

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**HAREKET ALGILAYAN DUYARGALAR İLE ÜÇ BOYUTLU UZAYDA
HAREKET KESTİRİMİ**

**YÜKSEK LİSANS TEZİ
Cem KÜÇÇÜK**

Anabilim Dalı : Bilgisayar Bilimleri

Programı : Bilgisayar Bilimleri

Tez Danışmanı: Yrd. Doç. Dr. D. Turgay Altılar

ARALIK 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

**HAREKET ALGILAYAN DUYARGALAR İLE ÜÇ BOYUTLU UZAYDA
HAREKET KESTİRİMİ**

**YÜKSEK LİSANS TEZİ
Cem KÜÇÇÜK
704071006**

Tezin Enstitüye Verildiği Tarih : 20 Aralık 2010

Tezin Savunulduğu Tarih : 20 Aralık 2010

**Tez Danışmanı : Yrd. Doç. Dr. D. Turgay ALTILAR (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Bülent ÖRENCİK (TÜBİTAK)
Yrd. Doç. Dr. M. Ersel KAMAŞAK (İTÜ)**

ARALIK 2010

Aileme ve arkadaşlarıma,

ÖNSÖZ

Tez çalışması süresince yardımlarını eksik etmeyen arkadaşlarım Alper Bodur, Ersin Özkan ve tez danışmanım D. Turgay Altılar'a teşekkürlerimi sunarım.

Aralık 2010

Cem Küçük
Bilgisayar Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ŞEKİL LİSTESİ.....	xi
ÖZET.....	xiii
SUMMARY	xv
1. GİRİŞ	1
1.1 Tezin Amacı	2
1.2 Ataletsel Ölçüm Ünitesi Kullanılarak Gerçekleştirilen Çalışmalar	2
2. HAREKET DUYARGALARI VE KULLANIMI.....	5
2.1 Ataletsel Ölçüm Ünitesi	5
2.1.1 İvmeölçer Duyargası	7
2.1.2 Açısal Hız Duyargası	8
2.1.3 Dengeli ve Sabit Ataletsel Ölçüm Üniteleri.....	8
2.2 Nintendo Wii Donanımı ve Özellikleri	9
3. HAREKET DUYARGALARI İLE KESTİRİM.....	13
3.1 Yönelim ve Doğrusal Hareket Hesaplamaları.....	14
3.1.1 Yönelim Hesaplamaları.....	14
3.1.2 Yönelim İfade Teknikleri.....	18
3.1.3 Doğrusal Hareket Hesaplamaları	22
3.2 Ataletsel Ölçüm Ünitelerinde Karşılaşılan Problemler	23
3.3 Kestirim ve Takip İşlemleri.....	24
3.3.1 Kalman Filtresi.....	25
3.3.2 Genişletilmiş Kalman Filtresi	27
4. ÜÇ BOYUTLU UZAYDA HAREKET KESTİRİMİ	29
4.1 Kestirim Öncesi Yüksek Frekanstaki Gürültülerin Süzülmesi	30
4.2 Yönelim Kestirimi.....	32
4.3 Doğrusal İvme, Hız ve Konum Kestirimi	37
4.4 Hareket Sonlandırma.....	40
4.5 Algoritma Sonuçları ve Karşılaştırmalar.....	41
5. YAZILIM GELİŞTİRME	63
5.1 Uygulama ve Kütüphane Yönetimi	64
5.2 Hareket Kestirim Kütüphanesi	65
5.3 Üç Boyutlu Arayüz Yazılımı.....	71
6. SONUÇ VE ÖNERİLER.....	73
KAYNAKLAR	77
EKLER.....	79

KISALTMALAR

AHRS	: İstikamet ve Durum Bilgi Sistemi (“Attitude and Heading Reference System”)
AÖÜ	: Ataletsel Ölçüm Ünitesi (“Inertial Measurement Unit”)
cm	: Santimetre
DCM	: Yönelimsel Kosinüs Matrisi (“Directional Cosine Matrix”)
DOF	: Serbestlik Derecesi (“Degrees Of Freedom”)
EEPROM	: Silinip Programlanabilir Salt Okunur Bellek (“Electrically Erasable Programmable Read-Only Memory”)
Hz	: Hertz
g	: Yerçekimi ivmesi
IRBM	: Orta Menzilli Balistik Füze (“Intermediate Range Ballistic Missile”)
I2C	: Karışıklık Tümlüşik Devre (“Inter-Integrated Circuit”)
kbit	: Kilo bit
m	: Metre
MEMS	: Mikro-Elektro-Mekanik Sistemler (“Micro-Electro-Mechanic Systems”)
mm	: Milimetre
m/s	: Metre / Saniye
m/s²	: Metre / Saniye Kare
rad/s	: Radyan / Saniye
SPI	: Seri Çevresel Arayüz Veri yolu (“Serial Peripheral Interface Bus”)
WinDDK	: Windows Aygıt Sürücüsü Gereci (“Windows Driver Device Kit”)
WDK	: Windows Sürücü Gereci (“Windows Driver Kit”)
°/s	: Derece / Saniye
3D	: Üç Boyutlu (“Three Dimensional”)

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : IRBM S3 Füzelerinin AÖÜ'sü.....	5
Şekil 2.2 : Cloud Cap “Crista” 6DOF AÖÜ	6
Şekil 2.3 : ADIS16350/ADIS16355 Blok Diyagramı	7
Şekil 2.4 : Dengeli AÖÜ.....	8
Şekil 2.5 : Sabit AÖÜ	9
Şekil 2.6 : Nintendo Wiimote	10
Şekil 2.7 : Nintendo Wii Motion Plus	11
Şekil 3.1 : Dikilme, Yuvarlanma ve Sapma Hareketleri	14
Şekil 3.2 : Yerçekimsiz Ortamda İvmeölçer.....	15
Şekil 3.3 : Yerçekimsiz Ortamda Hareket Eden İvmeölçer.....	15
Şekil 3.4 : Yerçekimi Etkisinde İvmeölçer.....	15
Şekil 3.5 : Yerçekimi Etkisinde Döndürülmüş İvmeölçer.....	16
Şekil 3.6 : Nesne Uzayı ve Referans Uzayı.....	18
Şekil 3.7 : Dördeyde Tek Eksene İndirgenmiş Dönüş.....	21
Şekil 3.8 : Durum Kestirimi Blok Diyagramı.....	25
Şekil 3.9 : Kalman Filtresi Algoritması.....	26
Şekil 4.1 : Sabit AÖÜ ile Hareket Kestirimi	29
Şekil 4.2 : Hareketsiz Durumda Ortalama Alan Filtre	31
Şekil 4.3 : İvme Verileri İçin Ortalama Alan Filtre.....	31
Şekil 4.4 : Açısal Hız İçin Ortalama Alan Filtre Performansı.....	32
Şekil 4.5 : Hareket Sonlandırmada Hata Noktası	40
Şekil 4.6 : Hareket I - Dikilme, Yuvarlanma ve Sapma Değişimi	42
Şekil 4.7 : Hareket I - Dikilme Açısı	43
Şekil 4.8 : Hareket I - Yuvarlanma Açısı	44
Şekil 4.9 : Hareket I - Sapma Açısı	45
Şekil 4.10 : Hareket I - Z Ekseni İvmeleri.....	46
Şekil 4.11 : Hareket II – Yuvarlanma ve Dikilme Hareketi	47
Şekil 4.12 : Hareket II - Yönelim Kestirimi	48
Şekil 4.13 : Hareket II – Hareket Durumu.....	49
Şekil 4.14 : Hareket III - Y Ekseninde Sağ ve Sol	49
Şekil 4.15 : Hareket III - Kestirim Sonuçları.....	51
Şekil 4.16 : Hareket IV - Yuvarlanarak Konum Değişimi	51
Şekil 4.17 : Hareket IV - Kestirim Sonuçları	53
Şekil 4.18 : Hareket V - Sapma Yapararak Konum Değişimi.....	53
Şekil 4.19 : Hareket V - Kestirim sonuçları	55
Şekil 4.20 : Hareket VI - Üç Eksen Hareket.....	56
Şekil 4.21 : Hareket VI - Dikilme Açısı	57
Şekil 4.22 : Hareket VI - Yuvarlanma Açısı.....	58
Şekil 4.23 : Hareket VI - Sapma Açısı	59
Şekil 4.24 : Hareket VI - Konum kestirimi (x).....	59
Şekil 4.25 : Hareket VI - Konum kestirimi (y).....	60
Şekil 4.26 : Hareket VI - Konum kestirimi (z).....	60

Şekil 4.27 : Hareket VI - Hareket Durumu İnceleme	61
Şekil 5.1 : Yazılım Mimarisi.....	65
Şekil 5.2 : Hareket Kestirim Kütüphanesi Sınıf Diyagramı	69
Şekil 5.3 : Ölçüleme Yapılandırma Dosyası İçeriği	70
Şekil 5.4 : 3D Arayüz Yazılımı	72

HAREKET ALGILAYAN DUYARGALAR İLE ÜÇ BOYUTLU UZAYDA HAREKET KESTİRİMİ

ÖZET

Son yıllarda bilgisayar bilimleri alanında sanal gerçeklik konusunda yapılan çalışmalarda büyük bir artış gözlemlenmektedir. Sanal gerçeklik kullanıcının, bilgisayar ortamında benzetimi yapılmış gerçek veya sanal bir ortam ile etkileşimde bulunmasına izin veren bir teknolojidir. Sanal gerçeklik ile çok yakından ilgili olan insan-makine ara yüzü tasarımı teknolojileri de kullanıcıların bilgisayar programları ile etkileşimini en üst düzeye çıkarma amacına sahiptir. Bu yaklaşım göz önünde bulundurularak tez kapsamında bir sanal gerçeklik uygulamasında kullanılmak üzere kullanıcının elindeki ve denetimindeki bir nesnenin üç boyutlu uzaydaki hareketinin tespit edilmesi hedeflendi.

Bu amaçla üç ekseninde doğrusal ivme ve açısal hız bilgilerine erişilebilen bir donanım kullanıldı. Üç eksen ivmeölçere (“accelerometer”) ve üç eksen açısal hız duyargasına (“gyroscope”) sahip olan bir ataletsel ölçüm ünitesi (AÖÜ) kullanılarak altı serbestlik derecesine sahip (6 DOF) bir sistem elde edilmektedir. Tez kapsamında elde edilen bu sistem için kalman filtresi kullanılarak üç boyutlu yönelim, hız ve konum kestirimi gerçekleştirildi.

AÖÜ’ler başlıca hava/kara/deniz araçlarında (insansızlar da dâhil) olmak üzere birçok farklı alanda kullanılmakta, bu alanlara göre farklı hassasiyetlerdeki ve teknik özelliklerdeki duyargalarla oluşturulmaktadır. İnsansız bir hava aracının ataletsel seyir sisteminde ele alınan hareket kestirimi problemi temelde bir oyunda kullanılan kumandanın hareket kestirimi ile aynı prensiplere sahiptir. Sanal gerçeklik uygulamalarında kullanılmak üzere üretilen ucuz maliyetli duyargaların kararlılığı ve doğruluğu, diğer alanlarda kullanılan duyargalara göre daha azdır. Bu gibi durumlarda kestirim algoritmalarının daha güçlü olması ve ham duyarga verisindeki hataları daha iyi şekilde süzmesi gerekmektedir. Üç eksen ivmeölçer ve açısal hız duyargasından oluşan bu dinamik sistemin durum denklemlerinin, durum geçişlerinin ve hata karakteristiklerinin çıkartılması, bu çıkarımlar kullanılarak bir kestirim ve takip algoritmasının çalıştırılması gerekmektedir. Kalman filtresi ataletsel seyir sistemleri dâhil birçok dinamik sistemin durum kestirimi ve takibi için yaygın olarak kullanılan etkin bir filtredir. Bu yüzden yapılan çalışmalarda kalman filtresi kullanıldı.

Yönelim hesaplamalarında kullanılan açı, koordinat dönüşüm matrisi ve dördey (“quaternion”) tabanlı yöntem sonuçları ile kalman filtresi ile düzeltilerek elde edilen sonuçlar karşılaştırıldı ve iyileştirmeler gözlemlendi. Konum kestirimine getirilen hareket sonlandırma yaklaşımı ile iyileştirmeler elde edildi. Yazılım mimarisi açısından modüler, geliştirmeye açık, sisteme uyarlaması kolay, platformdan bağımsız bir hareket kestirim kütüphanesi geliştirildi. Bu kütüphaneyi kullanan uygulamaya yönelik bir arayüz tasarlanarak gerçekleştirildi.

MOTION ESTIMATION IN THREE DIMENSIONAL SPACE WITH MOTION SENSING MODULES

SUMMARY

Virtual reality has become a very popular field in computer sciences and the research areas in this field have increased in last few years. Virtual reality is a term that applies to computer-simulated environments that can simulate places in the real world as well as in imaginary worlds. Human-computer interaction technology, which is closely related with the virtual reality, aims to improve interaction between users and computers. Considering to this approach, estimation and tracking the motion of an object, which is hold and controlled by a user in three-dimensional space is determined as the purpose of the project.

For this purpose, it was used a hardware which supplies three axis acceleration and three axis angular rate information of the object. A dynamic system with six degrees of freedom is obtained with three-axis acceleration and three dimensional orientation, velocity and position of this dynamic system is estimated and tracked by using a kalman filter design.

Inertial measurement units which are used mostly on aerial/land/surface/underwater vehicles including unmanned types, have become very popular in various areas like entertainment sector and have different sensitivity levels and specifications according to these sectors. The problem of estimation and tracking the motion of an unmanned aerial vehicle has essentially similar principles as the problem of estimation and tracking the motion of a gamepad in a virtual reality application. The small and low-cost sensors, which are produced, for gaming applications have less stability and accuracy, so the raw sensor outputs need to be filtered better and estimation algorithm must be robust. The state equations, state transitions and error characteristics of the dynamic system must be determined and estimation and tracking algorithm need to be run. Kalman filtering is an efficient and commonly used filter for estimation and tracking of different dynamic systems including mostly inertial navigation systems. Hence, kalman filtering is implemented for the estimation and tracking operations in the studies.

Angle, coordinate transformation matrix and quaternion based orientation representation techniques are implemented for orientation calculation. The results of these methods and Kalman filter based combined (gyro + accelerometer) orientation estimator are compared and expected improvement is observed. A motion ending technique is developed for position estimation and it is shown that developed technique improves results and provides better accuracy. A modular, reusable and platform independent motion estimation software framework is developed for system integrability. A user interface is designed and implemented specifically for this application by utilizing this framework.

1. GİRİŞ

Son yıllarda bilgisayar bilimleri alanında hareket algılayan duyargalar kullanılarak yapılan çalışmalarda büyük bir artış gerçekleşmektedir. Bir nesnenin üç boyutlu hareketinin tespit edilebilmesi endüstriyel uygulamalardan, eğlence sektörüne, eğitim ve benzetim amaçlı çalışmalardan sağlık sektörüne kadar birçok alanda yenilikçi çalışmaların yapılabilmesine olanak sağlamaktadır. Bu yaklaşım çerçevesinde hem üç boyutlu hareketi tespit etmede kullanılan donanımlarda, hem de bu donanımlar kullanılarak yapılan çalışmalarda karşılıklı hızlı bir geliştirme ve iyileştirme süreci devam etmektedir. Her geçen gün farklı uygulama alanlarına göre özelleşmiş yeni donanımlar geliştirilmekte, çalışmaların bu ürünleri yoğun bir şekilde kullanması da bu sektöre harcanan iş gücünün artmasını sağlamaktadır. Böylece daha gelişmiş ve kapsamlı donanımlar daha ucuz maliyetlerle üretilebilmekte ve kullanıcılar tarafından daha kolay temin edilebilmektedir. Boyutları ve maliyetleri kabul edilebilir oranlarda küçülen donanımlar ilk olarak insansız hava/deniz/kara araçlarında yaygın şekilde kullanılmaya başlandı. Bu donanımlar, çalışma yelpazesi genişleştikçe oyun sektöründe karakter hareketlerinin modellenmesi ve oyun kumandalarının daha gerçekçi hale getirilmesi amacının yanında tıptan benzetim projelerine kadar birçok farklı alandaki sanal gerçeklik uygulamasında başlıca kullanılan bir gereç haline geldi. Donanımların örüntü tanıma alanında kullanılması ile insan hareketlerin belirlenmesinden yaşamsal belirtilerin tespit edilmesine uzanan farklı kullanım alanları yaratılmış oldu. Donanımlarda bulunan duyargaların çeşitleri ve uygulama alanlarının farklılıkları verilerin işlenmesinde, yani belirlenen problemlerin çözümleri için oluşturulacak algoritmaların tasarlanmasında birbirinden farklı yaklaşımların ortaya çıkmasına sebep oldu. Farklı yaklaşımları bulunan özelleşmiş uygulamalar bir kenara bırakılırsa, AÖÜ kullanan birçok çalışmadaki ortak problemin temelinde üç boyutta yönelim ve konum tespiti bulunduğu görülmektedir.

1.1 Tezin Amacı

Yapılan çalışmada, daha çok sanal gerçeklik ve oyun alanında kullanılmak üzere tasarlanmış, küçük boyutta, ucuz maliyetli ve görece daha az hassas bir donanım kullanılarak üç boyutlu uzayda yönelim, hız ve konum kestirimine dayalı hareket takibi amaçlandı. Kullanıcının eli ile kullanacağı oyun kumandası yapısında bir donanımın hareketinin, kestirim konusunda yaygın olarak kullanılan, etkinliği birçok araştırmada kabul görmüş olan kalman filtresi ile kestirilmesi, yapılan kestirimde kullanılan yaklaşımların sebeplerinin ve sonuçlarının elde edilen iyileştirmeye etkisinin incelenmesi hedeflendi. Aynı zamanda hareket kestirimi için modüler, geliştirmeye açık, sisteme uyumu kolay bir yazılım kütüphanesi ve bu kütüphanenin kullanılmasıyla kestirim sonuçlarının görselleştirildiği üç boyutlu örnek bir grafik uygulaması geliştirilmesi hedeflendi.

1.2 Ataletsel Ölçüm Ünitesi Kullanılarak Gerçekleştirilen Çalışmalar

AÖÜ kullanılarak gerçekleştirilen çalışmaların büyük bir kısmı hava/kara/deniz araçları (insansız çeşitleri de dâhil) için ataletsel seyir sistemleri tasarlama hedefine yöneliktir. Son yıllarda duyarga teknolojisinin gelişmesi duyargaların küçülmesini ve ucuzlamasını sağlamış, böylece AÖÜ'ler birçok alanda yaygın bir şekilde kullanılmaya başlanmıştır.

Rahni ve Yahya (2007) el yapımı üç boyutlu bir ultrason tarayıcı modülünde üç boyutlu resmi oluşturabilmek için tarayıcının hareketini elde etme ihtiyacı duymuşlar ve kullandıkları bir MEMS 6-DOF AÖÜ üzerinde konum hesaplama üzerine bir çalışma yaparak geleneksel yöntemler ile karşılaştırmışlardır. Yaptıkları çalışmada duyarga verilerinin gürültülü olduğunu ve hatalı sonuçlar elde ettiklerini gözlemlemiş, alçak geçiren filtreler ve kayıklık için getirdikleri yaklaşımlarla biriken hataları azaltmaya çalışmışlardır. Çalışmalarında Xsens firmasının MT9-B modülünü kullanmışlardır. Açısal hız duyargasında oluşan kaymaların yönelim, yönelimde oluşan hataların ve ivmeölçer duyargası hatalarının da konum hesaplamalarında birikerek artan hatalar oluşturduğunu gözlemlemişlerdir. Uyguladıkları yaklaşımlarla büyük ölçüde hataları giderecekleri de belirledikleri uygulama alanında yeterli hassasiyete ulaşamadıkları için sistemin akustik ve optik duyargalar ile desteklenerek daha hassas bir hale getirilmesi gerektiği sonucunu çıkartmışlardır [1].

Foxlin, Harrington ve Altshule (1998) sanal gerçeklik ve benzetim projelerinde kullanılması amacı ile 6-DOF bir AÖÜ ile baş izleme (“head tracking”) uygulaması gerçekleştirmek istemişlerdir. Yaptıkları çalışmada kullanıcının kafasına yerleştirilen bir donanımda 6-DOF AÖÜ bulunmaktadır. Kullanıcının hareket alanı küçük bir bölge ile sınırlandırılmış ve bu bölgenin üzerine yatay olarak L şeklinde bir çubuk yerleştirilmiştir; bu çubuğun iki ucunda ve köşe noktasına ultrasonik mesafe ölçücü duyargalar yer almaktadır. Kullanıcının kafasındaki AÖÜ ile kestirilen konumun ultrasonik ölçümlerden hesaplanan konum bilgisi ile devamlı doğrulanması sağlanmış, böylece son derece hassas ve başarılı bir baş izleme sistemi gerçekleştirmişlerdir. AÖÜ’den ve ultrasonik duyargalardan oluşan dinamik sistemin modellenmesinde kalman filtresi kullanılmıştır [2].

Literatürde yapılan çalışmalara bakıldığında AÖÜ’nün tek başına kullanılmasının etkin sonuçlar üretmediği, destek sistemlerle birlikte kullanılması gerektiği gözlemlenmektedir. İnsansız hava/kara/deniz araçları üzerine yapılan çalışmalar kapalı bir alanda gerçekleştiriliyorsa kameralar ile konum kestirimi yapılarak ataletsel sistemle bütünleştirildiği, açık alanlarda gerçekleştiriliyorsa küresel konumlandırma sisteminin ataletsel sistemle bütünleştirildiği görülmektedir [3].

AÖÜ’ler kullanılarak gerçekleştirilen temel sistemlerden birisi de AHRS (“Attitude and Heading Reference System”)’dir. AHRS özellikle uçaklar için geliştirilen üç eksen veri üreten ataletsel duyargaları kullanarak yönelim bilgisi üreten aygıtlardır. AHRS’nin AÖÜ’den farkı AHRS’nin içersinde bir işlem birimi bulunması ve duyargalarını verilerini alarak yönelim kestirimini bu birim üzerinde yaparak kullanıcıya hesaplanmış yönelimi sağlamasıdır. Yönelim kestirimi dikilme, yuvarlanma ve sapma açılarını kapsamaktadır. Kestirimde kullanılan duyargalardan açısal hız duyargaları ile yönelimin takip edilebilmesi, ivmeölçer ve manyetik alan duyargaları ile yönelimin düzeltilebilmesi olanağı hassas yönelim kestirimini mümkün kılmaktadır. Buna karşın konum değişiminde bu olanak yalnız ataletsel duyargalar kullanıldığında elde edilememektedir. İvmeölçer duyargasına göre konum kestirimi yapılabilmektedir. Bu konumun doğrulanması ise ancak küresel konumlandırma sistemi ile yapılabilmektedir.

AÖÜ'lerin düşük maliyetle üretilebilmesi ve düşük boyutlara inmesi ile oyun sektöründe, mobil aygıtlarda ve sanal gerçeklik uygulamalarında kullanımını artmıştır. İlk olarak ivmeölçer duyargası ile oyun kumandalarına ve mobil cihazlara giren hareket algılama modüllerine yakın zamanda açısız hız duyargaları da eklenmiştir. Bu alanda öncülerden birisi dünyanın önde gelen oyun konsolu ve oyun geliştirici şirketlerinden Nintendo olmuştur. Ürettiği Wiimote oyun kumandasına ilk olarak ivmeölçer daha sonra da açısız hız duyargası ekleyerek ucuz maliyetli 6 DOF bir AÖÜ içerir hale getirmiştir. Açısız hız duyargasının eklenmesi ile daha hızlı hareketlerin daha hassas tespit edilebileceği savunulmuştur. Daha çok belirli bir hareket kümesinden örüntü tanıma tabanlı hareket tahmini yapan Wii oyunları 6 DOF yeni sistem ile daha gelişmiş bir yapıya bürünmektedir. Wiimote kumandasının ucuz elde edilebilirliği ve kişisel bilgisayarlara bağlanabilirliği birçok araştırmacıyı bu kumandayı kullanmaya teşvik etmiştir. Kumandanın kızılötesi ledleri takip edebilen kamera modülü kullanılarak birçok çalışma yapılmakta fakat içersindeki 6DOF AÖÜ kullanılarak üç boyutlu hareket tahminine yönelik henüz yeterli çalışma bulunmamaktadır. Bunda açısız hız duyargasının kumandaya yakın zamanda eklenmiş olmasının da etkisi vardır. Mobil platformlardan örnek verilecek olursa iPhone 4 sürümü ile aygıtta açısız hız duyargası eklenmiş ve 6 DOF bir AÖÜ içerir hale getirilmiştir. AÖÜ içeren modüllerin bu şekilde ucuz elde edilebilir ve kullanılabilir hale gelmesi birçok araştırmacı için bir fırsata dönüşmektedir ve bu alanda yapılan çalışmaların büyük ölçüde artışını sağlamaktadır.

2. HAREKET DUYARGALARI VE KULLANIMI

Bu bölümde ilk olarak AÖÜ'nün ne olduğu, ne amaçla hangi alanlarda kullanıldığı ve geçmişten günümüze gelişimi hakkında bilgi verilecektir. AÖÜ'nün temel olarak içerdiği duyargalar ve bu duyargalardan elde edilen veriler anlatılacaktır. Dengeli (“Gimbaled”) ve sabit (“Strapdown”) AÖÜ kavramları açıklandıktan sonra tez kapsamında kullanılan, sabit bir AÖÜ içeren Nintendo Wiimote oyun kumandası hakkında bilgi verilecektir.

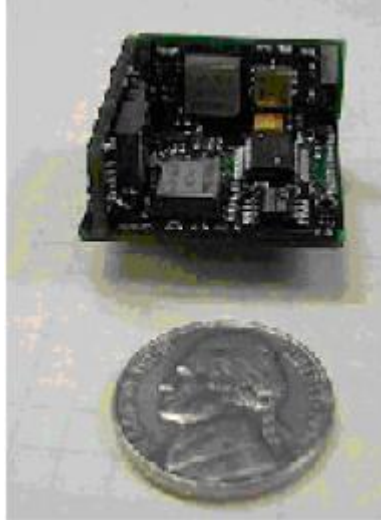
2.1 Ataletsel Ölçüm Ünitesi

AÖÜ'ler genellikle uçak, gemi, füze benzeri platformların hızı, yönelimi, eksenleri üzerine düşen yer çekimi ivmesi kuvveti gibi bilgilerin hesaplanabilmesi için kullanılan temel olarak ivmeölçerlerden ve açısal hız duyargalarından oluşan elektronik donanımlardır. Ataletsel seyir sistemlerinin temel bileşeni olan bu donanımsal modül ataletsel güdüm sistemlerinin tasarlanmasında kullanılır. Mekanik tasarım ağırlıklı, sistem hatasını mümkün olduğunca küçük tutmayı amaçlayan ve çok yüksek maliyetleri bulunan bu modüllere bir örnek Şekil 2.1'de verilmiştir [4]. Bu modül Fransız yapımı IRBM S3 füzelerinde bulunan AÖÜ'dür.



Şekil 2.1 : IRBM S3 Füzelerinin AÖÜ'sü

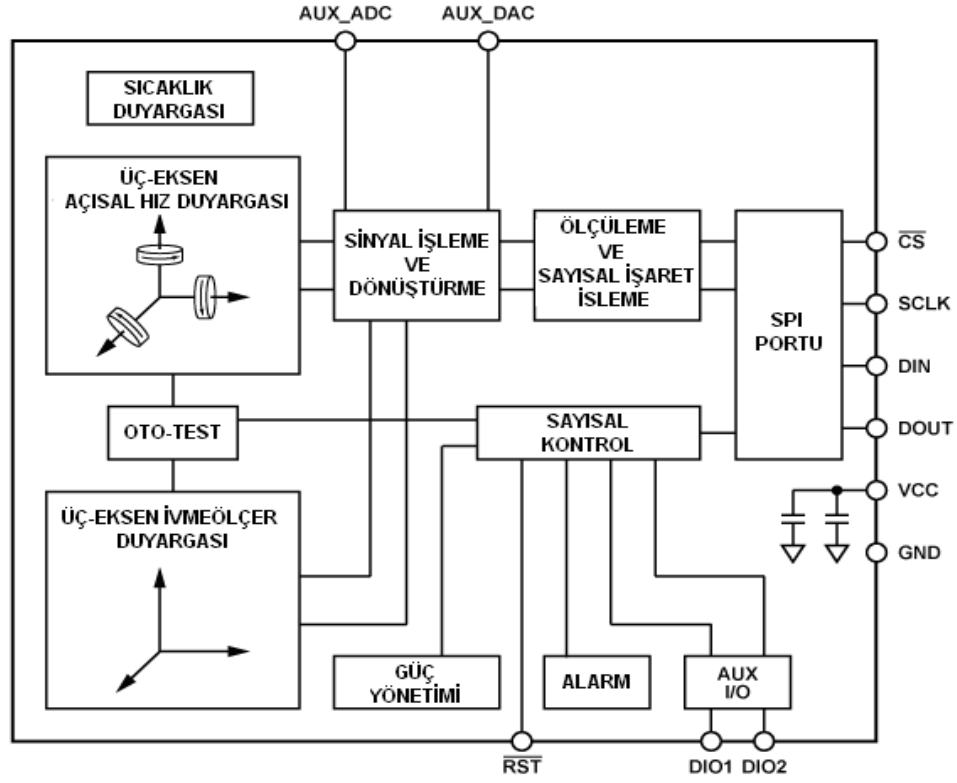
Son yıllarda MEMS teknolojisi çok hızlı bir gelişim göstermektedir. MEMS, mekanik elementlerin ve duyargaların silikon tabanlı mikro-üretim teknolojisi ile üretilmesi sonucu bu sistemlere verilen genel isimdir. Mikro-duyargaların üretilmesi ve mikro-devrelerle kontrol edilebilmesi tasarlanmak istenen bütün bir sistemin tek bir yonga üzerinde gerçekleştirilmesine olanak sağlamıştır. Günümüz şartlarında MEMS teknolojisinin gelişimiyle üretim olanağına erişilen 6 DOF modüllerden birine örnek Şekil 2.2’de verilmiştir;



Şekil 2.2 : Cloud Cap “Crista” 6DOF AÖÜ

Boyut olarak günümüz şartlarında kullanılan AÖÜ’lere göre yaklaşık 100 kat daha küçük ve daha düşük maliyetli olarak temin edilebilen bu ürünlerde üç eksen ivmeölçer ve üç eksen açısız hız duyargasının çıktılarını sayısal olarak elde edilebilmektedir. Bu da donanımın I2C/SPI gibi seri iletişim birimi içeren modüllerden herhangi biri ile kolaylıkla kullanılabilmesine olanak sağlamaktadır [5].

6 DOF AÖÜ’lerin genel yapısının görülmesi için Analog Devices’in ürettiği ADIS16350/ADIS16355 duyargasının fonksiyonel blok diyagramı Şekil 2.3’de verilmiştir [6]. Türkçeleştirilerek aktarılan şekilde de görüldüğü gibi donanımda üç eksen ivmeölçer, üç eksen açısız hız duyargası ve sıcaklık duyargası bulunmaktadır. Duyarga çıktıları bir mikro denetçi vasıtasıyla ilk olarak sinyal işleme ve çevrim birimine, daha sonrada ölçüleme (“calibration”) ve sayısal işarete çevrim birimine girer. Daha sonra elde edilen sayısal veriler SPI seri iletişim arabirimi üzerinden dışarıya verilir. Örnek modülde seri iletişim arabirimi olarak SPI kullanılmıştır, aynı zamanda I2C destekleyen modüller de yaygın bir şekilde bulunmaktadır.



Şekil 2.3 : ADIS16350/ADIS16355 Blok Diyagramı

2.1.1 İvmeölçer Duyargası

İvmeölçerler ivmeye bağlı kuvvetleri ölçen elektromekanik cihazlardır. İvmeölçerler bir nesneye etki eden yerçekimi kuvvetine bağlı statik ivmeyi ölçtüğü gibi bu nesnenin hareketine bağlı dinamik ivmeyi de aynı anda ölçmektedir. Ölçülen verinin yerçekimi ivmesinden mi yoksa nesnenin hareketinden mi kaynaklandığının ayrımı duyurga tarafından yapılamaz. İvmeölçerler G-kuvveti üzerinden ölçüm yaparlar. G-kuvveti serbest düşüşte olan bir nesneye (yerçekimi ivmesi ile düşen) etki eden kuvvettir. İvmeölçerler uygulama alanlarına göre farklı G-kuvveti hassasiyetlerinde çalışabilmektedir. Kapasitif, piezoelektrik, piezoresistif, hall etkisi, manyetoresistif ve ısı transferinden faydalanarak çalışan çeşitleri mevcuttur.

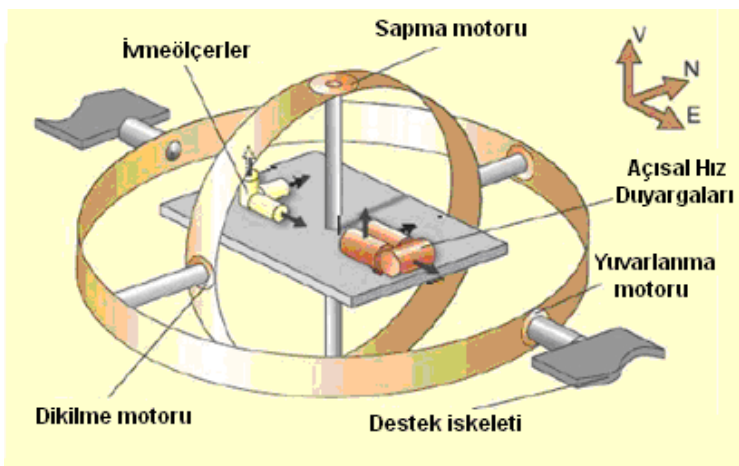
İvmeölçerden alınan ivme bilgisi ilgili eksendeki statik ve dinamik değer toplamını verir. Birden fazla eksende bilgi veren ivmeölçer duyargasının çıktıları kullanılarak dikilme, yuvarlanma, titreşim, hız ve konum gibi bilgilerinin elde edilmesi mümkündür.

2.1.2 Açısal Hız Duyargası

Açısal hız duyargaları bir nesnenin yönelimini belirlemek için açısal momentumun korunumu prensibine dayalı çalışan cihazlardır. Açısal hız duyargası hareketi merkezkaç ilkesine dayalıdır. Mekanik olarak tasarlanan ilk açısal hız duyargalarının temeli 1800’lü yıllara dayanmaktadır. 1900’lü yılların ortasından günümüze gelindiğinde açısal hız duyargası teknolojisinde büyük adımlar atılmış ve bu duyargalar uçakların ve gemilerin ataletsel seyir sistemlerinde kullanılan temel bir modül haline gelmiştir. MEMS teknolojisi ile çok küçük boyutlara inen ve ucuza mal edilebilen açısal hız duyargaları oyun kumandalarından cep telefonlarına kadar birçok üründe kullanılmaya başlanmıştır. Temel olarak açısal hız duyargası bir nesnenin ilgili eksenini etrafındaki dönüş hızını ($^{\circ}/s$) vermektedir. Bu yaklaşım nesnenin bütün eksenleri için düşünüldüğünde nesnenin yönelimini hesaplama için dikilme (“pitch”), yuvarlanma (“roll”) ve sapma (“yaw”) açılarının elde edilmesine olanak sağlamaktadır.

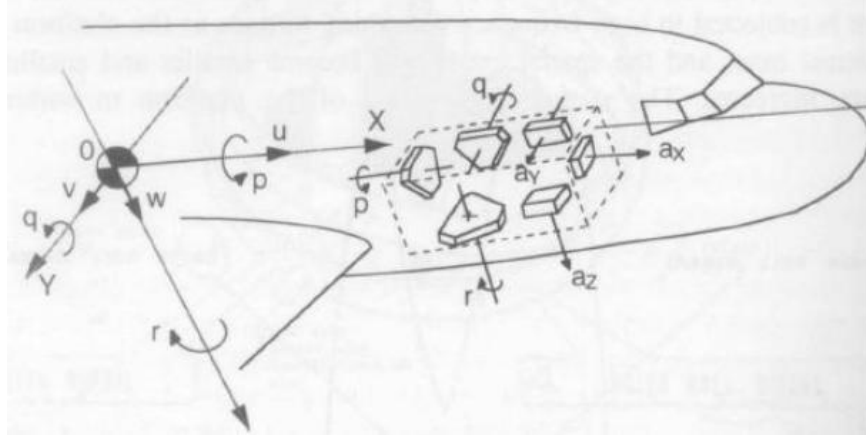
2.1.3 Dengeli ve Sabit Ataletsel Ölçüm Üniteleri

Dengeli (“Gimbaled”) AÖÜ’lerde ivmeölçer mekanik bir sistem sayesinde nesnenin yöneliminden etkilenmez böylece ivmeölçer eksenleri devamlı sabit kalır. İvmeölçer tarafından elde edilen doğrusal ivme bilgileri üzerinde nesnenin yönelimine göre düzeltme yapma ihtiyacı yoktur, bu da sistemin daha kararlı çalışmasını sağlar. Şekil 2.4’de örnek bir dengeli AÖÜ’nün yapısı verilmiştir [7].



Şekil 2.4 : Dengeli AÖÜ

Sabit (“Strapdown”) AÖÜ’lerde ise ivmeölçer ve açısal hız duyargası donanıma sabitlenmiştir ve ivmeölçerlerden elde edilen doğrusal ivme verileri yönetime göre üç boyutlu fiziksel dünyada farklı eksenleri ifade eder. Sabit AÖÜ’lerde bir nesnenin üç boyutlu uzaydaki ivmelerinin elde edilebilmesi için yönelim bilgisini kullanarak ivmeölçer ekseninden üç boyutlu uzay eksenlerine dönüşüm yapmak gerekmektedir. Bu da sistemin hatasını arttırmaktadır. Şekil 2.5’de örnek bir sabit AÖÜ’nin yapısı verilmiştir [7].



Şekil 2.5 : Sabit AÖÜ

Sabit AÖÜ’ler dengeli AÖÜ’lere göre daha kolay ve düşük maliyetle üretilebilir. Bu yüzden oyun kumandalarında, mobil aygıtlarda ve benzeri sistemlerde sabit AÖÜ’ler kullanılır. Tez kapsamında kullanılan örnek donanım da bir oyun kumandasıdır ve sabit AÖÜ içermektedir. Bir sonraki bölümde ilgili donanım tanıtılacaktır.

2.2 Nintendo Wii Donanımı ve Özellikleri

Tez çalışması kapsamında örnek donanım olarak dünyanın en büyük oyun konsolu ve oyun geliştirici firmalarından birisi olan Japon Nintendo’nun Wiimote oyun kumandası kullanılmıştır. Wiimote kullanıcının tek elde tutarak kullanması amacıyla tasarlanmış, 148 mm uzunluğunda, 36.2 mm genişliğinde ve 30.8 mm kalınlığında uzaktan kontrol tabanlı bir donanımdır, Şekil 2.6’da Wiimote kumandası verilmiştir.



Şekil 2.6 : Nintendo Wiimote

Wiimote oyun kumandasında ivmeölçer olarak Analog Devices'in ADXL330 modeli kullanılmıştır. ADXL330 4x4x1.45 mm boyutlarında, 1.8V'da 180uA akım ile son derece düşük güç tüketimine sahip, üç eksenli analog bir ivmeölçerdir. ADXL330 -3g ve +3g aralığında ölçüm yapabilmektedir, sıcaklık değişimlerine karşı kararlı çıktılar üretebilmektedir ve mobil cihazlarda, oyun kumandalarında, spor ve sağlık sektöründeki ürünlerde kullanılması amacıyla üretilmiştir.

Wiimote oyun kumandasında ataletsel ölçüm duyargası olarak yalnızca ivmeölçer bulunmaktadır. 2009 yılının ortalarında Nintendo Wiimote için Wii Motion Plus eklentisini çıkarmıştır. Şekil 2.7'de verilen Wii Motion Plus eklentisi kumandaya alt kısmından takılmaktadır, üç eksen açısal hız duyargası içermektedir ve daha karmaşık hareketlerin daha doğru bir şekilde algılanmasına olanak sağlamıştır. Üç eksenli ivmeölçer ve açısal hız duyargası ile hareketlerin gerçek zamanlı olarak yakalanabileceği ifade edilmiştir. Wii Motion Plus modülünde üç eksen açısal hızların elde edilebilmesi için Invensense firmasının ürettiği, dikilme ve yuvarlanma bilgilerini veren iki eksenli IDG-600 açısal hız duyargası ile birlikte sapma bilgisini veren tek eksen bir açısal hız duyargası birleştirilmiştir. IDG-600 4x5x1.2 mm boyutuyla MEMS piezoelektrik teknolojisi kullanılarak üretilmiş dünyanın en küçük açısal hız duyargalarından birisidir. İçersinde sıcaklık duyargası da bulunmaktadır. Oyun sektöründe kullanılacağı için benzerlerinden farklı olarak yüksek hızları yakalayabilmektedir. 440°/s ve 2000°/s'e kadar veri üretebilen iki modu bulunmaktadır ve bu özellik, oyun kumandasında agresif hareketlerin algılanabilmesini sağlamaktadır.



Şekil 2.7 : Nintendo Wii Motion Plus

Wiimote oyun kumandasının üst kısmında bir kamera duyargası bulunmaktadır. Ekranın altına veya üstüne yerleştirilen, sağ ve sol uçlarında kızılötesi ledler bulunan bir çubuk sayesinde kızılötesi ışık kaynakları kamera duyargasında tespit edilmekte ve bu bilgiler kullanılarak oyun kumandasının fare gibi kullanılabilmesi de mümkün olmaktadır.

Wiimote oyun kumandası 2 adet AA pil ile çalışmaktadır, Intel 8051 ailesinden bir mikro-işlemci ile çalışmaktadır, 128kbit'lik bir EEPROM'a sahiptir, duyurgalarla iletişim, I2C seri iletişim arabirimi üzerinden gerçekleştirilmektedir ve elde edilen duyurga bilgileri mavidiş (bluetooth) üzerinden kablosuz olarak aktarılmaktadır [8].

3. HAREKET DUYARGALARI İLE KESTİRİM

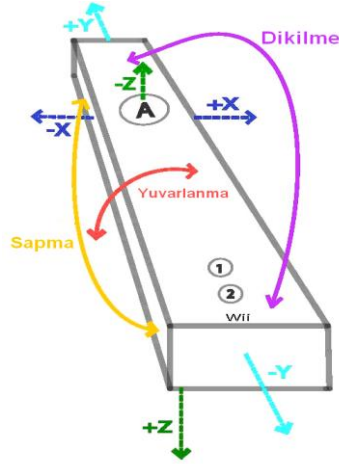
Bir nesnenin üç boyutlu uzaydaki hareketinin kestirilmesi, yöneliminin ve konumunun kestirilmesini ifade etmektedir. Bu bölümde ilk olarak yönelim hesaplamasının hangi duyarga verileri ile nasıl yapılabileceği anlatılacaktır. İvmeölçer ile anlık yönelim geçmiş yönelimden bağımsız olarak hesaplanabilmekteyken, açısal hız duyargası ile birikimsel olarak; yani geçmiş yönelim de kullanılarak hesaplanabilmektedir. Bu aşamada ardışıl yönelimlerin birbiri ile ilişkisi matematiksel olarak ifade edilmelidir ve bu özellik yönelim ifade tekniklerini önemli kılmaktadır. Belirtilen çerçevede nesne yöneliminin kolay anlaşılması açısından ilk olarak dikilme, yuvarlanma ve sapma açıları şeklindeki ifade tekniği daha sonra da koordinat dönüşüm matrisi ve dördey şeklindeki ifade teknikleri yararları ve sakıncalarıyla açıklanacaktır. Ardışıl yönelim ilişkilerinin denklemleri verilecek, farklı yönelim ifade tekniklerinin birbirlerine nasıl dönüştürülebileceği anlatılacaktır. Yönelim hesaplamalarının ardından konum hesaplamalarının hangi duyarga ile nasıl yapılacağı ve yönelimin konum hesaplamadaki etkileri açıklanacaktır. Hareket hesaplama işlemlerinin teorik yaklaşımları verildikten sonra neden gerçek hayatta zor ve karmaşık bir probleme dönüştüğü sebepleri ile ifade edilecektir. Bu problemlerin üstesinden gelmek için getirilmesi gereken kestirim (“estimation”) ve takip (“tracking”) kavramları anlatılacaktır. Kestirim ve takip işlemleri için tez kapsamında kullanılan Kalman Filtresi; dinamik sistemin filtreye uygun şekilde nasıl modellenmesi gerektiği ve filtrenin nasıl işlediği ayrıntılandırılarak açıklanacaktır.

3.1 Yönelim ve Doğrusal Hareket Hesaplamaları

Bir nesnenin üç boyutlu uzayda hareketinin algılanması için nesnenin dönüş hareketlerine göre yönelimi ve doğrusal hareketlerine göre hızı ve konumu hesaplanmalıdır. Bir nesnenin yöneliminin ve konumunun belirli bir süre boyunca takip edilmesi, o nesnenin bu sürede yaptığı hareketi ifade eder ve bu bilgiler için nesne üzerine yerleştirilen AÖÜ verileri kullanılır. Temel olarak, yönelim açısal hız duyargası, doğrusal hareket ise ivmeölçer ile hesaplanmasına rağmen bu iki duyarganın verilerinin birleştirilmesi ve ortak yorumlanması gereken durumlar problemi daha karmaşık hale getirmektedir.

3.1.1 Yönelim Hesaplamaları

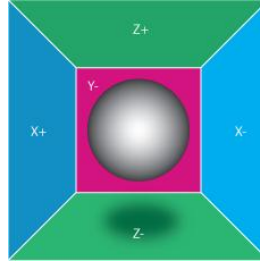
Yönelim temel olarak bir nesnenin dikilme, yuvarlanma ve sapma açıları ile ifade edilir ve bu açılar nesnenin x, y ve z eksenleri etrafında dönmesi sonucu elde edilir. Wiimote oyun kumandası üzerinde dikilme, yuvarlanma ve sapma hareketleri Şekil 3.1'de görülmektedir [9].



Şekil 3.1 : Dikilme, Yuvarlanma ve Sapma Hareketleri

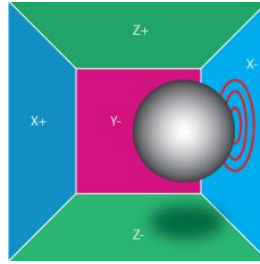
Yönelim bilgisi ivmeölçer ve açısal hız duyargası tarafından hesaplanabilmektedir. İvmeölçer kullanarak yönelim hesaplama, duyarganın yer çekimi ivmesini ölçebilmesi sayesinde mümkün olmaktadır. Bu yaklaşım ivmeölçere dinamik hareket verildiğinde kullanılamaz, çünkü eksenlerden gelen ivme verilerinin yerçekimi ivmesinden mi yoksa hareketinden mi kaynaklandığı bilinemez. Hareketsiz kabul edilebilecek bir ivmeölçerden elde edilen veriler yalnızca yerçekimi kuvvetinden kaynaklanacağı için eksenlere düşen kuvvetin oranlarından yönelim elde

edilebilmektedir. Yaklaşımın daha iyi anlaşılabilmesi için ivmeölçer Şekil 3.2’de görüldüğü gibi içersinde bir top bulunan kapalı bir küp şeklinde düşünülebilir.



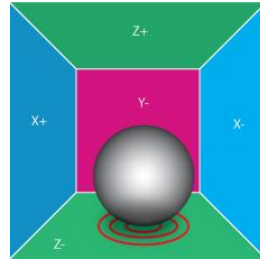
Şekil 3.2 : Yerçekimsiz Ortamda İvmeölçer

Yerçekimsiz bir ortamda ivmeölçer üzerine herhangi bir kuvvet etki etmemektedir, bu yüzden top hareketsiz şekilde küpün ortasında bulunmaktadır. Yerçekimsiz bir ortamda x eksen yönünde $1g$ 'lik bir ivme ile hareket mevcutsa Şekil 3.3’de görüldüğü gibi top x eksen üzerine $-1g$ 'lik kuvvet uygular.



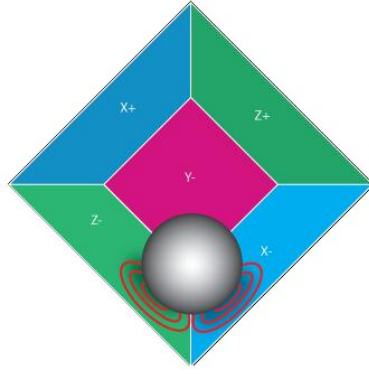
Şekil 3.3 : Yerçekimsiz Ortamda Hareket Eden İvmeölçer

İvmeölçer yaklaşımını yerçekimsiz ortamda değerlendirdikten sonra yerçekimi etkisine geçilebilir. İvmeölçer olarak düşünülen küpün yeryüzü üzerinde z eksenine yere gelecek şekilde hareketsiz şekilde durduğunu varsayarsak, Şekil 3.4’de gösterildiği gibi z eksenine $-1g$ 'lik yerçekimi kuvveti etki etmektedir.



Şekil 3.4 : Yerçekimi Etkisinde İvmeölçer

İvmeölçer olarak düşünülen küpün Şekil 3.5’de görüldüğü gibi yan çevrildiği düşünülürse, küp içersindeki top, hem x, hem de z eksenine kuvvet uygular ve toplam $1g$ 'lik yerçekimi kuvveti küpün duruş açısına göre bu iki eksene yayılır.



Şekil 3.5 : Yerçekimi Etkisinde Döndürülmüş İvmeölçer

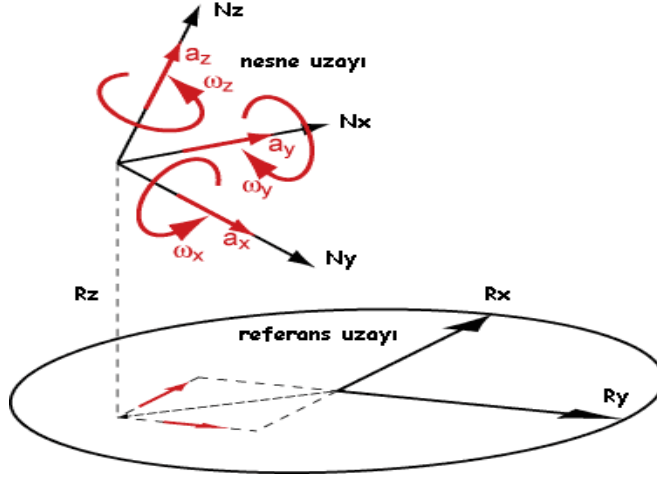
Hareketsiz durumdaki ivmeölçere etkileyen toplam kuvvet yaklaşık $1g$ 'dir ve bu kuvvetin eksenler üzerine düşen bileşenlerinin oranları ile dikilme ve yuvarlanma açıları elde edilebilmektedir. Nesnenin sapma açısının değişimi esnasında yerçekimi kuvvetinin etki ettiği eksenler değişmediği için ivmeölçer ile sapma açısı hesaplamak mümkün değildir, yani elde edilebilecek yönelim açıları yalnızca dikey eksen değişimlerinden elde edilebilmektedir, bu yüzden elde edilen yönelim üç boyutlu değildir. İvmeölçerden x , y ve z eksenleri için elde edilen değerlerin a_x , a_y , a_z , yuvarlanma açısının ϕ , dikilme açısının θ olduğu kabul edilirse belirtilen hesaplama aşağıdaki gibi yapılır (3.1).

$$\phi = \arctan(-a_y, -a_z) \quad (3.1)$$

$$\theta = \arcsin(-a_x / \sqrt{a_x^2 + a_y^2 + a_z^2})$$

Yönelim hesabında kullanılan temel duyarga açısal hız duyargasıdır. Üç eksenli bir açısal hız duyargası nesnenin x , y ve z eksenleri etrafındaki dönüş hareketleri için %/s cinsinden hız bilgisi vermektedir. Başlangıç yönelimi bilinen bir nesnenin hareketi süresince elde edilen hız bilgilerinin zamana göre integre edilmesiyle son yöneliminin bulunması mümkündür [10]. Bu integrasyon işlemi nesnenin açısal hız verileri ile bu verilerin yönelime etkisi arasındaki bağlantının çıkartılması gerekmektedir. Bu işlemdeki gereklilik nesneden elde edilen açısal hız verilerinin son yönelime göre yorumlanması ihtiyacından kaynaklanmaktadır. Basit bir örnekle bu durumu açıklayacak olursak; bütün yönelim açıları sıfır derece olan bir nesne x eksenleri etrafında 45 derecelik bir dönüş yaptığında, bu hareket dikilme açısında 45 derecelik bir değişimi ifade eder. Buna karşın yuvarlanma açısı 45 derece olan bir

nesne x eksenini etrafında 45 derecelik bir dönüş yaptığında bu yönelim hem sapma hem de dikilme açısından değişikliğe sebep olur. Bu yüzden nesnenin yönelimi kullanılarak açısal hız değerlerinde bir dönüşüm gerçekleştirme ihtiyacı doğmaktadır. Bir önceki yönelimin kullanılmasının gerekliliği açısal hız duyargası tabanlı yönelim hesaplamada birikimli bir yapı kullanılacağını gösterir. Nesne yöneliminin anlık açısal hızlar ve bir önceki yönelim ile ilişkisinin elde edilmesine ihtiyaç vardır. Bu sebeple nesne yöneliminin farklı ifade biçimleri değerlendirilmelidir. Başlangıçta dikilme, yuvarlanma ve sapma açıları temelli yaklaşım ele alınarak açıklama yapılmıştı. Bununla birlikte koordinat dönüşüm matrisi tabanlı ve dördey tabanlı yaklaşımlar ele alınacak; açısal hız verileri ile yönelim arasındaki bağıntılar ve farklı yönelim ifadelerinin birbirlerine dönüşümleri ile ilgili formüller verilecektir. Bu bölüme geçmeden önce ilerideki bölümlerde sıkça kullanılacak olan nesne uzayı ve referans uzayı kavramları açıklanacaktır. Bu kavramlar ile ilgili olarak Şekil 3.6 incelenirse [11]; nesne uzayı kullanılan donanımın koordinat eksenlerini ifade etmektedir. N_x , N_y ve N_z nesne koordinatlarıdır, ivmeölçer verileri bu eksenlerdeki ivme verileri (a_x , a_y , a_z) olarak elde edilir, açısal hız duyargası verileri de bu eksenler etrafındaki dönüş hızları (w_x , w_y , w_z) olarak elde edilir. Referans uzayı ise donanımın kullanıldığı ortamda belirlenen bir koordinat sistemini ifade eder. Örneğin donanım bir oda içerisinde televizyon karşısında kullanılıyorsa, oda içerisindeki konumunu belirtmek için belirlenen koordinat sistemidir. R_x , R_y , R_z referans koordinatları sırası ile donanımın bulunduğu noktadan televizyon ekranına çizilen eksen, donanımın bulunduğu noktadan geçen, televizyon ekranına paralel çizilen eksen ve donanımın bulunduğu noktadan tavana doğru dikine çizilen eksen olarak ifade edilebilir. Benzer şekilde bir uçağın hareketi düşünülürse, referans uzayı eksenlerinden R_x 'in kuzeyi, R_y 'nin doğuyu ve R_z 'nin yüksekliği gösterecek şekilde belirlendiği düşünülebilir. Donanımın hareketi referans uzayında tanımlıdır, dolayısıyla nesne koordinatlarına göre elde edilen duyarga verilerinin referans koordinatlarına dönüştürülmesi gerekmektedir. Bu dönüşüm donanımın yönelimi kullanılarak gerçekleştirilir ve daha ayrıntılı bilgi ilerleyen bölümde verilecektir.



Şekil 3.6 : Nesne Uzayı ve Referans Uzayı

3.1.2 Yönelim İfade Teknikleri

Bu bölümde euler açıları, koordinat dönüşüm matrisi ve dördey tabanlı yönelim ifade teknikleri ile ilgili teorik bilgiler verilecektir.

Euler Açıları Yöntemi

İlk yöntem yönelimi Euler açıları şeklinde tutmaktır. Böyle bir durumda; γ yönelim, θ dikilme açısı, ϕ yuvarlanma açısı, ψ sapma açısı olmak üzere herhangi bir k anındaki yönelim aşağıdaki gibi ifade edilir (3.2).

$$\gamma_k = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (3.2)$$

Nesne uzayına göre elde edilen açısal hız verilerinin referans uzayı eksenlerine dönüştürülmesi için nesnenin yönelimi kullanılarak B matrisi oluşturulur (3.3) [12].

$$B = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \quad (3.3)$$

γ_k ve γ_{k+1} ardışıl iki yönelimi, w_x x eksenini etrafındaki, w_y y eksenini etrafındaki, w_z z eksenini etrafındaki dönüş hızını ve t ; k ve k+1 zamanları arasındaki farkı gösterecek şekilde, yönelim aşağıdaki gibi hesaplanır (3.4).

$$\gamma_{k+1} = \gamma_k + B \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} t \quad (3.4)$$

B matrisinde dikilme açısı $\mp \pi/2$ değerlerine yaklaşırken, $\cos \theta$ değeri sıfıra yakınsayacağı için $\tan \theta$ ve $\sec \theta$ hesaplamalarında sıfıra bölme problemi oluşmaktadır. Bu nedenle sınır değerleri için ek denetimler yapılmalıdır. Yönelimi açı tabanlı ifade etmek bu yüzden etkin bir yöntem değildir.

Koordinat Dönüşüm Matrisi Yöntemi

İkinci yöntem yönelimi koordinat dönüşüm matrisi (3.5) şeklinde tutmaktır [13];

$$C_i^n = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \phi \sin \theta - \sin \psi \cos \phi & \cos \psi \sin \phi \cos \theta + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \theta & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (3.5)$$

Koordinat dönüşüm matrisi C_i^n alt indisten üst indise olan dönüşümü ifade eder. Buradaki alt indis i nesne uzayı, üst indis n ise referans uzayı ifade eder. Bu şekilde düşünülürse C_i^n matrisi nesne uzayından referans uzayına dönüşü ifade eder. C_i^n matrisinden tersine dönüşüm yapıldığında yönelim açıları elde edilir (3.6) [13];

$$\phi = \arctan(C_{i3,2}^n, C_{i3,3}^n) \quad (3.6)$$

$$\theta = -\arcsin(C_{i3,1}^n)$$

$$\psi = \arctan(C_{i2,1}^n, C_{i1,1}^n)$$

Koordinat dönüşüm matrisinin ardışıl zamanlardaki ilişkisini çıkartırken açısal hız vektörünün çarpık-bakışlımlı (“skew-symmetric”) matrisi η kullanılır (3.7) [13];

$$\eta = \begin{pmatrix} 0 & -w_z & -w_y \\ w_z & 0 & -w_x \\ w_y & w_x & 0 \end{pmatrix} \quad (3.7)$$

$C_i^n(k+1)$ ve $C_i^n(k)$ ardışıl iki yönelimi ve t aralarındaki zaman farkını, I_3 de 3×3 'lük birim matrisini ifade ederse, yönelim hesabı aşağıdaki gibi olur (3.8) [14].

$$C_i^n(k+1) = C_i^n(k)(I_3 + \eta t) \quad (3.8)$$

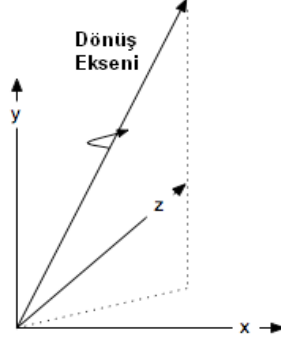
$$(I_3 + \eta t) = \begin{pmatrix} 1 & -w_z t & -w_y t \\ w_z t & 1 & -w_x t \\ w_y t & w_x t & 1 \end{pmatrix}$$

Dördey Yöntemi

Son olarak dördey tabanlı yönelim yaklaşımı ele alınırsa q_x , q_y , q_z ve q_s dördeyin bileşenleri olmak üzere q_k ; k anındaki yönelimi ifade eder (3.9).

$$q_k = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (3.9)$$

Dördey dört bileşeni olan hiper karmaşık (“hyper-complex”) bir sayıdır [13]. Karmaşık sayıların iki boyutlu düzlemde kutupsal koordinatlarla yönelim belirtmesi gibi hiper karmaşık sayılar da üç boyutlu uzayda yönelimi ifade etmektedir. q_x , q_y , q_z değişkenleri üç boyutlu uzayda bir eksen, q_s de bu eksen etrafındaki dönüşün büyüklüğünü ifade eder. Böylece bir cismin üç boyutlu dönme hareketi dördey yöntemi ile Şekil 3.7’deki gibi tek bir eksen etrafındaki belirli büyüklükteki bir dönüşle ifade edilir [15].



Şekil 3.7 : Dördeyde Tek Eksene İndirgenmiş Dönüş

Dördey şeklindeki yönelimin ardışıl zamanlardaki ilişkisini çıkartırken açısal hız vektörleri ile elde edilen dördey omega matrisi Ω kullanılır (3.10) [15].

$$\Omega = \begin{bmatrix} 0 & -w_x & -w_y & -w_z \\ w_x & 0 & w_z & -w_y \\ w_y & -w_z & 0 & w_x \\ w_z & w_y & -w_x & 0 \end{bmatrix} \quad (3.10)$$

T : dönüşüm matrisini, I_4 :4x4'lük birim matrisi ve t : k+1 ve k anları arasındaki zaman farkını ifade ederse [15];

$$T = I_4 + \frac{1}{2} \Omega t \quad (3.11)$$

q_{k+1} ve q_k ardışıl iki yönelimi ifade ederse ilişki aşağıdaki gibi verilir (3.12) [15];

$$q_{k+1} = T q_k \quad (3.12)$$

Dördeyin yönelimi ifade edebilmesi için normalize edilmesi gerekmektedir. Normalize edildikten sonra aşağıdaki gibi 3x3'lük DCM ("Directional Cosine Matrix") 'ye çevrilebilir (3.13) [15]. DCM matrisi koordinat dönüşüm matrisinin (3.5) devriğidir ("transpose").

$$DCM = D = \begin{bmatrix} 1 - 2(q_y^2 + q_z^2) & 2(q_x q_y + q_s q_z) & 2(q_x q_z - q_s q_y) \\ 2(q_x q_z - q_s q_y) & 1 - 2(q_x^2 + q_z^2) & 2(q_y q_z + q_s q_x) \\ 2(q_x q_z + q_s q_y) & 2(q_y q_z - q_s q_x) & 1 - 2(q_x^2 + q_y^2) \end{bmatrix} \quad (3.13)$$

Dördeyden euler açıları elde edilmek istenirse aşağıdaki dönüşüm denklemleri kullanılır (3.14) [15].

$$\phi = \arctan(2(q_y q_z + q_s q_x)/(1 - 2(q_x^2 + q_y^2))) \quad (3.14)$$

$$\theta = \arcsin(2(q_x q_z - q_s q_y))$$

$$\psi = \arctan(2(q_x q_y + q_s q_z)/(1 - 2(q_y^2 + q_z^2)))$$

3.1.3 Doğrusal Hareket Hesaplamaları

Bir nesnenin üç boyutlu uzayda konumunu hesaplamak için kullanılan temel duyurga ivmeölçerdir. Konumun zamana göre değişimi hızı, hızın da zamana göre değişimi ivmeyi verdiği için, ivmeölçerden alınan değerlerin zamana göre integre edilmesi ile eksenlerdeki hızlar, elde edilen hızların zamana göre integre edilmesi ile de konum bilgisi üretilebilmektedir. V_k ve V_{k+1} k ve k+1. örnekleme zamanındaki ardışıl hızları, X_{k+1} ve X_k ardışıl konumları ve a ivmeyi gösterirse integrasyon aşağıdaki gibi ifade edilir (3.15). Burada tek bir eksendeki hız ve konumun ifadesi verilmiştir. Üç boyutlu uzaydaki hız ve hareket için bu hesaplama üç eksen de yapılmalıdır.

$$V_{k+1} = V_k + at \quad (3.15)$$

$$X_{k+1} = X_k + V_k t + \frac{1}{2} at^2$$

Dengeli AÖÜ'lerin doğrusal hareket hesaplama işlemindeki avantajı nesnenin yönelimine göre ivmeölçer eksenlerinin değişmemesidir. Böylece ivmeölçerin x, y ve z eksenlerinden elde edilen ivme verileri referans uzayında da x, y ve z eksenlerine karşılık düşer, böylece hız ve konum hesapları doğrudan gerçekleştirilebilir. Sabit AÖÜ'lerde ise ivmeölçer platform üzerinde sabit olduğu için nesnenin yönelimi değiştikçe ivmeölçerin x, y ve z ekseninden referans uzayın x, y ve z eksenine dönüşüm yapma zorunluluğu doğmaktadır. Bu yüzden ilk olarak nesnenin yönelimi hesaplanmalı, daha sonra hesaplanan bu yönetime göre eksen dönüşümü gerçekleştirilmelidir. Bu dönüşümde koordinat dönüşüm matrisi (3.5) kullanılır. a_{nx} , a_{ny} , a_{nz} referans uzaydaki x,y ve z ivmelerini, a_x , a_y , a_z nesne uzayındaki ivmeleri ifade ederse dönüşüm aşağıdaki gibi yapılır (3.16) [16].

$$\begin{bmatrix} a_{nx} \\ a_{ny} \\ a_{nz} \end{bmatrix} = C_i^u \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad (3.16)$$

3.2 Ataletsel Ölçüm Ünitelerinde Karşılaşılan Problemler

AÖÜ kullanarak hareket hesaplama yalnızca teorik yaklaşım göz önüne alındığında basit bir problem olarak görünmekte, fakat ivmeölçer ve açısal hız duyargalarındaki hatalar da hesaplama katıldığında hassas ve gerçek sonuç bulmak karmaşık ve zor bir probleme dönüşmektedir. Hem ivmeölçer verilerine göre konum hesaplamada hem de açısal hız verilerine göre yönelim hesaplamada bir önceki konum ve yönelimin kullanılması gerekmektedir. Tekrarlamalı olarak çalışan bu sistemde hesaplanan veriler her döngüde sonuca eklenmekte, böylece hatalar birikerek gerçek hareketten farklı sonuçlar elde edilmektedir.

İvmeölçeri ele alırsak başta baz (“bias”) ve gürültü olmak üzere ölçeklendirme etkisi, sıcaklık modeli, analog-sayısal çevirici hatası, yerçekimi ivmesi değişimi gibi birçok etken duyarga çıktısını etkilemekte ve hatalara sebep olmaktadır. Baz ivmeölçere hiçbir kuvvet uygulanmadığında elde edilen çıktıyı ifade etmektedir. Gürültü ivmeölçerin iç ölçümlerinde oluşan rastgele hataları ve bulunduğu ortamda meydana gelen titreşimlerden kaynaklanan hataları ifade etmektedir. Ayrıca bir nesneye etki eden yerçekimi kuvvetinin nesnenin dünyadaki konumuna göre küçük farklılıklar gösterdiği göz önüne alınır, duyarga çıktısında ölçüleme hatalarının oluşması mümkündür. AÖÜ’nün kullanım amacına ve uygulamada hedeflenen hassasiyete göre gürültü modelleme ve giderme için ayrıntılı yaklaşımlar getirilebilmektedir.

Açısal hız duyargası ele alınır, ivmeölçerdeki gibi ölçeklendirme faktörü, sıcaklık modeli, analog-sayısal çevirici hatalarının yanında en önemli etkenlerden biri kayma hızı (“drift rate”) kabul edilir. Açısal hız duyargasının hız ölçümlerinde oluşan hatalar belirli bir süre sonra ölçülen yönelim açılarının belirli miktarda sapmasına yol açmaktadır. Bu yüzden açısal hız duyargasının kayma hızları tespit edilmeli ve hareket hesabına etkisi en aza indirilmelidir.

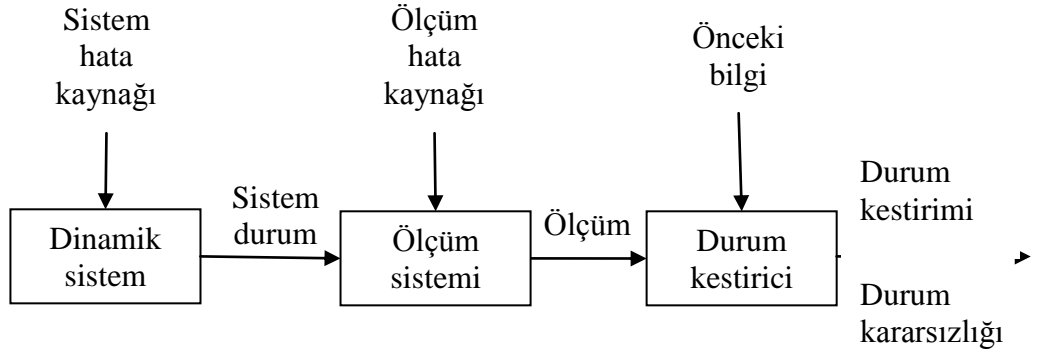
Bir nesnenin hareketini hesaplarken gerek konum, gerekse yönelim hesabında başlangıç durumları kabul edilir ve bu durumların üzerine hesaplanan değerlerin eklenmesi ile son durumlar elde edilir. Hesaplamalarda meydana gelen hatalar zamanla birikmekte ve yanlış sonuçların üretilmesine sebep olmaktadır. Özellikle sabit AÖÜ'lerde yönelim hesabına göre ivmeölçer eksenlerinden üç boyutlu uzaydaki x, y ve z eksenlerine dönüşüm yapıldığı için yönelim hesabında meydana gelecek hatalar ivmeölçer verilerinin değerlendirilmesinde de ek bir hataya sebep olmaktadır. Bu yüzden AÖÜ'lerde kullanılan donanım çok iyi incelenmeli, duyargaların hataları, davranışları çıkartılmalı; gürültü süzme ve kestirim işlemlerinde elde edilen bu veriler kullanılmalıdır. Ayrıca birçok farklı kullanım alanı için özelleşmiş AÖÜ'lerden uygulama hedefine yönelik donanımlar tercih edilmelidir.

3.3 Kestirim ve Takip İşlemleri

Kestirim kavramı, ilgilenilen bir konuda aldatici, hatalı ve kesin olmayan gözlemlerden elde edilen verilerden bir sonuç çıkarma anlamına gelmektedir. Kestirim işleminin amacına örnek olarak bir gezegenin yörüngesini belirlemeye çalışmak ya da bir uçağın konum ve hız bilgilerini hesaplamak verilebilir.

Takip kavramı hareket eden bir nesnenin dış ölçümler ile durumunun kestirilmesidir. Bu işlem sabit noktalara yerleştirilen duyargalarla ya da hareketli platform üzerinde bulunan duyargalarla gerçekleştirilmektedir. İlk bakışta takip, kestirimin özel bir durumu olarak görülse de aslında daha geniş bir kapsamı vardır. Takip, kestirim yaklaşımlarının yanı sıra yaygın istatistiksel karar teorilerini kullanma ihtiyacı duyar.

Kontrol sistemlerinde gürültülü dinamik sistemler için sinyal işleme yaygın bir şekilde kullanılmaktadır. Sisteme gelen sinyallerin gürültülerden süzülmesinin ardından ikinci kademe sistemin durumlarının kestirilmesidir. En uygun kestiricinin tanımı, gözlemleri yani elde edilen ölçümleri belirli bir ölçüte göre optimize edecek şekilde kestiren sayısal algoritma olarak verilebilir. Bu algoritmayı gerçekleştirmek için sistemle ilgili bilinenleri yani dış ölçümleri olabildiğince arttırmak ve sistem durumunu iyi modellemek gerekmektedir. Şekil 3.8'de bir sistemin durum kestiriminin genel bir blok diyagramı verilmiştir.



Şekil 3.8 : Durum Kestirimi Blok Diyagramı

Blok diyagramda da görüldüğü gibi dinamik bir sistemin durumu çıkartıldıktan sonra ölçüm hataları, sistem hataları, ölçümler ve daha önceden kestirilen bilgiler kullanılarak bir sonraki durum kestirilir. Bu işlem sistemin çalışması sürecinde devamlı tekrar eder [17]. Bir nesnenin yöneliminin kestirilmesi bu konuda açıklayıcı bir örnek olacaktır. Sistem durumu açısal hız verilerinin kullanılması ile yönelim hesaplamaya çalışırken ivmeölçer verilerinden elde edilen yönelim bilgileri dış ölçümleri oluşturur. Kısacası sistem yönelimi açısal hız duyargalarına göre hesaplarken ivmeölçer verileri ile düzeltilir.

3.3.1 Kalman Filtresi

Kalman filtresi doğrusal (“linear”) dinamik bir sistemin durumunu kestirmek için kullanılan, en küçük kareler yöntemine göre çalışan, matematiksel denklemlerden oluşan etkin bir filtredir. Filtrenin güçlü yönleri geçmiş, şu anı ve gelecek durumları tahmin edebilmesi ile birlikte kesin olarak sistem çalışma modeli bilinmeyen durumlarda bile etkin olarak çalışabilmesidir.

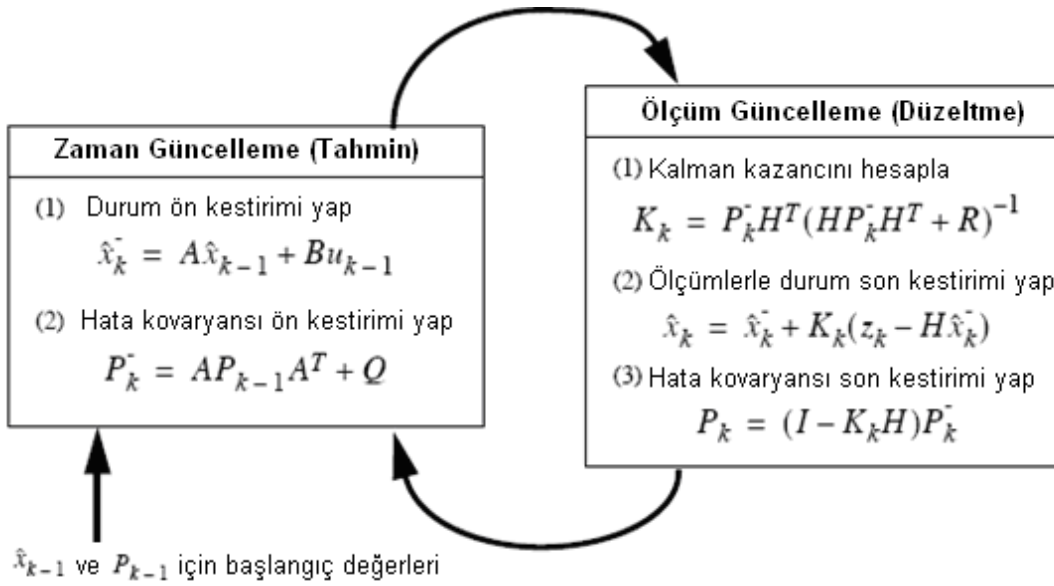
Kalman filtresinde birinci dereceden olasılıksal bir fark denklemi ile kontrol edilen ayrık zamanlı bir prosese ait x durumunun (3.17) z ölçümü (3.18) ile kestirilmeye çalışıldığı varsayalım.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (3.17)$$

$$z_k = Hx_k + v_k \quad (3.18)$$

Bu denklemlerde w_k ve v_k sırasıyla proses ve ölçüm gürültülerini ifade eden rastgele değişkenlerdir; birbirinden bağımsız oldukları, sıfır ortalamaya ve normal dağılıma sahip oldukları kabul edilir. u isteğe bağlı olarak kullanılan kontrol girişini ifade etmektedir.

Kalman filtresi sistem durumlarını geri besleme mekanizması kullanarak kestirir. Filtre ilk olarak geçmiş bilgileri kullanarak yeni durumu kestirir, daha sonra da gürültülü ölçüm değerleri ile geri besleme mekanizmasını çalıştırır. Bu yüzden kalman filtresindeki denklemler zaman güncelleme ve ölçüm güncelleme denklemleri olarak ikiye ayrılır. Zaman güncelleme denklemleri ile şimdiki durum ve hata kovaryansı tahminleri gerçekleştirilir ve bir sonraki adım için ön kestirim yapılmış olur. Ölçüm güncelleme denklemleri ile geri besleme mekanizması çalışır, yeni ölçüm değerleri ön kestirimler ile birleştirilerek iyileştirilmiş son kestirim elde edilir. Kalman filtresinin algoritması Şekil 3.9'de verilmiştir.



Şekil 3.9 : Kalman Filtresi Algoritması

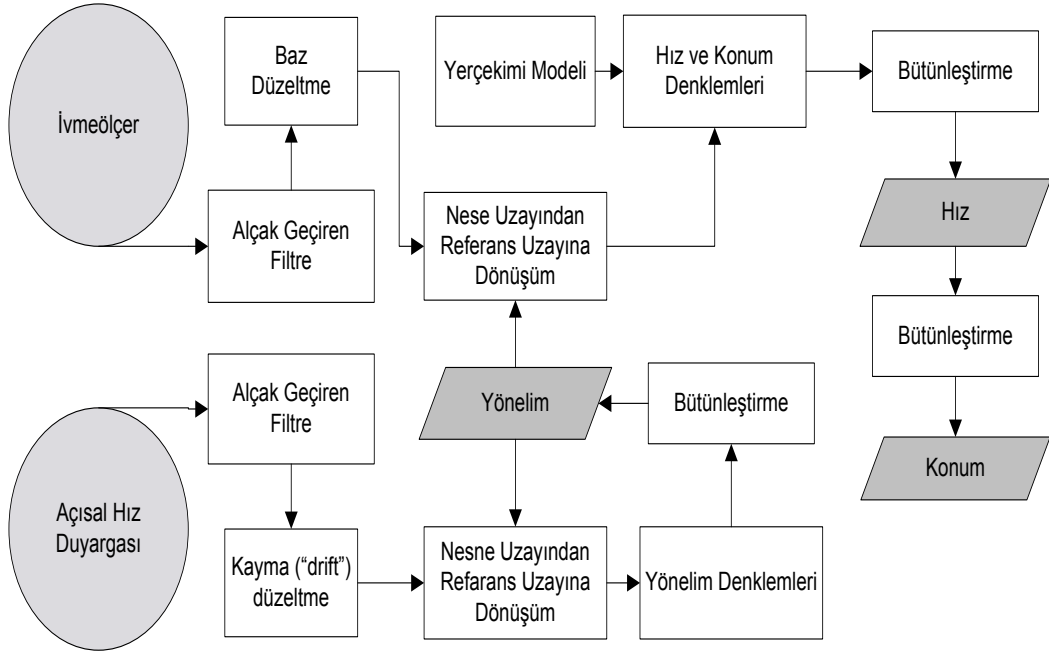
Şekilde de görüldüğü gibi zaman güncelleme denklemlerinin ilkinde sistem durumunun ön kestirimi, ikincisinde ise hata kovaryansının ön kestirimi yapılmaktadır. Ölçüm güncelleme denklemlerinin ilkinde ön kestirimi yapılmış hata kovaryansları ile filtre kazancı hesaplanmakta, ikincisinde ön kestirim ile ölçülen değerler filtre kazancı kullanılarak birleştirilmekte ve son kestirim gerçekleştirilmektedir. Ölçüm güncelleme denklemlerinin sonuncusunda ise filtre kazancı ve ön kestirimi yapılmış hata kovaryansları kullanılarak hata kovaryansının son kestirimi yapılmaktadır. Zaman güncelleme ve ölçüm güncelleme blokları devamlı birbirini takip eder [18].

3.3.2 Genişletilmiş Kalman Filtresi

Kalman filtresi daha önce de belirtildiği gibi doğrusal sistemler üzerinde çalışmaktadır. Eğer sistem doğrusal değil ise kalman filtresi anlık ortalama ve kovaryansı doğrusallaştırır. Bu yaklaşıma genişletilmiş kalman filtresi adı verilir. Taylor serisine benzer şekilde proses ve ölçüm denklemlerinin kısmi türevleri alınarak anlık kestirim doğrusallaştırılabilir. Genişletilmiş kalman filtresinin zayıf noktası kısmi türev alımı esnasında kullanılan Jacobian matris hesabında meydana gelebilecek olan ıraksamanın sonuçları kötü yönde etkilemesidir [18].

4. ÜÇ BOYUTLU UZAYDA HAREKET KESTİRİMİ

Üç boyutlu uzayda bir nesnenin hareketini üzerindeki bir AÖÜ ile kestirebilmek için bilinmesi gereken temel yaklaşımlar daha önceki bölümlerde açıklandı. İvmeölçer ve açısal hız duyargası verilerinin neler olduğu, hangi sonuçları elde etmek için nasıl kullanıldığı ve AÖÜ sisteminin hatalarının etkileri hakkında bilgiler verildi. Problemi kalman filtresi ile durum tasarımı ve kestirimi yöntemiyle ele almadan önce AÖÜ ile hareket kestirime genel bir yaklaşım verilecektir. Bu tüm resmin görülmesi ve problemin ana hatlarıyla özümsemesi için önemlidir ve Şekil 4.1'deki blok diyagramda gösterilmektedir. Blok diyagramda daireler sistem girdilerini, paralel kenarlar sistem çıktılarını ifade etmektedir.



Şekil 4.1 : Sabit AÖÜ ile Hareket Kestirimi

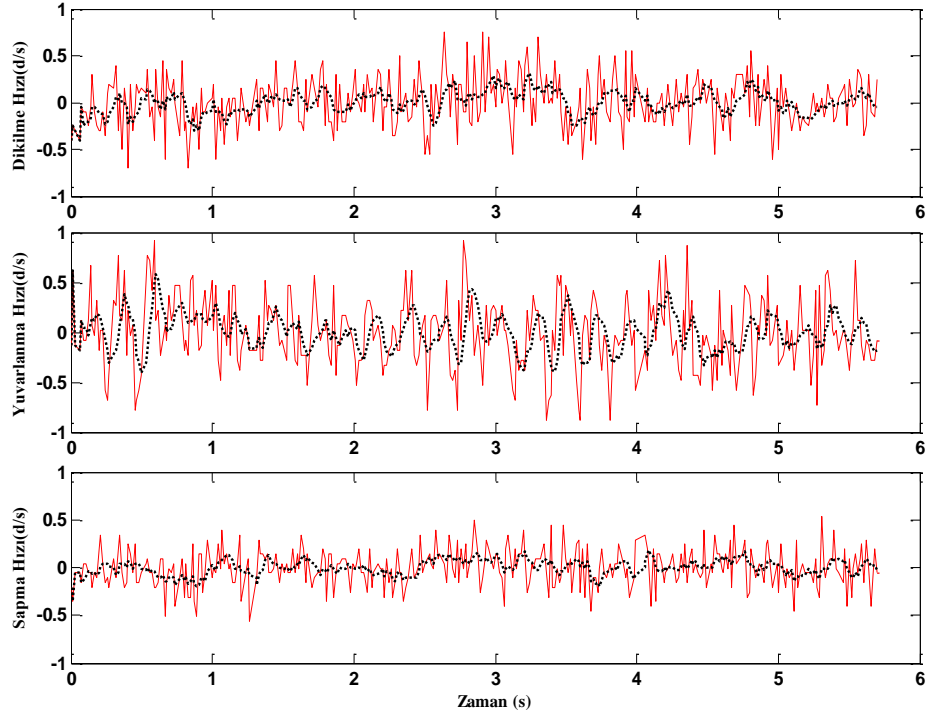
Ana hatları ile blok diyagramın üzerinden geçilecek olursa ilk olarak yapılması gereken yönelim kestirimidir. Açısal hız duyargası verisinin gürültüsü süzülmesi, daha sonra kayması düzeltilmeli ve elde edilen hız verisi bir önceki yönelim ile bütünleştirilerek şimdiki yönelim elde edilmelidir. İvmeölçer verisi filtrelenip, biasının düzeltilmesinin ardından hesaplanan yönelim bilgisi kullanılarak referans uzayı eksenlerine dönüşüm yapılması, yer çekimi etkisinin çıkartılması ve elde edilen

ivme deęerinin bir önceki hız ile bütünleştirilmesi ile şimdiki hız, şimdiki hızın bir önceki konumla bütünleştirilmesi ile şimdiki konum elde edilir.

Tez kapsamında gerçekleştirilen yöntem ilk olarak duyarga verilerinin alçak geçiren filtre ile süzülerek gürültülerinin azaltılması ile başlamaktadır. Ardından ilk olarak yönelim, daha sonra da konum kestirimi Kalman Filtresi kullanılarak gerçekleştirilmektedir. Kalman Filtresini kullanabilmek için sistemin durum deęişkenlerinin çıkartılması, fiziksel modele göre sistem denklemlerinin yazılması, ölçüm deęişkenlerinin belirlenmesi, proses ve ölçüm hatalarının modellenmesi gerçekleştirilmiştir. Kullanıcının hareketini bitirmesinin ardından hatalı filtre deęerleri ile hareketin devam etmesini engellemek için ise bir hareket sonlandırma mekanizması geliştirilmiştir.

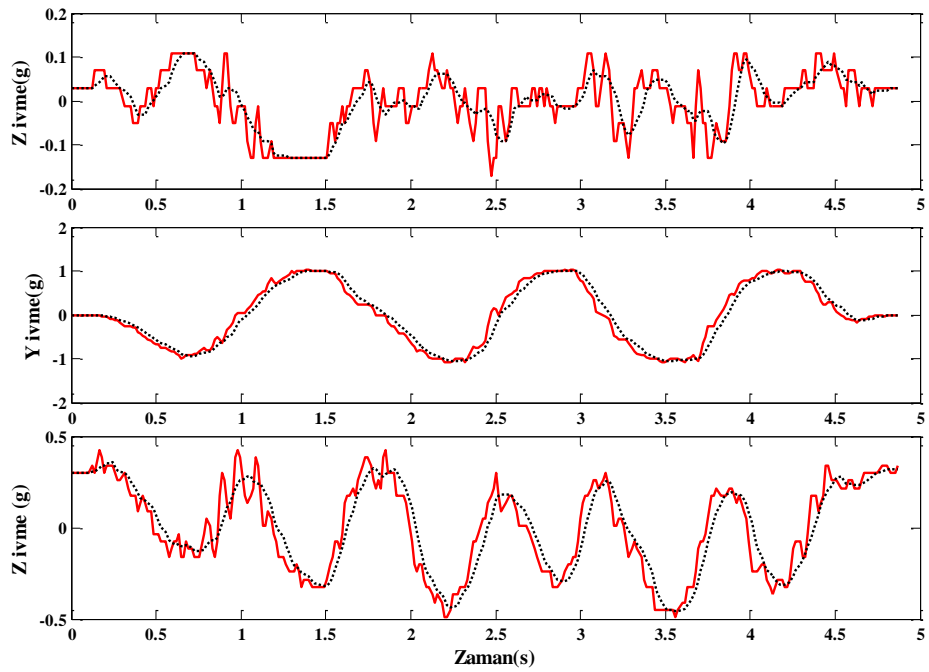
4.1 Kestirim Öncesi Yüksek Frekanstaki Gürültülerin Süzülmesi

Tez kapsamında kullanılan Nintendo Wiimote oyun kumandasının duyarga çıktıları incelendiğinde ilk olarak alçak geçiren filtreler ile sistem gürültüsünün azaltılması hedeflendi. AÖÜ hareketsiz durumda iken açısal hız duyargasının dikilme, yuvarlanma ve sapma hızlarının çıktıları belirlenmiştir. Üç eksenindeki çıktı sıfır ortalamaya (“zero-mean”) sahip olacak şekilde güncellenmiş standart sapma deęerleri dikilme, yuvarlanma ve sapma hızları için hesaplanmıştır. Ortalaması sıfıra çekilmiş fakat üzerinde başka bir işlem yapılmamış açısal hız duyargası çıktıları ile son beş deęerin ortalamasını alan bir alçak geçiren filtrenin sonuçları Şekil 4.2’deki grafikte verilmiştir. Noktalı biçimde çizdirilen sonuçlar ortalama alan filtre sonuçlarıdır. Açısal hız verilerindeki standart sapmaların yaklaşık %50 oranında giderildięi gözlemlenmiştir (Dikilme açısı için 0.2688’den 0.1210’a, yuvarlanma açısı için 0.3445’den 0.1806’ya, sapma açısı için 0.1937’den 0.08’e).



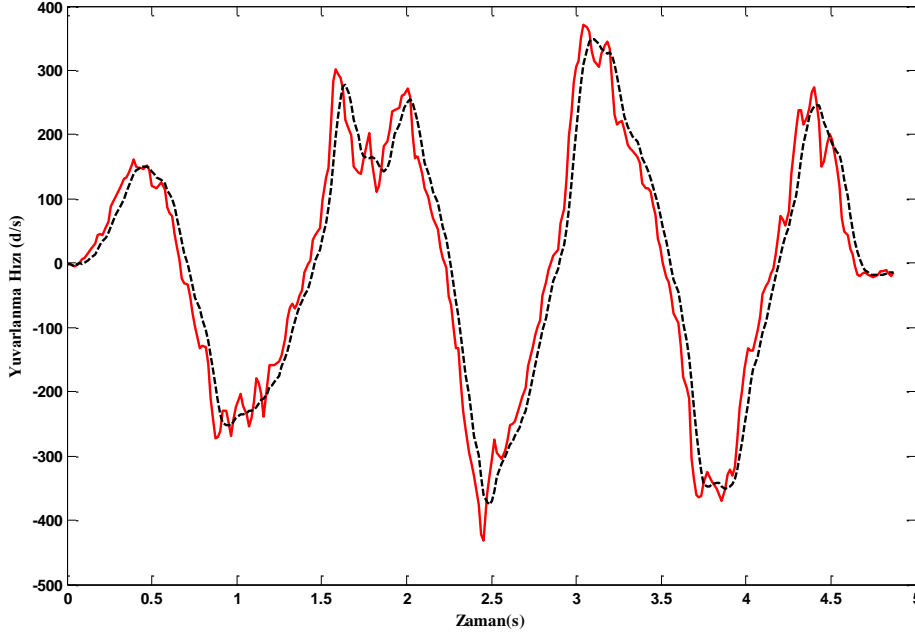
Şekil 4.2 : Hareketsiz Durumda Ortalama Alan Filtre

Filtrenin sonuçlarının daha iyi görülebilmesi için donanıma yuvarlanma açısı değişecek şekilde salınım yaptırılmıştır. Bu hareket esnasında ivmeölçer ve açısal hız duyargası verileri üzerinde elde edilen sonuçlar Şekli 4.3'de ve Şekil 4.4'de verilmiştir. Noktalı olarak çizdirilen sonuçlar ortalama alan filtre sonuçlarıdır.



Şekil 4.3 : İvme Verileri İçin Ortalama Alan Filtre

Hareketsiz durumdaki filtre sonuçlarını gösteren grafiğin dikey eksen aralığı $[-1 +1]^\circ/s$, hareketli durumda ise $[-500 +400]^\circ/s$ 'dir. Daha önce bahsedildiği gibi kullanılan açısal hız duyargası $2000^\circ/s$ 'e kadar çıktı üretebilmektedir.



Şekil 4.4 : Açısal Hız İçin Ortalama Alan Filtre Performansı

Ortalama alan filtrede dikkat edilmesi gereken nokta veri süzme işlemi sırasında hareketin yitirilmemesidir. Ortalama alınan eleman sayısını arttırmak, veride daha pürüzsüz geçişler sağlamasına rağmen Şekil 4.4'deki grafikte görülen tepelerdeki gibi ani değişimlerde işaretin kaybedilmesine sebep olmaktadır.

Kalman filtresine duyarga hatalarını gidermeden önce temel hedef, hareketi kaybetmeden yüksek frekanstaki gürültüleri olabildiğince saf dışı bırakmak ve duyarga gürültülerinin ortalamasını sıfıra çekmektir.

4.2 Yönelim Kestirimi

Nesne yönelim bilgilerinden dikilme ve yuvarlanma açısının ivmeölçer duyargası kullanılarak hesaplanması nesne dinamik ivmeye sahip değil iken mümkündür. Kalman filtresi tasarımında sistemin açısal hız duyargası verilerine göre modellenmesi ve geçerli ivmeölçer verilerine göre düzeltilmesi hedeflenmiştir. Bu yaklaşım yönelim kestirimi algoritmasının en önemli noktalarından birisidir.

Filtre tasarımına geçilecek olursa Kalman filtresi için sistem durum denklemlerinin çıkartılması gerekmektedir (3.1) (3.2). Yönelim ifade yönteminin seçimi, yöntemin etkinliği ve filtre tasarımına uygunluğu göz önüne alınarak belirlenmiştir. Euler açıları şeklinde tutulan yönelim filtre tasarımına uygun olmasına rağmen açı tabanlı yaklaşımın belirtilen zaafı yüzünden tercih edilmemiştir. Koordinat dönüşüm matrisi şeklinde tutulan yönelim ise filtre tasarımı açısından uygun bulunmamıştır. Yönelim açı tabanlı yaklaşımda üç değişkenle, dördey tabanlı yaklaşımda dört değişkenle tutulurken koordinat dönüşüm matrisi tabanlı yaklaşımda dokuz değişkenle ifade edilmektedir. Bu da sistemi daha karmaşık hale getirmektedir. Dördey tabanlı yöntem hem yöntemin etkinliği hem de filtre tasarımına uygunluğu göz önüne alınarak yönelim kestiriminde kullanılmıştır.

Kalman filtresi tasarımında ilk olarak sistemin durum değişkenlerini tanımlamak gerekmektedir. Sistemin durum değişkenleri yönelimi ifade eden değişkenlerdir, dolayısıyla x sistem durumunu ifade etmek üzere yönelimi ifade eden dördeyin bileşenlerinden oluşmaktadır (4.1).

$$x = \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (4.1)$$

Kalman filtresinin uygulanacağı sistemin tanımı (3.17) göz önünde bulundurulursa kullanılan sistem için kontrol girişi bulunmamaktadır ve dördeyler için ardışıl yönelim ilişkisini belirten denkleme (3.12) w_k proses gürültüsü eklenerek ifade edilir (4.2) .

$$q_{k+1} = T q_k + w_k \quad (4.2)$$

w_k proses gürültüsü $k+1$ anında hesaplanan dördey bileşenlerinde meydana gelebilecek sapmaları ($\omega_s, \omega_x, \omega_y, \omega_z$) ifade eder (4.3).

$$w_k = \begin{bmatrix} \omega_s \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (4.3)$$

Kalman filtresinde kullanılacak proses gürültüsünün kovaryansı aşağıdaki gibidir (4.4).

$$Q = \begin{bmatrix} \sigma^2_{\omega_s} & 0 & 0 & 0 \\ 0 & \sigma^2_{\omega_x} & 0 & 0 \\ 0 & 0 & \sigma^2_{\omega_y} & 0 \\ 0 & 0 & 0 & \sigma^2_{\omega_z} \end{bmatrix} \quad (4.4)$$

Bu değerin belirlenmesinde duyarga ölçümleri ile birlikte pratikteki sonuçlar kullanılmıştır. Açık tabanlı hesaplama yapıldığı düşünüldüğünde hız verilerinin standart sapmalarının 2.1×10^{-4} , 3.2×10^{-4} ve 1.4×10^{-4} rad/s şeklinde oldukları görülmektedir. Hız değerlerinin dördey bileşenlerine yansması dolaylı yoldan bir önceki yönetime göre belirlenmektedir. Pratikteki denemelerde bu değer 10^{-4} ile 10^{-7} arasında alınmıştır. Bu değerin işlevi daha basit ifade edilirse filtre sonuçlarının hesaplanmasında açısal hız duyargası verilerine olan güvenilirliğin arttırıp azaltılmasını sağlamaktadır. Bu değerler sıfır olarak alınırsa proses gürültüsü olmadığı varsayılır ve sonuçlar daha çok açısal hız duyargası verilerine dayanır.

Zaman güncelleme ile ilgili tanımlamalardan sonra ölçüm denklemi (3.18) belirlenmelidir. Sistemde ölçümü yapılabilen değerler ivmeölçer bilgilerinden çıkartılan θ (dikilme) ve ϕ (yuvarlanma) açılarıdır (3.1). Z_k matrisi ölçüm, H matrisi dördeyden euler açılarına dönüşüm matrisi ve v_k matrisi ölçüm gürültüsü olarak belirlenirse, ölçüm denklemi aşağıdaki gibi ifade edilir (4.5).

$$Z_k = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_k = H \begin{bmatrix} q_s \\ q_x \\ q_y \\ q_z \end{bmatrix} + v_k = Hq_k + v_k \quad (4.5)$$

H matrisi Jacobian matrisidir ve aşağıdaki gibi ifade edilir (4.6).

$$H = \begin{bmatrix} \frac{\partial \phi}{\partial q_s} & \frac{\partial \phi}{\partial q_x} & \frac{\partial \phi}{\partial q_y} & \frac{\partial \phi}{\partial q_z} \\ \frac{\partial \theta}{\partial q_s} & \frac{\partial \theta}{\partial q_x} & \frac{\partial \theta}{\partial q_y} & \frac{\partial \theta}{\partial q_z} \\ \frac{\partial \psi}{\partial q_s} & \frac{\partial \psi}{\partial q_x} & \frac{\partial \psi}{\partial q_y} & \frac{\partial \psi}{\partial q_z} \end{bmatrix} \quad (4.6)$$

H matrisinin çıkartımı matematiksel olarak mümkündür, fakat karmaşık ve uzun olduğu için yalnızca sonuçları verilmiştir (4.7) [15].

$$\begin{bmatrix} \frac{2D_{22}q_x}{D_{22}^2 + D_{12}^2} & \frac{2(D_{22}q_s + D_{12}q_x)}{D_{22}^2 + D_{12}^2} & \frac{2(D_{22}q_z + D_{12}q_y)}{D_{22}^2 + D_{12}^2} & \frac{2D_{22}q_y}{D_{22}^2 + D_{12}^2} \\ \frac{2q_y}{\sqrt{1-D_{02}^2}} & \frac{-2q_z}{\sqrt{1-D_{02}^2}} & \frac{2q_s}{\sqrt{1-D_{02}^2}} & \frac{2q_x}{\sqrt{1-D_{02}^2}} \\ \frac{2R_{00}q_z}{D_{00}^2 + D_{01}^2} & \frac{2R_{00}q_y}{D_{00}^2 + D_{01}^2} & \frac{2(D_{00}q_x + D_{01}q_y)}{D_{00}^2 + D_{01}^2} & \frac{2(D_{00}q_s + D_{01}q_z)}{D_{00}^2 + D_{01}^2} \end{bmatrix} \quad (4.7)$$

Bu denklemde v_k ölçüm gürültüsü k anında ölçümü yapılan (θ , ϕ açıları için ivmeölçerden elde edilen) açıların hesaplanması sırasında oluşabilecek hatayı (v_ϕ, v_θ, v_ψ) ifade eder (4.8).

$$v_k = \begin{bmatrix} v_\phi \\ v_\theta \\ v_\psi \end{bmatrix} \quad (4.8)$$

Kalman filtresinde kullanılacak ölçüm gürültüsünün kovaryansı da aşağıdaki gibidir (4.9).

$$R = \begin{bmatrix} \sigma_{v\phi}^2 & 0 & 0 \\ 0 & \sigma_{v\theta}^2 & 0 \\ 0 & 0 & \sigma_{v\psi}^2 \end{bmatrix} \quad (4.9)$$

Örneğin ivmeölçerden hesaplanan dikilme açısı ile gerçekteki dikilme açısının arasında 1° hata olabileceği düşünölsün. Algoritmada açılar radyan cinsinden tutulduđu için bu yaklaşık 1.74×10^{-2} radyana karşılık gelmektedir. Bu değerin karesi alındığında 3.04×10^{-4} değeri filtre girdisi olarak verilir. Pratik denemelerde bu değeler 10^{-2} ve 10^{-4} arasında alınmıştır. Pratikteki sonuçlarda dikilme açısında daha fazla hata oluştuđu göröldüğü için dikilme açısı ile ilgili değeri diğer açılara göre daha küçük (10^{-4}) alınmıştır. Böylece ivmeölçer verisinin güvenilirliği artırılarak hataların engellenmesi amaçlanmıştır.

Z_k ölçüm matrisi oluşturulurken ψ açısı ivmeölçerden hesaplanamadığı için bu değeri açısal hız verileri kullanılarak elde edilen değeri olarak alınır. Ölçüm güncelleme işleminde dikkat edilmesi gereken önemli noktalardan birisi de dairesel açı ifadesinin yaratabileceği hatalardır. İvmeölçer verilerine göre yönelim hesaplarırken arcsin ve arctan2 fonksiyonları kullanılmıştır. Bu fonksiyonlardan arcsin $[-90^\circ +90^\circ]$ aralığında arctan2 ise $[-180^\circ 180^\circ]$ aralığında değeri üretmektedir. Örnek olarak açısal hız verilerine göre dördey yöntemi ile kestirilen yönelim ϕ açısına çevrilince (3.14) 180° 'yi gösterebilir. İvmeölçerden hesaplanan ϕ açısı ise -179° 'yi gösterebilir. Bu açıların farkı 1° olmasına rağmen doğrudan filtreye sokulursa 359° derecelik bir fark varmış gibi işlem yapılacak ve kestirim işlemi çökecektir. Bu yüzden ölçüm güncelleme işleminde ön kestirimi yapılan açı değerleri ile ölçülen açı değerleri, birbirleri arasındaki $^\circ$ farkı doğru olacak şekilde yeniden düzenlenmelidir. Az önceki örnek göz önüne alınırsa ivmeölçerden hesaplanan -179° değeri 181° olarak düzeltilmelidir.

İvmeölçer verisi ile elde edilen açıların ölçüm güncelleme işleminde kullanılmasında dikkat edilmesi gereken önemli bir nokta vardır. Daha önce de belirtildiği gibi ivmeölçer verileri nesneye dinamik ivme kazandırıldığında yönelim hesabında hatalı sonuçlar üretir. Bu yüzden ölçüm güncellemeleri kısmında bu sorunu giderecek bir yaklaşım geliştirilmelidir.

Nesnenin dinamik ivmeye sahip olmadığı durum incelenirse aşağıdaki koşullar sağlanmalıdır:

1. Nesneye etki eden toplam ivme $1g$ olmalıdır (4.10).
2. Nesnenin referans uzaydaki z eksenine düşen yerçekimi ivmesi $1g$ olmalıdır (4.11).

Belirtilen koşullar sağlanması için belirli bir eşik değeri tanımlanmalıdır. Belirlenen eşik değeri λ kabul edilirse koşul kontrolü aşağıdaki gibi yapılır (4.10) (4.11).

$$abs(a_x^2 + a_y^2 + a_z^2 - 1) < \lambda \quad (4.10)$$

$$abs(a_{NZ}^2 - 1) < \lambda \quad (4.11)$$

Yukarıdaki formüllerde ivmeölçer verileri G-kuvveti şeklinde değerlendirilmektedir ve eşik değeri λ pratikte 0.2g olarak alınmıştır. Formüllerde verilen koşulların sağlanmadığı durumlarda θ , ϕ açılarının ölçüm güncelleme bloğundaki değerleri için Kalman filtresinin zaman güncelleme bloğunda tahmin edilen dördeyden elde edilen (3.14) açılar kullanılmıştır. Bu işlemle gerçekleştirilen mekanizma dinamik ivmeye sahip olunan durumlarda yalnızca açısal hız duyargası verisini kullanmakta, aksi durumlarda ise açısal hız duyargası verisi ile elde edilen sonuçları ivmeölçer verisi ile düzeltmektedir.

4.3 Doğrusal İvme, Hız ve Konum Kestirimi

Doğrusal ivme, hız ve konum kestiriminden önce ilk olarak yönelim bilgisi kullanılarak nesne uzayından referans uzayına dönüşümün (3.16) yapılması gerekir. Buradaki önemli nokta yönelim hesabında oluşacak hataların dönüşüm yüzünden doğrusal ivme, hız ve konum kestirimine etki edeceğidir [16].

Doğrusal ivme, hız ve konum kestirimi için yapılan işlemin daha kolay anlaşılabilmesi için dinamik sistemin durum denklemlerinin çıkartılması, proses ve ölçüm hatalarının modellenmesi tek eksenle ele alınacaktır. Üç eksenle yaklaşım tek eksenle yaklaşımın üç eksen için de gerçekleştirilmesinden oluşmaktadır.

İlk önce sistemin durum değişkenlerini tanımlamak gerekmektedir. Kullanılan sistemde ivmeölçerlerden elde edilen ivme verisi, bu ivme verisinden elde edilecek hız verisi ve bu hız verisinden elde edilecek konum verisi kestirilmek istenmektedir. Bu yüzden sistemin durumu x konum, hız ve ivme verilerinden oluşmaktadır (4.12).

$$x : \{x_p, x_v, x_a\} \quad x_p : \text{Konum}, \quad x_v : \text{Hız}, \quad x_a : \text{İvme} \quad (4.12)$$

Kalman filtresinin uygulanacağı sistemin tanımı (3.17) göz önünde bulundurulursa kullanılan sistem için kontrol girişi bulunmamaktadır ve ayrık Wiener prosesi ivme modeli sistemin tasarımı için kullanılabilir. (4.13)(4.14)(4.15) [19].

$$x_k = Ax_{k-1} + Lw_{k-1} \quad (4.13)$$

$$A = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

$$L = \begin{bmatrix} \frac{1}{2}T^2 \\ T \\ 1 \end{bmatrix} \quad (4.15)$$

Bu modelde proses gürültüsü (w_k) k . örnekleme zamanında meydana gelebilecek bir ivme artımıdır ve ortalaması sıfır olan bir beyaz gürültü serisi olduğu kabul edilir, ivme ayrık zamanlı bir Wiener procestir. Kalman filtresinde kullanılacak proses gürültüsünün kovaryansı da aşağıdaki gibidir (4.16) [19];

$$Q = L\sigma_w^2 L' = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^3 & T^2 & T \\ \frac{1}{2}T^2 & T & 1 \end{bmatrix} \sigma_w^2 \quad (4.16)$$

Bu model için σ_w duyarga verilerinden elde edilen sapmalar kullanılarak belirlenir. Pratikte sıfır ile meydana gelebilecek en büyük sapmanın yarısı arasında bir değer seçilmelidir [13]. Algoritmada ivmeölçerin ilgili eksenindeki çıktısının standart sapmaları kullanılmıştır ($\sim 10^{-1}$).

Zaman güncelleme ile ilgili tanımlamalardan sonra ölçüm denklemi (3.18) belirlenmelidir. Sistemde ölçülebilen tek değer ivme verisidir, bu yüzden z ölçüm vektörü yalnızca ivmeyi ifade edecektir (4.17) (4.18) (4.19).

$$z_k = Hx_k + Gv_k \quad (4.17)$$

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

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

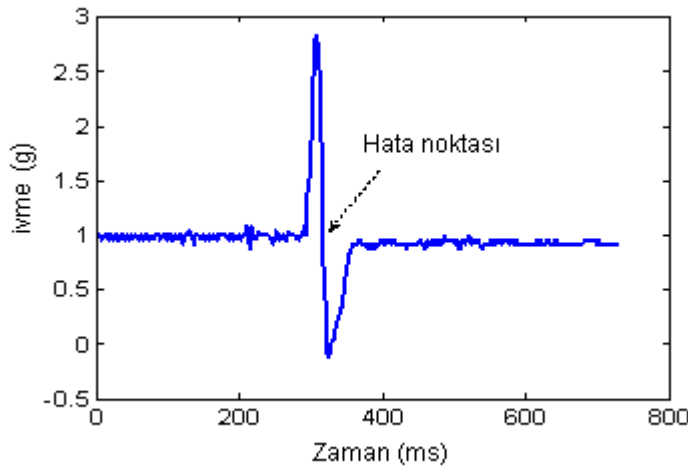
Bu modelde ölçüm gürültüsü (v^k) duyurga verisinin k ölçüm zamanında gerçek veri ile arasındaki meydana gelebilecek farkı ifade eder ve proses gürültüsü gibi ortalaması sıfır olan beyaz gürültü olduğu varsayılır. Kalman filtresinde kullanılacak ölçüm gürültüsünün kovaryansı da aşağıdaki gibidir (4.20);

$$R = G\sigma_v^2G' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \sigma_v^2 \quad (4.20)$$

Kalman filtresini uygulamak için gerekli matris tanımlamaları çıkartıldıktan sonra algoritma koşturulur. Yukarıda belirtilen sistemde proses ve ölçüm gürültülerinin yani σ_w ve σ_v değerlerinin belirlenme şekli önemlidir. Bu yapıda başka bir duyurgadan dış ölçüm elde edilmediği için proses ve ölçüm gürültüsü değerlerinin belirlenmesi ivmeölçer çıktısının istenilen ölçüde yumuşatılması amacına yöneliktir ve pratik deneyler sonucu 10^{-2} olarak belirlenmiştir. Örneğin bu değer 10^{-1} düzeyine yükseltilmesi hareket esnasında ivme sinyali genliğinin yaklaşık yarıya inmesine sebep olmaktadır.

4.4 Hareket Sonlandırma

Hareket sonlandırma işlemi nesnenin kullanım amacından kaynaklanan bir ihtiyaçtır. Üç boyutlu bir uygulamada kullanıcının oyun kumandası ile ani hareketler yaptığı varsayılabilir. Birçok ataletsel seyir sisteminde, sistemin çöküşüne sebep olabilecek düzeydeki agresif hareketler görece daha ucuz ve az doğruluğu olan bir AÖÜ ile gerçekleştirilmektedir. Bu durumda kullanıcı hareketini bitirdikten sonra sistem hataları birikmekte ve özellikle kestirilen doğrusal hız bilgilerinin hareket bitiminde sıfıra dönmemesi sonucu nesnenin hala harekete devam ettiği sanılmaktadır. Bu durumun engellenmesi için ivmeölçer verileri kullanılarak bir hareket sonlandırma yaklaşımı geliştirilmelidir. Yönelim kestirimi işlemlerinin anlatıldığı bölümde nesnenin dinamik ivmeye sahip olmadığı durumun koşulları belirtilmişti. Bu koşullar nesnenin hareketsiz olduğu her durumu karşılamamaktadır. Örnek bir hata durumu Şekil 4.5'deki grafikte verilmiştir. Grafikte z+ eksenini yönünde dikey bir hareket esnasında ivmeölçer çıktısı verilmiştir. Yalnızca anlık verilerin kontrolü yeterli olmamaktadır. Şekildeki hata noktası nesnenin hızının yavaşlamaya başladığı noktadır, belirtilen koşullara uymaktadır fakat bu anda henüz hareket sonlanmamıştır. Bu sorunun giderilmesi için anlık değerlerin yanında geçmişe yönelik tutulan ivmeölçer verilerinin de ortalaması kontrol edilmelidir. Geçmişe yönelik kaç değerın ortalamasına bakılacağı çeşitli hareketlerin incelenmesi ile belirlenir. Bu değer pratikte 20 olarak seçildi. Donanımdan elde edilen veriler ortalama 15 ms'de bir alınmaktadır ve bu değer yaklaşık son 300 ms'nin denetimini ifade etmektedir. Bu denetimle birlikte belirtilen sorun ortadan kalkar.



Şekil 4.5 : Hareket Sonlandırmada Hata Noktası

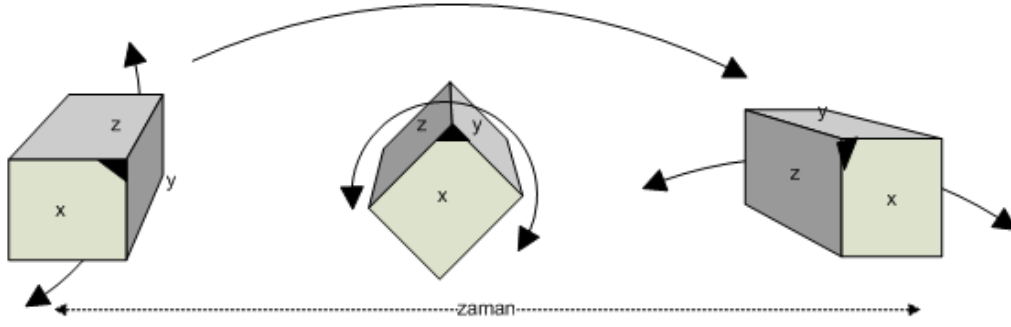
Getirilen bu yaklaşım için bilinmesi gereken önemli bir nokta hareketlerin dinamik ivme değişimleri göz önüne alınarak kestirildiği ve sabit hızda hareket eden bir nesnenin ivmesi olmayacağı için hareket sonlandırma denetimine takılacağıdır.

4.5 Algoritma Sonuçları ve Karşılaştırmalar

Bu kısımda belirli hareketler üzerine yönelim ve konum tahmini algoritmalarının nasıl çalıştığı gösterilecektir. Her hareket için ilk olarak hareketin ne olduğu şekliyle birlikte anlatılacak daha sonra hareketle ilgili algoritma performanslarını gösteren belirli çıktılar verilecektir. Hareketlerin görsel olarak anlatıldığı şekillerde nesne koordinatları ile birlikte x ekseninde üçgen şeklinde bir işaret verilecektir. Bu işaret yönelim değişimleri esnasında nesnenin dönüşünün daha iyi kavranması için eklenmiştir. Algoritma sonuçlarının verildiği grafiklerde yönelim açıları ile ilgili çıktılarda aşağıdakiler bilinmelidir;

- Kırmızı çizgi ivmeölçer verileri ile yapılan yönelim sonuçlarını verir.
- Mavi çizgi açısal hız verileri ile dördey tabanlı yönelim sonuçlarını verir.
- Siyah çizgi açısal hız verileri ile koordinat dönüşüm matrisi tabanlı yönelim sonuçlarını verir.
- Pembe çizgi açısal hız verileri ile euler açıları tabanlı yönelim sonuçlarını verir.
- Yeşil çizgi yönelimi dördey olarak tutan ve ivmeölçer verileri ile düzelten kalman filtresi sonuçlarını verir.
- Yapılan çalışmalarda donanımdan 15ms'de bir veri alınmıştır. Dolayısıyla örnekleme frekansı yaklaşık 67Hz.tir.

Hareket I : Bu hareket Şekil 4.6'da görüldüğü gibi sırası ile dikilme, yuvarlanma ve sapma açılarında yapılan değişimlerden oluşmaktadır. Bu hareketle birlikte yönelim ifade tekniklerinin performansları ve kalman filtresi ile iyileştirilmiş yönelim tahmini sonuçları karşılaştırılacaktır.

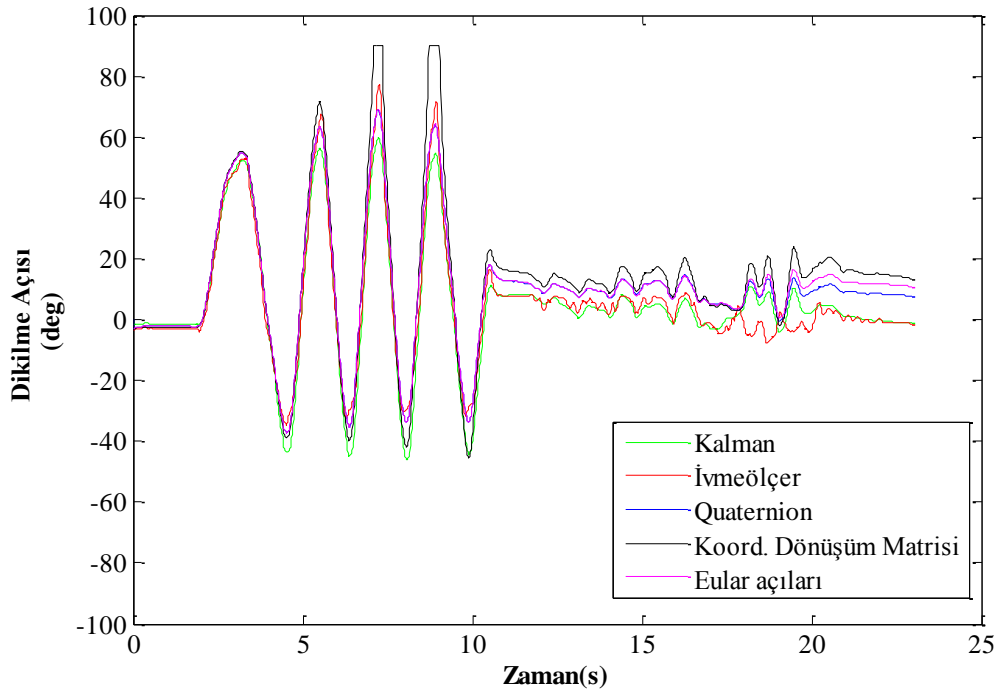


Şekil 4.6 : Hareket I - Dikilme, Yuvarlanma ve Sapma Değişimi

Şekil 4.6’da görüldüğü gibi ilk olarak dikilme açısı değişecek şekilde nesne hareket ettirilmiştir. Şekil 4.7’deki grafikte bu nesnenin hareketi boyunca yapılan yönelim kestirimi işlemlerinden dikilme açısı sonuçlarını göstermektedir. Grafikte yatay eksen saniye olarak zamanı, dikey eksen de derece olarak dikilme açısını göstermektedir. Grafik incelendiğinde;

- Yaklaşık 2. saniyeden başlayarak 11. saniyeye kadar dikilme açısının değiştirildiği görülmektedir. 11. saniyeden sonraki değişimler diğer açılar değiştirilirken istemsiz olarak meydana gelmiştir.
- 12. ve 16. saniyeler arasındaki değişimler yuvarlanma açısı değiştirilirken, 18. ve 22. saniyeler arasındaki değişimler sapma açısı değiştirilirken istemsiz olarak meydana gelmiştir (Şekil 4.8 ve Şekil 4.9’a bakılabilir).
- İvmeölçer sonuçlarının daha gürültülü olduğu görülmektedir.
- Sapma açısı değişimine kadar dördey ve euler açıları yöntemlerin yaklaşık olarak aynı sonuçları ürettiği, sapma açısı değişiminden sonra euler açıları yönteminin daha hatalı sonuçlar ürettiği görülmektedir.
- Koordinat dönüşüm matrisi yöntemi ile yaklaşık 70°’den sonra hatalı verilerin üretildiği görülmektedir. Bu yüzden en hatalı çıktıyı bu yöntem üretmiştir.
- İvmeölçerden elde edilen yönelimin nesnenin dinamik ivmesi olmadığı durumda daha doğru sonuçlar üreteceği varsayılırsa yalnızca açısal hız duyarısını kullanan yöntemlerden en yakın sonuçları dördey yönteminin ürettiği görülmektedir.

- En önemli nokta Kalman filtresi ile kestirilen yönelimin açısal hız duyargasına göre hareket ettiği ve kendisini ivmeölçer sonuçlarına göre düzelttiği görülmektedir. Bu sonuç en iyi 17. ve 24. saniyeler arasında bakılarak görülebilir. Bu aralıktaki hareket esnasındaki sonuçların yalnızca açısal hız kullanan yöntemlere benzer yapıda olduğu görülmekle birlikte hareket sonlanırken ivmeölçer verilerine göre sonuçların düzeltildiği ve diğer yöntemlerde oluşan kayıklığın giderildiği gözlemlenmiştir.

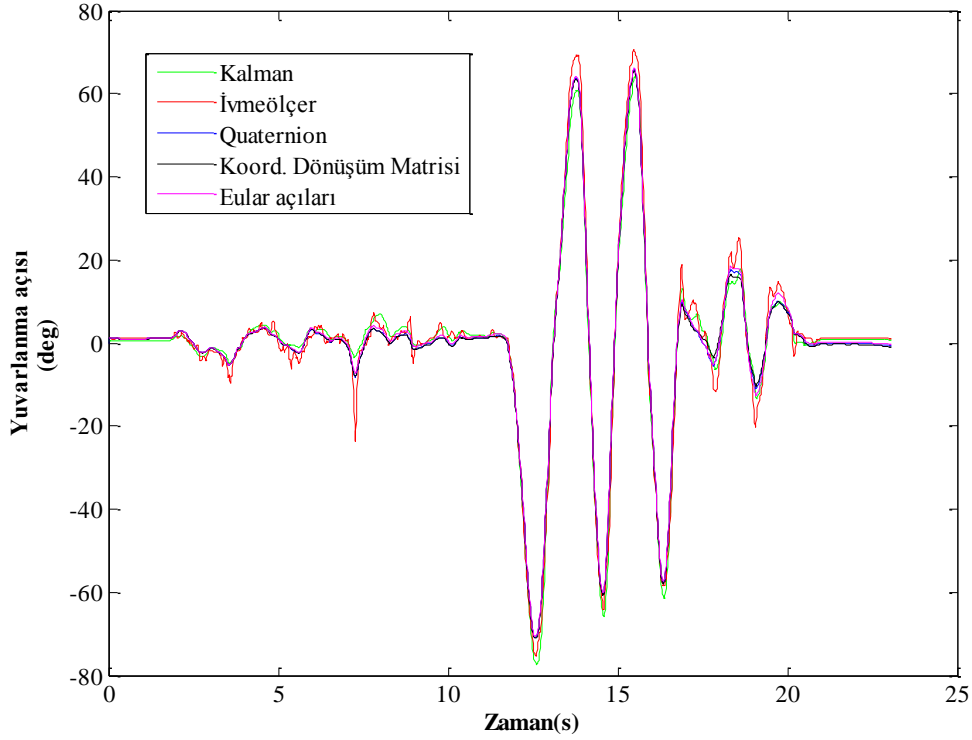


Şekil 4.7 : Hareket I - Dikilme Açısı

Şekil 4.8'deki grafikte nesnenin hareketi boyunca yapılan yönelim kestirimi işlemlerinden yuvarlanma açısı sonuçlarını göstermektedir. Grafikte yatay eksen saniye olarak zamanı, dikey eksen de derece olarak yuvarlanma açısını göstermektedir. Grafik incelendiğinde;

- Yaklaşık 12. saniyeden başlayarak 16. saniyeye kadar dikilme açısının değiştirildiği görülmektedir.
- İvmeölçer verisinin titreşimlerden önemli ölçüde etkilendiği bu grafikte görülmektedir.
- Dördey, euler açıları ve koordinat dönüşüm matrisi yöntemlerinin çok yakın sonuçlar verdiği görülmektedir.

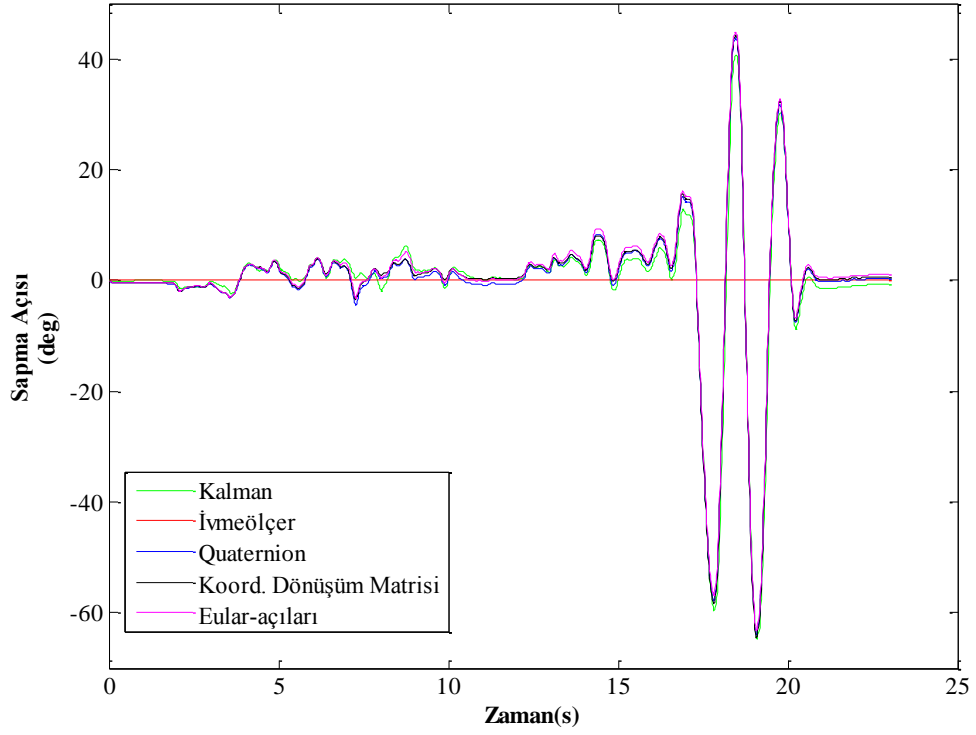
- Kalman filtresi sonucunun hareket bitimine yakın ivmeölçerden elde edilen açıya yakınsadığı ve meydana gelen küçük sapmayı giderdiği görülmektedir. Aynı zamanda bazı bölgelerde gürültülü ivmeölçer sonuçlarının kalman filtresi çıktılarını etkilediği de gözlemlenmektedir.



Şekil 4.8 : Hareket I - Yuvarlanma Açısı

Şekil 4.9'daki grafikte nesnenin hareketi boyunca yapılan yönelim kestirimi işlemlerinden sapma açısı sonuçlarını göstermektedir. Grafikte yatay eksen saniye olarak zamanı, dikey eksen de derece olarak sapma açısını göstermektedir. İvmeölçer verilerinden sapma açısı hesaplanmadığı için ivmeölçer sonuçları sıfır olarak gösterilmiştir. Grafik incelendiğinde;

- Sapma açısı değerinde dış ölçüm kullanılarak doğrulama yapılmamıştır.
- Kalman dışındaki bütün yöntemlerin benzer sonuçlar ürettiği görülmektedir.
- Kalman filtresi sonucunun diğer sonuçlarla küçük bir farklılık gösterdiği görülmektedir. Bunun sebebi kalman filtre içersinde gürültü modellerinin çalışmasıdır. Diğer modellerde sadece ardışıl yönelimlerin ilişki denklemleri kullanılmaktadır.

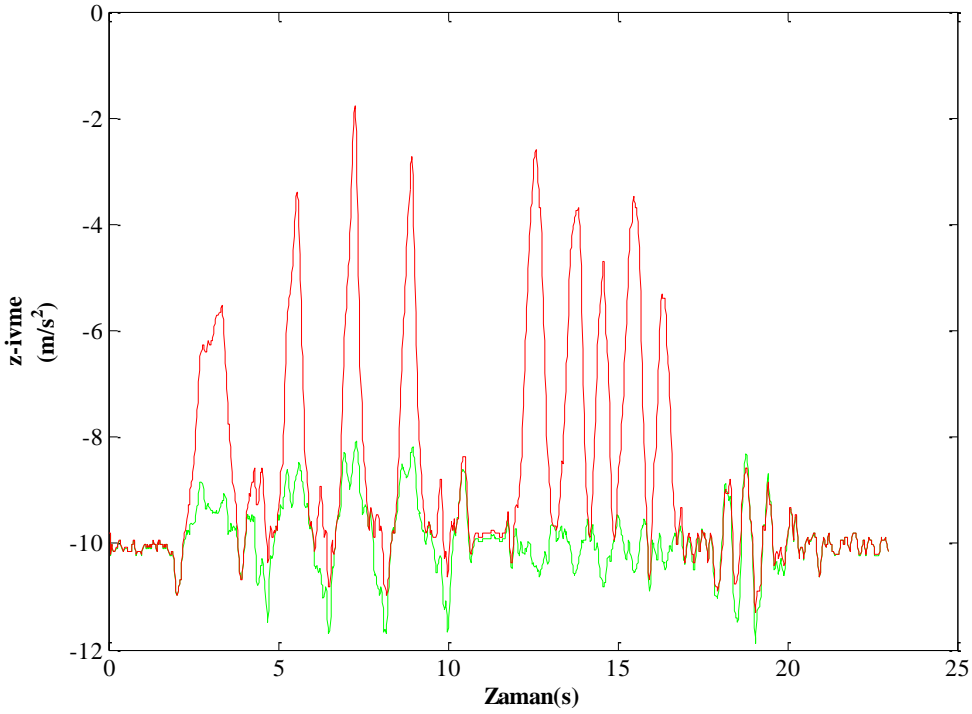


Şekil 4.9 : Hareket I - Sapma Açısı

Şekil 4.10'daki grafikte nesne uzayındaki z eksenine düşen yerçekimi ivmesi kırmızı renkte ve referans uzaydaki z eksenine düşen yerçekimi ivmesi yeşil renkte gösterilmektedir. Grafikte yatay eksen saniye olarak zamanı, dikey eksen de m/s^2 olarak ivmeyi göstermektedir. Grafik incelendiğinde;

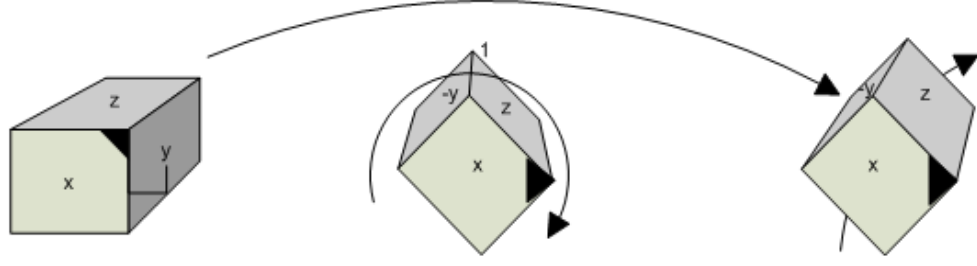
- Dikilme (2. ve 11. saniye arası) ve yuvarlanma (12. ve 16. saniye arası) hareketleri sırasında yerçekimi kuvvetinin nesnenin z eksenine üzerindeki etkisinin değiştiği görülmektedir.
- Sapma hareketi (18. ve 22. saniye arası) sırasında yerçekimi kuvvetinin nesnenin z eksenine üzerindeki etkisinin diğer hareketlere oranla çok az değiştiği görülür. Bunun sebebi sapma hareketinin etkisi değil, sapma hareketi yapılırken diğer eksenlerde yapılan istemsiz yönelim değişimleridir.
- Referans uzayı nesnenin hareket ettiği ortamı tanımladığı için yerçekimi ivmesi nesnenin yönelimi ne olursa olsun referans uzaydaki z eksenine düşmektedir. Bu yüzden koordinat dönüşüm matrisi kullanılarak ivme değerlerinin nesne eksenlerinden referans eksenlerine dönüşümün doğru yapıldığı görülmektedir.

- Yerçekimi ivmesinin yaklaşık 9.8 m/s^2 olduğu düşünülürse yönelim değişimleri esnasında bu değer 8 m/s^2 ve 12 m/s^2 arasında değişim gösterebildiği görülmekte. Bu da yönelim hareketinin ivmeölçer verilerini etkilediği dolayısıyla beklenen değerlerden sapmalar meydana geldiğini gösterir. Nesnenin dinamik ivmeye sahip olmadığı durumlar denetlenirken neden eşik bir değer belirlenmesi gerektiği bu grafikte anlaşılmaktadır. Örneğin eşik değer 2 m/s^2 belirlenirse bütün hareket boyunca ivmeölçer verileri ile doğrulama yapılır, buna karşın eşik değer 0.2 m/s^2 belirlenirse yalnızca yönelimin değişmediği ara noktalarda düzeltme yapılır.



Şekil 4.10 : Hareket I - Z Eksenİ İvmeleri

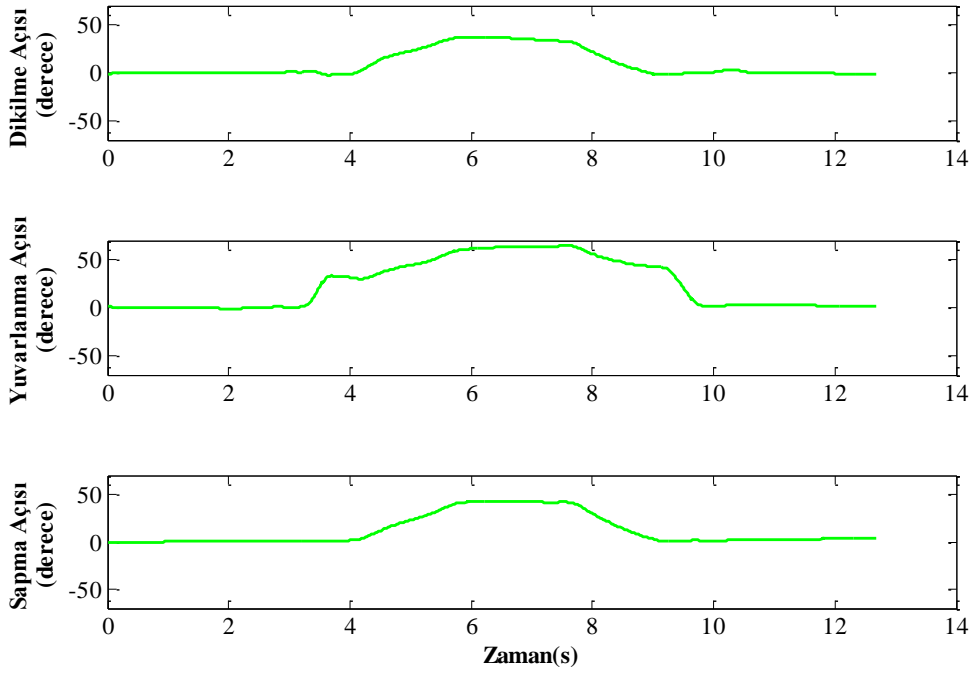
Hareket II : Bu hareketle Şekil 4.11’de gösterildiği gibi nesnenin yuvarlanma açısı yaklaşık 45 derece değiştirildikten sonra nesne koordinatlarında yaklaşık 45 derecelik bir dikilme hareketi yapılmaktadır. Ardından bu hareketin tersi gerçekleştirilerek başlangıç durumuna dönüş yapılmaktadır. Amaç yuvarlanma açısı farklı iken nesne koordinatlarında yapılan bir dikilme hareketinin referans koordinatlarına nasıl yansıdığını görmektir.



Şekil 4.11 : Hareket II – Yuvarlanma ve Dikilme Hareketi

Şekil 4.11’de belirtilen hareketin nesne yönelimini nasıl değiştirdiği Şekil 4.12’deki grafikte gösterilmektedir. Grafikte yatay eksen saniye cinsinden zamanı ve dikey eksenler kalman filtresi sonucu yönelim açılarını ifade etmektedir. Grafik incelendiğinde;

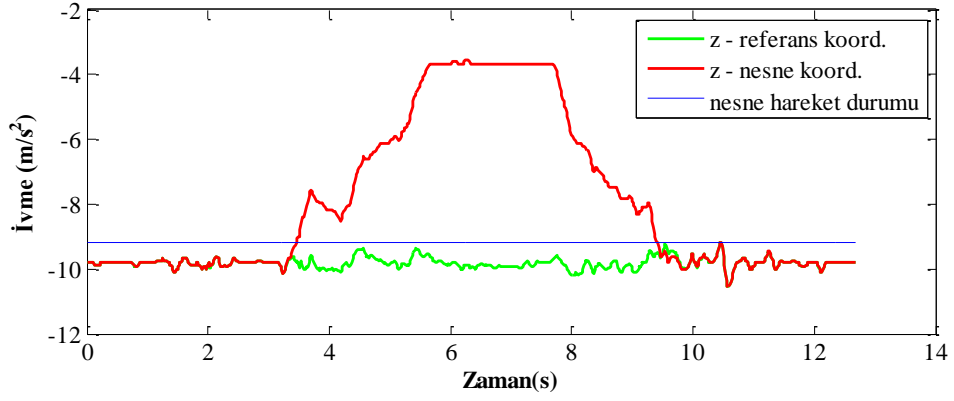
- Nesnenin yönelimi ilk olarak yalnızca yuvarlanma hareketi ile değiştirilmektedir ve bu hareketin yaklaşık olarak 2. ve 3. saniyeler arasında meydana geldiği görülmektedir. Grafikte bu aralıkta dikilme ve sapma açıları değişim göstermezken yuvarlanma açısı yaklaşık 45° düzeyine gelmiştir.
- Yaklaşık 4. saniyede nesne koordinatlarındaki dikilme hareketi başlamaktadır. Bu hareket referans koordinatlarda her üç yönelim açısına da değişime sebep olmaktadır ve 6. saniyede sona ermektedir.
- 8. saniyenin hemen öncesinde aynı hareketin ters yönde başladığı görülmektedir. Yine nesne koordinatlarındaki dikilme hareketi üç eksene de etki eder ve yaklaşık 45° ’lik yuvarlanma açısına sahip yönelime geri dönülür.
- Yaklaşık 9. ve 10. saniyeler arası 2. ve 3. saniyeler arası meydana gelen yönelim değişiminin tersi gerçekleşir ve yuvarlanma açısında değişiklik meydana gelir. Son olarak sistem başlangıç durumuna geri döner.



Şekil 4.12 : Hareket II - Yönelim Kestirimi

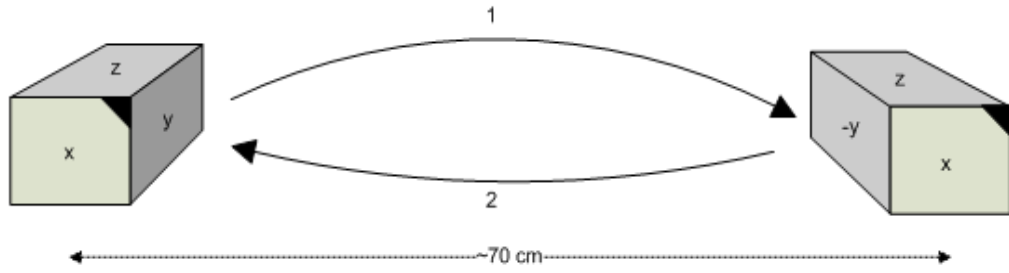
Şekil 4.13'deki grafikte Şekil 4.10'daki grafiktekine benzer şekilde yönelim değişimi boyunca nesne uzayındaki z eksenine ve referans koordinatlarındaki z eksenine düşen yerçekimi ivmesi verilmiştir. Bu grafikte ek olarak mavi çizgi ile nesnenin dinamik ivmeye sahip olup olmadığı belirtilmiştir. Mavi çizgi, denetimin dinamik ivmeye sahip olduğunu gösterdiği durumda yeşil çizginin en alt noktasından, aksi durumda ise en üst noktasından geçecek şekilde çizdirilir. Grafik incelendiğinde;

- Yönelimin değişmesi sürecinde nesne uzayındaki z eksenine düşen yerçekimi ivmesi değeri değişirken referans uzayda aynı kalmıştır.
- Şekil 4.10'daki grafikte karşılaştırıldığında yönelim değişiminin daha yavaş yapıldığı görülmektedir. İvmeölçer verisinde daha az değişim yaşanmış dolayısıyla referans uzaydaki z eksenine düşen yerçekimi ivmesi Şekil 4.10'daki duruma göre daha az sapma göstermiştir.
- Referans uzayındaki z eksenine düşen yerçekimi fazla sapma göstermediği için nesne hareket boyunca dinamik ivmeye sahip değil olarak algılanmış ve mavi çizgi devamlı yeşil çizginin en üst noktasından çizilmiştir.



Şekil 4.13 : Hareket II – Hareket Durumu

Hareket III : Şekil 4.14'de görüldüğü gibi nesnenin yönelimi değiştirilmeden y ekseninde yaklaşık 70 cm + yönde, daha sonra yaklaşık 70 cm – yönde bir hareketi ifade etmektedir.



Şekil 4.14 : Hareket III - Y Ekseninde Sağ ve Sol

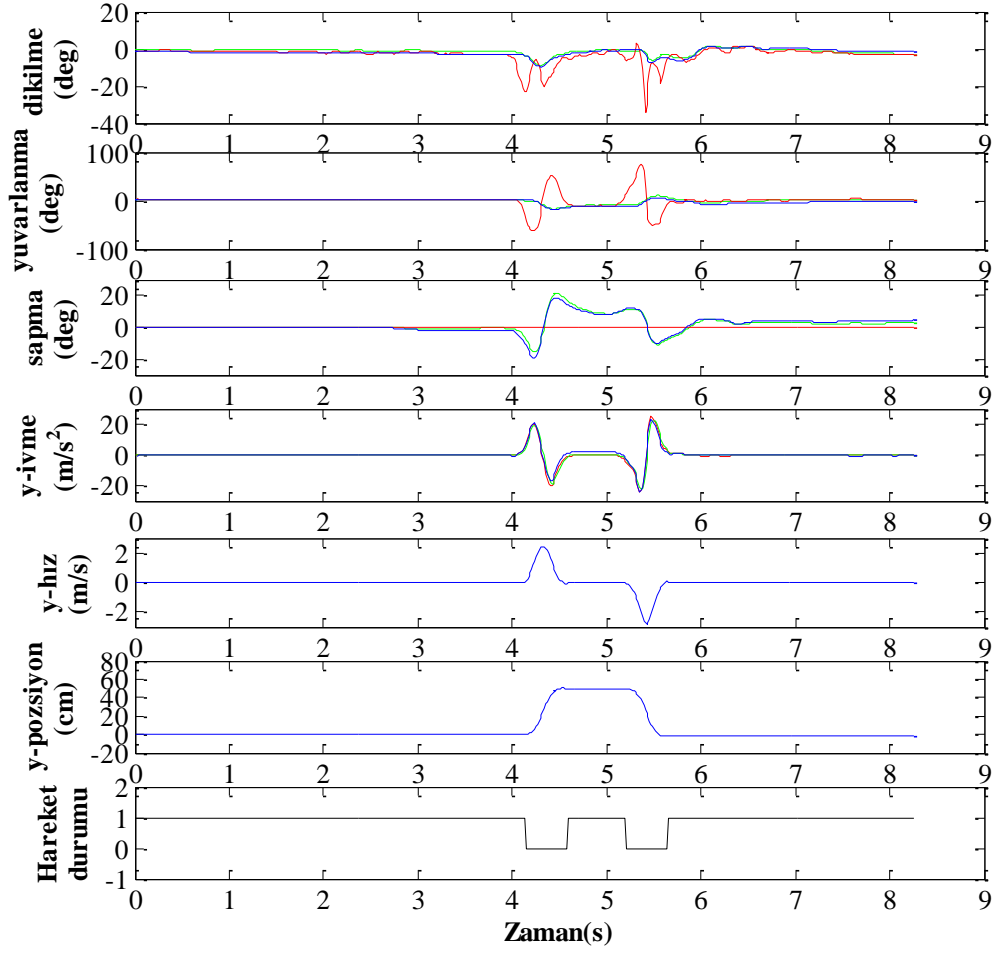
Şekil 4.14'de belirtilen hareket için hareket kestirim algoritmasından elde edilen sonuçlar Şekil 4.15'deki grafikte verilmiştir. Grafik için aşağıdakiler bilinmelidir;

- Üstten başlanarak sırası ile dikilme, yuvarlanma ve sapma açıları, nesne ve referans uzayındaki y eksen ivmeleri, referans uzaydaki y eksen hızı, konumu ve nesne hareket durumu denetimi sonuçları verilmiştir.
- Yönelim açılarının gösteriminde yalnızca dördey, yalnızca ivmeölçer ve kalman filtresi sonuçları gösterilmektedir.
- İvme gösteriminde mavi renk nesne uzayındaki eksen ivmesini, kırmızı renk referans uzay eksen ivmesini, yeşil renk ise kalman filtresi sonucu referans uzay eksen ivmesini gösterir.

- Hareket durumu gösteriminde 0 değeri hareketsiz durumu, 1 değeri hareketli durumu ifade etmektedir.

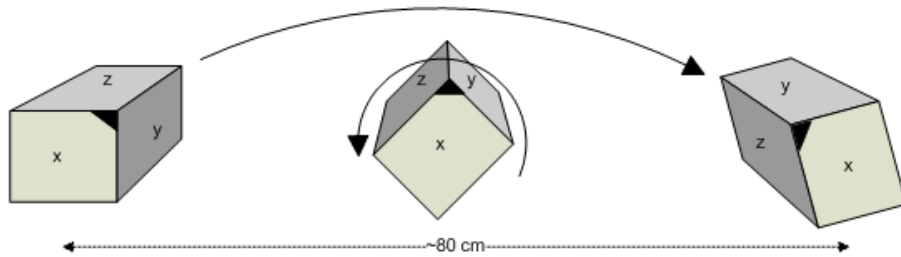
Grafiklerin alt alta verilme sebebi hareket sürecince algoritmanın işleyişini en iyi ifade eden sonuçların aynı zaman ekseninde incelenmesi gerekliliğidir. Grafikler incelendiğinde;

- Hareket durumu denetiminin 0 olarak sonuçlandığı zaman aralıkları ile ivme değişimlerinin gerçekleştiği aralıklar uyuşmakta, bu da denetim mekanizmasının doğru çalıştığını göstermektedir.
- Hareketlerin gerçekleştiği aralıklar için ivmeölçer tarafından hesaplanan dikilme ve yuvarlanma açılarının yanlış olduğu görülmektedir. İvmeölçerden elde edilen bilgilerle yapılan düzeltmeler hareketsiz durumda yapıldığı için bu aralıklardaki yanlış sonuçların dikkate alınmadığı ve kalman filtresi sonuçlarında açısal hız duyargasının etkin olduğu görülmektedir.
- Hareket gerçekleştirilirken sapma açısında istemsiz değişimlerin meydana geldiği görülmektedir.
- y eksenini için ivme verileri incelendiğinde nesne yöneliminde büyük değişiklikler olmadığı için nesne uzayındaki y eksenine üzerine düşen ivme ile referans uzayındaki y eksenine düşen ivmenin yaklaşık olarak aynı olduğu görülmektedir.
- y eksenini için hareket aralıklarında hız değerinin nasıl kestirildiği görülmektedir.
- y eksenini için hareket süresince konum değerinin değişimi görülmektedir. Hareket yaklaşık 70cm olarak yapılmasına rağmen filtre sonucu yaklaşık 60cm olarak algılanmıştır. Filtre değerleri bu hareketin daha hassas algılanması için ayarlanabilir fakat yönelim değişiminin olmadığı bu hareket için özelleştirilen değerler yönelim değişiminin bulunduğu başka hareketlerde daha fazla hatalar üretilmesine sebep olabilir. Bu yüzden çeşitli hareketler için toplamda en az hatayı üreten değerler belirlenmelidir.



Şekil 4.15 : Hareket III - Kestirim Sonuçları

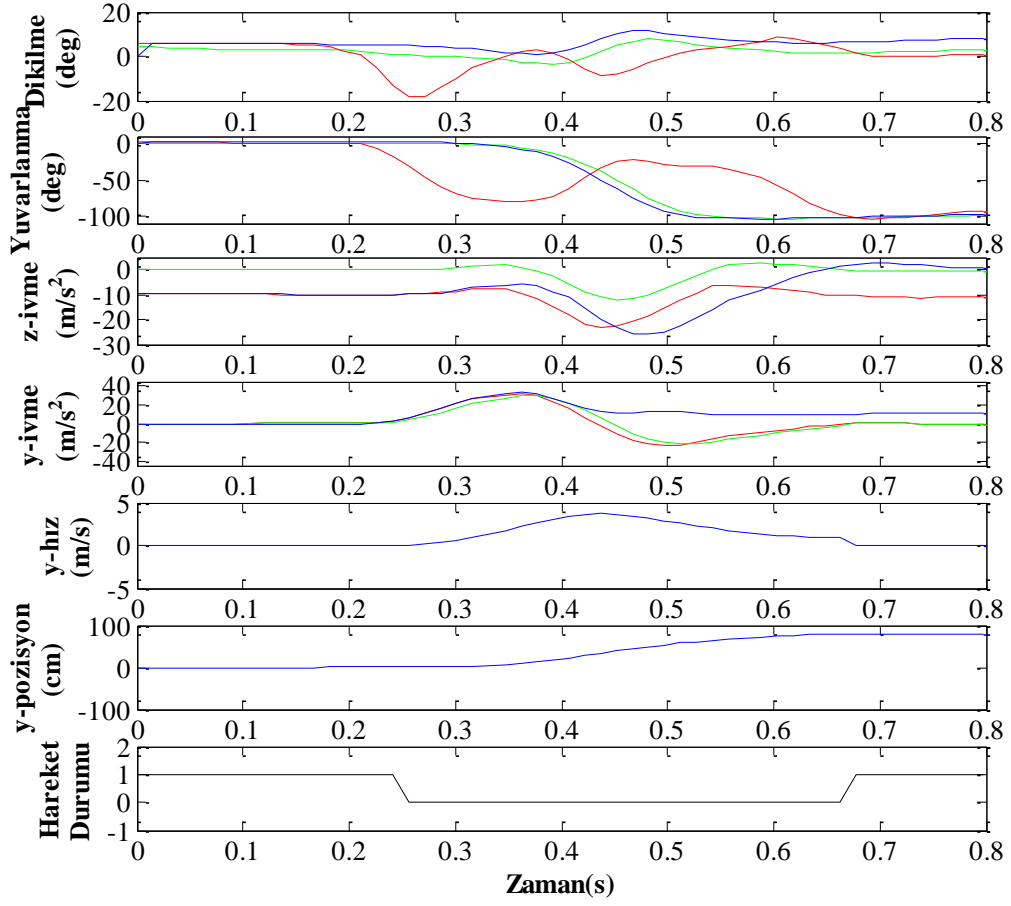
Hareket IV : Şekil 4.16’da gösterildiği gibi nesnenin yuvarlanma açısı yaklaşık 100° değiştirilerek referans koordinattaki y eksenini üzerinde yaklaşık 80 cm’lik bir hareketi ifade etmektedir. Nesnenin yönelimi değişirken yapılan bu hareketle algoritmanın performansı incelenecektir.



Şekil 4.16 : Hareket IV - Yuvarlanarak Konum Değişimi

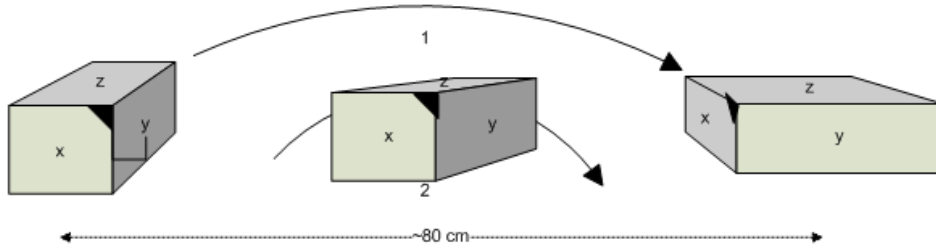
Şekil 4.16'da gösterilen hareket için hareket kestirim algoritmasından elde edilen sonuçlar Şekil 4.17'de verilmiştir. Grafikler incelendiğinde;

- Hareketin yaklaşık olarak [0.25 0.65] zaman aralığında meydana geldiği görülmektedir.
- Hareket III'de yönelim değişmediği için referans uzayı y eksenini ile nesne uzayı y eksenini aynıdır. Bu harekette ise Şekil 4.6'da görülebileceği gibi nesne uzayında y ekseninden z eksenine bir dönüş meydana gelmektedir. Bu değişim grafiklerde incelenirse;
 - Hareketin nesne uzayındaki y-eksende başladığı, [0.25 0.4] zaman aralığında y-eksenindeki ivme değerinin (mavi) baskın olduğu, dönüş hareketi ile birlikte z-ekseni ivme değerine (mavi) etki etmeye başladığı ve [0.4 0.65] aralığında z-ekseninde baskın olduğu görülmektedir. Daha basit olarak ifade edilirse hareketin nesne uzayındaki y ekseninde başladığı ve nesne uzayındaki z ekseninde bittiği görülmektedir.
 - Referans uzaydaki y eksenini ivmesi konum değişiminde kullanılacak ivmedir, y-ivme grafiğindeki kalman filtresi çıktısına bakılırsa (yeşil); yönelime göre dönüşümün yapıldığı ve referans uzaydaki y ekseninde toplam hareketin ifade edildiği görülmektedir. Nesne uzayındaki y eksenini ivmesi (mavi) ile referans uzaydaki y eksenini ivmesi (yeşil) karşılaştırıldığında bu sonuç açıkça görülebilir.
- Dikilme ve yuvarlanma açıları için hareket esnasında ivmeölçerden yanlış değerler üretildiği görülmektedir. Bu değerler hareket aralığında kalman girdisi olarak kullanılmamakta, dolayısıyla yönelim açısal hız duyargası sonuçlarına yakınsamaktadır. Hareket bitiminde ise açı değerlerinin ivmeölçer çıktılarına yakınsadığı görülmektedir.
- Yuvarlanma açısının hareket boyunca 0°'den -100°'e indiği görülmektedir.
- Hareket süresince referans uzayı y eksenini hız ve pozisyon çıktıları görülmektedir. Yaklaşık ~90 cm'lik bir konum değişimi kestirimi yapılmıştır.



Şekil 4.17 : Hareket IV - Kestirim Sonuçları

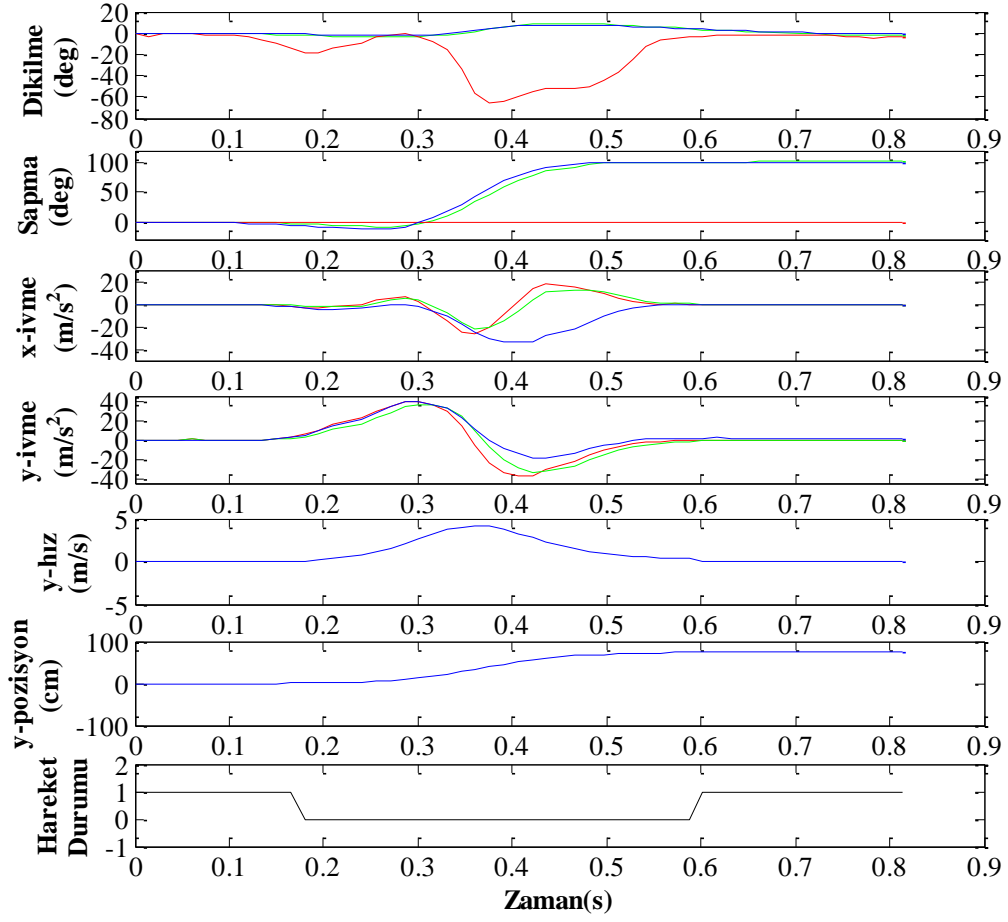
Hareket V : Şekil 4.18’da gösterildiği gibi nesnenin sapma açısı yaklaşık 100° değiştirilerek referans koordinattaki y ekseninde yaklaşık 80 cm’lik bir hareketi ifade etmektedir. Nesnenin yönelimi sapma açısı ile değiştirken yapılan bu hareketle algoritmanın performansı incelenecektir.



Şekil 4.18 : Hareket V - Sapma Yaparak Konum Değişimi

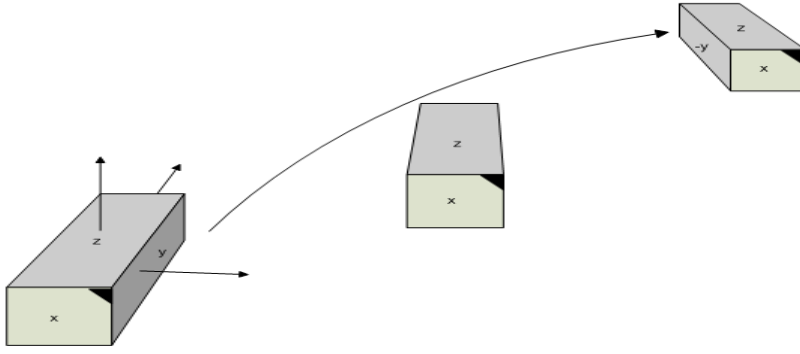
Şekil 4.18’de gösterilen hareket için hareket kestirim algoritmasından elde edilen sonuçlar Şekil 4.19’de verilmiştir. Grafikler incelendiğinde;

- Hareket yine nesne uzayındaki y ekseninde başlamakta fakat bu sefer x ekseninde sonlanmaktadır. Yönetime göre referans uzaydaki y eksenine düşen ivme ile nesne uzayındaki y eksenine düşen ivme farkı y-ivme grafiğinde görülmektedir.
- Nesne uzayındaki x ekseninde ivme görülmesine rağmen ideal olarak referans uzaydaki x ekseninde ivme görülmemesi gerekir. x-ivme grafiğinde özellikle [0.4 0.5] aralığına bakılırsa nesne uzayındaki x ekseninde ivme görülmesine rağmen referans uzaydaki x ekseninde bu ivme görülmemektedir. Buna rağmen ideal sonuca tam yaklaşamadığı görülmüştür. Bu yüzden hareket sonucunda x ekseninde de küçük bir konum değişimi üretilecektir.
- Hareket süresince dikilme açısının yanlış hesaplandığı fakat kalman filtresine yansımadağı görülmüştür.
- Sapma açısının hareket sürecince 0° ’den 100° ’ye çıktığı görülmektedir.
- Hareket süresince referans uzayı y eksenini hız ve pozisyon çıktıları görülmektedir. Yaklaşık 90 cm’lik bir konum değişimi kestirimi yapılmıştır.



Şekil 4.19 : Hareket V - Kestirim sonuçları

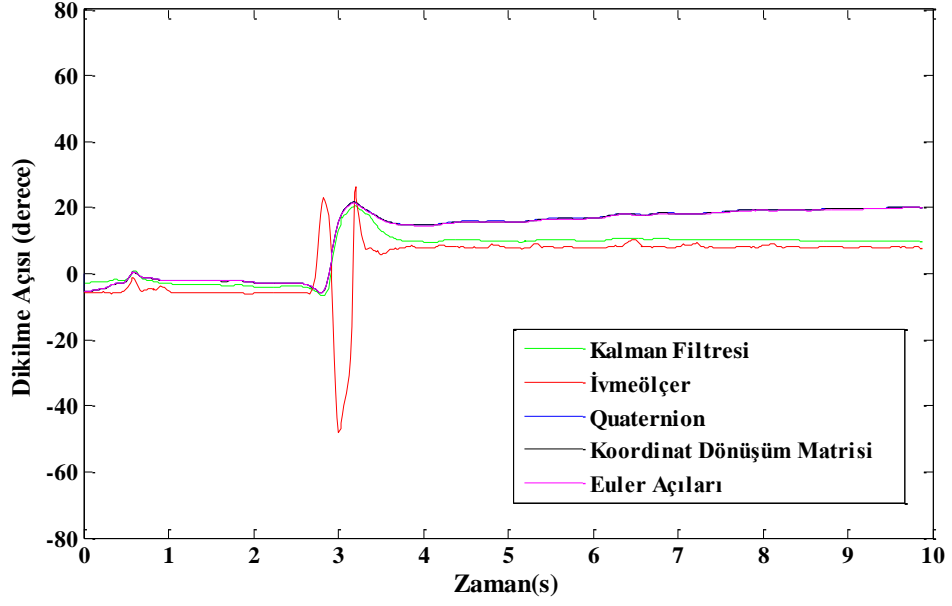
Hareket VI : Şekil 4.20’de görüldüğü gibi nesnenin yönelimi değiştirilmeden referans uzaydaki x, y ve z eksenleri doğrultusunda yapılan bir hareketi ifade etmektedir. Hareket ile yaklaşık olarak x ekseninde 30cm, y ekseninde 70cm ve z ekseninde 70 cm konum değişimi yapılmıştır. Bu hareketin amacı gerek yönelim gerek konum kestirimindeki her bir adımın nasıl çalıştığını incelemektir. Aynı zamanda yönelim ifade teknikleri performansları da karşılaştırılacaktır.



Şekil 4.20 : Hareket VI - Üç Eksen Hareket

Bu hareket için dikilme açısının çıktıları Şekil 4.21'deki grafikte verilmiştir. Grafik incelendiğinde;

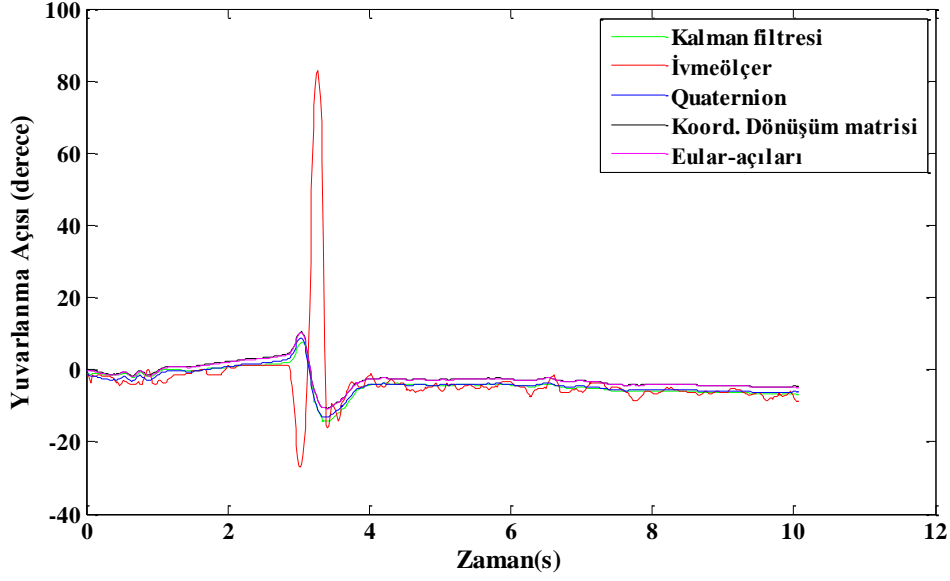
- Konum değişim hareketinin yaklaşık [2.5 3.5] zaman aralığında meydana geldiği görülmektedir.
- Hareket esnasında istemsiz olarak dikilme açısında yaklaşık 10° 'lik bir yönelim yapılmıştır.
- Hareket sırasında ivmeölçer verilerinden elde edilen yönelim sonucunun yanlış olduğu görülmektedir.
- Dördey, koordinat dönüşüm matrisi ve euler açıları yöntemlerinin aynı sonucu ürettikleri görülmektedir. Hareket bitiminde gerçek dikilme açısı ile bu yöntemleri sonuçları arasında yaklaşık 10° 'lik bir fark meydana geldiği görülmektedir.
- Kalman filtresi sonucunun hareketin bitimi ile ivmeölçer sonucuna yakınsadığı ve açısal hız duyargası hatasını düzelttiği görülmektedir.
- Hareket bittikten sonra (4. saniyeden sonra) ivmeölçerin titreşimlerden etkilenmesi yüzünden çıktısında oluşan gürültünün kalman filtresi sonuçlarında bulunmadığı görülmektedir.



Şekil 4.21 : Hareket VI - Dikilme Açısı

Bu hareket için yuvarlanma açısının çıktıkları Şekil 4.22'deki grafikte verilmiştir. Grafik incelendiğinde;

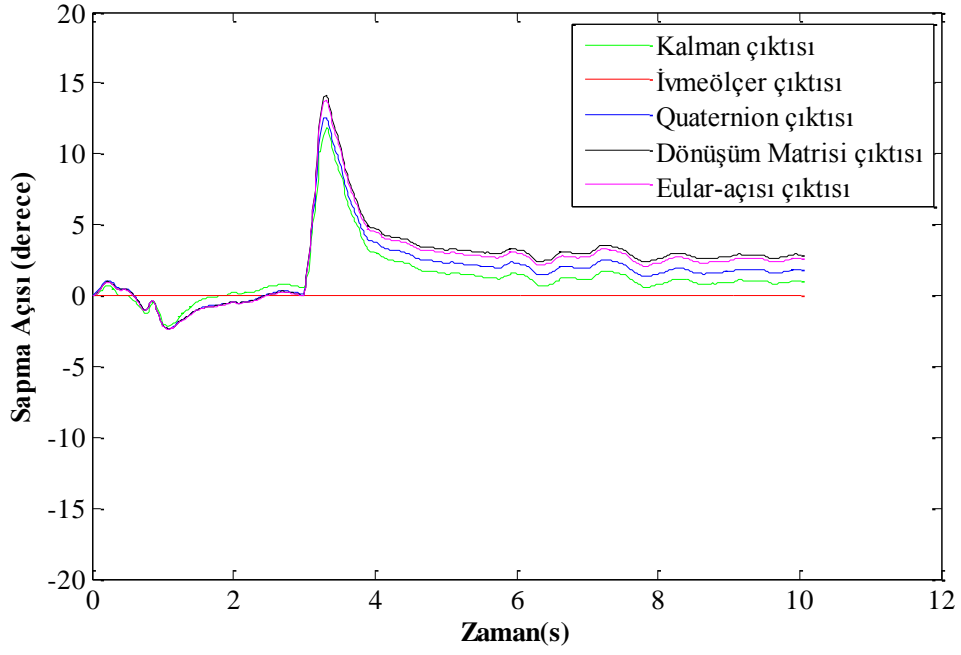
- Hareket esnasında istemsiz olarak dikilme açısında yaklaşık 5° 'lik bir yönelim yapılmıştır.
- Hareket sırasında ivmeölçer verilerinden elde edilen yönelim sonucunun yanlış olduğu görülmektedir.
- Koordinat dönüşüm matrisi ve euler açıları yöntemlerinin aynı sonuçları ürettikleri görülmektedir. İvmeölçerden elde edilen yuvarlanma açısı ile yaklaşık 2° 'lik bir fark bulunmaktadır.
- Dördey yönteminin ivmeölçer ve ivmeölçer ile düzeltilmiş kalman filtresi sonucuna daha yakın olduğu görülmektedir.
- İvmeölçerden çıkartılan yuvarlanma sonuçlarının gürültülü olduğu görülmektedir.



Şekil 4.22 : Hareket VI - Yuvarlanma Açısı

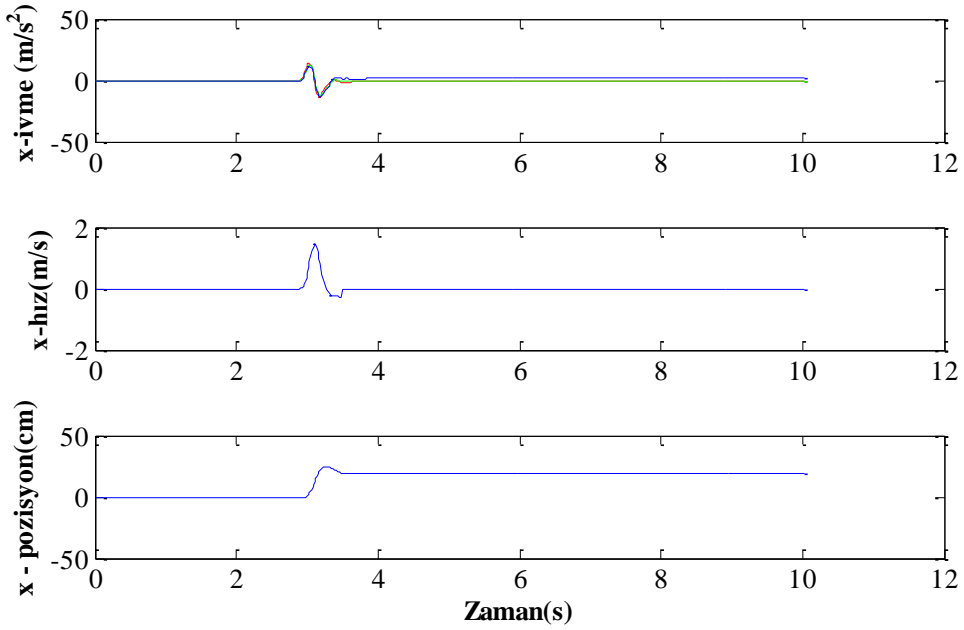
Bu hareket için sapma açısının çıktıları Şekil 4.23'deki grafikte verilmiştir. Grafik incelendiğinde;

- Hareket esnasında istemsiz olarak sapma açısında değişiklik meydana gelmiştir fakat ivmeölçer doğrulaması olmadığı için bu değişikliğin kesin olarak büyüklüğü bilinemez.
- Yönelim ifade teknikleri arasında yaklaşık birer derecelik farklar olduğu görülmektedir.
- Kalman filtresi sonucunun diğer yöntemlere göre daha düşük bir açı değişimini işaret ettiği görülmektedir.

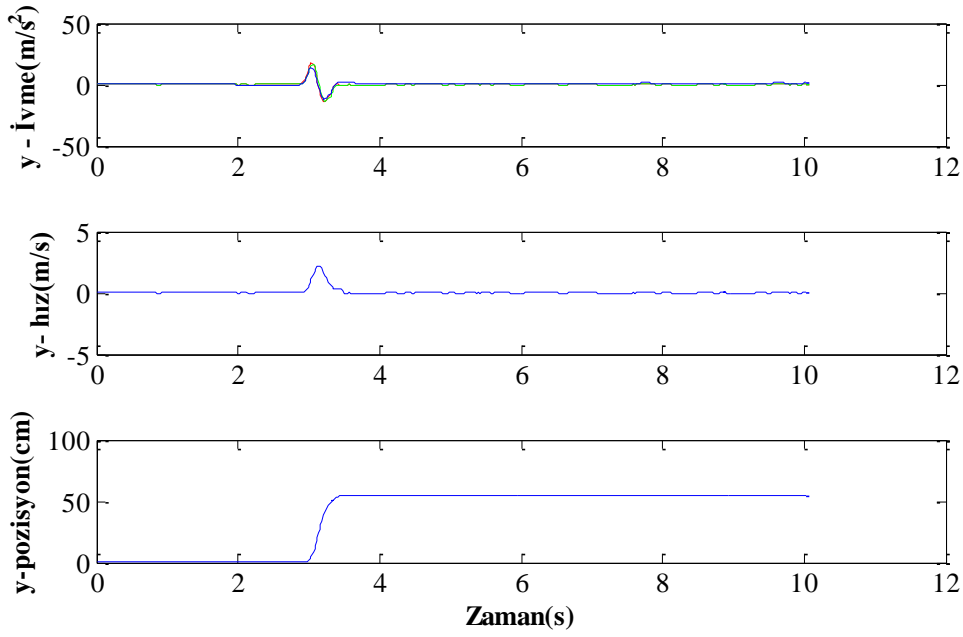


Şekil 4.23 : Hareket VI - Sapma Açısı

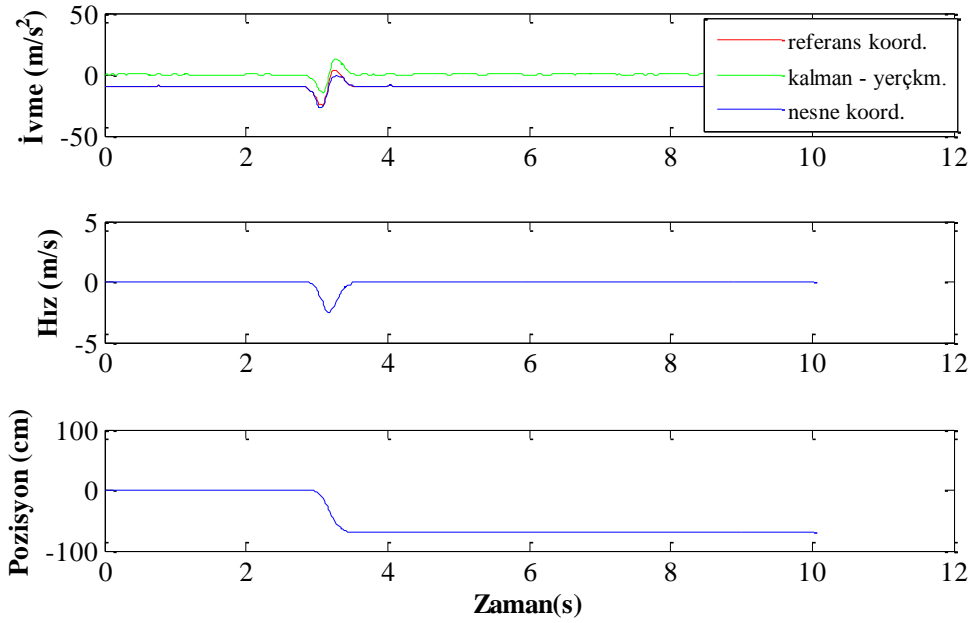
Bu hareket sürecinde her eksen için ivme, hız ve konum kestirimlerinin nasıl yapıldığı Şekil 4.24, Şekil 4.25 ve Şekil 4.26'da verilmiştir. Şekil 4.26'daki ivme grafiği z eksenini ifade ettiği için nesne uzayında yerçekimi etkisi ile birlikte gösterilmiş fakat referans uzayında yerçekimi etkisinden arındırılmış hali gösterilmiştir.



Şekil 4.24 : Hareket VI - Konum kestirimi (x)

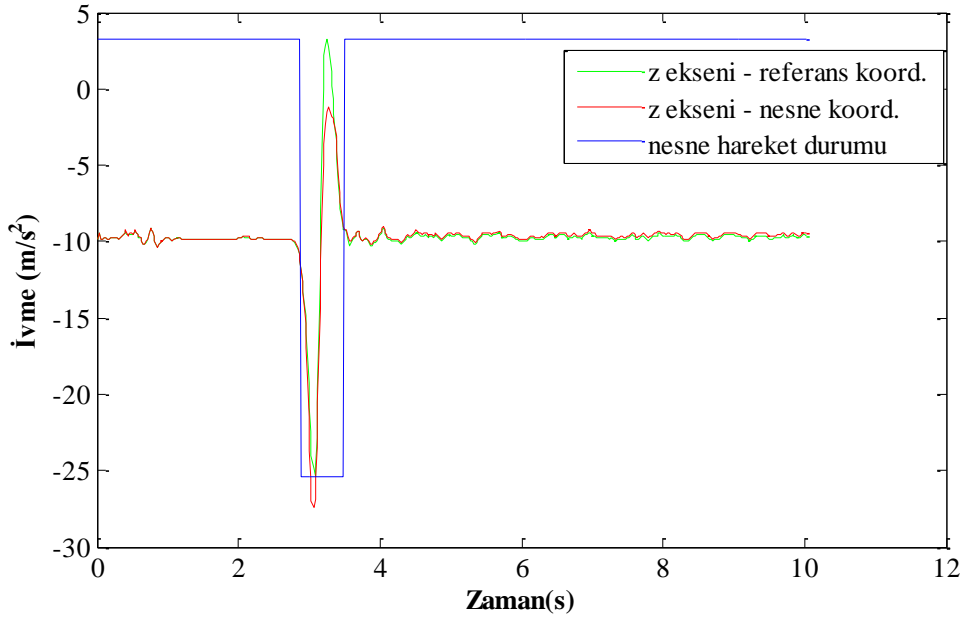


Şekil 4.25 : Hareket VI - Konum kestirimi (y)



Şekil 4.26 : Hareket VI - Konum kestirimi (z)

Şekil 4.27'deki grafikte ise nesnenin hareketlilik durumu denetleme sonuçları nesne uzayındaki ve referans uzaydaki z eksenine etki eden toplam ivme değerleri ile birlikte verilmiştir. Mavi olarak gösterilen çizgi hareket olduğu sonucuna varılırsa yeşil çizginin en alt noktasından aksi takdirde en üst noktasından geçecek şekilde çizdirilmiştir. Hareket durumunun doğru olarak algılandığı görülmektedir.



Şekil 4.27 : Hareket VI - Hareket Durumu İnceleme

Algoritma sonuçlarına alçak geçiren filtre, yönelim ifade teknikleri, kalman filtresi proses ve ölçüm gürültü kovaryansları, hareket sonlandırma işleminde kullanılan değişkenler etki etmektedir. Bu değişkenler teorik bilgiler ve birçok pratik deney sonucu belirlenmiştir. Farklı hareket çeşitleri için toplamda en az hatayı üretecek değerler seçilmeye çalışılmıştır. Buna rağmen aynı algoritmalar için daha etkin sonuçlar üretebilecek değişken kombinasyonlarının bulunabileceği gerçeği göz ardı edilmemelidir.

5. YAZILIM GELİŞTİRME

Tez kapsamındaki yazılım geliştirme çalışmaları başlıca örnek donanımın çıktılarının elde edilebileceği bir yazılım kütüphanesi, hareket kestirim algoritmalarının gerçekleştirileceği bir yazılım kütüphanesi ve bu iki kütüphanenin kullanımıyla yönetilen görsel bir üç boyutlu uygulama yazılımından oluşmaktadır.

İlk aşama belirlenen bir yazılım kütüphanesi ile Wiimote verilerinin elde edilmesinden oluşmaktadır. Bu kısım ilgili donanıma sürücü geliştirmeyi değil; duyarga verilerini elde eden hazır bir yazılım kütüphanesi kullanarak uygulama katmanına adaptör geliştirmeyi ifade etmektedir. Asıl önemli kısım hareket kestirim algoritmalarının gerçekleştirileceği yazılım kütüphanesidir. Algoritma geliştirme ve elde sonuçların değerlendirilmesi açısından görselleştirme önemli bir gereksinim oluşturmaktadır. Tez kapsamında duyarga verilerinin, algoritmalarda kullanılan parametrelerin, eşik değerlerin işaretlere ve sonuçlara etkisinin incelenmesi ve bu sonuçların görselleştirilmesi ihtiyacı doğmuştur. Bu yüzden duyarga verilerinin, tasarlanan filtrelerin ve algoritma sonuçlarının incelenmesinde MATLAB kullanımına başvurulmuştur. Wiimote verilerini okuyabilen çeşitli MATLAB kütüphaneleri bulunmasına rağmen açısız hız duyargası verilerine erişemedikleri gözlenmiştir. Bu yüzden gerekli veriler C++ kütüphanesi kullanılarak kaydedilmiş ve MATLAB ortamında incelenmiştir. Dolayısıyla algoritmik yazılım parçaları algoritmaların MATLAB ortamında kodlanarak incelenmesinin ardından C++'a aktarılması ile gerçekleştirilmiştir. Bir önceki bölümde algoritma sonuçlarının aktarıldığı grafikler MATLAB ortamında geliştirilen model yazılımının çıktılarıdır. Algoritma yazılımları MATLAB ortamından C++ ortamına aktarılırken gelişmiş bir yazılım kütüphanesine dönüştürülmesi hedeflendi. Yazılım geliştirmedeki temel hedefler, hareket kestirim kütüphanesinin kullanılan donanıma doğrudan bağımlı olmayacak şekilde geliştirilmesi ve hareket kestirim kütüphanesindeki bileşenlerin modüler yapıda olması olarak belirlenmiştir. Bu bölümde ilk olarak donanım, hareket kestirimi ve uygulama katmanlarının yazılım mimarisindeki yerleri ve geliştirilen modüllerin yazılımsal özellikleri aktarılacak, ardından hareket kestirim kütüphanesi

sınıf diyagramı ile birlikte açıklanarak modüler yapı anlatılacak ve son olarak uygulama yazılımı tanıtılacaktır.

5.1 Uygulama ve Kütüphane Yönetimi

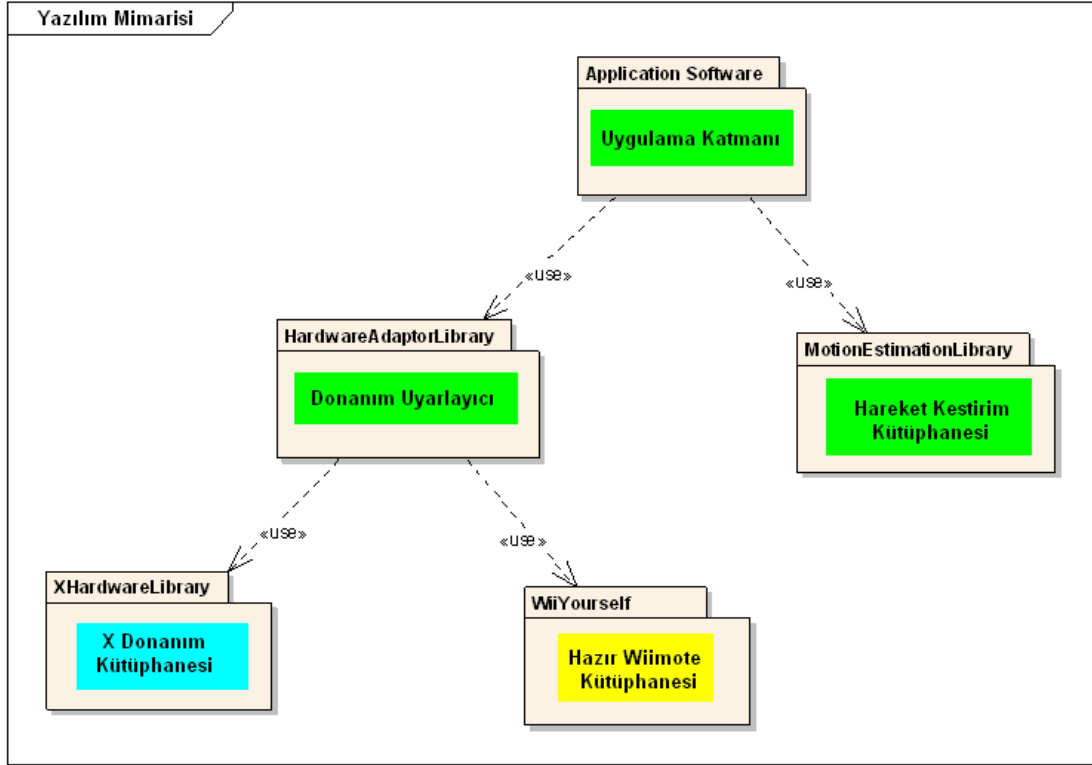
Tez kapsamındaki gerçekleştirilen temel yazılım geliştirme mimarisi Şekil 5.1’de verilmiştir. WiiYourself modülü Wiimote verilerinin elde edilmesinde kullanılan hazır bir C++ kütüphanesidir. HardwareAdaptorLibrary ise WiiYourself kütüphanesini kullanarak duyurga bilgilerini alan ve uygulama katmanına ulaştıran birimdir. Bu birim daha sonra farklı bir X donanımının kullanılması halinde bu donanıma ait XHardwareLibrary modülünü kullanarak benzer bir arayüze sahip adaptör oluşturacak ve uygulama katmanına iletecektir. Şekil 5.1’de yeşil renkle belirtilen modüller tez kapsamında geliştirilmiştir. WiiYourself modülü hazır olduğu için sarı renkle, XhardwareLibrary modülü geliştirilmediği fakat başka bir donanım için geliştirilmesi gerektiği için mavi renkle gösterilmiştir.

WiiYourself kütüphanesi Windows için yazılmış bir Native C++ kütüphanesidir ve WinDDK (“Windows Driver Device Kit”) kullanılarak geliştirilmiştir. Windows WinDDK Microsoft tarafından daha fazla desteklenmemektedir ve yerini WDK (“Windows Driver Kit”) almıştır. WiiYourself kütüphanesi WDK ile çalışır hale getirilmiştir (Microsoft Visual Studio 2008 C++ Project). HardwareAdaptorLibrary modülü geliştirilirken açık kaynaklı Microsoft Visual Studio 2008 ortamına uyarlanabilen QT/C++ kütüphanesinin 4.6.2 sürümü kullanılmıştır.

MotionEstimationLibrary modülü hareket kestiriminin yapıldığı modüldür ve statik bir Native C++ kütüphanesi olarak Visual Studio 2008 ortamında gerçekleştirilmiştir. İhtiyaç duyulan matris işlemleri için açık kaynaklı BOOST C++ kütüphanelerinin 1.42 sürümü kullanılmıştır. Bu modül standart C++ ve BOOST kütüphanelerini kullanmaktadır ve platformdan bağımsızdır. Kaynak kodlar farklı işletim sistemlerindeki farklı derleyiciler ile derlenebilmektedir. Bu modül şekilde de görüldüğü gibi kullanılan donanım ile doğrudan bağlantılı değildir. Farklı uygulamalarda farklı donanımların verileri ile kullanılabilir.

Uygulama katmanı donanım adaptörünü kullanarak elde ettiği donanım verileri ile hareket kestirim kütüphanesini çağırır ve elde ettiği sonuçları 3D grafik arayüzünde

gösterir. 3D görselleştirme için OPENGL Kütüphanesi ve pencere yönetimi için QT/C++ kütüphanesinin 4.6.2 sürümü kullanılmıştır.



Şekil 5.1 : Yazılım Mimarisi

5.2 Hareket Kestirim Kütüphanesi

Hareket kestirim kütüphanesi gerçekleştirilirken temel hedef modüler yapıya sahip, geliştirmeye açık, belirli modüllerin tak-çıkart şeklinde değiştirilip sisteme uyarlanabileceği bir yapı yaratmaktır. Bu amaçla Köprü ("Bridge"), Fabrika ("Factory"), Strateji ("Strategy"), Adaptör ("Adapter") gibi yazılım tasarım kalıplarına başvurulmuştur. Şekil 5.2'de hareket kestirim kütüphanesinin sınıf diyagramı verilmiştir (Ayrıntılı diyagram için EK-A'ya ya bakınız). Aşağıda her bir yazılım bileşeni ve bileşenlerden oluşan modüler yapının nasıl kullanılacağı açıklanacaktır.

KalmanFilter: Soyut("Abstract") bir C++ sınıfı olarak gerçekleştirilmiştir. Herhangi bir sistem modeli kalman filtresi tasarlanarak kestirilmek ve takip edilmek isteniyorsa baştan yeni bir filtre yazılımı yapılmasına gerek yoktur. Bu sınıfta filtrede kullanılan

matrisler tanımlanmış, zaman güncelleme ve ölçüm güncelleme fonksiyonları yazılmış, değişkenlerin başlangıç değerlerine alınmasını ve her iterasyonda sistem modeline ve ölçümlere göre belirlenmesi gereken matrisler için sanal (“virtual”) metotlar tanımlanmıştır. Bu yüzden belirli bir modele göre kalman filtresi oluşturmak için bu sınıftan yeni bir sınıf türetilmeli, soyut sınıflar gerçekleştirilmeli, zaman güncelleme ve ölçüm güncelleme metotları doğru bir şekilde çağrılmalıdır. Türetilmiş sınıfta modele özgü yaklaşımlar gerçekleştirilir, modelden bağımsız temel özellikler ve metotlar kalıtımsal olarak elde edilir.

DTWPBasedPositionEstimator: KalmanFilter soyut sınıfından türetilmiş bir C++ sınıfıdır. KalmanFilter sınıfındaki soyut metotlar tez kapsamında anlatılan konum kestirim modeline göre gerçekleştirilmiştir. Ek olarak bu modele özgü gürültü kovaryanslarını elde etmek için gerekli değerlerin atandığı değişkenlere ve fonksiyonlara sahiptir.

QuaternionBasedOrientationEstimator: KalmanFilter soyut sınıfından türetilmiş bir C++ sınıfıdır. KalmanFilter sınıfındaki soyut metotlar tez kapsamında anlatılan yönelim kestirim modeline göre gerçekleştirilmiştir. Ek olarak bu modele özgü gürültü kovaryanslarını elde etmek için gerekli değerlerin atandığı değişkenlere ve fonksiyonlara sahiptir.

BaseLowPassFilter: Arayüz (“Interface”) sağlayan bir C++ sınıfı olarak gerçekleştirilmiştir. Tüm metotları sanaldır. Gerekli duyarga verilerinin alınacağı fonksiyonun ve sonuçlarının tutulacağı değişkenlerin tanımını içermektedir. Herhangi bir alçak geçiren filtre tasarlanmak istendiğinde bu sınıftan türetilen olacaktır. Duyarga verilerinin alındığı sanal metot gerçekleştirilerek filtre algoritmasının sonuçları ilgili değişkenlere aktarılacaktır. Böylece bu sınıf tipinden alçak geçiren filtre nesnesine sahip bir modül bu sınıftan türetilmiş bütün alçak geçiren filtre sınıflarının nesnelere ile uyumlu çalışabilecektir.

AveragingLPFilter: BaseLowPassFilter sınıfından türetilmiş bir C++ sınıfıdır. BaseLowPassFilter sınıfındaki soyut metotlar gerçekleştirilmiş, ortalama olarak sonuç değişkenlerini güncelleyen bir filtre haline getirilmiştir. Ek olarak filtreye özgü değerlerin tutulacağı listelere, kaç elemanın ortalamasının alınacağını belirttiği değişkenlere ve bu değişkenleri atandığı metoda sahiptir.

BaseOrientationEstimator: Arayüz (Interface) sağlayan bir C++ sınıfı olarak gerçekleştirilmiştir. Tüm metotları sanaldır. Gerekli duyarga verilerinin alınacağı fonksiyonun ve sonuçlarının tutulacağı değişkenlerin tanımını içermektedir. Herhangi bir yönelim hesaplaması yapan algoritma tasarlanmak istendiğinde bu sınıftan türetilen olacaktır. Duyarga verilerinin alındığı sanal metot gerçekleştirilerek filtre algoritmasının sonuçları (dikilme, yönelme ve sapma açıları, koordinat dönüşüm matrisi ve dördey) ilgili değişkenlere aktarılacaktır. Böylece bu sınıf tipinden yönelim hesaplama nesnesine sahip bir modül bu sınıftan türetilmiş bütün yönelim hesaplama algoritmaları içeren nesnelere uyumlu çalışabilecektir.

CemOrientationEstimator: BaseOrientationEstimator sınıfından türetilmiş bir C++ sınıfıdır. BaseOrientationEstimator sınıfındaki soyut metotlar gerçekleştirilmiştir. QuaternionBasedOrientationEstimator sınıfından bir filtreye sahiptir ve bu filtreyi kullanarak yönelim kestirimi ve takibi yapmaktadır. Bu sınıfta ölçüm güncelleme işlemindeki kumandanın dinamik ivmeye sahip olmadığı durumların kontrolleri ve bu kontrollere göre filtreye sağlanacak veriler belirlenmektedir. Elde edilen sonuçlara göre tanımlı değişkenler güncellenmektedir.

BasePositionEstimator: Arayüz (“Interface”) sağlayan bir C++ sınıfı olarak gerçekleştirilmiştir. Tüm metotları sanaldır. Gerekli duyarga verilerinin alınacağı fonksiyonun ve sonuçlarının tutulacağı değişkenlerin tanımını içermektedir. Herhangi bir konum hesaplaması yapan algoritma tasarlanmak istendiğinde bu sınıftan türetilen olacaktır. Duyarga verilerinin alındığı sanal metot gerçekleştirilerek filtre algoritmasının sonuçları (referans uzaydaki X, Y ve Z koordinatları) ilgili değişkenlere aktarılacaktır. Böylece bu sınıf tipinden konum hesaplama nesnesine sahip bir modül bu sınıftan türetilmiş bütün konum hesaplama algoritmaları içeren nesnelere uyumlu çalışabilecektir.

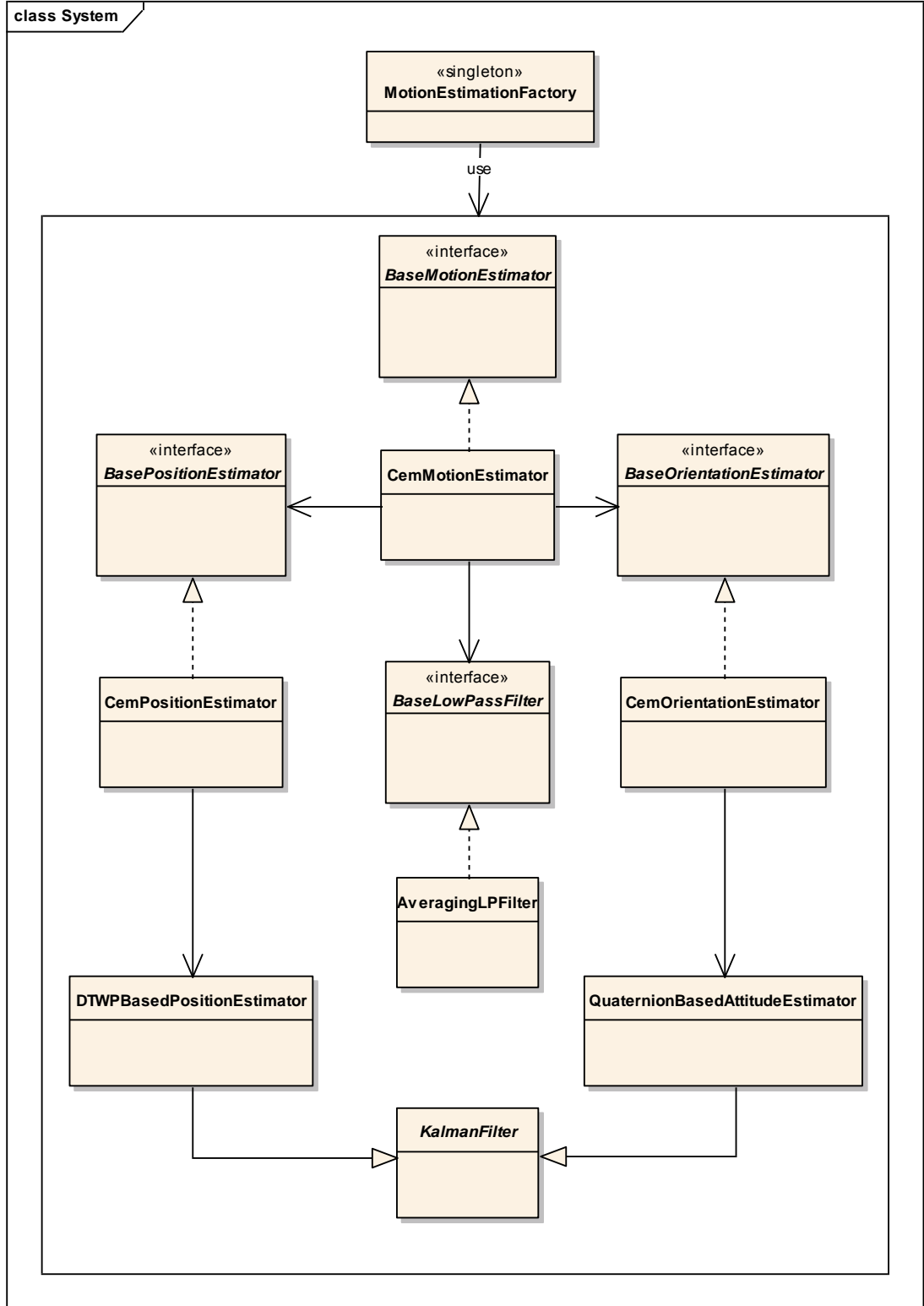
CemPositionEstimator: BasePositionEstimator sınıfından türetilmiş bir C++ sınıfıdır. BasePositionEstimator sınıfındaki soyut metotlar gerçekleştirilmiştir. DTWPBasedPositionEstimator sınıfından bir filtreye sahiptir ve bu filtreyi kullanarak konum kestirimi ve takibi yapmaktadır. Bu sınıfta kumandanın hareket edip etmediği durumların kontrolleri ve bu kontrollere göre filtre değerlerinin yönetilmesi gerçekleştirilmektedir. Elde edilen sonuçlara göre tanımlı değişkenler güncellenmektedir.

BaseMotionEstimator: Arayüz (“Interface”) sağlayan bir C++ sınıfı olarak gerçekleştirilmiştir. Tüm metotları sanaldır. Gerekli duyarga verilerinin alınacağı fonksiyonun ve sonuçlarının tutulacağı yönelim ve konum birlikte olmak üzere tüm değişkenlerin tanımını içermektedir. Herhangi bir yönelim ve konum hesaplaması yapan algoritma tasarlanmak istendiğinde bu sınıftan türetilecektir. Duyarga verilerinin alındığı sanal metot gerçekleştirilerek filtre algoritmasının sonuçları (yönelim ve konum bilgileri) ilgili değişkenlere aktarılacaktır. Böylece bu sınıf tipinden hareket hesaplama nesnesine sahip bir modül bu sınıftan türetilmiş farklı hareket hesaplama algoritmaları içeren nesnelere uyumlu çalışabilecektir.

CemMotionEstimator: BaseMotionEstimator sınıfından türetilmiş bir C++ sınıfıdır. BaseMotionEstimator sınıfındaki soyut metotlar gerçekleştirilmiştir. BaseLowPassFilter sınıfından bir alçak geçiren filtreye, BaseOrientationEstimator sınıfından bir yönelim hesaplama algoritmasına ve BasePositionEstimator sınıfından bir konum hesaplama algoritmasına sahiptir. Bu sınıfın içerdiği nesnelere alt (“base”) sınıf tiplerinden referanslar şeklinde tanımlanmıştır, bu sınıflardan türetilen AveragingLPFilter, CemOrientationEstimator ve CemPositionEstimator tipinden nesnelere referansları atandığında tez kapsamında açıklanan model çalıştırılmış olur. CemMotionEstimator tipinden bir nesne farklı bir alçak geçiren filtre ile çalıştırılmak isteniyorsa yapılması gereken BaseLowPassFilter sınıfından istenen algoritmaya sahip bir sınıf üretmek ve bu sınıftan yaratılan bir nesnenin referansının CemMotionEstimator tipinden yaratılan nesneye atanmasını sağlamaktır. Aynı yaklaşımla yönelim ve konum hesaplama algoritmalarının değiştirilmesi de mümkündür.

MotionEstimationFactory: Tekil (“Singleton”) bir C++ sınıfı olarak gerçekleştirilmiştir. Yukarıdaki modüllerde kalıtsal yaklaşımla modüller bir yapının nasıl elde edildiği açıklandı. Sisteme genel olarak bakıldığında birbirleri ile uyumlu çalışabilen farklı parçaların bir bütün haline getirilebilmesi mümkün hale gelmiştir. Bu sınıf farklı parçaların bir bütün haline getirilmesi işlemini gerçekleştirmekte ve uygulama katmanından soyutlamaktadır. Uygulama katmanı bu sınıfa tek bir çağrı yapmakta ve kullanacağı hareket kestirim nesnesinin referansını istemektedir. Bu sınıf algoritma tasarımcısı tarafından her bir modülün yaratılmasını ve birleştirilmesini sağlar. Tez kapsamında belirtilen yaklaşım düşünüldüğünde CemPositionEstimator, CemOrientationEstimator ve AveragingLPFilter sınıflarından birer nesne yaratır. Ardından CemMotionEstimator sınıfından bir nesne yaratır ve az

önceki nesnelerin referanslarını bu nesneye aktarır. Sınıflar için gerekli ölçüleme değerlerinin girdisini sağlar. Artık gerekli parçalar oluşturulmuş ve CemMotionEstimator sınıfından yaratılan nesnede bütünleştirilmiş durumdadır. Bu nesnenin referansını isteyen module metod çıktısı olarak döndürür.



Şekil 5.2 : Hareket Kestirim Kütüphanesi Sınıf Diyagramı

İvmeölçer ve açısal hız duyargası için ölçüleme verileri doğrudan değer olarak yazılım içersinde atanabileceği gibi aynı zamanda belirlenen biçime göre hazırlanmış bir ölçüleme yapılandırma (“configuration”) dosyası ile de gerçekleştirilebilir. MotionEstimator tipinden bir nesne Şekil 5.3’de içeriği verilen “Calibration.xml” dosyasını okumaya çalışır, okuyamazsa ivmeölçer ve açısal hız duyargası verilerinde hiçbir hata ve sapma olmadığını farz ederek varsayılan değerler kullanır. Aynı zamanda ölçüleme işlemi için başka bir yapılandırma dosyasının kullanılması ilgili dosya isminin nesneye aktarılması ile mümkündür.

```
- <CalibrationParameters>
- <Accelerometer>
- <X>
  <Min>-1.04</Min>
  <Max>1</Max>
  <Std>0.01</Std>
</X>
- <Y>
  <Min>-0.96</Min>
  <Max>1</Max>
  <Std>0.01</Std>
</Y>
- <Z>
  <Min>-1</Min>
  <Max>1</Max>
  <Std>0.01</Std>
</Z>
</Accelerometer>
- <Gyroscope>
- <R>
  <Mean>0.50</Mean>
  <Std>0.1</Std>
</R>
- <P>
  <Mean>-3.33</Mean>
  <Std>0.1</Std>
</P>
- <Y>
  <Mean>-0.05</Mean>
  <Std>0.07</Std>
</Y>
</Gyroscope>
</CalibrationParameters>
```

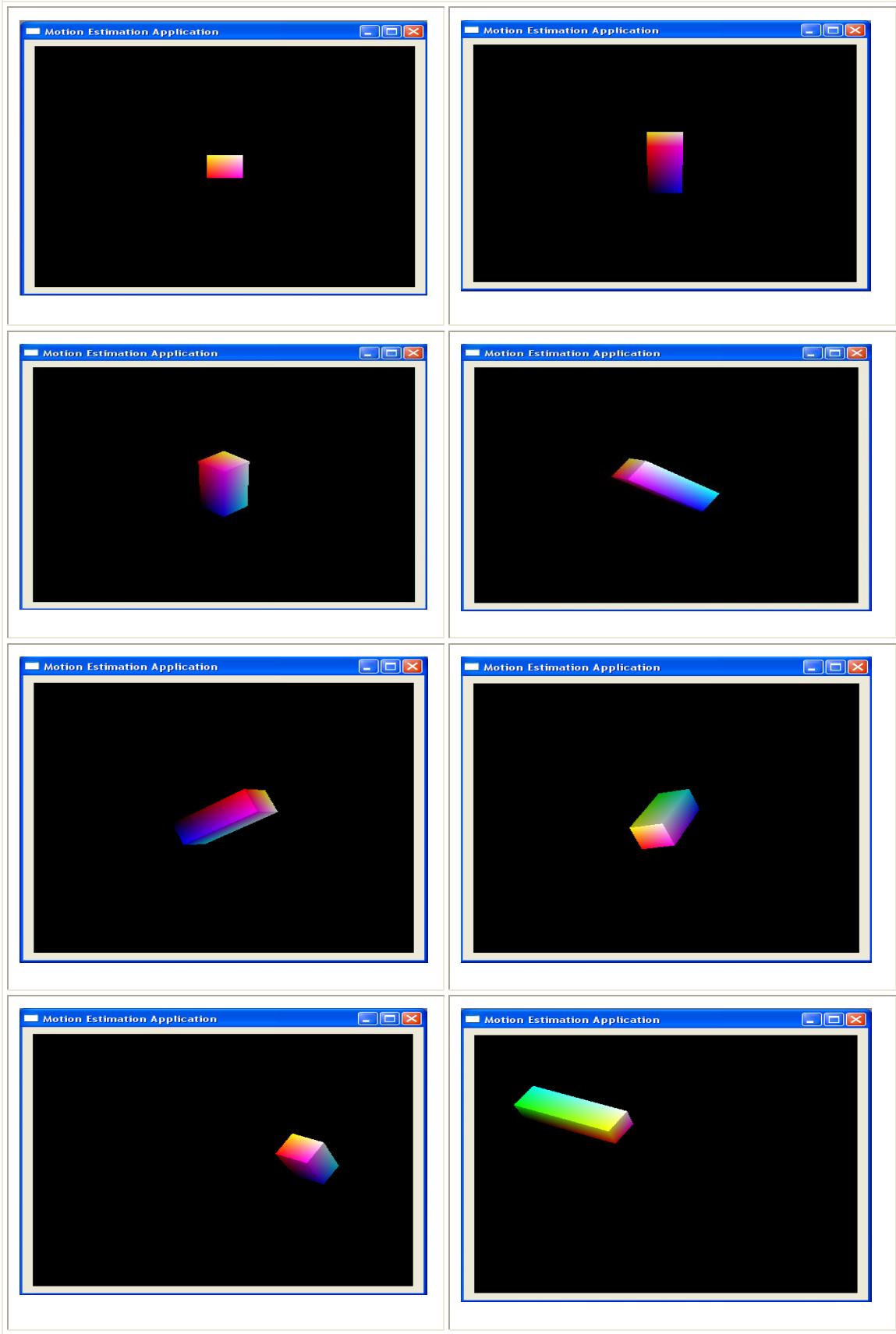
Şekil 5.3 : Ölçüleme Yapılandırma Dosyası İçeriği

Şekil 5.3’de verilen ölçüleme yapılandırma dosyası XML biçiminde tutulmaktadır ve Wiimote oyun kumandası için belirlenen değerleri içerir. İvmeölçer için hareketsiz durumda (yalnız yerçekimi kuvveti etkisinde iken) x, y ve z eksenlerinde donanımın verdiği en küçük ve en büyük değerler ile standart sapmalar (G-kuvveti) verilmektedir. Açısal hız duyargası için hareketsiz durumdaki üç eksen hızların ortalamaları ve standart sapmaları (°/s) verilmektedir. Tez kapsamında bu değerlerin elde edilmesi için SensorCalibraton adında bir uygulama geliştirilmiştir. Bu

uygulama çeşitli durumlarda ölçüm yaparak gerekli değerlerin elde edilmesini sağlamaktadır. Yazılım mimarisi katmanları göz önüne alındığında bu yazılım bağımsız ve bağlantısız durumdadır, bu yüzden konu dışına çıkmamak için ayrıntılarına girilmeyecektir.

5.3 Üç Boyutlu Arayüz Yazılımı

Geliştirilen donanım uyarlayıcı modülün ve hareket kestirim kütüphanesinin kullanıldığı ve sonuçların üç boyutlu olarak görülebileceği basit bir arayüz yazılımı geliştirildi. Tasarlanan genel yazılım mimarisi sayesinde donanımla haberleşme ve hareket kestirim işlemlerinin ayrıntıları uygulama katmanına yansıtılmamaktadır. Uygulama katmanı bir adet donanım adaptörü nesnesi yaratmakta ve bu nesnenin gönderdiği verileri bir metot ile elde etmektedir. Bu metot içerisinde MotionEstimationFactory nesnesinden aldığı hareket kestirici nesneye gerekli girdileri sağlamak ve metot sonunda algoritma sonuçlarını elde etmektedir. Tek bir metot ile gerekli bilgiler üretilmiş olur. Arayüz uygulamasında Wiimote kumandasına benzer şekilde bir dikdörtgenler prizması çizdirilmektedir, yönelim ve konum bilgileri uygun şekilde bu birime aktarılmaktadır. Şekil 5.4'de arayüz programına ilişkin ekran çıktıları verilmiştir. Programda çizdirilen dikdörtgenler prizması hareketlerin daha belirgin olması için renklendirilmiştir. Kullanılan örnek donanım Wiimote kumandasının yönelimi değiştirildikçe aynı yönelim değişimi arayüzdeki dikdörtgenler prizmasına yansıtılmaktadır. Kullanıcının yaptığı konum değişimi de arayüzdeki koordinatlara göre ölçeklendirilerek gösterilir. Arayüz yazılımı aynı zamanda Wiimote kumandasının arkasında bulunan B tuşuna basıldığında hareket kestiricinin başlangıç durumuna (bütün yönelim açılarının ve konum bilgilerinin sıfır olduğu durum) dönmesi için gerekli metodunu çağırır. Böylece kullanıcı herhangi bir hareket yaptıktan sonra Wiimote kumandasını başlangıç durumuna getirerek bu tuşa basar. Böylece hareket kestirim algoritmalarındaki veriler sıfırlanır ve sistem yeni hareket için hazır duruma gelir. Arka arkaya çeşitli hareketlerin gerçekleştirilmesi bu yöntemle olanaklı hale getirilmektedir.



Şekil 5.4 : 3D Arayüz Yazılımı

6. SONUÇ VE ÖNERİLER

Tez kapsamında hedeflendiği üzere ilk olarak örnek donanım belirlenen Wiimote oyun kumandasının ivmeölçer ve açısal hız duyargası verilerine erişildi ve gürültü karakteristikleri çıkartıldı. Dinamik sistem tasarımı ve kestirim işlemlerine geçmeden önce kullanılması düşünülen alçak geçiren filtre için denemeler yapıldı ve ortalama alan filtrenin hareketi kaybetmeyecek düzeyde tutulması ile yeterli iyileştirmeyi sağladığı gözlemlendi.

Yönelim hesaplama işleminde açı tabanlı, koordinat dönüşüm matrisi tabanlı ve dördey tabanlı yöntemler değerlendirildi ve çıktıları karşılaştırıldı. İvmeölçer verileri ile yönelim hesabında düzeltme yapılması için kalman filtresi tasarlandı. Hem hesaplama yönteminin etkinliği hem de kalman filtresi için sistem modelleme işlemine uygunluğu göz önüne alınarak dördey tabanlı yönelim hesaplaması gerçekleştirildi. Yalnızca açısal hız verileri kullanılarak elde edilen yönelim bilgileri ve ivmeölçer verileri ile düzeltilen yönelim bilgileri karşılaştırıldı. İvmeölçer verileri ile düzeltilmiş yönelim bilgilerinde dikilme ve yuvarlanma açılarında meydana gelen kayıklıkların giderildiği gözlemlendi. Kalman filtresi kullanımında proses ve ölçüm gürültü kovaryanslarının düzenlemesi ile elde edilen sonuçların tahmin edilen ya da ölçülen değerlere yaklaştırılabildiği gözlemlendi. Daha açık ifade edilirse sonuçların elde edilmesinde açısal hız duyargası ve ivmeölçer duyargası verilerinin ağırlıkları değiştirilmiş oldu. Yönelim hesabında diğer bir kritik nokta kumandanın dinamik ivmeye sahip olmadığı durumun denetimiydi. İvmeölçerden elde edilen toplam ivmenin ve referans uzayı z eksenine düşen ivmenin yerçekimi ivmesine yakınlığı belirlenen bir eşik değer ile kontrol edildi. Bu eşik değer büyüdükçe ivmeölçer doğrulaması artmakta fakat ivmeölçerden alınan verinin doğruluğu azalmaktadır. Bu yüzden eşik değer bu iki ölçüte göre en uygun şekilde belirlenmesi gerektiği görüldü.

Konum hesaplama işleminde gerçekleştirilen kalman filtresi yönelim kestirimindeki yapıdan daha farklıdır. Yönelim hesaplamada temel hedef açılal hız duyargası verilerinden elde edilen sonuçların ivmeölçer verileri ile düzeltilmesidir. Konum kestiriminde ise başka bir ölçüm kaynağına göre düzeltme gerçekleştirilmemektedir. Konum kestiriminde ivmeölçer duyargası hataları modellenmiştir, proses ve ölçüm gürültüsü kovaryanslarının belirlenmesinde parametreler değiştirilerek ivmeölçer sinyallerinin hareketi algılama düzeyleri belirlenmiştir. Gerek ivmeölçer duyargasının titreşimlere daha duyarlı olmasından, gerekse konum elde etmek için ivmeölçer verisinde ikinci bir integrasyona gerek duyulmasından dolayı konum kestiriminde yönelim kestirimine göre daha hatalı sonuçlar elde edilmektedir. Bu yüzden konum kestirimindeki diğer önemli nokta ise hareket sonlandırma işlemidir. Yönelim kestirimine benzer fakat ek denetimler içeren bir hareketsizlik kontrolü yapılmıştır. Bu işlemde ek olarak belirli bir süreç boyunca ölçülen ivmenin yerçekimi ivmesine yakınlığı da kontrol edilmektedir. Bu kontroldeki sürecin uzunluğu ve kısalığı da filtre performansını etkileyen diğer bir etken olarak gözlenmiştir. Süreç çok uzun tutulursa hareketsizlik geç algılanmakta, çok kısa tutulursa da hareketli bir durumun hareketsiz algılanmasına sebep olmaktadır. Bu değer belirtilen ölçütler göz önüne alınarak en uygun şekilde seçilmelidir. Yapılan çalışmalarda duyarga hatalarının hedef uygulama kapsamında büyük ölçüde giderilmesi ve gerçekleştirilen bir üç boyutlu grafik uygulaması ile görselleştirilmesi sağlandı. Yazılım geliştirme konusunda en önemli kısım tasarım kalıpları kullanılarak modüler bir hareket kestirim kütüphanesinin geliştirilmesiydi. Bu kütüphanenin kullanımı ile yeni modüller geliştirme ve sisteme uyarlama kolaylığı sağlandı.

Tez kapsamında gerçekleştirilen çalışmanın devamı için donanım katmanında, hareket kestirimi katmanında ve uygulama katmanında olmak üzere çeşitli geliştirmeler yapılabilir.

İlk olarak kestirim işleminde kullanılan kalman filtresinin hata-durum (“error-state”) yaklaşımı ile tasarlanması denenebilir. Kalman filtresindeki hata-durum yaklaşımı, sistem durum denklemlerinde hata değişkenlerinin bulunmasını, yani filtrenin sistemde oluşabilecek hataları tahmin etmesi üzerine geliştirilmiştir ve istenen sonuçlara bu yaklaşım ile varılıp varılamayacağı üzerine çalışılabilir.

Daha etkin bir çözüm donanım katmanında iyileştirmeye gitmektir. Bu, elde edilen sonuçların manyetik alan duyargası gibi başka duyarga verileri ile bütünleştirilmesi ve daha az hataya sahip bir hareket kestiriminin ek girdilerle sağlanması anlamına gelmektedir. Fakat bu yaklaşım için farklı bir donanım ihtiyacı doğmaktadır.

Uygulama katmanındaki iyileştirme ise uygulamanın hedefine göre tanımlanacak hareket modelleri ile hareket kestirimi sonuçlarının sentezlenmesi ile elde edilebilir. Örneğin üç boyutlu bir bilgisayar oyununda eldeki donanım ile kılıç kullanıldığı varsayalım. Kılıç hareketlerinin doğrudan hareket kestirim algoritması sonuçlarına göre belirlenmesi yerine aynı zamanda kılıç hareketleri için belirlenen bir menzil ve hareket modeline uydurulması sağlanabilir. Nintendo'nun Wiimote için geliştirdiği oyunlara bakıldığında algoritmaların daha çok örüntü tanıma ağırlıklı çalıştığı gözlemlenmiştir. Örneğin tenis oyununda kumandanın hareketinin birebir raket olarak gösterimi değil sadece farklı vuruş örüntülerine dönüştürülerek gösterimi gerçekleştirilir. Yapılan hareket tenis oyunu için belirlenen vuruş örüntüleri kümesinde en yakın örüntüye benzetilir. Aynı yaklaşım golf oyunu için de benzerdir. Tez kapsamında yaklaşım ile Wii oyunlarındaki yaklaşımın temel farkı budur. Tez kapsamında doğrudan duyarga verileri ile gerçek hareket kestirilmeye çalışılmıştır. Bu yaklaşım bahsedilen örüntü kümesinin daha fazla çeşitlendirilebilmesine olanak sağlamaktadır. Gerçeğe yakın hareket tespiti uygulama kapsamındaki hareket örüntülerine uydurulduğunda çok daha kapsamlı hareket modelleri çıkartmak mümkündür.

Ayrıca donanım ve hareket kestirimi katmanında yapılan işlemler ataletsel seyir ve güdüm sistemlerinde yapılan işlemlere benzerdir. Örneğin tez kapsamındaki yönelim kestirimi örnek bir AHRS işlevi görmektedir. Daha hassas ivmeölçer ve açısal hız duyargalarına ek olarak manyetik alan duyargaları içeren bir donanım ile daha gelişmiş bir AHRS tasarlanabilir. Bu donanıma GPS modülünün eklenmesi ile bütünleştirilmiş seyir ve güdüm sistemlerine geçiş yapılabilir. Bu çalışmalarda temel hedef tez kapsamında da olduğu gibi maliyeti görece daha düşük olan donanımlarla, daha hassas ve daha üst sınıftaki AÖÜ'lere yakın hareket kestirimi yapabilen sistemler geliştirmek olacaktır.

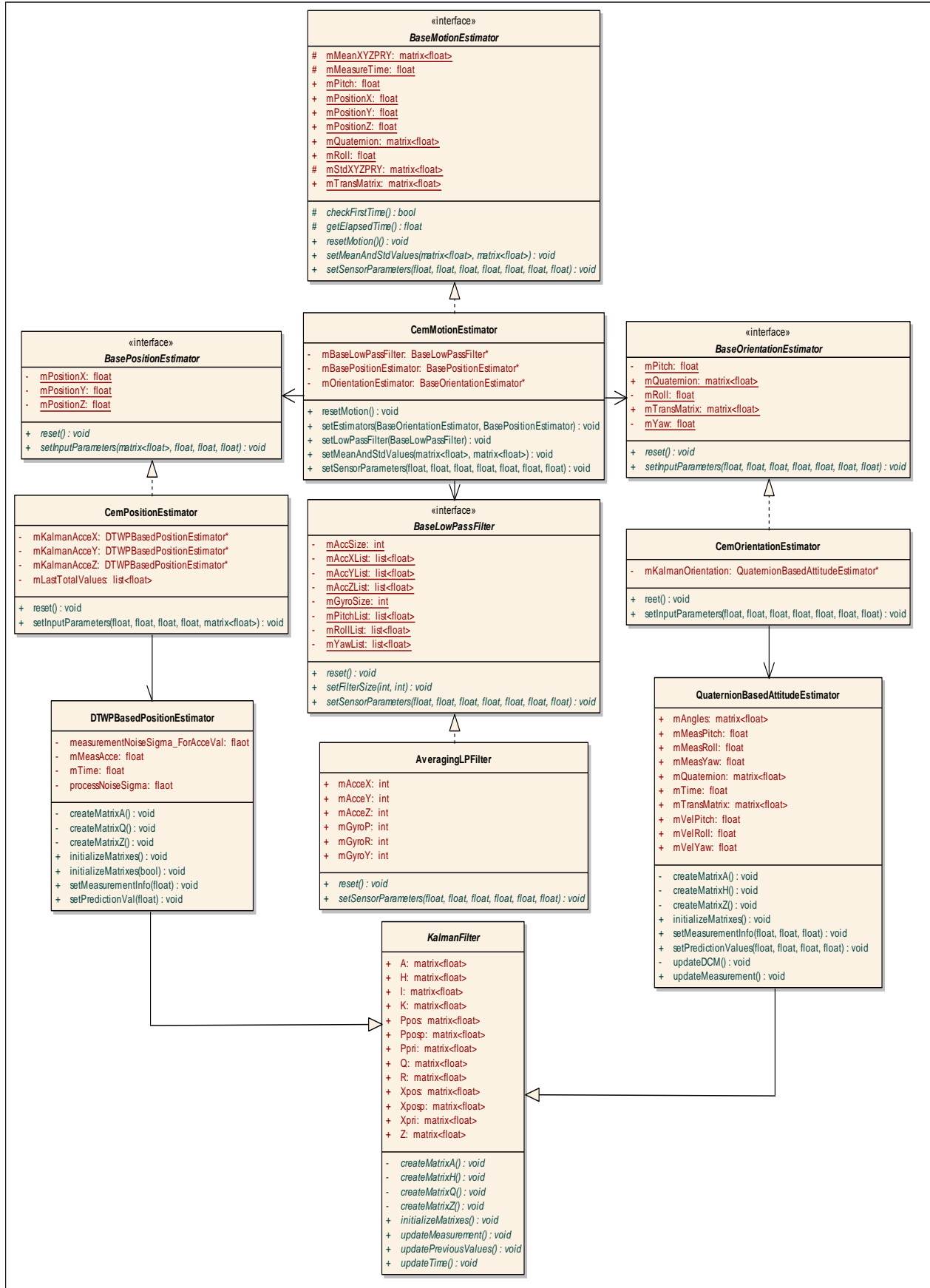
KAYNAKLAR

- [1] **Rahni, A.A.A. and Yahya, I.**, 2007: Obtaining Translation from a 6-DOF MEMS IMU – an Overview: *2007 Asia-Pasific Conference On Alpllied Electromagnetics Proceedings*, Melaka, Malaysia, December 4-6, 2007
- [2] **Foxlin, E., Harrington, M. and Altshuler, Y.**, 1998: Miniature 6-DOF inertial system for tracking HMDs in *SPIE vol. 3362, Helmet and Head-Mounted Displays III, AeroSense 98*, Orlando, FL, April 13-14, 1998
- [3] **Huang, Z.J. and Fang, J.C.**, 2005: Integration of MEMS Inertial Sensor-Based GNC of a UAV in *International Journal of Information Technology*, Vol. 11 No. 10, 2005
- [4] <<http://www.answers.com/topic/inertial-navigation-system-1>>, alındığı tarih 05.02.2010
- [5] **Ross, L.**, 2009: Obtaining Commercialization of MEMS Technology: Current Trends, poster session presented at *the The Audacity Of Imagination: Microelectronics' New Millenium*, Kitchener Waterloo, Ontario, Canada, April 14-15, 2009
- [6] <<http://healthcare.analog.com/en/otherproducts/multichip/adis16350/products/product.html>>, alındığı tarih 01.05.2010
- [7] <<http://www.tkt.cs.tut.fi/kurssit/>>, alındığı tarih 03.01.2010
- [8] <http://www.sparkfun.com/commerce/tutorial_info.php?tutorials_id=43/>, alındığı tarih 03.01.2010
- [9] <<http://www.symbolicsound.com/cgi-bin/bin/view/Learn/UsingTheNintendoWiimoteWithKyma/>>, alındığı tarih 02.05.2010
- [10] <<http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>>, alındığı tarih 02.01.2010
- [11] <<http://www.oxts.com/default.asp?pageRef=38&category=10&faq=30/>>, alındığı tarih 02.05.2010
- [12] **Arkadani, H. A., Bridges, T. J.**, 2010, “*Review of the 3-2-1 Euler Angles: a yaw-pitch-roll sequence*”, pp.7-8, Teknik Rapor, Matematik Bölümü, Surrey Üniversitesi, İngiltere.
- [13] **Groves, P.D.** , 2008: Navigation Mathematics in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, pp. 17-36. ISBN-13: 978-1-58053-255-6.
- [14] **Groves, P.D.** , 2008: Inertial Navigation in *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, pp. 122-129. ISBN-13: 978-1-58053-255-6.

- [15] **Marmion, M.**, 2006: “*Airborne attitude estimation using a Kalman filter*”, pp.15-18, 57-65, Yüksek Lisans Tezi, Svalbart Univesite Merkezi, Norveç.
- [16] **Bar-Shalom, Y., Li, X.R. and Kirubarajan, T.**, 2001: Coordinate Transformation in *Estimation with Applications to Tracking and Navigation*, p. 515, A Wiley-Interscience Publication, New York, USA
- [17] **Bar-Shalom, Y., Li, X.R. and Kirubarajan, T.**, 2001: Introduction in *Estimation with Applications to Tracking and Navigation*, pp. 1-6, A Wiley-Interscience Publication, New York, USA
- [18] **Welch, G. and Bishop, G.**, 2001: An Introduction to the Kalman Filter, <<http://www.cs.unc.edu/~welch/kalman/>>, alındığı tarih 29.10.2009.
- [19] **Bar-Shalom, Y., Li, X.R. and Kirubarajan, T.**, 2001: Estimation For Kinematic Models in *Estimation with Applications to Tracking and Navigation*, pp. 72-77, A Wiley-Interscience Publication, New York, USA

EKLER

EK A : Hareket Kestirim Kütüphanesi Ayrıntılı Sınıf Diyagramı



ÖZGEÇMİŞ

Ad Soyad: Cem Küççük
Doğum Yeri ve Tarihi: Nazilli, 04.11.1985
Adres: TÜBİTAK BİLGEM BTE Gebze/KOCAELİ
Lisans Üniversitesi: İstanbul Teknik Üniversitesi

Yayın Listesi:

- **Küççük, C.** , Özkan, E. , Altılar, D.T., 2010: A Novel Software Framework for 3D Motion Estimation with A 6-DOF MEMS IMU for A Virtual Reality Environment. *International Science and Technology Conference*, Turkish Republic of Northern Cyprus, October 27-29, 2010.
- Özkan, E. , **Küççük, C.** , Altılar, D.T., 2010: A Hardware Framework Design for 3D Motion Sensing, Detection and. Estimation. *International Science and Technology Conference*, Turkish Republic of Northern Cyprus, October 27-29, 2010.