T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SENSÖR FÜZYONU ALGORİTMALARI İLE AÇISAL KONUM REFERANS SİSTEMİ TASARIMI

OBEN ÖZKAN

YÜKSEK LİSANS TEZİ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI
MEKATRONİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
DOÇ. DR. AYDIN YEŞİLDİREK

İSTANBUL, 2019

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SENSÖR FÜZYONU ALGORİTMALARI İLE AÇISAL KONUM REFERANS SİSTEMİ TASARIMI

Oben ÖZKAN tarafından hazırlanan tez çalışması 11.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç.Dr.Aydın YEŞİLDİREK

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç.Dr.Aydın YEŞİLDİREK

Yıldız Teknik Üniversitesi _____

Prof. Dr. Fikret ÇALIŞKAN

İstanbul Teknik Üniversitesi _____

Dr.Öğr.Üyesi Mehmet Selçuk ARSLAN

Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Bu çalışmada, açısal konum referans sistemlerinde kullanılan sensörler ve sensör bütünleştirme algoritmaları incelenmiş ve bu sistem kullanılarak bir yayanın adım takibi uygulaması için sensör hatalarını azaltan bir filtre tasarımı verilmiştir.

Açısal konum referans sistemleri ataletsel seyrüsefer sisteminin ayrılmaz bir parçasıdır bu sistemler içerisinde bulundurduğu ataletsel sensörler sayesinde dünya eksenine göre açısal konumun elde edilmesini sağlarlar. Hava, uzay, kara, denizaltı araçlarında, akıllı mühimmatlarda, pilot kasklarında ve oyun konsollarında açısal konum bilgisi açısal konum referans sistemleri sayesinde elde edilirler.

Çalışmamda öncelikle açısal konum referans sistemini oluşturan sensörleri inceledim, bu sensörlerden elde edilen verileri uygun bir sensör bütünleştirme algoritması ile bütünleştirdim ve bir açısal konum referans sistemi elde ettim. Daha sonra tezimin asıl amacı olan yaya adım takibi yapabilen bir ataletsel konum bulma sistemi için özgün hata giderme filtresi tasarlayıp sistemimi gerçekledim.

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

Temmuz, 2019

Oben ÖZKAN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÇİZELGE LİSTESİ x

ÖZET xi

ABSTRACT xiii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 4
1.3 Hipotez 4

BÖLÜM 2

AÇISAL KONUM REFERANS SİSTEMİ 6

2.1 Ataletsel Ölçüm Sensörleri 6
2.2 Rotasyon Matrisinin Matematiksel Çıkarımı 7
2.2.1 DCM (Rotasyon) Matrisi Türevinin Matematiksel Çıkarımı 9
2.2.2 Euler Açıları 10
2.2.3 Kuaterniyon Gösterimi 11
2.3 Sensörler 12
2.4 Hata Modellemesi 13
2.4.1 İvmeölçer Sensörü 13
2.4.2 Jiroskop Sensörü 14
2.4.3 Manyetometre 15
2.4.4 AHRS Birimi 16

BÖLÜM 3

SENSÖR FÜZYONU ALGORİTMALARI 21

3.1 Kalman Filtresi 21
3.2 Bütünleyici Filtre 23
3.2.1 Vektör Gözlemcisi ile Oryantasyon 25
3.2.2 Filtre Füzyon Algoritması 29
3.2.3 Manyetik Bozulma Telafisi 30
3.2.4 Jiroskop Sapması Telafisi 31
3.2.5 Filtre Kazancı 33

BÖLÜM 4

İÇ ORTAM SEYRÜSEFER UYGULAMASI 34

4.1 Sıfır Hız Algılama(SHA) 34
4.2 Uygulama 35
4.3 Heuristic Drift Elimination (HDE) 41
4.4 Sezgisel Kayma Eliminasyonu (SKE) 42
4.4.1 Yaya Hareket Algılaması 43
4.4.2 Baskın Yön Hesaplaması 43
4.4.3 Pozisyon Güncelleme 46
4.5 Pozisyon Tahmin Sistemine Genel Bakış 53

BÖLÜM 5

SONUÇ VE ÖNERİLER 55

KAYNAKLAR 57

EK-A

AHRS GÖMÜLÜ C++ KODU 60

EK-B

BÜTÜNLEYİCİ FİLTRE + SKE + SHA MATLAB KODU 65

ÖZGEÇMİŞ 74

SİMGE LİSTESİ

ω	Açısal dönüş vektörü
ω_{en}^{n}	Navigasyon ekseninin dünya eksenine göre açısal dönüş hızının araç eksenine izdüşüm vektörü
Ω	Skew-simetrik açısal hız matrisi
Ω_{ib}^{b}	Gövde eksen takımına bağlı jiroskopların ataletsel eksen takımına göre ölçtükleri açısal hız vektörlerinden oluşan skew-simetrik açısal hız matrisi
Ω_{ib}^{b}	Navigasyon ekseninin ataletsel eksen takımına göre olan ve gövde eksen takımında ifade edilen açısal hız vektörünün skew-simetrik gösterimidir
R_{n}^{b}	Navigasyon ekseninin ataletsel eksen takımına göre rotasyon matrisi
R_{b}^{n}	Ataletsel eksen takımının navigasyon eksenine göre rotasyon matrisi
g	Yerçekim ivmesi
θ	Yunuslama(pitch) açısı
Φ	Yuvarlanma (roll) açısı
Ψ	Savrulma(yaw) açısı
q	Quaterniyon vektörü
$\otimes$	Quaterniyon vektör çarpım operatörü
∇	Gradyan iniş operatörü
β	Filtre kazanç katsayısı
ζ	Filtre kazanç katsayısı
∂	Türev operatörü
γ_t	Sensör çerçevesinin dünya çerçevesine göre tahmini yönelimine uygulanan katsayı
μ_t	Yakınsama hızını sağlayan değer
α	İvmeölçer ve manyetometre ölçümlerinde gürültüyü hesaba katan μ_t'nin katsayısı

KISALTMA LİSTESİ

AHRS	Attitude and Heading Referance System
CF	Complementary Filter
DCM	Driect Direction Cosine Matrix
EHDE	Enhanced Heuristic Drift Elimination
EKF	Extended kalman filter
GPS	Global positioning system
HDE	Heuristic Drift Elimination
INS	Inertial navigation system
KF	Kalman Filter
MEMS	Micro-Electro-Mechanical Systems
NED	Kuzey doğu aşağı ekseni
SHA	Sıfır Hız Algılama
SKE	Sezgisel Kayma Eliminasyonu
ZUPT	Zero Update Velocity

ŞEKİL LİSTESİ

Sayfa

Şekil 2. 1 Navigasyon ekseni ile araç ekseni arası ilişki [18] 7
Şekil 2. 2 Sapma, yunuslama ve yuvarlanma açılarının rotasyonal etkisi [18]..... 7
Şekil 2. 3 Euler açıları 10
Şekil 2. 4 Kuaternion gösterimi 12
Şekil 2. 5 Solda LEV-3 sağda ST-124-M3 AHRS birimleri 17
Şekil 2. 6 Bir halka lazer dönüölçeri ve çalışma mantığı 18
Şekil 2. 7 Northrop Grumman LN-2000 AHRS 18
Şekil 2. 8 Çalışmada kullanılan AHRS şematiği 20
Şekil 3. 1 Kalman filtresi algoritması 22
Şekil 3. 2 Lineer kalman filtresi tabanlı pozisyon takip algoritması 22
Şekil 3. 3 Kalman ve Bütünleyici filtrelerin performans karşılaştırılması 23
Şekil 3. 4 B çerçevesinin $^{A}\hat{\imath}$ ekseni etrafındaki oryantasyonu. 24
Şekil 3. 5 Altı serbestlik dereceli bütünleyici filtre 30
Şekil 3. 6 MARG uygulaması için komple oryantasyon filtresinin blok şeması.....32
Şekil 4. 1 Sıfır hız algılama blok diyagramı 35
Şekil 4. 2 AHRS ivmeölçer verileri 37
Şekil 4. 3 AHRS ivmeölçer verileri 8-15sn aralığı..... 37
Şekil 4. 4 Filtrelenmiş ivme büyüklüğü ve durağan dönemler 38
Şekil 4. 5 Filtrelenmiş ivme büyüklüğü ve durağan dönemler 38
Şekil 4. 6 SHA ile hız tahmini 39
Şekil 4. 7 SHA ile pozisyon tahmini 40
Şekil 4. 8 Yürüyüş rotasının X-Y-Z ekseninlerinde gösterimi 40
Şekil 4. 9 Yürüyüş rotasının X-Y ekseninde gösterimi 41
Şekil 4. 10 AHRS + SHA uygulaması 42
Şekil 4. 11 Önceki 5 adımın pozisyonu ve mevcut adım 44
Şekil 4. 12 SKE blok diyagramı 47
Şekil 4. 13 SKE+SHA pozisyon tahmini 48
Şekil 4. 14 SKE+SHA pozisyon 3B gösterimi 48
Şekil 4. 15 SKE+SHA pozisyonun X-Y ekseninde görünümü 49
Şekil 4. 16 Beşli adım gurubu ile SKE kullanılarak yapılan spiral merdiven testi 50
Şekil 4. 17 Beşli adım gurubu ile spiral merdiven testi X-Y düzleminde görünüş 50
Şekil 4. 18 İkili adım gurubu ile SKE kullanılarak yapılan spiral merdiven testi.....51
Şekil 4. 19 İkili adım gurubu ile spiral merdiven testi X-Y düzleminde görünüş.....51

Şekil 4. 20 Dairesel yürüyüş testi 52
Şekil 4. 21 Sistemin genel blok diyagramı 53

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2. 1 BNO55 ivmeölçer sensörünün parametreleri 14
Çizelge 2. 2 BNO55 dönüölçer sensörünün parametreleri 15
Çizelge 2. 3 BNO55 manyetometre sensörünün parametreleri 16

ÖZET

SENSÖR FÜZYONU ALGORİTMALARI İLE AÇISAL KONUM REFERANS SİSTEMİ TASARIMI

Oben ÖZKAN

Mekatronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Aydın YEŞİLDİREK

Açısal konum referans sistemleri hava, deniz, uzay araçlarında, akıllı mühimmatlarda, pilot kasklarında, akıllı cep telefonlarında veya sanal gerçeklik gözlükleri gibi 3 boyutlu uzayda açısal durum bilgisinin elde edilmesi gereken yerlerde başlangıç durumu veya belirli bir referansa göre açısal durum bilgisini veren sensörler bütünüdür.

Literatürde "attitude and heading reference system" olarak geçen bu cihazlar piyasada AHRS kısaltması ile bilinirler. AHRS üç eksende yuvarlanma, yunuslama ve yalpalama açılarının bilgilerini verecek sensörlerle donatılmış cihazlardır. Bu sensörler açısal konum bilgilerini dünya üzerindeki doğal etkilerden elde etmek durumundadır bu sebeple ataletsel ve manyetik alan sensörleri AHRS için sıklıkla kullanılan sensörlerdir. AHRS bu sensörlerden aldığı bilgileri sensör füzyonu algoritmaları ile birleştirerek ihtiyaç duyulan 3 eksen açısal konum bilgisini sağlamaktadır. AHRS sensörlerinde ataletsel olarak ivmeölçer ve jiroskop sensörleri bulunmaktadır manyetik alan bilgisi hesaplamaya dahil edilecekse manyetometre sensörleri de dahil edilmektedir bunun dışında barometre ve sıcaklık sensörleri de hassasiyeti arttırmak için kullanılmaktadır. Sensörler günümüzde genellikle mikro elektro-mekanik (MEMS) sensör paketleri halinde kullanılsa da özellikle taktik ve navigasyon seviyesindeki cihazlarda jiroskop hatalarını en aza indirgemek için sagnac etkisini kullanarak açısal hız elde eden lazer jiroskoplar kullanılmaktadır. Ancak uygun maliyetli AHRS sistemlerinde üreticiler MEMS teknolojisine yönelmete ve

sensörlerden kaynaklanan hataları etkili sensör füzyonu algoritmaları ile telafi etmeyi amaçlamaktadırlar. AHRS sistemleri için sensör füzyonu algoritmaları literatürde yoğun bir şekilde çalışılmıştır ve bu tezde sıklıkla kullanılan algoritmalardan bahsedilmiştir.

Ataletsel navigasyon sistemi, harici bir referans sisteme ihtiyaç duymadan ataletsel sensörlerden aldığı verileri kullanarak pozisyon, hız ve açısal konum bilgilerini hesaplayan cihazlardır. Bu cihazlar bir çok alanda kullanılmakla birlikte iç ortam navigasyon ve yaya takip sistemlerinde de kullanılmaktadır. Literatürde birçok iç ortam navigasyon uygulaması üzerinde çalışılmış olup tez çalışmamda benzer bir uygulamaya yer verilmiştir.

Bu tez çalışmasında, bir açısal konum referans sisteminin tasarımı verilmiştir. Düşük maliyetli MEMS tabanlı atalet sensörlerinin verileri, doğru bir AHRS sistemi elde etmek için sıfır hız algılama, sezgisel sapma giderme ve bütünleyici filtreleme ile birleştirilmiştir. Harici kaynaklara ihtiyaç duymayan böyle bir navigasyon sisteminin, iç mekan navigasyon sistemleri dahil olmak üzere bazı uygulamalarda önemli avantajları vardır. Bu uygulamalardan biri yaya izleme sistemleridir. Bu sistemlerde, yürüyüşün doğasından elde edilen ivme değerleri, hız ve konum tahminlerini almak için entegre edilmiştir. Düşük maliyetli MEMS tabanlı IMU'lar gürültüye karşı çok hassastır, bir yardım mekanizması olmazsa büyük ölçüde sürüklenirler. Tez çalışmasında tasarlanan açısal konum referans sistemi yürüyen bir kişinin ayağına monte edildi, iç mekan navigasyon sistemi incelendi ve önerilen filtre algoritmalarıyla sensörlerin neden olduğu hataları en aza indiren umut verici sonuçlar verdi. Çalışmanın sonunda, filtre doğrulandı ve sensörlerin neden olduğu uzaysal kaymanın ortadan kaldırıldığı gözlemlendi.

Anahtar Kelimeler: Açısal konum referans sistemi, Ataletsel sensörler, Sensör Füzyonu, Ataletsel navigasyon, Konum tahmini

ABSTRACT

ATTITUDE AND HEADING REFERENCE SYSTEM DESIGN WITH SENSOR FUSION ALGORITHMS

Oben ÖZKAN

Department of Mechatronics Engineering

MSc. Thesis

Adviser: Asoc.Prof.Dr. Aydın YEŞİLDİREK

Attitude and heading reference systems are a set of sensors that provide angular position information in an air, sea, spacecraft, smart ammunition, pilot helmet, smartphone or virtual reality glasses, where angular condition information is required in three-dimensional space or where the angular position information is relative to a particular reference.

These devices, known as "attitude and heading reference system " in the literature, are known on the market with the acronym of AHRS. AHRS is equipped with sensors that provide information on roll, pitch and yaw angles in three axes. These sensors have to obtain angular position information from natural effects around the world, so inertial and magnetic field sensors are commonly used sensors for AHRS. By combining the information received from these sensors with the sensor fusion algorithms, AHRS provides the required 3-axis angular position information. The AHRS sensors are equipped with accelerometers and gyroscope sensors. Magnetic field information is included in the calculation. Magnetometer sensors are also included. Barometer and temperature sensors are as well used to increase sensitivity. Although sensors are often used in microelectro-mechanical (MEMS) sensor packs, laser gyroscopes with angular velocity are used to minimize gyroscope errors, especially in tactical and navigation-level devices. However, in cost-effective AHRS systems, manufacturers aim to compensate

for MEMS technology and to compensate for errors due to sensors by effective sensor fusion algorithms. Sensor fusion algorithms for AHRS systems have been studied extensively in the literature and frequently used algorithms are mentioned in this thesis study.

The inertial navigation system is used to calculate position, speed and angular position information using the data received from inertial sensors without the need for an external reference system. These devices are used in many areas, but they are also used in indoor navigation and pedestrian tracking systems. In the literature, many interior environment navigation applications have been studied and a similar application has been included in my thesis study.

In this thesis study, an angular position reference system design is given. Low-cost MEMS based inertial sensors data is fused by zero velocity sensing, heuristic drift elimination and complementary filtering to obtain an accurate AHRS system. Such a navigation system not needing external sources has significant advantages in some application including indoor navigation systems. One of such applications is the pedestrian tracking systems. In these systems, the acceleration values derived from the nature of the walk are integrated to get the speed and position estimates. Low-cost MEMS based IMU's are very sensitive to noise and they drift drastically without an aiding mechanism. The angular position reference system designed in the article was mounted on the foot of a walking person, the indoor navigation system was studied and the errors caused by the sensors were minimized by the proposed filter algorithms showed promising results. In the end of the study, the filter was verified and it was observed that the spatial shift caused by the sensors has been eliminated.

Keywords: AHRS,Inertial Navigation,Inertial Sensors,Sensor Fusion,Dead Reckoning

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Açısal konum referans sistemleri havacılık ve uzay uygulamalarında sıkça kullanılan AHRS olarak bilinen sistemlerdir, terim attitude and heading reference system'in baş harflerinin kısaltmasından gelir ve bu makalede açısal konum referans sistemi kısaltması olarak AHRS kullanılacaktır. AHRS üç eksende yönelim ve açısal konum bilgisi veren sistemlerdir. AHRS uygulaması herhangi bir dış kaynağa ihtiyaç kalmadan ataletsel kuvvetleri ölçen ivmeölçer ve jiroskop sensörlerinin kalman filtresi, bütünleyici (complementary) filtre veya herhangi bir sensör füzyonu algoritması ile birleştirilerek sensör ekseninin dünya eksenine göre açısal konumunun tahmin edilmesi uygulamasına dayanır. Ataletsel sensörler IMU olarak bilinen sensör paketleri halinde piyasada bulunurlar.

Bir iç ortam yaya konumlandırma sistemi, kapalı bir ortamda yaya konum bilgilerine gerçek zamanlı erişim için bir sistemdir [1]. Yaygın olarak kullanılan Global Konumlandırma Sistemi (GPS) dış mekanda çok hassas konumlandırma bilgileri elde edebilir. Bununla birlikte, kapalı ortamlarda, uydu sinyali bina tarafından kolayca bozulduğundan GPS doğru yaya konumlandırma bilgisi sağlayamaz [2]. Günümüzde, iç ortam konumlandırma teknolojisi, kablosuz ve ataletsel konumlandırma teknolojisi içinde sınıflandırılmaktadır. Kablosuz konumlandırma teknikleri kızılötesi [3], ultrasonik [4], Bluetooth [5], Wi-Fi [6], ZigBee [7], Radyo Frekansı Tanımlama (RFID) [8], Ultra Geniş Bant (UWB) [9], görsel [10] ve kablosuz ağ konumlandırma teknolojisi [11] gibi teknolojilerle gerçekleştirilir. Bu konumlandırma teknolojileri, görüş hattı dışında kalma

(Non-Line Of Sight) faktörleri ve çok yollu faktörler gibi dış faktörlerden etkilenir [12]. Bu nedenle, kablosuz konumlandırmada doğruluk yeterince yüksek değildir ve kararlılık zayıftır. Ataletsel konumlandırma teknolojisi [13], bir ivmeölçer ve jiroskopa dayalı olarak yaya hızı, konumu ve açısal tutumu elde eder. Ataletsel navigasyon hataları dış ortamdan etkilenmez, fakat ataletsel navigasyon sistemi uzun bir süre boyunca kümülatif hatalara eğilimlidir. Fiyat, yapı, hacim ve ağırlık bakımından avantajları olan Mikroelektromekanik-Ataletsel Ölçüm Birimlerine (MEMS-IMU) dayalı ataletsel seyir sistemleri son yıllarda büyük ilgi görmüştür [14].

İç ortam navigasyon sistemlerinde dış kaynak düzeltmesi kullanılamadğı için başka doğal etkiler kullanılır. Bu makalenin konusu olan yaya takip sistemlerinde genellikle sıfır hız algılama (zero velocity update) yöntemi kullanılmaktadır. Sıfır hız algılama yönteminde IMU sensörlerinden kaynaklanan hataları gidermek için yürümenin doğasından gelen bir takım doğal etkileri kullanan algoritma adım atma sırasında oluşan belirli ivme modellerini takip ederek yürüyen kişinin ayağına bağlı sensörlerin durağan ve hareketli anlarını takip etmektedir. Algoritma durağan dönemlerde yani ayak yerde sabitken hız verisini sıfıra çekmekte, kişi adım atmaya başladığında yani ayak hareketliyken hız verisini sensörlerden gelen bilgi ile hesaplamaktadır böylece sadece hareketli anlarda ölçüm yapan algoritma sensör hatalarından en az şekilde etkilenmeyi amaçlar. Literatürde genelde çeşitli sıfır hız algılama yöntemlerini tek başına kullanmak yeterli görülse de bu yöntemin yetersiz kaldığı uygulamalarda Borenstein ve Lauro Ojeda'nın çalışmasında [20] olduğu gibi ilave filtreler kullanılmaktadır.

AHRS tasarımı için çeşitli füzyon teknikleri üzerinde çalışılmıştır. Leccadito yüksek lisans tezinde bir ataletsel ölçüm sisteminin gereksinimi olan AHRS sistemi için ivmeölçer,jiroskop ve manyetometre sensörlerinin hata modellerini çıkarmış ve bu gürültülerin etkisini en aza indirmek için kalman filtresi üzerinde çalışmıştır [15]. Lai çalışmasında AHRS için bir bütünleyici filtre kullanmıştır IMU sensörlerinin hata modellerinin parametrelerini bulmak için 3 eksende 360 derece dönebilen bir platform oluşturmuş ve bir sensör kalibrasyon sistemi inşa etmiştir. Oluşturduğu bu sistemde doğru açı değerleri ile filtre değerlerini karşılaştırarak sensörlerin hata modelini bulma çalışmasında bulunmuştur [16]. Mahony R. Makalesinde AHRS sensör füzyonu algoritmasında tahmin kısmı için fikri kinematik denklemin girişini düzeltmektir. Jiroskop

kaymasından kaynaklanan hatayı düzeltmek için bir PI kontrolcüden faydalanmıştır. Mahony ivmeölçer ver jiroskop verilerinden faydalanmış manyetometreyi sistemine dahil etmemiştir jiroskop ve ivmeölçer verilerini yerçekimi ile karşılaştırmış ve dünya çerçevesine göre sistemin açısal konumunu bulmaya çalışmıştır [17]. Navigasyon sistemlerindeki ataletsel ve GPS/INS destekli referans çerçevelerin hesabı, ECEF koordinat sistemleri, rotasyon matrislerinin hesapları AHRS ve ataletsel navigasyon sistemelrinin kinematik eşitlikleri Farrel'ın kitabında işlenmiştir [18]. Madgwick açısal konum bilgisi için quaternion yaklaşımını kullanmıştır. Bu yaklaşımda açısal konum tahminini iyileştiren füzyon algoritmasını minimizasyon problemi olarak formülüze ederek bu problemi gradyan iniş algoritması ile çözmüştür bu yaklaşım Extended Kalman Filteresinin (EKF) tahmin doğruluğuna yakın sonuçlar üretmesinin yanında düşük işlem gücü ile bile çalışabilmekte böylece 8 bitlik mikrodenetleyicilerde bile çalışabilecek algoritma ortaya çıkmaktadır [19]. İç ortam navigasyon sistemi için Borenstein ve Lauro Ojeda tarafından Heuristic Drift Elimininationn (HDE) adı verilen bir yöntem ile sürüklenme eliminasyonu için sezgisel bir yöntem önermişlerdir [20]. S.K. Park ve Y.S Suh çalışmasında Hidden Markov Model ile sadece tek bir jiroskop verisi kullanarak yeni bir sıfır hız algılama algoritması üzerinde çalışmış ve durağan dönemleri daha güvenilir şekilde tespit etmişlerdir [22]. Foxlin,E ayakkabıya monteli takip sistemi çalışmasında sıfır hız algılama algoritmasının hız hatalarını EKF ile düzelterek daha doğru pozisyon tahmin verileri elde etmiştir [23].

Bu makaledeki çalışma sıfır hız algılama yöntemine ilave yeni bir filtre geliştirerek daha doğru konum bilgisi sunan yaya navigasyon sisteminin geliştirilmesi üzerinedir. Sistemimizi literatürden ayıran çalışma, algoritmamızın yürüyen kişinin baskın yönünü bulmaya ve sürüklenmeleri baskın yöne göre elimine etmeye yönelik olmasıdır. Sistemimizde IMU sensörleri yürüyecek kişinin ayakkabısına bağlanmıştır, ivmeölçer verilerinden önce sıfır hız algılama yöntemi ile pozisyon bilgisi alınmakta daha sonra makalenin çalışma konusu olan Sezgisel Kayma Eliminasyonu filtresi ile pozisyon bilgisi bir kez daha düzeltilmekte ve nihai pozisyon bilgisi kaydedilmektedir. Çalışmanın sonunda Sezgisel Kayma Eliminasyonu filtresi gerçeklenmiş ve yalnızca sıfır hız algılama filtresi kullanan navigasyon sistemlerine göre daha doğru pozisyon bilgisi tahmin ettiği doğrulanmıştır.

1.2 Tezin Amacı

Bu tezin amacı harici bir referans sinyaline ihtiyaç duymadan ataletsel kuvvetleri ve dünyanın manyetik alanını kullanarak herhangi bir cismin açısal konumunu ve cismin hareket doğrultusunu etkin ve sınırlı bir bütçe ile ölçebilecek yöntemleri araştırmak ve geliştirmektir. Çalışmamızda ilk olarak açısal konum referans sistemi için teorik alt yapı sunulacak ve çalışan bir konsept model yapılacaktır. Daha sonra bu sistem yürüyüş sırasında oluşan ataletsel değerleri kaydeden bir veri toplama sistemine dönüştürülecek ve toplanan verilerden konum tahmini yapan önerdiğimiz filtre algoritmasının çalışması incelenecektir. Bu tez çalışması ile aynı zamanda bu alanda yapılmış az sayıdaki Türkçe literatür çalışmasını da arttırmak hedeflenmiştir.

Bu amaç doğrultusunda literatürdeki çalışmalar taranmış ve çalışmalar iki ana gurup halinde incelenmiştir. İlk gurupta açısal konum referans sistemleri ve bu sistemlerde kullanılan filtre algoritmaları incelenmiş ve tez çalışmasında kullanılan sistem için en uygun filtre algoritması seçilmiş ve uygulanmıştır. İkinci gurupta ise ataletsel navigasyon sistemlerinden biri olan yaya takip sistemlerinde kullanılan yöntemlerden bahsedilmiş tez çalışması için önerdiğimiz filtre algoritması uygulanmış ve hata düzeltmedeki etkinliği karşılaştırmalı grafikler ile gösterilmiştir.

1.3 Hipotez

İnsan yürüyüş sırasında ayağı yerden kaldırma, ayağı ileri doğru atma ve yere basma işlemlerini gerçekleştirerek yürür bu sırada her bir işleme özgü ayakta 3 eksende ivmelenme değerleri oluşur. Eğer bu ivmelenme değerlerini tespit edersek yürüşün hangi aşamada olduğunu ve bir adımın atılıp atılmadığını belirleyebiliriz. Bu tezin amacı sensörlerden kaynaklanan ve integral ile kümülatif artan hataların yürüyüş analizine etkisini azaltmak olduğu için öncelikle bir adımın atılıp atılmadığını belirlemek ayağın ileri doğru gitme sırasında konum değişimini ölçmek ve yürüyen kişinin hangi doğrultuda yürüdüğünü tespit ederek diğer doğrultularda hatadan kaynaklı sapmaları elemek ölçüm sırasında meydana gelen sapmaları engelleyecektir.

Tez çalışmasında yürüyüşün baskın yönü bulunarak diğer yönlerde sensör hatasından dolayı konum bilgisini etkileyen kaymaların filtrelenmesi ile daha doğru bir iç ortam navigasyon sisteminin gerçeklenelip gerçeklenmeyeceği sınanmıştır.

BÖLÜM 2

AÇISAL KONUM REFERANS SİSTEMİ

Açısal Konum Referans Sistemi, bir cismin uzayda 3 eksendeki açısal konumunun bilgisini vermeye yarayan çoğunlukla ivmeölçer, jireskop ve manyetometreden oluşan bir sensörler bütünüdür. İngilizcede Attitude and Heading Reference System (AHRS) adıyla da bilinirler ve bu tezde AHRS kısaltması ile kullanılacaktır.

AHRS sensörleri hava, deniz ve kara araçlarında ve platformlarda kullanılmakta olup verdikleri açılsal konum bilgisi ile araçların yönelim açılarının bilgisini sağlar. AHRS 3 eksendeki yönelim bilgisini jiroskoptan alınan açısal dönme bilgisinin integralini alarak cismin yönelim açılarını belirler. Jiroskoptan kaynaklanan kaymaları ise yerçekimi ve dünyanın manyetik alanlarını kullanan ivmeölçer ve manyetometre sensörleri ile telafi etmeye çalışır. Bu telafi için sensör füzyonu algoritmaları kullanılır ve böylece cismin açısal konumu hatasız bir şekilde ölçülmüş olur. Tezin ikinci bölümünde sensör füzyonu uygulamasını daha iyi anlamak için çeşitli füzyon algoritmalarının bir analizi ve karşılaştırılması yapılmıştır. Bu bölümde 2 temel filtre incelenmiştir.

- Bütünleyici (Complementary) filtre
- Kalman Filtresi

2.1 Ataletsel Ölçüm Sensörleri

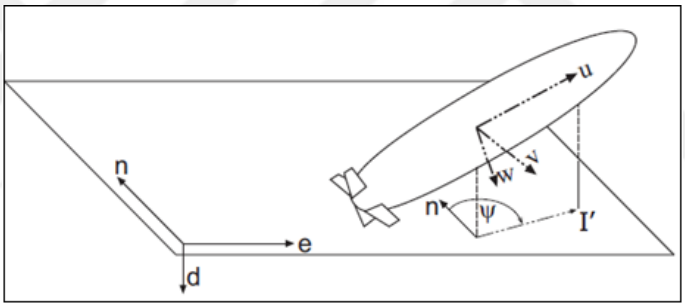
Ataletsel ölçüm sensörü, üç eksenli ölçüm yapabilen jiroskop ivmeölçer ve manyetometre sensörlerinin birleşimidir. IMU sensörleri, yeryüzü merkezli atalet eksenine göre çıkış ürettiği için ve hataların toplanarak artmasından ötürü sadece

integrasyon işlemi ile dönüş açılarını, öteleme hız ve deplasmanları hesaplanamaz. Bu sensörler yerçekimi, koriyolis etkileri, gürültü, ofset ve sürüklenmeye maruz kalmaktadır.

Açısal konum kestirimi yapabilmek için dönmekte olan bir cismin kinematik denklemini yazmak gerekir.

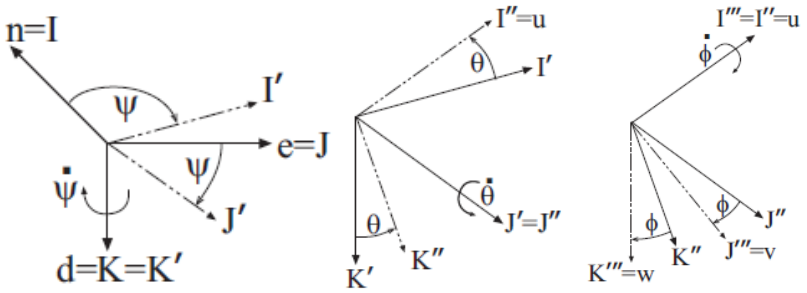
2.2 Rotasyon Matrisinin Matematiksel Çıkarımı

Bir eksen takımındaki vektörü başka bir eksen takımına göre ifade etmek için bu vektörün eksen takımları arasındaki dönüşümü tanımlayan rotasyon matrisi ile çarpmamız gerekir. Aşağıdaki şekilde rotasyon matrisi, navigasyon ekseninden araç eksenine dönüşümde gereklidir. Ψ, θ ve Φ açıları sırasıyla savrulma, yunuslama ve yalpalam açılarıdır. Bu açılar aracın ekseninin navigasyon eksenine göre dönüşümünü belirleyen açılardır. Bu açılar yaw, pitch, roll açıları olarak geçer.



Şekil 2. 1 Navigasyon ekseni ile araç ekseni arası ilişki [18]

Şekil 2.1'de araç ekseni u, v, w navigasyon ekseni ise n, e, d eksenleri ile gösterilmiştir. Navigasyon ekseninden araç eksenine dönüşüm yapabilmek için navigasyon eksenini rotasyon matrisi ile çarpmak gerekmektedir.



Şekil 2. 2 Sapma, yunuslama ve yuvarlanma açılarının rotasyonal etkisi [18]

Şekil 2.2'de görüldüğü gibi NED navigasyon koordinat takımı d(K) temel ekseninde Ψ yaw açısı kadar döndürülürse elde edilecek olan x'y'z' ekseni aşağıdaki matris çarpımı ile elde edilir.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n \\ e \\ d \end{bmatrix} \quad (2.1)$$

Ardışıl dönme kullanılarak x'y'z' eksen takımı y' eksenindei θ pitch açısı ile döndürülürse yeni x'',y'',z'' ekseni takımı aşağıdaki matris çarpımı ile bulunur.

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (2.2)$$

x'',y'',z'' ekseni Φ roll açısı ile döndürülürse araç eksenine ulaşmış oluruz

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & sin\Phi \\ 0 & -sin\Phi & cos\Phi \end{bmatrix} \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (2.3)$$

Navigasyon ned eksen takımı ardışıl ZYX temel eksenlerde sırasıyla Ψ, θ, Φ euler açılarıyla döndürülerek araç eksen takımı ulaşıldığı varsayıldığın da, tüm işlem tek bir dönme matrisiyle (2.1) ve (2.2) eşitlikleri (2.3)'de yerine yazılırak

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} cos\Psi & sin\Psi & 0 \\ -sin\Psi & cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\Phi & sin\Phi \\ 0 & -sin\Phi & cos\Phi \end{bmatrix} \begin{bmatrix} n \\ e \\ d \end{bmatrix} \quad (2.4)$$

eşitliği elde edilir. Denklem 2.4'deki matrisler çarpılırsa

$$R_n^b = \begin{bmatrix} C(\Psi)C(\theta) & S(\Psi)C(\theta) & -S(\theta) \\ -S(\Psi)C(\Phi) + C((\Psi))S(\theta)S(\Phi) & C(\Psi)C(\Phi) + S(\Psi)S(\theta)S(\Phi) & C(\theta)C(\Phi) \\ S(\Psi)S(\Phi) + C((\Psi))S(\theta)C(\Phi) & -C(\Psi)S(\Phi) + S(\Psi)S(\theta)C(\Phi) & C(\theta)C(\Phi) \end{bmatrix} \quad (2.5)$$

(2.5) navigasyon ekseninden araç eksenine dönüşümü temsil eden DCM(Directional Cosine Matrix) dir. Rotasyon matrisinin değeri küçük açı değişimleri için bulunmak istenirse (2.5) sıfır açılar etrafında lineerleştirilerek (kosinüs değerleri 1 e sinüs açı değerleri ise açının kendisi olarak radyan cinsinden açı değerine eşit olduğundan).

$$R_n^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & S(\Phi) \\ 0 & -S(\Phi) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -S(\theta) \\ 0 & 1 & 0 \\ S(\theta) & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & S(\Psi) & 0 \\ -S(\Psi) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

(2.6) deki matrisler çarpılırak yada direk olarak (2.5) kullanılarak aşağıdaki şekilde bulunur.

$$R_n^b = \begin{bmatrix} 1 & \Psi & -\theta \\ -\Psi & 1 & \Phi \\ \theta & -\Phi & 1 \end{bmatrix} \quad (2.7)$$

Küçük açı değişimleri, açısal hız ile birim zamanın çarpımı olarak ifade edilir.

$$R_{n(t)}^{n(t+\delta t)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & -\Psi & \dot{\theta} \\ \Psi & 0 & -\dot{\Phi} \\ -\dot{\theta} & \dot{\Phi} & 0 \end{bmatrix} \delta t \quad (2.8)$$

Denklem 2.8'da açısal hızların ters simetrik matris gösterimi olan Ω_{bn}^n terimi yazılırsa 2.9 eşitliği elde edilir.

$$R_{n(t)}^{n(t+\delta t)} = \mathrm{I}_{3\mathrm{X}3} - \Omega_{3X3}\, \delta\mathrm{t} \quad (2.9)$$

2.2.1 DCM (Rotasyon) Matrisi Türevinin Matematiksel Çıkarımı

Δt zaman sonra araç ekseni (b) nin navigasyon ekseni (n) ye göre DCM matrisinin türevi denklem 2.10 ile ifade edilir

$$\dot{R}_b^n(t) = \lim \frac{R_b^n(t+\delta\mathrm{t}) - R_b^n(t)}{\delta(t)} \quad (2.10)$$

δt zaman sonra oluşan rotasyon matrisi iki çarpana ayrılabilir.

$$R_b^n(t+\delta t) = R_{n(t)}^{n(t+\delta t)} R_b^n(t) \quad (2.11)$$

δt zaman değişiminde denklem 2.9 daki ifade denklem 2.10 de yerine yazılırsa denklem 2.12 eşitliği ortaya çıkar.

$$\dot{R}_b^n(t) = lim_{\delta t \to 0} \frac{(I - \Omega_{bn}^n\, \delta t\,) R_b^n(t) - R_b^n(t)}{\delta t} = -\Omega_{bn}^n\, R_b^n(t) \quad (2.12)$$

Denklem 2.12'de $\Omega_{bn}^n = R_b^n \Omega_{bn}^n\, R_n^b$ ile $-\Omega_{bn}^n = \Omega_{nb}^n$ eşitliği ile 2.13 eşitliği elde edilir. 2.13 denklemi rotasyon matrisinin türev denklemidir

$$\dot{R}_b^n = -\Omega_{bn}^n\, R_b^n = R_b^n \Omega_{nb}^b = R_b^n (\Omega_{ib}^b - \Omega_{in}^b) \quad (2.13)$$

Ω_{ib}^b matrisi aracın atalet eksenine göre açısal hızını belirtir bu matristen navigasyon ekseninin atalet eksenine göre açısal hızını belirten Ω_{ib}^b matrisi çıkarılırsa aracın navigasyon eksenine göre açısal hız matrisi Ω_{nb}^b bulunur. Dünyanın atalet eksenine göre

açısal hızı ve navigasyon eksenini dünyanın eksenine göre açısal hızının toplamı navigasyon eksenin atalet eksenine göre açısal hızına eşittir ve $\Omega_{in}^{b} = \Omega_{ie}^{b} + \Omega_{en}^{b}$ ve eşitliği ile gösterilir.İlgili eşitliklerden $\Omega_{nb}^{b} = \Omega_{ib}^{b} - \Omega_{ie}^{b} - \Omega_{en}^{b}$ eşitliği çekilir ve 2.13 denkleminde yerine yazılırsa.

$$\dot{R}_{b}^{n} = R_{b}^{n}(\Omega_{ib}^{b} - \Omega_{ie}^{b} - \Omega_{en}^{b}) \tag{2.14}$$

2.14 eşitliği elde edilir ve rotasyon matris çarpımı içeri dağıtıldığında

$$\dot{R}_{b}^{n} = R_{b}^{n}\Omega_{ib}^{b} - R_{b}^{n}\Omega_{ie}^{b} - R_{b}^{n}\Omega_{en}^{b} \tag{2.15}$$

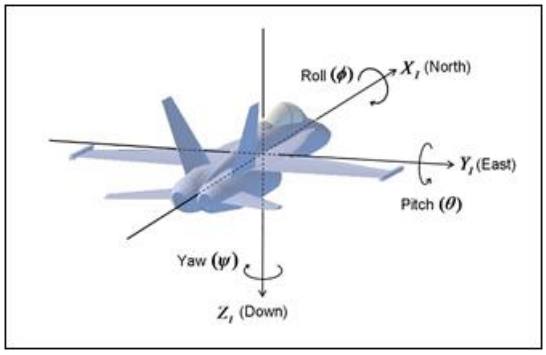
Denklem 2.15'de $\Omega_{ie}^{b} = R_{n}^{b}\Omega_{ie}^{n}R_{b}^{n}$ ve $\Omega_{en}^{b} = R_{n}^{b}\Omega_{en}^{n}R_{b}^{n}$ eşitlikleri ile

$$\dot{R}_{b}^{n} = R_{b}^{n}\Omega_{ib}^{b} - (\Omega_{ie}^{n} + \Omega_{en}^{n})R_{b}^{n} \tag{2.16}$$

Eşitliği elde edilir. Bu bölümde rotasyon matrisi eşitlikleri çıkarılmıştır.

2.2.2 Euler Açıları

AHRS sensörleri hem euler hem de quaternion gösteriminde oryantasyon açı bilgilerini sağlayabilir. Euler gösterimi quaternion gösterimine göre daha basittir. Öte yandan, denklem 2.17 den gelen "Gimbal Kilidi" adlı bir fenomen ile sınırlıdır. Sensörün hiçbir zaman +/- 90 derecelik açı açılarına yakın çalışmayacağı uygulamalarda, Euler Açıları iyi bir seçim olabilir.



Şekil 2. 3 Euler açıları

$$\frac{d}{dt}\begin{bmatrix}\Phi\\ \theta\\ \Psi\end{bmatrix} = \begin{bmatrix}1 & sin\Phi tan\theta & cos\Phi tan\theta\\ 0 & cos\theta & -sin\theta\\ 0 & sin\Phi/cos\theta & cos\Phi/cos\theta\end{bmatrix}\begin{bmatrix}\omega_{ib,x}^{b}\\ \omega_{ib,y}^{b}\\ \omega_{ib,z}^{b}\end{bmatrix} \tag{2.17}$$

Denklem 2.17'de görüleceği üzere θ yunuslama açısı 90 derece olduğunda matris hesabı yapılamaz ve sonuç tanımsız olur bu euler gösteriminin basitliğinin yanında bir dezavantajdır.

2.2.3 Kuaterniyon gösterimi

Matematikte, dördeyler (ya da Quaternion, kuaternion), Kuaterniyonlar 3 boyutlu uzayda dönme ve rotasyon değişikliğini yapmak için tanımlanmış sayı sistemidir. İlk defa İrlanda'lı matematikçi Sir William Rowan Hamilton tarafından 1843 yılında tanımlanmış ve 3 boyutlu uzaydaki matematiğe uygulanmışlardır.

Bir kuaterniyon, gerçek bir skalar ve karmaşık bir vektörden oluşan karmaşık sayıların bir uzantısıdır. Vektör kısmı, i, j ve k olarak adlandırılan üç karmaşık sayı içerir. Kuaterniyonlar üç boyutlu bir kartezyen koordinat sisteminin temel vektörleri olarak yorumlanabilirler. Bir kuaterniyon (2.18) eşitliği ile ifade edilir

$$q = w + xi + yj + zk = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} \tag{2.18}$$

Karmaşık kısımlar aşağıdaki özelliklere sahiptir.

$$ii = -1\,,\quad jj = -1,\quad kk = -1,\quad ij = k = -ji\,,\quad jk = i = -kj,\quad ki = j = -ik$$

Kuaterniyon yaklaşımında, karmaşık vektör bölümü bir dönme ekseni tanımlayarak iki koordinat çerçevesinin yönünü temsil eder skalar bölüm ise bu eksen etrafında dönecek açıyı belirtir. Euler teoremine göre, bu yaklaşım üç boyutlu uzayda herhangi bir yönelimi tanımlamak için yeterlidir.

Kuaterniyon işlemleri, genel vektör işlemleriyle neredeyse aynıdır.

$q1 = w1 + x1i + y1j + z1k$ ve $q2 = w2 + x2i + y2j + z2k$ olmak üzere 2 kuaternion çarpımı şu şekildedir.

$$\begin{aligned} q1 * q2 = {} & (w1w2 - x1x2 - y1y2 - z1z2) \\ & + (x1w1 + w1x2 - z1y2 + y1z2) \\ & + (y1w2 + z1x2 + w1y2 - x1z2)j \\ & + (z1w2 - y1x2 + x1y2 + w1z2)k \end{aligned} \tag{2.19}$$

Kuaternion eşleniği = $q = w + xi + yj + zk = (w, \vec{v})$

$$\tilde{q} = (w - xi - yj - zk) = (w, -\vec{v}) \tag{2.20}$$

Kuaternion normu

$$\mathrm{N(q)} = \sqrt{q * \widetilde{q}} = \sqrt{w^2 + x^2 + y^2}+z^2 \tag{2.21}$$

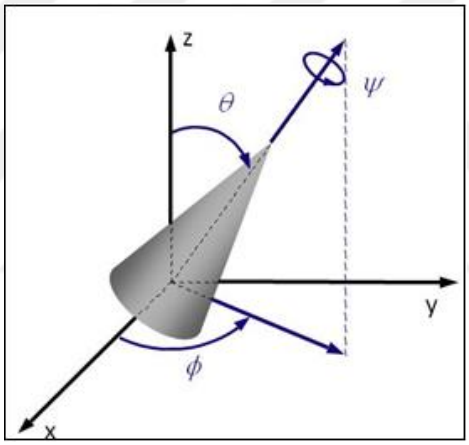
Normalizasyonu

$$q_{nornalization} = q/N(q) \tag{2.22}$$

Kuanternion tersi

$$q^{-1} = \frac{\tilde{q}}{N^2(q)} = \frac{\tilde{q}}{\tilde{q}q} \ , \qquad q^{-1} = \tilde{q} \tag{2.23}$$

Kuaternion yaklaşımının euler yaklaşımına göre avantajı gimbal kilidi fenomenine yakalanmamasıdır.



Şekil 2. 4 Kuaternion gösterimi

2.3 Sensörler

Bu bölüm, mevcut çalışmada kullanılan sensörlerin ve sensör modülünün bir tanımını sunar. Tez çalışmasında üç farklı sensör tipi, 3 eksenli jiroskop, 3 eksenli ivmeölçer ve 3 eksenli manyetometre kullanılmıştır. Sensörler Mikro elektro-mekanik sistemlerden üretilmişlerdir. MEMS sensörlerinin avantajı sistemlerin küçük paket boyutunda düşük güç tüketimi, ucuz ve yüksek kullanılabilirlikte olmasını sağlamaktadır.

2.4 Hata Modellemesi

Genel olarak, sensörler belirli bir fiziksel miktarı ölçer etkinin kuvvetine ve yönüne bağlı bir değer üretir. Çıktı değeri genellikle ilgilenilen miktar ile aynı alanda bulunan bir değer elde etmek için işlenmesi gereken bir elektrik voltajıdır. Hata modellemede önişleme genellikle bias çıkarma ve ölçeklendirme işlemleri ile yapılır. Bias sensörün doğrudan çıkış vermesini engelleyen eşik değeridir.

Tez çalışmasında her sensör için tipik olan doğrusal bir sensör modeli kullanacağız. Bu doğrusal model aşağıdaki kalibrasyon parametrelerinden oluşur.3 eksenli bir sensör tipi için bias, ölçeklendirme ve yanlış hizalama parametreleri dahil olmak üzere basit bir hata modeli (2.24)'deki gibidir.

$$\vec{z}_{out} = s_{nominal}(I_{3X3} + M_{sa})\vec{z}_{input} + \vec{b}_z \quad (2.24)$$

Burada $\vec{z}_{out}$ ve $\vec{z}_{input}$ üç sensörün çıkış ve giriş değerleridir. Giriş vektörü bir nominal ölçek faktörü "$s_{nominal}$" ile ölçeklendirilir. M_{sa} herbir sensör için ölçek faktörü sapmaları ve giriş ekseni için yanlış hizalamaları içerir. İlave bias değerleri $\vec{b}_z$ olarak belirtilir.

2.4.1 İvmeölçer Sensörü

Doğrusal bir ivmeölçer, tek bir ya da çoklu uzaysal eksende ivme ölçen bir cihazdır. Mevcut sensörlerin tiplerinin ortak prensibi, bir kütle üzerindeki kuvveti hissetmek ve Newton'un F = ma hareketinin ikinci yasasına göre ivmeyi tespit etmektir.

İvmeölçerler kabaca mekanik ve katı hal cihazı olarak sınıflandırılabilirler. Mekanik ivmeölçerler, yaylarla asılı bir kütlenin yer değiştirmesini ölçer. Katı hal cihazları ise bir ucunda serbest hareketli bir prova kütlesine tutturulmuş ve diğer ucundaki kasaya sağlam şekilde tutturulmuş bir konsol kirişinden oluşur. İvmelenme, prova kütlesinin konumunu değiştirir ve ortaya çıkan kuvvet kirişin gerilimi ölçülerek belirlenebilir. MEMS ivmeölçerleri, mekanik ve katı hal cihazları ile aynı prensiplere dayanmaktadır.

Tez çalışmasında BOSCH firması tarafından üretilmiş olan BNO55 sensörünün ivmeölçeri kullanılmıştır 2g-16g arasında ölçeklendirilebilen bu sensör 14 bit çözünürlükte 1000Hz frekansta I2C protokolü ile veri sağlayabilmektedir.

Çizelge 2. 1 BNO55 ivmeölçer sensörünün parametreleri

Parametre	Sembol	Birim	Min	Typ	Max
Sensitivity	S	Lsb/mg		1	
Sensitivity Temperature Drift	TCS	%/K		+-0.03	
Zero g offset	Off_{xyz}	mg	-150	+-80	+150
ADC Noise	n_{rms}	ug/$\sqrt{Hz}$		150	190
Alignment Err	E_a	°		0.5	2
Sensitivity supply voltage drift	S_{vdd}	%/V		0.065	0.2

İvmeölçer (2.25)'deki gibi modellenebilir.

$$a = \dot{v} - g + b_a + n_a \tag{2.25}$$

Burada, $\dot{v}$, g, b_a ve n_a sırasıyla bir anlık doğrusal hızlanma, yerçekimi hızlanması, ivmeölçer önyargısı ve gürültüyü temsil eder.

2.4.2 Jiroskop Sensörü

Jiroskop açısal dönmeyi ölçer. Bu sensörlerin ölçümünü etkileyen en önemli sorun bias ve sürüklenme ve konusudur. Jiroskop dönme hızına ek olarak gürültü ile birlikte ve ortogonal olarak monte edilmiş üç eksen etrafında yanlılığı ölçer. MEMS jiroskopları için sıcaklık, darbe kuvveti ve diğer faktörler de hataya neden olabilir. Genel olarak jiroskop ölçümü aşağıdaki gibi modellenebilir

$$\Omega_g^s(t) = \Omega(t) + s_g(t) + n_g(t) \tag{2.26}$$

MEMS jiroskoplarının çıktısı doğrusal ivmelenmelere karşı da hassasiyet gösterir. Bu karşılıklı bağımlılık çok küçüktür ve uzun süre boyunca yerçekiminden daha büyük statik ivmelere maruz kalmayan çoğu uygulama için, bu bağımlılık güvenle göz ardı edilebilir. Bütünlüğü sağlamak için üreticiler, jiroskop sensör modelinin bir parçası olarak lineer hızlanma kalibrasyon parametrelerine gyro hassasiyetini sağlar. Fabrika kalibrasyonu sırasında, jiroskopları değişen doğrusal ivmelere maruz bırakılır ve bu kalibrasyon parametrelerini en küçük kareler tahmini kullanarak tahmin edilir.

Ω_g^s jiroskop ölçüm değeri Ω jiroskobun gerçek ölçüm değeri s_g ve n_g ise sürüklenme ve bias değerleridir. BNO55 Jiroskop 125°/s den 2000°/s aralığında 12-523Hz aralığında örnekleme verebilen bir sensördür.

Çizelge 2. 2 BNO55 dönüölçer sensörünün parametreleri

Parametre	Sembol	Birim	Min	Typ	Max
Sensitivity	S	Lsb/°/s Rad/s		16 900	
Sensitivity Temperature Drift	TCS	%/K		+-0.03	+-0.03
Zero rate offset	$Off_{\Omega x,y,z}$	°/s	-3	1	+3
Output Noise	n_{rms}	°/s		0.1	0.3
Nonlinearity	NL	%FS		0.05	0.2

2.4.3 Manyetometre

Bir manyetometre, yerel manyetik alanın gücünü ve yönünü ölçen bir sensör türüdür. Ölçülen manyetik alan hem dünyanın manyetik alanının hem de yakındaki nesneler tarafından oluşturulan manyetik alanların bir birleşimi olacaktır. Manyetik alan, sensör referans çerçevesinde ölçülür.

Dünyanın manyetik alanı, bir ucu Dünya'nın coğrafi Kuzey Kutbu yakınında, diğeri de dünyanın coğrafi Güney Kutbu yakınında bulunan manyetik bir dipole benzeyen kendiliğinden devam eden bir manyetik alandır. Bu manyetik alanın gücü, Güney Amerika'da 0,3 Gauss'a ve kuzey Kanada'da 0,6 Gauss'a kadar olan güçlerle Dünya genelinde değişmektedir. Bu manyetik alan zaman içinde nispeten kararlı olmasına rağmen, iyonosferdeki elektrik akımları yüzey manyetik alanlarını bir derece kadar saptayabilen günlük değişikliklere neden olabilir. Normal olarak, alan kuvvetindeki günlük değişiklikler, rotadaki yaklaşık 0.03 derecelik bir değişikliğe eşit olacak olan 0.25 mGauss derecesindedir. Bu pozisyondaki bir değişimin küçüklüğü, çoğu MEMS bazlı manyetometrenin çözünürlüğü ile aynı büyüklüktedir, bu nedenle çoğu durumda Dünya'nın manyetik alanı zamana göre sabit sayılabilir.

MEMS manyetometrelerinin çoğu, bir manyetik alan uygulandığında direnci değiştiren bir direnç veya hall effect temelli çalışır. Sapma, ölçek ve bias etkilerinden kaynaklanan sensör hatalarının yanı sıra, manyetik alan ölçümleri sensörün çevresindeki ferromanyetik elemanlar tarafından bozulmaktadır ve bunlar hard iron ve soft iron bozulmaları olarak adlandırılmaktadır. Hard iron bozulmalarına, sabit, ilave bir manyetik alan sergileyen malzemeler neden olurken, soft iron bozulmaları, malzemenin kendinden kaynaklanan bozulmaların sonucudur. Bu hatalar kalibrasyon sırasında dikkate alınmalı ve bu ölçüm işlemini daha karmaşık hale getirmektedir. BNO55 Manyetometre sensörü 1300uT(x-y ekseni), 2500uT(z ekseni) aralığında 2-30Hz bantgenişliğinde sensör verisi sunabilen bir sensördür.

Çizelge 2. 3 BNO55 manyetometre sensörünün parametreleri

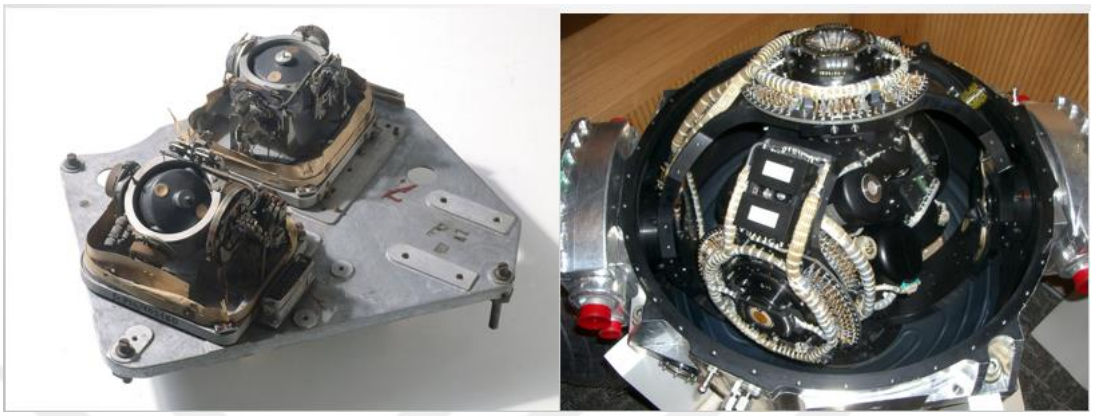
Parametre	Sembol	Birim	Min	Typ	Max
Sensitivity	S	Lsb/°/s Rad/s		16 900	
Sensitivity Temperature Drift	TCS	%/K		+-0.01	+-0.03
Zero B offset Withoutcallibration vs calibration	$Off_{\Omega x,y,z}$	uT		2	40
Output Noise	n_{rms}	uT		0.6	1
Full Scale Nonlinearity	NL	%FS		1	
Sensitivity supply voltage drift	S_{vdd}	%/V		<0.4	

2.4.4 AHRS Birimi

AHRS olarak daha iyi bilinen Açısal Konum Referans Sistemleri, 3 eksenli Atalet Ölçüm Birimi (IMU) ile birleştirilmiş 3 eksenli manyetik sensörden ve sensör füzyonu algoritmalarını hesaplayan gömülü bir işlemciden oluşan yaw(sapma),pitch(yunuslama) ve roll(yuvarlanma) açılarını üzerinde bulunduğu objeye göre hesaplayan cihazlara verilen isimdir.

Açısal konum referans sistemlerinin 100 yılın üzerinde bir geliştirme süreci vardır. 2.dünya savaşında Almanların ünlü A4-V2 roketleri balistik rotaları için gereken açısal

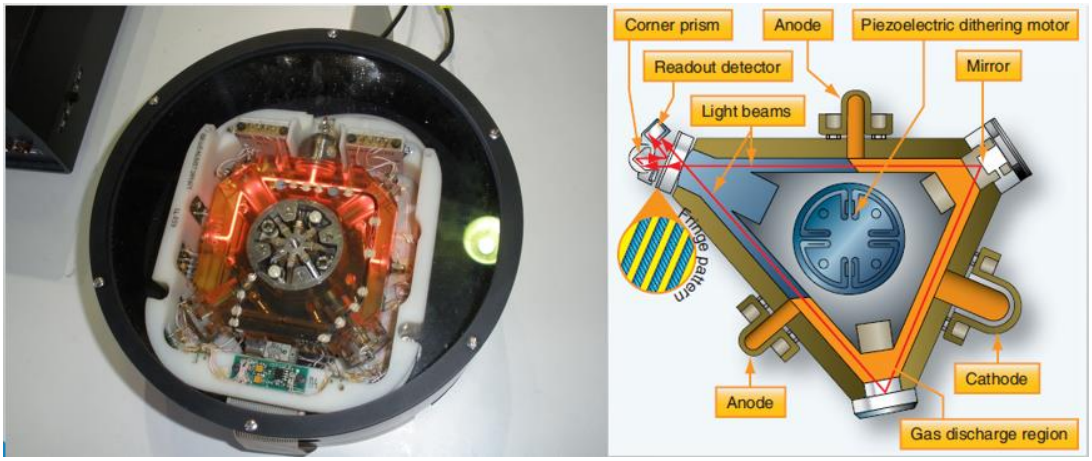
konum bilgisi LEV-3 adındaki 2 mekanik jiroskoptan oluşan bir ataletsel platformdan elde etmekteydi. Aya ilk kez insan taşıyan Saturn V roketleri yine uzayda konum belirlemek için ST-124-M3 cihazından faydalanmıştır bu cihazda 3 gimball her eksende 3 ivmeölçer 2 sarkaç mekanizmasından oluşmaktaydı.



Şekil 2. 5 Solda LEV-3 sağda ST-124-M3 AHRS birimleri

AHRS sensörleri öncelikle uçaklarda bulunan büyük geleneksel mekanik jiroskopik uçuş aletlerinin yerini almak ve daha iyi güvenilirlik ve doğruluk sağlamak üzere tasarlanmıştır. Tipik olarak bir AHRS, bir fiber optik (FOG) veya MEMS 3 eksenli açısal hız gyro triadından, 3 eksenli MEMS ivmeölçerinden ve manyetometre olarak bilinen 3 eksenli manyetik sensörden oluşur. Bu çeşitli sensörleri kullanarak oryantasyon çözümünü hesaplamak için yerleşik bir sensör füzyon algıritması (Kalman filtresi veya complementary filtre) kullanılır. Bazı AHRS sensörleri, jiroskop sapmasını dengelemek ve atalet hızlanma vektörünün daha doğru bir tahminini sağlamak için GPS kullanır.

Halka lazer jiroskopu (RLG), aynı yol üzerinde iki bağımsız karşı-yayılma rezonans modunda bir halka lazeri içerir, bu iki farklı lazer birleşim noktasında girişime uğrar jiroskop dönerken sagnac etkisiyle lazerlerden birinin birleşim noktasına giden yolu uzayacağı için girişim deseninde farklılaşma oluşur ve faz kayması görülür ve bu kayma rotasyonu tespit etmek için kullanılır.



Şekil 2. 6 Bir halka lazer dönüölçeri ve çalışma mantığı

Pek çok üst düzey taktiksel kalite AHRS sensörü inşa edilmiştir. Örneğin Northrop Grumman LN-200'ün tarafından üretilen LISA 200 AHRS, 7x4x4.5 inç boyutunda ve 4.5 pound ağırlığa sahiptir.



Şekil 2. 7 Northrop Grumman LN-2000 AHRS

Bu AHRS, UH-60 Blackhawk dahil olmak üzere birçok platformda kullanılmaktadır. Diğer bir popüler AHRS, Avrupa'da da üretilen LCR 92 ve LCR 93 AHRS üniteleridir. Her iki sistem de fiber optik jiroskopları kullanır. LCR 92, yerçekimi yönünü algılamak için bir kabarcık seviyesi kullanırken LCR 93, MEMS bazlı ivmeölçerler kullanır.

Şu anda piyasada düşük maliyetli (<10.000 $) AHRS sensörleri sunan çok sayıda şirket vardır. Bu sensörlerin tümü MEMS jiroskopları, ivmeölçerler ve manyetometreleri kullanır. Piyasadaki düşük maliyetli AHRS sensörleri arasındaki temel fark, sensörde

kullanılan tutum tahmin algoritmasıdır. Bu sensörler iki farklı gruba ayrılabilir. Bir yanda, piyasada, üst seviye AHRS sistemlerinde kullanılana benzer bir Kalman filtre algoritması kullanan tam olarak AHRS sensörleri vardır. Bu tür bir algoritma, gerçek zamanlı olarak zamanla değişen gyro kaymasını tahmin ve elimine etmek için manyetik ve ivme ölçümlerini kullanır. Bu sistemler yüksek performans sunar ancak işlemci için hesaplanacak yoğun bir işlem yükü getirir. Bu üst seviye algoritmalarda, matematiksel tekillikler ve gimball lock fenomenine yakalanmamak için kuaterniyon matematiğini kullanılır. Diğer sensör grubu, sensörün oriantasyonunu gerçek zamanlı olarak tahmin etmek için Kalman olmayan filtre bazlı algoritmalar kullanır. Bu tür bir algoritma, ileri teknoloji AHRS sistemlerinde kullanılanlardan farklıdır ve performansın üstün kalitesi üreticiye bağlıdır ancak genel olarak işlemciye daha az işlem yükü getirir.

AHRS'lerde genellikle çok hassas jiroskoplar kullanılır çünkü jiroskopların kalitesi genel performansın üzerinde büyük bir etkiye sahiptir. Taktik dereceli AHRS'ler, çok sabit açısal oran ölçümleri sağlamak için fiber optik jiroskopları(FOG) kullanır. Son zamanlarda AHRS üreticileri düşük güç ve düşük maliyetlerinden dolayı MEMS jiroskoplarını kullanmaya başlamıştır. MEMS jiroskoplarındaki son gelişmeler, düşük seviye fiber optik jiroskoplarla karşılaştırılabilecek performans seviyelerine ulaşmıştır.

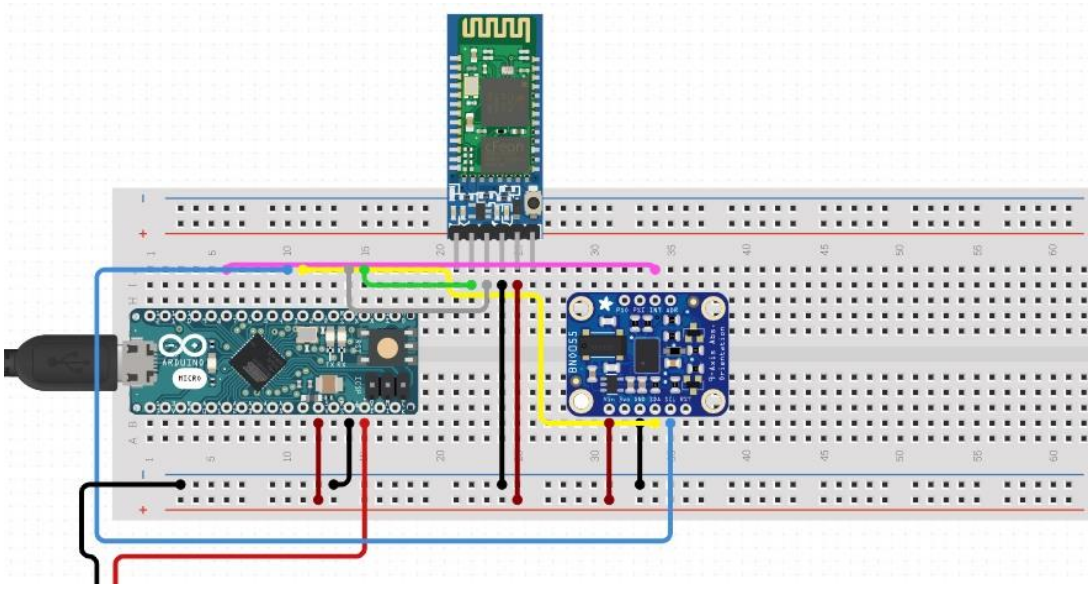
İleri teknoloji AHRS sistemleri genel olarak katı hal silikon ivmeölçer kullanır. AHRS'de kullanılan ivmeölçerler uzun vadeli önyargı kararlılığına sahip olacak şekilde seçilir. AHRS'de sabit bir ivmeölçer önemlidir, çünkü yatay seviye düzleminin bir ölçümü ile yerleşik algoritmayı sağlar. İleri teknoloji AHRS sistemleri genel olarak tüm çalışma koşullarında 2 mg'dan daha iyi hassasiyette taktiksel sınıflandırma ivmeölçerleri kullanır.

AHRS'de kullanılan manyetik sensörler sistemin kalitesine göre değişiklik gösterir, genel olarak yüksek kaliteli AHRS uygulamaları için flux-gate manyetometreleri düşük kaliteli AHRS sistemleri için ise MEMS Anisotropik Manyeto Rezistif (AMR) sensörler kullanılır. Genel olarak uçak sistemlerinde manyetometre AHRS'den ayrı bir uzak yere monte edilir. Manyetometre, elektrik motorları tarafından üretilen manyetik alanlar gibi dış manyetik ortamdan etkilenebileceği için, tipik olarak manyetik ortamın en az olduğu bir şekilde seçilmiş yerlere monte edilir. Düşük maliyetli birçok AHRS sistemi, uzak bir manyetometre bağlamak için bir yol sağlamaz. MEMS tabanlı manyetometreler daha az

güç gereksinimi ve geleneksel flux-gate tasarımlarından çok daha küçük bir paket boyutu sunmaktadır. Yakındaki nesnelerin etkilerini telafi etmek için kullanmadan önce her iki manyetometrenin uçağa veya platforma kurulduktan sonra kalibre edilmesi gerekecektir.

AHRS sensörleri genel olarak üç yoldan biriyle kullanılır. İlk olarak, platformun oriantasyonunun uçuş verilerinin zamanın bir fonksiyonu olarak kaydedilmesini sağlamak için bir araç olarak kullanılabilirler. İkincisi, platform dengeleme için bir AHRS kullanılabilir. Açısal hız çıkışları, platform dengesini sağlamak için doğrudan bir kontrol döngüsüne bağlanabilir. AHRS ayrıca bir açısal referans kontrol sistemi olarak da kullanılabilir. Uçaklar, helikopterler, dört rotorlular, zeplinler ve hatta su altı robotik araçlarının tümü, kapalı devre bir açısal konum kontrol sistemi oluşturmak için AHRS kullanımından yararlanabilir. Tahmini yönlendirme açıları ile birlikte yüksek bir bant genişliğindeki açısal hız çıktıları, platformun doğrudan istenen tutum açıları ile komutlandırılmasını mümkün kılabilir.

Tez çalışmasında aşağıda bağlantı şeması görülen yapı kullanılmıştır BOSCH marka BNO55 IMU sensörü verileri I2C ile Atmel denetleyiciye göndermekte denetleyici sensör füzyonu algoritması ile kuaterniyonları hesapladıktan sonra Hc05 bluetooth modülü ile verileri bilgisayara aktarmaktadır.



Şekil 2. 8 Çalışmada kullanılan AHRS şematiği

BÖLÜM 3

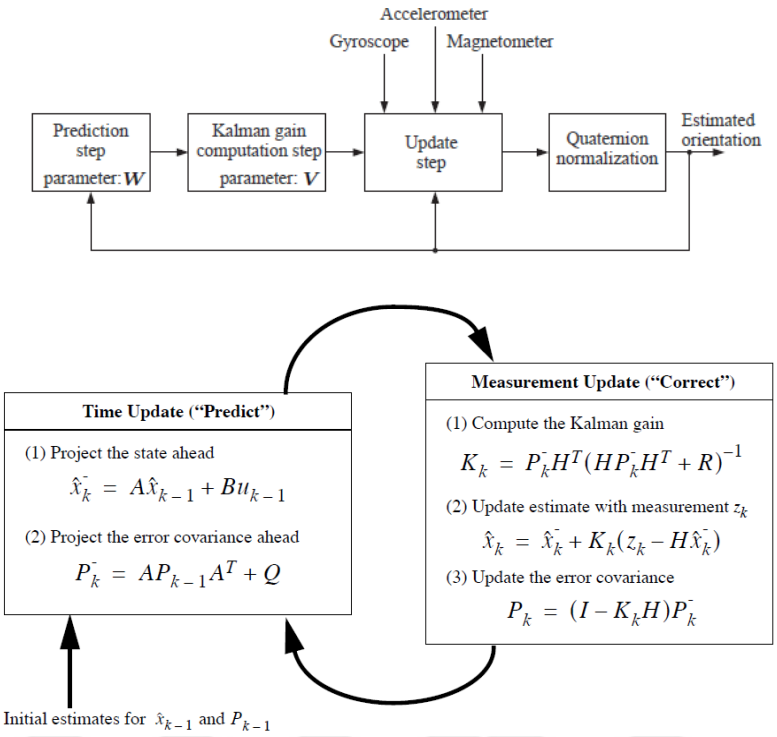
SENSÖR FÜZYONU ALGORİTMALARI

MEMS Sensörleri 2. Bölümde tartışıldığı ekonomik olmaları yanında çeşitli dış etkenlere karşı duyarlıdır ve bu durum sensörlerin çıkışlarında önemli oranlarda hatalara ve gürültülere sebep olur. İvmeölçerler hız değişimlerine karşı son derece hassas sensörlerdir böylece statik ölçümlerde güvenilir sonuçlar verirler ancak dinamik öçümlerde çok fazla hata üretirler jiroskoplar ise statik ölçümlerde sensörün doğasından kaynaklanan bias etkilerine karşı duyarlıdır ancak dinamik ölçümlerde son derece iyi iş çıkarırlar. Güvenilir bir açısal konum referans sensörü için çeşitli sensörlerin dinamik ve statik ölçüm özelliklerinden faydalanmak gerekir işte bu sebeple çeşitli AHRS algoritmaları geliştirilmiştir. 2 temel AHRS algoritması uygulamalarda yer edinmiştir bunlar Kalman filtresi uygulaması ve Bütünleyici (Complementary) filtre uygulamalarıdır.

3.1 Kalman Filtresi

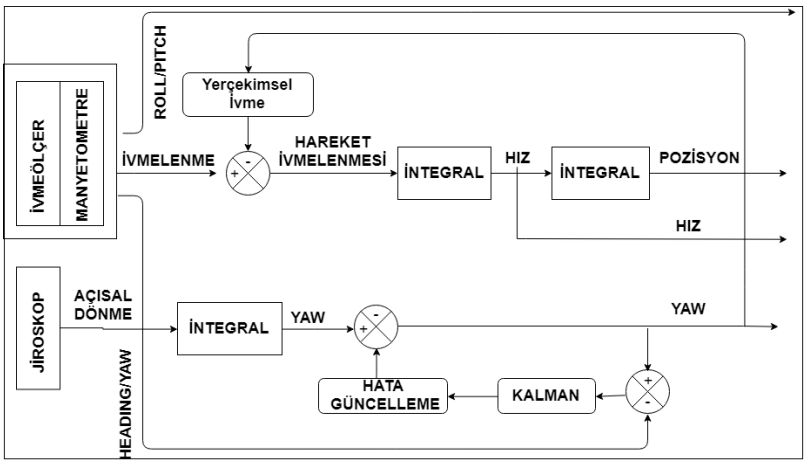
Kalman filtresi, modelin önceki bilgileriyle birlikte giriş ve çıkış bilgilerinden sistemin durumlarını tahmin edilebilen filtredir. Kalman filtresinin bir örneği çölde ilerleyen bir robotla açıklanabilir. Robot, önceden bir ölçümü yapılmış bilinen bir yerden başlar ileri doğru hareket etmeye başladıkça gözlemci tekerleklerinin dönme miktarına bağlı olarak konumunu tahmin eder. Robotun tekerlerinin kaymasından dolayı konum ölçümü hatası gerçekleşir. Bununla birlikte, tekerlerle beraber GPS okuması alındığını düşünelim gps okuması kısa mesafe aralıklarında gürültülüdür ve doğru sonuç vermez ancak tekerlerle yapılan ölçüm ise kısa mesafe aralıklarında doğru sonuç verdiği gibi uzun mesafelerde

teker kayması ile birlikte ölçüm sonuçları yanlış olmaya başlar. Bu işlem tekrarlanır ve iki sensör kaynaştırılarak tahmini çıktı iyileştirilir.



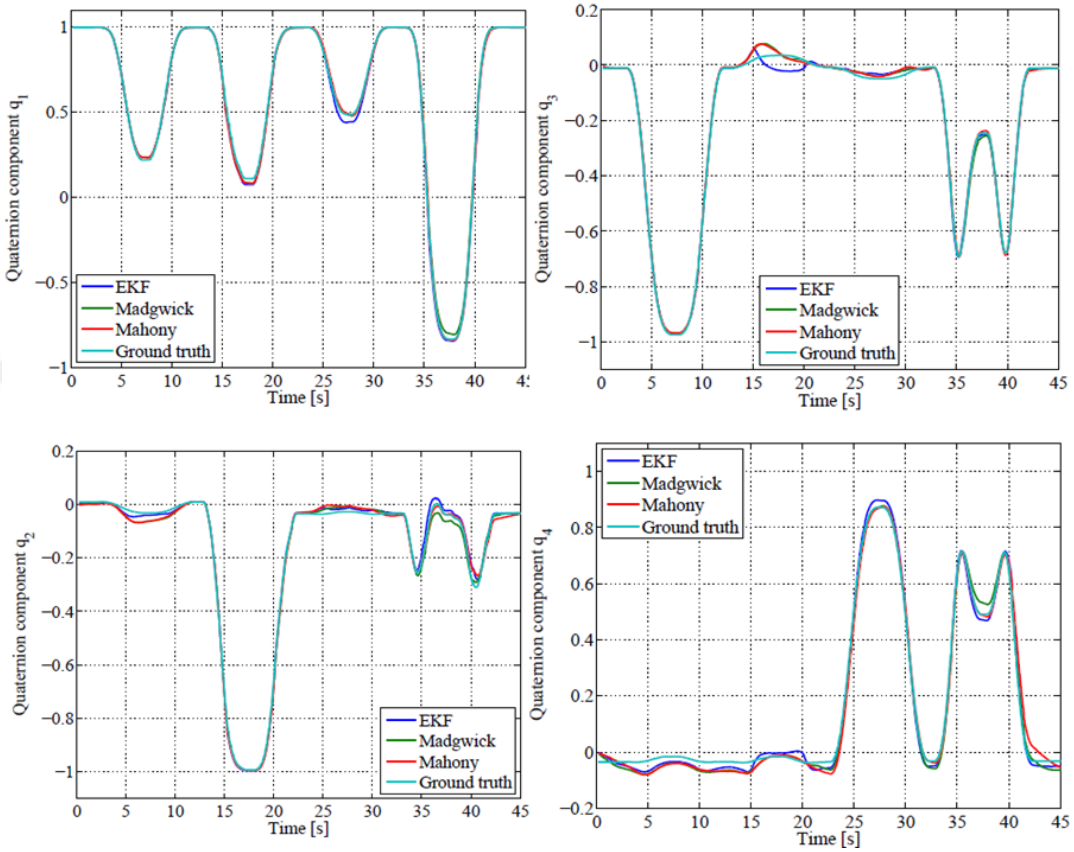
Şekil 3. 1 Kalman filtresi algoritması

Kalman filtresi iki aşamalı bir döngüden oluşur: Şekil 3.1'de gösterildiği gibi güncellenir ve tahmin edilir. Tahmin aşamasında Kalman filtresi belirsizlikleri ile birlikte mevcut durum değişkenlerinin tahminlerini üretir. Güncelleme adımında, Kalman filtresi en son ölçüm kümesini alır, tahmin ile karşılaştırır ve Kalman kazanımı olarak adlandırılan ağırlıklı bir ortalama katsayısı K ile tahmini günceller.



Şekil 3. 2 Lineer kalman filtresi tabanlı pozisyon takip algoritması

Şekil 3.3'de görüldüğü üzere kuaterniyon çıkışlarına göre bütünleyici filtreler ile EKF arasında performans bakımında fark bulunmamaktadır bu sebepten dolayı kullandığımız mikrodenetleyiciye daha az hesaplama yükü getiren bütünleyici filtre uygulaması tercih edilmiştir.



Şekil 3. 3 Kalman ve bütünleyici filtrelerin performans karşılaştırılması (Cirillo vd.[26])

Kalman filtresi iyi bir hata filtreleme algoritmasıdır ancak düşük işlem kapasiteli işlemciler için yoğun bir işlem yükü getirir bu sebepten dolayı bu çalışmada daha düşük işlem yükü ile çalışan ve kalman filtresi kadar etkili olan Madwick'in önerdiği bütünleyici filtre algoritması [19] kullanılmıştır.

3.2 Bütünleyici Filtre

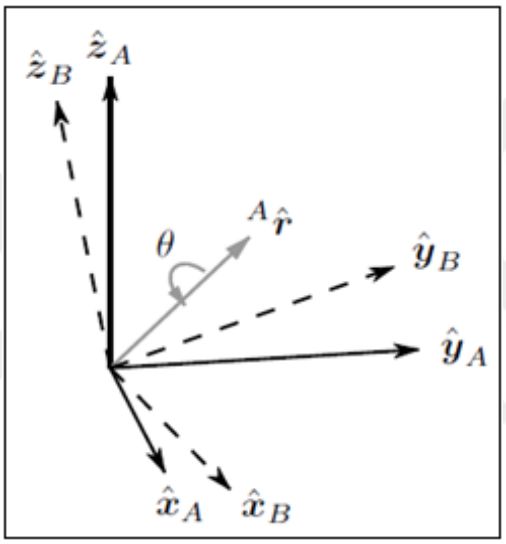
Bir kuaterniyon, sert bir cismin veya koordinat çerçevesinin yönünü üç boyutlu uzayda yönlendirmek için kullanılabilecek dört boyutlu bir karmaşık sayıdır. B çerçevesinin A çerçevesine göre rasgele yönlendirilmesi, A çerçevesindeki ${}^{A}\hat{r}$ ekseni etrafında θ açısının dönmesi ile sağlanabilir. ${}^{A}_{B}\hat{q}$, A çerçevesine göre B çerçevesinin oryantasyonunu tarif

eder ve ${}^{A}\hat{r}$, A çerçevesinde tanımlanan bir vektördür. Kuaterniyon aritmetiği, genellikle, oryantasyonu tanımlayan bir kuaterniyonun ilk önce normalize edilmesini gerektirir. Bu nedenle, bir yönelimi tanımlayan tüm kuaterniyonlar için birim uzunluktadır.

$$ {}^{A}_{B}\hat{q} = [q_1\ q_2\ q_3\ q_4] = [\cos\left(\frac{\theta}{2}\right) - r_x\sin\left(\frac{\theta}{2}\right) - r_y\sin\left(\frac{\theta}{2}\right) - r_z\sin\left(\frac{\theta}{2}\right)] \quad (3.1) $$

* İle gösterilen kuaterniyon konjugatı, bir oryantasyonla tarif edilen bağıl kareleri yerdeğiştirmek için kullanılabilir. Örneğin, ${}^{B}_{A}\hat{q}$, ${}^{A}_{B}\hat{q}$'nun konjugatıdır ve çerçeve A'ya göre A çerçevesinin oryantasyonunu tanımlar. ${}^{A}_{B}\hat{q}$'nin konjugatı, (3.2) ile tanımlanır.

$$ {}^{A}_{B}\hat{q}^{*} = {}^{B}_{A}\hat{q} = [q_1\ -q_2\ -q_3\ -q_4] \quad (3.2) $$



Şekil 3. 4 B çerçevesinin ${}^{A}\hat{r}$ ekseni etrafındaki oryantasyonu

⊗ ile belirtilen kuaterniyon işlemi, bileşik yönelimleri tanımlamak için kullanılabilir. Örneğin, ${}^{A}_{B}\hat{q}$ ve ${}^{B}_{A}\hat{q}$ tarafından açıklanan iki yönelim için, bileşik yönelim ${}^{A}_{C}\hat{q}$ denklemi ile tanımlanabilir.

$$ {}^{A}_{C}\hat{q} = {}^{B}_{C}\hat{q} \otimes {}^{A}_{B}\hat{q} \quad (3.3) $$

İki kuaterniyon için, a ve b, kuaterniyon işlemi Hamilton kuralı kullanılarak belirlenebilir ve () ile tanımlanabilir. Buradaki işlemin değişmeli özelliği yoktur $a \otimes b \neq b \otimes a$

$$ a \otimes b = [a_1\ a_2\ a_3\ a_4] \otimes [b_1\ b_2\ b_3\ b_4] \quad (3.4) $$

$$= \begin{bmatrix} a_1b_1 - \ a_2b_2 - a_3b_3 - a_4b_4 \\ a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3 \\ a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2 \\ a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1 \end{bmatrix}^T$$

Üç eksenli bir jiroskop, sırasıyla x, y ve z eksenlerinin w_x, w_y ve w_z olarak adlandırılan sensor çerçevesinin açısal hızı ölçer. Bu parametreler radyan cinsinden s_w vektörü ile gösterildiğinde (3.5), dünya çerçevesinin sensör çerçevesine göre kuaternion göstermi olarak olarak formülize edilmiş hali olan ${}^S_E\dot{q}$ (3.6)'da gösterilmiştir.

$$s_w = [0 \quad w_x \quad w_y \quad w_z] \tag{3.5}$$

$${}^S_E\dot{q} = \frac{1}{2} {}^S_E\hat{q} \ \circledast \ {}^S w \tag{3.6}$$

Dünya çerçevesinin t zamanındaki sensör çerçevesine göre yönü ${}^E_S\dot{q}_{w,t}$ kuaterniyon türevinin sayısal olarak entegrasyonu ile hesaplanabilir ${}^S_E\dot{q}_{w,t}$. Bu denklemlerde s_{wt} , t anında ölçülen açısal orandır Δt örnekleme periyodudur ${}^S_E\hat{q}_{est,t-1}$ önceki yönelim tahminidir.

$${}^S_E\dot{q}_{w,t} = \frac{1}{2} {}^S_E\hat{q}_{est,t-1} \ \circledast \ {}^S w_t \tag{3.7}$$

$${}^S_E q_{w,t} = \frac{1}{2} {}^S_E\hat{q}_{est,t-1} + {}^S_E\dot{q}_{w,t}\Delta t \tag{3.8}$$

3.2.1 Vektör Gözlemcisi ile Oryantasyon

Bir üç eksenli ivmeölçer, sensörün hareketinden dolayı doğrusal ivme ile birleştirilmiş sensör çerçevesindeki yerçekimi alanının büyüklüğünü ve yönünü ölçecektir. Benzer şekilde, üç eksenli bir manyetometre, yerel manyetik alan ve çarpıtmalarla birleştirilmiş sensör çerçevesinde dünyadaki manyetik alanın büyüklüğünü ve yönünü ölçecektir. Bir oryantasyon bağlamında, başlangıçta bir ivmeölçerin sadece yerçekimini ölçeceği, bir manyetometrenin yalnızca dünyanın manyetik alanını ölçeceği varsayılacaktır.

Dünya manyetik alanının yönü gözlemci çerçevesinden (earth reference frame) biliniyorsa, alanın sensör çerçevesi içindeki yönünün ölçülmesi, sensör çerçevesinin gözlemci çerçevesine göre yönlendirilmesine olanak sağlayacaktır. Bununla birlikte, herhangi bir ölçüm için, benzersiz bir sensör oryantasyon çözümü olmayacaktır, bunun yerine, gerçek oryantasyonun manyetik Alana paralel bir eksen etrafında dönmesi ile elde edilen tüm oryantasyonlarla temsil edilen sonsuz çözümler olacaktır. Bazı

uygulamalarda, eksik bir çözüme izin veren 2 açısı bilinen bir açısı bilinmeyen bir Euler açı gösterimi kullanmak kabul edilebilir. Bilinmeyen açı manyetik alanın yönü ile paralel bir eksen etrafında dönmedir. Sensörün yönelimi bir optimizasyon probleminin formülasyonu yoluyla sağlanabilir.

$$\min f({}^{S}_{E}\hat{q}\,, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s}\,) \tag{3.9}$$

$$f\left({}^{S}_{E}\hat{q}\,, {}^{E}_{\cdot}\hat{d}\,, {}^{S}_{\cdot}\hat{s}\right) = {}^{S}_{E}\hat{q}^{*} \circledcirc {}^{E}_{\cdot}\hat{d} \circledcirc {}^{S}_{E}\hat{q} - {}^{S}_{\cdot}\hat{s} \tag{3.10}$$

$${}^{S}_{E}\hat{q} = [q_1 \quad q_2 \quad q_3 \quad q_4] \tag{3.11}$$

$${}^{E}_{\cdot}\hat{d} = [0 \quad d_x \quad d_y \quad d_z] \tag{3.12}$$

$${}^{S}_{\cdot}\hat{s} = [0 \quad s_x \quad s_y \quad s_z] \tag{3.13}$$

Pek çok optimizasyon algoritması var ancak Gradient descent (gradyan iniş) algoritması, hem uygulaması hem de hesaplaması en basitlerinden biri. Gradient descent, bir fonksiyonun minimumunu bulmak için birinci mertebeden yinelemeli optimizasyon algoritmasıdır. Eğim inişini kullanarak yerel minimum bir nokta bulmak için, geçerli noktadaki fonksiyonun gradyanının negatifiyle orantılı adımlar atılır. (3.14), bir oryantasyon tahmini ile sonuçlanan iterasyonlar için gradient descent algoritmasını tarif eder. Sensörün gradyan iniş algoritması için başlangıç oryantasyonu ${}^{S}_{E}q_{k+1}$, oryantasyonu ${}^{S}_{E}\hat{q}_k$, denklem adımı μ ile ifade edilir. (3.15), nesnel işlev ve Jacobian tarafından belirlenen çözelti yüzeyinin gradyanını hesaplar.

$${}^{S}_{E}q_{k+1} = {}^{S}_{E}\hat{q}_k - \mu\frac{\nabla f({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s})}{\|\nabla f({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s})\|}, k = 0,1,2,\dots n \tag{3.14}$$

$$\nabla f\left({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s}\right) = J^{T}({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}) f({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s}) \tag{3.15}$$

$$f\left({}^{S}_{E}\hat{q}_k, {}^{E}_{\cdot}\hat{d}, {}^{S}_{\cdot}\hat{s}\right) = \begin{bmatrix} 2d_X\left(\frac{1}{2} - q_3^2 - q_4^2\right) + 2d_y(q_1q_4 + q_2q_3) + 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_X(q_2q_3 - q_1q_4) + 2d_y\left(\frac{1}{2} - q_2^2 - q_4^2\right) + 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_X(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + 2d_z\left(\frac{1}{2} - q_2^2 - q_3^2\right) - s_z \end{bmatrix} \tag{3.16}$$

$$J\left({}^{S}_{E}\hat{q}_k, {}^{E}\hat{d}\right) = \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 \\ -2d_xq_4 + 2d_zq_2 & 2d_xq_3 - 4d_yq_2 + 2d_zq_1 \\ 2d_xq_3 - 2d_yq_2 & 2d_xq_4 - 2d_yq_1 - 4d_zq_2 \\ -4d_xq_3 + 2d_yq_2 - 2d_zq_1 & -4d_xq_4 + 2d_yq_1 + 2d_zq_2 \\ 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix} \quad (3.17)$$

(3.14) ile (3.17) arasındaki denklemler, uygulanacak filtre algoritmasının genel formunu tanımlar. Bununla birlikte, alanın yönünün yalnızca küresel koordinat çerçevesinin ana ekseninin 1 veya 2'sinde bileşenlere sahip olduğu varsayılabilirse, denklemler basitleştirilir. Yerçekimi yönünün denklem (3.18)'de gösterildiği gibi dikey z eksenini belirlediğini varsaymak uygun bir kural, olacaktır. Normalize edilmiş ivme vektörü değerleri denklem (3.19)'daki gibidir.

$$E_{\hat{g}} = [0 \quad 0 \quad 0 \quad 1] \quad (3.18)$$

$$S_{\hat{g}} = [0 \quad a_x \quad a_y \quad a] \quad (3.19)$$

$$f_g({}^{S}_{E}\hat{q}, {}^{S}\hat{a}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2\left(\frac{1}{2} - q_2^2 - q_3^2\right) - a_z \end{bmatrix} \quad (3.20)$$

$$J_g({}^{S}_{E}\hat{q}) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix} \quad (3.21)$$

Dünyanın manyetik alanı, bir yatay eksende ve dikey eksende bileşenlere sahip olarak kabul edilebilir. Bu (3.22) ile temsil edilmiştir. Sırasıyla ${}^{E}\hat{b}$ ve normalize edilmiş manyetometre ${}^{S}\hat{m}$ ve ${}^{E}\hat{d}$ ölçümü için ${}^{S}\hat{s}$, (3.24) ile tanımlanan objektif fonksiyonu ve (3.25) ile tanımlanan sadeleştirilmiş Jakobian denklemi verir.

$${}^{E}\hat{b} = [0 \quad b_x \quad 0 \quad b_z] \quad (3.22)$$

$${}^{S}\hat{m} = [0 \quad m_x \quad m_y \quad m_z] \quad (3.23)$$

$$f_g\left({}^{S}_{E}\hat{q}, {}^{S}\hat{b}, {}^{S}\hat{m}\right) = \begin{bmatrix} 2b_x(0.5 - q_3^2 - q_4^2) + 2b_z(q_2q_4 - q_1q_3) - m_x \\ 2b_x(q_2q_3 - q_1q_4) + 2b_z(q_1q_2 + q_3q_4) - m_y \\ 2b_x(q_1q_3 + q_2q_4) + 2b_z(0.5 - q_2^2 - q_3^2) - m_z \end{bmatrix} \quad (3.24)$$

$$J_g\left({}^S_E\hat{q}, {}^E\hat{b}\right)=$$
$$\begin{bmatrix} -2b_z q_3 & 2b_z q_4 & -4b_x q_3 - 2b_z q_1 & -4b_x q_4 + 2b_z q_2 \\ 2b_x q_4 + 2b_z q_2 & 2b_x q_3 + 2b_z q_1 & 2b_x q_2 + 2b_z q_4 & -2b_x q_1 + 2b_z q_3 \\ 2b_x q_3 & 2b_x q_4 - 4b_z q_2 & 2b_x q_1 - 4b_z q_3 & 2b_x q_2 \end{bmatrix} \quad (3.25)$$

Daha önce tartışıldığı gibi, yerçekimi veya yalnızca dünyanın manyetik alanının ölçümü, sensörün benzersiz bir yönelimini sağlamayacaktır. Bunu yapmak için, her iki alanın ölçümleri ve referans yönleri (3.26) ve (3.27) ile tarif edildiği gibi birleştirilebilir.

$$f_{g,b}\left({}^S_E\hat{q}, {}^S_{\cdot}\hat{a}, {}^E_{\cdot}\hat{b}, {}^S_{\cdot}\hat{m}\right) = \begin{bmatrix} f_g\left({}^S_E\hat{q}, {}^S_{\cdot}\hat{a}\right) \\ f_g\left({}^S_E\hat{q}, {}^E_{\cdot}\hat{b}, {}^S_{\cdot}\hat{m}\right) \end{bmatrix} \quad (3.26)$$

$$J_{g,b}\left({}^S_E\hat{q}, {}^E_{\cdot}\hat{b}\right) = \begin{bmatrix} J_g^T\left({}^S_E\hat{q}\right) \\ J_b^T\left({}^S_E\hat{q}, {}^E_{\cdot}\hat{b}\right) \end{bmatrix} \quad (3.27)$$

Optimizasyona yönelik geleneksel bir yaklaşım, her yeni yönelim ve karşılık gelen sensör ölçümleri için hesaplanacak olan birden fazla denklem yinelemesi (3.14) gerektirecektir. Etkili bir algoritma için, her bir yinelemeyi optimal bir değere getirmek için μ adım büyüklüğünü ayarlamak gerekecektir. (3.28) tahmini yönlenme olan ${}^S_E q_{\nabla,t}$'i bir önceki t zaman tabanlı tahmini yönlenme olan ${}^S_E\hat{q}_{kest,t-1}$ 'i hesaplar ∇f ise t zamanında ölçülen ${}^S_{\cdot}\widehat{a_t}$ ve ${}^S_{\cdot}\widehat{m}$ ile tanımlanır. ∇f 'nin formu kullanılan sensöre göre (3.29)'dan seçilir. $\nabla f\ için$ sensör sadece ivmeölçerden oluşuyorsa (3.29)'da 1. yaklaşım eğer ivmeölçer ve manyetometre birlikte kullanılıyorsa 2. yaklaşım tercih edilir. ∇ kuaterniyonun gradyan inişi kullanılarak hesaplandığını gösterir.

$${}^S_E q_{\nabla,t} = {}^S_E\hat{q}_{est,t-1} - \mu_t \frac{\nabla f}{\|\nabla f\|} \quad (3.28)$$

$$\nabla f = \begin{cases} J_g^T\left({}^S_E\hat{q}_{est,t-1}\right) f_g\left({}^S_E\hat{q}_{est,t-1}, {}^S_{\cdot}\widehat{a_t}\right) & (1) \\ J_{g,b}^T\left({}^S_E\hat{q}_{est,t-1}, {}^E_{\cdot}\hat{b}\right) f_{g,b}\left({}^S_E\hat{q}_{est,t-1}, {}^S_{\cdot}\hat{a}, {}^E_{\cdot}\widehat{b}, {}^S_{\cdot}\widehat{m}\right) & (2) \end{cases} \quad (3.29)$$

Optimum μ_t değeri, ${}^S_E q_{\nabla,t}$'nin yakınsama hızını sağlayan değer olarak tanımlanabilir. Bu değer gereksiz yere büyük bir adım boyutu nedeniyle aşmayı önlediği için fiziksel yönelim oranı ile sınırlıdır. μ_t (3.30)'daki gibi hesaplanır Δt örnekleme periyodu ${}^S_E\hat{q}_{w,t}$ jiroskop ile ölçülen fiziksel yönelim oranı α ivmeölçer ve manyetometre ölçümlerinde gürültüyü hesaba katan μ_t'nin katsayısıdır.

$$\mu_t = \alpha \left\| {}^{S}_{E}\hat{q}_{w,t} \right\| \Delta t \, , \alpha > 1 \tag{3.30}$$

3.2.2 Filtre Füzyon Algoritması

Sensör çerçevesinin dünya çerçevesine göre tahmini yönelimi olan ${}^{S}_{E}q_{est,t}$, oryantasyon hesaplamaları olan ${}^{S}_{E}q_{w,t}$ ve ${}^{S}_{E}q_{\nabla,t}$ (3.8) ve (3.28)'de belirtildiği gibi sensör füzyonu algoritması ile birleştirilerek elde edilir. (3.31)'de ${}^{S}_{E}q_{w,t}$ ${}^{S}_{E}q_{\nabla,t}$ yönelim ölçümlerinin füzyonu gösterilmiştir γ_t ve $(1-\gamma_t)$ yönelimlere uygulanan katsayılardır.

$${}^{S}_{E}q_{est,t} = \gamma_t {}^{S}_{E}q_{\nabla,t} + (1-\gamma_t) {}^{S}_{E}q_{w,t} \, , 0 \leq \gamma_t \leq 1 \tag{3.31}$$

Optimum γ_t değerinin hesaplanması (3.32)'de gösterilmiştir. Burada $\frac{\mu_t}{\Delta t}$ ${}^{S}_{E}q_{\nabla,t}$'nın yakınsama oranı β ise jiroskop ölçüm hatasına karşılık gelen bir kuaterniyon türevinin büyüklüğü olarak ifade edilen ${}^{S}_{E}q_{w,t}$ 'nin ağırlıklı sapmasıdır. (3.33), (3.32)'in yeniden ifade edilmiş halidir.

$$(1-\gamma_t)\beta = \gamma_t \frac{\mu_t}{\Delta t} \tag{3.32}$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \tag{3.33}$$

(3.31)-(3.33) ${}^{S}_{E}q_{\nabla,t}$'in yakınsama oranının yönettiği fiziki oryantasyon değişim hızına eşit veya daha büyük olduğunu varsayarsak ${}^{S}_{E}q_{w,t}$ ile ${}^{S}_{E}q_{\nabla,t}$ nin optimal füzyonudur. Bu nedenle α 'nın üst sınırı yoktur. Eğer (3.30)'da α μ_t den çok daha büyük olduğu varsayılırsa oryantasyon filtresi tasarımı basitleşir. (3.28)'de büyük bir μ_t değeri kullanıldığında ${}^{S}_{E}\dot{q}_{est,t-1}$ değeri önemsiz olur ve (3.34)'deki gibi sadeleştirilebilir.

$${}^{S}_{E}q_{\nabla,t} \cong \mu_t \frac{\nabla f}{\|\nabla f\|} \tag{3.34}$$

(3.33)'de belirtilen γ_t eşitliğinde paydadaki β ihmal edilebilir hale gelir ve denklem, (3.35) olarak yeniden yazılabilir.

$$\gamma_t \cong \frac{\beta \Delta t}{\mu_t} \tag{3.35}$$

(3.8),(3.34) ve (3.35)'deki denklemler (3.31)'de yeniden ikame edilirse (3.36) olarak yazılabilir.

$${}^{S}_{E}q_{\nabla,t} = \frac{\beta \Delta t}{\mu_t}\left(-\mu_t \frac{\nabla f}{\|\nabla f\|}\right) + (1-0)({}^{S}_{E}\dot{q}_{est,t-1} + {}^{S}_{E}\dot{q}_{w,t}\Delta t) \tag{3.36}$$

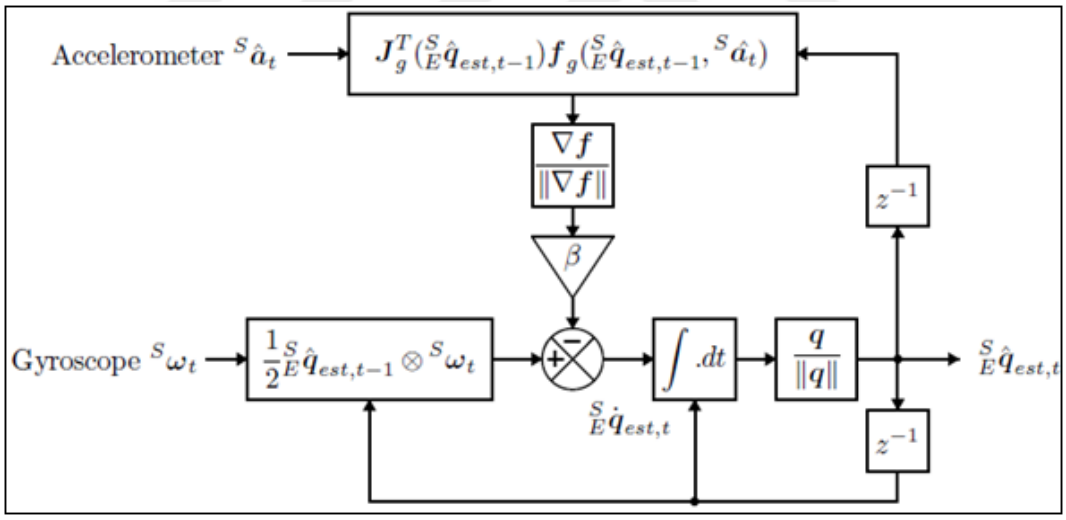
(3.36) basitleştirildiğinde (3.37) olarak yazılabilir burada ${}^{S}_{E}\dot{q}_{est,t}$ (3.38) ile belirlenen oryantasyon değişim hızı tahminidir. ${}^{S}_{E}\dot{\hat{q}}_{est,t}$, (3.39) ile tanımlanmıştır ve ${}^{S}_{E}\dot{q}_{est,t}$ 'nın hatasının yönüdür.

$$ {}^{S}_{E}q_{est,t} = {}^{S}_{E}\hat{q}_{est,t-1} + {}^{S}_{E}\dot{q}_{est,t}\Delta t \tag{3.37} $$

$$ {}^{S}_{E}\dot{q}_{est,t} = {}^{S}_{E}\dot{q}_{w,t} - \beta\, {}^{S}_{E}\dot{\hat{q}}_{est,t}\Delta t \tag{3.38} $$

$$ {}^{S}_{E}\dot{\hat{q}}_{est,t} = \frac{\nabla f}{\|\nabla f\|} \tag{3.39} $$

(3.37)'den (3.38)'e kadar olan denklemlerde görülüdüğü gibi filtre ${}^{S}_{E}q_{est}$ oryantasyonunu tahmini oryantasyon oranı olan ${}^{S}_{E}\dot{q}_{est,t}$'i sayısal integrasyon yöntemi ile hesaplamaktadır. Filtre ${}^{S}_{E}\dot{q}_{est,t}$'i jiroskoplar tarafından ölçülen oryantasyon değişim oranını jiroskop ölçüm hatasının büyüklüğü ile birlikte hesaplar. β ile bu hata tahmini hata yönünde kaldırıldır. ${}^{S}_{E}\dot{\hat{q}}_{est}$ ivmeölçer ve manyetometre ölçümlerinden hesaplanır. Şekil 3.4, bir IMU için oryantasyon filtresi uygulamasının bir blok diyagramını göstermektedir.



Şekil 3. 5 Altı serbestlik dereceli bütünleyici filtre

3.2.3 Manyetik Bozulma Telafisi

Yerkürenin manyetik alanının ölçümleri, manyetometrenin yakınında ferromanyetik elemanların bulunmasıyla bozulacaktır. Oryantasyon sensörünün performansı üzerindeki manyetik bozulmaların etkisine yönelik araştırmalar, elektrikli cihazlar, metal mobilyalar ve bir bina inşaatı içindeki metal yapılar dahil kaynaklar tarafından önemli

hataların ortaya çıkabileceğini göstermiştir. Sensör çerçevesine sabitlenmiş, hard iron biases (sert demir önyargı) adı verilen girişim kaynakları kalibrasyon ile kaldırılabilir. Soft iron distortion (yumuşak demir bozulmaları) olarak adlandırılan yeryüzü çerçevesindeki parazit kaynakları, dünyanın manyetik alanının ölçülen yönünde hatalara neden olur. Sapma hataları, yani yeryüzü yüzeyine göre yatay düzlemde olan sapma hataları, ek bir referans ölçüm olmadan düzeltilemez. Yeryüzü yüzeyine göre düşey düzlemde olan eğim hataları, ivmeölçer sensörünün tutumunun ek bir ölçümünü sağladığı için telafi edilebilir.

Dünyadaki manyetik alanın dünya çerçevesinde t zamanında ölçülen yönü ${}^{E}\hat{h}_t$

$${}^{E}\hat{h}_t = [0 \quad h_x \quad h_y \quad h_z] = {}^{S}_{E}\hat{q}_{est,t-1} \circledcirc {}^{E}\hat{m}_t \circledcirc {}^{S}_{E}\hat{q}^{*}_{est,t-1} \tag{3.40}$$

$${}^{E}\hat{b}_t = \begin{bmatrix} 0 & \sqrt{h_x^2 + h_y^2} & 0 & h_z \end{bmatrix} \tag{3.41}$$

Manyetik çarpılmaların bu şekilde telafi edilmesi, manyetik bozulmaların sadece yönelimin tahmini başlık bileşenini etkilemek için sınırlı kalmasını sağlar. Bu yaklaşım ayrıca, dünyanın manyetik alanının referans yönünün önceden tanımlanması ihtiyacını da ortadan kaldırır; bu diğer yönlendirme filtre tasarımlarının potansiyel bir dezavantajıdır.

3.2.4 Jiroskop Sapması Telafisi

Jiroskop zero bias (sıfır önyargı) zamanla, sıcaklıkla, hareketle sürüklenir ve dünyanın kendi ekseni etrafındaki dönmesine duyarlıdır. Bir IMU veya MARG sensör dizisinin herhangi bir pratik uygulamasında bu hesaba katılmalıdır. Kalman tabanlı yaklaşımların bir avantajı, jiroskop önyargısını sistem modeli içinde ek bir durum değişkeni olarak eklenerek tahmin edebilmeleridir. Ancak, Mahony [17], jiroskop sapması sürüklenmesinin, oryantasyon değişikliği oranında hatanın integral geri beslemesi yoluyla daha basit yönlendirme ile telafi edilebileceğini göstermiştir. Burada da benzer bir yaklaşım kullanılacaktır.

Oryantasyon değişikliği oranındaki tahmini hatanın normalleştirilmiş yönü, ${}^{S}_{E}\dot{\hat{q}}_{\epsilon}$, (3.42) kullanılarak her jiroskop ekseninde açısal hata olarak ifade edilebilir. Jiroskop biası, ${}^{S}w_b$, ${}^{S}w_e$'nin DC bileşeni ile temsil edilir ve uygun bir kazanç, ζ ile ağırlıklandırılarak ${}^{S}w_e$'nin

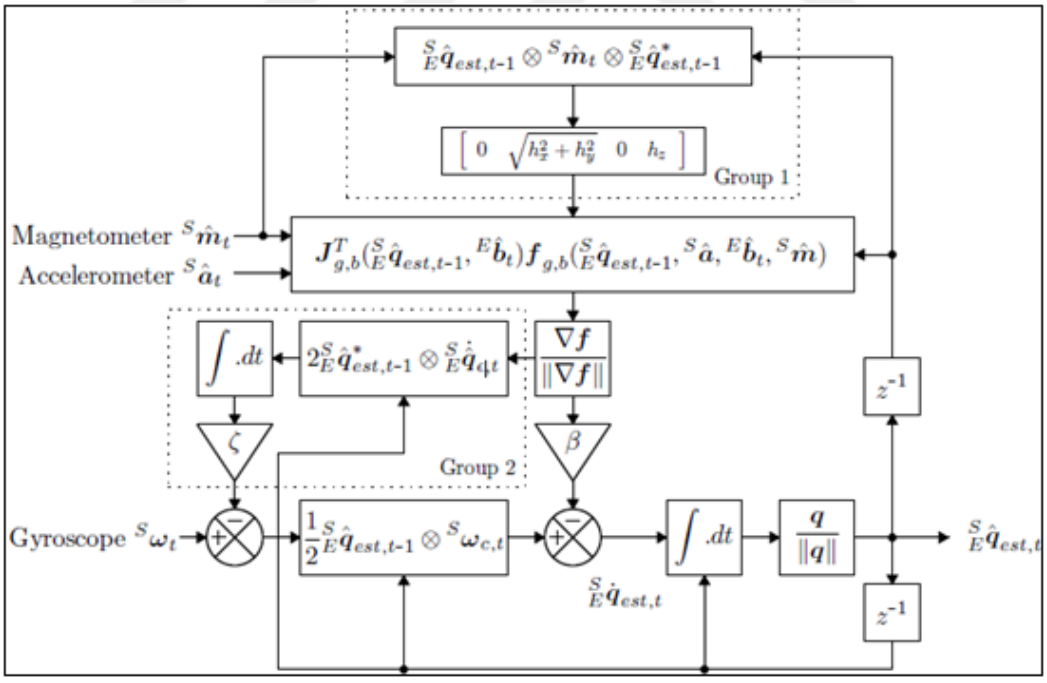
integrasyonundan çıkarılabilir. Bu, (3.43) ve (3.44) 'de gösterildiği gibi, kompanse edilmiş jiroskop ölçümleri olan ${}^{S}w_c$'i verir. ${}^{S}w_c$'nin ilk elemanının her zaman 0 olduğu varsayılır.

$$ {}^{S}w_{e,t} = 2 {}^{S}_{E}\hat{q}^{*}_{est,t-1} \circledast {}^{S}_{E}\dot{\hat{q}}_{\epsilon,t} \tag{3.42} $$

$$ {}^{S}w_{b,t} = \zeta \sum_{t} {}^{S}w_{e,t}\,\Delta t \tag{3.43} $$

$$ {}^{S}w_{c,t} = {}^{S}w_{t} - {}^{S}w_{b,t} \tag{3.44} $$

Kompanse edilmiş jiroskop ölçümleri ${}^{S}w_c$ daha sonra jiroskopi ölçümleri ${}^{S}w$ yerine, (3.6)'da kullanılabilir. Her eksendeki açısal hatanın büyüklüğü, ${}^{S}w_e$, birim uzunluktaki bir kuaterniyon türevine eşittir. Bu nedenle, integral kazancı ζ, tahmini jiroskop biasının(${}^{S}w_b$) yakınsaklık oranını doğrudan belirler, ve bu kuaterniyon türevinin büyüklüğü olarak ifade edilir. Bu işlem tam bir oryantasyon için filtre tahmininin ${}^{S}_{E}\hat{q}_{est}$ kullanılmasını gerektirdiğinden sadece filtrenin MARG uygulaması için geçerlidir. Şekil 3.5, manyetik distorsiyon ve jiroskop sapması sapması telafisi dahil olmak üzere, bir MARG sensör dizisi için komple filtre uygulamasının bir blok diyagramını göstermektedir.



Şekil 3. 6 MARG uygulaması için komple oryantasyon filtresinin blok şeması

3.2.5 Filtre Kazancı

Filtre kazancı β, bir kuaterniyon türevinin büyüklüğü olarak ifade edilen ortalama sıfır jiroskop ölçüm hatalarını temsil eder. Hata kaynakları şunları içerir: sensör gürültüsü, sinyal örtüşme hataları, kuantalama hataları, kalibrasyon hataları, sensör yanlış hizalaması, non-ortogonallık(sensör ekseni dikliksizliği) ve frekans response özellikleri. Filtre kazancı ζ, aynı zamanda bir kuaterniyon türevinin büyüklüğü olarak ifade edilen, sıfır anlamına gelmeyen jiroskop ölçüm hatalarını kaldırmak için yakınsama oranını temsil eder. Bu hatalar jiroskop biasını (önyargısını) temsil eder. Sırasıyla, β ve ζ açısal miktarlarını kullanarak w_β ve $\dot{w}_\zeta$ 'in ince ayarlarının yapılması uygun olup, burada $\widetilde{w}_\beta$, her eksenin tahmini ortalama sıfır jiroskop ölçüm hatasını ve $\dot{w}_\zeta$, her eksendeki tahmini jiroskop sapması sürüklenme hızını temsil eder. (3.6) kullanılarak β 'in herhangi bir birim kuaterniyon olduğu denklem (3.45) ile tanımlanabilir burada $\hat{q}$, birim kuaterniyondur. Benzer şekilde, ζ (3.46)denklemiyle tanımlanabilir.

$$\beta = \left\| \frac{1}{2} \hat{q} \otimes [0 \ \widetilde{w}_\beta \ \widetilde{w}_\beta \ \widetilde{w}_\beta \ ] \right\| = \sqrt{\frac{3}{4}} \widetilde{w}_\beta \tag{3.45}$$

$$\zeta = \sqrt{\frac{3}{4}} \dot{\widetilde{w}}_\zeta \tag{3.46}$$

BÖLÜM 4

İÇ ORTAM SEYRÜSEFER UYGULAMASI

Navigasyon uygulamalarında , Pedestrian Dead Reckoning (PDR), önceden belirlenmiş veya sabit bir pozisyon kullanarak yayaların yörüngesini ve pozisyonunu tahmin etme ve zaman içinde bilinen veya tahmin edilen hızlara veya ivmelere dayanarak bu pozisyonu ilerletme sürecidir.

Önceki bölümlerde, kalman ve bütünleyici filtreler kullanarak tatmin edici bir tutum tahmini elde etmenin mümkün olduğunu gösterilmiştir. Bu projenin ilgili hedefi, yürüyen bir yayanın konumunu tahmin etmektir. Bu bölümde sıkça kullanılan yaklaşımlar ve önerdiğimiz filtre algoritması açıklanacaktır.

4.1 Sıfır Hız Algılama(SHA)

Geleneksel kişisel dead reconing sistemleri, bir pedometre veya bir ivmeölçer kullanarak adımları tespit eder. Daha sonra hangi yöne bakan taraf olduğunu tespit etmek için manyetometre verileri alınır. Bir adım tespit edildiğinde, pozisyon tahminini bir adım ileri o yöne doğru ileriye hareket ettirir. Adım uzunluğu, genellikle, uzuv dönme açısını tespit eden bel veya bacaklara monte edilmiş bir sensör ile hesaplanır.

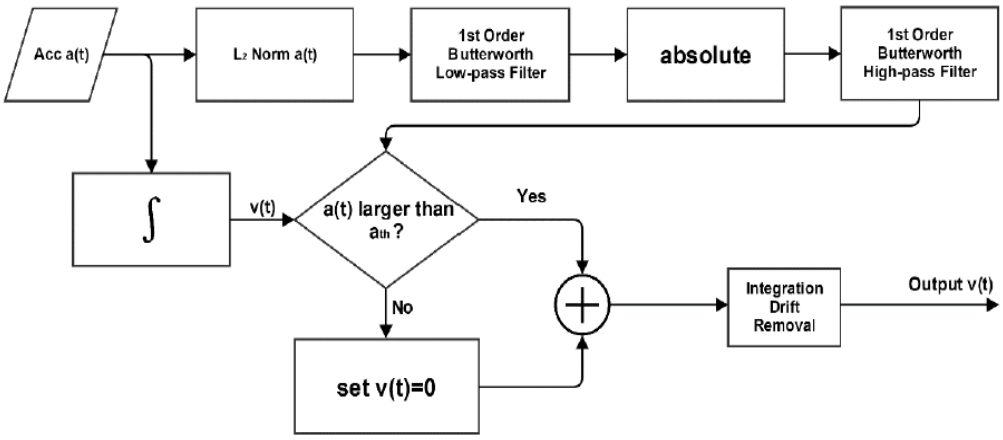
Bu sistemler için, adım uzunluğu önceki tespit edilen adımların ortalamasıdır. Bu nedenle gecikmelerden muzdariptir ve tatmin edici anlık pozisyon tahmini vermez. Dahası, bu sistemler pozisyon tahminini her zaman bir adım öne doğru çıkarır; yürüyen kişi geriye doğru hareket ederse, tahmin edilen pozisyon hala öne doğru gider.

Sıfır hız algılama, ataletsel navigasyon sisteminde hayati bir parçadır. Atalet sensörleri sürüklenmeye maruz kaldıklarından, durağanlık periyotları zaman zaman tespit edilmezse, ivmelenme hataları sert sürüklenmeye yol açan hıza ve konuma entegre edilecektir. Sıfır hız, hızı sıfırlamak için gereken bilgileri sağlar [22].

Sıfır hızı tespit etmek için temelde iki yaklaşım vardır. Birincisi, duruş aşamasını tespit etmek için insanın hareket kalıpları bilgisini kullanmak. Tipik olarak, bu tür yöntemler, tekrarlayan bir dizi toplayıcı vuruşyere basma, durma, ayak kaldırma ve sallanma dizisi olarak yürümeyi modellemektedir. Bu yöntemler yalnızca yürümek için çalışır ve koşma, gezinme veya geriye doğru yürüme gibi hareketleri algılayamaz [23].

İkinci ve daha genel olan seçenek, sadece atalet sensöründen gelen verileri kullanır. Sensör sabitken (okuma 0), hızın sıfır olduğunu ve açısal hızın da 0 olduğunu varsayar. Önemli bir nokta, ivme sabit bir hızda hareket ederse, algoritmanın hareketi hareketsiz olarak yanlış değerlendireceğidir. Ancak, ivmeölçer ivmelere veya kuvvetlere karşı çok hassas olduğundan ve aynı zamanda yürümek oldukça karmaşık bir hızlanma ve yavaşlama süreci olduğundan, sıfır hızın tespiti basit ivme nedeniyle başarısız olması pek mümkün değildir.

4.2 Uygulama



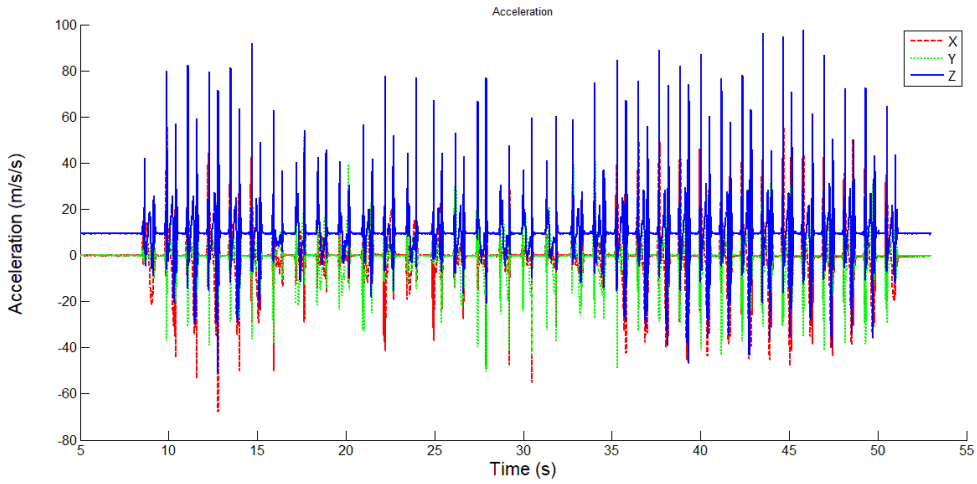
Şekil 4. 1 Sıfır hız algılama blok diyagramı

- İvme büyüklüğünü elde etmek için her üç eksendeki ivmeölçer okumalarındaki l2 normunu hesaplanır.

- Hızlandırmanın büyüklüğünü verileri düzeltmek için birinci dereceden bir Butterworth alçak geçirgen filtreden geçirilir.
- Filtrelenmiş verilerin mutlak değerini hesaplanır.
- DC değerini çıkarmak için verileri birinci dereceden bir Butterworth yüksek geçişli filtreden geçirilir.
- a (t) ivme değeri ampirik olarak belirlenmiş bir eşik değerden daha büyük olup olmadığına karar verilir, büyük ise, hiçbir şey yapılmaz, eğer değilse, v (t) sıfıra ayarlanır. Filtrelenen ivme verileri atılır çünkü tek amacı durağan süreleri tespit etmektir.
- Tüm hız veri örnekleri güncellenene kadar adım 1'den itibaren tekrarlanır.
- Düzeltilmiş v(t) 'yi durağan periyotlar sırasında ve v (t) durağan olmayan periyotlar boyunca bir araya getirerek sıfır hızda v (t) elde edin.
- Sadece durağan periyotlardaki hız sıfıra ayarlandığı için, durağan olmayan periyotlarda sabit integral hala mevcuttur. Ayrıca, 5. adımda, hız eşik değerine göre sıfıra zorlandığı için, kritik noktadan önceki ve sonraki hız süreksiz olacaktır. Bu nedenle entegrasyonun neden olduğu sürüklenme giderilmelidir.

Entegrasyon hızının kaymasını kaldırma yolu: önce durağan olmayan bir dönemin başlangıcı ile bitişi arasındaki hız farkını bulunur, daha sonra, bu durağan olmayan dönemde sürüklenme oranını elde etmek için bu fark örnek sayısına bölünür, daha sonra bu noktadaki sürüklenme değerini elde etmek için sürüklenme hızını karşılık gelen veri indeksi ile çarpılır. Ardından, doğrusal hız elde etmek için kayma önceden hesaplanan hızdan çıkarılır.

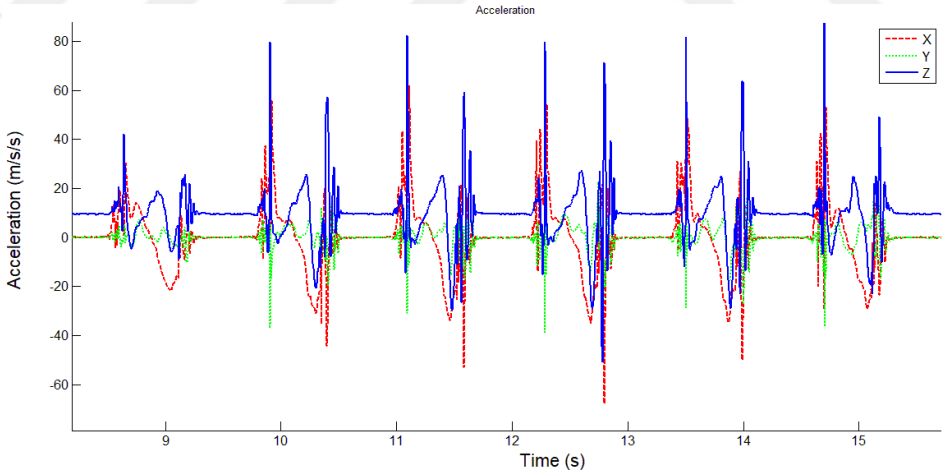
Uygulamada AHRS sensör birimi yürüyen kişinin ayağına sabitlenmiş ve bir koridorda yürürken sensörlerin ham verileri kaydedilmiştir. Yürüyen kişi bir koridor ile merdivene ulaşmış üst kata çıktıktan sonra alt kattaki koridora paralel bir şekilde 25 metre daha yürümüştür AHRS biriminin ivmeölçer verileri şekil 4.2’deki gibi gözlemlenmiştir.



Şekil 4. 2 AHRS ivmeölçer verileri

Şekilde görüldüğü gibi yürüyüş sırasında her bir eksende farklı ivmelenme değerleri oluşmakta ayrıca zaman ekseninde bakıldığında bu ivmelenme değerlerinin tekrarlı bir şekilde devam ettiği görülmektedir. Bu ivmelenme değerleri bir algoritma ile tanımlanırsa kişinin adım atıp atmadığı ayrıca adımın hangi aşamasında olduğu kolaylıkla bulunabilir.

Şekil 4.3'de bu ivme değerleri daha yakından incelenebilmektedir.



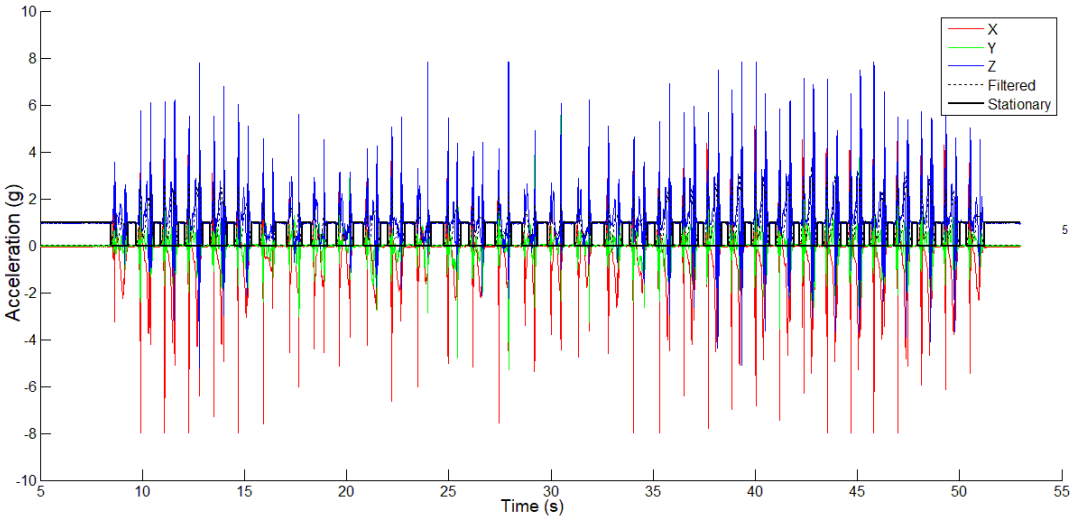
Şekil 4. 3 AHRS ivmeölçer verileri 8-15sn aralığı

İlk olarak, yürümeye başlamadan önce, Z ekseni üzerindeki ivmeölçer okuması X ve Y eksenindeki değerlerden çok daha büyüktür, çünkü yerçekimi nedeniyle ivme artmıştır.

İkinci olarak, X ve Y eksenlerindeki verilerin 0 ile mükemmel şekilde hizalanmadığını gözlemliyoruz. Bu, sensörün ayakkabıya tam olarak dik durmadığını (dünya çerçevesine göre) yerleştirmediğini gösterir, bu nedenle ağırlık bölümleri kesinlikle iyi olan X ve Y

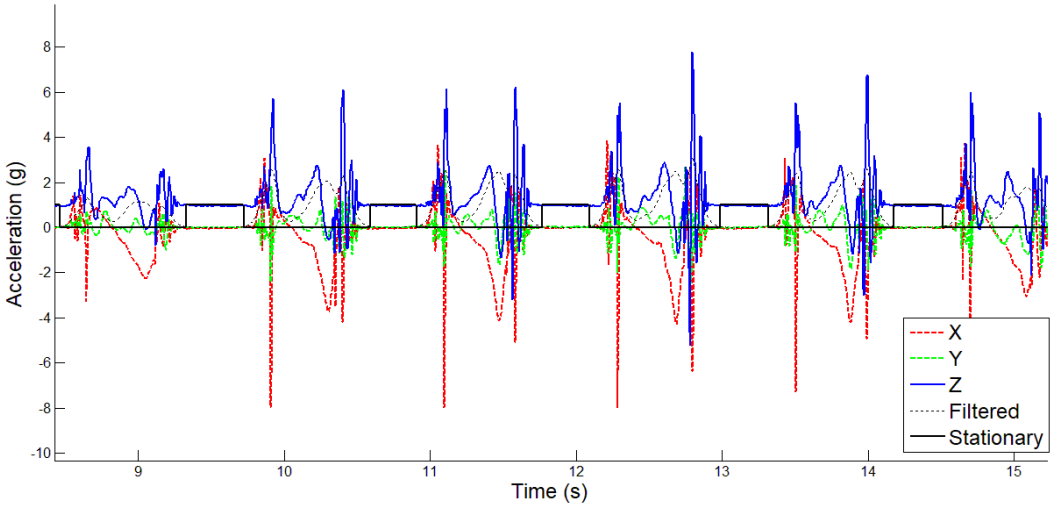
eksenlerinde görülür. Yayalar, sensörleri tamamen dik şekilde takmak zorunda değildir, bu imkansızdır. Sensör çerçevesi ile dünya çerçevesi arasındaki tutarsızlık AHRS algoritması ile telafi edilebilir.

Üçüncüsü, Z ekseni üzerindeki verilerde gürültü veya darbe kuvvetleri veya başka nedenlerden dolayı birkaç ani ivme kolayca gözlenebilir. Bu sivrilikler sıfır hız güncellemesini yanıltabilir ve yanlış kararlara neden olabilir. Bu durum, ivme verisinin karar mantığına gönderilmeden önce düzeltilmesi için low pass bir filtreden geçirilmesi gerekmesinin nedenidir. Bunlarla beraber, her darbe bir basamağı temsil ettiğinden kolayca tanınabilecek darbeler vardır.



Şekil 4. 4 Filtrelenmiş ivme büyüklüğü ve durağan dönemler

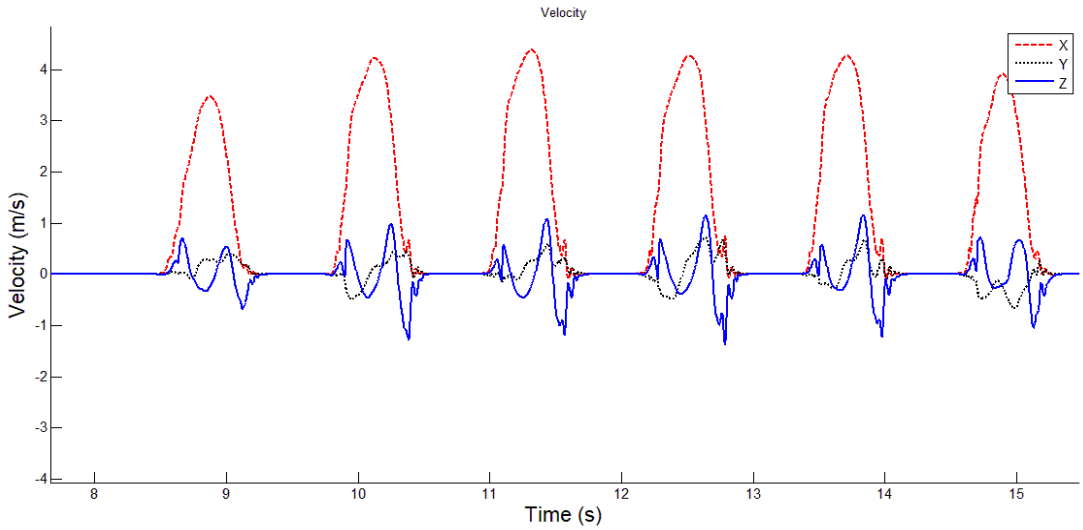
Şekil 4.5'de grafiğe daha detaylı bakıldığında bakılmış halidir.



Şekil 4. 5 Filtrelenmiş ivme büyüklüğü ve durağan dönemler

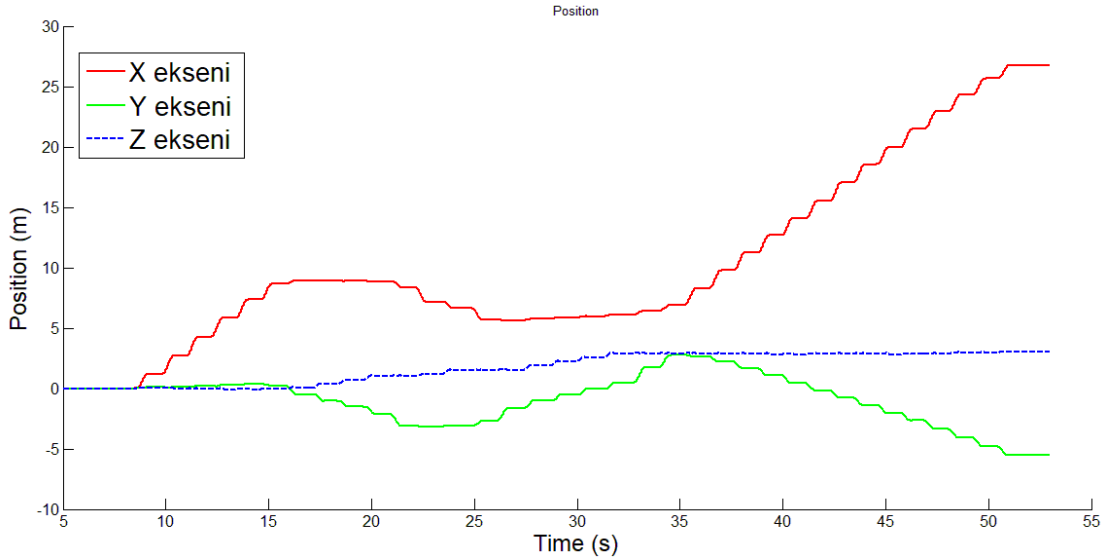
Şekil 4.5'de ham sensör verilerinin low pass ve high pass filtrelerden geçirildikten sonra 0.05g threshold değeri ile yayanın yürüyüş sırasında adımının durağan ve durağan olmayan periyotlarının ayrılmış halidir.

Sıfır hız algılama ile hız tahmini şekil 4.6'de gösterilmektedir. Grafik alçak geçiren filtreden geçirilmiş ivmeölçer verilerinin integrasyonundan hız sürüklenmesinin çıkarılması ile oluşturulmuştur. Grafiğin [13]'de kullanılan sıfır hız algılama algoritmasına göre durağan dönemleri daha iyi ayırt ettiğini görebiliriz. 3 eksende hız, sensörün bağlı bulunduğu ayak yere sabitken sıfıra eşitlenir, ivmelenme 0.05g lik eşik değeri aşınca ayağın adım attığı algılanır ve ivme değerleri ayrık zamanda integral alıcısı ile toplanarak 3 eksende hız değerleri elde edilir. Hız değerlerinin X ekseninde daha fazla görülmesinin nedeni hareket doğrultusunun X ekseni olmasındandır. Z eksenideki hız değeri adım atarken ayağın yerden kalkma anı ve ayağın yere basma anı arasında oluşan hız değeridir Y eksenindeki hız değeri adım sırasında ayağın yalpalaması ve yana kayması gibi etkilerden oluşan hızlanma değeridir.



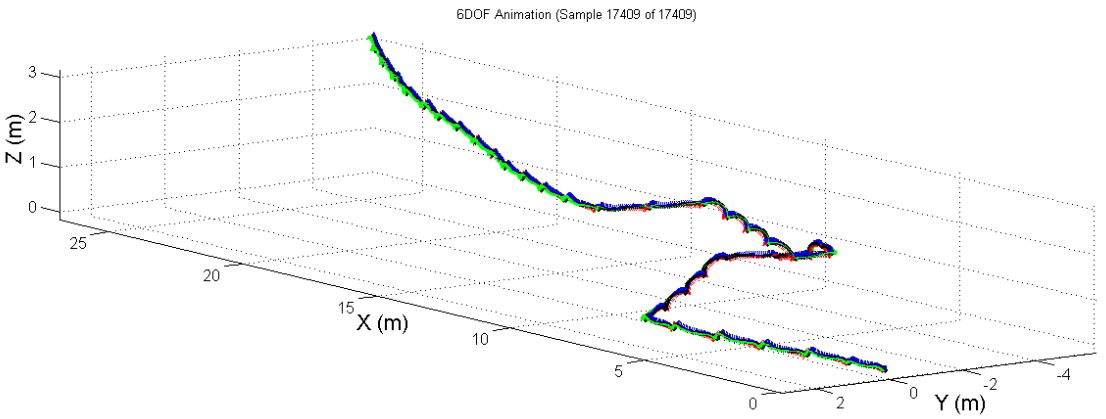
Şekil 4. 6 SHA ile hız tahmini

Grafiğin makul derecede tatmin edici olduğunu görebiliriz. 3 eksende hız darbeler arasındaki sabit dönemlerde sıfıra hizalanır. Kayma ortadan kalkar ve her eksendeki hızın büyüklüğü makuldür, X en yüksek, Y ve Z düşüktür çünkü düz yürüme temelde X ekseni boyuncadır. Bir sonraki hesaplama pozisyon tahmini olacaktır pozisyon da hız verilerinin integrasyonu ile elde edilmektedir.

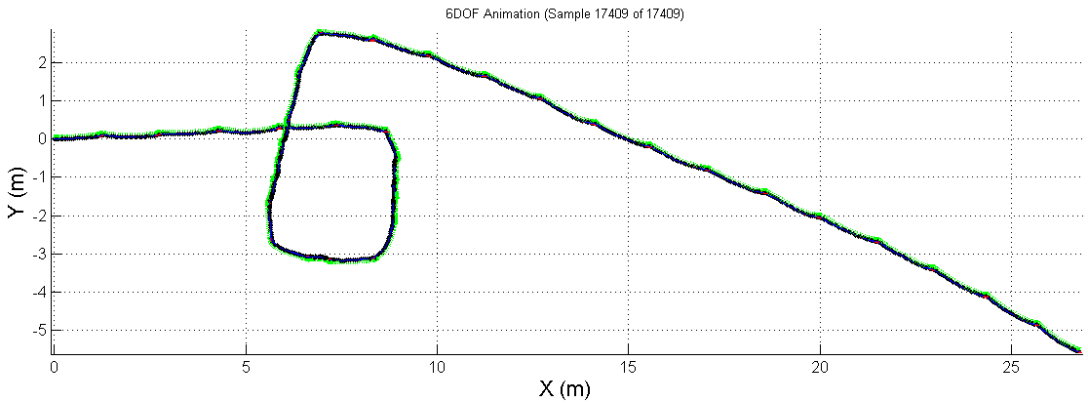


Şekil 4. 7 SHA ile pozisyon tahmini

Şekil 4.7, X ekseni doğrultusunda 26 metrelik mesafenin sıfır hız algılama algoritması kullanarak elde edilen pozisyon tahminini göstermektedir. Sensör integrasyonu sırasında oluşan sürüklenmenin etkilerini Y ekseni boyunca görebiliriz bu eksende meydana gelen sürüklenme X eksenindeki hareketin Y eksenine olan etkisi ile açıklanabilir. Merdivenden çıktıktan sonra Y ekseninde sürüklenme yaklaşık 8 metredir. İdealde Y ekseninde 3 metre civarında hareket olmuştur. X ekseni boyunca sapma yaklaşık 1.5 metredir bu hareket doğrultusudur ve sapma %5.7 civarınadır. Ayrıca eğrilerdeki dalgalanmaları görebiliriz ki bunların her biri yaya tarafından atılan adımı temsil eder. Z ekseni 3 metre yükselmiştir bu da kat yüksekliği ile uyuşmaktadır.



Şekil 4. 8 Yürüyüş rotasının X-Y-Z ekseninlerinde gösterimi

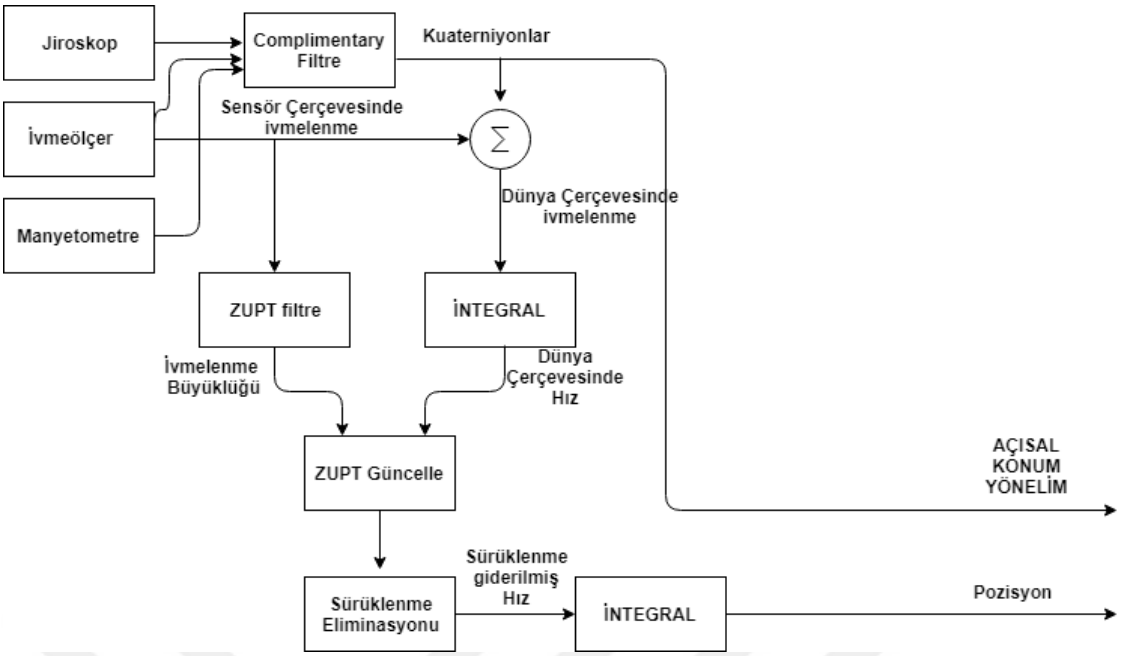


Şekil 4. 9 Yürüyüş rotasının X-Y ekseninde gösterimi

Ayrıca dikkate değer bir fenomen vardır, doğrusal yer değiştirme sadece X ekseni üzerinde değildir, Y üzerinde de biraz etkisi vardır. Bu muhtemelen sensörün dik olarak yerleştirilmediğinden kaynaklanmaktadır. Ancak bu çok önemli bir mesele değildir çünkü pozisyon noktaları arasındaki göreceli ilişki devam ettiği sürece bu sapmayı istediğimiz zaman düzeltebiliriz. Bu bölümde zero velocity update (sıfır hız güncellemesinin) uygulaması gösterilmiştir. Aynı zamanda sensörün neden ayakkabılara takıldığı sorusuna da cevap verilmiştir.

4.3 Heuristic Drift Elimination (HDE)

Önceki bölümlerde, kuaterniyonlar ve bütünleyici ve kalman filtresi açıklanmış ve değerlendirilmiştir. Ayrıca, pozisyon tahmin uygulamasında sıfır hız güncelleme yöntemi uygulanmıştır. Bu iki teknik, temel bir fonksiyonel yaya izleme algoritmasını içeren bir blok şemasında şekil 4.10' da gösterilmiştir.



Şekil 4. 10 AHRS + SHA uygulaması

SHA algoritması tek başına doğrusal bir rotada başarılı işler çıkarabilir ancak şekil 4.9'da görüldüğü üzere merdiven gibi görece daha karmaşık bir rotadan sonra Y ekseninde gözle görülür bir sürüklenmeye maruz kalmıştır. İşte bu noktada tezin çalışma konusu olan sezgisel sürüklenme eliminasyonu algoritması devreye girecektir.

MEMS jiroskoplarındaki önyargı ve hataları gidermek için 2010 yılında Johann Borenstein ve Lauro Ojeda tarafından Heuristic Drift Elimininationn (HDE) adı verilen bir yöntem önerildi [20].

Sıfır hız güncellemesinde olduğu gibi, mevcut algoritmalar kaymayı düzeltmek için yeterli olmadığından, daha fazla bilgi sağlamak için dış gerçekleri kullanırız. SHA, yürüme sırasında ayaklardan herhangi birinin tekrarlayan bir durma ve hareket sırasına maruz kalmasından yararlanır. HDE, çoğu binada, koridorların birbirine paralel veya dik olması gerçeğini göz önünde bulundurur, böylece HDE koridorların baskın yönlerini yönlendirir.

4.4 Sezgisel Kayma Eliminasyonu (SKE)

Bu tezde, 3 boyutlu olarak çalışan ve yapılandırılmış iç mekan ortamında gyro sürüklenmesini tamamen kaldırabilen yeni bir HDE algoritması önerilmektedir. Johann Borenstein ve Lauro Ojeda'nın HDE'si ile karşılaştırıldığında, SKE algoritması her biri

sadece bir kaç dolara mal olan çok daha ucuz MEMS IMU'ları üzerinde çalışırken, Johann Memsense Inc. tarafından üretilen 1300 dolarlık Nano IMU adlı bir sensörü kullanmıştır.

SKE genel olarak üç bölümden oluşmaktadır: yaya hareketi tespiti, baskın yön hesaplaması ve konum güncellemesi.

4.4.1 Yaya Hareket Algılaması

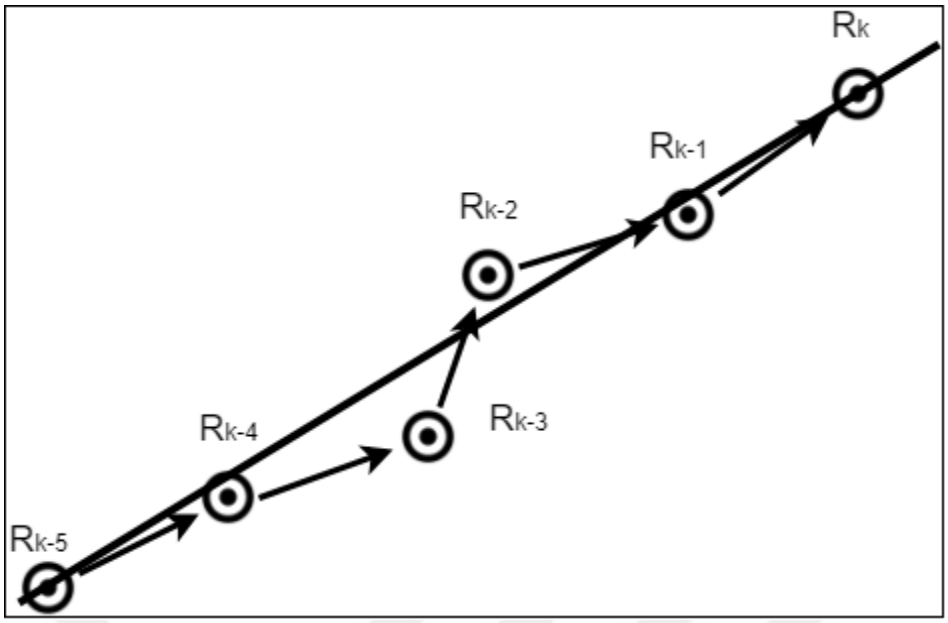
Sürüklenmenin ortadan kaldırılmasının başlangıcında, SKE verinin ilk beş adımını sayar ve beş adımın düz bir çizgi halinde olup olmadığına karar verir. Düşük maliyetli bir MEMS IMU kullandığımız için jiroskop gerçekten hızlı sürüklenmektedir. Bu yüzden daha fazla adım saymak jiroskop hatalarını yanında getirecektir.

Yayanın düz yürümesine karar vermek için, önce adım vektörlerini konum verilerinden hesaplarız. Sonra ilk beş adım sayılır ve bitişik iki adım arasındaki açı hesaplanır. Tüm açıların mutlaklığı bir eşikten azsa, yaya düz yürüyor demektir. Olmazsa, yaya düz bir çizgide yürümüyordur.

Yaya düz yürümüyorsa, SKE hiçbir şey yapmaz. Yaya düz bir çizgide yürüyorsa, SKE baskın yön hesaplaması olan bir sonraki adıma geçer.

4.4.2 Baskın Yön Hesaplaması

Baskın yön hesaplaması iki adımdan oluşur: ilk adım, hakim yönlerin X-Y düzlemindeki çıkıntılarını hesaplamaktır. İkinci adım, 3B alanda baskın yönler oluşturmaktır. Baskın yön projeksiyonu yukarıda belirtilen beş adımdan hesaplanır. Beş kademeli vektörler 6 konumdan hesaplanır. Şekil 4.11'de gösterildiği gibi, X-Y düzlemindeki ilk 6 konumun izdüşümleri boyunca dik dengelemeye dayalı düz bir çizgiye uyan doğrusal regresyon gerçekleştiriyoruz.



Şekil 4. 11 Önceki 5 adımın pozisyonu ve mevcut adım

X-Y düzleminde düz bir çizgi varsayıyoruz bu çizgi y =ax+c formunda baskın yönü belirtir.

Analitik düzlemde A(X_1, Y_1)noktasının $ax + by + c = 0$ doğrusuna olan uzaklığı

$$|AH| = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} \tag{4.1}$$

formülü ile bulunur.

Dolayısıyla şekil 4.11'deki noktalardan düz çizgiye olan mesafe şöyledir:

$$d_i = \frac{|y_i \ - (c + ax_i)|}{\sqrt{1 + a^2}} \tag{4.2}$$

Buradan, mesafelerin karelerinin toplamı:

$$D = \sum_{i=k-5}^{k} d_i^2 \tag{4.3}$$

Buradan, en uygun çizginin a ve c katsayıları aşağıdaki denklemler çözülerek bulunabilir:

$$|AH| = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} \tag{4.4}$$

$$\frac{\partial \mathrm{D}}{\partial \mathrm{c}} = \frac{2}{1 + a^2} \sum_{i=k-5}^{k} (y_i - (c + ax_i))^2(-1) = 0 \tag{4.5}$$

$$\frac{\partial \mathrm{D}}{\partial \mathrm{a}} = \frac{2}{1+a^2}\sum_{i=k-5}^{k}(y_i - (c + ax_i))^2(-x_i) + \sum_{i=k-5}^{k}\frac{(y_i-(c+ax_i))^2(-1)(2a)}{(1+a^2)^2} = 0 \quad (4.6)$$

$$c = \bar{y} - a\bar{x} \quad (4.7)$$

$$a = B \pm \sqrt{B^2 + 1} \quad (4.8)$$

$$B = \frac{\left(\sum_{i=k-5}^{k} y_i^2 - \frac{1}{6}(\sum_{i=k-5}^{k} y_i)^2\right) - (\sum_{i=k-5}^{k} x_i^2 - \frac{1}{6}(\sum_{i=k-5}^{k} x_i)^2)}{2(6\overline{xy} - \sum_{i=k-5}^{k} x_i y_i)} \quad (4.9)$$

Böylece, baskın yönün X-Y düzlemi üzerindeki yansımasını [1, c] alırız.

3B alanda baskın yönü hesaplamak için farklı şekilde ele alınması gereken iki durum vardır. Binalardaki katlar genellikle sadece merdiven veya asansörlerle birbirine bağlandığı için, bir yayanın yavaş yavaş inmesi veya yükselmesi mümkün değildir. Sonuç olarak sadece iki durum mümkündür: yatay yürüyüş veya merdiven çıkma.3D baskın yönünün belirlenmesi aşağıda açıklanmıştır.

A. **Yaya düz bir zeminde yürüyor**: Geçerli adım vektörünün üçüncü elemanının, yani Z ekseni üzerindeki dikey elemanın mutlaklığını bir eşikten geçirilir. Eşik değerin altındaysa yaya'nın düz bir zeminde yürüdüğünü söyleriz. Yani 3B'de baskın yön [1 c 0]. Bu yalnızca yayaların başlangıçta XP'ye (X ekseni üzerinde pozitif) olan yüzüne dönük yönüdür. Sağ el koordinatlar kuralına göre, diğer üç baskın yönü sırasıyla YP (Y ekseninde pozitif), XN (X ekseninde negatif), YN (Y ekseninde negatif) olarak adlandırılan XP'den kolayca hesaplayabiliriz.

   YN = $[1 \quad -\frac{1}{c} \quad 0]$, XN = $[-1 \quad -c \quad 0]$, YP = $[-1 \quad \frac{1}{c} \quad 0]$

B. **Yaya merdivenlerde yukarı ya da aşağı yürüyor:** Mevcut adım vektörünün Z bileşeninin mutlaklığı eşikten büyükse, yürüyenlerin merdiven çıkacağını belirleriz. Bu durumda, baskın yönün XY koordinatlarını koruyoruz ve baskın yönü, mevcut adımla aynı dradyana sahip olacak şekilde yatırıyoruz**.**

Mevcut adımın vektör ile temsil edildiğini varsayalım: $step_i = [x_i \quad y_i \quad z_i]$

Dolayısıyla mevcut adımın gradyanı:

$$g = \frac{z_i}{\sqrt{x_i^2 + y_i^2}} \quad (4.10)$$

Bu nedenle, gradyanı korumak için, egemen yön XP, basitçe g'yi XP'nin ilk iki elementinin normuyla çarpmak suretiyle bulunabilir:

$$XP = \begin{bmatrix} 1 & c & g * \sqrt{1^2 + a^2} \end{bmatrix} = \begin{bmatrix} 1 & c & \frac{z_i\sqrt{1+a^2}}{\sqrt{x_i^2+y_i^2}} \end{bmatrix} \quad (4.11)$$

Benzer şekilde, diğer üç baskın yön:

$$YN = \begin{bmatrix} 1 & -\frac{1}{c} & \frac{z_i\sqrt{1+\frac{1^2}{c}}}{\sqrt{x_i^2+y_i^2}} \end{bmatrix} \quad (4.12)$$

$$XN = \begin{bmatrix} -1 & -c & \frac{z_i\sqrt{1+c^2}}{\sqrt{x_i^2+y_i^2}} \end{bmatrix} \quad (4.13)$$

$$YP = \begin{bmatrix} -1 & \frac{1}{c} & \frac{z_i\sqrt{1+\frac{1^2}{c}}}{\sqrt{x_i^2+y_i^2}} \end{bmatrix} \quad (4.14)$$

4.4.3 Pozisyon Güncelleme

SKE'nin son adımı pozisyon güncellemesidir. Bu adımda, ilk olarak SKE mevcut adım ile her baskın yön arasındaki yönelim farkını hesaplar ve yönelim farkının bir eşikten düşük olup olmadığını kontrol eder. Eşik küçük tutulur, böylece baskın yönü şaşırma olasılığı düşer. Mevcut adım, baskın yönlerin herhangi bir olası aralığına ait değilse, SKE hiçbir şey yapmaz.

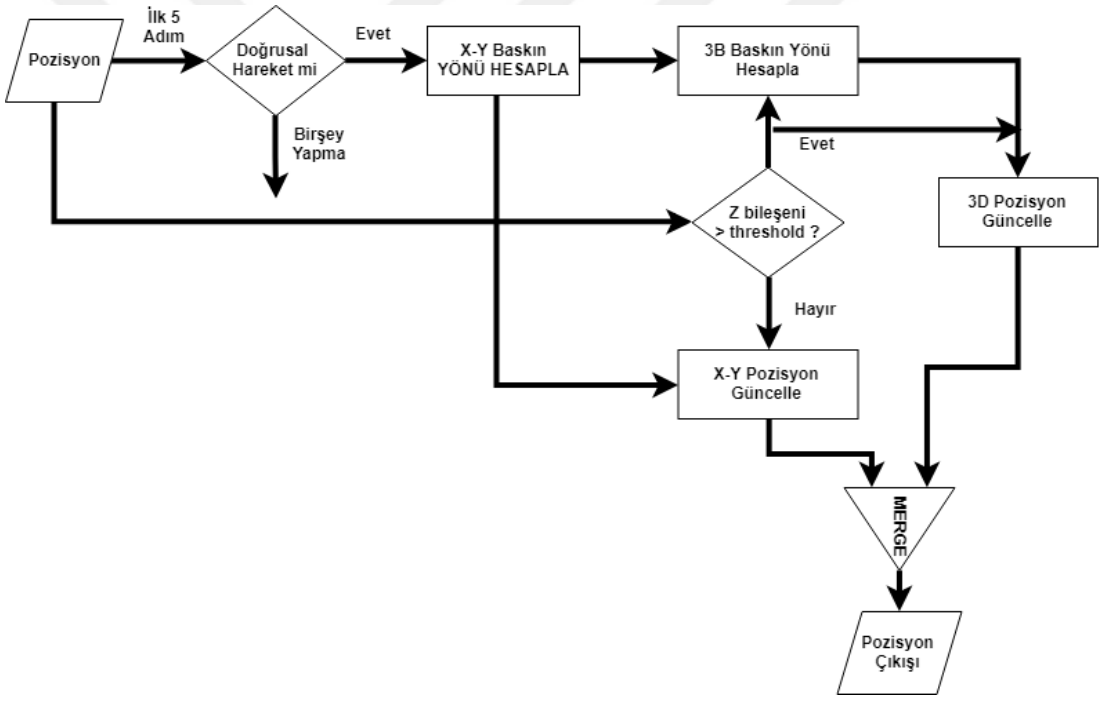
Mevcut adım dört baskın yönden birine düşerse, aşağıdaki adımlar gerçekleştrilir

Adım 1: EK B' deki yazılımda verilen geçerli adım ile baskın yön arasındaki kuaterniyon "HDEquat" hesaplanır

Adım 2: HDEquat kullanılarak geçerli adımın tüm pozisyon noktaları baskın yöne doğru döndürülür. Tüm noktaları döndürmenin nedeni, düşük maliyetli bir MEMS IMU kullandığımız için oldukça hızlı bir şekilde kaymasıdır. Sadece bir adım vektör düzeltilirse, takip eden adımların başka bir baskın yönün bölgesine sürüklenmesi muhtemeldir.

Adım 3: Kuaterniyon rotasyonu belirli bir rotasyon etrafında değil sabit bir eksen etrafındaki belirli derecelerde bir dönüşdür. Bu yüzden rotasyondan sonra, kalan yörünge önceki adıma paralel olmasına rağmen, sürekli değildir. Bu yüzden son adım, yörüngeyi sürekli hale getirmek için “kesme noktası” nın iki ucu arasındaki farkı çıkarmak zorundadır.

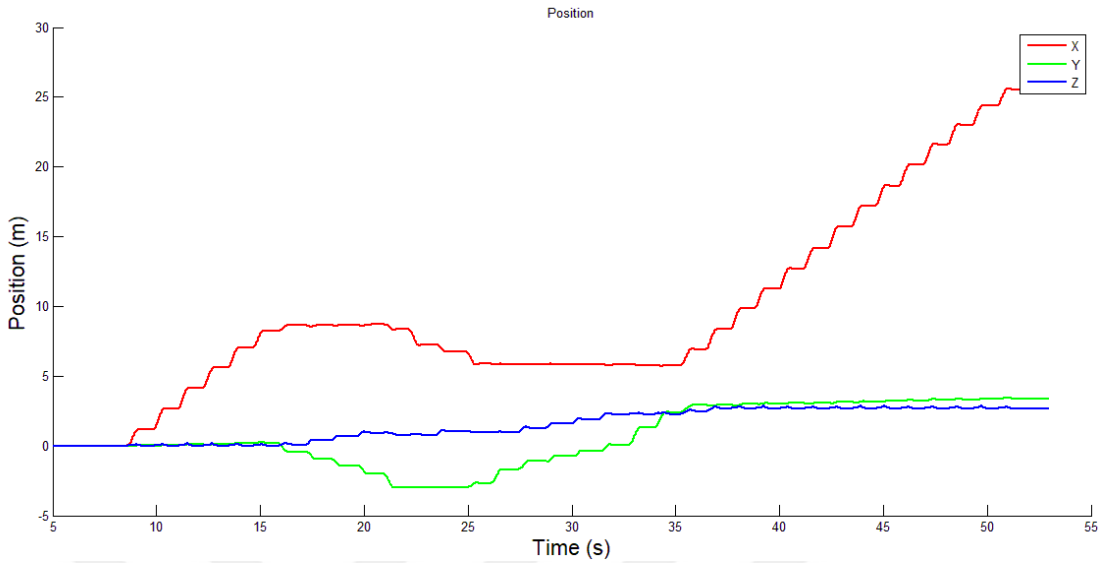
SKE 'nin blok şeması, Şekil 4.12'de gösterilmiştir. Blok diyagramdan SKE 'nin pozisyon verilerini aldığını ve güncellenmiş pozisyon verilerini verdiğini görebiliriz. Bu AHRS algoritmasından bağımsızdır. Baskın yönün sadece X ve Y eksenlerinde tutulmasının ve aynı gradyanın Z ekseninde kalmasının nedeni Z eksenindeki kayma önemsizdir. Genellikle katlar arasında en fazla 20 basamak vardır, bu nedenle kayma varsa kabul edilebilir bir aralıktadır.



Şekil 4. 12 SKE blok diyagramı

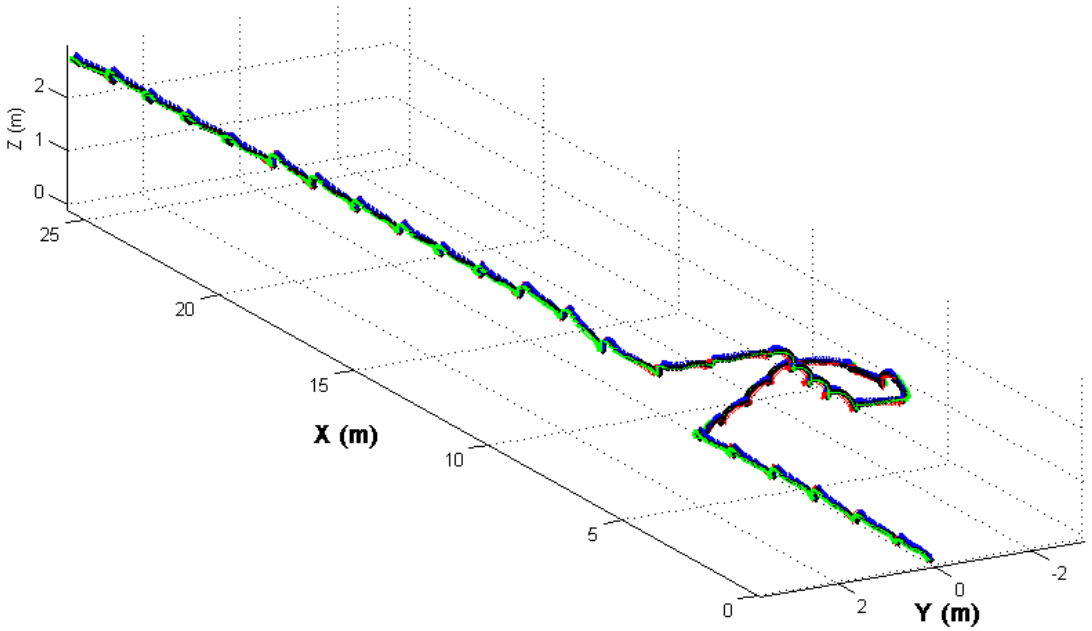
Sezgisel kayma eliminasyonu algoritması devreye girdiğinde ise ilk olarak şekil 4.13’de pozisyon grafiğini inceleyecek olursak Y eksenindeki sürüklenmenin şekil 4.7’ ye göre ortadan kalktığını görüyoruz. Algoritma baskın yönü doğru hesaplamış ve Y eksenindeki kaymayı gidermiştir. Uyguladığımız yöntem [13]’de uygulanan merdiven çıkma testine benzerlik göstermekte söz konusu makalede uygulanan yöntemde pozisyon hatası %1.61 olarak verilmektedir. Pozisyon hataları kullandığımız algoritma ile düzeltildiğinde

X ekseninde 25.7 metre ilerlendiği ölçülmüş böylece X eksenindeki kayma 0.3 metre civarına inmiştir bu da X ekseninde %1.15 hataya inildiğini göstermektedir.

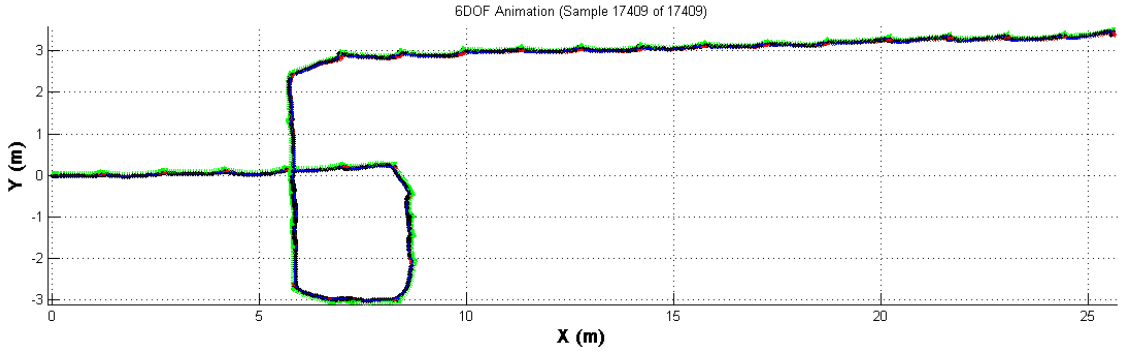


Şekil 4. 13 SKE+SHA pozisyon tahmini

Şekil 4.14'deki grafik MATLAB ortamında çizilmiş olup önerilen algoritmanın kullanımından sonra toplanan verilerle oluşturulmuş 3 boyutlu bir gösterimdir. Yayanın bina içinde aldığı yol kullanılan merdiven ve koridor açık şekilde belli olmaktadır. Kullandığımız algoritmanın nasıl bir düzeltme yaptığını daha iyi anlamak için şekil 4.9 ve şekil 4.15 incelendiğinde Y ekseninin baskın yönde düzeltildiği görülmektedir. [13] ve [24]'de kullanılan test güzergahı bizim verileri elde ettiğimiz güzergaha göre daha düzlemseldir. Bu bakımdan sensörlerimizin sürüklenmeye uğraması daha muhtemel olmasına rağmen iyi bir performans sergilemiştir. [23]'de yapılan testlerde daha uzun yolda ancak düz zeminde yürüme testi yapılmıştır bu sebeple sistemimizle tam performans karşılaştırması yapılamamıştır. [20]'de yapılan navigasyon sisteminde 2,065 metre boyunca test edilmiş ve %0.26 hata ile test tamamlanmıştır bu sebeple performans bakımından bizim sistemimizden ileridedir.



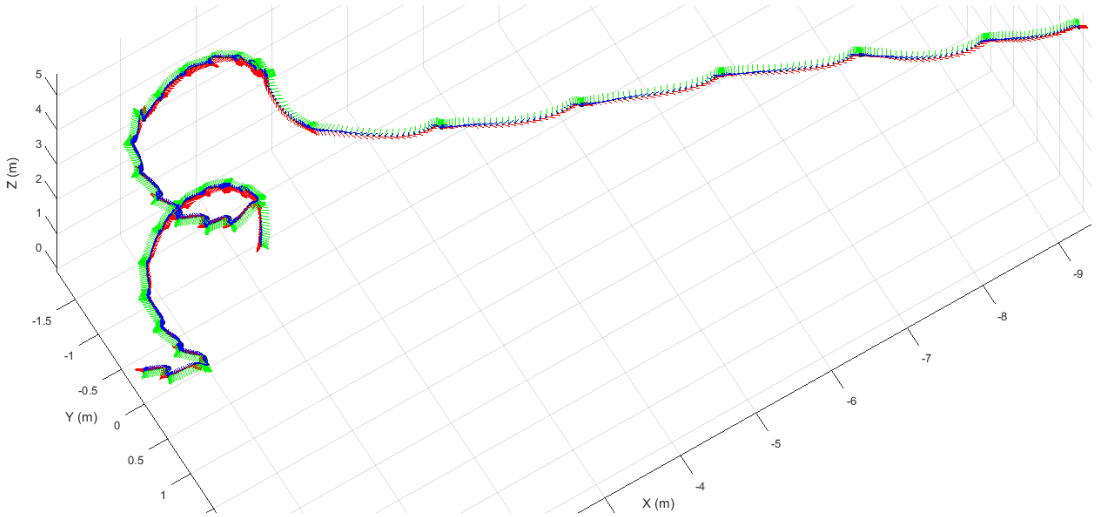
Şekil 4. 14 SKE+SHA pozisyon 3B gösterimi



Şekil 4. 15 SKE+SHA pozisyonun X-Y ekseninde görünümü

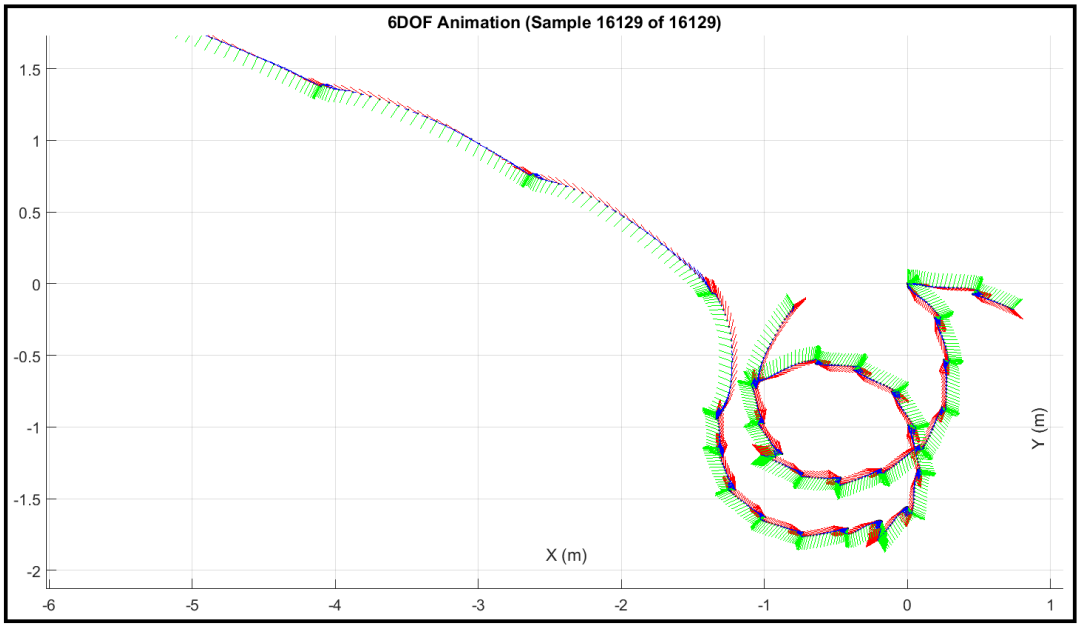
Sezgisel kayma eliminasyonu filtresi doğrusal yöndeki hareketleri algılamaya ve düzeltmeye yönelik çalışmaktadır adımlar arasında doğrusallık bulunduğunda algoritma konumu baskın yönde düzeltmektedir bu düzeltmenin dairesel yollarda bozucu bir etkisi olacağı tahmin edildiğinden çalışmada dairesel ve spiral yol testi yapılmıştır.

Lineer yolların yanı sıra lineer olmayan yollardaki konum tahmin performansını ölçmek için spiral bir merviden testi yapılmıştır. Spiral dönme testinde 5'li adım guruplarının etkisi şekil 4.16 ve 4.17' de görülmektedir ilk dönüşten sonra sistem stabilitesini kaybetmiştir, konum tahmini merdivenin dışına taşmış ve yol üzerindeki süreklilik kaybedilmiştir. Bu durum filtrenin dezavantajı olarak görülebilir ancak yine de önlenebilir bir durumdur.



Şekil 4. 16 Beşli adım gurubu ile SKE kullanılarak yapılan spiral merdiven testi

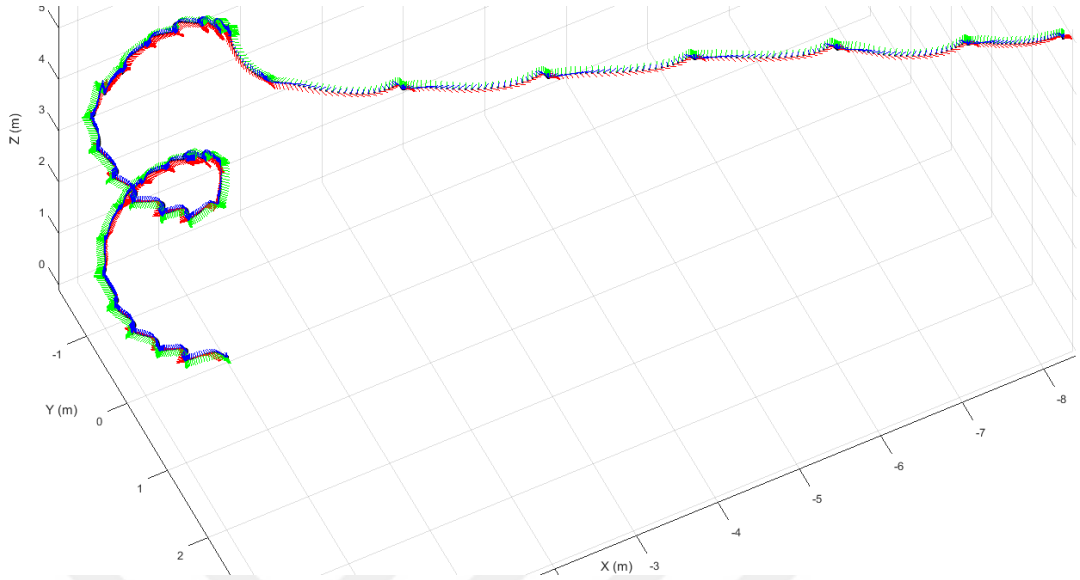
Şekil 4.17'de X-Y düzleminde baktığımızda yol tahminindeki süreksizlik görülmektedir.



Şekil 4. 17 Beşli adım gurubu ile spiral merdiven testi X-Y düzleminde görünüş

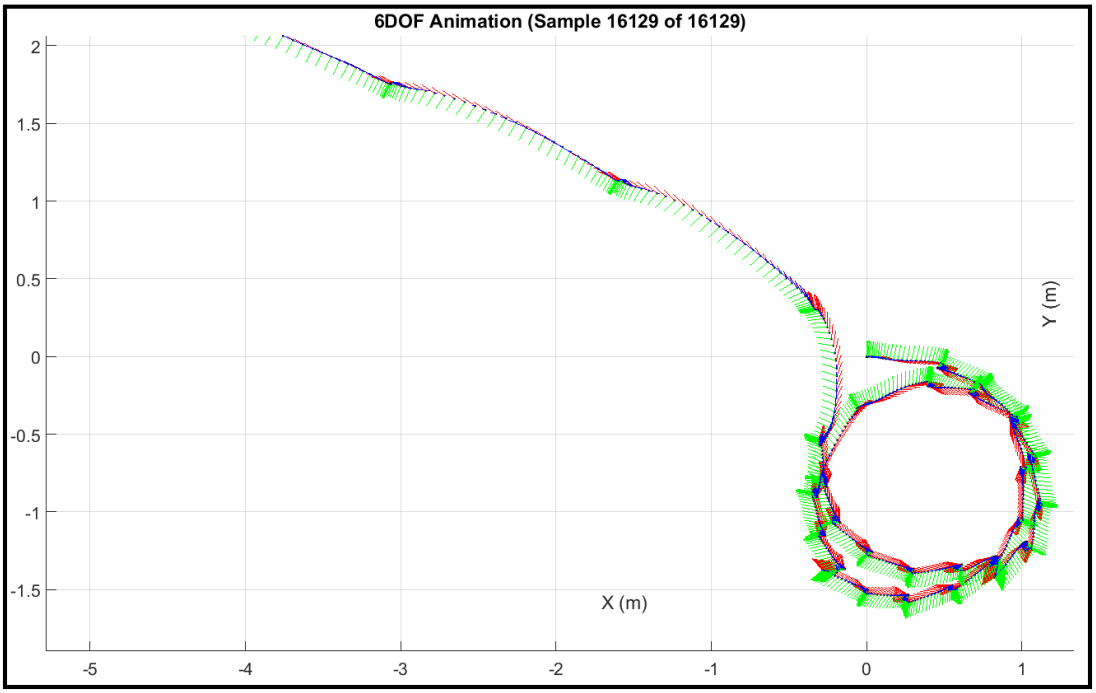
Algoritmanın 5 adımlık konumu baskın yönde düzeltmeye çalışması spiral ve dairel yollarda büyük hatalara sebebiyet vermektedir bu sebeple baskın yol arama adım sayısı 2 ye düşürülmüştür böylece dairesel yollarda daha az düzeltme uygulanmıştır şekil 4.18'de görüldüğü üzere konum tahmininde dairesel form bozulmamıştır. Ancak bu testte de bir tam dairesel dönüş sonucunda sistemin tahmini tam olarak belli olmamaktadır Z ekseninden kaynaklanan etkiyi kaldırıp yayanın düz zeminde daire

etrafında yürüyerek başladığı yere gelmesi bize daha doğru bir tahmin performansı verisi sunacaktır bu sebeple şekil 4.20'deki test yapılmıştır.



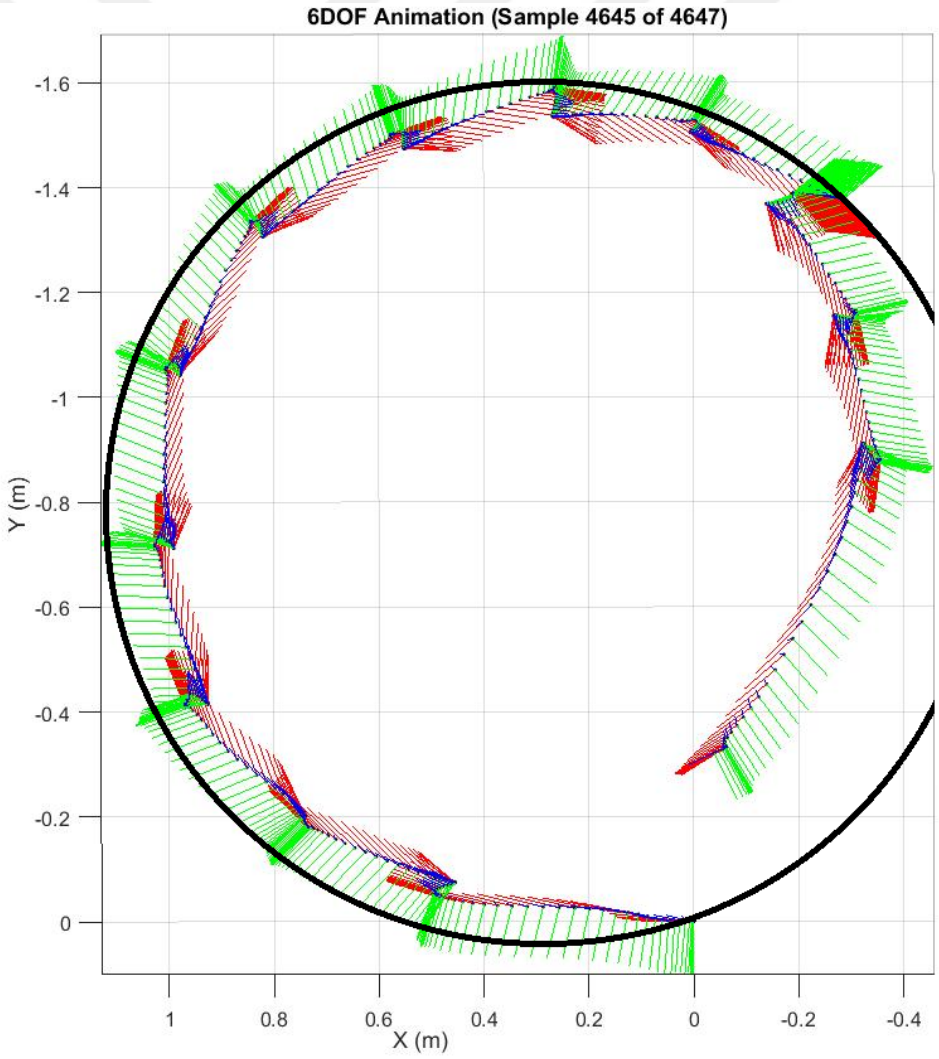
Şekil 4. 18 İkili adım gurubu ile SKE kullanılarak yapılan spiral merdiven testi

Şekil 4.19'da görüldüğü üzere ikili guruplarla SKE uygulandığında dairesel formda süreksizlikler giderilmiştir.



Şekil 4. 19 İkili adım gurubu ile spiral merdiven testi X-Y düzleminde görünüş

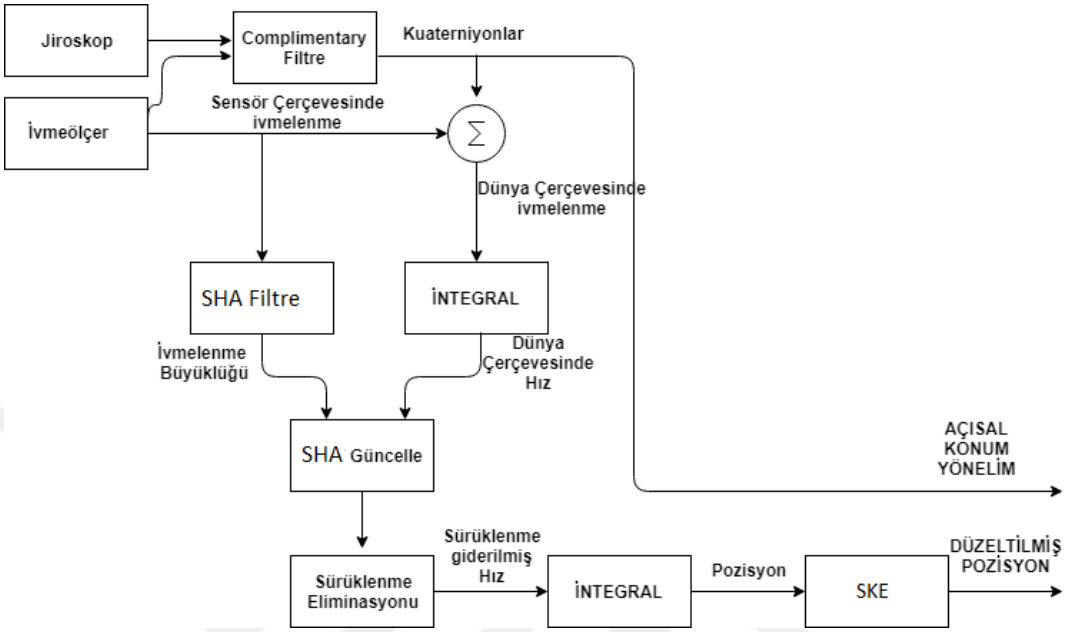
Sistemin dairesel harekette nasıl bir tahmin ortaya koyacağını test etmek amacıyla spiral merdivenlerin çevresinde merdivenleri çıkmadan yürüyüş testi yapılmıştır bu testte Z ekseninde tırmanma olmamıştır ve yürüyen kişi başladığı noktaya geri gelmiştir. Sistem 2'li adım gurupları ile çalıştırılmıştır. Yarı çapı 0.8 m olan bir dairede yürünülen çevre 5.4 metre kadardır ve yürüyüş sonunda yol tahmininde y-yönünde 0.3 metre ve x-yönünde sıfıra yakın hata elde edilmiştir. Bu lineer yürüyüşte elde edilen tahminine göre daha fazla olmakla birlikte SKE'nin hata düzeltme payının azalmasının sonucunda ortaya çıkmıştır. Ayrıca dairesel bir yolda yürümek IMU sensörlerinde daha fazla hata oluşmasına sebep olmaktadır. Şekil 4.20'de siyah daire gerçek yol hattını göstermek amacıyla grafiğe eklenmiştir konum tahmininin dairesel formda olduğu ancak başlangıç ve bitiş noktalarının birleşmediği göze çarpmaktadır.



Şekil 4. 20 Dairesel yürüyüş testi

4.5 Pozisyon Tahmin Sistemine Genel Bakış

Bu bölümde, tüm sistemi gözden geçirilip genel blok diagramı çizilmiş ve sistemin MATLAB ve gömülü C++ kodları açıklanmıştır.



Şekil 4. 21 Sistemin genel blok diyagramı

Üst seviye uygulama aşağıdaki adımlardan oluşur:

Adım 1: İvmeölçer, jiroskop ve manyetometre verileri bütünleyici filtreden geçirilir ve filtre tüm veri serisinin kuaterniyonlarını çıkarır. Kuaterniyonlar veya bunların euler gibi başka bir rotasyon temsiline dönüşümü arzu ettiğimiz yönelim açılarını verir. Matlab kodunda kuaterniyonlar, önceki bölümlerde gördüğümüz bir 3D yörünge izleme animasyonu oluşturmak için rotasyon matrisine dönüştürülür.

Adım 2: İvmeölçer verilerini sensör çerçevesinden Dünya çerçevesine döndürmek için 1. adımda hesaplanan kuaterniyonlar kullanılır.

Adım 3: Dünya çerçevesinde hız elde etmek için ivmelenme verileri integrasyon işlemine alınır.

Adım 4: SHA filtresi için gerekli durağan periyotları tespit etmek için ivmeölçer 1. derece Butterworth low pass ve 1. sıra Butterworth high pass filtrelerden geçirilir.

Adım 5: Dünya çerçevesinde SHA filtresinin hızlarını elde etmek için 3. adımda hesaplanan hızı durağan periyotları bulunur ve bu periyotlarda hız sıfırlanır.

Adım 6: Entegrasyon nedeniyle biriken sürüklenme durağan olmayan periyotlardan giderilir

Adım 7: Dünya çerçevesinde pozisyon bulmak için 6. adımda hesaplanan doğrusal hızın integrali alınır

Adım 8: Konum verileri SKE'ye iletilir ve nihai sürüklenme eliminasyonu işlemi gerçekleştirilir.

Böylelikle adım 1'de rotasyon tahmini adım 8'de ise pozisyon tahmini elde edilmiş oldu.

BÖLÜM 5

SONUÇ VE ÖNERİLER

Bu tezde, insanların yürüme biçimlerini gözlemlemek için uygulanabilir düşük maliyetli bir MEMS IMU bazlı konum ve rotasyon izleme çözümünü üzerinde çalışılmıştır. Bunun ötesinde sistem, iç mekan haritalama vb. alanlarda kullanılma potansiyeline sahiptir. Bu hedefe ulaşmak için bu çalışmada, düşük maliyetli atalet algılayıcılarının ciddi kayma sorununu aşabilecek hafif bir sistem kurulmaya çalışılmıştır. Ayrıca, navigasyon ve izleme sistemleri için teoriden uygulamaya kadar bir dizi yöntemler sunulmuştur.

Çalışmada, ataletsel takip için birkaç sürüklenme azaltma tekniği uygulanmış kendi kendine yeten, ancak hafif ve düşük maliyetli bir sistem geliştirilmiştir. Aynı zamanda bu tezde kullanılan MEMS sensörlerin çalışma prensiplerinden sensör ölçüm modellerine dair içerikler bulunmakla beraber, EK A ve EK B'de tüm test verileri ve sistemde koşan algoritmalar sunulmuştur.

Sunulan sonuçlara göre, tezde önerilen SKE sürüklenme önleme yaklaşımı iç ortamdaki sürüklenmenin çoğunluğunu ortadan kaldırdığı görülmüştür.

CF + SHA + SKE sistemi yapılan deneylerde oldukça iyi çalışsa da, bazı sınırlamaları vardır.

İlk olarak, SKE'nin baskın yönleri hesaplayabilmesi için yaya baskın yön boyunca düz bir koridordan başlamalıdır.

İkincisi, SKE'nin karmaşık bir binada veya zikzak paterni gibi karmaşık yürüyüş paternlerinde başarısız olması muhtemeldir. SKE, baskın yönü uzun süre bulamazsa, bu süre boyunca sürüklenmeye doğru yörüngeden tamamen sapabilir.

Mevcut sıfır algılama yöntemleri [22,25] normal yürüyüş döngüleri için genellikle iyi çalışır, ancak iç ortam koşullarında merdiven, basamak kıvrılan koridorlar gibi sensörleri etkileyecek etmenler olduğunda pozisyon tahmini iyi değildir. Bu sebeple sıfır hız algılama algoritmasına ek olarak ilave kalman tabanlı [23,24], veya sezgisel [20] düzeltici filtreler kullanılmaktadır.

Önerdiğimiz algoritma yapılan deneylerde [13] ve [25]'e göre daha iyi çalışsa da [20]'deki sistemin gerisinde kalmaktadır ayrıca algoritmanın bazı sınırlamaları vardır. İlk olarak, algoritmanın baskın yönleri hesaplayabilmesi için yaya baskın yön boyunca düz bir koridordan başlamalıdır. İkincisi, algoritmanın karmaşık bir binada, dairesel veya zikzaklı karmaşık yürüyüş yollarında istenilen başarıyı elde edememektedir. Sezgisel kayma eliminasyonu, baskın yönü uzun süre bulamazsa, bu süre boyunca sürüklenmeye doğru yörüngeden tamamen sapabilir.

Bu çalışma ile birlikte yürümenin doğasından kaynaklanan etkiler kullanılarak bir iç ortam navigasyon sisteminde IMU sensörlerinden kaynaklanan kaymanın eliminasyon yöntemi üzerinde çalışılmıştır. Performansı düşüren ölçüm yöntemi, sensör kalitesi, algoritma performansı gibi konular ileri bir çalışmada araştırılacaktır.

KAYNAKLAR

[1] Pasku V., Angelis A.D., Moschitta A., Carbone P., Nilsson J.O., (2016)"A magnetic ranging aided dead-reckoning indoor positioning system for pedestrian applications;" Proceedings of the IEEE Instrumentation and Measurement Technology Conference; 23–26 May 2016, Taipei, Taiwan.

[2] Xu, L., Xu, Q., (2017) "A Reliable Fusion Positioning Strategy for Land Vehicles in GPS-Denied Environments Based on Low-Cost Sensors" IEEE Transactions on Industrial Electronics, 64: 3205–3215.

[3] Wang, X., Wang, X, D. Feng, J. Fu, Z. Kang., (2016) "Research on matching and localization of characteristic unanimous infrared dim and small targets". J. Electron. Meas. Instrum. 30: 1405-1410.

[4] Sinelnikov Y., Sutin A., Salloum H., Sedunov N., Sedunov A., (2015)"Mice ultrasonic detection and localization in laboratory environment." J. Acoust. Soc. Am. 138:1791.

[5] Faragher, R., Harle, R., (2015) "Location fingerprinting with bluetooth low energy beacons". IEEE Journal on Selected Areas in Communications, 33: 2418 – 2428.

[6] Chen, X., Zou, S., (2017) "Improved Wi-Fi indoor positioning based on particle swarm optimization". IEEE Sensors Journal, 17: 7143 – 7148.

[7] Alvarez, Y., Las Heras, F., (2016) "ZigBee-based Sensor Network for Indoor Location and Tracking Applications". IEEE Latin America Transactions, 14: 3208 – 3214.

[8] Yang, P., Wu, W., (2014) "Efficient particle filter localization algorithm in dense passive RFID tag environment". IEEE Transactions on Industrial Electronics, 61: 5641 – 5651.

[9] Muqaibel, A., Alahmari, A., Kousa, M., Mesbah, A., Landlosi, A., (2014) "NOLS mitigation for UWB positioning". International Journal of Remote Sensing, 35: 7959-7977.

[10] Liu, P., Yang, P., Wang, C., Huang, K., Tieniu, T., (2017) “A semi-supervised method for surveillance-based visual location recognition”. IEEE Transactions on Cybernetics, 47: 3719 – 3732.

[11] Lasla, N., Younis, MF., Ouadjaout, A., Nadjib, B., (2015) “ An effective area-based localization algorithm for wireless networks”. IEEE Transactions on Computers, 64: 2103 – 2118.

[12] Wang, G., So, A.M.C., Li, Y., (2016) “ Robust convex approximation methods for TDOA-based localization under NLOS conditions”. IEEE Transactions on Signal Processing, 64:3281 – 3296.

[13] Ren, M., Pan, K., Liu, Y., Guo, H., Zhang, X., Wang, P., (2016) “ A novel pedestrian navigation algorithm for a foot-mounted inertial-sensor-based system”. Sensors, 16: 139.

[14] Ma, L., Chen, W., Li, B., You, Z., Chen, Z., (2014) “ Fast field calibration of MIMU based on the powell algorithm”. Sensors, 14: 16062-16081.

[15] Leccadito, M., (2013) A Kalman Filter Based Attitude Heading Reference System Using a Low Cost Inertial Measurement, Yüksek Lisans Tezi, Virginia Commonwealth University,Institute of Science, Richmond, Virginia.

[16] Lai, Y.C. , Jan, S.S., (2010) “Development of a Low-Cost Attitude and Heading Reference System Using a Three-Axis Rotating Platform,” Sensors, 10:2472-2491.

[17] Mahony, R., Hamel, T., (2005) “Complementary filter design on the special orthogonal group SO(3)” Proceedings of the 44th IEEE Conference on Decision and Control, 15 Dec. 2005, Seville, Spain.

[18] Farrel, J. A., (2008) AIDED NAVIGATION GPS with High Rate Sensors, Mc Graw-Hill Book Co, New York.

[19] Madgwick, S.O.H., Harrison, A.J.L., Vaidyanathan, R., (2011) “Estimation of IMU and MARG orientation using a gradient descent algorithm” IEEE International Conference on Rehabilitation Robotics, 29 June-1 July 2011, Zurich.

[20] Borenstein, J., Ojed, L., (2010) “Heuristic Drift Elimination for Personnel Tracking Systems” THE JOURNAL OF NAVIGATION, 63:591–606.

[21] EuclideanSpace,“Math-Quaternions,”. http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/ 28 Nisan 2019

[22] Park, S.K., Suh, Y.S., (2010) “A Zero Velocity Detection Algorithm Using Inertial Sensors for Pedestrian Navigation Systems” Sensors, 10:9163-9178.

[23] Foxlin, E., (2005). "Pedestrian Tracking with Shoe-Mounted Inertial Sensors". IEEE Computer Graphics and Applications, 25: 38-46.

[24] Fan, Q., Zhang, H., Sun, Y., Zhu, Y., Zhuang, X., Jia, J., (2018) “An Optimal Enhanced Kalman Filter for a ZUPT-Aided Pedestrian Positioning Coupling Model”. Sensors, 18(5): 1404.

[25] Wang, Y., Li, X., Zou, J., (2018) “A Foot-Mounted Inertial Measurement Unit (IMU) Positioning Algorithm Based on Magnetic Constraint”. Sensors, 18(3): 741.

[26] Cirillo, A., Cirillo, P., De Maria, G., Natale, C., Pirozzi, S., (2014) “Experimental Comparison of Sensor Fusion Algorithms for Attitude Estimation” Proceedings of the 19th World CongressThe International Federation of Automatic ControlCape Town, South Africa. August 24-29, 2014, 47(3): 7585-7591.

EK-A

AHRS GÖMÜLÜ C++ KODU

Genel Bilgiler

Bu kod ATMEGA 32U4 işlemcisinde çalışmaktadır sensör verilerini I2C yöntemi ile sensörden almakta verileri bütünledikten sonra bilgisayara USB aracılığıyla hesaplanan kuaternionları göndermektedir

```
#include "Math.h"

static float GyroMeasError = PI * (40.0f / 180.0f);
// gyroscope measurement drift in rad/s/s (start at 0.0 deg/s/s)
static float GyroMeasDrift = PI * (0.0f  / 180.0f);
// fusion scheme.
static float beta = sqrt(3.0f / 4.0f) * GyroMeasError;   // Compute
beta
// Compute zeta, the other free parameter in the Madgwick scheme
usually
// set to a small or zero value
static float zeta = sqrt(3.0f / 4.0f) * GyroMeasDrift;



// Vector to hold integral error for Mahony method
static float eInt[3] = {0.0f, 0.0f, 0.0f};
// Vector to hold quaternion
static float q[4] = {1.0f, 0.0f, 0.0f, 0.0f};

void MadgwickQuaternionUpdate(float ax, float ay, float az, float gx,
float gy, float gz, float mx, float my, float mz, float deltat)
```

```
{
  // short name local variable for readability
  float q1 = q[0], q2 = q[1], q3 = q[2], q4 = q[3];
  float norm;
  float hx, hy, _2bx, _2bz;
  float s1, s2, s3, s4;
  float qDot1, qDot2, qDot3, qDot4;

  // Auxiliary variables to avoid repeated arithmetic
  float _2q1mx;
  float _2q1my;
  float _2q1mz;
  float _2q2mx;
  float _4bx;
  float _4bz;
  float _2q1 = 2.0f * q1;
  float _2q2 = 2.0f * q2;
  float _2q3 = 2.0f * q3;
  float _2q4 = 2.0f * q4;
  float _2q1q3 = 2.0f * q1 * q3;
  float _2q3q4 = 2.0f * q3 * q4;
  float q1q1 = q1 * q1;
  float q1q2 = q1 * q2;
  float q1q3 = q1 * q3;
  float q1q4 = q1 * q4;
  float q2q2 = q2 * q2;
  float q2q3 = q2 * q3;
  float q2q4 = q2 * q4;
  float q3q3 = q3 * q3;
  float q3q4 = q3 * q4;
  float q4q4 = q4 * q4;

  // Normalise accelerometer measurement
  norm = sqrt(ax * ax + ay * ay + az * az);
  if (norm == 0.0f) return; // handle NaN
  norm = 1.0f/norm;
  ax *= norm;
  ay *= norm;
  az *= norm;
```

```
  // Normalise magnetometer measurement
  norm = sqrt(mx * mx + my * my + mz * mz);
  if (norm == 0.0f) return; // handle NaN
  norm = 1.0f/norm;
  mx *= norm;
  my *= norm;
  mz *= norm;

  // Reference direction of Earth's magnetic field
  _2q1mx = 2.0f * q1 * mx;
  _2q1my = 2.0f * q1 * my;
  _2q1mz = 2.0f * q1 * mz;
  _2q2mx = 2.0f * q2 * mx;
  hx = mx * q1q1 - _2q1my * q4 + _2q1mz * q3 + mx * q2q2 + _2q2 * my *
q3 +
       _2q2 * mz * q4 - mx * q3q3 - mx * q4q4;
  hy = _2q1mx * q4 + my * q1q1 - _2q1mz * q2 + _2q2mx * q3 - my * q2q2
+ my * q3q3 + _2q3 * mz * q4 - my * q4q4;
  _2bx = sqrt(hx * hx + hy * hy);
  _2bz = -_2q1mx * q3 + _2q1my * q2 + mz * q1q1 + _2q2mx * q4 - mz *
q2q2 + _2q3 * my * q4 - mz * q3q3 + mz * q4q4;
  _4bx = 2.0f * _2bx;
  _4bz = 2.0f * _2bz;

  // Gradient decent algorithm corrective step
  s1 = -_2q3 * (2.0f * q2q4 - _2q1q3 - ax) + _2q2 * (2.0f * q1q2 +
_2q3q4 - ay) - _2bz * q3 * (_2bx * (0.5f - q3q3 - q4q4) + _2bz * (q2q4
- q1q3) - mx) + (-_2bx * q4 + _2bz * q2) * (_2bx * (q2q3 - q1q4) +
_2bz * (q1q2 + q3q4) - my) + _2bx * q3 * (_2bx * (q1q3 + q2q4) + _2bz
* (0.5f - q2q2 - q3q3) - mz);
  s2 = _2q4 * (2.0f * q2q4 - _2q1q3 - ax) + _2q1 * (2.0f * q1q2 +
_2q3q4 - ay) - 4.0f * q2 * (1.0f - 2.0f * q2q2 - 2.0f * q3q3 - az) +
_2bz * q4 * (_2bx * (0.5f - q3q3 - q4q4) + _2bz * (q2q4 - q1q3) - mx)
+ (_2bx * q3 + _2bz * q1) * (_2bx * (q2q3 - q1q4) + _2bz * (q1q2 +
q3q4) - my) + (_2bx * q4 - _4bz * q2) * (_2bx * (q1q3 + q2q4) + _2bz *
(0.5f - q2q2 - q3q3) - mz);
  s3 = -_2q1 * (2.0f * q2q4 - _2q1q3 - ax) + _2q4 * (2.0f * q1q2 +
_2q3q4 - ay) - 4.0f * q3 * (1.0f - 2.0f * q2q2 - 2.0f * q3q3 - az) +
(-_4bx * q3 - _2bz * q1) * (_2bx * (0.5f - q3q3 - q4q4) + _2bz * (q2q4
- q1q3) - mx) + (_2bx * q2 + _2bz * q4) * (_2bx * (q2q3 - q1q4) + _2bz
* (q1q2 + q3q4) - my) + (_2bx * q1 - _4bz * q3) * (_2bx * (q1q3 +
q2q4) + _2bz * (0.5f - q2q2 - q3q3) - mz);
  s4 = _2q2 * (2.0f * q2q4 - _2q1q3 - ax) + _2q3 * (2.0f * q1q2 +
_2q3q4 - ay) + (-_4bx * q4 + _2bz * q2) * (_2bx * (0.5f - q3q3 - q4q4)
+ _2bz * (q2q4 - q1q3) - mx) + (-_2bx * q1 + _2bz * q3) * (_2bx *
(q2q3 - q1q4) + _2bz * (q1q2 + q3q4) - my) + _2bx * q2 * (_2bx * (q1q3
+ q2q4) + _2bz * (0.5f - q2q2 - q3q3) - mz);
```

```
  norm = sqrt(s1 * s1 + s2 * s2 + s3 * s3 + s4 * s4);    // normalise
step magnitude
  norm = 1.0f/norm;
  s1 *= norm;
  s2 *= norm;
  s3 *= norm;
  s4 *= norm;

  // Compute rate of change of quaternion
  qDot1 = 0.5f * (-q2 * gx - q3 * gy - q4 * gz) - beta * s1;
  qDot2 = 0.5f * (q1 * gx + q3 * gz - q4 * gy) - beta * s2;
  qDot3 = 0.5f * (q1 * gy - q2 * gz + q4 * gx) - beta * s3;
  qDot4 = 0.5f * (q1 * gz + q2 * gy - q3 * gx) - beta * s4;

  // Integrate to yield quaternion
  q1 += qDot1 * deltat;
  q2 += qDot2 * deltat;
  q3 += qDot3 * deltat;
  q4 += qDot4 * deltat;
  norm = sqrt(q1 * q1 + q2 * q2 + q3 * q3 + q4 * q4);    // normalise
quaternion
  norm = 1.0f/norm;
  q[0] = q1 * norm;
  q[1] = q2 * norm;
  q[2] = q3 * norm;
  q[3] = q4 * norm;
}

  // Apply feedback terms
  gx = gx + Kp * ex + Ki * eInt[0];
  gy = gy + Kp * ey + Ki * eInt[1];
  gz = gz + Kp * ez + Ki * eInt[2];

  // Integrate rate of change of quaternion
  pa = q2;
  pb = q3;
  pc = q4;
  q1 = q1 + (-q2 * gx - q3 * gy - q4 * gz) * (0.5f * deltat);
  q2 = pa + (q1 * gx + pb * gz - pc * gy) * (0.5f * deltat);
  q3 = pb + (q1 * gy - pa * gz + pc * gx) * (0.5f * deltat);
```

```
  q4 = pc + (q1 * gz + pa * gy - pb * gx) * (0.5f * deltat);

  // Normalise quaternion
  norm = sqrt(q1 * q1 + q2 * q2 + q3 * q3 + q4 * q4);
  norm = 1.0f / norm;
  q[0] = q1 * norm;
  q[1] = q2 * norm;
  q[2] = q3 * norm;
  q[3] = q4 * norm;
}

const float * getQ () { return q; }
```

EK-B

BÜTÜNLEYİCİ FİLTRE + SKE + SHA MATLAB KODU

```
clear;
close all;
clc;
addpath('Quaternions');
addpath('ximu_matlab_library');

% ------------------------------------------------------------------------
-----
% Select dataset (comment in/out)

%filePath = 'Datasets/straightLine';
%startTime = 6;
%stopTime = 26;

 filePath = 'Datasets/stairsAndCorridor';
 startTime = 5;
 stopTime = 53;

% filePath = 'Datasets/spiralStairs';
% startTime = 4;
% stopTime = 47;

% ------------------------------------------------------------------------
-----
% Import data

samplePeriod = 1/256;
xIMUdata = xIMUdataClass(filePath, 'InertialMagneticSampleRate',
1/samplePeriod);
time = xIMUdata.CalInertialAndMagneticData.Time;
gyrX = xIMUdata.CalInertialAndMagneticData.Gyroscope.X;
gyrY = xIMUdata.CalInertialAndMagneticData.Gyroscope.Y;
gyrZ = xIMUdata.CalInertialAndMagneticData.Gyroscope.Z;
accX = xIMUdata.CalInertialAndMagneticData.Accelerometer.X;
accY = xIMUdata.CalInertialAndMagneticData.Accelerometer.Y;
accZ = xIMUdata.CalInertialAndMagneticData.Accelerometer.Z;
magX = xIMUdata.CalInertialAndMagneticData.Magnetometer.X;
```

```
magY = xIMUdata.CalInertialAndMagneticData.Magnetometer.Y;
magZ = xIMUdata.CalInertialAndMagneticData.Magnetometer.Z;
clear('xIMUdata');

% ------------------------------------------------------------------------
-----
% Manually frame data

% startTime = 0;
% stopTime = 10;

indexSel = find(sign(time-startTime)+1, 1) : find(sign(time-
stopTime)+1, 1);
time = time(indexSel);
gyrX = gyrX(indexSel, :);
gyrY = gyrY(indexSel, :);
gyrZ = gyrZ(indexSel, :);
accX = accX(indexSel, :);
accY = accY(indexSel, :);
accZ = accZ(indexSel, :);
magX = magX(indexSel, :);
magY = magY(indexSel, :);
magZ = magZ(indexSel, :);
% --------------------------

% ------------------------------------------------------------------------
-----
% Detect stationary periods

% Compute accelerometer magnitude
acc_mag = sqrt(accX.*accX + accY.*accY + accZ.*accZ);

% HP filter accelerometer data
filtCutOff = 0.001;
[b, a] = butter(1, (2*filtCutOff)/(1/samplePeriod), 'high');
acc_magFilt = filtfilt(b, a, acc_mag);

% Compute absolute value
acc_magFilt = abs(acc_magFilt);

% LP filter accelerometer data
filtCutOff = 5;
[b, a] = butter(1, (2*filtCutOff)/(1/samplePeriod), 'low');
acc_magFilt = filtfilt(b, a, acc_magFilt);

% Threshold detection
stationary = acc_magFilt < 0.05;

% ------------------------------------------------------------------------
-----
% Plot data raw sensor data and stationary periods

figure('Position', [9 39 900 600], 'NumberTitle', 'off', 'Name',
'Sensor Data');
ax(1) = subplot(2,1,1);
    hold on;
    plot(time, gyrX, 'r');
    plot(time, gyrY, 'g');
    plot(time, gyrZ, 'b');
    title('Gyroscope');
```

```
    xlabel('Time (s)');
    ylabel('Angular velocity (^\circ/s)');
    legend('X', 'Y', 'Z');
    hold off;
ax(2) = subplot(2,1,2);
    hold on;
    plot(time, accX, 'r');
    plot(time, accY, 'g');
    plot(time, accZ, 'b');
    plot(time, acc_magFilt, ':k');
    plot(time, stationary, 'k', 'LineWidth', 2);
    title('Accelerometer');
    xlabel('Time (s)');
    ylabel('Acceleration (g)');
    legend('X', 'Y', 'Z', 'Filtered', 'Stationary');
    hold off;
linkaxes(ax,'x');

% ------------------------------------------------------------------------
-----
% Compute orientation

quat = zeros(length(time), 4);
AHRSalgorithm = AHRS('SamplePeriod', 1/256, 'Kp', 1, 'KpInit', 1);

% Initial convergence
initPeriod = 2;
indexSel = 1 : find(sign(time-(time(1)+initPeriod))+1, 1);
for i = 1:2000
    AHRSalgorithm.UpdateIMU([0 0 0], [mean(accX(indexSel))
mean(accY(indexSel)) mean(accZ(indexSel))]);
end

% For all data
for t = 1:length(time)
    if(stationary(t))
        AHRSalgorithm.Kp = 0.5;
    else
        AHRSalgorithm.Kp = 0;
    end
    AHRSalgorithm.UpdateIMU(deg2rad([gyrX(t) gyrY(t) gyrZ(t)]),
[accX(t) accY(t) accZ(t)]);
    quat(t,:) = AHRSalgorithm.Quaternion;
end

% ------------------------------------------------------------------------
-----
% Compute translational accelerations

% Rotate body accelerations to Earth frame
acc = quaternRotate([accX accY accZ], quaternConj(quat));

% % Remove gravity from measurements
% acc = acc - [zeros(length(time), 2) ones(length(time), 1)];     %
unnecessary due to velocity integral drift compensation

% Convert acceleration measurements to m/s/s
acc = acc * 9.81;

% Plot translational accelerations
```

```
figure('Position', [9 39 900 300], 'NumberTitle', 'off', 'Name',
'Accelerations');
hold on;
plot(time, acc(:,1), 'r');
plot(time, acc(:,2), 'g');
plot(time, acc(:,3), 'b');
title('Acceleration');
xlabel('Time (s)');
ylabel('Acceleration (m/s/s)');
legend('X', 'Y', 'Z');
hold off;

% ------------------------------------------------------------------
-----
% Compute translational velocities

acc(:,3) = acc(:,3) - 9.81;

% Integrate acceleration to yield velocity
vel = zeros(size(acc));
for t = 2:length(vel)
    vel(t,:) = vel(t-1,:) + acc(t,:) * samplePeriod;
    if(stationary(t) == 1)
        vel(t,:) = [0 0 0];     % force zero velocity when foot
stationary
    end
end


% Compute integral drift during non-stationary periods
velDrift = zeros(size(vel));
stationaryStart = find([0; diff(stationary)] == 1);
stationaryEnd = find([0; diff(stationary)] == -1);
for i = 1:numel(stationaryEnd)
    driftRate = vel(stationaryEnd(i)-1, :) / (stationaryEnd(i) -
stationaryStart(i));
    enum = 1:(stationaryEnd(i) - stationaryStart(i));
    drift = [enum'*driftRate(1) enum'*driftRate(2)
enum'*driftRate(3)];
    velDrift(stationaryStart(i):stationaryEnd(i)-1, :) = drift;
end

% Remove integral drift
vel = vel - velDrift;

% Plot translational velocity
figure('Position', [9 39 900 300], 'NumberTitle', 'off', 'Name',
'Velocity');
hold on;
plot(time, vel(:,1), 'r');
plot(time, vel(:,2), 'g');
plot(time, vel(:,3), 'b');
title('Velocity');
xlabel('Time (s)');
ylabel('Velocity (m/s)');
legend('X', 'Y', 'Z');
hold off;

% ------------------------------------------------------------------
-----
```

```
% Compute translational position

% Integrate velocity to yield position
pos = zeros(size(vel));
for t = 2:length(pos)
pos(t,:) = pos(t-1,:) + vel(t,:) * samplePeriod; % integrate velocity
to yield position
end


%--------------xxxxxxxxxxxxxxxxx

% get position of the first 5 steps
first5X = pos(stationaryEnd(1:5),1);
first5Y = pos(stationaryEnd(1:5),2);
% use polyfit to calculate best fit straight line for the first five
step and serve as X-Y dominant direction
p = polyfit(first5X,first5Y,1);



% Heuristic Drift Elimination
step = zeros(length(stationaryEnd),3);
headingDifferenceXP = zeros(length(step),1); % heading difference
between current step and dominant XP direction
headingDifferenceYP = zeros(length(step),1); % heading difference
between current step and dominant YP direction
headingDifferenceXN = zeros(length(step),1); % heading difference
between current step and dominant XN direction
headingDifferenceYN = zeros(length(step),1); % heading difference
between current step and dominant YN direction

HDEquat = zeros(length(step),4);


% start EHDE correction
for i = 1:length(stationaryEnd)

    % step calculated from position
step(i,:) = pos(stationaryStart(i),:) - pos(stationaryEnd(i),:);

% walking on flat floor
if abs(step(i,3)) < 0.2
fprintf('into plane correction.\n');

% dominant direction in 3D determined
domDirXP = [1,p(1),0];
domDirYN = [1,-1/p(1),0];
domDirXN = [-1,-p(1),0];
domDirYP = [-1,1/p(1),0];

% heading difference between current step and each of four dominant
directions
headingDifferenceXP(i) = getAngle(step(i,:),domDirXP);
headingDifferenceYP(i) = getAngle(step(i,:),domDirYP);
headingDifferenceXN(i) = getAngle(step(i,:),domDirXN);
headingDifferenceYN(i) = getAngle(step(i,:),domDirYN);

% determine which possible zone does current step fall into
if abs(headingDifferenceXP(i)) < 0.2
```

```
% calculate quaternion from current step to determined dominant
direction
HDEquat(i,:) = calcQuat(step(i,:), domDirXP);

% rotate current step tp nearest dominant direction
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
fprintf('plane correction to XP made %dth iteration.\n',i);

% if not the first step, parallel move the remaining whole trajectory
to the end of previous step
if i >= 2
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );
end

elseif abs(headingDifferenceYP(i)) < 0.2
HDEquat(i,:) = calcQuat(step(i,:), domDirYP);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
fprintf('plane correction to YP made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );

elseif abs(headingDifferenceXN(i)) < 0.2
HDEquat(i,:) = calcQuat(step(i,:), domDirXN);
fprintf('plane correction to XN made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );

elseif abs(headingDifferenceYN(i)) < 0.2
HDEquat(i,:) = calcQuat(step(i,:), domDirYN);
fprintf('plane correction to YN made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
```

```
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );
end

% if walking along stairs
elseif abs(step(i,3)) > 0.2
fprintf('into stair correction.\n');

% dominant direction in 3D determined, gradient remained
domDirXP = [1,p(1),step(i,3) * norm([1,p(1)]) / norm(step(i,1:2))];
domDirYN = [1,-1/p(1),step(i,3) * norm([1,-1/p(1)]) /
norm(step(i,1:2))];
domDirXN = [-1,-p(1),step(i,3) * norm([-1,p(1)]) / norm(step(i,1:2))];
domDirYP = [-1,1/p(1),step(i,3) * norm([-1,1/p(1)]) /
norm(step(i,1:2))];

% heading difference between current step and each of four dominant
directions
headingDifferenceXP(i) = getAngle(step(i,:),domDirXP);
headingDifferenceYP(i) = getAngle(step(i,:),domDirYP);
headingDifferenceXN(i) = getAngle(step(i,:),domDirXN);
headingDifferenceYN(i) = getAngle(step(i,:),domDirYN);

% determine which possible zone does current step fall into and
correct current step to nearest dominant direction

if abs(headingDifferenceXP(i)) < 0.4
HDEquat(i,:) = calcQuat(step(i,:), domDirXP);
fprintf('stair correction to XP made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );

elseif abs(headingDifferenceYP(i)) < 0.4
HDEquat(i,:) = calcQuat(step(i,:), domDirYP);
fprintf('stair correction to YP made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
```

```
pos(stationaryStart(i-1),3) );

elseif abs(headingDifferenceXN(i)) < 0.4
HDEquat(i,:) = calcQuat(step(i,:), domDirXN);
fprintf('stair correction to XN made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );

elseif abs(headingDifferenceYN(i)) < 0.4
HDEquat(i,:) = calcQuat(step(i,:), domDirYN);
fprintf('stair correction to YN made %dth iteration.\n',i);
pos(stationaryEnd(i):length(pos),:) =
quaternRotate(pos(stationaryEnd(i):length(pos),:),HDEquat(i,:));
pos(stationaryEnd(i):length(pos),1) =
pos(stationaryEnd(i):length(pos),1) - ( pos(stationaryEnd(i),1) -
pos(stationaryStart(i-1),1) );
pos(stationaryEnd(i):length(pos),2) =
pos(stationaryEnd(i):length(pos),2) - ( pos(stationaryEnd(i),2) -
pos(stationaryStart(i-1),2) );
pos(stationaryEnd(i):length(pos),3) =
pos(stationaryEnd(i):length(pos),3) - ( pos(stationaryEnd(i),3) -
pos(stationaryStart(i-1),3) );
end
end

end
% % calculate heading difference between curret step and previous step
in right hand coordinate
% headingBetweenSteps = zeros(length(step)-1,1);
% for i = 1:length(headingBetweenSteps)
% headingBetweenSteps(i) = getAngle(step(i,:),step(i+1,:));
% end;

%--------------xxxxxxxxxxxxxxxxx

% Plot translational position
figure('Position', [9 39 900 600], 'NumberTitle', 'off', 'Name',
'Position');
hold on;
plot(time, pos(:,1), 'r');
plot(time, pos(:,2), 'g');
plot(time, pos(:,3), 'b');
title('Position');
xlabel('Time (s)');
ylabel('Position (m)');
legend('X', 'Y', 'Z');
hold off;

% ------------------------------------------------------------------
-----
% Plot 3D foot trajectory
```

```
% % Remove stationary periods from data to plot
% posPlot = pos(find(~stationary), :);
% quatPlot = quat(find(~stationary), :);
posPlot = pos;
quatPlot = quat;

% Extend final sample to delay end of animation
extraTime = 20;
onesVector = ones(extraTime*(1/samplePeriod), 1);
posPlot = [posPlot; [posPlot(end, 1)*onesVector, posPlot(end,
2)*onesVector, posPlot(end, 3)*onesVector]];
quatPlot = [quatPlot; [quatPlot(end, 1)*onesVector, quatPlot(end,
2)*onesVector, quatPlot(end, 3)*onesVector, quatPlot(end,
4)*onesVector]];

% Create 6 DOF animation
SamplePlotFreq = 4;
Spin = 120;
SixDofAnimation(posPlot, quatern2rotMat(quatPlot), ...
                'SamplePlotFreq', SamplePlotFreq, 'Trail', 'All', ...
                'Position', [9 39 1280 768], 'View',
[(100:(Spin/(length(posPlot)-1)):(100+Spin))',
10*ones(length(posPlot), 1)], ...
                'AxisLength', 0.1, 'ShowArrowHead', false, ...
                'Xlabel', 'X (m)', 'Ylabel', 'Y (m)', 'Zlabel', 'Z
(m)', 'ShowLegend', false, ...
                'CreateAVI', false, 'AVIfileNameEnum', false,
'AVIfps', ((1/samplePeriod) / SamplePlotFreq));
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Oben ÖZKAN
Doğum Tarihi ve Yeri	:26.08.1991 , Kayseri
Yabancı Dili	:İngilizce
E-posta	:oben.ozkan91@gmail.com

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Lisans	Mekatronik Mühendisliği	Erciyes Üniversitesi	2015

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2019	ASELSAN A.Ş	Mekatronik Test Sistemleri Geliştirme Mühendisi
2018	Anova Arge Teknolojileri A.Ş	Test Sistemleri Geliştirme Mühendisi
2017	Mir Araştırma ve Geliştirme A.Ş	Elektronik ve Yazılım Mühendisi

YAYINLARI

Makale

1. Özkan, O., Yeşildirek, A., (2019)“Sensör Füzyonu Algoritmaları ile Açısal Konum Referans Sistemi Tasarımı”. Bayburt Üniversitesi Fen Bilimleri Dergisi, 2(1):93-103