$y = Cx + Du \quad (3.10)$$

Burada;

x = Durum vektörü,

$\dot{x}$ = Durum vektörünün zamana göre türevi,

y = Çıkış vektörü,

u = Giriş ya da kontrol vektörü,

A = Sistem matrisi,

B = Giriş matrisi,

C = Çıkış matrisi,

D = Geri besleme matrisi,

Bir dinamik sistemin gözlenebilir olabilmesi, durum ve çıktı denklemi arasındaki ilişkinin bir özelliğidir. Bu yüzden A ve C matrisleriyle bağıntılıdır [79].

Eşitlik (3.9) ve (3.10)'da açıklanan sistemin gözlenebilir olabilmesi için eşitlik (3.11)'deki $(n \times n)$ boyutlu V matrisinin rankının n 'ye eşit olması gerekir [80].

$$V = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (3.11)$$

3.2.2.2 Kontrol edilebilirlik

Kontrol edilebilirlik, dinamik kontrol sistemini rastgele bir başlangıç durumundan hedeflenen bir son duruma, izin verilen kontroller kümesi yardımıyla yönlendirilebilmesi olarak nitelendirilebilir [81].

($n \times n$) boyutlu S matrisinin (eşitlik (3.12)), aşamasının (rankının) n olması durumunda sistem kontrol edilebilirdir [80].

$$S = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (3.12)$$

3.2.2.3 Luenberger gözlemci algoritması

Luenberger gözlemcisini tercih edilen bir tasarım metodu haline getiren temel etken gözlemci tasarımının basit olmasıdır [79]. Doğrusal zamanla değişmeyen eşitlik (3.13) ve eşitlik (3.14)'de tanımlanan sistemi ele alalım.

$$\dot{x} = Ax(t) + Bu(t), \quad x(t_0) = x_0 \quad (3.13)$$

$$y(t) = Cx(t) \quad (3.14)$$

$t = t_0$ başlangıç zamanında $x(t_0) = x_0$ başlangıç koşuluna sahip bu sistem kontrol edilebilir ise, sistemi eşitlik (3.15)'deki giriş sinyali ile istenen durum değişkeni değerine götürmek mümkün olmaktadır.

$$u(t) = -Kx(t) \quad (3.15)$$

Burada K matrisi kutup yerleştirme yöntemi yardımıyla elde edilebilir. İlk olarak sisteme $\{p_1, p_2, p_3, \dots, p_n\}$ şeklinde n tane kutup atanır ve kutup değerleri eşitlik (3.16)'da ki gibi çözülür [82].

$$|\lambda I - (A - BK)| = (\lambda - p_1)(\lambda - p_2) \dots (\lambda - p_n) \quad (3.16)$$

Sistem durumu $x(t)$ 'nin aksine bazı durumlarda sadece $y(t)$ çıkışı ölçülebilmelidir. Böyle durumlarda eşitlik (3.15)'deki durum geri beslemesi olan $u(t)$ kullanılamaz. Bu problemi çözmek için $x(t)$ sistem durumu lineer Luenberger gözlemci yardımıyla tahmin edilir [82].

Eşitlik (3.17)'de $\hat{x}(t)$ durum tahmini, $\hat{y}(t)$ tahmin çıktısıdır (eşitlik (3.18)). L matrisi tasarımın anahtar bir ögesi olarak gözlemci kazancıdır ve tahmin hatası da eşitlik (3.19) ile tanımlanır.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)), \quad \hat{x}(0) = 0 \quad (3.17)$$

$$\hat{y}(t) = C\hat{x}(t) \quad (3.18)$$

$$\tilde{x}(t) = x(t) - \hat{x}(t) \quad (3.19)$$

Eşitlik (3.13) ve Eşitlik (3.17)'den eşitlik (3.20) elde edilir.

$$\dot{\tilde{x}} = A\tilde{x}(t) - LC\tilde{x}(t) = (A - LC)\tilde{x}(t), \quad \tilde{x}(0) = x_0 \quad (3.20)$$

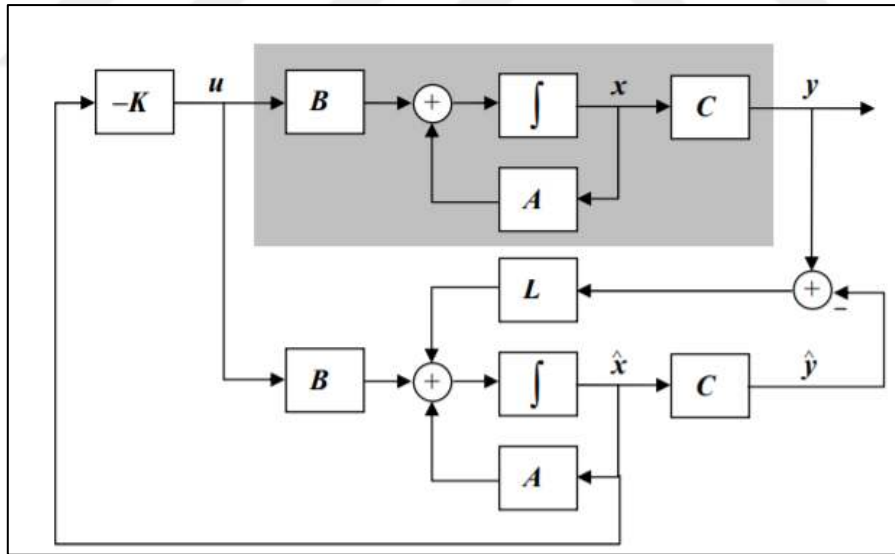
$(A - LC)$ 'nin öz değerlerinin tümünün karmaşık düzlemin sol yarısında olması durumunda tahmin hatası $t \rightarrow \infty$ sıfıra yaklaşacaktır (eşitlik (3.21) veya (3.22)).

$$\tilde{x}(t)|_{t \rightarrow \infty} = x(t) - \hat{x}(t)|_{t \rightarrow \infty} \rightarrow 0 \quad (3.21)$$

$$\hat{x}(t)|_{t \rightarrow \infty} \rightarrow x(t)|_{t \rightarrow \infty} \quad (3.22)$$

Zamanın ilerlemesiyle beraber durum tahmini, durumun gerçek değerine $x(t)$ 'ye doğru yaklaşır. Eşitlik (3.15)'deki geri beslemenin yerine eşitlik (3.23) konulabilir. Sistemin blok şeması Şekil 3.4'te verilmiştir [82].

$$u(t) = -K\hat{x}(t) \quad (3.23)$$



Şekil 3.4. Luenberger gözlemci blok şeması [82]

Son olarak da L matrisinin tasarımı için öncelikle $(A - LC)$ 'nin öz değerleri eşitlik (3.24) çözülerek bulunur. Denklemin çözümünde L matrisi bulunurken eşitlik (3.16)'nın çözümü yapılırken kullanılan kutup yerleştirme yöntemi kullanılır. Burada (A, C) matris çifti gözlenebilir olması koşuluyla $(A - LC)$ öz değerleri eşitlik (3.24) ile kolaylıkla bulunabilir [83].

$$|\lambda I - (A - LC)| = 0 \quad (3.24)$$

Sonuç olarak durum uzayı cinsinden Luenberger gözlemcisi eşitlik (3.25)'deki gibi elde edilir.

$$\hat{x}(t + 1) = (A - LC)\hat{x}(t) + Bu(t) + Ly(t) \quad (3.25)$$

3.2.3. Kalman filtresi

Belirli bir sürede gözlemlenen verilerden bazı değişkenlerin değerlerini tahmin etmek için kullanılan yöntem Kalman filtreleme denir [84]. Farklı varyasyonlarıyla tahmin problemlerinin analizinde ve sonuca ulaştırılmasında Kalman Filtresi temel bir yöntem olmuştur [85]. Temelinde Kalman filtresi, bazı koşullar altında tahmin edilen hata kovaryansını minimum yapan bir tahmin edici ve tahminleri düzeltici matematiksel formüller kümesidir. Tahmin yaparken geçmiş, şimdiki ve gelecek durum tahminlerini destekler. Sistem modelinin yapısı mutlak bir şekilde bilinmediği durumlarda dahi bunu yapabilmesi filtrenin güçlü yönlerindendir [86].

Özellikle otonom ve yardımcı seyrüsefer sistemleri alanında Kalman filtresi kapsamlı bir şekilde kullanılmıştır. Bunun en büyük etkenlerinden birisi de teknolojinin gelişmesiyle birlikte dijital hesaplama alanındaki ilerlemeler ve filtrenin görece olarak basit ve sağlam yapısından kaynaklanır. Ayrıca günümüzde daha farklı uygulama alanları da mevcuttur. Bir nesnenin konum ya da hız tahmininde, bir hedefin takibi için kullanılan radar uygulamalarında ya da hava durumu tahmini, konuşma geliştirme, ekonomi, otomatik pilot gibi geniş kullanım alanları mevcuttur [87].

3.2.3.1 Kalman filtresi algoritması

Kalman filtresi bir sistemin durum kestirimlerini yapan algoritmadır. Söz konusu sistemin durum kestirimi yapılabilmesi için doğrusal dinamik bir sistem olması gerekmektedir. Durum uzayı formatında doğrusal dinamik sistemin süreç modeli eşitlik (3.26)'da verilmiştir. Eşitlik (3.26)'da k zaman indisi, F durum geçiş matrisi, x_{k-1} önceki durum vektörü, B kontrol girdi matrisi, u_{k-1} kontrol vektörü, w_{k-1} süreç gürültü vektörü ve Q kovaryans matrisini göstermektedir. Sistemin durum değişkenlerinin sıfır ortalamaya sahip olduğu varsayılır (eşitlik (3.27)).

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.26)$$

$$w_k \sim (0, Q) \quad (3.27)$$

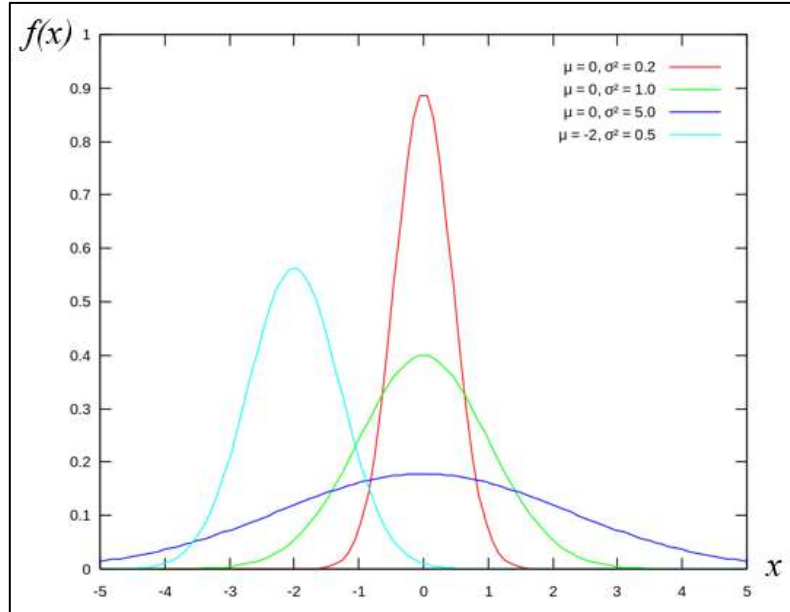
Eşitlik (3.28)'de ölçüm modeli verilmiştir. Burada z_k ölçüm vektörü, H ölçüm matrisi, v_k ölçüm gürültü vektörü, R kovaryans matrisidir. Ölçüm değişkenlerinin sıfır ortalamaya sahip olduğu varsayılır (eşitlik (3.29)).

$$z_k = Hx_k + v_k \quad (3.28)$$

$$v_k \sim (0, R) \quad (3.29)$$

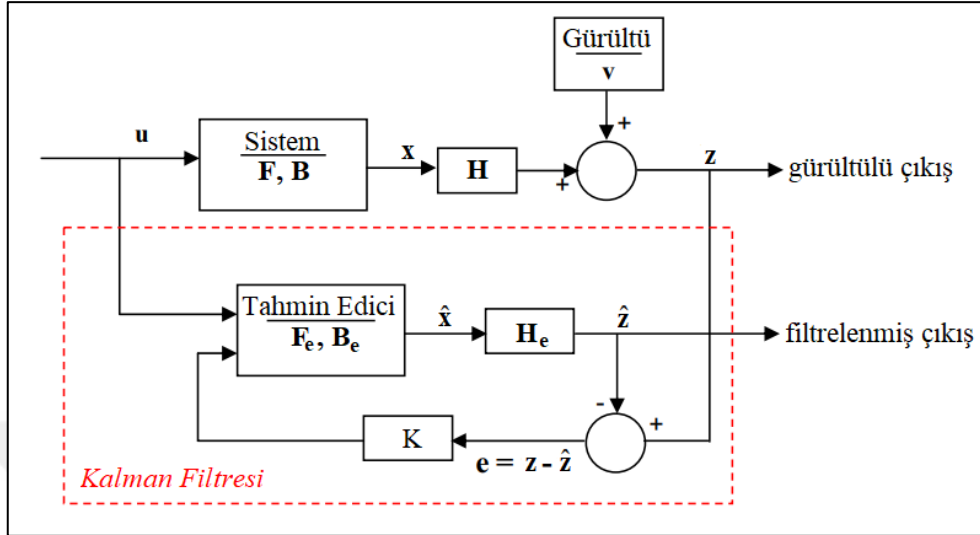
Sonuç olarak süreç gürültü vektörü ve ölçüm gürültü vektörünün kendi aralarında bağımsız ve normal dağılımlı olduğu kabul edilir. Bu normal dağılım, Gauss dağılımı (eşitlik (3.30)) olarak da bilinir. Eşitlikte (3.30)'da (σ) standart sapmayı, (μ) ortalamayı, (e) ve (π) ise matematiksel sabit sayıları gösterir. Birçok alanda tercih edilen sürekli ve olasılık dağılımlı bir matematiksel fonksiyondur. Şekil 3.5'te Gauss grafiği verilmiştir.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.30)$$



Şekil 3.5. Gauss olasılık yoğunluk grafiği

Kalman filtresinin amacı sensörlerden elde edilen z_k çıkış vektörü değerlerinden sistemin durum değişkeni olan x değerlerini tahmin etmektir. Şekil 3.6’da Kalman filtresi blok şeması yer almaktadır.



Şekil 3.6. Kalman filtre blok şeması [88]

Kalman filtresi algoritması durum tahmini yapmak için başlangıç parametrelerine ihtiyaç duyar. Bunlar ilk durum tahmini ($\hat{x}_0^+$) ve ilk hata kovaryans matrisi (P_0^+)dir.

Tahmin ve güncelleme adımı olmak üzere Kalman filtresi iki basamaktan oluşmaktadır. Kalman filtresi, durum tahmini yaparken geri besleme sinyali kullanır. Filtre öncelikle durum tahmini yapar ve daha sonra gürültülü sensör verilerinden geri bildirim alır. Bundan dolayı Kalman filtresi algoritması iki aşamadan oluşur [86].

Denklemlerde $\hat{\cdot}$ operatör kullanımı o değişkenin tahmini anlamına gelir. Yani $\hat{x}$ gösterimi x 'in tahminidir. Üst işaretlerde kullanılan $-$ ve $+$ sembolleri de sırasıyla önceki durumu ve sonraki durumu göstermektedir. Eşitlik (3.31) ve (3.32) de Kalman filtresi algoritmasının tahmin adımı verilmiştir. Eşitlik 3.32 tahmin edilen hata kovaryansı denklemdir.

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1} \quad (3.31)$$

$$P_k^- = FP_{k-1}^+F^T + Q \quad (3.32)$$

Kalman filtresi algoritmasının güncelleme adımı eşitlik (3.33-3.34-3.35-3.36) ile verilmiştir. Eşitlik (3.33) inovasyon olarak bilinen ölçüm kalıntısı denklemdir. Burada ölçüm kalıntısı gerçek ölçüm (z_k) ile ölçüm tahmini ($\hat{x}$) arasındaki farktır. Güncelleme adımında ilk adım olarak ölçüm kalıntısı ($\tilde{y}_k$) hesaplanır. Filtre mevcut ölçümü tahmin etmek için durum tahminini ölçüm matrisiyle çarpar. Durum tahmini ile ölçüm verilerine ağırlık atayan Kalman kazancı (K), eşitlik (3.34) ile hesaplanır. Sonrasında eşitlik (3.35)'te $\hat{x}_k^-$ tahminine $K\tilde{y}_k$ düzeltmesi sağlamak için ölçüm kalıntısı ile Kalman kazancı (K) çarpılır. Durum tahminini elde eden Kalman filtresi bir sonraki adıma geçebilir. Ardından eşitlik (3.36)'da güncellenmiş hata kovaryansı (P_k^+) bulunarak sonraki zaman adımında kullanılır [84].

$$\tilde{y}_k = z_k - H\hat{x}_k^- \quad (3.33)$$

$$K = P_k^- H^T (R + H P_k^- H^T)^{-1} \quad (3.34)$$

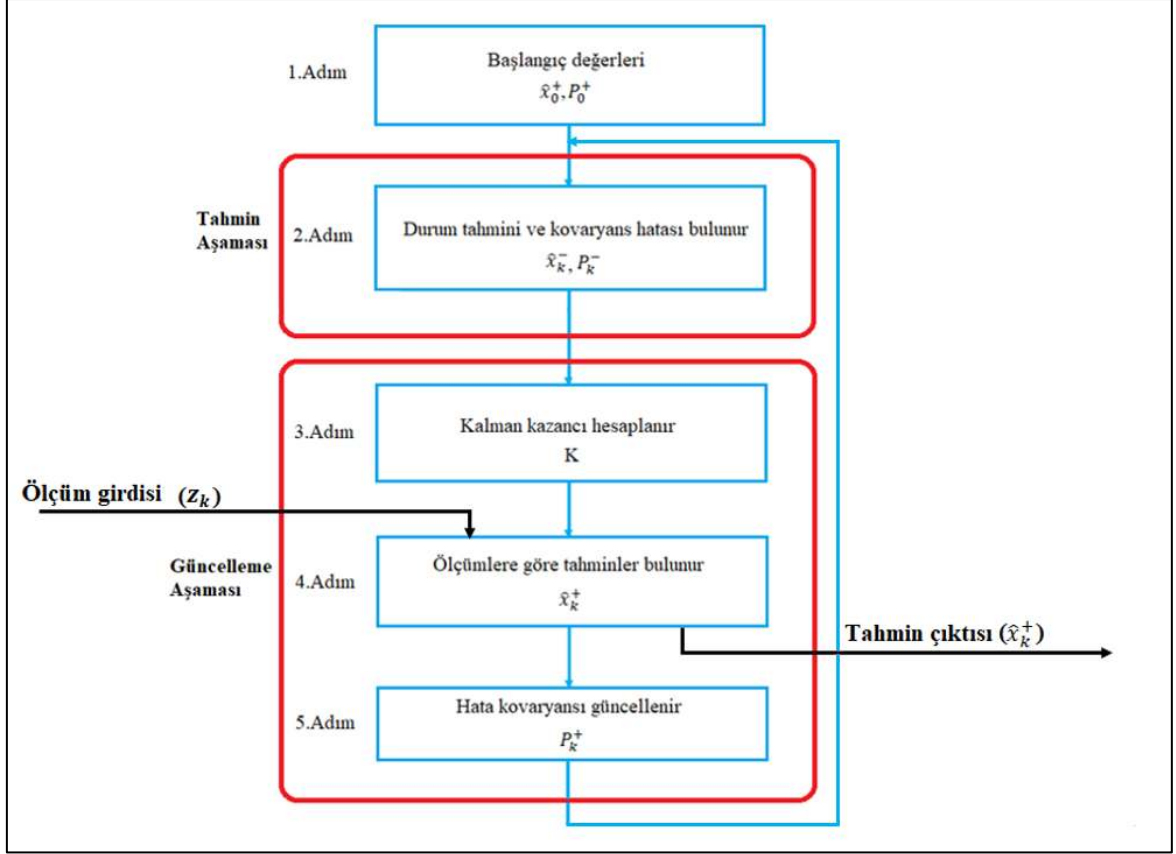
$$\hat{x}_k^+ = \hat{x}_k^- + K\tilde{y}_k \quad (3.35)$$

$$P_k^+ = (I - KH)P_k^- \quad (3.36)$$

Kalman filtresi algoritmasında ölçüm ve güncelleme denklemlerinde Q ve R değişkenleri mevcuttur. Burada Q işlem gürültüsü kovaryansı ve R ölçüm gürültüsü kovaryansıdır. Sistem modelinin ve ölçümlerin içerdiği hata oranlarına göre Q ve R matrisleri değişiklik göstermektedir. Modelle ilgili hatalar Q matrisi ile, ölçümle ilgili hatalar R matrisi ile alakalıdır. Sistem modeli ile ilgili sıkıntıların yüksek olduğu durumlarda Q matrisi R matrisinden büyük seçilir. Eğer ölçüm değerlerine güvenilmiyorsa R matrisi Q matrisinden daha yüksek seçilerek işlem yapılır. Filtrenin tahmin başarısına göre Q ve R matrisleri farklı değerlerle tekrar denenebilir. Bu matrislerin değerlerinin tam bilindiği durumlarda Kalman filtresi çok başarılı sonuçlar vermektedir. Gerçek dünya problemlerinde bu matrislerin değerlerinin hesaplanması olası değildir [89].

Kalman filtresi algoritma işleyişi Şekil 3.7'de verilmiştir. Şekil 3.7'de 1.adımda Kalman filtresi uygulanabilmesi için başlangıç değerleri olan başlangıç durum kestirimi ($\hat{x}_0^+$) ve başlangıç hata kovaryansı (P_0^+) işleme konulur. 2. Adımla beraber Kalman filtresi tahmin aşamasına geçilir ve durum tahmini ($\hat{x}_k^-$) ile hata kovaryansı (P_k^-) bulunur. Sıradaki adımda güncelleme aşamasına geçilir ve 3. Adımda Kalman kazancı (K) bulunur. Filtre kazancının bulunmasıyla beraber 4.adımda ölçüm girdisi (z_k) işleme alınır ve ölçümlere göre durum

tahmini ($\hat{x}_k^+$) bulunur. Durum tahmini elde edildikten sonra 5. Adımda hata kovaryansı güncellenir ve güncel hata kovaryansı (P_k^+) tahmin aşamasına dönerek 2. adımda işleme alınır ve bu şekilde filtreye geri besleme yapılmış olur. Kalman filtresi bu şekilde yinelemeli olarak çalışmaya devam eder.



Şekil 3.7. Kalman filtresi algoritma şeması

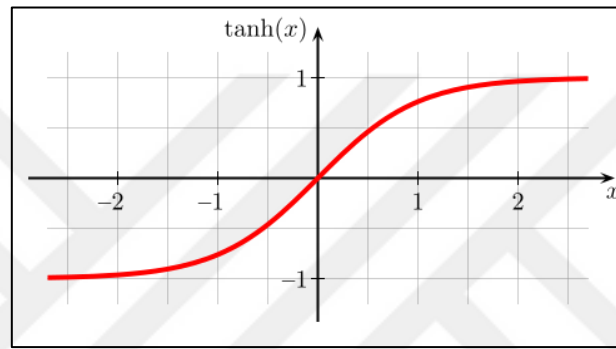
3.2.4. Uzun-kısa vadeli bellek (Long-short term memory, LSTM)

Sepp Hochreiter ve Jürgen Schmidhuber tarafından 1997 senesinde literatüre kazandırılan bir yinelemeli sinir ağı (Recurrent Neural Network, RNN) modelidir. Bu yapıların geleneksel ileri beslemeli sinir ağlarından farkı geri besleme yapılarına sahip olmasıdır. LSTM, ses işleme [90], görüntü işleme [91], dil işleme [92] gibi pek çok alanda uygulamalar mevcuttur.

LSTM'nin temel çıkış noktası olan RNN, yapılarında tek bir $\tanh$ katmanı bulundururken; LSTM birbirleriyle bağlantılı olan dört farklı katman içerir. $\tanh$ fonksiyonu, hiperbolik bir fonksiyondur.

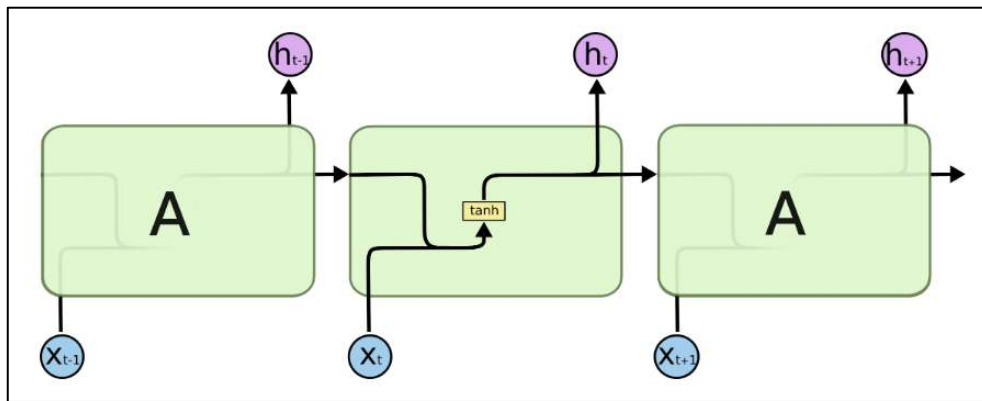
Trigonometride kullanılan sıradan fonksiyonların analoglarına hiperbolik fonksiyonlar denir. Trigonometride temel hiperbolik fonksiyonlar ise $\sinh$ ve $\cosh$ fonksiyonlarıdır. Bu iki fonksiyondan yararlanarak hiperbolik tanjant fonksiyonu $\tanh$ oluşturulur. Eşitlik (3.37)'de $\tanh$ fonksiyonunun formülü verilmiştir. Bir başka deyişle hiperbolik tanjant fonksiyonu, tanjant dairesel fonksiyonun hiperbolik analoguna denir. Şekil 3.8'de $\tanh$ fonksiyonu gösterilmiştir.

$$\tanh x = \left(\frac{\sinh x}{\cosh x} \right) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}} \right) = \left(\frac{e^{2x} - 1}{e^{2x} + 1} \right) \quad (3.37)$$



Şekil 3.8. $\tanh$ Fonksiyonu

Standart bir LSTM bloğu bir hücre, bir giriş kapısı (input gate), bir çıkış kapısı (output gate) ve unutma kapısından (forget gate) meydana gelir. RNN ve LSTM yapılarında ait bloklar sırasıyla Şekil 3.9 ve Şekil 3.10'da verilmiştir [93].

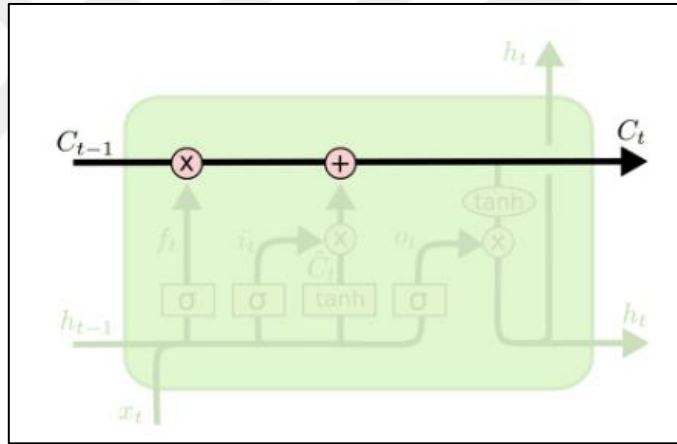


Şekil 3.9. Standart bir RNN bloğu [93]

Şekil 3.10'da verilen LSTM bloğunun en üstünden geçen yatay çizgi (Şekil 3.11) doğrusal işlemlerin yapıldığı hattır. Bu hatta taşınması gereken verilerin ne olduğu yapı

Şekil 3.10. Standart bir LSTM bloğu [93]

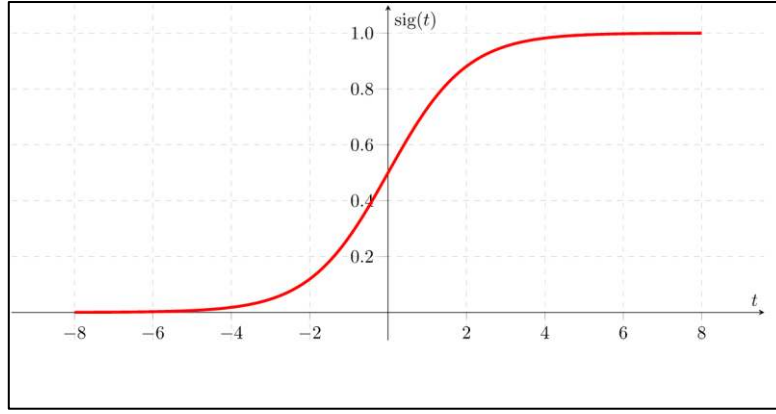
Şekil 3.10. Standart bir LSTM bloğu [93]



Şekil 3.11. LSTM bloğundaki doğrusal işlem hattı [94]

Sigmoid, sıfır ile bir arasında çıktı üreten “S” harfine benzeyen matematiksel bir fonksiyondur. Fonksiyona ait matematiksel model eşitlik (3.38) ile Sigmoid fonksiyon grafiği ise Şekil 3.12’de verilmektedir.

$$sig(t) = \frac{1}{1 + e^{-at}} \quad (3.38)$$

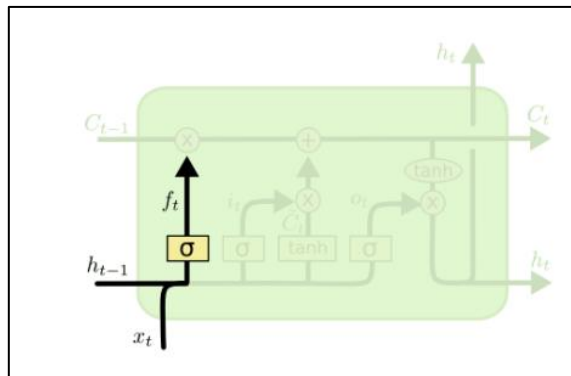


Şekil 3.12. Sigmoid fonksiyon grafiği

LSTM yapısı içerisindeki sigmoid fonksiyon çıktısı, sıfır olduğunda herhangi bir verinin geçişini durduran; bir olduğunda ise tüm verilerin geçişine izin veren bir katman olarak karşımıza çıkar.

LSTM bloğunda birincil görev, verinin bir sonraki katmana geçip geçmemesine karar vermektir. Bu kararın verildiği bölüm unutma kapısı (f_t) olarak isimlendirilir. Bir önceki katmandan gelen veri (h_{t-1}) ile mevcut veri (x_t) sigmoid fonksiyonuna (σ) girdi olarak verilerek elde edilen sonuca göre verilerin bir sonraki katmana taşınabilirliği kontrol edilir. Matematiksel olarak yapılan işlemler eşitlik (3.39)'da verilmiştir. Eşitlikte; W_f ağırlıkları, b_f ise kapıya ait önyargıyı gösterir. Şekil 3.13'te LSTM bloğunun ilk katmanı görülmektedir.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.39)$$



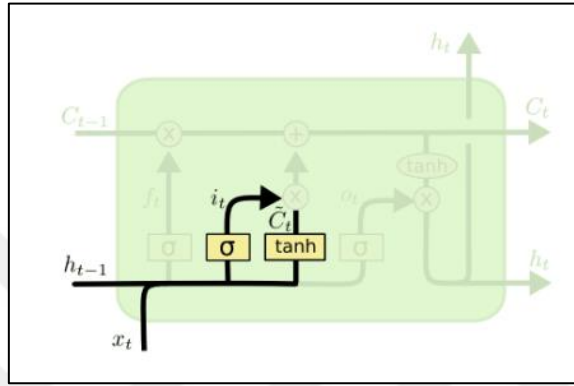
Şekil 3.13. LSTM bloğunun ilk katmanı [94]

LSTM bloğunda ikincil görev, gelen yeni verinin depolama durumuna karar vermektedir. Öncelikle giriş kapısı (i_t) olarak adlandırılan bu bölümde sigmoid fonksiyonu (σ) hangi değerlerin güncelleneceğine karar verir. Ardından $\tanh$ fonksiyonu, yeni aday

değerleri ($\tilde{C}_t$) barındıran bir vektör oluşturur ve sigmoid fonksiyon çıktısı (i_t) ile $\tanh$ fonksiyon çıktısı ($\tilde{C}_t$) çarpılarak (Şekil 3.14) eski hücre durumu olan C_{t-1} ile f_t çarpımına eklenir (Şekil 3.15). LSTM bloğunun ikinci katmanına ait matematiksel işlemler eşitlik (3.40) ve eşitlik (3.41)'de verilmiştir. Eşitliklerde, b_i ve b_c ilgili kapılara ait önyargılardır.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.40)$$

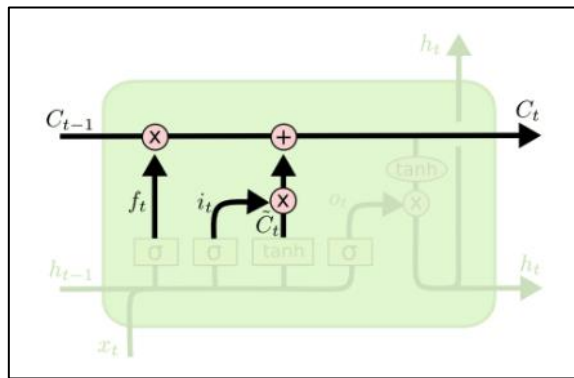
$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3.41)$$



Şekil 3.14. LSTM bloğunun ikinci katmanının ilk hali [94]

LSTM bloğunun ikinci katmanının sonraki hali için yapılan matematiksel işlemler eşitlik (3.42)'de verilmiştir. Eşitlikte, C_t ve $\tilde{C}_t$ sırasıyla hücre durumu ve aday hücre durumudur.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.42)$$



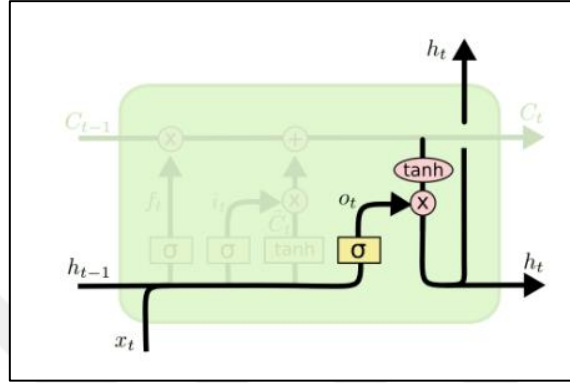
Şekil 3.15. LSTM bloğunun ikinci katmanının sonraki hali [94]

LSTM bloğundaki son görev ise çıkışın ne olacağına karar vermektir. Öncelikle yine bir sigmoid fonksiyonu, önceki veri ile mevcut veri girdilerini işler. Sonrasında hücre durumunda var olan veri, $\tanh$ fonksiyonunda işleme alınarak sigmoid çıktısı ile çarpılır.

Son aşamada ise filtrelenmiş bu çıktı bir sonraki hücre girişine (h_t) girdi olarak verilir. Şekil 3.16'da LSTM bloğunun son katmanı, çıkış kapısı (o_t) görülmektedir LSTM bloğunun son katmanına ait matematiksel işlemler eşitlik (3.43) ve eşitlik (3.44)'te verilmiştir.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3.43)$$

$$h_t = o_t * \tanh(Ct) \quad (3.44)$$



Şekil 3.16. LSTM bloğunun son katmanı [94]

LSTM yapısı, günümüz literatüründe çift yönlü uzun kısa süreli bellek (Bidirectional Long Short-Term Memory, BLSTM), geçitli tekrarlayan birim (Gated recurrent unit, GRU) gibi farklı versiyonlara sahiptir.

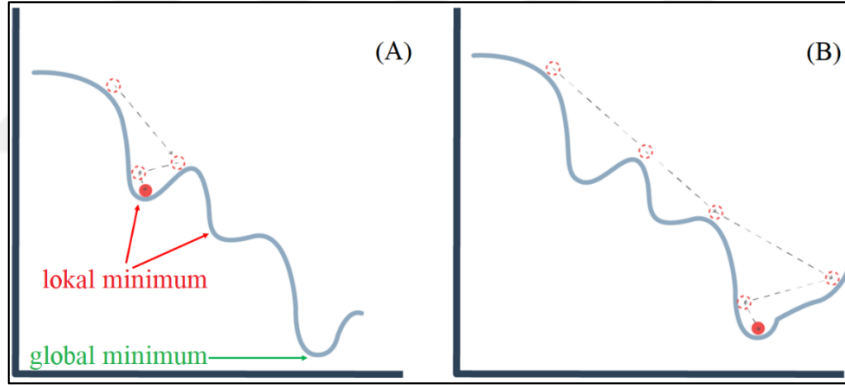
Temel anlamda derin öğrenme ağlarında öğrenme işlemi bir optimizasyon problemidir. Doğrusal olmayan bir problem de optimum noktanın bulunması istendiğinde çözüm için optimizasyon yöntemleri kullanılır. Derin öğrenme ağlarında ise en yaygın olarak kullanılan optimizasyon araçları aşağıdaki gibidir;

- 1) Kök Ortalama Kare Yayılımı (Root Mean Square Propagation, RMSprop)
- 2) Stokastik Gradyan İnişi (Stochastic Gradient Descent)
- 3) Uyarlanabilir moment tahmini (Adaptive moment estimation, Adam)
- 4) Adagrad
- 5) Nesterov accelerated gradient:
- 6) Adadelata
- 7) Adamax

Bu algoritmalar, başarı ve hız bakımından farklılıklar oluşturmaktadır [95]. Bu tez kapsamında; RMSprop, Stokastik Gradyan İnişin momentumlu versiyonu ve Adam optimize edicileri kullanılmıştır. Optimize ediciler hakkında tanımlamalar yapılmadan önce öğrenme oranı, öğrenme hızı ve momentum terimleri bilinmelidir.

Öğrenme hızı, her güncelleme adımında gradyanın boyunu belirler. Bu parametre 0 ile 1 arasında değişir ve gradyanın ne kadar güncelleneceği hakkında ayarlamalar yapar. Öğrenme hızının seçilebilecek en büyük değeri olan 1 değeri gradyanın %100 olarak güncelleneceğini belirtir. Ancak çoğu zaman büyük değerler seçilmez.

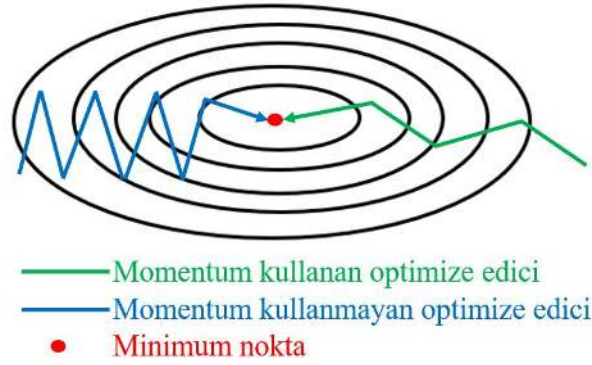
Şekil 3.17’de bir topu tepeden aşağı serbest bıraktığımızda eğimden dolayı giderek artan bir hızla top yuvarlanacaktır. Hızlanarak giden bu topun bir sonraki adımda yerinin hesaplanması istendiği zaman, o sonraki adımla bir önceki adım arası uzunluğa öğrenme oranı denir [96]. Bir sinir ağının eğitimi sırasında öğrenme oranının büyüklüğü önemlidir. Öğrenme oranının küçük seçilmesi durumunda top küçük adımlarla ilerler, ancak ilerleme yavaştır. Bu durumda top lokal bir minimum noktaya takılıp küresel minimum noktasını kaçırabilir. Tam tersi durumda öğrenme oranı çok büyük seçilmesi halinde ise global minimum hiçbir zaman yakalanamayabilir [96]. Şekil 3.17’de görüldüğü gibi (A) durumunda küçük öğrenme adımı seçildiğinde lokal minimuma takılmış ve global minimum kaçırılmıştır. Ancak (B) durumunda daha uygun bir öğrenme oranı seçildiğinde başarılı bir şekilde global minimum bulunmuştur.



Şekil 3.17. Öğrenme oranı seçimine göre sonuç karşılaştırması şekil [96]

Momentum, minimum noktaya ulaşılırken algoritmayı hızlandırmak için salınımı azaltan bir yöntemdir. Gradyanlar aynı yönü gösterdiğinde momentum artar ve gradyanların yön değiştirdiği noktalarda güncellemelerde azalmalar olur. Böylece hedefe yaklaşma sırasında geçen süreyi kısaltırken salınımlar da sönümlenmiş olur. Öğrenme hızı konusunda verilen top örneğine bakıldığında, yokuş aşağı bırakılan top, eğimden dolayı topun hızı sürekli artacaktır ve bu durumda momentum parametresi kullanılmazsa top her ayırık güncelleme adımında nerede olduğu hakkında herhangi bir bilgiye sahip olmayacaktır. Böylece momentum kullanılmadan geçmişe dair hiçbir bilgi olmadan sadece o güncelleme adımına ait gradyanı temel alınacaktır [96]. Şekil 3.18’de momentumun ilgili noktaya

ulaşmadaki etkisi gösterilmiştir. Şekil 3.18’de de görüldüğü üzere dikey anlamda salınımı azaltarak hedef noktaya (minimum nokta) daha hızlı ulaşılmasını sağlamıştır.

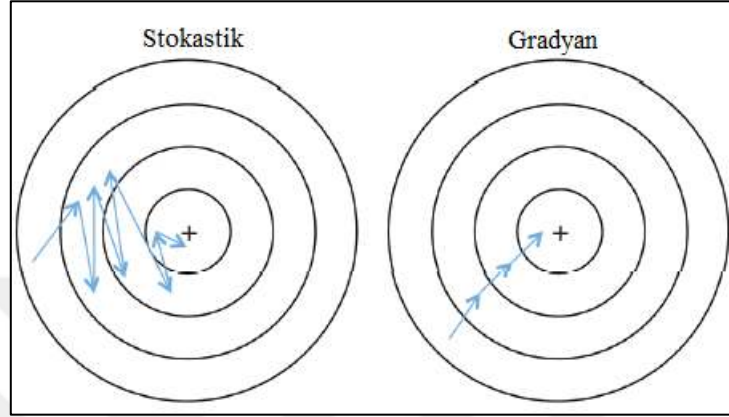


Şekil 3.18. Momentumun optimize edicilerdeki etkisi

RMSprop yöntemi; Geoffrey Hinton tarafından önerilmiştir. Gradyan tabanlı bir optimize edici olan bu teknik, sinir ağlarının eğitiminde kullanılır. Veri fonksiyonların da yayılım gösteren fonksiyon gradyanları, yayılım aşamasında bazı sıkıntılarla karşılaşma eğilimindedir. Bu aşamada gradyanların çok büyümesi ile gradyan patlaması ya da çok küçülmesi sonucu gradyan kaybolması olabilir. Bu da eğitim verilerinin öğrenilememesine sebep olur. Gradyan da oluşabilecek bu tür problemlerin üstesinden gelmek için RMSprop, kare gradyanların hareketli ortalamasını alır. Bu işlem sonucunda adım boyutu yani momentum düzenlenir. Böylece patlayan gradyanların önüne geçmek için adım uzunluğu azaltılırken, kaybolan gradyanlar için adım uzunluğu artırılır. RMSprop, kısacası öğrenme oranını bir hiper parametreymiş gibi seçmek yerine öğrenme hızını zamana göre değiştirerek kullanır [97].

Stokastik Gradyan İnişi (Stochastic Gradient Descent, SGD); kelime olarak stokastik, rastlantısal anlamına gelmektedir. Stokastik gradyan inişi ise her yineleme adımında tüm veri setini kullanmak yerine rastgele seçilecek birkaç örnek ile işlemler yapar. Gradyan inişin stokastik versiyonu bazı problemlere çözüm getirmek amacıyla türetilmiştir. Gradyan inişte gradyan hesabı yapılırken, her yineleme adımında veri kümesinin sahip olduğu toplam örnek sayısı kullanılır ve bu örnek sayısı yüksek olduğu durumlarda çözüme ulaşmak oldukça uzun sürebilir. Bu problemin üstesinden gelmek amacıyla gradyan inişin stokastik versiyonu ortaya atılmıştır. SGD’nin, her yineleme aşamasında veri kümesi karıştırılır ve içerisinden rastgele bir numune seçilerek yineleme işlemi gerçekleştirilir. Burada SGD optimize edici, örnek seçimini rastgele yaptığından dolayı minimum noktaya gidiş yolunda algoritmanın izlediği yol, ağırlık ayarlaması yaparken belirli düzeyde gürültü içerir ve bu SGD’yi gradyan

inişin tipik bir versiyonuna göre daha gürültülü bir yol yapar. Ancak daha kısa bir eğitim süresi elde edileceğinden dolayı yolun gürültülü olması fazla bir önem teşkil etmez ve etkili bir yöntem olduğundan dolayı yaygın olarak kullanılır. Bunun yanı sıra SGD yöntemi olasılıksal olmasından dolayı minimum noktaya ulaşmak için daha fazla yinelemeye ihtiyaç duysa da tipik bir gradyan inişe göre hesaplamalar daha az maliyetlidir [98]. SGD yönteminin gradyan iniş algoritmasıyla karşılaştırması Şekil 3.19’da verilmiştir.



Şekil 3.19. Stokastik gradyan iniş ile gradyan iniş algoritmasının karşılaştırması [99]

SGD optimize edicinin momentum kullanan versiyonuna momentumlu Stokastik Gradyan İniş (Stochastic Gradient Descent with Momentum, SGDM) denir. SGDM, daha hızlı yakınsama sağlamak amacıyla gradyan vektörlerini doğru yönlerde hızlandırmaya yardımcı olan optimize edicidir. Derin öğrenme ağlarının eğitiminde kullanılır. SGDM optimize edici, SGD optimize ediciye göre genelde daha iyi sonuçlar verir. SGDM optimize edicinin daha iyi sonuçlar vermesinin ana sebeplerinden birisi, üstel ağırlıklı ortalama kullanarak tahminlerin gürültüden daha az etkilenmesini sağlamasıdır [100].

Uyarlanabilir Moment Tahmini (Adaptive Moment Estimation, Adam): RMSprop ve SGD optimize edicilerin momentumlu bir kombinasyonudur. RMSprop optimize edici gibi kare gradyanları kullanarak öğrenme oranını ölçekler. Momentumdan yararlanmak için SGD optimize edicinin de yaptığı gibi gradyanın kendisinden yararlanmak yerine gradyanın hareketli ortalamasıyla işlem yapar. Adam, öğrenme oranını uyarlayarak işleme koyan bir yöntemdir ve her bir parametrenin öğrenme oranlarını hesaplar. Temelde adı uyarlanabilir moment tahmininden türetilmiştir. Çünkü birinci (ortalama) ve ikinci gradyan anlarının (merkezlenmemiş varyans) tahminleriyle sinir ağında mevcut her bir ağırlığa öğrenme oranı uyarlar. Ayrıca Adam optimize edici, veri sayısının büyük olduğu durumlarda oldukça verimlidir ve daha az bellek ihtiyacı duyar ve hızlıca iyi sonuçlar elde ettiğinden derin öğrenme alanında oldukça popülerdir [101].

Bir derin öğrenme ağı mimarisinde test sonuçlarının doğruluğunu etkileyen parametreler mevcuttur. Bu tez kapsamında derin öğrenme ağlarının performansına yönelik 17 adet parametre kullanılmıştır. Bunlar:

- a) Maksimum eğitim turu (Max. epoch)
- b) Gradyan eşiği (Gradient threshold)
- c) Başlangıç öğrenme oranı (Initial learn rate)
- d) Öğrenme oranı çizelgesi (Learn rate schedule)
- e) Öğrenme adım büyüklüğü (Learn rate drop period)
- f) Öğrenme adım düşme faktörü (Learn rate drop factor)
- g) Gizli birimlerin sayısı (Number of hidden units)
- h) Mini parti boyutu (Mini batch size)
- i) Karıştırma (Shuffle)
- j) L2 Düzenleyici (L2 regularization)
- k) Gradyan bozunma faktörü (Gradient decay factor)
- l) Kare gradyan bozunma faktörü (Squared gradient decay factor)
- m) Doğrulama sıklığı (Validation frequency)
- n) Gradyan eşiği yöntemi (Gradient threshold method)
- o) Momentum
- p) Sıra uzunluğu (Sequence length)
- r) Sıra doldurma değeri (Sequence padding value)

Bu parametrelerin açıklamaları aşağıda verilmiştir.

a) Maksimum eğitim turu (Max. epoch): Değer olarak pozitif tamsayılardan seçilen ve eğitim sırasında verilerin derin öğrenme ağına gösterilme sayısı olarak tanımlanır [102].

b) Gradyan eşiği (Gradient threshold): Kademeli olarak artan gradyan değeri eğitimi kararsız hale getirir ve buna gradyan patlaması (gradient exceeded) denir. Ağın eğitimi alakasız değerlerden arındırmak ve öğrenme oranlarını artırarak gradyan patlamasını engellemek amacıyla gradyan kırpma (gradient clipping) işlemi yapılır. Bu işlem, pozitif skaler ya da sonsuz sayılardan seçilebilen gradyan eşiği sayısı ile kontrol edilir. Gradyan değeri, bu eşik değerini geçmesi durumunda gradyan eşik değerine göre kırpma işlemi yapılır [103].

c) Başlangıç öğrenme oranı (Initial learn rate): Derin öğrenme ağının eğitimi için kullanılan bu parametre pozitif skaler sayılardan seçilir. Bu sayının çok küçük olması durumunda ağın eğitimi için geçen süre uzun olurken; büyük olması durumunda ise ağın eğitimi yetersiz olur [102].

d) Öğrenme oranı çizelgesi (Learn rate schedule): Öğrenme hızının değişimini belirleyen bir parametre olarak tanımlanmıştır. Mevcut yöntemler, yok (none) ve parça parça (piecewise) olarak belirtilmiştir. Öğrenme oran çizelgesi “yok” şeklinde seçilirse, eğitim boyunca öğrenme hızında herhangi bir değişim olmaz. Bu yapının “parça parça” seçilmesi durumunda, öğrenme hızı eğitim boyunca kademeli bir şekilde azalır. Bunun yanı sıra “parça parça” durumunda iki parametre daha mevcuttur. Bunlar, düşme süresi (drop period) ve düşme oranı faktörü (drop rate factor)’dür. Düşme süresi ve düşme oranı faktörü sırasıyla, eğitim süresi boyunca öğrenme oranının düştüğü çarpım faktörü ve eğitim gerçekleşirken öğrenme oranındaki değişimler sırasında geçen devir sayısı olarak tanımlanır [102]. Bu parametrelere ait detaylı açıklamalar aşağıda verilmiştir.

e) Öğrenme adım büyüklüğü (Learn rate drop period): Pozitif tam sayılardan seçilen bu parametre, belirtilen eğitim tur sayısı her geçildiğinde genel öğrenme oranı düşme faktörü ile çarpılır [103].

f) Öğrenme oranı düşme faktörü (Learn rate drop factor): Bu parametre 0 ile 1 arasında bir değer alır ve skalerdir. Belirli bir sayıda eğitim turu geçildiğinde öğrenme oranına uygulanan bu parametre çarpımsal bir faktördür [103].

g) Gizli birimlerin sayısı (Number of hidden units): Değeri pozitif tam sayılardan seçilir başka bir deyişle gizli boyut olarak da adlandırılır. Zaman serisi verilerinin gizli durumları arasındaki hatırlanan bilgi oranına gizli birim sayısı denir. Gizli durum, dizi uzunluğuna bakılmaksızın zaman adımlarının daha önceki durumlarından da bilgiler barındırabilir. Gizli birimlerin sayısının çok büyük seçilmesi durumunda kullanılan derin öğrenme ağı eğitim veri seti üzerinde gereğinden fazla çalışarak ezber yapmaya başlar ve bu da aşırı uyuma sebep olur. Bundan dolayı test verisiyle model denendiğinde büyük ihtimalle düşük başarı elde edilir. Genelde gizli birimlerin sayısı onlar ile binler arasında değişen bir sayıdır [104].

h) Mini parti boyutu (Mini batch size): Bu parametrenin değeri pozitif tam sayılar arasından belleğe sığabilmesi için 2’nin katları olacak şekilde seçilir ve eğitim kümesinin bir alt kümesidir. Veriler eğitileceği zaman her yineleme adımı için aynı anda kaç adet verinin kullanılacağını belirler, ağırlıkları günceller ve kayıp fonksiyonunun gradyan değerlendirmesini yapan bir parametredir [102].

i) Karıştırma (Shuffle): Verileri karıştırmak için kullanılmaktadır. Bu parametre ‘once’ olarak belirlenirse, veriler eğitim aşamasına geçmeden önce doğrulama ve eğitim için kullanılacak veriler bir kez karıştırılır. Eğer bu parametre ‘never’ olarak belirtilirse veriler hiç karıştırılmadan devam edilir. Son olarak bu parametrenin ‘every-epoch’ olarak

belirlenmesi halinde her bir eğitim döneminde eğitimde kullanılacak veriler karıştırılır ve bunun yanı sıra her ağ doğrulaması yapılacağı zaman doğrulama işleminde kullanılacak veriler de karıştırılır. Ayrıca eğitim aşamasında kullanılacak örneklerin sayısını mini parti boyutu tam olarak bölmediği zaman, eğitim ağının her turunda artan veriler atılır. Bu atılma aşamasında her zaman aynı verilerin atılmaması için karıştırma parametresi 'every epoch' olarak belirlenmelidir [102].

j) L2 Düzenleyici (L2 regularization): Negatif olmayan sayılar arasından bir değer alır. Derin öğrenme ağında ezberlemeyi engellemek amacıyla kayıp işlevinin ağırlıklarına bir düzenleme parametresi ekler. Başka bir deyişle ağırlık azaltması olarak da bilinir [102].

k) Gradyan bozunma faktörü (Gradient decay factor): Adam optimize ediciler için kullanılan ve değeri 0 ile 1 arasında virgülle ayrılan sayılardan seçilen bir parametredir [102].

l) Kare gradyan bozunma faktörü (Squared gradient decay factor): Bu parametre 0 ile 1 arasında bir değer alır. Rmsprop optimize edici için varsayılan değer 0,9 iken Adam optimize edici için bu değer 0.999 olarak belirtilmiştir [102].

m) Doğrulama sıklığı (Validation frequency): Pozitif tam sayılar arasından bir değer alır. Doğrulama ölçütleri arasındaki tekrarlama sayısıdır [102].

n) Gradyan eşiği yöntemi (Gradient threshold method): Gradyan eşiğini aşan gradyanların kırılması için kullanılan bu parametre üç farklı şekilde tanımlanır. Bu parametrenin 'l2norm' olarak seçilmesi durumunda öğrenilebilir bir parametrenin değeri L2'nin değerinden büyük olması halinde gradyanın değerinin L2'nin değerine eşit olacak şekilde kırpar. Eğer bu parametre 'global-l2norm' olarak seçilir ise global L2'nin değeri L değerinden büyük olması halinde tüm gradyanların değerlerini gradyan eşiği yöntemini kullanarak büyüklüğünü ayarlar. Bu ayarlama işleminde global L2 öğrenilebilir parametrelerin hepsini göz önünde bulundurarak yapar. Son olarak gradyan eşiği yöntemi 'absolute-value' olarak seçilir ve bu durumda öğrenilebilir parametrelerin her birinin kısmi türevleri alınır ve mutlak değerleri bulunur. Bu değerlerden herhangi birisi gradyan eşiğini aşması durumunda kısmi türevin işareti korunarak gradyan eşiğine eşit olacak şekilde ölçeklendirilir [102].

o) Momentum: Bu parametre değer olarak 0 ile 1 arasında bir sayıdır. Bu değer 0 olarak seçilmesi halinde momentumlu gradyan inişin önceki tekrarlama adımından gelen bir parametre güncellemesinin mevcut duruma katkısının olmadığı anlamına gelir. Eğer bu sayı

1 olarak seçilseydi önceki adımdan gelen güncellemenin mevcut duruma katkısının maksimum olduğu anlaşılır [102].

p) Sıra uzunluğu (Sequence length): Giriş dizilerini ayarlayan bir parametredir. Bu parametre üç farklı şekilde seçilebilir. Bunlar, en uzun (longest), en kısa (shortest) ve direk olarak bir pozitif tam sayı atamasıdır. Parametrenin “longest” seçilmesi durumunda; her mini partide yer alan diziler, en uzun veri ile aynı uzunluğa gelmesi için dizilere dolgu eklenir. Bu işlem sırasında herhangi bir veri atılmaz. Ancak dolgu işlemi sırasında ağdaki gürültü artabilir. Sıra uzunluğu parametresi “shortest” olarak seçilirse mini partide yer alan dizilerin uzunluğunu en kısa dizi uzunluğuna eşitlemek için verilerin bir kısmını atar. Eğer parametreye direk tam sayı ataması yapılırsa dizileri belirtilen uzunlukta daha küçük dizilere böler [102].

r) Sıra doldurma değeri (Sequence padding value): Sıra uzunluğu “longest” seçildiğinde veya bir tam sayı atandığında kullanılabilen bir parametredir. Skaler bir değer seçilir. Giriş dizilerine uygulanacak dolgu değeridir [102].

Tez kapsamında tasarlanan derin öğrenme ağı algoritmasında dört adet katman bulunmaktadır. Bu katmanların görevleri aşağıda ayrıntılı bir şekilde anlatılmıştır.

Sıra giriş katmanı (sequence input layer), bir ağa dizi verilerinin girilmesini sağlayan katmandır. Bu katman bir giriş katmanı oluşturarak, girişte kullanılan verilerin özellik sayısını ve giriş katmanının boyutu ayarlanır [105].

LSTM katmanı (LSTM layer), zaman serisinden oluşan verilerin zaman adımları arasındaki bağımlılıkları öğrenir. Aynı zamanda eğitim aşamasında uzun diziler boyunca gerçekleşen gradyan akışını daha iyi hale getirmek için ek işlemler uygular. LSTM katmanı çıkış durumu olarak da bilinen gizli durum ve hücre durumundan meydana gelir. Bir t adımında bulunan gizli durum, aynı zaman adımında bu katmanın çıktısını da içerir. Her zaman adımında öğrenilen bilgiler hücre durumunun içinde bulunur. Geçen her bir zaman adımında LSTM katmanı hücre durumuna bilgi ekleme ve çıkarma işlemi yapar. Katman hücre durumundaki bu güncellemeleri kapıları kullanarak yapar. Tez çalışmalarında kullanılmak için oluşturulan derin öğrenme algoritmasında bu LSTM katmanı ile gizli birimlerin sayısı ayarlanmıştır [104].

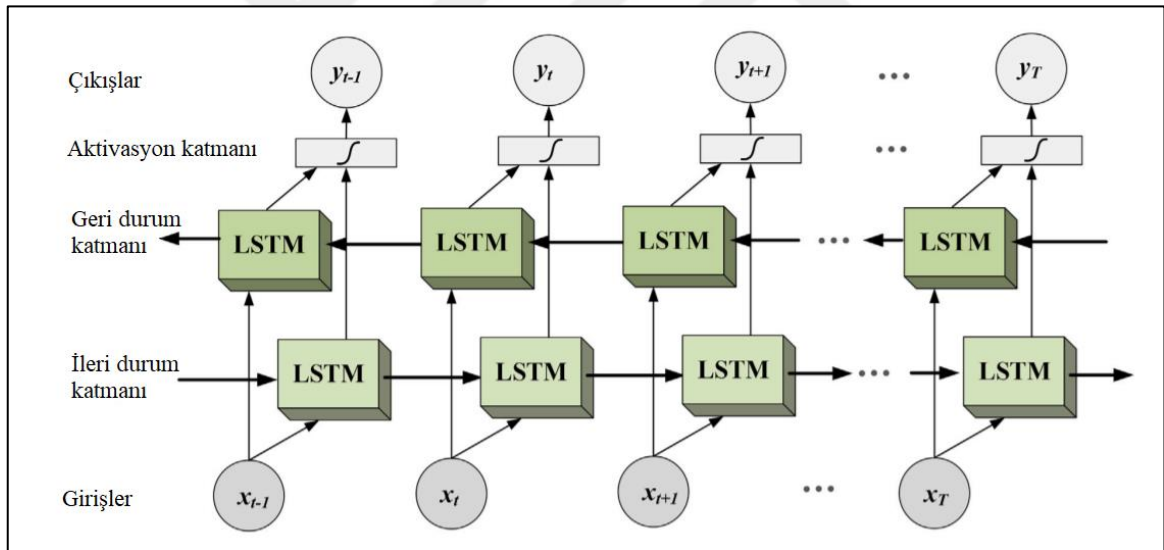
Tam bağlaşımlı katman (fully connected layer), ilk olarak bir ağırlık matrisi kullanarak girişi bu matrisle çarpıp ve çarpma işleminin ardından bir sapma vektörü ekler. Bu katmanın

adından da anlaşılacağı üzere önceki katmanlarda bulunan nöronlar, tam bağlaşımlı katmanda bulunan nöronlarla bağlanır [106].

Regresyon katmanı (Regression layer), regresyon analizi gerçekleştirmek amacıyla yarım ortalama kare hata kaybını hesaplayarak ağı çıktısını üretir.

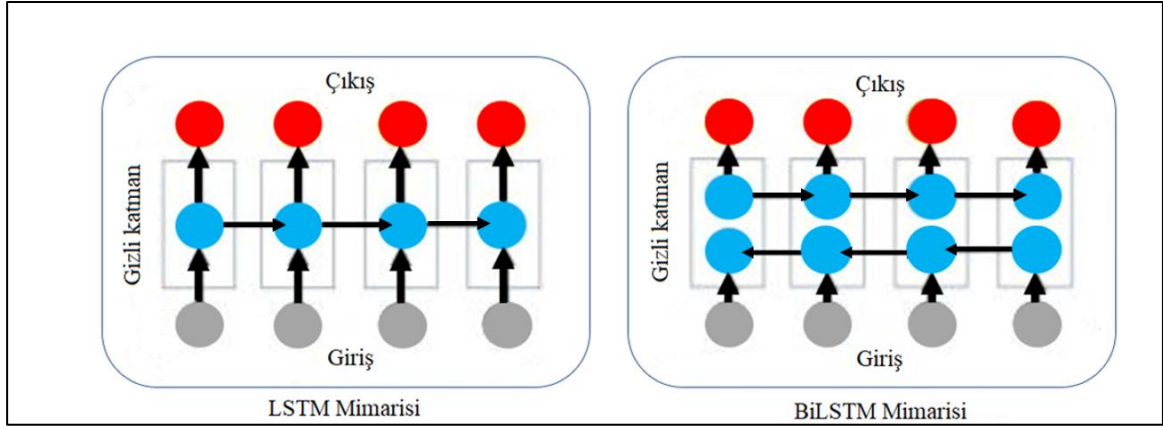
3.2.5. Çift yönlü uzun kısa süreli bellek (Bidirectional long short-term memory, BLSTM)

1997 yılında Schuster ve Paliwall tarafından ortaya atılan bir geri beslemeli sinir ağı çeşididir. Bu sinir ağı yapıları hem bir sonraki katmandan hem de bir önceki katmandan gelen verileri işlemeye olanak sağlar. Yani geleneksel bir RNN yapısından farkı iki katmandan gelen değerlere göre hesaplama yapılmasıdır. Katmanlardan biri girişini ileri yönden alırken diğeri geri yönden alır. Bu şekilde LSTM'nin iki kez uygulanması, öğrenme işleminin uzun vadeli verilerde daha iyi olmasını ve dolayısıyla da model doğruluğunun artırılmasına vesile olur [107]. Şekil 3.20'de BLSTM genel yapısı verilmiştir.



Şekil 3.20. BLSTM yapısı [108]

LSTM verileri sıralı bir şekilde tek yönde işler. Bunun yanı sıra BLSTM sinir ağında bilgi akışı her iki yönde de gerçekleşir. Bu yapay sinir ağında, yönleri birbirine ters olacak şekilde iki LSTM ağı ile eğitim gerçekleştirilir [109]. LSTM ve BLSTM arasındaki yapısal fark Şekil 3.21'de gösterilmiştir.



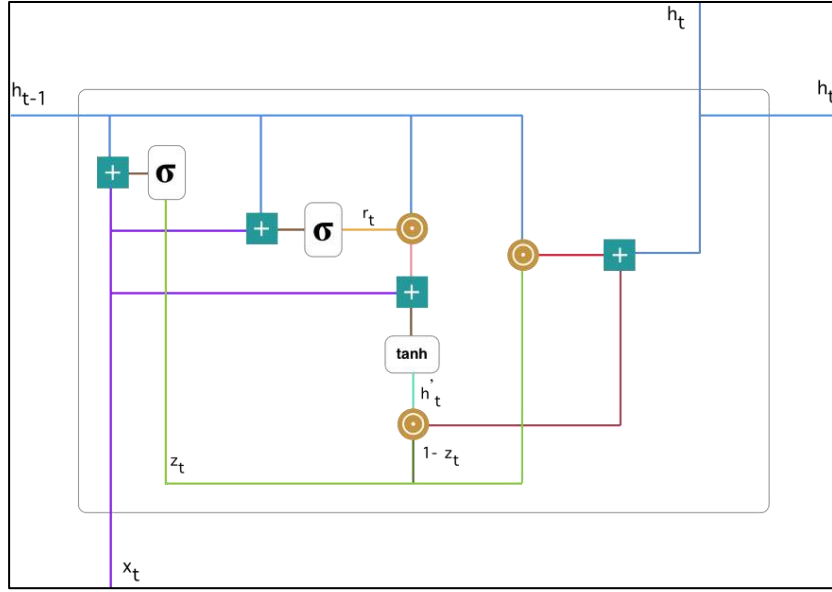
Şekil 3.21. LSTM ve BLSTM mimarisi karşılaştırması [109]

3.2.6. Geçitli tekrarlayan birim (Gated recurrent unit, GRU)

Kyunghyun Cho ve ekibi tarafından 2014 yılında ortaya atılan GRU sinir ağları LSTM benzerliğiyle dikkat çeker. Ancak LSTM'ye oranla daha az parametreye sahip olup daha küçük veri setlerinde daha iyi performans sergilediği ortaya atılmıştır [110]. Literatürde de GRU sinir ağının LSTM sinir ağına oranla daha başarılı sonuçlar elde ettiği çalışmalar bulunmaktadır [111],[112].

GRU bilgilerin sinir ağındaki ilerleyişini kontrol etmek için özel yapılar vardır ve Bunlar “kapı” olarak adlandırılır ve kapılar önceki zamandan gelen bilgilerin unutulup unutulmayacağına ya da bu bilgilerin güncellenip güncellenmeyeceğine karar veren iki tür kapı mevcuttur. Bunlar sırasıyla sıfırlama ve güncelleme kapısı olarak bilinir. Bu kapılar belirli ölçüde gündelik hayattaki kapılarla benzerlik taşımaktadır. Giriş verileri ile çıkış verileri arasındaki karmaşık bağıntıyı öğrenme yeteneği sağlar [113].

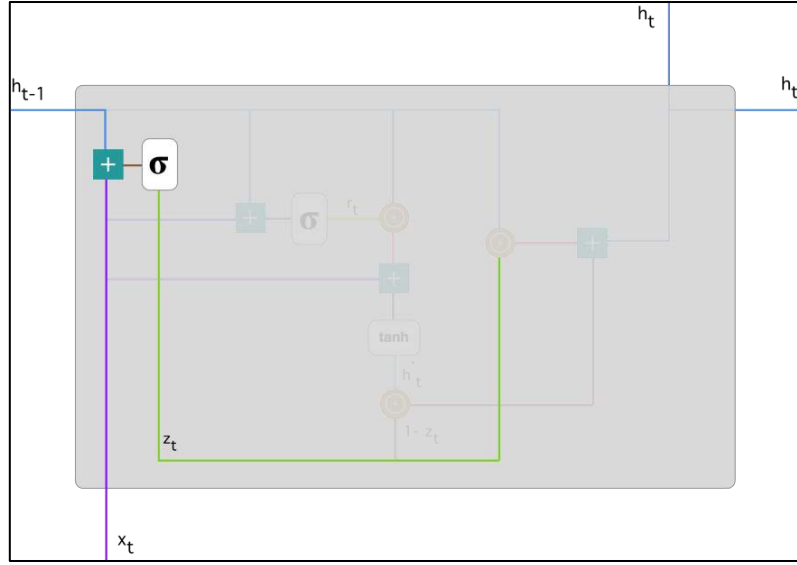
Doğal dil işleme, konuşma sinyallerini modelleme gibi pek çok uygulama alanı bulunmaktadır. GRU yapısının daha az parametreye sahip olmasının sebebi LSTM de kullanılan çıkış kapısının olmamasıdır. Standart bir GRU yapısı Şekil 3.22’de gösterilmiştir.



Şekil 3.22. Standart bir GRU yapısı [110]

Yapının ilk aşamasında kapı güncellemesi yapılır. Bu işlemde geçmişten gelen ne kadar verinin geleceğe aktarılacağına karar verir. Girdi verisi x_t ile bir önceki zamandan gelen h_{t-1} kendi ağırlık katsayılarıyla çarpılarak toplanır ve bir sigmoid fonksiyonundan (σ) geçirilerek oluşan sinyal z_t olarak isimlendirilir. (Şekil 3.23). Aynı LSTM yapısındaki gibi sıfır ile bir arasında değer üreten sigmoid fonksiyonu ne kadar verinin bir sonraki aşamaya geçeceğine karar verir. GRU kapı güncellemesi aşamasında yapılan matematiksel işlemler eşitlik (3.45)'te verilmiştir. Eşitlikte; z_t güncelleme kapısını gösterirken, $W^{(z)}$, x_t girdi verilerinin ağırlıklarını ve $U^{(z)}$ ise bir önceki zamandan gelen h_{t-1} verilerinin ağırlıklarını gösterir.

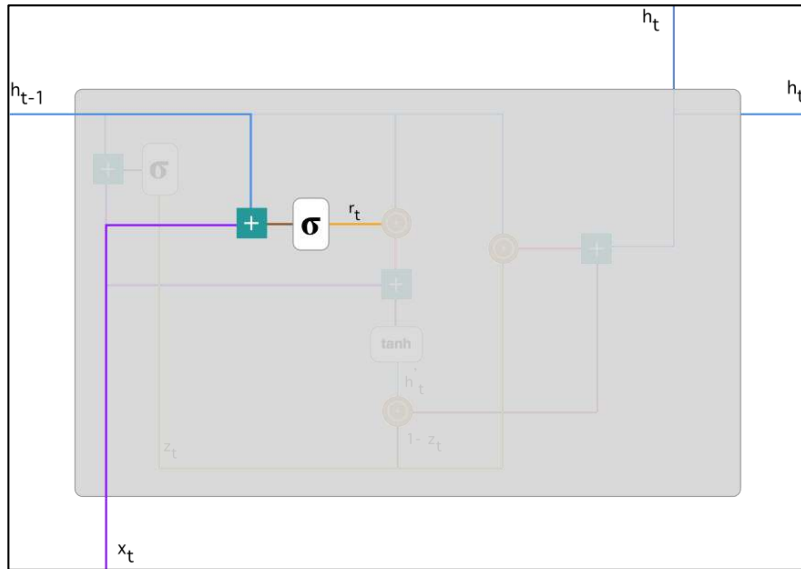
$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (3.45)$$



Şekil 3.23. Kapı güncelleme işlemi [110]

Yapının bir sonraki adımında, geçmişten gelen ne kadar verinin unutulacağına karar verilir, bu adım sıfırlama kapısı (r_t) olarak isimlendirilir. Buradaki matematiksel işlemler kapı güncelleme aşamasında yapılan işlemlerle aynıdır. Var olan küçük farklılık ağırlık katsayılarından kaynaklanır (Şekil 3.24). Sıfırlama kapısına ait matematiksel işlemler eşitlik (3.46)'da gösterilmiştir. Eşitlikte $W^{(r)}$, parametresi x_t girdi verilerinin ağırlıklarını ve $U^{(r)}$ ise bir önceki zamandan gelen h_{t-1} verilerinin ağırlıklarını gösterir.

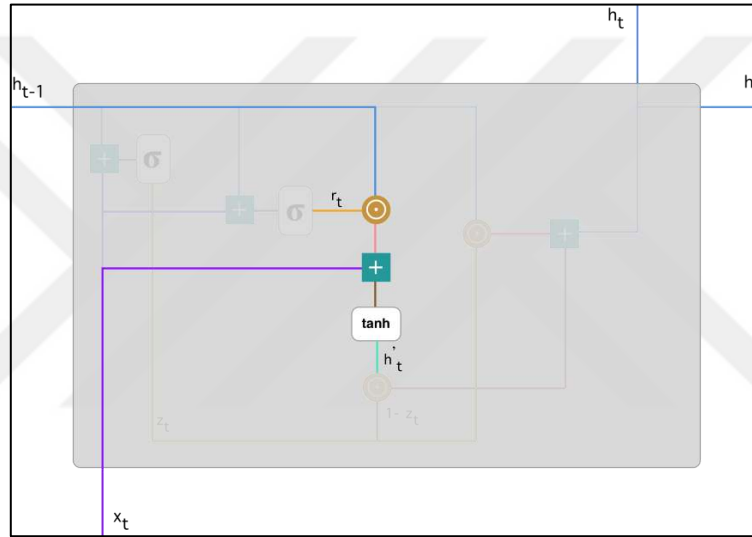
$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (3.46)$$



Şekil 3.24. Sıfırlama kapısı [110]

Yapının bir önceki aşamalarından alınan kapı çıktıların son aşamadaki çıktıyı ne kadar etkilediğine ise üçüncü adımda karar verilir. İlk olarak sıfırlama kapısından gelen veri (r_t), bir önceki zamandan gelen veri (h_{t-1}) ile kendi ağırlıklarıyla çarpılır ve ardından Hadamard (Öge bazlı) çarpım işlemi gerçekleştirilir. Çarpım sonucu girdi verisi (x_t) toplanarak $\tanh$ fonksiyonuna uygulanır (Şekil3.25). GRU yapısının üçüncü adımında yapılan matematiksel işlemler eşitlik (3.47)'de verilmiştir. Eşitlikte $\odot$ sembolü Hadamard çarpımı sembolünü, h'_t ise $\tanh$ çıktısıdır. W parametresi, x_t girdi verilerinin ağırlıklarını, U ise bir önceki zamandan gelen h_{t-1} verilerinin ağırlıklarıdır.

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (3.47)$$

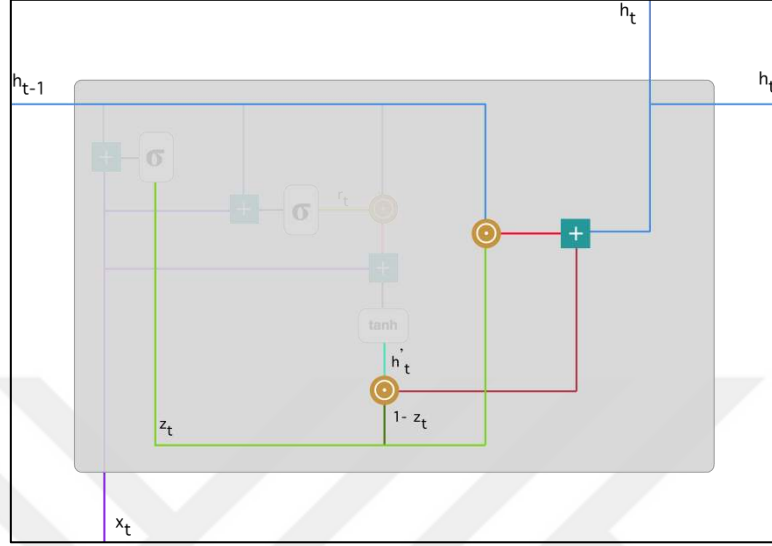


Şekil 3.25. GRU yapısının üçüncü adımı [110]

Son aşamada ise h_t vektörünün hesaplaması yapılarak bir sonraki aşamaya o anki birimden ne kadar verinin geçirileceğine karar verilir. Bu işlem için güncelleme kapısı çıktısına (z_t) ihtiyaç vardır. Güncelleme kapsısı vektörü z_t çıktısı 1 değerine yakın olursa $1 - z_t$ işlemi sıfıra yakınsayarak o andaki güncel verinin büyük bir kısmının atılacağı anlamına gelir. Model z_t vektörünün 1 değerine yakınsamasını öğrenerek bir önceki adımdan gelen büyük miktarda verinin tutulmasını sağlar. İlk olarak $1 - z_t$ ile bir önceki katmanda kullanılan $\tanh$ çıktısı (h'_t) Hadamard çarpımı ($\odot$) yapılır. Aynı zamanda bir önceki zaman verisi (h_{t-1}) ile güncelleme kapısı çıktısı da aynı çarpım işlemine tabi tutulur. İki çarpım sonucu toplanarak h_t vektörünü meydana getirir (Şekil 3.26). GRU yapısının son işlem basamağında yapılan matematiksel işlemler eşitlik (3.48)'de verilmiştir.

Anlatıldığı gibi GRU modeli, her seferinde yeni bir giriş almayarak ağın ilgili verilerini tuttuğu için daha iyi öğrenebilme yetisine sahip olduğu yönünde yorum yapılabilir.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (3.48)$$

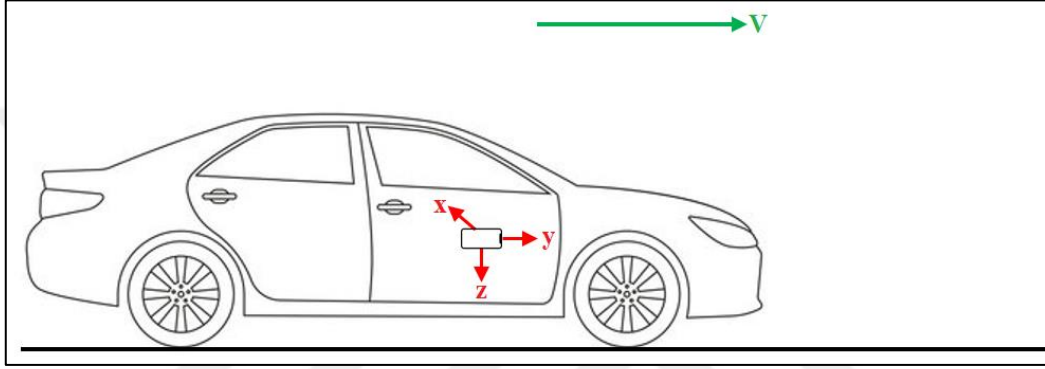


Şekil 3.26. GRU yapısının son işlem basamağı [110]

4. BENZETİM ÇALIŞMALARI

4.1. Geleneksel Kestirim Yöntemleri İçin Kinematik Modelin Oluşturulması

Tez kapsamında gerçekleştirilen çalışmalarda Bölüm 3.1’de ayrıntılı bir şekilde anlatılan veri seti kullanılmıştır. Bu veri seti, akıllı telefon ivmeölçer ve jiroskop algılayıcılarından istifade ederek veri üretmektedir. Kestirim çalışmalarında bir otomobilin şehirler arası yolda seyir halindeyken alınan veriler kullanılmıştır. Kullanılan veri seti için sensör yerleşimlerini temsil eden görsel Şekil 4.1’de gösterilmektedir.



Şekil 4.1. Sistemin temsili gösterimi

Yapılan çalışmalarda, geleneksel yöntemlerle kestirim uygulaması Kalman filtresi ve Luenberger gözlemcisi kullanılarak yapılmıştır. Bu yöntemler, uygulandığı sistemin matematiksel modeline ihtiyaç duyar.

Sistemin matematiksel modeli oluşturulurken Newton’un hareket denklemlerinden ve ayrık beyaz gürültü modelinden (Gaussian White Noise) yararlanılmıştır. Bu tez kapsamında x ve y eksenindeki veriler kullanıldığından; sistem modeli, x ve y eksenindeki konum ve hız parametrelerinden oluşmaktadır. Bu durumda oluşturulan sistem dinamiği denklemleri eşitlik (4.1)’den (4.15)’e kadar verilmektedir.

Eşitlik (4.1) ve (4.2)’de sırasıyla x ve y eksenini için hızın sonraki adımdaki değeri ile şimdiki adımdaki değerinin karşılaştırması verilmiştir.

$$V_x(k + 1) = V_x(k) \quad (4.1)$$

$$V_y(k + 1) = V_y(k) \quad (4.2)$$

V_x = x eksenindeki hız

V_y = y eksenindeki hız

Denklemlerde k zaman indisidir. Formüller oluşturulurken hız sabit olarak kabul edilir ve sistem buna göre modellenir.

Eşitlik (4.3) ve (4.4)'te ise x ve y eksenleri için bir sonraki adımdaki konum bilgisi şimdiki konum, şimdiki hız ve zaman bilgisi cinsinden verilmiştir.

$$X_x(k+1) = X_x(k) + \Delta t V_x(k) \quad (4.3)$$

$$X_y(k+1) = X_y(k) + \Delta t V_y(k) \quad (4.4)$$

X_x = x eksenindeki konum

X_y = y eksenindeki konum

Δt = Sensör örnekleme periyodu

Yukarıda elde edilen denklemler doğrultusunda sistemin durum matrisi eşitlik (4.5)'teki gibi elde edilir.

$$\begin{bmatrix} X_x(k+1) \\ V_x(k+1) \\ X_y(k+1) \\ V_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_x(k) \\ v_x(k) \\ x_y(k) \\ v_y(k) \end{bmatrix} + G w_n \quad (4.5)$$

Denklemlerde geçen Δt terimi verilerin algılayıcılardan hangi periyotlarla alındığını göstermektedir. Bölüm 3.1'de anlatılan veri setinde akıllı telefon algılayıcılarının ölçüm frekansı 20 Hz'dir. Diğer bir ifadeyle her 50 ms de bir veri elde edilmektedir. Dolayısıyla Δt değeri 0.05 sn olmaktadır.

Eşitlik (4.5)'te yer alan $G w_n$ modeli (ayrık beyaz gürültü modeli) eklendiğinde eşitlik (4.6) elde edilir.

$$\begin{bmatrix} X_x(k+1) \\ V_x(k+1) \\ X_y(k+1) \\ V_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_x(k) \\ v_x(k) \\ x_y(k) \\ v_y(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} w_x(n) \\ w_y(n) \end{bmatrix} \quad (4.6)$$

Sistemin çıkış tepkisi eşitlik (4.7) ile verilmektedir.

$$\begin{bmatrix} V_x(k) \\ V_y(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_x(k) \\ V_x(k) \\ X_y(k) \\ V_y(k) \end{bmatrix} + v_n \quad (4.7)$$

Hızı sabit kabul etmek yerine ivmenin sabit kabul edildiği bir sistem modeli hazırlanırsa sistem dinamikleri eşitlik (4.8) ve (4.9)'da ki gibi tanımlanmaktadır.

$$a_x(k+1) = a_x(k) \quad (4.8)$$

$$a_y(k+1) = a_y(k) \quad (4.9)$$

a_x = x eksenindeki ivme

a_y = y eksenindeki ivme

Eşitlik (4.10) ve (4.11)'de sabit ivmeye göre hız formülleri türetilmiştir.

$$V_x(k+1) = V_x(k) + a_x(k) \cdot \Delta t \quad (4.10)$$

$$V_y(k+1) = V_y(k) + a_y(k) \cdot \Delta t \quad (4.11)$$

Eşitlik (4.12) ve (4.13)'te x ve y eksenlerindeki konum bilgilerine ilişkin sistem dinamikleri verilmektedir.

$$x_x(k+1) = x_x(k) + \Delta t \cdot V_x(k) + a_x(k) \cdot \frac{\Delta t^2}{2} \quad (4.12)$$

$$x_y(k+1) = x_y(k) + \Delta t \cdot V_y(k) + a_y(k) \cdot \frac{\Delta t^2}{2} \quad (4.13)$$

Eşitlik (4.14)'te sistemin durum denklemi matris gösterimi verilmiştir.

$$\begin{bmatrix} X_x(k+1) \\ V_x(k+1) \\ a_x(k+1) \\ X_y(k+1) \\ V_y(k+1) \\ a_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_x(k) \\ V_x(k) \\ a_x(k) \\ X_y(k) \\ V_y(k) \\ a_y(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 1 & 0 \\ 0 & \frac{\Delta t^2}{2} \\ 0 & \Delta t \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} w_x(k) \\ w_y(k) \end{bmatrix} \quad (4.14)$$

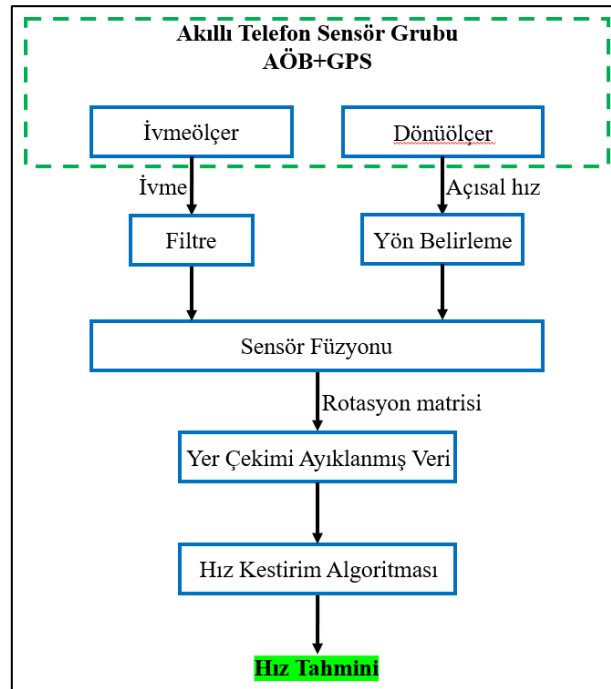
İvme sabit olarak kabul edildiğinde sistemin çıkış tepkisi eşitlik (4.15)'te verilmektedir.

$$\begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} + v_n \quad (4.15)$$

4.2. Geleneksel Kestirim Yöntemleri İçin Benzetim Sonuçları

Bölüm 4.1’de detayları verilen sistemin matematiksel modeli kullanılarak hız kestirimi yapılmıştır. Kestirim yapılırken, Kalman filtresi ve Luenberger gözlemcisi ele alınmıştır. Kalman filtresiyle kestirim yapılabildiği halde Luenberger gözlemcisi ile birtakım sınırlamalardan dolayı sorunla karşılaşmıştır. Luenberger gözlemcisinin uygulanamamasının sebepleri Bölüm 4.2.2’de ayrıntılı bir şekilde açıklanmıştır.

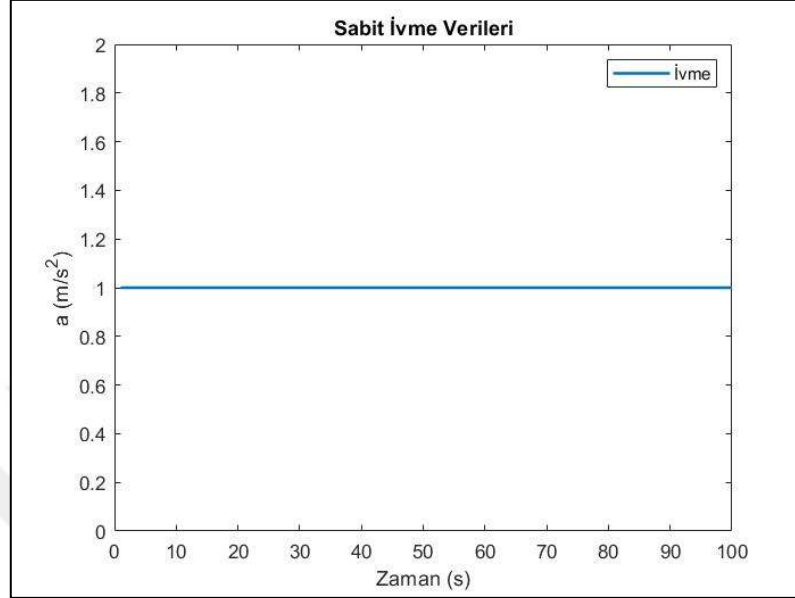
Geleneksel yöntemler ile hız kestirimi yapılmadan önce akıllı telefon yardımı ile elde edilen ivme verilerindeki gürültülü değerler filtreden geçirilerek ivme verileri temizlenmiştir. Ardından akıllı telefonun dönüölçer sensörlerinden elde edilen açı değerleri, Bölüm 3.2.1’de anlatılan Rotasyon Matrisi yönteminde kullanılarak ivme verilerinden yer çekimi etkisi kaldırılmış ve geleneksel hız kestirim algoritmaları için ivme değerleri kullanılabilir hale getirilmiştir. Bu tez kapsamında kullanılan ivme verilerinin hazırlanmasındaki işlemler akış diyagramı olarak Şekil 4.2’de verilmiştir.



Şekil 4.2. Geleneksel hız kestirim yöntemleri için ivme verilerini hazırlama aşamaları

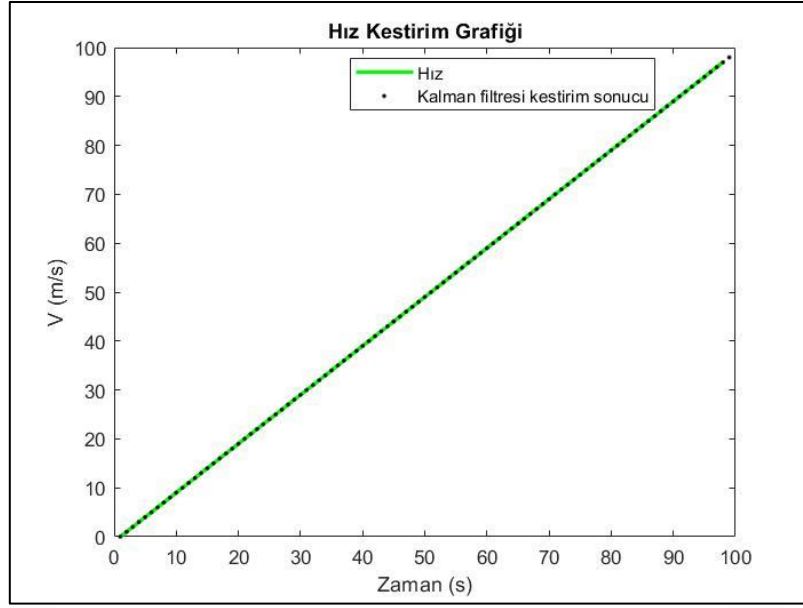
4.2.1. Kalman filtresi benzetim sonuçları

Oluşturulan Kalman filtresi algoritmasının doğruluğunun teyit edilebilmesi amacıyla sabit ivmeli bir yapay model oluşturulmuştur. Şekil. 4.3'te yapay verilerin ivme-zaman grafiği verilmiştir.



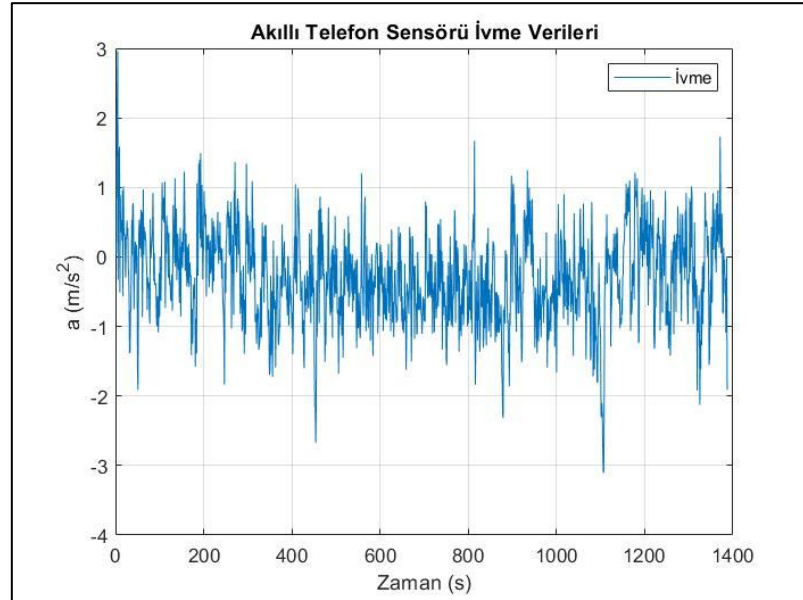
Şekil 4.3. Yapay verilerin ivme-zaman grafiği

Sabit ivme ile oluşturulan yapay ivme veri seti, bu tez kapsamında tasarlanan Kalman filtresi algoritmasında işleme konulmuştur. Yapay veri setine ait kestirim sonuçları Şekil 4.4'te verilmiştir. Şekil 4.4'teki grafikte yeşil çizgi hızı, siyah çizgi ise hız kestirim değerlerini göstermektedir. Grafiğin çıktısı incelendiğinde yapay hız değerlerinin tahmini hız değerleriyle üst üste gelerek başarılı bir kestirim yapıldığı görülmektedir. Böylece, tasarlanan Kalman filtresi algoritmasının başarılı olduğu ve bu tez kapsamında incelenecek olan veri setinin de bu tasarlanan Kalman filtresi algoritmasıyla denenebileceği ortaya çıkmaktadır.



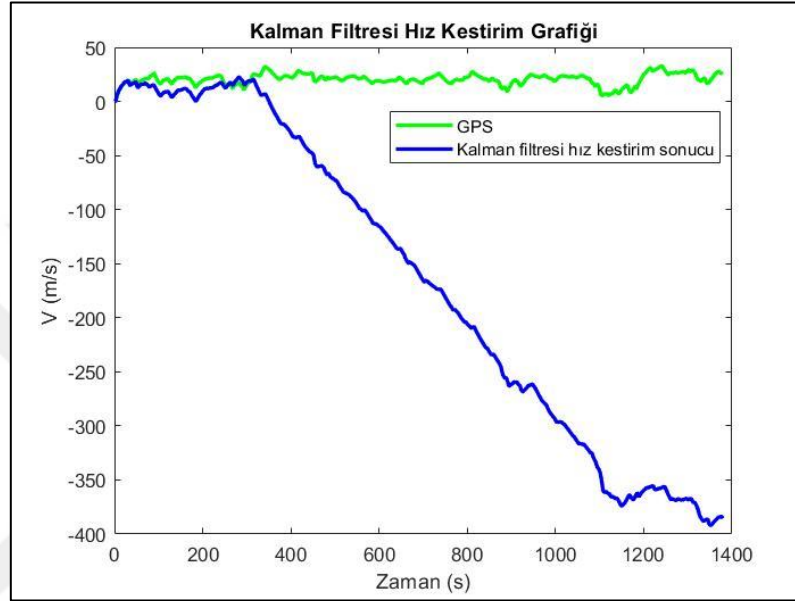
Şekil 4.4. Oluşturulan yapay ivme veri setinden elde edilen hız kestirim sonucu

Bu tez kapsamında Bölüm 3.1’de açıklanan veri setinin akıllı telefon yardımıyla elde edilen ivme verilerinin y eksenini çıktıkları Şekil 4.5’te verilmiştir. Grafikte yer alan ivme verileri otomobilin şehirler arası yolda değişken hızlarda seyir halindeyken alınan akıllı telefon ivmeölçer verilerinden oluşmaktadır. Şekil 4.5’te 1380 saniyelik ivme verileri yer almaktadır. Bu verilerden hız kestirimi yapılırken; veri setine akıllı telefon yardımı ile kaydedilen GPS verileri ile kıyaslanarak hız kestirimi yapılmıştır.



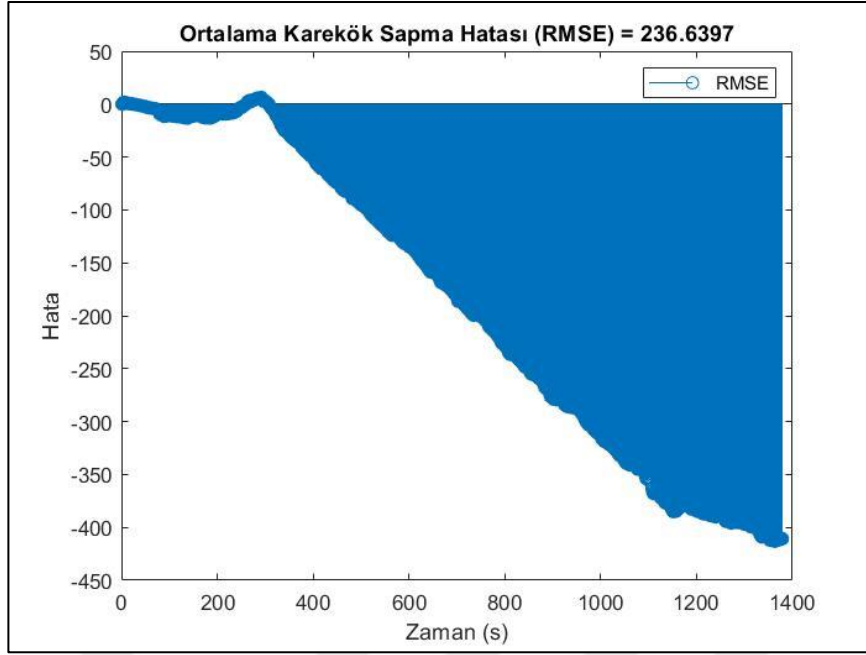
Şekil 4.5. Akıllı telefon ivmeölçer algılayıcılarından elde edilen y eksenini ivme verileri

Şekil 4.6’da Kalman Filtresi ile hız kestirimi sonucu çıkan grafik verilmiştir. Grafik incelendiğinde 310 saniye boyunca gayet başarılı bir hız kestirimi yaptığı görülmekte ve 310. Saniyeden sonra sert bir şekilde aşağı yönlü negatif hız kestirimi yapmaktadır. Bunun sebebi Kalman Filtresinin yapısından kaynaklanmakta ve ivme verilerindeki hatalardan dolayı birikerek giden tahmin kayması bir süre sonra büyük hatalara sebep olmaktadır. Ayrıca kaymanın sebebinin başka sebeplere bağlı olabileceği düşünülmektedir. Bu duruma dair ayrıntılı bilgi ilerleyen kısımlarda açıklanmıştır.



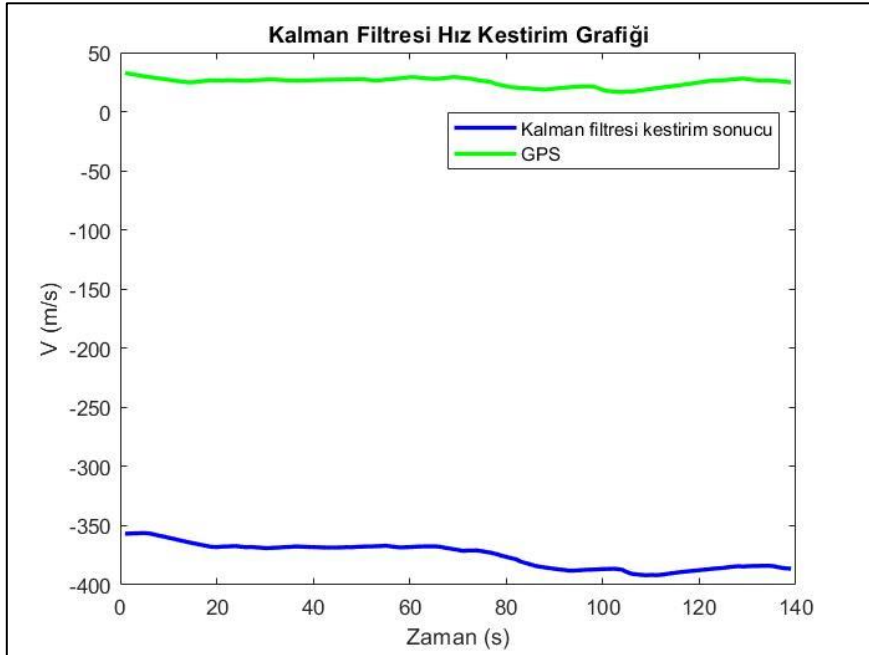
Şekil 4.6. Kalman filtresi hız kestirim grafiği

Veri setinin tamamıyla yapılan hız kestirimin RMSE değeri 236,6397 olarak bulunmuş ve grafiği Şekil 4.7’de verilmiştir. Grafik incelendiğinde sensör ölçümlerindeki küçük hataların birikerek, Kalman filtresini kullanılamaz hale getirmektedir. Daha önce de belirtildiği gibi grafiğin ilk bölümünde (310. Saniyeye kadar) RMSE değerleri daha kabul edilebilirdir.



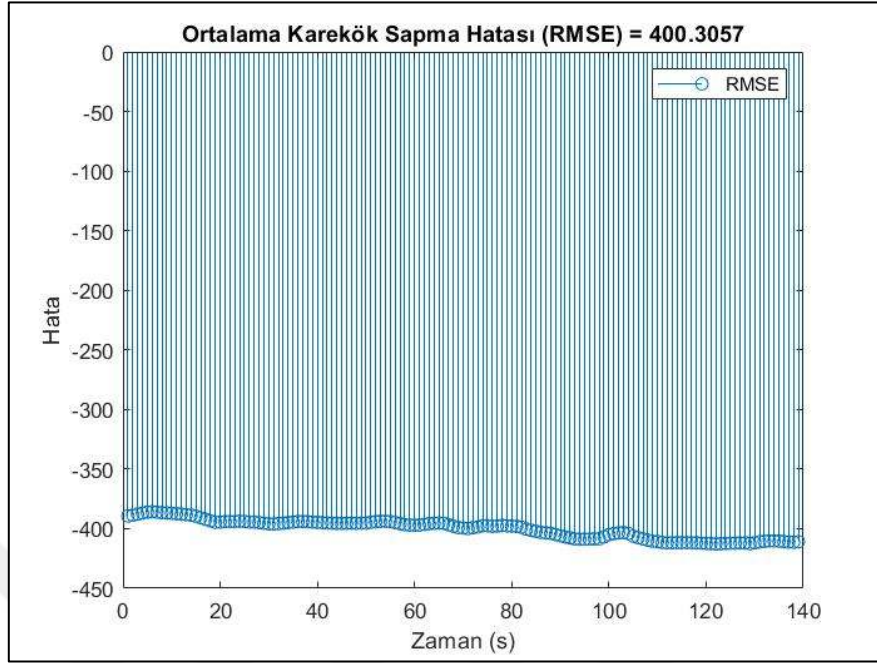
 ekil 4.7. T m veri setinin ortalama karek k sapma hatas 

Kalman filtresi h z tahminlerini, derin  ğrenme tabanlı kestirim sonu larıyla kıyaslamak amacıyla  ekil 4.6'da verilen grafiğın son %10'luk kısmı  ekil 4.8'de verilmiřtir. Grafik incelendiğinde Kalman filtresi zaman ge tikte sens r  l  m hatalarından dolayı birikmeli bir hataya maruz kalır ve bir s re sonra h z kestirimi ile GPS değeri arasında b y k bir fark oluřmaktadır.



 ekil 4.8. Kalman Filtresi h z kestirim grafiğı (138 saniye)

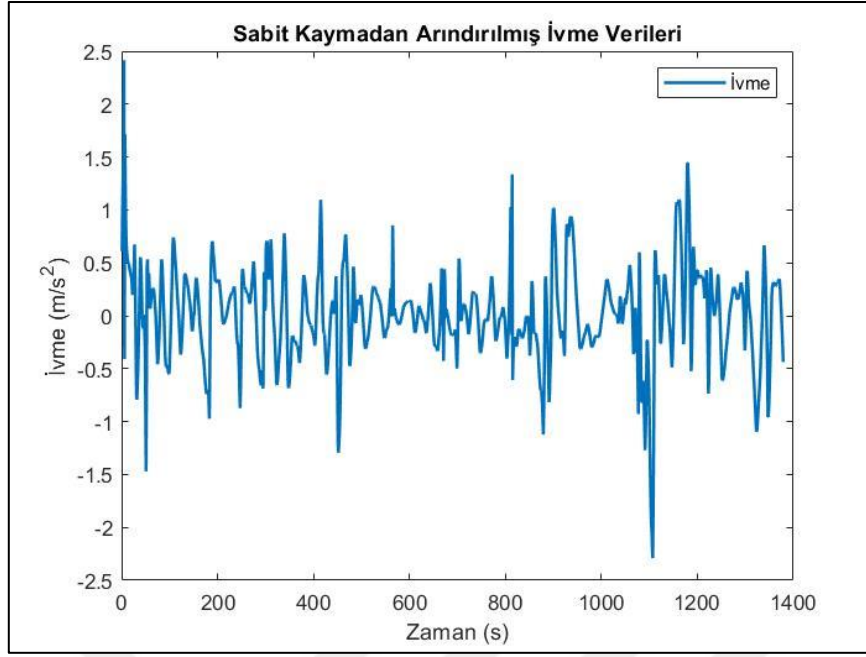
Kalman filtresi ile yapılan hız kestirim sonucunun son 138 saniyelik kısmında, RMSE değeri 400,3057 çıkmıştır ve grafiği Şekil 4.9’da verilmiştir.



Şekil 4.9. Ortalama karekök sapma hatası (138 saniye)

Kalman Filtresi ile elde edilen hız kestirim sonuçları incelendiğinde Şekil 4.6’da da görüldüğü üzere kestirim sonuçları 310.saniyeden sonra neredeyse doğrusal bir şekilde negatif yönde kaymaktadır. Bu kaymanın sebebinin, gürültü birikmesinin etkisi olduğu gibi sensörlerden ölçüm alınması sırasında sıcaklık değişimi ve sensör kalibrasyon hatalarından kaynaklı ivme verilerinde sabit bir kayma olabileceği düşünülmektedir. Bu sebeple akıllı telefon sensör verilerinin sabit kayması olabildiğince temizlenmeye çalışılmıştır.

Akıllı telefon ivme verilerinin hareketli ortalaması kullanılarak ivme verilerinin sabit kayması arındırılmıştır. İvme verilerinin arındırılmış hali Şekil 4.10’da verilmiştir.

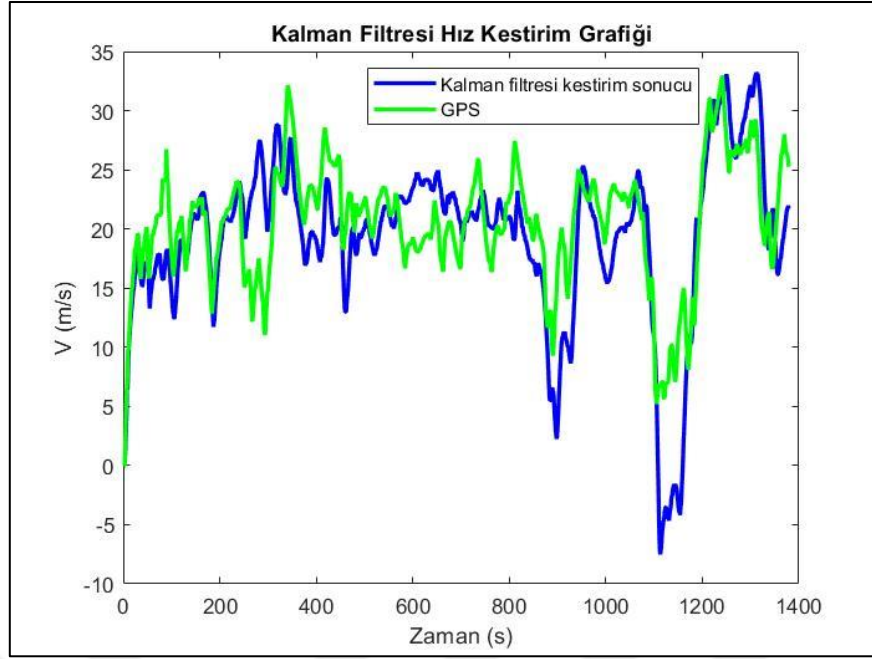


Şekil 4.10. Sabit kaymadan arındırılmış ivme verileri

Sabit kaymadan arındırılmış ivme verileri, Kalman Filtresi hız kestirim algoritmasına uygulanmıştır. Hız kestirim sonuçları Şekil 4.11’de verilmiştir. Arındırılmış ivme verileri ile hız kestirim grafiği incelendiğinde görülmektedir ki Şekil 4.6’daki hız kestirimine göre çok daha başarılı kestirim sonuçları elde edilmiştir. Şekil 4.6’daki doğrusal kayma büyük oranda giderilmiştir.

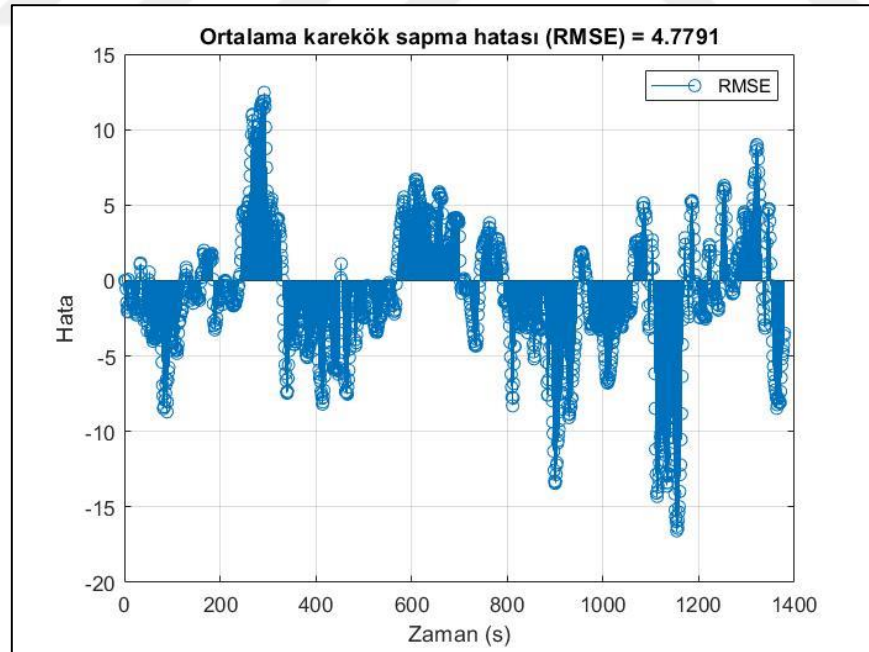
Şekil 4.11’de hız kestirim çıktılarında ufak kaymalar mevcuttur. Hız kestirim çıktılarında kayma olmasına rağmen hız kestirim çıktıları, GPS hız verilerindeki değişimlere de tepki vermektedir.

Sonuç olarak daha önce Kalman Filtresi ile yapılan kestirim sonuçlarındaki büyük kaymalar, akıllı telefon ivme verilerinin hareketli ortalaması alınarak ölçüm verilerindeki sabit kaymalar kaldırılmış ve daha başarılı sonuçlar elde edilmiştir.



Şekil 4.11. Arındırılmış ivme verileri ile hız kestirim grafiği

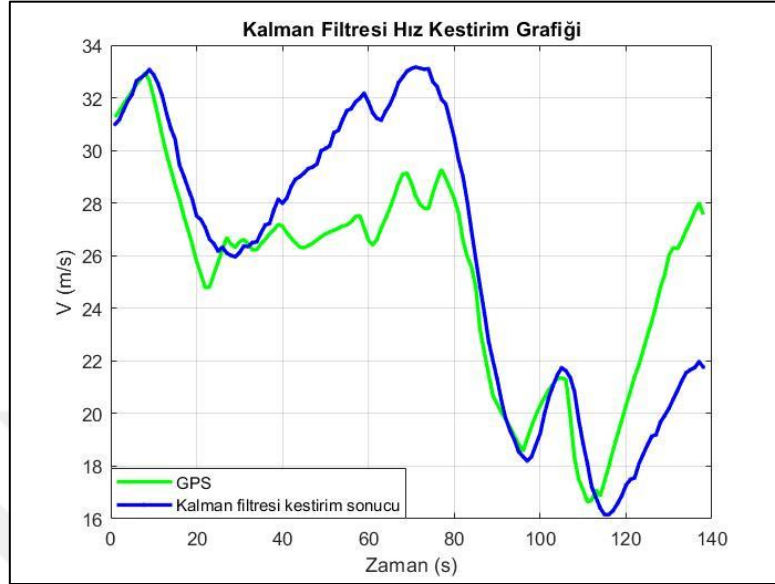
Arındırılmış ivme verileri ile elde edilen kestirim sonuçlarına ait RMSE değerleri Şekil 4.12’de verilmiştir. Şekil 4.7’de ki RMSE değerleri ile kıyaslandığında daha başarılı kestirim sonuçları elde edildiği net bir şekilde görülmektedir.



Şekil 4.12. Kestirim sonucu elde edilen RMSE değerleri

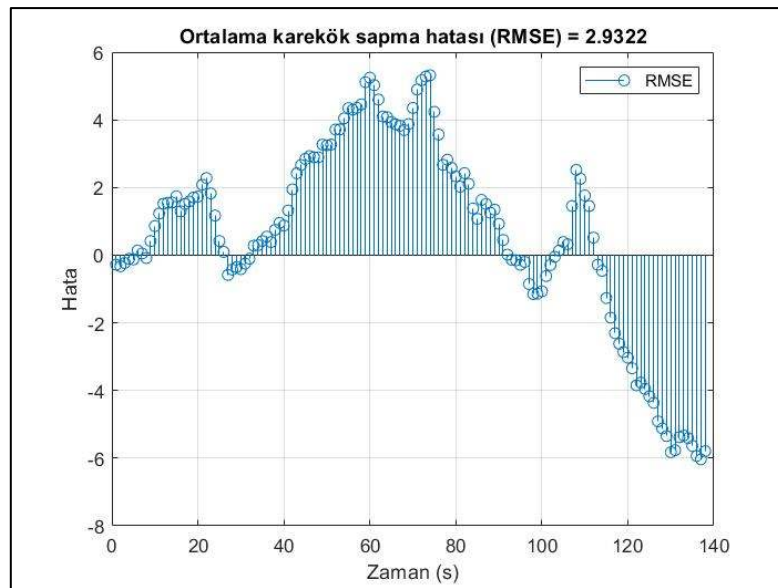
Sabit kaymadan arındırılmış ivme verileri kullanılarak Kalman Filtresi ile yapılan hız kestirimleri, derin öğrenme tabanlı kestirim sonuçlarıyla kıyaslamak amacıyla Şekil 4.11’de

verilen grafiğin son %10'luk kısmı Şekil 4.13'te verilmiştir. Grafik incelendiğinde Kalman filtresi hız kestirim sonuçlarında kaymalar mevcuttur. Ancak Şekil 4.8'deki ilk hız kestirim durumuyla karşılaştırıldığında daha başarılı sonuçlar vermiştir.



Şekil 4.13. Arındırılmış ivme verileri ile Kalman Filtresi hız kestirimi (138 saniye)

Sabit kaymadan arındırılmış ivme verileri kullanılarak Kalman filtresi ile yapılan hız kestirim sonucunun son 138 saniyelik kısmında, RMSE değeri 2,9322 çıkmıştır ve grafiği Şekil 4.14'te verilmiştir. Görüldüğü üzere Şekil 4.9'da verilen bir önceki durumun kestirim sonuçlarına göre daha başarılı sonuçlar elde edilmiştir.



Şekil 4.14. Kestirim sonucu elde edilen RMSE değerleri (138 saniye)

4.2.2. Luenberger gözlemcisi benzetim alışmaları

Luenberger gözlemcisi ile kestirim yapılabilmesi için sistem modelinin gözlenebilir ve kontrol edilebilir olması gerekmektedir.

Bölüm 4.1’de özetlenen sistemin kinematik modelinin matris gösterimi eşitlik (4.6)’da verilmiştir. Aşağıdaki denklemlerde Luenberger gözlemcinin uygulanma koşulları olan gözlenebilirlik ve kontrol edilebilirlik durumlarının incelenebilmesi için sistem matrisleri verilmiştir. Bölüm 3.2.2’de açıklanan, bir sistemin gözlenebilir ve kontrol edilebilir olup olmadığı, yine aynı bölüm başlığı altında verilen eşitlik (3.11) ile (3.12) kullanılarak sınanır. Bu sınama işlemi sırasında; sistem matrisi A, giriş matrisi B ve çıkış matrisi C’ye ihtiyaç duyulmaktadır. Eşitlik (4.16)’da sistem matrisi olan A matrisi verilmiştir.

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

Sistemin kontrol edilebilir olup olmadığını sınamak amacıyla kullanılan giriş matrisi B, eşitlik (4.17)’de verilmiştir.

$$B = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 0 & \frac{\Delta t^2}{2} \\ 0 & \Delta t \end{bmatrix} \quad (4.17)$$

Sistemin gözlenebilir olup olmadığını sınamak amacıyla çıkış matrisi olan C matrisi eşitlik (4.18)’de verilmiştir.

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

Sistem matrisi, giriş matrisi ve çıkış matrisi yardımıyla eşitlik (3.11)’de verilen gözlenebilirlik matrisi olan V matrisi ile eşitlik (3.12)’de formülü verilen kontrol edilebilirlik matrisi olan S matrisi elde edilir.

Gözlenebilirlik matrisi olan V matrisi eşitlik (4.19)’da verilmiştir. V matrisinin gözlenebilir olması için V matrisinin rankı tam çıkmalıdır. Bu durumda V matrisini rankı 4 çıkmazsa sistem gözlenebilir değildir.

$$V = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.19)$$

V matrisinin rankı hesaplandığında 2 çıkar. Rank(V)≠4 olması nedeniyle sistem gözlenebilir değildir.

Eşitlik (4.20)'de kontrol edilebilirlik matrisi olan S matrisi verilmiştir. Sistemin kontrol edilebilir olması için S matrisinin rankı tam olmalıdır. Bu durumda S matrisinin rankı 4 çıkmazsa sistem kontrol edilebilir değildir.

$$S = \begin{bmatrix} 0.0013 & 0 & 0.0038 & 0 & 0.0063 & 0 & 0.0088 & 0 \\ 0.05 & 0 & 0.05 & 0 & 0.05 & 0 & 0.05 & 0 \\ 0 & 0.05 & 0 & 0.0038 & 0 & 0.0063 & 0 & 0.0088 \\ 0 & 0.0013 & 0 & 0.05 & 0 & 0.05 & 0 & 0.05 \end{bmatrix} \quad (4.20)$$

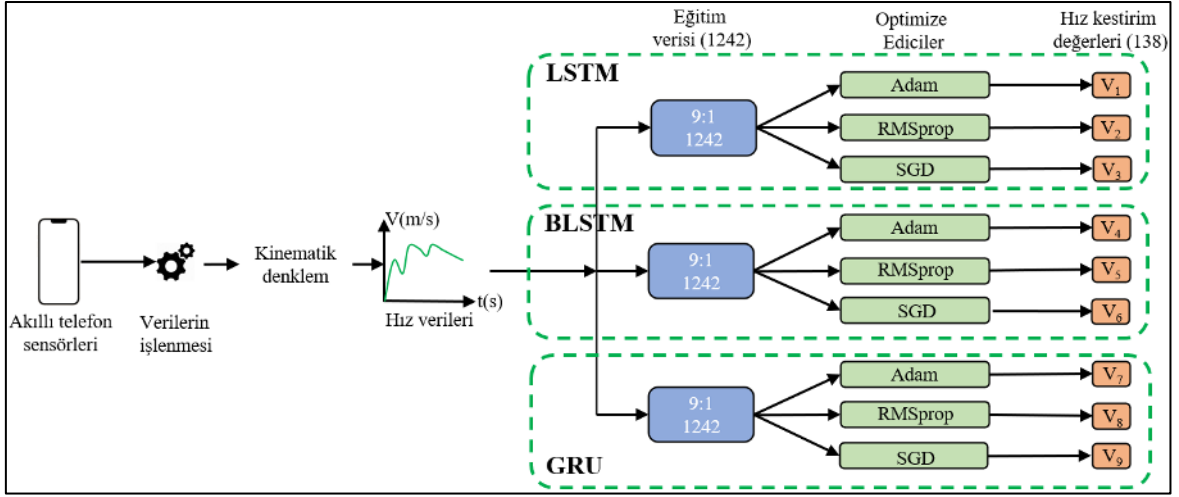
S matrisinin rankı hesaplandığında 4 çıkar. Rank(S)=4 olduğunda dolayı sistem kontrol edilebilirdir.

Sonuç olarak sistem kontrol edilebilir olsa da gözlenebilirlik şartını sağlamadığından dolayı Luenberger gözlemci algoritması bu sisteme uygulanamaz.

4.3. Derin Öğrenme Tabanlı Yöntemler ile Kestirim

Bu tez çalışmasında son yıllarda en sık kullanılan derin öğrenme tabanlı algoritmalar tercih edilmiştir. Bunlar: LSTM, BLSTM ve GRU'dur.

Tez çalışmalarında kullanılan veri seti, eğitim ve test seti olarak sırasıyla 9:1 olarak ayrılmıştır. Veri setinden 1380 örnek alınmış ve bunlardan 1242 tanesi Derin Öğrenme Ağı modelini oluşturmada kullanılırken; kalan 138 tanesi modeli test etmek amacıyla kullanılmıştır. Derin öğrenme ağları hız kestirim akış diyagramı Şekil 4.15'te verilmiştir.

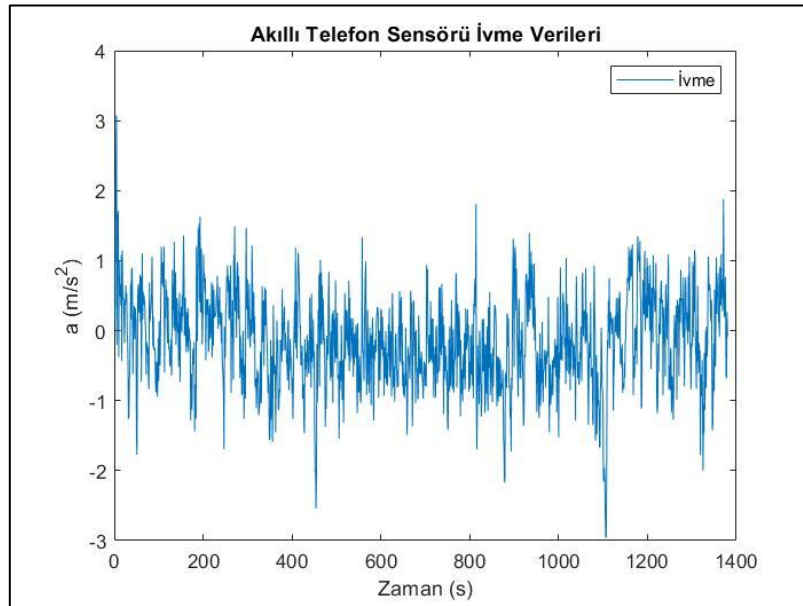


Şekil 4.15. Derin öğrenme ağları hız kestirim akış diyagramı

4.3.1. LSTM algoritması ile alınan benzetim sonuçları

LSTM algoritması, verideki uzun vadeli bağımlılıkları öğrenmek adına bir eğitim setine ihtiyaç duyar. LSTM derin öğrenme ağında kullanılacak veri seti Şekil 4.15'te gösterildiği gibi 9:1 oranında ayrılmıştır.

Veriler eğitim ve test veri seti olarak ayrıldıktan sonra, ivme değerleri normalize edilmiştir ve Bölüm 4.1'de verilen Newton'un hareket yasaları kullanılarak hız değerlerine çevrilmiştir. Akıllı telefon sensörlerine ait derin öğrenme ağlarında hız kestirimi yapmak için kullanılan ham veri seti Şekil 4.16'da grafiksel olarak gösterilmiştir.



Şekil 4.16. LSTM ağında kullanılan ham veri seti

Standardize edilmiş hız verileri, LSTM katmanına girdi olarak verilmiş, tam bağlaşımlı katman ile tek vektör haline getirilmiş ve son aşamada regresyon katmanı ile model çıktıları üretilmiştir. Algoritmada, Maksimum eğitim turu (Max. epoch), Gradyan eşiği (Gradient threshold), Başlangıç öğrenme oranı (Initial learn rate), Gizli birimlerin sayısı (Number of hidden units), Mini parti boyutu (Mini batch size), Karıştırma (Shuffle), L_2 düzenleyici (L_2 regulariation), Doğrulama sıklığı (Validation frequency), Gradyan eşiği yöntemi (Gradient threshold method), Gradyan bozunma faktörü (Gradient decay factor), Kare gradyan bozunma faktörü (Squared gradient decay factor), Sıra uzunluğu (Sequence length), Sıra doldurma değeri (Sequence padding value) ve momentum parametreleri sabit tutulmuştur. Ayrıca; Adam, RMSprop ve SGDM optimize edicileri kullanılarak algoritma başarısını en çok etkileyen Öğrenme adım büyüklüğü (Learn rate drop period) ve Öğrenme adım düşme faktörü (Learn rate drop factor) değiştirilerek model en iyilenmeye çalışılmıştır.

Bölüm 3.2.4'te özetlenen derin öğrenme algoritmalarının tahmin doğrulukları üzerine doğrudan etkisi olan parametreler ve optimize ediciler farklı koşullar altında denenerek tahmin doğruluğuna etkileri incelenmiştir.

Bu tez kapsamında kullanılan LSTM yönteminde 3 çeşit optimizasyon algoritması mevcuttur. Her bir optimizasyon algoritması için dört parametre grubu seçilerek farklı durumlar altında derin öğrenme ağının RMSE değerleri elde edilmiştir. Bu tez çalışmasında elde edilen RMSE değerleri Tablo 4.1'de verilmektedir. Algoritmanın tahmin başarısına öğrenme oranının etkilerini incelemek için başlangıç öğrenme oranı (Initial Learn Rate) 0,005 olarak seçilmiş ve her parametre durumu için bu değer sabit tutulmuştur. Başlangıç öğrenme oranının, LSTM ağının eğitim aşaması boyunca güncellenebilir bir yapıya sahip olabilmesi amacıyla, Öğrenme Adım Büyüklüğü (Learn Rate Drop Period, LDP) ve Öğrenme Adım Düşme Faktörü (Learn Rate Drop Factor, LDF) parametreleri yardımıyla belirli adımlar boyunca değiştirilmiştir. Tablo 4.1'te yer alan parametre durumları takip eden paragraflarda özetlenmiştir.

Birinci parametre durumu için LDP değeri 2, LDF 0,99 olarak belirlenmiştir. Başka bir deyişle burada LDP parametresi ile algoritma kaç adımda bir öğrenme oranına güncelleme yapılacağı seçilmiş ve LDF parametresi ile belirlenen bir katsayı ile öğrenme oranı çarpılarak öğrenme oranı değiştirilmiştir. Birinci durumda her 2 adımda bir öğrenme oranı 0,99 ile çarpılarak oldukça küçük değişiklikler öğrenme oranına uygulanmıştır, ancak bu değişiklik her iki adımda bir olduğundan sık bir güncelleme gerçekleştirilmiş olmaktadır. Tez kapsamında yapılan çalışmalarda ağın eğitim tur sayısı 100 olarak belirlenmiş ve sabit

tutulmuştur. Böylece birinci parametre durumu için her 2 adımda bir yapılan güncelleme ile öğrenme oranı 50 kez güncellenerek eğitim tamamlanmıştır.

İkinci parametre durumunda güncelleme sayısı biraz düşürülmüştür. Her 5 adımda bir yapılan güncelleme ile eğitim boyunca 20 kez güncelleme yapılmıştır. Ancak bu sefer LDF katsayısı biraz daha küçük seçilerek öğrenme oranı her güncelleme sonucunda önceki adıma göre daha hızlı bir şekilde küçülmüştür.

Üçüncü parametre durumunda LDP parametresi 10 seçilerek eğitim boyunca yapılacak güncelleme sayısı bir önceki adıma göre iki katına çıkarılarak eğitim boyunca 10 kez güncelleme yapılması sağlanmıştır. Ancak LDF katsayısı 0,5 olarak belirlendiğinden başlangıç öğrenme oranı her güncelleme adımında LDF ile çarpıldığında değerini yarı yarıya azaltmıştır.

Son parametre durumu olan dördüncü adımda başlangıç öğrenme oranı sabit tutularak derin öğrenme ağının eğitimi boyunca öğrenme oranına herhangi bir güncelleme uygulanmamıştır.

Oluşturulan dört parametre durumunun optimize edicilere göre RMSE değerleri Tablo 4.1’de verilmiş ve tablo oluşturulurken her parametre durumu için algoritma 10 kere çalıştırılarak RMSE değerlerinin ortalaması alınmıştır. Bu tablo incelendiğinde en iyi sonuç Adam optimize edici için 2,5547 ile birinci parametre durumunda elde edilmiştir. Diğer optimize edicilerde bu değere yaklaşıl原因 durumlar olmuştur. Ancak Adam optimize edici ile gerçekleştirilen birinci parametre durumu kadar istikrarlı sonuçlar alınamamıştır.

Netice olarak, bu tez kapsamında yapılan çalışmalarda LSTM ağı ile hız tahmini için en uygun optimize edicinin Adam optimize edici olduğu ortaya çıkmaktadır. Ancak SGDM optimize edici ile elde edilen sonuçlarda da umut vadeden değerler mevcuttur.

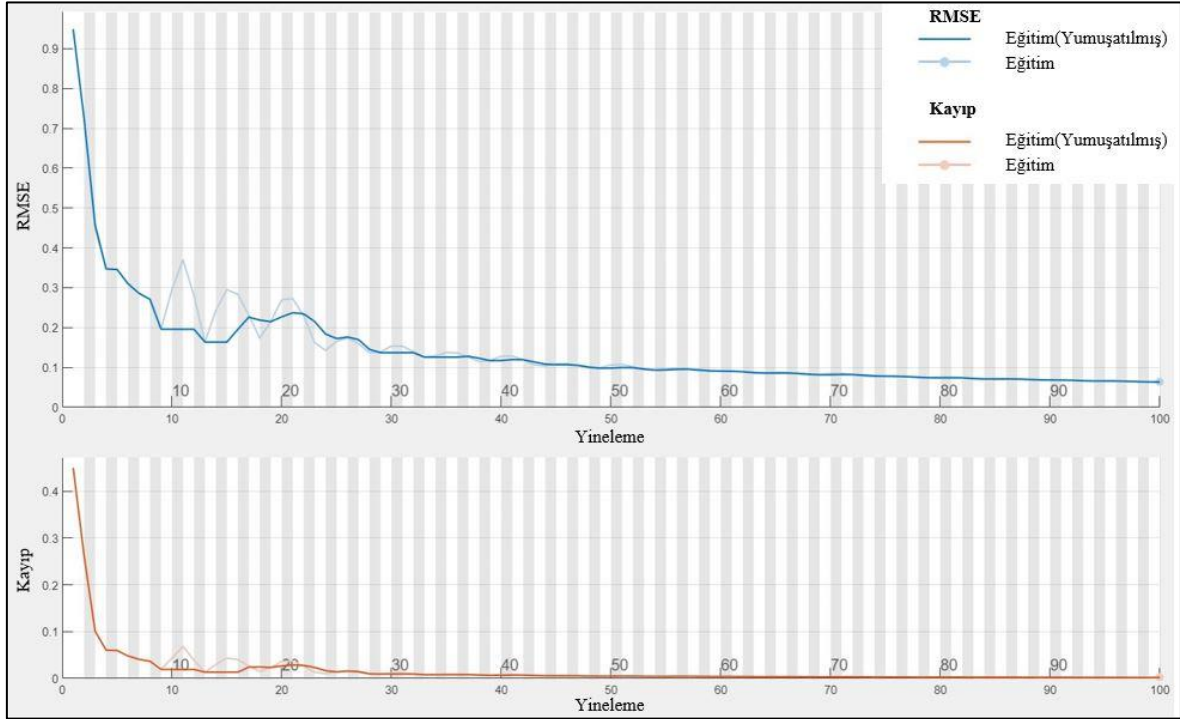
Tablo 4.1 LSTM derin öğrenme ağı hız kestirim sonuçları

Optimize Edici	Parametre Durumu	Parametre	Değer	RMSE
Adam	1	LDP	2	2,5547
		LDF	0,99	
	2	LDP	5	2,81425
		LDF	0,89	
	3	LDP	10	2,84424
		LDF	0,5	
	4	LDP	none	2,9675
		LDF	none	
RMSProp	1	LDP	2	3,32687
		LDF	0,99	
	2	LDP	5	2,9655
		LDF	0,89	
	3	LDP	10	2,8963
		LDF	0,5	
	4	LDP	none	2,7691
		LDF	none	
SGDM	1	LDP	2	2,56546
		LDF	0,99	
	2	LDP	5	2,57395
		LDF	0,89	
	3	LDP	10	3,4473
		LDF	0,5	
	4	LDP	none	2,85125
		LDF	none	

LSTM derin öğrenme ağı ile hız kestirimi sırasında üç farklı optimize edici kullanılmıştır. Bu optimize ediciler ile elde edilen eğitim süreci grafikleri aşağıda açıklanmıştır.

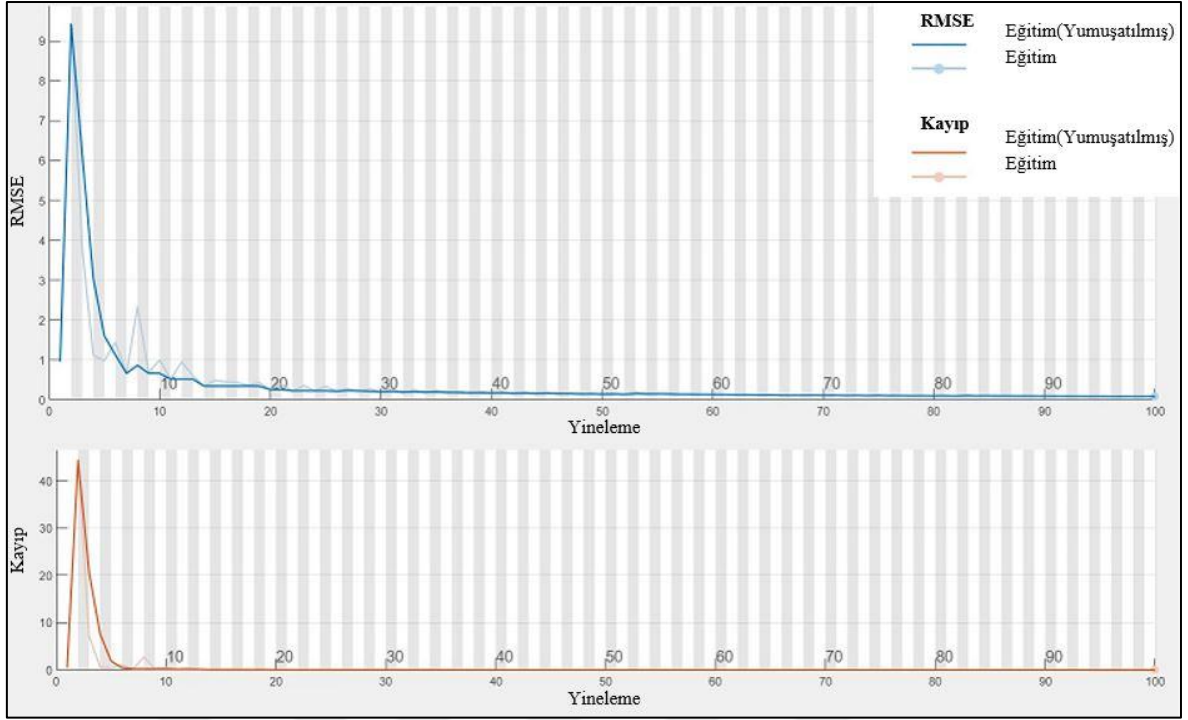
İlk olarak LSTM derin öğrenme ağında kullanılan Adam optimize edicinin eğitim grafiği Şekil 4.17’de verilmiştir. Grafikte mavi renk ile gösterilen RMSE değerlerine bakıldığında 0,95 değerinden başlayarak 5 yinleme adımında RMSE olarak 0,3 değerinin altına hızlı bir şekilde inerek eğitim sürecini stabil hale getirmiştir. Turuncu çizgiler ile gösterilen kayıp değerleri de bu durumu özetler niteliktedir. Kayıp grafiği incelendiğinde ilk

5 yineleme adımından sonra eğitim sürecindeki kayıplar 0.1 değerinin altına inerek oldukça düşük kayıplarla eğitim sürecini tamamlamıştır.



Şekil 4.17. Adam optimize edici eğitim süreci grafiği

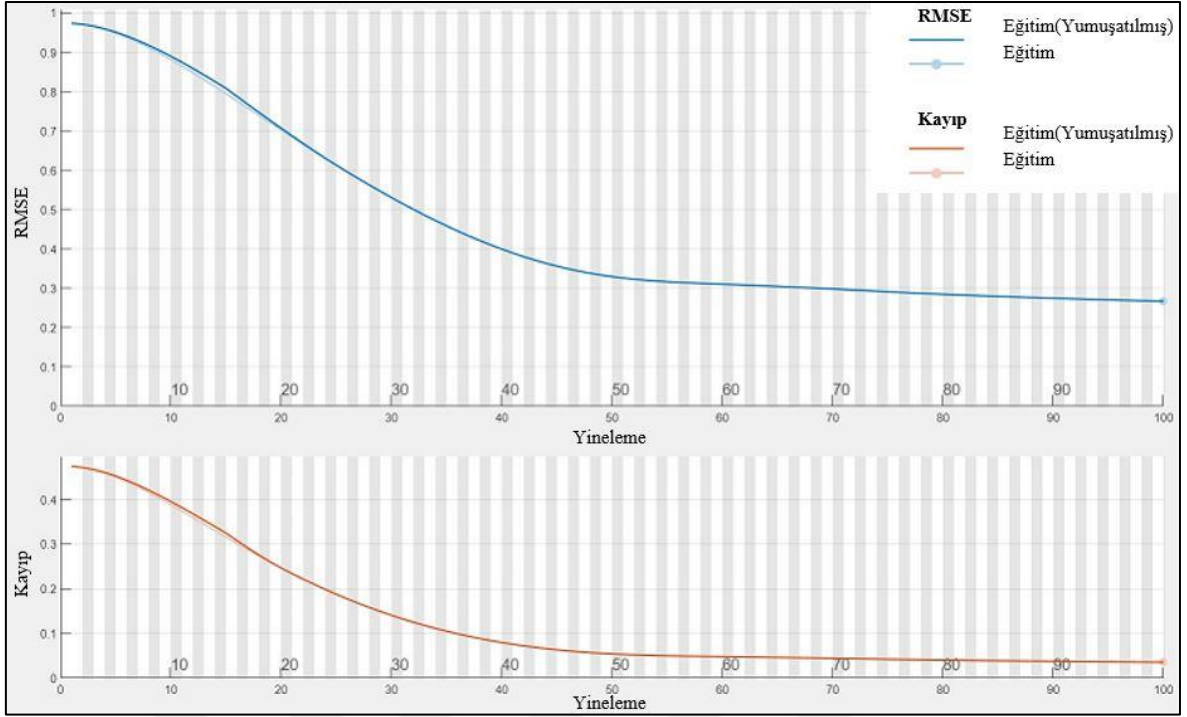
İkinci olarak LSTM derin öğrenme ağında kullanılan RMSprop optimize edicinin eğitim grafiği Şekil 4.18'de verilmiştir. Grafikte mavi çizgi ile gösterilen RMSE değerleri incelendiğinde 1 değerinden 9'lara kadar çıkan bir RMSE değeri görülmektedir. Daha önce grafiği Şekil 4.17'de verilen Adam optimize edicide 5 yineleme adımında 0,3 değerlerine kadar inen RMSE değeri, RMSprop optimize edicide ise 10. Yineleme adımına gelene kadar hata değerleri yükselmiş ve 10. yineleme adımında 1 değerinin altına inebilmiştir. Turuncu renk ile gösterilen kayıp grafiği de incelendiğinde ilk 10 yineleme adımında yüksek kayıplar yaşamıştır. Bu da göstermektedir ki RMSprop optimize edici başlangıçta iyi bir eğitim süreci geçirememiştir.



Şekil 4.18. RMSprop optimize edici eğitim süreci grafiği

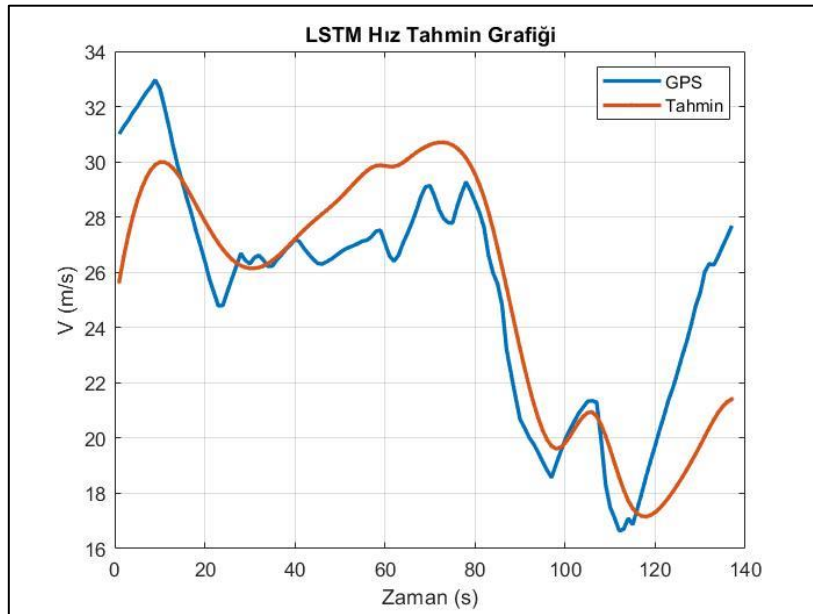
Son olarak LSTM derin öğrenme ağında kullanılan SGDM optimize edicinin eğitim grafiği Şekil 4.19’da verilmiştir. Grafikte mavi çizgi ile gösterilen RMSE değerleri incelendiğinde RMSE değeri 1’den başlamış ve 55. yineleme adımında 0,3 değerine inebilmiştir. SGDM optimize edicinin eğitim aşamasında stabil hale gelmesinin uzun sürmesinin sebebi, optimize edicinin stokastik yapısından kaynaklanmaktadır.

Adam optimize edici ile karşılaştırıldığında SGDM optimize edicinin düşük RMSE değerlerine inmesi daha uzun sürmüştür. Turuncu çizgi ile gösterilen kayıp grafiği incelendiğinde ise kaybın stabil hale gelmesi uzun sürmesine rağmen yüksek kayıplara uğramamıştır. Bu da göstermektedir ki SGDM optimize edici de hız kestirim uygulamalarında başarılı sonuçlar elde edebilir.



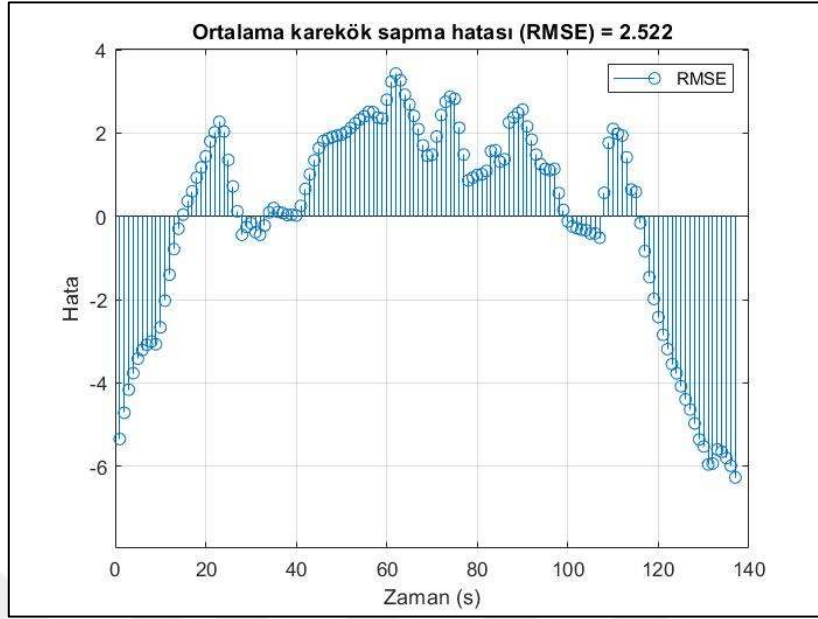
Şekil 4.19. SGDM optimize edici eğitim süreci grafiği

LSTM hız kestirim RMSE değerleri ele alındığında Adam optimize edici ile elde edilen en iyi başarı olarak nitelendirilen birinci parametre durumu ve bu senaryoda ortaya çıkan en düşük RMSE değerine sahip kestirim grafiği Şekil 4.20'de verilmiştir. Grafik incelendiğinde 138 saniyeden oluşan bir hız kestirimi mevcuttur. Kestirim grafiğindeki kaymaların ana sebebi ivme verilerindeki gürültüden kaynaklanmaktadır.



Şekil 4.20. LSTM hız tahmin grafiği

Şekil 4.21’de her bir saniyeye düşen RMSE değeri gösterilmiştir. Özellikle bitiş noktalarında ciddi kaymalar gözlemlenmiştir.



Şekil 4.21. LSTM hız kestirimi RMSE değerleri grafiği

4.3.2. BLSTM algoritması ile alınan benzetim sonuçları

LSTM algoritmasında olduğu gibi, BLSTM yöntemi de eğitim ve test setine gereksinim duyar. LSTM modelinden elde edilen başarı ile kıyaslama yapabilmek adına, LSTM algoritmasında kullanılan veri seti ve parametreleri tercih edilmiştir. LSTM derin öğrenme ağına uygulandığı gibi mevcut üç optimize ediciye dört farklı parametre durumu için hız kestirimi yaptırılmış ve sonuçlar tablolastırılmıştır. Alınan çıktılar Tablo 4.2’de verilirken en düşük RMSE değeri, birinci parametre durumuyla SGDM optimize edici gerçekleştirmiştir. Ancak birinci parametre durumundaki Adam optimize edicide SGDM hız kestirimine yakın bir değer almasına rağmen LSTM derin öğrenme ağına kıyasla daha uzun eğitim süresi gerektirdiğinden dolayı LSTM derin öğrenme ağı daha avantajlıdır. BLSTM derin öğrenme ağının bir başka dikkat çeken yönü ise parametre değerleri güncelleme almadığı zaman genelde daha başarılı sonuçlara ulaşmasıdır.

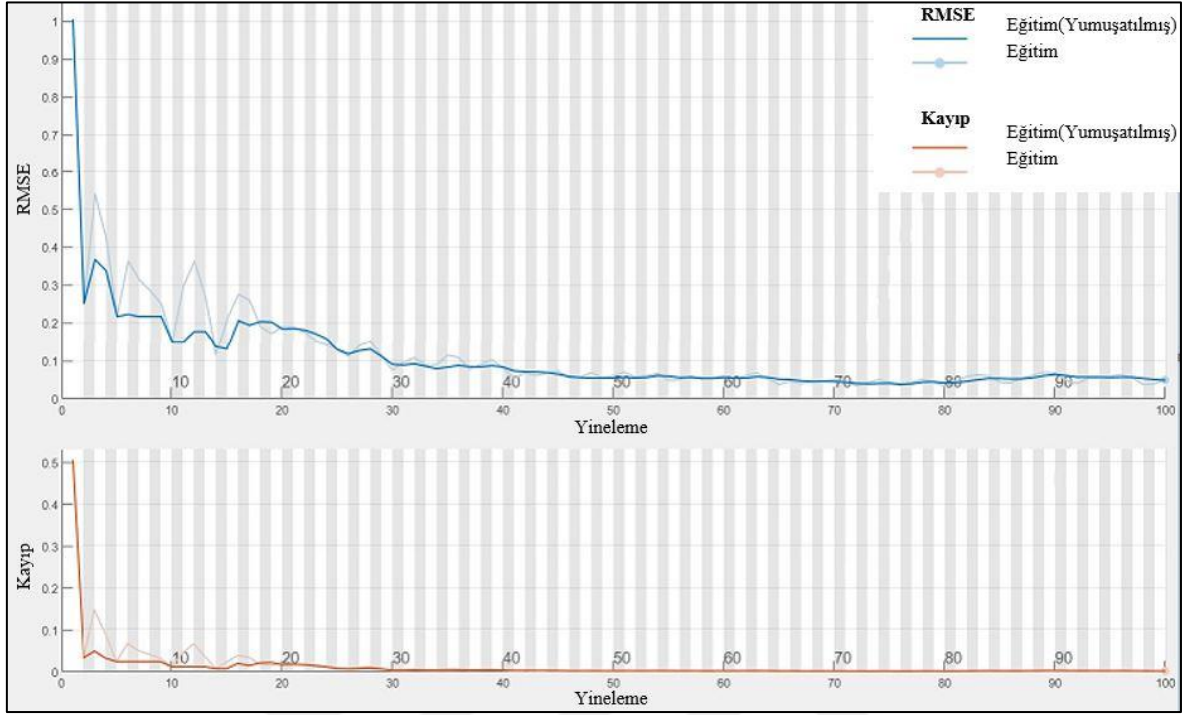
Tablo 4.2. BLSTM derin öğrenme ağı hız kestirim sonuçları

Optimize Edici	Parametre Durumu	Parametre	Değer	RMSE
Adam	1	LDP	2	2,9296
		LDF	0,99	
	2	LDP	5	2,7709
		LDF	0,89	
	3	LDP	10	3,0256
		LDF	0,5	
	4	LDP	none	2,7812
		LDF	none	
RMSProp	1	LDP	2	2,84423
		LDF	0,99	
	2	LDP	5	4,08124
		LDF	0,89	
	3	LDP	10	3,6455
		LDF	0,5	
	4	LDP	none	2,82871
		LDF	none	
SGDM	1	LDP	2	2,7592
		LDF	0,99	
	2	LDP	5	4,3457
		LDF	0,89	
	3	LDP	10	5,81412
		LDF	0,5	
	4	LDP	none	2,9105
		LDF	none	

BLSTM derin öğrenme ağı ile hız kestirimi sırasında, LSTM derin öğrenme ağında olduğu gibi üç farklı optimize edici kullanılmıştır. Bu optimize ediciler ile elde edilen eğitim süreci grafikleri aşağıda açıklanmıştır.

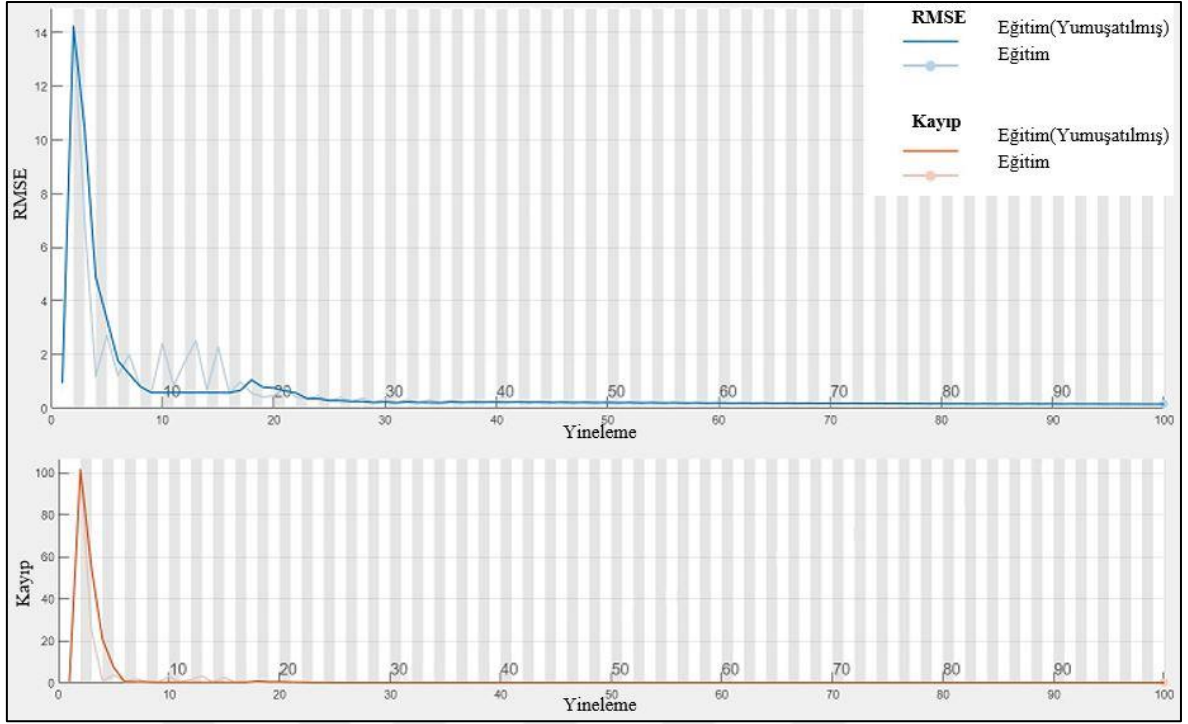
İlk olarak BLSTM derin öğrenme ağında kullanılan Adam optimize edicinin eğitim grafiği Şekil 4.22’de verilmiştir. Mavi çizgi ile gösterilen RMSE değerlerine bakıldığında, 1 değerinden başlayarak 10 yinleme adımında 0,2 değerinin altına düşmüştür. Grafiğin genelinde ise RMSE değerlerinin stabil hale gelmesi sırasında dalgalanmalar yaşanmıştır. Turuncu çizgi ile gösterilen kayıp değerleri incelendiğinde 2. yinleme adımından sonra ciddi kayıplar yaşanmamıştır. Adam optimize edici, genel olarak başarılı bir eğitim süreci

geçmesini sağlamıştır. Tablo 4.2’de Adam optimize ediciye ait değerler incelendiğinde genel olarak diğer optimize edicilere göre daha iyi değerler elde edildiği görülmektedir.



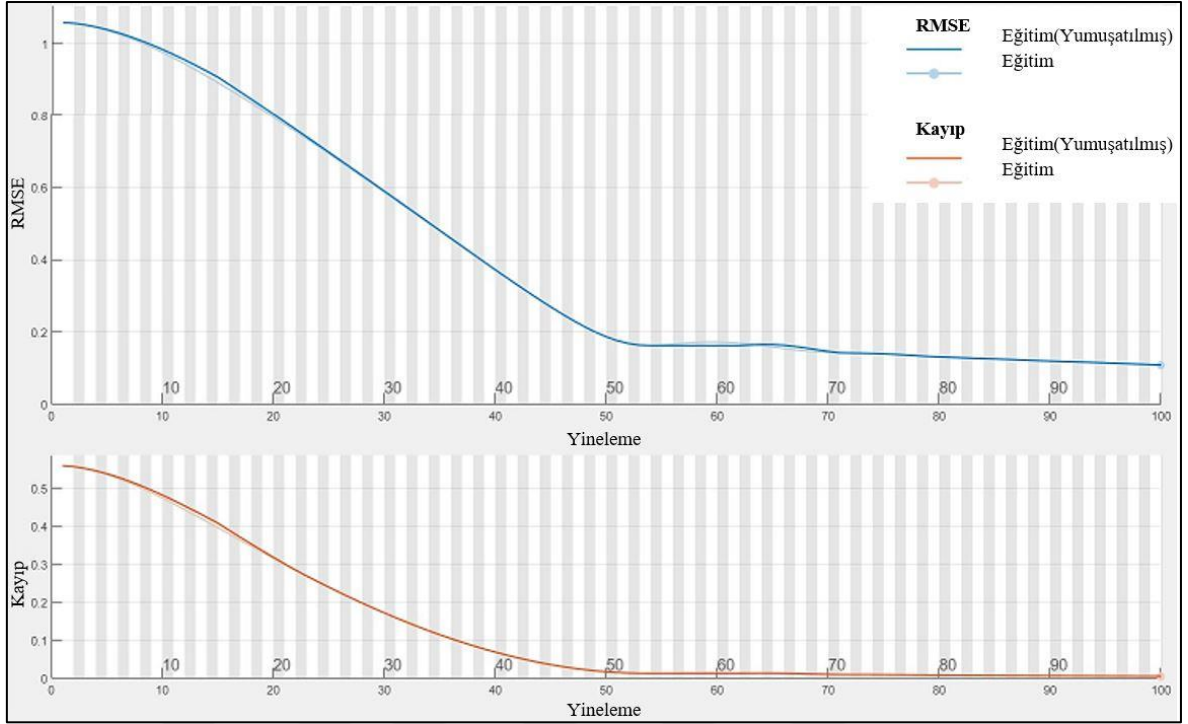
Şekil 4.22. Adam optimize edici eğitim süreci grafiği

İkinci olarak BLSTM derin öğrenme ağında kullanılan RMSprop optimize edicinin eğitim grafiği Şekil 4.23’te verilmiştir. Mavi çizgi ile gösterilen RMSE değerlerine bakıldığında 1’den başlamış ve 14’e kadar yükselmiştir. Turuncu çizgi ile gösterilen kayıp grafiği incelendiğinde, eğitim aşamasının ilk 5 adımında yüksek kayıplar yaşamış ve bu kayıplar hız kestirim sonuçlarına da yansımıştır. Tablo 4.2’de RMSprop optimize ediciye ait kestirim sonuçları da bu durumu destekler niteliktedir.



Şekil 4.23. RMSprop optimize edici eğitim süreci grafiği

Son olarak BLSTM derin öğrenme ağında kullanılan SGDM optimize edicinin eğitim grafiği Şekil 4.24’te verilmiştir. Mavi çizgi ile gösterilen RMSE değerleri incelendiğinde 50. yineleme adımına kadar RMSE değeri düşerek ilerlemiş ve ilerleyen yineleme adımlarında çok küçük bir dalgalanma yaparak düşük hatalı eğitim sürecine maruz kalmıştır. Turuncu çizgi ile gösterilen kayıp grafiğine de bakıldığında eğitim sürecindeki kayıpların minimuma inmesi uzun sürmüştür. Ancak yaşanan kayıplara bakıldığında, RMSprop optimize edicinin aksine 0,5 ile 0,01 arasında oldukça düşük değerlerdir.



Şekil 4.24. SGDM optimize edici eğitim süreci grafiği

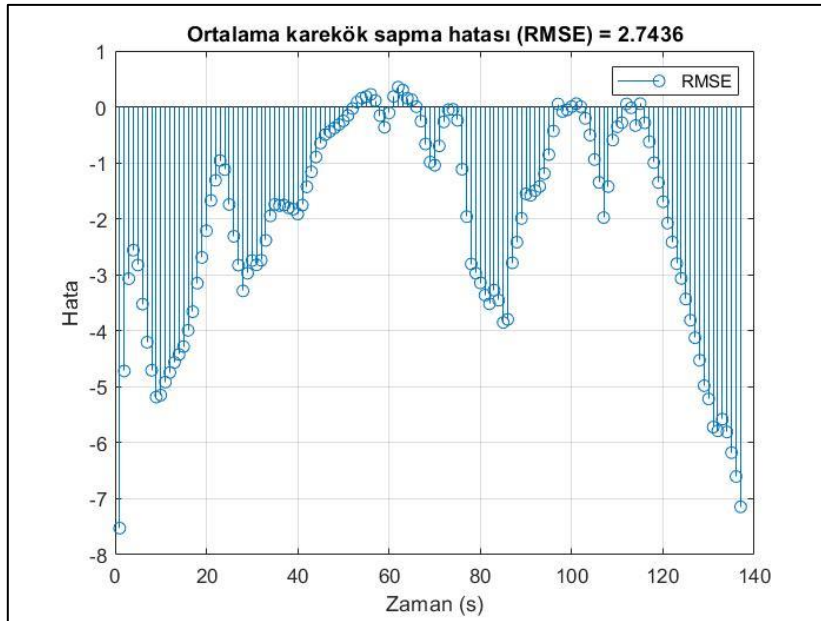
Genel olarak incelediğimiz üç optimize edicinin eğitim grafiği ortak olarak değerlendirildiğinde, eğitim sürecinde düşük kayıplar yaşayıp hızlı bir şekilde düşük kayıp seviyelerine düşen Adam optimize edici ve Adam optimize ediciye oranla daha yavaş stabil hale gelen SGDM optimize edici de genel anlamda başarılı sonuçlar elde etmiştir. Ancak RMSprop optimize edici derin öğrenme ağı eğitim aşamasında diğer optimize edicilere göre yüksek kayıplar yaşamıştır.

BLSTM derin öğrenme ağının hız kestirim RMSE değerleri ele alındığında en başarılı bulunan ve SGDM optimize edici ile yapılan birinci parametre durumu hız kestirim grafiği Şekil 4.25'te verilmiştir. GPS ile karşılaştırılan hız tahmin değerlerinde kayma gözlemlenmektedir. LSTM derin öğrenme ağı ile BLSTM derin öğrenme ağının RMSE değerleri genel olarak karşılaştırıldığında BLSTM, LSTM derin öğrenme ağına göre daha tutarsızdır ve BLSTM derin öğrenme ağının eğitim süresi daha uzun sürmektedir. Bundan dolayı bu tür kestirim problemlerinde BLSTM derin öğrenme ağının kullanılması mantıklı bulunmamıştır.



Şekil 4.25. BLSTM hız tahmin grafiği

BLSTM derin öğrenme ağının hız kestirim sonuçlarının her saniyedeki RMSE değerleri Şekil 4.26’da verilmiştir. Şekil 4.26 incelendiğinde özellikle başlangıç hız kestirimlerinde oldukça yüksek kaymalar mevcuttur. Aynı kaymalar bitiş noktasında da yüksek değerlere ulaşmıştır. Ancak hız kestirim grafiğinin orta noktalarında hata değerleri düşüktür. Genel anlamda 138 saniyelik hız kestiriminde başlangıç ve bitiş noktaları dışında başarılı sayılabilecek sonuçlara ulaşılmıştır.



Şekil 4.26. BLSTM hız kestirimi RMSE değerleri grafiği

4.3.3. GRU algoritması ile alınan benzetim sonuçları

Literatürde GRU, LSTM algoritmasına alternatif olarak sunulmuş, yapısal olarak LSTM'den farklılaştırılmış bir derin öğrenme ağı modelidir. Bu sebeple, tez çalışmalarında kullanılan LSTM ve BLSTM yöntemleri ile performans kıyaslaması yapılmak istenmiştir.

Önceki derin öğrenme ağ modellerinde kullanılan parametreler bu yapıda da kullanılmıştır. LSTM ve BLSTM derin öğrenme ağında kullanılan Adam, RMSprop ve SGDM optimize ediciler GRU sinir ağında da kullanılmıştır. Önceki derin öğrenme ağlarıyla RMSE değerlerini karşılaştırabilmek açısından parametre durumları da aynı seçilmiştir. Sonuçlar Tablo 4.3'te sunulmuştur.

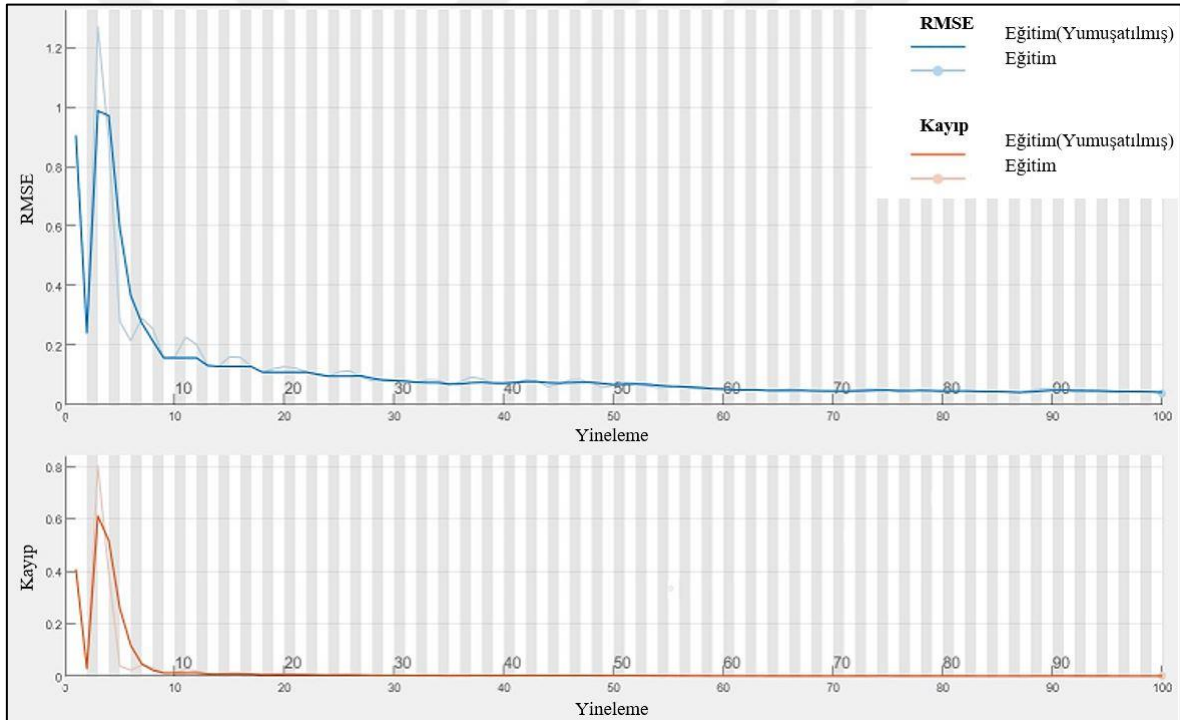
Tablo 4.3. GRU derin öğrenme ağı hız kestirim sonuçları

Optimize Edici	Parametre Durumu	Parametre	Değer	RMSE
Adam	1	LDP	2	2,81824
		LDF	0,99	
	2	LDP	5	2,626
		LDF	0,89	
	3	LDP	10	2,6324
		LDF	0,5	
	4	LDP	none	2,6847
		LDF	none	
	RMSProp	LDP	2	5,0164
		LDF	0,99	
		LDP	5	2,8642
		LDF	0,89	
		LDP	10	3,2396
		LDF	0,5	
		LDP	none	2,9462
		LDF	none	
SGDM	1	LDP	2	2,5414
		LDF	0,99	
	2	LDP	5	2,5576
		LDF	0,89	
	3	LDP	10	3,1982
		LDF	0,5	
	4	LDP	none	2,6726
		LDF	none	

En başarılı hız kestirim sonuçlarına SGDM optimize edici ile ulaşılmıştır. SGDM optimize edici genel anlamda başarılı hız kestirimleri yaparken birinci parametre durumunda en iyi RMSE değerine ulaşmıştır.

GRU derin öğrenme ağı ile hız kestirimi sırasında, LSTM ve BLSTM derin öğrenme ağlarındaki gibi üç farklı optimize edici kullanılmıştır. Bu optimize ediciler ile elde edilen eğitim süreci grafikleri aşağıda açıklanmıştır.

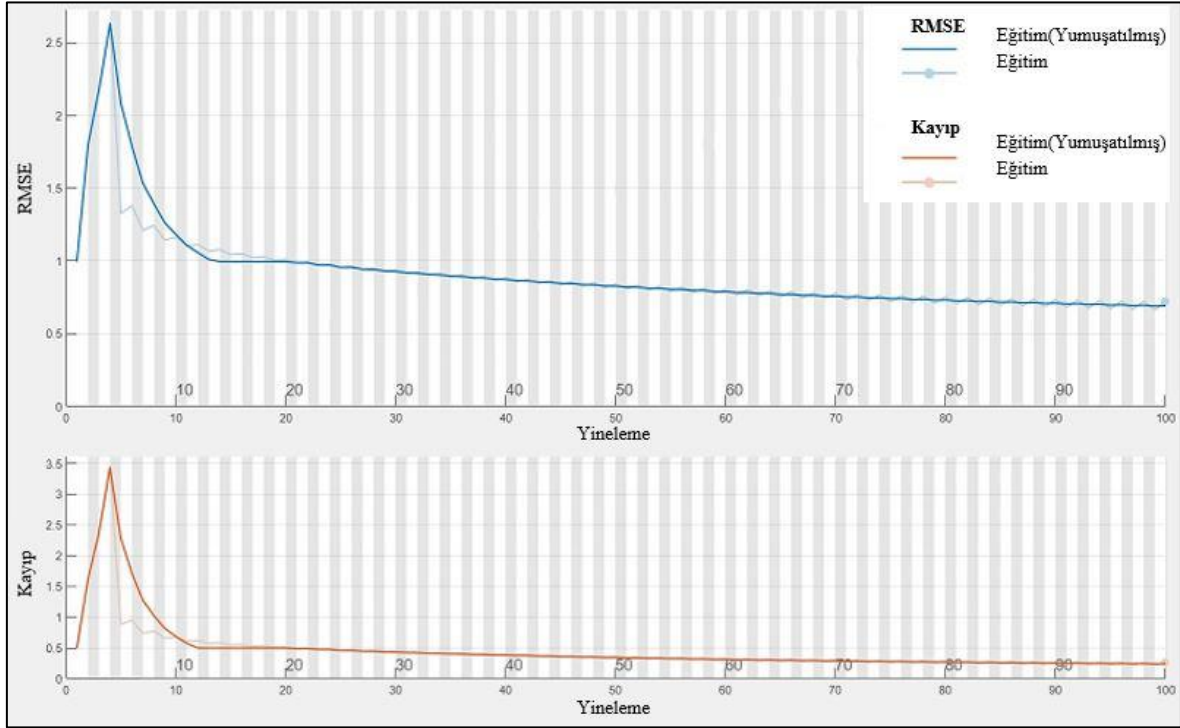
İlk olarak GRU derin öğrenme ağında kullanılan Adam optimize edicinin eğitim grafiği Şekil 4.27’de verilmiştir. Mavi çizgi ile gösterilen RMSE değerleri incelendiğinde 10. Yineleme adımımda en düşük RMSE değerini gören optimize edici olmuştur. Turuncu çizgi ile gösterilen kayıp grafiği de bunu destekler niteliktedir. Genel olarak bakıldığında LSTM ve BLSTM’de de başarılı eğitim süreci geçiren Adam optimize edicinin, bu tür zaman serisi kestirimlerinde kullanılabileceği sonucuna varılmıştır.



Şekil 4.27. Adam optimize edici eğitim süreci grafiği

İkinci olarak GRU derin öğrenme ağında kullanılan RMSprop optimize edicinin eğitim grafiği Şekil 4.28’de verilmiştir. Mavi çizgi ile gösterilen RMSE değerleri incelendiğinde 1 değerinden 2,7 değerlerine çıkmış ve RMSE olarak 0,5 değerini altına inememiştir. Turuncu çizgiler ile gösterilen kayıp grafiği de eğitim aşamasının sonuna kadar kayıpların sürdüğünü göstermektedir. Genel olarak RMSprop optimize edici LSTM ve BLSTM derin öğrenme ağlarında da eğitim süreci için çok başarılı sonuçlar verememiştir. Bu da göstermektedir ki

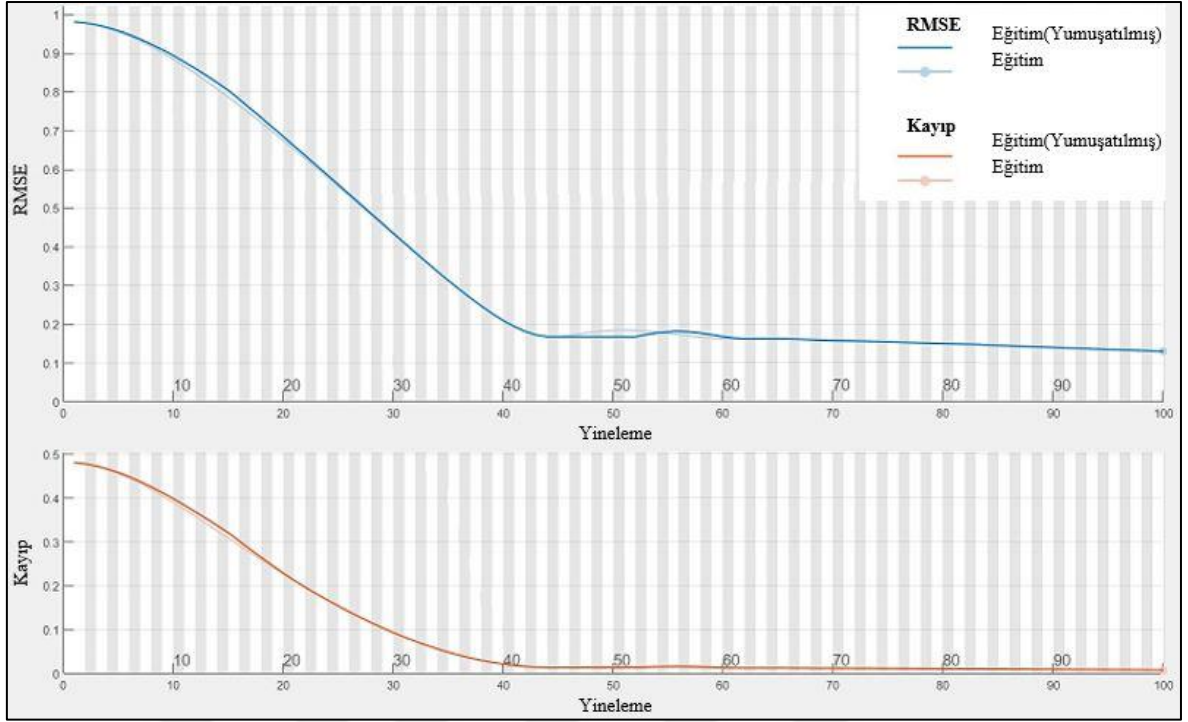
RMSprop optimize edici bu tür zaman serisi kestirimlerinde çok başarılı bir eğitim süreci geçirememiştir.



Şekil 4.28. RMSprop optimize edici eğitim süreci grafiği

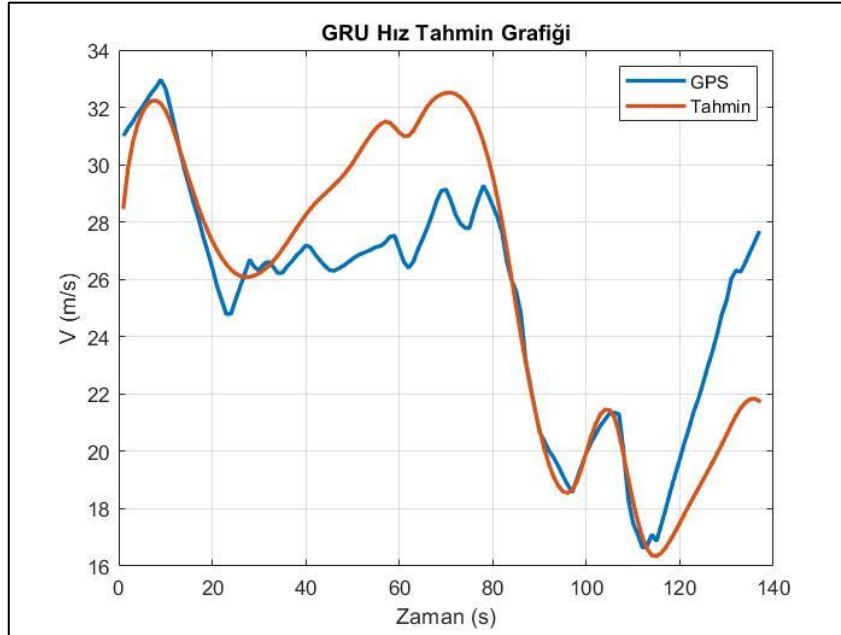
Son olarak GRU derin öğrenme ağında kullanılan SGDM optimize edicinin eğitim grafiği Şekil 4.29'da verilmiştir. Mavi çizgiler ile gösterilen RMSE değerlerine bakıldığında 1'den 0,2'ye kadar kararlı bir şekilde düşmektedir. İlerleyen yineleme adımlarında ufak bir dalgalanma yaşasa da turuncu renkle gösterilen kayıp grafiğinde görüldüğü gibi derin öğrenme ağının eğitimi sırasında düşük kayıplar yaşamıştır. Tablo 4.3'teki SGDM optimize edicinin RMSE çıktıları da bu durumu destekler niteliktedir.

Bu da göstermektedir ki LSTM, BLSTM ve GRU derin öğrenme ağlarında başarılı eğitim süreci geçiren SGDM optimize edici bu tür zaman serisi kestirimlerinde kullanılabilir bir yöntemdir. Daha önce eğitim grafikleri incelenen LSTM ve BLSTM derin öğrenme ağlarında da SGDM optimize edici RMSE değerleri Adam optimize edici RMSE değerlerine göre daha geç stabil hale gelmiştir. Bunun sebebi ise SGDM optimize edicinin olasılıksal yapısından dolayı ağırlık ayarlaması yaparken gürültü içermesinden kaynaklanmaktadır.



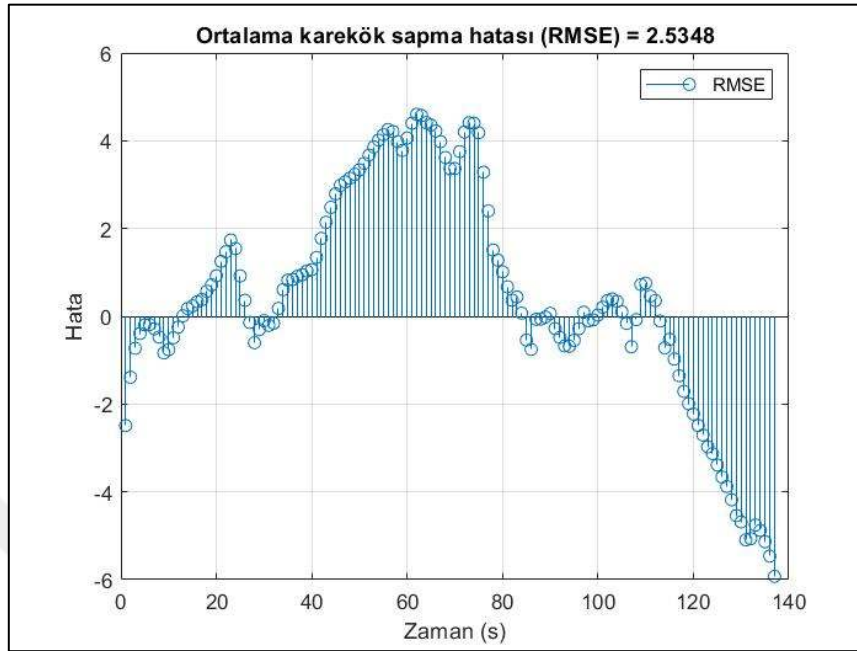
Şekil 4.29. SGDM optimize edici eğitim süreci grafiği

GRU derin öğrenme ağıyla elde edilen en iyi RMSE değerlerine sahip hız tahmin sonuçları Şekil 4.30'da verilmiştir. LSTM derin öğrenme ağı başlangıç durumlarında hız tahmin hatası yaparken GRU derin öğrenme ağı bu problemi ortadan kaldırmıştır. GRU derin öğrenme ağı 138 saniyelik hız tahmininin genelinde LSTM ve BLSTM derin öğrenme ağlarından da başarılı sonuçlar vermiştir. Ancak bitiş tahminlerinde hala kayma mevcuttur.



Şekil 4.30. GRU hız tahmin grafiği

GRU derin öğrenme ağının hız tahmin RMSE grafiği Şekil 4.31’de verilmiştir. Grafik incelendiğinde başlangıç noktasında başarılı hız tahminleri yapsa da zamanla kaymalar olmuştur. Ancak tahmin çıktıları genel anlamda GPS hız değerleri ile uyum içerisindedir.



Şekil 4.31. GRU derin öğrenme ağı RMSE grafiği

Derin öğrenme ağları ile yapılan hız kestirim çalışmaları derlendiğinde, 2,5414’lük RMSE değeri ile en başarılı kestirimi GRU derin öğrenme ağı yapmıştır. LSTM ve BLSTM ise sırasıyla 2,5547 ve 2,7592’lik RMSE değerleri ile hız kestirim sonuçları vermiştir. Bu sonuçların yanı sıra derin öğrenme ağlarının eğitimi esnasında harcanan süreler göz önünde bulundurulduğunda en fazla eğitim süresi BLSTM derin öğrenme ağı için geçmiştir. BLSTM derin öğrenme ağını da sırasıyla LSTM ve GRU derin öğrenme ağları takip etmiştir. Tablo 4.4’te derin öğrenme ağlarına ait en başarılı kestirim sonuçları verilmiştir.

Tablo 4.4. Derin öğrenme ağlarına ait en başarılı kestirim sonuçları

Kestirim Yöntemi	RMSE
LSTM Derin Öğrenme Ağı	2,5547
BLSTM Derin Öğrenme Ağı	2,7592
GRU Derin Öğrenme Ağı	2,5414

5. SONUÇLAR VE ÖNERİLER

Bu tez kapsamında GPS gibi küresel konumla sistemlerinin sinyal kaybı gibi senaryolarda ortaya çıkabilecek mağduriyetlere alternatif bir yol olan Ataletsel Navigasyon Sistemleri (Inertial Navigation System, INS) için geleneksel kestirim yöntemleri ve son zamanlarda popülaritesini artıran ve kullanım alanı gün geçtikçe genişleyen derin öğrenme tabanlı yöntemlerle hız kestirimi yapılarak, kestirim yöntemlerinin başarılarının kıyaslanması amaçlanmıştır. Çalışmada bir otomobilin hız kestirimi yapılmıştır ancak sistem modeli değiştirilerek yapılan çalışma bir hava aracı ya da bir yayanın yürüyüş hız kestirimi için de kullanılabilir. Çalışmada kullanılan veri seti, uygun ölçüm ortamı sağlanarak bir akıllı telefon ile MATLAB mobil programı çalıştırılarak akıllı telefonun sensörlerinden bir otomobile ait ivme, dönü ve GPS verileri elde edilmiştir. Veri seti oluşturulduktan sonra, verileri gürültüden olabildiğince arındırmak için filtreden geçirilmiş ve yer çekimine maruz kalan ivme verileri, rotasyon matrisleri yardımıyla yer çekiminden de arındırılmıştır. Kestirim algoritmaları için kullanılabilir hale getirilen bu veriler ile otomobilin hız kestirimi yapılmıştır.

Yapılan analizlerde geleneksel kestirim yöntemlerinden Kalman filtresi, sensör verilerinde bulunan çok küçük hatalardan bile etkilenmektedir ve bu hatalar birikmeli olarak hız kestirim sonuçlarında büyük kaymalara sebep olabilmektedir. Başka bir geleneksel kestirim yöntemi olan Luenberger gözlemcisinde ise tasarlanan sistem modeli, kontrol edilebilir olsa da gözlenebilir olmadığından dolayı kullanılamamıştır. Bu analiz sonuçlarına bakıldığında Kalman Filtresinin sensör gürültülerine karşı çok hassas olduğu ve hareketli ortalama alma yöntemi ile sabit kaymalar ivme verilerinden çıkartılarak Kalman Filtresi ile daha başarılı sonuçlar alınmıştır. Ancak, geleneksel yöntemlerden olan Luenberger gözlemcisinin ise bu tür hız kestirim problemlerinde kullanılamayacağı anlaşılmıştır. Derin öğrenme tabanlı yapılan kestirim çalışmalarında LSTM, BLSTM ve GRU derin öğrenme ağları kullanılmıştır. Tahmin doğrulukları üzerine doğrudan etkisi olan derin öğrenme ağı parametreleri farklı koşullar altında sınanmıştır. Derin öğrenme ağlarında Adam, RMSprop ve SGDM optimize edicileri kullanılarak optimize edicilerin hız kestirim sonuçlarına etkisinin incelenmesi amaçlanmıştır. LSTM derin öğrenme ağı ile yapılan hız kestiriminde en iyi sonuç Adam optimize edici ile 2,5547'lik bir RMSE değeri elde edilmiştir. BLSTM derin öğrenme ağı ile yapılan kestirim sonucunda 2,7592'lik RMSE değeri ile en başarılı sonuç SGDM optimize edici ile elde edilmiştir. Son olarak GRU derin öğrenme ağı ile

yapılan kestirin sonuçlarında ise 2,5414'lük RMSE değeri ile yine SGDM optimize edici en başarılı sonuca ulaşmıştır. Yapılan analizler sonucunda görülmektedir ki hız kestiriminde en iyi kestirim sonuçlarını veren LSTM ve GRU derin öğrenme ağları kestirim problemleri için uygunken, BLSTM derin öğrenme ağları ile yapılan hız kestirimlerin de diğer derin öğrenme ağı yöntemlerine göre daha düşük başarı elde edilmiştir. Ek olarak derin öğrenme ağlarının eğitim süreçleri de göz önünde bulundurulduğunda eğitim süreci en uzun süren yöntem BLSTM derin öğrenme ağı olmuştur. Bundan dolayı BLSTM derin öğrenme ağı, hız kestirimi için uygun bulunmamıştır. Yapılan analizler, optimize ediciler açısından incelendiğinde SGDM ve Adam optimize ediciler ile en iyi kestirim sonuçları elde edilmiştir. Böylece, SGDM ve Adam optimize ediciler, kestirim problemleri için uygunken RMSprop ile doğru kestirim sonuçlarına ulaşmanın daha zor olduğu düşünülmektedir.

Gelecek çalışmalarda Kalman Filtresindeki hız kestirim kaymalarının Uyarlanabilir Kalman Filtresi (Adaptive Kalman Filter, AKF) ile giderilebileceği düşünülmekte ve Kalman filtresi ile derin öğrenme ağlarının entegre edilerek daha başarılı kestirim sonuçları elde edilebileceği düşünülmektedir.

KAYNAKLAR

- [1] H. Fourati, “Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 221–229, 2015, doi: 10.1109/TIM.2014.2335912.
- [2] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, “RNN-Aided Human Velocity Estimation from a Single IMU,” *Sensors*, vol. 20, no. 13, p. 3656, Jun. 2020, doi: 10.3390/s20133656.
- [3] M. Jonasson, Å. Rogenfelt, C. Lanfelt, J. Fredriksson, and M. Hassel, “Inertial Navigation and Position Uncertainty during a Blind Safe Stop of an Autonomous Vehicle,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4788–4802, 2020, doi: 10.1109/TVT.2020.2971667.
- [4] Y. Cho, H. Cho, and C. M. Kyung, “Accurate and Robust Walking Speed Estimation with Adaptive Regression Models for Wrist-Worn Devices,” *IEEE Sens. J.*, vol. 20, no. 18, pp. 10744–10755, 2020, doi: 10.1109/JSEN.2020.2994616.
- [5] J. Svacha, G. Loianno, and V. Kumar, “Inertial Yaw-Independent Velocity and Attitude Estimation for High-Speed Quadrotor Flight,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1109–1116, 2019, doi: 10.1109/LRA.2019.2894220.
- [6] S. C. Lee and S. K. Hong, “Velocity-aided attitude estimation for helicopter aircraft using microelectromechanical system inertial-measurement units,” *Sensors (Switzerland)*, vol. 16, no. 12, pp. 1–12, 2016, doi: 10.3390/s16122102.

- [7] M. Zrenner, S. Gradl, U. Jensen, M. Ullrich, and B. M. Eskofier, “Comparison of different algorithms for calculating velocity and stride length in running using inertial measurement units,” *Sensors (Switzerland)*, vol. 18, no. 12, 2018, doi: 10.3390/s18124194.
- [8] K. L. Wang and J. Xu, “A speed regression using acceleration data in a deep convolutional neural network,” *IEEE Access*, vol. 7, pp. 9351–9356, 2019, doi: 10.1109/ACCESS.2019.2890967.
- [9] H. Deng, U. Arif, Q. Fu, Z. Xi, Q. Quan, and K. Y. Cai, “Visual–inertial estimation of velocity for multicopters based on vision motion constraint,” *Rob. Auton. Syst.*, vol. 107, pp. 262–279, 2018, doi: 10.1016/j.robot.2018.06.010.
- [10] K. Hori *et al.*, “Inertial Measurement Unit-Based Estimation of Foot Trajectory for Clinical Gait Analysis,” *Front. Physiol.*, vol. 10, no. January, pp. 1–12, 2020, doi: 10.3389/fphys.2019.01530.
- [11] C. J. de Ruiter and J. H. van Dieën, “Stride and step length obtained with inertial measurement units during maximal sprint acceleration,” *Sports*, vol. 7, no. 9, 2019, doi: 10.3390/sports7090202.
- [12] G. Ariante, U. Papa, S. Ponte, and G. Del Core, “Velocity and attitude estimation of a small unmanned aircraft with micro pitot tube and inertial measurement unit (IMU),” *2020 IEEE Int. Work. Metrol. AeroSpace, Metroaerosp. 2020 - Proc.*, pp. 314–318, 2020, doi: 10.1109/MetroAeroSpace48742.2020.9160298.
- [13] S. Liu *et al.*, “Pedestrian indoor navigation using foot-mounted IMU with multi-sensor data fusion,” *Int. J. Model. Identif. Control*, vol. 30, no. 4, pp. 261–272, 2018, doi: 10.1504/IJMIC.2018.095833.

- [14] C. Gençoğlu and H. Gümüş, “Standing Handball Throwing Velocity Estimation with a Single Wrist-Mounted Inertial Sensor,” *Ann. Appl. Sport Sci.*, vol. 8, no. 1, pp. 2–8, 2020, doi: 10.29252/aassjournal.893.
- [15] R. H. Rogne, T. H. Bryne, T. I. Fossen, and T. A. Johansen, “On the Usage of Low-Cost MEMS Sensors, Strapdown Inertial Navigation, and Nonlinear Estimation Techniques in Dynamic Positioning,” *IEEE J. Ocean. Eng.*, vol. 46, no. 1, pp. 24–39, 2021, doi: 10.1109/JOE.2020.2967094.
- [16] M. Li *et al.*, “Radar-Updated Inertial Landing Navigation with Online Initialization,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3360–3374, 2020, doi: 10.1109/TAES.2020.2987405.
- [17] W. Y. Kim, H. Il Seo, and D. H. Seo, “Nine-Axis IMU-based Extended inertial odometry neural network,” *Expert Syst. Appl.*, vol. 178, no. April, p. 115075, 2021, doi: 10.1016/j.eswa.2021.115075.
- [18] E. Simonetti, E. Bergamini, G. Vannozzi, J. Bascou, and H. Pillet, “Estimation of 3d body center of mass acceleration and instantaneous velocity from a wearable inertial sensor network in transfemoral amputee gait: A case study,” *Sensors*, vol. 21, no. 9, 2021, doi: 10.3390/s21093129.
- [19] A. Jain, D. Kulemann, and S. Schon, “Improved velocity estimation in urban areas using Doppler observations,” *2021 Int. Conf. Localization GNSS, ICL-GNSS 2021 - Proc.*, 2021, doi: 10.1109/ICL-GNSS51451.2021.9452243.

- [20] Y. Mao, T. Ogata, H. Ora, N. Tanaka, and Y. Miyake, “Estimation of stride-by-stride spatial gait parameters using inertial measurement unit attached to the shank with inverted pendulum model,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–10, 2021, doi: 10.1038/s41598-021-81009-w.
- [21] A. M. Bassam, A. B. Phillips, S. R. Turnock, and P. A. Wilson, “Ship speed prediction based on machine learning for efficient shipping operation,” *Ocean Eng.*, vol. 245, no. August 2021, p. 110449, 2022, doi: 10.1016/j.oceaneng.2021.110449.
- [22] H. Acikgoz and D. Korkmaz, “Long Short-Term Memory Network-based Speed Estimation Model of an Asynchronous Motor,” *12th Int. Symp. Adv. Top. Electr. Eng. ATEE 2021*, 2021, doi: 10.1109/ATEE52255.2021.9425346.
- [23] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems, 2nd edition [Book review]*, vol. 30, no. 2. 2015.
- [24] A. Nouredin, T. B. Karamat, and J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration; Chapter 4 Inertial Navigation System*. 2013.
- [25] O. J. Woodman, “An introduction to inertial navigation,” Cambridge, 2007.
- [26] M. Kayton and W. R. Fried, *Avionics Navigation Systems*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1997.
- [27] B. E. Chan, “2017 Tech Notes Strapdown Inertial Navigation Systems,” 2017.

- [28] D. K. Shaeffer, “MEMS inertial sensors: A tutorial overview,” *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 100–109, 2013, doi: 10.1109/MCOM.2013.6495768.
- [29] N. Yazdi, K. Najafi, and F. Ayazi, “Micromachined Inertial Sensors,” Michigan, 1998.
- [30] S. S. Bhattacharyya, E. F. Deprettere, R. Leupers, and J. Takala, *Handbook of signal processing systems*. 2018.
- [31] J. A. Farrell, *Aided Navigation Systems: GPS and High Rate Sensors*. 2008.
- [32] Z. Mohammed, I. (Abe) M. Elfadel, and M. Rasras, “Monolithic multi degree of freedom (MDoF) capacitive MEMS accelerometers,” *Micromachines*, vol. 9, no. 11, pp. 1–20, 2018, doi: 10.3390/mi9110602.
- [33] M. Bao, *Analysis and Design Principles of MEMS Devices*, no. 1. Elsevier, 2005.
- [34] Refet Fırat Yazıcıoğlu, “Surface Micromachined Capacitive Accelerometers Using MEMS Technology,” Middle East Technical University, 2003.
- [35] M. A. Erişmiş, “MEMS accelerometers and gyroscopes for inertial measurement unit,” Middle East Technical University, 2004.
- [36] Z. Ghemari, “Study and analysis of the piezoresistive accelerometer stability and improvement of their performances,” *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, pp. 1520–1526, 2017, doi: 10.1007/s13198-017-0622-8.

- [37] S. Kavitha, R. J. Daniel, and K. Sumangala, “A simple analytical design approach based on computer aided analysis of bulk micromachined piezoresistive MEMS accelerometer for concrete SHM applications,” *Meas. J. Int. Meas. Confed.*, vol. 46, no. 9, pp. 3372–3388, 2013, doi: 10.1016/j.measurement.2013.05.013.
- [38] E. Balaban, A. Saxena, P. Bansal, K. F. Goebel, and S. Curran, “Modeling, detection, and disambiguation of sensor faults for aerospace applications,” *IEEE Sens. J.*, vol. 9, no. 12, pp. 1907–1917, 2009, doi: 10.1109/JSEN.2009.2030284.
- [39] S. K. Reguieg, Z. Ghemari, T. Benslimane, and S. Saad, “Modeling and Enhancement of Piezoelectric Accelerometer Relative Sensitivity,” *Sens. Imaging*, vol. 20, no. 1, 2019, doi: 10.1007/s11220-018-0222-y.
- [40] C. S. Silva, J. Pontes, J. C. Viana, L. A. Rocha, and J. Gaspar, “A fully integrated three-axis thermal accelerometer,” *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, pp. 963–966, 2013, doi: 10.1109/I2MTC.2013.6555558.
- [41] G. Krishnan, C. U. Kshirsagar, G. K. Ananthasuresh, and N. Bhat, “Micromachined High-Resolution Accelerometers,” no. July 2007, 2007.
- [42] E. Abbaspour-Sani, R. S. Huang, and C. Yee Kwok, “A linear electromagnetic accelerometer,” *Sensors Actuators A. Phys.*, vol. 44, no. 2, pp. 103–109, 1994, doi: 10.1016/0924-4247(94)00792-6.
- [43] C. B. W. Pedersen and A. A. Seshia, “On the optimization of compliant force amplifier mechanisms for surface micromachined resonant accelerometers,” *J. Micromechanics Microengineering*, vol. 14, no. 10, pp. 1281–1293, 2004, doi: 10.1088/0960-1317/14/10/001.

- [44] A. A. Seshia *et al.*, “A vacuum packaged surface micromachined resonant accelerometer,” *J. Microelectromechanical Syst.*, vol. 11, no. 6, pp. 784–793, 2002, doi: 10.1109/JMEMS.2002.805207.
- [45] V. Ferrari, A. Ghisla, D. Marioli, and A. Taroni, “Silicon resonant accelerometer with electronic compensation of input-output cross-talk,” *Sensors Actuators, A Phys.*, vol. 123–124, pp. 258–266, 2005, doi: 10.1016/j.sna.2005.03.067.
- [46] H. Xie, G. K. Fedder, and R. E. Sulouff, “Accelerometers,” *Compr. Microsystems*, vol. 2, pp. 135–180, 2007, doi: 10.1016/B978-044452190-3.00053-7.
- [47] C. Aszkler, “Acceleration, Shock and Vibration Sensors,” *Sens. Technol. Handb.*, pp. 137–159, 2005, doi: 10.1016/B978-075067729-5/50045-8.
- [48] R. Mukhiya *et al.*, “Design, modelling and system level simulations of DRIE-based MEMS differential capacitive accelerometer,” *Microsyst. Technol.*, vol. 25, no. 9, pp. 3521–3532, 2019, doi: 10.1007/s00542-018-04292-0.
- [49] C. L. Chu, C. H. Lin, and K. C. Fan, “Two-dimensional optical accelerometer based on commercial DVD pick-up head,” *Meas. Sci. Technol.*, vol. 18, no. 1, pp. 265–274, 2007, doi: 10.1088/0957-0233/18/1/033.
- [50] M. N. Armenise, C. Ciminelli, F. Dell’Olio, and V. M. N. Passaro, *Advances in Gyroscope Technologies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

- [51] W. Case, “The gyroscope: An elementary discussion of a child’s toy,” *Am. J. Phys.*, vol. 45, no. 11, pp. 1107–1109, Nov. 1977, doi: 10.1119/1.10957.
- [52] Z. Wang, C. Hu, W. Wang, X. Si, and Z. Zhou, “An off-online fuzzy modelling method for fault prognosis with an application,” in *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, May 2012, no. May, pp. 1–7, doi: 10.1109/PHM.2012.6228783.
- [53] D. Titterton and J. Weston, “Strapdown inertial navigation technology - 2nd edition - [Book review,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 20, no. 7, pp. 33–34, Jul. 2005, doi: 10.1109/MAES.2005.1499250.
- [54] V. M. N. Passaro, A. Cuccovillo, L. Vaiani, M. De Carlo, and C. E. Campanella, “Gyroscope Technology and Applications: A Review in the Industrial Perspective,” *Sensors*, vol. 17, no. 10, p. 2284, Oct. 2017, doi: 10.3390/s17102284.
- [55] G. T. Schmidt, “Navigation sensors and systems in GNSS degraded and denied environments,” *Chinese J. Aeronaut.*, vol. 28, no. 1, pp. 1–10, Feb. 2015, doi: 10.1016/j.cja.2014.12.001.
- [56] H. M. Abdelzaher *et al.*, “Noise reduction in optical gyroscope signals based on hybrid approaches,” *J. Opt.*, vol. 51, no. 1, pp. 5–21, Mar. 2022, doi: 10.1007/s12596-020-00617-3.
- [57] R. Ma, H. Yu, K. Ni, Q. Zhou, X. Li, and G. Wu, “Angular velocity measurement system based on optical frequency comb,” in *Semiconductor Lasers and Applications IX*, Nov. 2019, no. June 2021, p. 36, doi: 10.1117/12.2537826.

- [58] N. B. Phillips, I. Novikova, E. E. Mikhailov, D. Budker, and S. Rochester, “Controllable steep dispersion with gain in a four-level N -scheme with four-wave mixing,” *J. Mod. Opt.*, vol. 60, no. 1, pp. 64–72, Jan. 2013, doi: 10.1080/09500340.2012.733433.
- [59] R. Antonello and R. Oboe, “MEMS Gyroscopes for Consumers and Industrial Applications,” in *Microsensors*, no. June, InTech, 2011.
- [60] J. Mate, “Improvement of the calibration, test and verification process of an interferometric fiber optic gyroscope Bachelor’s thesis in mechatronics engineering,” Chalmers University of Technology, 2020.
- [61] M. N. Armenise, C. Ciminelli, F. De Leonardis, R. Diana, V. Passaro, and F. Peluso, “Gyroscope technologies for space applications,” *Proc. 4th Round Table Micro/Nano Technol. Sp.*, pp. 1–26, 2003.
- [62] S. Nasiri and S. Clara, “A Critical Review of MEMS Gyroscopes Technology and Commercialization Status,” *Sensors (Peterborough, NH)*, p. 8, 2009, [Online]. Available: invensense.com.
- [63] E. W. McAllister, “Measurement,” in *Pipeline Rules of Thumb Handbook*, Elsevier, 2014, pp. 541–624.
- [64] H. Zeng and Y. Zhao, “Sensing Movement: Microsensors for Body Motion Measurement,” *Sensors*, vol. 11, no. 1, pp. 638–660, Jan. 2011, doi: 10.3390/s110100638.
- [65] A. M. Shkel, “Type I and Type II Micromachined Vibratory Gyroscopes,” in *2006 IEEE/ION Position, Location, And Navigation Symposium*, 2006, vol. 2006, no. January 2006, pp. 586–593, doi: 10.1109/PLANS.2006.1650648.

- [66] M. Reze and M. Osajda, “MEMS sensors for automotive vehicle stability control applications,” in *MEMS for Automotive and Aerospace Applications*, Elsevier, 2013, pp. 29–53.
- [67] M. Ben-Ari and F. Mondada, “Robotic Motion and Odometry,” in *Elements of Robotics*, Cham: Springer International Publishing, 2018, pp. 63–93.
- [68] H. Dong, “Design and Analysis of a MEMS Comb Vibratory Gyroscope,” University of Bridgeport, 2009.
- [69] B. Fan, A. Chen, W. Hu, F. Zheng, D. Xu, and S. Guo, “Investigation of air damping in different vibrational modes of lateral-axis gyroscopes,” in *2017 IEEE SENSORS*, Oct. 2017, vol. 2017-Decem, no. August, pp. 1–3, doi: 10.1109/ICSENS.2017.8234141.
- [70] B. Moaveni, M. Khosravi Roqaye Abad, and S. Nasiri, “Vehicle longitudinal velocity estimation during the braking process using unknown input Kalman filter,” *Veh. Syst. Dyn.*, vol. 53, no. 10, pp. 1373–1392, Oct. 2015, doi: 10.1080/00423114.2015.1038279.
- [71] F. Sandhu, H. Selamat, S. E. Alavi, and V. Behtaji Siahkal Mahalleh, “FPGA-Based Implementation of Kalman Filter for Real-Time Estimation of Tire Velocity and Acceleration,” *IEEE Sens. J.*, vol. 17, no. 17, pp. 5749–5758, Sep. 2017, doi: 10.1109/JSEN.2017.2726529.
- [72] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

- [73] M. Musiol, “Speeding up Deep Learning Computational Aspects of Machine Learning Deep Learning in Flight Phase Recognition View project Speeding up Deep Learning Computational Aspects of Machine Learning,” no. January, 2016, [Online]. Available: [researchgate.net/publication/308414212](https://www.researchgate.net/publication/308414212).
- [74] I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning: Machine Learning Book,” p. 785, 2016, [Online]. Available: deeplearningbook.org/.
- [75] C. Yi and J. Cho, “Sensor Fusion for Accurate Ego-Motion Estimation in a Moving Platform,” *Int. J. Distrib. Sens. Networks*, vol. 2015, pp. 1–6, 2015, doi: 10.1155/2015/831780.
- [76] A. Poulouse, O. S. Eyobu, and D. S. Han, “An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data,” *IEEE Access*, vol. 7, pp. 11165–11177, 2019, doi: 10.1109/ACCESS.2019.2891942.
- [77] G. Ellis, “The Luenberger Observer: Correcting Sensor Problems,” in *Observers in Control Systems*, Elsevier, 2002, pp. 67–96.
- [78] I. O. Rataichuk and V. I. Kortunov, “INS errors compensation algorithm based on Luenberger Observer,” in *2013 IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD)*, Oct. 2013, pp. 126–128, doi: 10.1109/APUAVD.2013.6705303.
- [79] M. S. Mahmoud, “Advanced Control Design with Application to Electromechanical Systems,” in *Advanced Control Design with Application to Electromechanical Systems*, Elsevier, 2018, pp. 1–41.
- [80] B. C. Kuo, *Otomatik Kontrol Sistemleri*. Literatür Yayınları, 2006.

- [81] J. Klamka, “Controllability of dynamical systems. A survey,” *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 61, no. 2, pp. 335–342, Jun. 2013, doi: 10.2478/bpasts-2013-0031.
- [82] Y.-P. Chen, “19. Linear Luenberger Observer Design.”
- [83] A. Bemporad, “Automatic Control-State estimation and linear observers,” *Course PPT*, pp. 1–13, 2011.
- [84] Y. Kim and H. Bang, “Introduction to Kalman Filter and Its Applications,” in *Introduction and Implementations of the Kalman Filter*, IntechOpen, 2019, pp. 1–16.
- [85] L. A. McGee and S. F. Schmidt, “Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry,” *NASA Tech. Memo.*, no. November, p. 21, 1985, [Online]. Available: ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19860003843_1986003843.pdf.
- [86] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” 1997.
- [87] M. Laaraiedh, “Implementation of Kalman Filter with Python Language,” no. 3, pp. 1–5, Apr. 2012, [Online]. Available: arxiv.org/abs/1204.0375.
- [88] J. E. Carrion and B. F. Spencer, “Model-based Strategies for Real-time Hybrid Testing,” *NSEL Rep. Ser. Rep. 6*, no. January, p. 211, 2007.
- [89] C. Kasnakoğlu, “Kalman Filtresi Uygulamaları,” 2014.
- [90] J. Wang, M. Xue, R. Culhane, E. Diao, J. Ding, and V. Tarokh, “Speech emotion recognition with dual-sequence LSTM architecture,” *ICASSP 2020 - 2020 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 6469–6473, 2020.

- [91] L. Wang, X. Xu, H. Dong, R. Gui, R. Yang, and F. Pu, “Exploring convolutional LSTM for polsar image classification,” pp. 8452–8455, 2018.
- [92] Q. Zhan, L. Zhang, H. Deng, and X. Xie, “An Improved LSTM For Language Identification,” in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Aug. 2018, vol. 2018-Augus, pp. 609–612, doi: 10.1109/ICSP.2018.8652445.
- [93] O. Akköse, “Uzun-Kısa Vadeli Bellek(LSTM),” 2022. medium.com/deep-learning-turkiye/uzun-kısa-vadeli-bellek-lstm-b018c07174a3.
- [94] İ. Dölek, “LSTM,” 2018. ishakdolek.medium.com/lstm-d2c281b92aac.
- [95] N. Çarkacı, “Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler,” 2018. [medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4#:~:text=Bu problemi çözmek için%3B veri,mini-batch” olarak adlandırılmaktadır](https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4#:~:text=Bu%20problemi%20çözmek%20için%3B%20veri,mini-batch%20olarak%20adlandırılmaktadır.).
- [96] “Stochastic gradient descent.” peltarion.com/knowledge-center/documentation/modeling-view/run-a-model/optimizers/stochastic-gradient-descent.
- [97] R. Sanghvi, “A Complete Guide to Adam and RMSprop Optimizer,” 2021. medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be.
- [98] “Stochastic Gradient Descent (SGD).” setscholars.net/machine-learning-mastery-stochastic-gradient-descent-sgd/.

- [99] K. A. Carpenter, D. S. Cohen, J. T. Jarrell, and X. Huang, “Deep learning and virtual drug screening,” *Future Med. Chem.*, vol. 10, no. 21, pp. 2557–2567, Nov. 2018, doi: 10.4155/fmc-2018-0314.
- [100] V. Buşaeu, “Stochastic Gradient Descent with momentum,” 2017. towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d.
- [101] V. Bushaev, “Adam — latest trends in deep learning optimization.” towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c.
- [102] “TrainingOptionsADAM.” mathworks.com/help/deeplearning/ref/nnet.cnn.trainingoptionsadam.html.
- [103] “trainingOptions.” mathworks.com/help/deeplearning/ref/trainingoptions.html.
- [104] “lstmLayer.” mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstm.html;jsessionid=fb8d0942481364b3e28e3863d9bc.
- [105] “sequenceInputLayer.” mathworks.com/help/deeplearning/ref/nnet.cnn.layer.sequenceinputlayer.html.
- [106] “fullyConnectedLayer.” mathworks.com/help/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html.
- [107] B. Babüroğlu, A. Tekerek, and M. Tekerek, “Türkçe için Derin Öğrenme Tabanlı Doğal Dil İşleme Modeli Geliştirilmesi,” p. 8, 2019.

- [108]Ö. Yildirim, “A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification,” *Comput. Biol. Med.*, vol. 96, no. March, pp. 189–202, May 2018, doi: 10.1016/j.compbimed.2018.03.016.
- [109]A. T. Mohan and D. V. Gaitonde, “A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks,” no. April, Apr. 2018, [Online]. Available: arxiv.org/abs/1804.09269.
- [110]S. Kostadinov, “Understanding GRU Networks.” towardsdatascience.com/understanding-gru-networks-2ef37df6c9be.
- [111]R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nov. 2016, no. November 2016, pp. 324–328, doi: 10.1109/YAC.2016.7804912.
- [112]P. T. Yamak, L. Yujian, and P. K. Gadosey, “A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting,” in *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, Dec. 2019, pp. 49–55, doi: 10.1145/3377713.3377722.
- [113]C. Hu, T. Ou, H. Chang, Z. Yu, and L. Zhu, “Deep GRU Neural-Network Prediction and Feedforward Compensation for Precision Multi-Axis Motion Control Systems,” *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1–1, 2020, doi: 10.1109/TMECH.2020.2975343.