

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

**GEOMETRİK UZAYLAR
VE UYGULAMALARI**

Hatice KUŞAK

Tez Danışmanı : Prof. Dr. Ali ÇALIŞKAN

Matematik Anabilim Dalı

Bilim Dalı Kodu : 403.02.01

Sunuş Tarihi : 21.01.2011

Bornova-İZMİR

2011

Hatice KUŞAK tarafından doktora tezi olarak sunulan “Geometrik Uzaylar ve Uygulamaları” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 21.01.2011 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı : Prof. Dr. Ali ÇALIŞKAN

Raportör Üye: Prof.Dr. H.Hüseyin UĞURLU

Üye : Prof. Dr. Urfat NURIYEV

Üye :Doç. Dr. Fuat OKUMUŞ

Üye : Yard. Doç. Dr. Serap TOPAL

ÖZET**GEOMETRİK UZAYLAR VE UYGULAMALARI**

KUŞAK, Hatice

Doktora Tezi, Matematik Bölümü

Tez Yöneticisi: Prof. Dr. Ali ÇALIŞKAN

Ocak 2011, 82 sayfa

Bu tezde geometrik uzaylardaki bazı yapılar teorik olarak incelenmiş, her yapı için robotik, mekanik, kinematik ve tasarım gibi alanlarda bazı uygulamalar verilmiştir.

Tezde çeşitli sayı sistemleri, Clifford cebirleri ve Öklid uzayı ile ilgili temel kavramlar verilmiştir. Temel dönüşümler hatırlatılarak, düzlemsel kinematik ile ilgili teorik bir çalışma yapılmıştır. BDGT deki interpolasyon teknikleri ile mekanik bir uygulama çalışması yapılmıştır. Öklid uzayındaki geometrik yapılar, herhangi bir zaman skalasında incelenerek daha genel sonuçlar elde edilmiştir. Dual uzayların temel kavramları yardımıyla robotik hareketler ile ilgili bir uygulama verilmiştir. Galile düzleminin temel kavramları ve bazı bilgisayar uygulamaları ele alınarak, trigonometrik fonksiyonlar incelenmiştir.

Bu tezin amacı, geometrik uzaylar arasındaki benzerlikleri ve farklılıkları dikkate alarak geometrinin; robotik çalışmalarda, mekanikte, kinematikte, bilgisayar bilimlerinde ve özellikle tasarımda etkin bir biçimde nasıl kullanıldığını bilgisayar programları yardımıyla ortaya koymaktır.

Anahtar sözcükler: Geometrik uzaylar, Galile düzlemi, Dual Uzay, Kuaterniyon, Zaman skalası, BDGT.

ABSTRACT

GEOMETRIC SPACES AND THEIR APPLICATIONS

KUŞAK, Hatice

Ph.D. in Mathematics

Supervisor: Prof. Dr. Ali ÇALIŞKAN

January 2011, 82 pages

In this thesis, different structures are being investigated, some applications are done at the subjects like robotic, mechanical, kinematical, design for each space.

In the thesis various number systems and Clifford Algebras are introduced, and the concepts about Euclidean space are given. Some transformations in Euclidean Geometry are mentioned also, a theoretical study is done about plane kinematic and a mechanical application study is done by the interpolation methods. More general results are obtained by investigating the geometric structures on any time scale. An application is given about robotic motions by the main terms of dual spaces. Basic properties of Galilean plane and its computer applications are given.

The main purpose of this thesis is; the similarities and the differences between the spaces in geometry at the robotic studies, mechanical, kinematical, computer sciences also especially the usage in designment are the provided.

Keywords: Geometric spaces, Galilean space, Dual space, Quaternions, Time scale, CAGD.

TEŞEKKÜR

Bu uzun ve zorlu çalışma süresince desteklerini ve görüşlerini aldığım değerli danışman hocam Sayın Prof. Dr. Ali ÇALIŞKAN'a çok değerli katkılarından dolayı teşekkürü bir borç bilirim. Değerli katkılarını aldığım değerli hocam Sayın Prof. Dr. Hasan Hüseyin UĞURLU'ya emeklerinden dolayı yürekten teşekkür ederim. Tez izleme jürisinde olup, ilgisini ve zamanını esirgemeyen hocam Sayın Prof. Dr. Urfat NURİYEV'e katkılarından dolayı çok teşekkür ederim. Zamanlarını esirgemeyip savunma sınavına katkıda buldukları için sınav komitesi üyelerine teşekkür ederim. Tez projeme E.Ü. Bilimsel Araştırma Projeleri (BAP) kapsamında destek verdiği için Ege Üniversitesi'ne teşekkür ederim. Ayrıca çok büyük sabır ve hoşgörü gösteren annem Hafize, kardeşim Hasibe, babam Çetin KUŞAK ve sevgilerini esirgemeyen ailemin diğer fertlerine teşekkürü bir borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
ABSTRACT	vii
TEŞEKKÜR	ix
ŞEKİLLER DİZİNİ.....	xv
ÇİZELGELER DİZİNİ	xvii
SİMGELER VE KISALTMALAR DİZİNİ.....	xix
1.GİRİŞ	1
2. SAYI SİSTEMLERİ VE CLIFFORD CEBİRLERİ.....	3
2.1 Karmaşık Sayılar	3
2.2 Dual Sayılar.....	5
2.3 Kuaterniyonlar.....	6
2.4 Kuaterniyon Cebri için Bir Uygulama Çalışması	8
2.5 Dual Kuaterniyonlar.....	12
3. ÖKLİD UZAYI.....	14
3.1 Temel Kavramlar	14
3.2 Öklid Uzayında Mekanik Bir Uygulama	15
3.2.1 Önbilgiler ve tarihçe.....	15

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.2.2 Engebeli arazi için BDGT interpolasyon metodu	16
3.2.3 Yeni bir tekerlek yapısı için bazı diferansiyel geometrik yaklaşımlar..	18
3.2.4 Sonuç	23
3.3 Öklid Uzayında Dönüşümler	24
3.4 Öklid Düzlem Dönüşümlerinin Bir Kinematik Uygulaması	25
3.4.1 Türev Denklemleri	27
3.4.2 Bağıl, Mutlak ve Sürüklenme Hız Vektörlerinin Birleşimi	28
3.4.3 Ani Shearing Polü ve Pol Eğrisi.....	30
3.4.4 Sonuçlar.....	31
4. ZAMAN SKALASINDA GEOMETRİK YAPILAR.....	35
4.1 Zaman Skalası ile İlgili Temel Bilgiler	35
4.2 Zaman Skalasında Türev Dönüşümü ve Doğal Konneksiyon	37
5. DUAL UZAY.....	44
5.1 Temel Kavramlar.....	44
5.2 Robotik Hareketler	46

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
5.3 Dual Kuaterniyon ve Dual Uzayların Robotik Bir Uygulaması.....	49
5.3.1 İki dual vektörün çarpımı ile dual kuaterniyon elde etme.....	50
5.3.2 Bir vidayı dual kuaterniyon yardımıyla döndürme.....	52
5.3.3 Bir dual kuaterniyon ile belirlenen vida dönmelerinin dizisel metod ile hareketleri	53
5.3.4 Bir dual kuaterniyonun dönme hareketi ile tanımlanan modüler metod	54
5.3.5 Dual küresel mekanizmalarda kuaterniyonlarla tanımlanan dirsek hareketi	55
5.3.6 Dual küresel dirsek hareketleri sonucu oluşan yörünge yüzeyleri olarak regle yüzeyleri	56
5.3.7 Sonuç	57
6. GALİLE DÜZLEMİ	58
6.1 Galile Düzlemi Temel Kavramları	58
6.2 Galile Düzleminde Dönüşümler	65
6.3 Galile Düzleminde Trigonometri.....	66
7. GENEL SONUÇLAR.....	76

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
KAYNAKLAR.....	77
ÖZGEÇMİŞ.....	81
EKLER	
Ek 1 Kuaterniyon Cebri Hesap Makinesi Programının Kodları	
Ek 2 Shearing Kinematik ile İlgili Mapple Kodları	
Ek 3 Shear Dönüşümü olmadan Normal Kinematiğin Mapple Kodları	
Ek 4 Galile Dönüşümlerinin Bilgisayar Uygulamalarının Program Kodları	

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1Kuaterniyon Cebri Hesap Makinesi Programının klasör simgesi.....	8
2.2 Hesap Makinesinin Veri Girme Sayfası.....	8
2.3a Verilen Kuaterniyonun bileşenlerini klavyeden girme.....	9
2.3b Verilen Kuaterniyonun bileşenlerinin bazlarla yazılışı.....	9
2.3c Verilen Kuaterniyonun tersi	9
2.3d Verilen Kuaterniyonun normu	9
2.3e Verilen Kuaterniyonun eşleniği.....	9
2.4a Verilen kuaterniyonların bileşenlerinin klavyeden girilişi	10
2.4b Verilen kuaterniyonların çarpımın sonucu	10
2.5a Verilen kuaterniyonun bileşenlerinin klavyeden girilişi	10
2.5b Verilen kuaterniyonun üssü.....	10
2.6a Verilen kuaterniyonun bileşenlerinin klavyeden girilişi	10
2.6b Verilen kuaterniyonun toplamı.....	10
2.7a Verilen kuaterniyonun bileşeninin klavyeden girilişi.....	11
2.7b Verilen kuaterniyonun skalerle çarpımı	11
2.8a Verilen kuaterniyonun dönme açısının klavyeden girilişi.....	11
2.8b Verilen kuaterniyonun dönme matrisinin hesabı	11
2.8c Verilen kuaterniyon vektörünün döndürülmüş hali.....	11

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
2.9 Vida hareketi	13
3.1 Kübik splaynın uç noktaları ve teğet vektörleri	17
3.2 Kübik splayn modellenmesi	18
3.3 Üç boyutlu uzayda bir çember	19
3.4 Üç boyutlu uzaydaki Çemberlerle Tekerlek Modellemesi.....	20
3.5 Üç boyutlu uzayda oskülatör çember	20
3.6 Bir kübik splayn üzerindeki eğrilik değişimlerinin bir örneği	21
3.7 Üç boyutlu uzayda bir küre	22
3.8 Üç boyutlu uzayda oskülatör küre.....	22
3.9 Üç boyutlu uzayda küresel tekerleklerle araç modellenmesi	23
3.10a \vec{u} öteleme vektörünün shear hareketi	32
3.10b \vec{u} öteleme vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı.....	32
3.11a \vec{u} öteleme vektörünün türev vektörünün shear hareketi	32
3.11b \vec{u} öteleme vektörünün türev vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı	32

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
3.12a \vec{v}_x vektörünün shear hareketi.....	33
3.12b \vec{v}_x vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı	33
3.13a \vec{v}_r bağıl hız vektörünün shear hareketi	33
3.13b \vec{v}_r bağıl hız vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı	33
3.14a \vec{v}_f bağıl hız vektörünün shear hareketi.....	33
3.14b \vec{v}_f bağıl hız vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı	33
3.15a \vec{v}_a bağıl hız vektörünün shear hareketi.....	34
3.15b \vec{v}_a bağıl hız vektörünün doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı	34
3.16a Pol eğrisinin shear hareketindeki.....	34
3.16b Pol eğrisinin doğal hareketindeki Mapple çıktısı	34
5.1 Döner tip eklem	47
5.2 Kayar tip eklem	47
5.3 Kartezyen manipülatör	47

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
5.4 Silindirik manipülatör	48
5.5 Küresel manipülatör	48
5.6 Eklemlili-İnsan kolu manipülatör	49
5.7 Dual vektörler arasındaki dual açı.....	51
5.8 Bir vidayı bir eksen etrafında döndürme.....	52
5.9 Bir vidanın dizisel metod ile hareketi	53
5.10 Bir vidanın modüler metod ile hareketi.....	54
5.11 Küresel mekanizmanın çizdiği küresel eğri	55
5.12 Dual küresel mekanizmanın çizdiği regle yüzey.....	56
6.1a Öklid düzleminde dik bir doğru ve l' herhangi bir doğru	58
6.1b l doğrusunun ve l' doğrusunun Galile düzlemindeki karşılığı.....	58
6.2 Galile dönüşümünün doğru parçalarına etkisi.....	59
6.3 Galile dönüşümü için PASCAL programında bir örnek	60
6.4 Öklid düzlemindeki AB doğru parçasının Galile düzlemindeki CD doğru parçasına dönüştüren programın GSP de Grafik çıktısı.....	60

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
6.5a Galile düzleminde uzaklık.....	61
6.5b Galile düzleminde özel uzaklık.....	61
6.6 Galile çemberi	62
6.7 Galile açısı.....	63
6.8 Galile düzleminde özel doğrular	63
6.9 Galile düzleminde iki doğru arasındaki uzaklık.....	64
6.10 Galile düzleminde diklik	64
6.11 Çemberin Öklid Düzlemindeki Winplot program görüntüsü.....	65
6.12 Çemberin Galile Düzlemindeki Winplot program görüntüsü.....	66
6.13 Birim Galile çemberi.....	66
6.14 Galile trigonometrik fonksiyonları.....	67
6.15 Galile kosinüs fonksiyonu.....	67
6.16 Galile sinüs fonksiyonu.....	68
6.17 Galile tanjant fonksiyonu	69
6.18 Galile kotanjant fonksiyonu	69

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
6.19 Galile sing fonksiyonunun grafiği	72
6.20 Galile cosg fonksiyonunun grafiği	72
6.21 Galile tang fonksiyonunun grafiği.....	73
6.22 Galile cotg fonksiyonunun grafiği.....	74
6.23 Galile düzleminde bir üçgenin kenar uzunlukları ve açıları.....	75

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
6.1 Öklid düzleminde bazı trigonometrik değerler.....	71
6.2 Galile düzleminde bazı trigonometrik değerler	71
6.3 Galile düzleminde sing fonksiyonunun trigonometrik değerleri.....	72
6.4 Galile düzleminde sing fonksiyonunun trigonometrik değerleri.....	72
6.5 Galile düzleminde tang fonksiyonunun trigonometrik değerleri.....	73
6.6 Galile düzleminde cotg fonksiyonunun trigonometrik değerleri.....	73

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
IR	Reel sayılar kümesi.
\mathbb{C}	Karmaşık sayılar kümesi.
ID	Dual sayılar kümesi.
q	Kuaterniyon.
Q	Dual kuaterniyon.
Θ	Dual açı.
ϵ	Dual birim.
(β)	Beta eğrisi.
$H_{S_1} = E/E'$	E nin E' ne göre shear hareketi.
\vec{v}_f	Sürüklenme hız vektörü.
\vec{v}_r	Bağlı hız vektörü.
\vec{v}_a	Mutlak hız vektörü.
\mathbb{D}	Dual sayılar halkası.
Sing	Galile sinüsü.
 <u>Kısaltmalar</u>	
BDGT	Bilgisayar Destekli Geometrik Tasarım.
CAGD	Computer Aided Geometric Design.

1. GİRİŞ

Bilim adamları yüzyıllardır geometride çeşitli uzaylar keşfetmiş ve bu uzayların robotik, mekanik, mühendislik, bilgisayar, fizik vs. gibi alanlarda kullanımının nasıl elverişli olabileceği üzerinde araştırmalar yapmışlardır. Bu nedenle, bu tezde genel olarak, geometrik uzaylardaki yapıların çeşitli bilim sahalarıyla ilişkileri araştırılmaya çalışılmıştır. Farklı geometrik uzayların cebirsel yapıları hakkında bazı temel kavramlar hatırlatıldıktan sonra, bu geometrik uzaylar üzerinde analiz, kinematik, mekanik ve tasarım açısından bazı uygulamalara yer verilmiştir.

İkinci bölümde, Clifford cebirlerinin çıkış noktası olan karmaşık sayılar tanıtılmış, dual sayılar ve kuaterniyon ile ilgili temel kavramlar verilmiştir. Ayrıca kullanım pratikliğini sağlamak amacıyla kuaterniyon cebirini esas alan bir hesap makinesi programı yapılmıştır.

Karmaşık sayılar, ilk defa, İtalyan matematikçi Gerolamo Cardano (1501-1576) tarafından bulunmuş ve tasarlanmıştır (Hillman et al., 2005). Daha sonraları, karmaşık sayıyı reel düzlemde bir nokta olarak alan Fransız matematikçisi ARGAND' dır. R^2 ile gösterilen düzlemdeki tüm (x, y) ikilileri artık bir karmaşık sayı olarak düşünülmektedir (Dönmez, 1999). Böylece, Öklid uzayını karmaşık sayılar yardımı ile inceleme fırsatı ortaya çıkmıştır. Düzlemde dönmeyi ifade eden karmaşık sayılar, uzay için yetersiz kalmaktadır. Bunun üzerine, uzaydaki dönme dönüşümlerinin kullanımını mekanik anlamda kolaylaştıran farklı bir yapı olarak, 1843 yıllarında William Rowan Hamilton tarafından “**kuaterniyonlar**” tanımlanmıştır. Bir kuaterniyon, karmaşık sayıların genelleştirilmesi ile oluşan üç sanal elemanlı dört bileşenli bir yapıdır. Hamilton kuaterniyonları tanımlamakla iki vektör için, bölümün de mümkün olabileceğini göstermiş, yeni bir çarpım işlemini vektör cebirine dahil etmiş ve böylece üç boyutlu uzayda hareketlerin tetkikini kolaylaştırmıştır (Hamilton R., 1844).

Üçüncü bölümde, Öklid uzayının temel kavramları verilmiş ve sonrasında araçların hareketini sağlayan tekerlek yapısına Öklid uzayındaki diferensiyel geometrik özelliklerden yararlanılarak mekanik açıdan yeni bir bakış açısı getirilmiştir. Ayrıca, Öklid uzayında gerçekleşebilen düzlemsel kinematiğe shear dönüşümü ile farklı bir geometrik anlam yüklenmiştir.

Dördüncü bölümde, analizde kullanımı oldukça yaygın olan zaman skalasının geometrik uygulamalarına yer verilmiştir. Son zamanlarda uygulamalı bilimler ve analizde zaman skalasının kullanımı hızlı bir biçimde artmaktadır. İlk olarak, Stefan Hilger (1989) doktora tezinde zaman skalası teorisini ele alarak sürekli analiz ve ayrık analizi birleştirme ve genelleme imkanı bulmuştur. Bu tezde, bazı geometrik yapılar zaman skalasında incelenmiştir.

Beşinci bölümde, dual uzayın ve mühendislikteki robotik hareketlerinin temel kavramları ele alınmış ve dual uzaylardaki vida yapıları ile robotik hareketler incelenmiştir.

Yüzyıllardır bir taraftan çeşitli sayı kümeleri üzerine cebirsel anlamda araştırmalar yapılırken, diğer taraftan bu sayı kümelerinin geometrik yapıları ile ilgili önemli araştırmalar yapılmıştır. Bu sayı kümelerinden birisi de dual sayılardır. Dual sayılar 19. yüzyılda Clifford tarafından tanımlanmış, Kotelnikov ve Study tarafından geliştirilerek katı cisim kinematiğine uygulamaları verilmiştir. Ayrıca dual sayı cebri üzerine (Blaschke, W. 1949), (Müller, 1963), (Köse, 1974), (Veldkamp, 1976), (Agrawal, 1985) ve diğer matematikçiler tarafından günümüze kadar özgün çalışmalar yapılmış ve konu ile ilgili araştırmalar devam edilmektedir. Kuaterniyonlar gibi uzayda dönme hareketlerini karakterize eden yapıların yanında, dönme ve ötelemeyi birlikte ifade eden farklı bir yapı olarak, dual kuaterniyonlar keşfedilmiştir (Yang 1964).

Altıncı bölümde, Galile düzleminin temel kavramları verilmiş ve Galile bu düzlemin trigonometrisi ile ilgili temel formüller ispatlanmış ve geometrik yorumları verilmiştir.

2. SAYI SİSTEMLERİ VE CLIFFORD CEBİRLERİ

2.1 Karmaşık Sayılar

Teorem 2.1.1: Karmaşık sayılar kümesi \mathbb{C} , bir cisim yapısına sahiptir.

Teorem 2.1.2: $z \in \mathbb{C}$ olsun. $w^2 = z$ olacak şekilde $w \in \mathbb{C}$ vardır.

Tanım 2.1.1: $a, b \in \mathbb{R}$ ve $(a, b) \in \mathbb{C}$ için $(a, b) = (a, 0) + (0, b)$ yazılabilir. Öte yandan $(b, 0) \cdot (0, 1)$ çarpımı yapıldığında $(0, b)$ eşit olduğundan (a, b) karmaşık sayısı

$$(a, b) = (a, 0) + (0, 1) \cdot (b, 0) = a + ib$$

“standart biçimi” ile tanımlanabilir.

Tanım 2.1.2: $a, b \in \mathbb{R}$ olmak üzere $z = a + ib$ karmaşık sayısının eşleniği “karmaşık sayısının eşleniği” $\bar{z} = a - ib$ dir.

Tanım 2.1.3: $x_j, y_j \in \mathbb{R}$ ve $j = 1, 2$ olmak üzere $z_j = x_j + iy_j$ “karmaşık sayılarında işlemler” aşağıdaki gibi tanımlanır:

$$1) \text{ Toplama : } z_1 + z_2 = (x_1 + x_2) + (y_1 + y_2)i$$

$$2) \text{ Çarpma : } z_1 \cdot z_2 = (x_1x_2 - y_1y_2) + (x_1y_2 + x_2y_1)i$$

olarak tanımlanır.

Tanım 2.1.4: $x, y \in \mathbb{R}$ olmak üzere $z = x + iy$ karmaşık sayısının \mathbb{R}^2 düzleminde başlangıç noktasına olan uzaklığı, bu sayının modülü olarak isimlendirilen

$$|z| = |x + iy| = \sqrt{x^2 + y^2}$$

reel sayısı olarak tanımlanır.

Tanım 2.1.5: $z = (x, y)$ karmaşık sayıların kümesine “karmaşık düzlem” denir.

Tanım 2.1.6: $x_j, y_j \in \mathbb{R}$ ve $j = 1, 2$ olmak üzere $z_1 = x_1 + iy_1$ ve $z_2 = x_2 + iy_2$ noktaları arasındaki “uzaklık”

$$|z_1 - z_2| = |AB| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

denklemleri ile tanımlıdır.

Tanım 2.1.7: $x, y \in \mathbb{R}$ olmak üzere $z = x + iy$ standart formunda verilen karmaşık sayının

$$z = |z|(\cos \theta + i \sin \theta)$$

biçiminde yazılışına “karmaşık sayının kutupsal formu” denir.

Tanım 2.1.8: $\forall n \in \mathbb{Z}$ için karmaşık sayılarda

$$z^n = |z|^n (\cos n\theta + i \sin n\theta)$$

eşitliği ile “De Moivre kuralı” tanımlanır (Hillman et al., 2005).

Tanım 2.1.9: $e^{i\theta} = \cos \theta + i \sin \theta$ ile gösterilen

$$\mathbb{R} \rightarrow \mathbb{C}$$

$$\theta \rightarrow e^{i\theta}$$

operatörüne “kompleks sayı operatörü” denir.

Tanım 2.1.10: $\alpha, \beta \in \mathbb{R}$ olmak üzere

$$\begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix}$$

formundaki matrislerin cümlesi $z = \alpha + i\beta$ biçimindeki karmaşık sayılardan oluşan \mathbb{C} kompleks sayılar cümlesine izomorfdur (Yang, 1963).

2.2 Dual Sayılar

Tanım 2.2.1: $x, x^* \in IR$ ve $\varepsilon^2 = 0$ olmak üzere $\hat{x} = x + \varepsilon x^*$ şeklinde tanımlanan sayılara “**dual sayılar**” denir. Dual sayıların kümesi

$$ID = \{\hat{x} = x + \varepsilon x^* : x, x^* \in IR\}$$

ile gösterilir. Burada x ve x^* bileşenleri \hat{x} dual sayısının, sırasıyla, reel ve dual kısımlarıdır.

Tanım 2.2.2: Dual sayılarda dört işlem aşağıdaki şekilde verilebilir:

1) Toplama: $\hat{x} + \hat{y} = (x + \varepsilon x^*) + (y + \varepsilon y^*) = (x + y) + \varepsilon(x^* + y^*)$

2) Çarpma: $\hat{x}\hat{y} = (x + \varepsilon x^*)(y + \varepsilon y^*) = xy + \varepsilon(xy^* + x^*y)$

3) Eşitlik: $\hat{x} = \hat{y}$ ancak ve ancak $x = y$ ve $x^* = y^*$ olur.

4) Skalerle Çarpma: $\lambda(x + \varepsilon x^*) = \lambda x + \varepsilon \lambda x^*$

5) Bölme: $\frac{\hat{x}}{\hat{y}} = \frac{x + \varepsilon x^*}{y + \varepsilon y^*} = \frac{(x + \varepsilon x^*)(y - \varepsilon y^*)}{(y + \varepsilon y^*)(y - \varepsilon y^*)} = \frac{x}{y} + \varepsilon \frac{x^*y - xy^*}{y^2}$ ve $y=0$ için

sonsuzdur (Köse, 1974).

Teorem 2.2.1: $(ID, +, \bullet)$ üçlüsü bir cisim değildir.

Tanım 2.2.3: Dual sayılar $x, x^* \in IR$ ve $\varepsilon^2 = 0$ olmak üzere $\hat{x} = x + \varepsilon x^*$ şeklinde gösterilebildiği gibi aynı zamanda $\hat{x} = (x, x^*)$ sıralı ikilileriyle de gösterilebilmektedirler.

Tanım 2.2.4: $z = (x, x^*)$ dual sayılarının oluşturduğu noktalar kümesine “**dual düzlem**” denir ve ID ile gösterilir. Her bir (x, x^*) ikilisine de dual düzlemin bir “**dual noktası**” denir.

Tanım 2.2.5: $z = x + \varepsilon x^*$ “**dual sayısının eşleniği**” diye $z = x - \varepsilon x^*$ dual sayısına denir.

Tanım 2.2.6: G, ID dual sayılar kümesinin boştan farklı bir alt kümesi olsun. G üzerindeki bir f fonksiyonu her \hat{x} için G de bir \hat{y} dual sayısına götürüyorsa \hat{y} dual sayısına f fonksiyonunun \hat{x} deki değeri denir, ve $\hat{y} = f(\hat{x})$ şeklinde gösterilir. \hat{y} nin

reel kısmı ve dual kısmı, sırasıyla, $\text{Re } \hat{y} = u(x, x^*)$ ve $\text{Du } \hat{y} = u^*(x, x^*)$ şeklindedir. Böylece

$$\hat{y} = f(\hat{x}) = u(x, x^*) + \varepsilon u^*(x, x^*)$$

denklemini ile “**f dual değişkenli fonksiyonu**” reel ve dual kısımlarıyla birlikte yazılabilir (Köse, 1974).

Teorem 2.2.2: $a, a^* \in \mathbb{R}$ olmak üzere

$$f(A) = \begin{bmatrix} a & a^* \\ 0 & a \end{bmatrix}$$

formundaki matrislerin cümlesi \mathbb{IM} ise $(\mathbb{IM}, +, \cdot)$ ve $(\mathbb{ID}, \oplus, \otimes)$ halkaları izomorftur (Yang, 1963).

2.3 Kuaterniyonlar

Tanım 2.3.1: Sanal bileşenleri olarak $\mathbf{i}, \mathbf{j}, \mathbf{k}$ vektörlerinin

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1; \mathbf{ij} = \mathbf{k} = -\mathbf{ji}; \mathbf{jk} = \mathbf{i} = -\mathbf{kj}; \mathbf{ki} = \mathbf{j} = -\mathbf{ik};$$

koşullarını sağladığı bilindiğine göre bu vektörlerin birleşimi olan

$$q = w + \mathbf{i}x + \mathbf{j}y + \mathbf{k}z$$

dörtlü yapısına “**kuaterniyon**” denir (Hamilton, 1844).

İki $q_0 = w_0 + x_0\mathbf{i} + y_0\mathbf{j} + z_0\mathbf{k}$ ve $q_1 = w_1 + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$ kuaterniyonu verilsin. Kuaterniyon cebri aşağıdaki gibi tanımlanmaktadır:

Tanım 2.3.2: q_0 ve q_1 kuaterniyonlarının **toplamı ve farkı**

$$\begin{aligned} q_0 \pm q_1 &= (w_0 + x_0\mathbf{i} + y_0\mathbf{j} + z_0\mathbf{k}) \pm (w_1 + x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}) \\ &= (w_0 \pm w_1) + (x_0 \pm x_1)\mathbf{i} + (y_0 \pm y_1)\mathbf{j} + (z_0 \pm z_1)\mathbf{k} \end{aligned}$$

ile tanımlıdır.

Tanım 2.3.3: q_0 ve q_1 kuaterniyonlarının “**çarpımı**”

$$\begin{aligned} q_0 q_1 &= (w_0 + x_0 \mathbf{i} + y_0 \mathbf{j} + z_0 \mathbf{k})(w_1 + x_1 \mathbf{i} + y_1 \mathbf{j} + z_1 \mathbf{k}) \\ &= (w_0 w_1 - x_0 x_1 - y_0 y_1 - z_0 z_1) + \\ &\quad (w_0 x_1 + x_0 w_1 + y_0 z_1 - z_0 y_1) \mathbf{i} + \\ &\quad (w_0 y_1 - x_0 z_1 + y_0 w_1 + z_0 x_1) \mathbf{j} + \\ &\quad (w_0 z_1 + x_0 y_1 - y_0 x_1 + z_0 w_1) \mathbf{k} \end{aligned}$$

olarak elde edilir.

Tanım 2.3.4: $q = w + xi + yj + zk$ kuaterniyonunun “**eşleniği**”

$$q^* = (w + xi + yj + zk)^* = w - xi - yj - zk$$

ile verilir.

Tanım 2.3.5: $q = w + xi + yj + zk$ kuaterniyonunun “**normu**”

$$N(q) = N(w + xi + yj + zk) = w^2 + x^2 + y^2 + z^2$$

şeklinde tüm bileşenlerinin karelerinin toplamına eşittir.

Tanım 2.3.6: Bir q kuaterniyonunun “**tersi**” $q^{-1} = q/N(q)$ ile bulunur (Shoemaker, 1987).

Tanım 2.3.7: $q \neq 0$ olmak üzere p kuaterniyonunu bir q kuaterniyonuna bölmek için kuaterniyonlarda değişme özelliği olmadığından p yi q^{-1} ile çarpmak gerekir.

$$r_1 = pq^{-1}$$

çarpımına p nin q ile “**soldan bölümü**” denir. Benzer şekilde sağdan bölümü de tanımlamak mümkündür.

Tanım 2.3.8: $q_0 = \cos \theta + \vec{S}_0 \sin \theta$ bir birim kuaterniyon, \vec{S}_0 birim vektör olmak üzere, q_0 birim kuaterniyonuna “**kuaterniyon operatörü**” denir. Böylece bir q_0 kuaterniyon operatörü ile \vec{S}_0 eksenini etrafında θ açısı kadar döndürme hareketi sağlanır (Yang, 1963).

2.4 Kuaterniyon Cebri için Bir Uygulama Çalışması:

Bu çalışma için kuaterniyon cebirindeki tüm cebirsel işlemler C++ programında ile yazılarak kuaterniyon hesap makinesi kodları oluşturulmuştur, (Bkz. Ek1). Sonrasında Hesap makinesinin görseelliğini artırmak için C# programından kullanılmıştır. Programı paket program tarzında hazırlayarak kurulum aşamasını daha kolaylaştırmak için ve ayrıca C# programını yükleme gereği duymadan doğrudan bilgisayara indirebilmek için hesap makinesinin kendi kurulumu oluşturulmuştur. Kuaterniyon Cebri Hesap Makinesi Programını bilgisayara yükledikten sonra masaüstünde Şekil 2.1 deki klasör simgesi görülmektedir.



Şekil 2.1 Kuaterniyon Cebri Hesap Makinesi Programının klasör simgesi

Şekil 2.2 Hesap Makinesinin Veri Girme Sayfası

Klasörü tıklayınca Şekil 2.2 deki gibi hesap makinesinin veri girme sayfası açılır. Veri girme sayfasında kuaterniyonun bileşenlerinin girilmesi, skalalarla çarpım, iki kuaterniyonun toplamı, farkı ve çarpımını hesaplattıran butonlar vardır. Böylelikle kuaterniyonlarla yapılan geometrik işlemler daha hızlı ve güvenilir bir şekilde hesaplatılabilecektir. Aynı zamanda hesap makinesinde bir vektörün kuaterniyon yardımıyla döndürülmesini ve döndürülmüş vektörün bileşenlerini hesaplamak mümkündür. Şimdi hesap makinesindeki her bir butonun işleyişini örneklerle görelim:

Örnek 2.4.1: $q = 10 + 12\vec{e}_1 + 15\vec{e}_2 + 20\vec{e}_3$ kuaterniyonunu hazırladığımız hesap makinesinde bileşenleri girildikten sonra bileşenleri bazları ile birlikte bir mesaj kutusu içinde versin. Sonra girilen kuaterniyonun tersi, normu ve eşleniği hesaplatılsın (Bkz.Şek. 2.3).

(a)

(b)

(c)

(d)

(e)

Şekil 2.3 Verilen Kuaterniyonun (a) bileşenlerini klavyeden girme

(b) bileşenlerinin bazlarla yazılışı (c) tersi (d) normu (e) eşleniği

Örnek 2.4.2: $q_1 = 1 + 3\vec{e}_1 + 5\vec{e}_2 + 7\vec{e}_3$ ve $q_2 = 2 + 4\vec{e}_1 + 6\vec{e}_2 + 8\vec{e}_3$

kuaterniyonlarının kuaterniyon çarpımı hesaplatılsın (Bkz. Şek. 2.4).



(a)

(b)

Şekil 2.4 Verilen kuaterniyonların (a) bileşenlerinin klavyeden girilişi (b) çarpımın sonucu

Örnek 2.4.3: $q_1 = 1 + 2\vec{e}_1 + 3\vec{e}_2 + 4\vec{e}_3$ kuaterniyonunun 5. dereceden kuvveti alınsın (Bkz. Şek. 2.5).

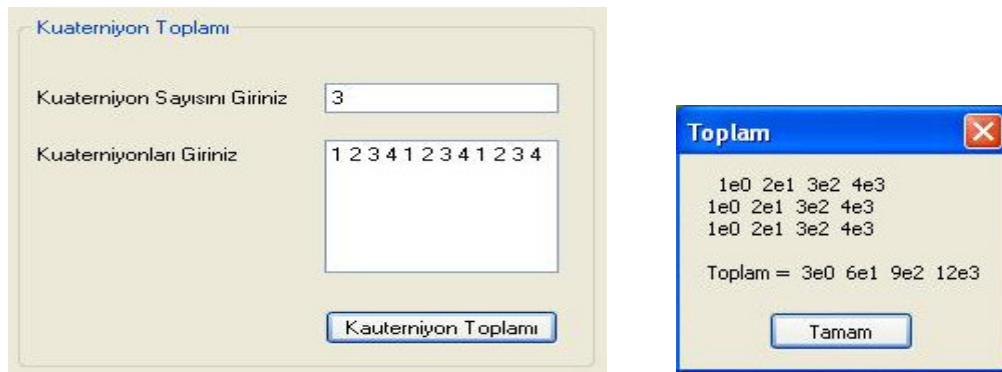


(a)

(b)

Şekil 2.5 Verilen kuaterniyonun (a) bileşenlerinin klavyeden girilişi (b) üssü

Örnek 2.4.4: $q_1 = 1 + 2\vec{e}_1 + 3\vec{e}_2 + 4\vec{e}_3$, $q_2 = 1 + 2\vec{e}_1 + 3\vec{e}_2 + 4\vec{e}_3$ ve ayrıca $q_3 = 1 + 2\vec{e}_1 + 3\vec{e}_2 + 4\vec{e}_3$ kuaterniyonlarının toplamı hesaplatılsın (Bkz. Şek. 2.6).



(a)

(b)

Şekil 2.6 Verilen kuaterniyonun (a) bileşenlerinin klavyeden girilişi (b) toplamı

Örnek 2.4.5: $q_1 = 2 + 10\vec{e}_1 + 3\vec{e}_2 + 14\vec{e}_3$ kuaterniyonu 5 skalerli ile çarpılsın (Şekil 2.7).

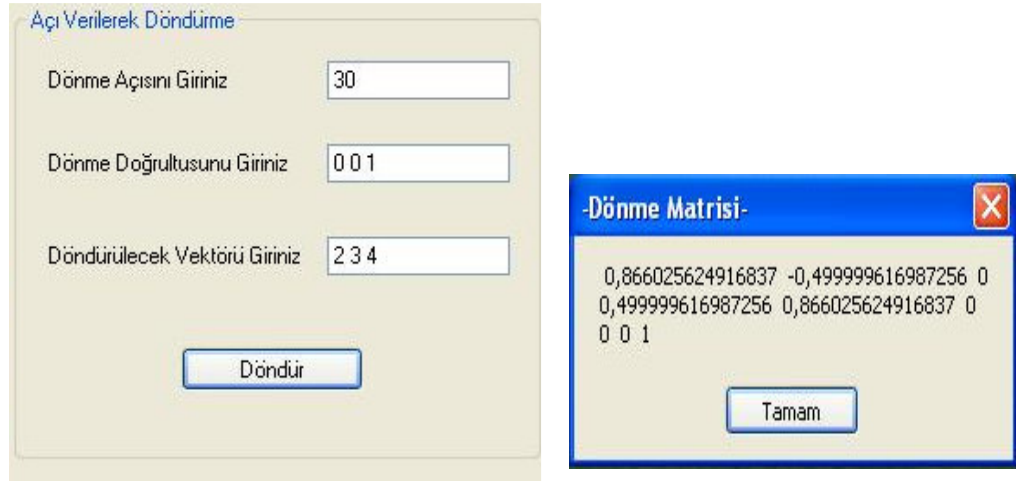


(a)

(b)

Şekil 2.7 Verilen kuaterniyonun (a) bileşeninin klavyeden girilişi (b) skalerle çarpımı

Örnek 2.4.6: Dönme açısı $\theta = 30$ dönme doğrultusu $\vec{S}^* = (0,0,1)$ olmak üzere $q^* = \cos \theta + \vec{S}^* \sin \theta$ kuaterniyonu ile (2,3,4) vektörünü döndürerek döndürülmüş vektörün koordinatları hesaplatılsın (Bkz. Şek. 2.8).



(a)

(b)

(c)

Şekil 2.8 Verilen kuaterniyonun (a) dönme açısının klavyeden girilişi

(b) dönme matrisinin hesabı (c) vektörün döndürülmüş hali

2.5 Dual Kuaterniyonlar

Tanım 2.5.1: Dual vektörlere benzer şekilde iki reel kuaterniyon

$$\begin{cases} q = d + a\vec{e}_1 + b\vec{e}_2 + c\vec{e}_3 \\ q^* = d^* + a^*\vec{e}_1 + b^*\vec{e}_2 + c^*\vec{e}_3 \end{cases}$$

olmak üzere $Q = q + \varepsilon q^*$, $\varepsilon^2 = 0$ şeklinde tanımlanan ya da

$$D = d + \varepsilon d^*, \quad A = a + \varepsilon a^*, \quad B = b + \varepsilon b^*, \quad C = c + \varepsilon c^*$$

dual sayıları için

$$Q = D + A\vec{e}_1 + B\vec{e}_2 + C\vec{e}_3$$

biçiminde yazılan dördlüye “**dual kuaterniyon**” denir.

Dual kuaterniyonlar cebri aşağıda verilmiştir:

Tanım 2.5.2: İki “**dual kuaterniyonun toplamı**” karşılıklı olarak bu iki kuaterniyonun reel ve dual kısımlarının toplamıyla elde edilir. Aynı kaide fark için de vardır.

Tanım 2.5.3: İki dual kuaterniyon

$$\begin{cases} Q = q + \varepsilon q^* \\ P = p + \varepsilon p^* \end{cases}$$

olsun. İki “**dual kuaterniyonun çarpımı**”

$$Q \times P = q \times p + \varepsilon(q \times p^* + q^* \times p)$$

$$P \times Q = p \times q + \varepsilon(p \times q^* + p^* \times q)$$

ile verilir. Dual kuaterniyon çarpımları, genel olarak komutatif değildir.

Tanım 2.5.4: Bir $Q = q + \varepsilon q^*$ “**dual kuaterniyonunun eşleniği**”

$$K_Q = K_q + \varepsilon K_{q^*} = D - A\vec{e}_1 - B\vec{e}_2 - C\vec{e}_3$$

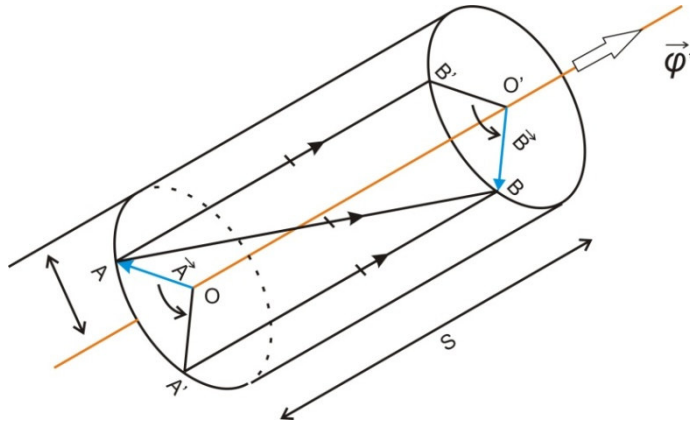
olarak tanımlanır.

Tanım 2.5.5: $Q \times (K_Q)$ çarpımına Q “**dual kuaterniyoun normu**” denir ve N_Q ile gösterilir.

Tanım 2.5.6: Bir Q “**dual kuaterniyonunun tersi**” $Q^{-1} = Q/N(Q)$ ile tanımlanır.

Tanım 2.5.7: $Q \neq 0$ olmak üzere P kuaterniyonunu bir Q kuaterniyonuna bölmek için kuaterniyonlarda değişme özelliği olmadığından P yi Q^{-1} ile çarpmak gerekir. $R_1 = PQ^{-1}$ çarpımına P nin Q ile soldan “**bölümü**” denir. Benzer şekilde sağdan bölümü de tanımlamak mümkündür.

Tanım 2.5.8: \vec{A} ve \vec{B} iki birim dual vektör olsun (Bkz. Bölüm 5). \vec{A} ve \vec{B} birim dual vektörlerinin arasındaki dual açı $\Theta = \theta + \varepsilon\theta^*$ olsun. $\theta^* = 0$ ise oluşan $Q^* = \cos \Theta + \vec{\varphi}^* \sin \Theta$ dual kuaterniyonuna “**dönme operatörü**”, $\theta = 0$ iken oluşan dual kuaterniyona “**kayma operatörü**” ve $\theta \neq 0, \theta^* \neq 0$ durumunda oluşan dual kuaterniyona ise “**vida operatörü**” denir (Yang, 1963)(Bkz Şek.2.9).



Şekil 2.9 Vida hareketi (Hacısalıhoğlu, 1983).

3. ÖKLİD UZAYI

3.1 Temel Kavramları

Tanım 3.1.1: Öklid geometrisine göre iki $A(x, y)$ ve $A_1(x_1, y_1)$ noktaları arasındaki “uzaklık” $d_{AA_1} = \sqrt{(x_1 - x)^2 + (y_1 - y)^2}$ formülü ile tanımlanır.

Tanım 3.1.2: l ve l_1 doğrularının denklemleri $y = kx + s$ ve $y_1 = k_1x + s_1$ dir ve doğrular arasındaki δ_{ll_1} açısı için $\tan \delta_{ll_1} = \frac{k_1 - k}{kk_1 + 1}$ formülüne “iki doğru arasındaki açının tanjantı” denir.

Tanım 3.1.3: l ve l_1 doğrularının x -ekseni ile yaptıkları açılar φ ve φ_1 olmak üzere; $\tan \varphi = k, \tan \varphi_1 = k_1$ için

$$\tan \delta_{ll_1} = \tan(\varphi_1 - \varphi) = \frac{\tan \varphi_1 - \tan \varphi}{1 + \tan \varphi_1 \cdot \tan \varphi} = \frac{k_1 - k}{kk_1 + 1}$$

formülüne “iki doğru arasındaki açı” denir. Eğer $k = k_1$ eğimleri eşit ise l ve l_1 birbirine “paraleldir” denir ve doğrular arasındaki δ_{ll_1} açısı sıfır olur.

Tanım 3.1.4: $|s_1 - s|$ ifadesi l ve l_1 doğruları arasındaki y ekseni üzerindeki doğru parçasının uzunluğu olmak üzere

$$d_{ll_1} = \frac{|s_1 - s|}{\sqrt{k^2 + 1}}$$

uzaklığına “paralel doğrular arasındaki uzaklık” denir. Her bir l ve l_1 doğrularının x ekseni ile yaptığı açı φ olmak üzere

$$\sqrt{k^2 + 1} = \sqrt{\tan^2 \varphi + 1} = \frac{1}{\cos \varphi}$$

olur. Dikkat edilirse δ_{ll_1} ve d_{ll_1} lerin nitelikleri farklıdır. δ_{ll_1} açısal ölçü birimi d_{ll_1} ise uzaklık birimidir (Yaglom, 1979).

3.2 Öklid Uzayında Mekanik Bir Uygulama

Bu çalışmada, engebeli arazi aracının uygun arazi modellenmesi için, Bilgisayar Destekli Geometrik Tasarım (BDGT) teknikleri ile yeni bir yaklaşım sunulacaktır. Arazi yüzeylerinde üç boyutlu kübik bir eğri, test yolu olarak alınmış ve tekerlek modellemesi için iki farklı geometrik metod verilmiştir: Birinci metotta tekerlekler oskülatör çemberler olarak düşünülmüş ve ikinci metotta ise tekerlekler oskülatör küre olarak alınmıştır.

3.2.1 Önbilgiler ve Tarihçe:

Süspansiyon sistemi aracın tekerlek sistemiyle ilgili bir mekanizmadır. Süspansiyon sistemi oluşturulurken bilmemiz gereken en önemli şey engebeli arazilerde yol alırken ani ve değişen pozisyonlardan onu nasıl koruyacağımızdır, (Kilit, 2005). Bu nedenle, birçok tekerlek çeşidi, süspansiyon sistemini dizayn etmek için yüzyıllardır üretilmeye devam edilmiştir.

Birleşik devletler “rocker-bogie” tasarımı ile ilgilenerek 1997 de Mars Pathfinder “Sojourner” aracı inşa edilmişti, (Patel et al., 2002).

“Bogie” terimi eski demir yolu sisteminden gelir. Bogie, altı tekerleklerden oluşan ve eğrilen ray boyunca mil etrafında dönen trenin alt takımıdır. “Rocker” terimi de diferensiyel tasarımından gelir. Rocker tekerlekle araç gövdesini çeşitli pozisyonlarda sallanmaya bağlı olarak dengede kalmasını sağlar (Thianwiboon et al., 2001). Bu yüzden engebeli arazi modellemeleri ve araç tekerlek yapıları geometrik metodlar kullanılarak dizayn edilmelidir.

Son çalışmalarda, engebeli arazi için iki boyutlu üçüncü mertebeden eğri kullanılmış ve tekerlekler bu mekanizmanın kinetik özellikleri kullanılarak elde edilmiştir (Kilit, 2005).

Ancak bu çalışmalarda yol yapısının iki boyutlu olduğu varsayıldığından bu çözümlerin, gerçekçi üç boyutlu uzay görüntüsü için yetersiz kaldığı görülmektedir.

Bu çalışmada engebeli araziler için BDGT da kullanılan üç boyutlu üçüncü mertebeden parametrik kübik eğrilerin interpolasyon teknikleri kullanılmıştır. Rocker-bogie gövdesindeki tekerleklerin süspansiyon mekanizması, oskülatör

çember ve oskülatör küreler ile 3 boyutlu uzayda diferensiyel geometrik yöntemler kullanılmasıyla elde edilmiştir (Goetz, 1970; Heinrich, 1963; Gray, 1997).

3.2 .2 Engebeli arazi için BDGT interpolasyon metodu

$\vec{\beta}(t)$, eğri üzerindeki bir nokta olmak üzere bir parametrik kübik eğri

$$\vec{\beta}(t) = \sum_{i=0}^3 \vec{a}_i t^i \quad 0 \leq t \leq 1, \quad (3.1)$$

ile tanımlanır. (3.1) denkleminin açık hali

$$\vec{\beta}(t) = \vec{a}_3 t^3 + \vec{a}_2 t^2 + \vec{a}_1 t^1 + \vec{a}_0 \quad 0 \leq t \leq 1 \quad (3.2)$$

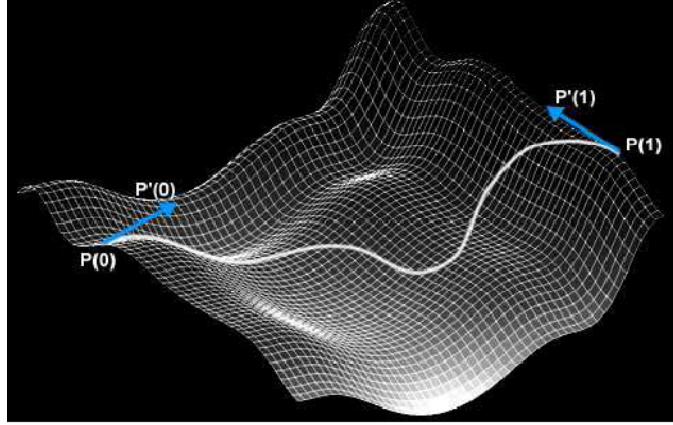
dir. Bu eşitlik $\vec{\beta}(t)$ nin üç bileşenine

$$x(t) = a_{3x} t^3 + a_{2x} t^2 + a_{1x} t^1 + a_{0x}$$

$$y(t) = a_{3y} t^3 + a_{2y} t^2 + a_{1y} t^1 + a_{0y}$$

$$z(t) = a_{3z} t^3 + a_{2z} t^2 + a_{1z} t^1 + a_{0z}$$

biçiminde ayrılabilir. Yukarıdaki denklemi çözebilmek için, bilinmeyen on iki katsayı, cebirsel katsayı olarak bilinmeli ve belirtilmelidir. Değişken değerli eğrinin şeklini kontrol etmek zordur. Daha iyi bir yaklaşım için geometrik kısıtlamaları sağlayan uygun sınır şartları kurulmalıdır. Sınır şartları oniki cebirsel katsayının hesaplanması için gerekli olan oniki denklemi kurmaya uygun olmalıdır. Herbir bölümün bilinen uç noktalarının koordinatları ihtiyaç duyulan oniki denklemin altısından elde edilir. Diğer altı denklem, uç noktalardaki iki teğet vektörün bulunmasıyla elde edilir. Teğet vektörlerinin doğrultuları, eğrilerin uç noktalarındaki eğimleriyle bulunur. Uç noktalarındaki teğet vektörlerinin büyüklüklerindeki değişiklikler eğrinin şeklinin değişimini gösterir (Bkz. Şekil 3.1).



Şekil 3.1 Kübik splaynın uç noktaları ve teğet vektörleri

Bir kübik eğriyi tanımlamak için uç noktalarının ve teğet vektörlerinin kullanılması işlemi bir Hermit interpolasyonu formundadır. Herbir kübik eğri parçası 0 dan 1 e parametrelenir, böylece iki uç noktası, t parametrik değişkeninin limit değeri olan $\vec{\beta}(0)$ ve $\vec{\beta}(1)$ ye karşılık gelir. (3.2) denkleminde $t = 0$ ve $t = 1$ yerine konulursa, iki uç noktalarının vektörleri ve

$$\vec{\beta}(0) = \vec{a}_0 \quad \vec{\beta}(1) = \vec{a}_3 + \vec{a}_2 + \vec{a}_1 + \vec{a}_0$$

cebirsal katsayıları arasındaki ilişkisi bulunmuş olur. Teğet vektörlerini bulabilmek için (3.2) denkleminde t ye göre diferansiyellenebilmesiyle

$$\vec{\beta}'(0) = 3a_{3z}t^2 + 2a_{2z}t + a_1$$

bağıntısı bulunur. Uç noktalarındaki teğet vektörleri $t=0$ ve $t=1$ yerine konulmasıyla

$$\vec{\beta}'(0) = \vec{a}_1 \quad \vec{\beta}'(1) = 3\vec{a}_3 + 2\vec{a}_2 + \vec{a}_1$$

bulunur. (3.2) denklemindeki a_i cebirsal katsayısı, uç noktalarının ve teğet vektörlerinin sınır şartları açıkça

$$\vec{a}_0 = \vec{\beta}(1) \quad \vec{a}_1 = \vec{\beta}'(0)$$

$$\vec{a}_2 = -3\vec{\beta}(0) + 3\vec{\beta}(1) - 2\vec{\beta}'(0) - \vec{\beta}'(1)$$

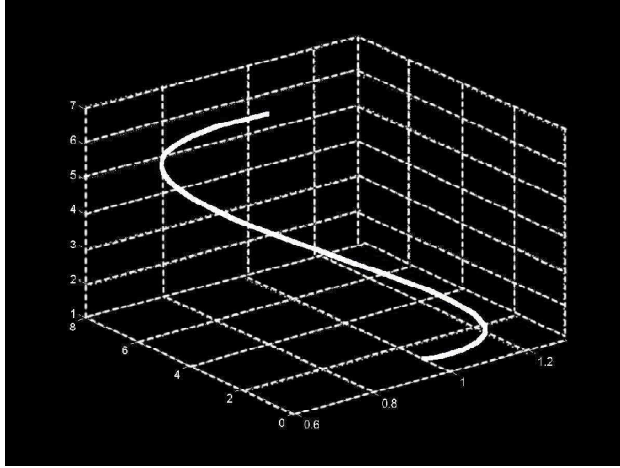
$$\vec{a}_3 = 2\vec{\beta}(0) - 2\vec{\beta}(1) + \vec{\beta}'(0) + \vec{\beta}'(1)$$

şeklinde yazılabilir. \vec{a}_i nin bu değerlerini (3.2) denkleminde yerine yazıp tekrar düzenlediğimizde

$$\vec{\beta}(t) = (2t^3 - 3t^2 + 1)\vec{\beta}(0) + (-2t^3 + 3t^2)\vec{\beta}(1) + (t^3 - 2t^2 + t)\vec{\beta}'(0) + (t^3 - t^2)\vec{\beta}'(1)$$

bağıntısı bulunur. $\vec{\beta}(0)$, $\vec{\beta}(1)$, $\vec{\beta}'(0)$ ve $\vec{\beta}'(1)$ değerleri geometrik katsayılar olarak adlandırılırlar ve bilinen vektörlerin niteliklerini gösterirler. Bu vektör büyüklüklerinin polinom katsayıları genellikle “**blending fonksiyonları**” olarak bilinirler. T parametresini bu blending fonksiyonlarda 0 dan 1 ye kadar değiştirerek eğri parçasının üzerinde çeşitli noktalar bulunabilir (David, 1990).

Örnek 3.2.1: Burada BDGT kullanılmasıyla bir kübik splayn modellenmesinin bir bilgisayar programı vardır (Bkz. Şekil 3.2).



Şekil 3.2 Kübik splayn modellenmesi

3.2.3 Yeni bir tekerlek yapısı için bazı diferensiyel geometrik yaklaşımlar:

Arazi aracının engebeli zeminde daha kararlı hareket etmesi için tekerlek süspansiyon mekanizması, pürüzlü zeminle güçlü bir temas kurmalıdır. Bu sebepten temas problemini çözebilmek için birçok metod oluşturulmuştur. Burada iki tane tekerlek modelleme tekniği sunulmuş ve geometrik açıdan ele alınarak incelenmiştir. Birinci metod üç boyutlu uzayda tekerlekleri oskülatör çember olarak gösterir. İkinci metotta bir küresel tekerlek yapısı kullanılmış ve bu düşünce üç boyutlu kübik splayn üzerinde uygulanmıştır.

3.2.3.1 Üç boyutlu uzayda çemberlerle araç tekerleklerinin modellenmesi

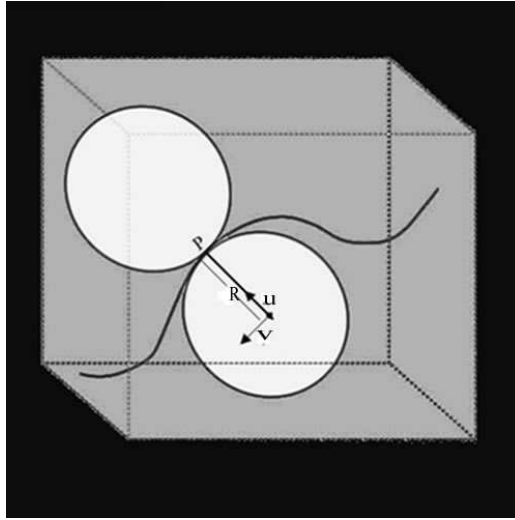
Bu bölümde üç boyutlu uzayda çember tekerleğinin yeni bir geometrik metodunu sunacağız. Bir arazi zeminini aşağıdaki gibi tanımlayabiliriz:

$$\vec{\beta}(t) = (x_i(t), y_i(t), z_i(t)) = \left(\sum_{j=1}^3 \sum_{i=0}^3 a_i t^i e_j \right).$$

Şekil 3.3 den görülüyor ki (β) kübik splayn eğrisi (x_0, y_0, z_0) değme noktasında tekerleğe değ. Üç boyutlu uzayda bir P çemberi

$$\vec{P} = \vec{C} + R_1 \cos \alpha \vec{U} + R_1 \sin \alpha \vec{V}$$

denklemleri ile gösterilir. \vec{V} vektörü düzlemin birim normal vektörü, C çemberin merkezi, R_1 yarıçap, U vektörü C den çember üzerindeki bir noktaya olan bir birim vektör olsun. α , x-ekseni ve (x_0, y_0, z_0) noktasındaki eğrinin teğeti arasındaki açıdır. P eğri üzerinde bir noktadır (Bkz. Şekil 3.3).



Şekil 3.3 Üç boyutlu uzayda bir çember

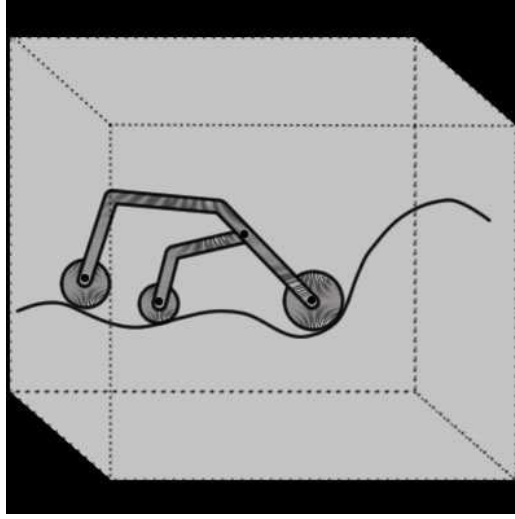
Bu denklemde P noktasının değme noktası olarak alındığında

$$\vec{P} = (x_0, y_0, z_0)$$

olur. Tekerleğin konumunu elde etmek için aşağıdaki denklem ile verilen üç boyutlu çemberin merkezine ihtiyacımız vardır:

$$\begin{aligned}\vec{C} &= \vec{P} - R_1 \cos \alpha \vec{U} - R_1 \sin \alpha \vec{V} \\ &= (x_0, y_0, z_0) - R_1 \cos \alpha \vec{U} - R_1 \sin \alpha \vec{V}\end{aligned}\quad (3.3)$$

Tekerleği bu yolla tanımlayabileceğimiz aşıkardır. Çünkü tekerlek merkezi Şekil 3.4 de görüldüğü gibi eğri boyunca tüm kübik splayn koordinatlarının hesaplanması ile bulunabilir.

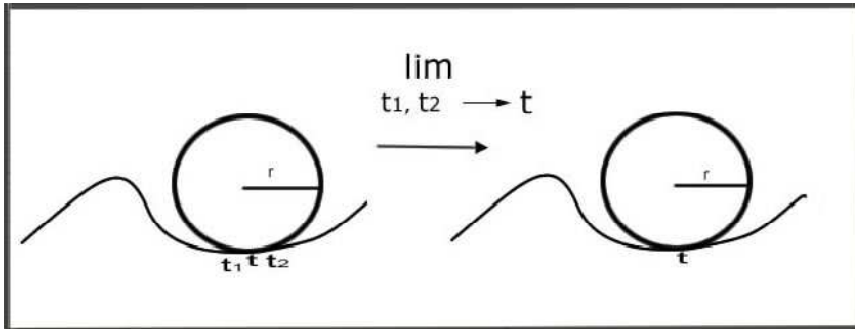


Şekil 3.4 Üç boyutlu uzaydaki Çemberlerle Tekerlek Modellemesi

$P(t, t_1, t_2)$ çemberinin $\vec{\beta}(t) = \sum_{i=0}^3 \vec{a}_i t^i$ eğrisini t, t_1, t_2 gibi üç noktada kesitiğini düşünelim. Bu iki t_1, t_2 noktalar eğri üzerindeki her t noktasına sonsuzda yakınsaktır. Böylece $P(t, t_1, t_2)$ çemberinin limiti t noktasında “**Oskülatör çember**” olarak adlandırılır ve

$$\lim_{t_1, t_2 \rightarrow t} P(t, t_1, t_2)$$

formülü ile gösterilir.

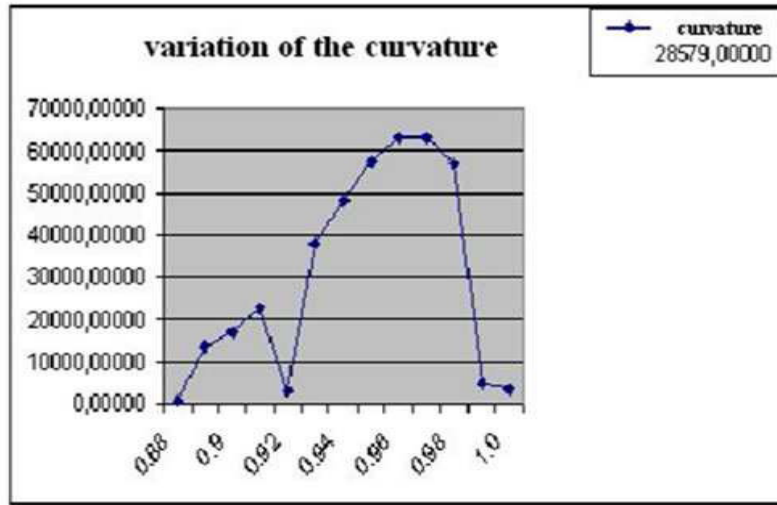


Şekil 3.5 Üç boyutlu uzayda oskülatör çember

Öyleyse R_1 oskülör çemberinin yarıçap formülünün kullanılmasıyla aşağıdaki denklemi tanımlayabileceğimiz görülür (Bkz. Şekil 3.5).

$$R_1 = \frac{(\overleftarrow{\beta^2})^{\frac{3}{2}}}{\sqrt{(\overleftarrow{\beta} \times \overleftarrow{\beta})}} = \frac{1}{k_1}.$$

Burada k_1 , (β) eğrisinin P noktasındaki eğriliğidir. Şekil 3.6 daki kübik splayn üzerinde eğriliklerin değişimlerinin bir örneği yer almaktadır.



Şekil 3.6 Bir kübik splayn üzerindeki eğrilik değişimlerinin bir örneği

3.2.3.2 Üç boyutlu uzayda araç tekerleğinin küreyle modellenmesi

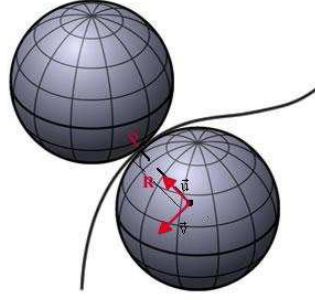
Bu bölümde, araç için geometrik tasarım metodlarını kullanan yeni bir bogie sistemi yapılandırılmaya çalışılmıştır. Daha önce açıklandığı gibi, bu çalışmada, daha fazla gerçeklik alabilmek için üç boyutlu uzayda tekerlek-yol değme modelini hesaba kattık. Üç boyutlu uzayda $S(t, t_1, t_2, t_3)$ küresini tekerlek eğrisi ve arazi yolunu ise yine $\overleftarrow{\beta}(t) = \sum_{i=0}^3 a_i t^i$ $0 \leq t < 1$, üç boyutlu parametrik kübik eğrisi olarak düşündük. Burada yeni bir bogie sistem için

$$x_c = R_1 \cos \alpha$$

$$y_c = R_1 \sin \alpha$$

$$z_c = \sqrt{R_2^2 - R_1^2}$$

gibi özel bir küre denklemi kullanılmıştır (Bkz. Şekil 3.7).



Şekil 3.7 Üç boyutlu uzayda bir küre

Burada, tanımlanan kürenin merkezi (x_c, y_c, z_c) , yarıçapı R_2 ve (3.3) denklemindeki oskülatör çemberinin yarıçapı R_1 dir. Küre merkezinin, (x_0, y_0, z_0) noktasına ötelenmesiyle merkez noktasının koordinatlarını tekrar biçimlendirirsek

$$x_c = x_0 + R_1 \cos \alpha$$

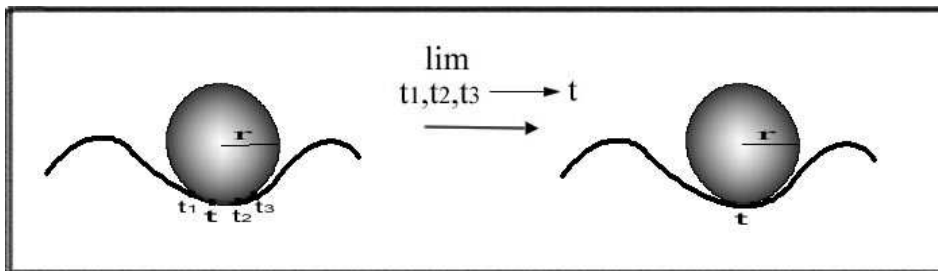
$$y_c = y_0 + R_1 \sin \alpha$$

$$z_c = z_0 + \sqrt{R_2^2 - R_1^2}$$

olur. t_1, t_2, t_3 noktaları parametrik kübik eğri üzerinde herhangi bir t noktasına yakınsadığında $S(t, t_1, t_2, t_3)$ küresinin limitine

$$S = \lim_{t_1, t_2, t_3 \rightarrow t} S(t, t_1, t_2, t_3)$$

t noktasındaki “**Oskülatör küresi**” denir (Bkz. Şekil 3.8).

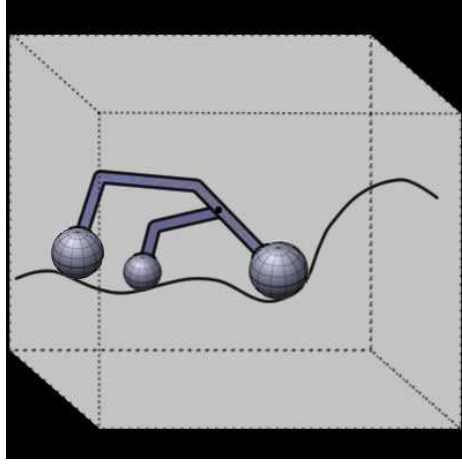


Şekil 3.8 Üç boyutlu uzayda oskülatör küre

Oskülatör küresinin yarıçapı

$$R_2^2 = (R_1)^2 + \left[\frac{1}{k_2} R_1\right]^2$$

denklemden elde edilebilir. Bu denklemdeki k_2 sembolü, (x_0, y_0, z_0) noktasındaki eğrinin burulmasını gösterir. Böylece küresel tekerleğin merkezini ve yarıçapını hesaplamış oluruz, ayrıca küresel tekerlek bogie sisteminin tasarımını gerçekleştirmiş oluruz (Bkz. Şekil 3.9).



Şekil 3.9 Üç boyutlu uzayda küresel tekerleklerle araç modellenmesi

3.2 .4 Sonuçlar

Bu çalışmada engebeli arazi modellemesi için BDGT deki interpolasyon metodu kullanıldı. Bu metodun kullanılmasıyla arazi üzerinde bir yol, uç noktaları ile kolayca kontrol altına alınabilir. Sonra, üç boyutlu arazi üzerindeki bir araç için bazı tekerlek çeşitleri düşünüldü. Yapılandırılan bu tekerlekler diferensiyel geometrik özelliklerle incelendi.

Sonuç olarak; bu metodların, üç boyutlu uzayda BDGT deki kübik splayn interpolasyonları ile bir arazi aracının hareketini geliştirmek için ileri kontrol etmede kullanılabileceği anlaşıldı, (Kusak and Çalışkan, 2009).

3.3 Öklid Uzayında Dönüşümler

Tanım 3.3.1: Bir nesnenin x-eksenine göre veya y-eksenine göre aynı doğrultudaki değişimine “**iki boyutlu öteleme (2D-translation)**” denir. Bir nesnenin üç boyutlu koordinat sisteminde belirtilen tüm noktalarının x, y ve z koordinatlarına sırasıyla Δx , Δy ve Δz öteleme miktarlarının eklenmesine “**Üç Boyutta Öteleme (3D-Translation)**” denir.

Tanım 3.3.2: Ölçekleme işlemi, bir nesnenin boyutunun x ve y eksenleri yönünde değiştirilmesine “**iki boyutlu ölçekleme (2D-Scaling)**” denir. Ölçekleme işlemi, bir nesnenin boyutunun x,y,z eksenleri yönünde değiştirilmesine “**üç boyutlu ölçekleme (3D-Scaling)**” denir .

Tanım 3.3.3: y ekseninin x eksenine ile sağa doğru meyilli hale getirilmesine “**iki boyutlu aksel meyillendirme (Shearing)**”denir (Bkz.Bölüm 7).

Tanım 3.3.4: Bir nesneyi tanımlayan noktaların bir referans noktası etrafında bir β açısı kadar döndürülmesine “**iki boyutlu aksel döndürme(Rotating)**” denir. Nesnenin x,y,z –eksenleri etrafında belirli açılar altında döndürülmesine “**üç boyutlu aksel döndürme (Rotating)**” denir.

Tanım 3.3.5: Nesnelerin simetrik görüntülerine “**iki boyutlu yansıma (Reflection)**” denir. Nesnenin x, y, z–eksenlerine göre simetrik görüntüsünü almaya “**üç boyutlu yansıma (Reflection)**”denir, (Rogers et al.,1989).

3.4 Öklid Düzlem Dönüşümlerinin Bir Kinematik Uygulaması

Bu çalışmada, iki metal düzlem levhadan, biri sabit iken diğ erinin shear hareketi ile esnemesi sonucu oluşan düzlem hareketi incelenmektedir. Düzlem hareketlerinde, genel olarak, iki düzlem birbirinin aynıdır. Hareketli düzlem üzerinde alınan sabit noktanın hareketi, hem dönme, hem öteleme dönüşümleri ile tanımlanmaktadır. Ancak düzlemlerin birbirine göre hareketleri incelenirken deformasyonlar gözönünde bulundurulmamaktadır. Burada levhalardan birinin sabit, diğ erinin fiziksel bir etki ile deformasyona uğ ratılmış olduğunu düşünüyörüz. Düzlemdeki bu deformasyonu shear hareketi ile göstereceğ iz. Böylece hareket sonucu oluşan hız bağıntılarını ve pol eğrisini hesaplayacağız.

Geometrik tasarımda nesnelerin hareketleri için dönme, öteleme, ölçekleme ve shearing gibi çeşitli dönüşümler kullanılmaktadır, (Rogers et al.,1989). Kinematikte düzlem hareketini, koordinat sisteminden dolayı, dönme ve öteleme dönüşümleri ile tanımlarız, (Müller, 1963).

Kinematikte, düzlemsel harekette kullanılan düzlemler birebir aynıdır. Teorik olarak bu mümkündür ancak bazı fiziksel koşullarda düzlemin yapısında bir deformasyon söz konusu olabilir. Düzlem bir metal levha olarak alınırsa, ısıdan dolayı bir deformasyon olabilir, ya da düzlem plastik bir yapıda ise esnemeneden kaynaklanan deformasyon alınabilir. Biz çalışmamızda oluşabilecek bu deformasyonları shearing (kırpma-kaykılma) hareketi ile geometrik olarak anlamlandırıyoruz. Artık burada hareketimizi shearing ve öteleme ile tanımlıyoruz. Böylelikle düzlemsel harekete, bu anlamda yöntem olarak kinematiğ e başka bir açıdan bakmış oluyoruz.

Düzlem hareketinde baz vektörlerinin dönme hareketini

$$\vec{e}_1 = \vec{e}'_1 \cos \varphi + \vec{e}'_2 \sin \varphi$$

$$\vec{e}_2 = -\vec{e}'_1 \sin \varphi + \vec{e}'_2 \cos \varphi$$

denklemleri(Bkz. Tanım 3.3.4), düzlemler arasındaki ötelemeyi(Bkz. Tanım 3.3.1) ise

$$\vec{u} = \vec{e}_1 u_1 + \vec{e}_2 u_2$$

denklemlerle gösteriyoruz. (x, y) noktasına shear dönüşümü uygularsak

$$(x', y') = (x, ax + y)$$

yazılabilir(Bkz. Tanım 3.3.3 ve Bölüm 6). Burada a sayısı açığı verir. Çalışmamızda düzlemin baz vektörlerini shear dönüşümü altında dönüştürüyoruz. Sonra iki düzlemin arasındaki öteleme vektörünü tanımlayarak hareketimizi shearing ve öteleme hareketiyle oluşturuyoruz. Hareketimizde E hareketli düzleminin ortogonal olmayan birim çatisı $\{0, e_1, e_2\}$ olsun. E' sabit düzleminin ortogonal çatisını $\{0', e'_1, e'_2\}$ alalım. Bu iki koordinat sistemini E ve E' düzlemleri ile temsil edecek ve hareketli koordinat sisteminin sabit koordinat sistemine göre bir parametrelili hareketini ele alacağız, ve böylece çalışmamızda düzlemsel hareketi, öteleme ve shearing hareketleri ile tanımlamış olacağız. Hareketli sistemin başlangıç noktasından sabit sistemin başlangıç noktasına giden öteleme vektörü

$$\overrightarrow{OO'} = \vec{u} = \vec{e}_1 u_1 + \vec{e}_2 u_2$$

şeklinde. Her iki koordinat sisteminin birbirine göre yaptıkları açığı shearing katsayısı olarak alalım. Bu durumda hareketli düzlemin birim baz vektörlerini shearing dönüşümü ile aşağıdaki gibi verebiliriz:

$$\begin{aligned} \vec{e}_1 &= \vec{e}'_1 \\ \vec{e}_2 &= \vec{e}'_1 \frac{\varphi}{\sqrt{1+\varphi^2}} + \vec{e}'_2 \frac{1}{\sqrt{1+\varphi^2}} \end{aligned} \quad (3.4)$$

Tanım 3.4.1: \vec{u} öteleme ve φ shear dönüşümleri altında tanımlanan E ve E' düzlemsel hareketine “**shear dönüşümlü bir parametrelili düzlemsel hareket**” denir ve $H_{S1} = E/E'$ şeklinde gösterilir.

u_1, u_2, φ büyüklüklerini aşağıdaki gibi t (burada t parametresi zaman olarak kabul edilebilir) temel parametresinin yeterli derecede türevlenebilen fonksiyonları olduğunu varsayalım:

$$u_1 = u_1(t) \quad u_2 = u_2(t) \quad \varphi = \varphi(t)$$

fonksiyonları tanımlıdır. Hareketli E düzlemi ve sabit E' düzlemi üzerinde alınan bir X noktasının koordinatları, sırasıyla, (x_1, x_2) ve (x'_1, x'_2) olsun. Bu noktalara karşılık gelen konum vektörleri

$$\vec{X} = \overrightarrow{OX} = \vec{e}_1 x_1 + \vec{e}_2 x_2$$

$$\vec{X}' = \overrightarrow{O'X'} = \vec{e}'_1 x'_1 + \vec{e}'_2 x'_2$$

biçiminde ifade edilebilir. Bu durumda vektörel toplama kuralından

$$\overrightarrow{O'X} = \overrightarrow{O'O} + \overrightarrow{OX} = -\overrightarrow{OO'} + \overrightarrow{OX}$$

$$\vec{x}' = -\vec{u} + \vec{x} = \vec{e}_1(-u_1 + x_1) + \vec{e}_2(-u_2 + x_2). \quad (3.5)$$

bağıntısı elde edilir. X noktasının x_1, x_2 koordinatları t zamanından bağımsız ise yani x_1, x_2 sabit ise X noktasının hareketli E - düzleminde tespit edilmiş olduğunu düşünür veya kısaca X, E nin bir noktası, tersine olarak x'_1, x'_2 sabit sayılar ise X, E' de sabit kalır veya X, E' nin bir noktası olur.

3.4.1 Türev Denklemleri:

Bir X noktasının hem sabit hem hareketli düzleme göre hızlarını araştırmak için hareketin türev denklemlerini elde etmemiz gerekir. Bunun için \vec{e}'_1 ve \vec{e}'_2 vektörlerini sabit kabul ederek (3.4) denklemlerinin t zamanına göre türevleri hesaplanır.

Teorem 3.4.1: Shearing dönüşümü ile tanımlanan $H_{S1} = E/E'$ düzlemsel hareketinin $\{\vec{e}_1, \vec{e}_2\}$ baz vektörlerinin t zamanına göre türevi

$$\dot{\vec{e}}_1 = 0$$

$$\dot{\vec{e}}_2 = \frac{\dot{\phi}}{\sqrt{1+\phi^2}} \vec{e}_1 - \frac{\phi \dot{\phi}}{1+\phi^2} \vec{e}_2$$

şeklindedir.

İspat: E düzleminde t ye göre türev alırken \vec{e}'_1 ve \vec{e}'_2 baz vektörlerini sabit olarak düşünmeliyiz. Bu yüzden türev denklemleri

$$\dot{\vec{e}}_1 = 0 \quad \text{ve} \quad \dot{\vec{e}}_2 = \dot{\varphi} \left(\frac{\vec{e}_1}{\sqrt{1+\varphi^2}} - \frac{\varphi \vec{e}_2}{1+\varphi^2} \right)$$

şeklinde bulunur.

Teorem 3.4.2: $H_{S1} = E/E'$ bir parametrelili düzlemsel hareketin t zamanına göre türevi

$$\dot{\vec{u}} = \vec{e}_1 \left(\dot{u}_1 + \frac{\varphi u_2}{\sqrt{1+\varphi^2}} \right) + \vec{e}_2 \left(\frac{\varphi \dot{\varphi}}{1+\varphi^2} u_2 + \dot{u}_2 \right)$$

olarak elde edilir.

İspat: $\vec{u} = \vec{e}_1 u_1 + \vec{e}_2 u_2$ vektörünün türevini alırken \vec{e}_1 ve \vec{e}_2 vektörleri hareketli düzlemde alındığı için bu vektörler sabit olmayacağından

$$\begin{aligned} \dot{\vec{u}} &= \dot{\vec{e}}_1 u_1 + \vec{e}_1 \dot{u}_1 + \dot{\vec{e}}_2 u_2 + \vec{e}_2 \dot{u}_2 \\ &= 0 \cdot u_1 + \vec{e}_1 \dot{u}_1 + \dot{\varphi} \left(\frac{\vec{e}_1}{\sqrt{1+\varphi^2}} - \frac{\varphi \vec{e}_2}{(1+\varphi^2)} \right) u_2 + \vec{e}_2 \dot{u}_2 \\ &= \vec{e}_1 \left(\dot{u}_1 + \frac{\varphi u_2}{\sqrt{1+\varphi^2}} \right) + \vec{e}_2 \left(\frac{\varphi \dot{\varphi}}{1+\varphi^2} u_2 + \dot{u}_2 \right) \end{aligned}$$

elde edilir.

Tanım 3.4.2: Shear dönüşümü ile yeniden yapılandırılan $H_{S1} = E/E'$ bir parametrelili hareketinin türev denklemleri

$$\begin{aligned} \dot{\vec{e}}_1 &= 0, \quad \dot{\vec{e}}_2 = \frac{\dot{\varphi}}{\sqrt{1+\varphi^2}} \vec{e}_1 - \frac{\varphi \dot{\varphi}}{1+\varphi^2} \vec{e}_2, \\ \dot{\vec{u}} &= \left(\dot{u}_1 + \frac{\varphi u_2}{\sqrt{1+\varphi^2}} \right) \vec{e}_1 + \left(\dot{u}_2 + \frac{\varphi \dot{\varphi}}{1+\varphi^2} u_2 \right) \end{aligned}$$

baz ve ötleme vektörlerinin türevleri ile verilir.

Tanım 3.4.3: Hareketin türev denklemlerinde geçen $\dot{\varphi}$ türevine shearing dönüşümünün “açısal hızı” denir.

3.4.2 Bağıl, Mutlak ve Sürüklenme Hız Vektörlerinin Birleşimi

E düzlemi E' düzlemine göre bir parametrelili bir hareketi gerçekleştirirken bir X noktası da hareketli E düzlemindeki yerini t zamanı ile değiştirsin. Böylece oluşan hareketten bağıl, mutlak ve sürüklenme hız vektörleri olmak üzere üç farklı hız oluşur.

Tanım 3.4.4: E üzerindeki (X) eğrisine ait olan X noktasının E ye göre hız vektörü, X noktasının “**bağıl hız vektörü**” olarak adlandırılır ve \vec{v}_r ile

$$\vec{v}_r = \dot{X} = \vec{e}_1 \dot{x}_1 + \vec{e}_2 \dot{x}_2$$

biçiminde tanımlıdır.

Tanım 3.4.5: E üzerindeki (X') eğrisine ait olan X' noktasının E' ye göre hız vektörü, X' noktasının “**mutlak hız vektörü**” olarak ve \vec{v}_a ile

$$\vec{v}_a = -\dot{u} + \dot{x}$$

biçiminde tanımlıdır.

Tanım 3.4.6: $H_{S1} = E/E'$ düzlemsel hareketinde, E hareketli düzlemi üzerinde X noktası sabit kalırken X noktasının E' düzlemine göre hız vektörüne X noktasının “**sürüklenme hız vektörü**” denir ve \vec{v}_f ile gösterilir.

Teorem 3.4.3: $H_{S1} = E/E'$ hareketinde mutlak hız vektörünü, bağıl hız ve sürüklenme hız vektörlerinin toplamına eşittir, yani $\vec{v}_a = \vec{v}_f + \vec{v}_r$ dir.

İspat: Tanım 3.5.2 ve Tanım 3.5.4 ün kullanılmasıyla (3.5) denkleminin t parametresine göre türevi

$$\begin{aligned} \vec{v}_a &= -\dot{u} + \dot{x} \\ &= -\vec{e}_1 \left(\dot{u}_1 + \frac{\dot{\varphi} u_2}{\sqrt{1+\dot{\varphi}^2}} \right) - \vec{e}_2 \left(\frac{\dot{\varphi} \varphi}{(1+\dot{\varphi}^2)} u_2 + \dot{u}_2 \right) + \dot{\varphi} \left(\frac{\vec{e}_1}{\sqrt{1+\dot{\varphi}^2}} - \frac{\varphi \vec{e}_2}{(1+\dot{\varphi}^2)} \right) x_2 \end{aligned}$$

dir. Tanım 3.5.6 den dolayı $\vec{v}_r = \vec{0}$ olduğunda sürüklenme hız vektörü şöyledir:

$$\vec{v}_f = \vec{e}_1 \left[-\dot{u}_1 - \frac{\dot{\varphi}u_2}{\sqrt{1+\dot{\varphi}^2}} + \frac{\dot{\varphi}x_2}{\sqrt{1+\dot{\varphi}^2}} \right] + \vec{e}_2 \left[\frac{-\dot{\varphi}\varphi}{(1+\varphi^2)}u_2 - \dot{u}_2 - \frac{\dot{\varphi}\varphi x_2}{1+\varphi^2} \right]. \quad (3.6)$$

Not 3.1: X noktası E düzlemi üzerinde sabit kalıyorsa yani $\vec{v}_r = \vec{0}$ ise mutlak hız vektörü

$$\vec{v}_a = \vec{v}_f$$

denkleminde görüldüğü üzere sürüklenme hız vektörüne eşit olur.

3.4.3 Ani Shearing Polü ve Pol Eğrisi

Burada bir parametrelili hareketimizin her t anında sürüklenme hızı sıfır olan noktalarını araştıracağız.

Teorem 3.4.4: Her ani t parametresi için sürüklenme hız vektörü sadece bir noktada sıfır olur, yani $H_{S1} = E/E'$ hareketi üzerinde $\dot{\varphi} \neq 0$ kısıtlamasıyla her iki E ve E' düzlemde sabit kalan sadece bir nokta vardır.

İspat: $H_{S1} = E/E'$ hareketinde her ani t parametresi için sürüklenme hızının sıfır olduğu nokta sadece E düzleminde sabit kalmaz aynı zamanda E' düzlemi üzerinde de sabit kalır. $\vec{v}_f = \vec{0}$ alınmasıyla

$$-\dot{u}_1 - \frac{\dot{\varphi}u_2}{\sqrt{1+\dot{\varphi}^2}} + \frac{\dot{\varphi}x_2}{\sqrt{1+\dot{\varphi}^2}} = 0 \quad (3.7)$$

$$\frac{-\dot{\varphi}\varphi}{(1+\varphi^2)}u_2 - \dot{u}_2 - \frac{\dot{\varphi}\varphi x_2}{1+\varphi^2} = 0 \quad (3.8)$$

denklemlerini elde ederiz. Burada $\dot{\varphi} \neq 0$ ve $p \neq 0$ olduğundan (3.7) ve (3.8) denklemlerinden aşağıdaki bağıntıları buluruz. Shearing hareketinde $\dot{\vec{e}}_1 = 0$ olduğu için hareket boyunca \vec{e}_1 bazının sabit kaldığı anlaşılmaktadır. Bu nedenle, harekette sabit ve hareketli düzlemin anlık hareketlerinde her iki düzlemde noktaların x_1 apsisi aynı kalır. Böylece

$$p_1 = x_1 \quad p_2 = \frac{\sqrt{1+\dot{\varphi}^2}}{\dot{\varphi}}\dot{u}_1 + u_2 = -\frac{1+\varphi^2}{\dot{\varphi}\varphi}\dot{u}_2 - u_2$$

denklemleri pol noktasını verir.

Tanım 3.4.7: $\overline{OP} = \vec{P} = \vec{e}_1 p_1 + \vec{e}_2 p_2$ konum vektörü ile verilen $P(p_1, p_2)$ noktası H_{S1} hareketinin polü ya da ani dönme polü ya da ani dönme merkezi olarak adlandırılır.

Not 3.2: (3.6) denkleminde verilen sürüklenme hızında \dot{u}_1 ve \dot{u}_2 yerine yazarsak sürüklenme hızı pol noktasının bileşenleri ile

$$\vec{v}_f = (x_2 - p_2) \cdot \left[\frac{1}{\sqrt{1 + \phi^2}} \vec{e}_1 - \frac{\phi}{1 + \phi^2} \vec{e}_2 \right] \cdot \dot{\phi}.$$

biçiminde yazılabildiği görülür.

3.4.4 Sonuçlar:

1. Hareketli koordinat düzleminde t anında alınan X noktası ve aynı andaki P pol noktasının oluşturduğu vektör

$$\overline{PX} = \vec{e}_2 (x_2 - p_2)$$

olur. \overline{PX} vektörünün \vec{v}_f sürüklenme vektörü sıfır olmadığı için bu vektörlerin dik olmadığı görülür. Yani shear dönüşümü yardımıyla oluşturulan düzlemsel harekette

$$\langle \overline{PX}, \vec{v}_f \rangle = - \frac{(x_2 - p_2)^2 \phi}{1 + \phi^2}$$

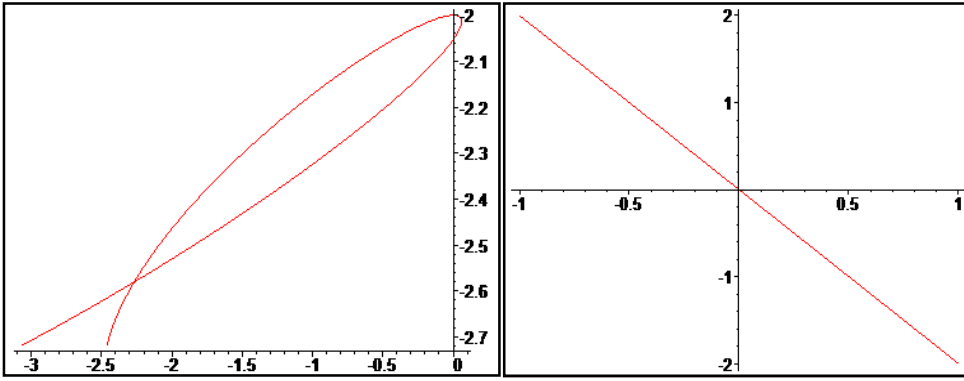
denklemini sağlar.

2. \vec{v}_f sürüklenme hız vektörünün uzunluğu shear dönüşümlü düzlemsel harekette aşağıdaki gibidir:

$$|\vec{v}_f| = |x_2 - p_2| \dot{\phi} \sqrt{\frac{1}{1 + \phi^2} + \frac{\phi^2}{(1 + \phi^2)^2}}.$$

3. Shearing dönüşümü ile bir parametrelili H_{S1} düzlemsel hareket üzerinde, E düzleminin hareketli (P) pol eğrisi E' düzleminin sabit (P') pol eğrisi üzerinde kaymadan yuvarlanamaz.

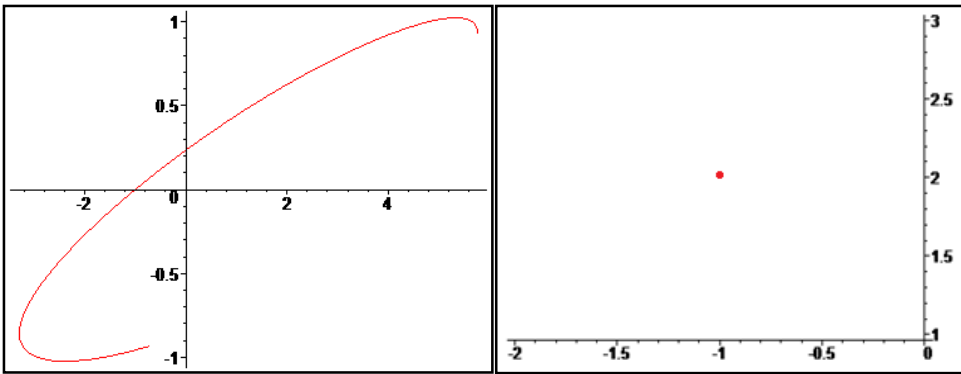
Örnek3.4.1 E düzlemi sabit E' düzlemi üzerinde başlangıç noktası $(t\cos t + 2t\sin t, t\sin t - 2t\cos t)$ olan ve dönme açısı $\varphi(t) = t$ olacak şekilde hareket etsin. Bu durumda E düzlemi üzerindeki bir $X = (x_1, x_2)$ noktasının E' düzlemine göre ve E nin E' düzlemine göre hem Öklid uzayında dönme, öteleme içeren düzlemsel harekete göre hem de Öklid uzayında shear hareketi içeren düzlemsel hareketine göre oluşan hareket dönüşümlerini ve hızlarının hesabını **MAPPLE 11** programı ile bularak sonuç eğrilerinin kıyaslarını yapalım, (Bkz Ek2 ve Ek3).



(a)

(b)

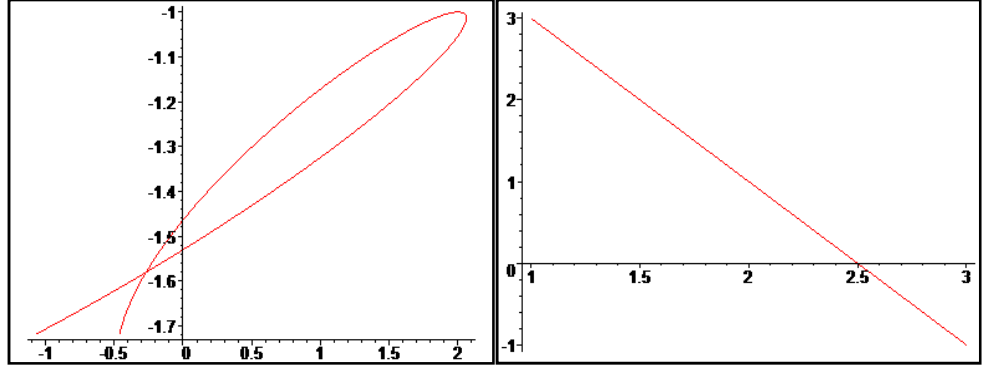
Şekil 3.10 \vec{u} öteleme vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

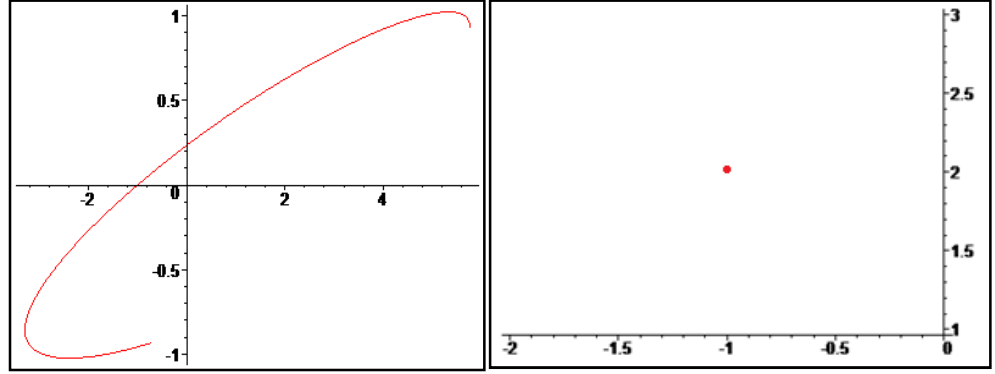
Şekil 3.11 \vec{u} öteleme vektörünün türev vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

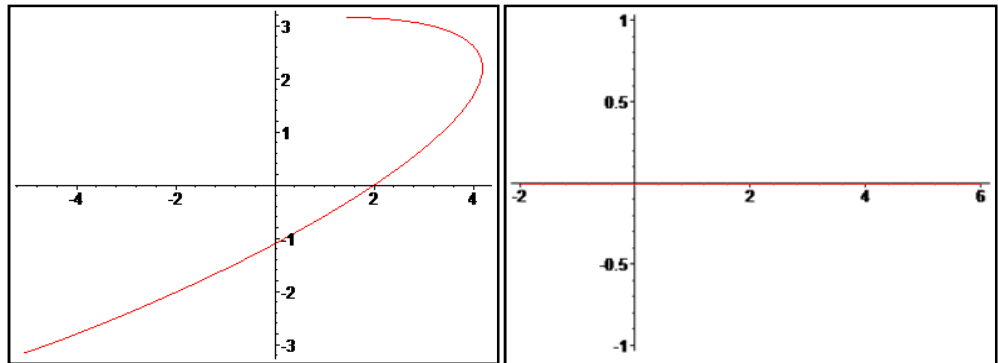
Şekil 3.12 \vec{x} vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

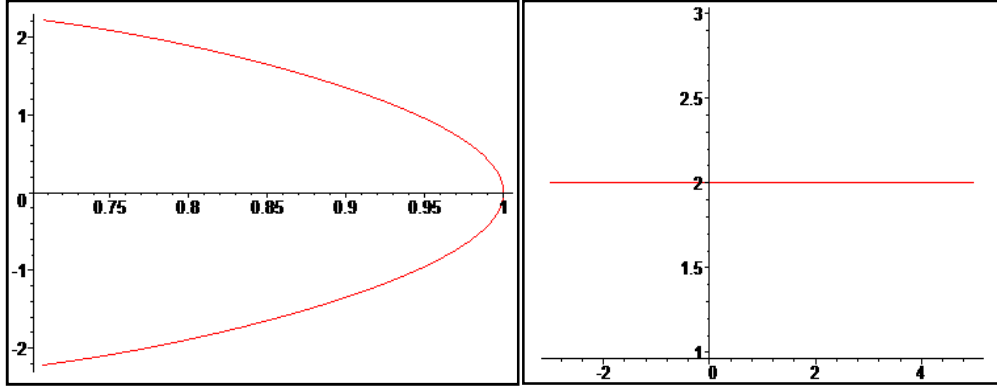
Şekil 3.13 \vec{v}_r bağıl hız vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

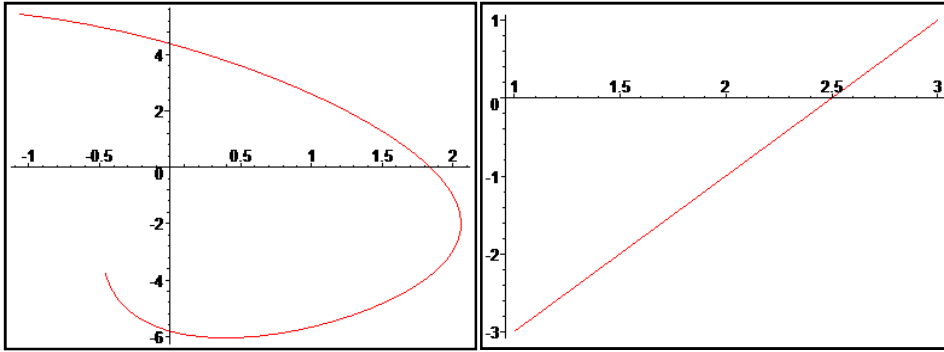
Şekil 3.14 \vec{v}_f bağıl hız vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

Şekil 3.15 \vec{v}_a bağılı hız vektörünün (a) shear hareketi (b) doğal hareketi boyunca çizdiği eğrisinin Mapple çıktısı



(a)

(b)

Şekil 3.16 Pol eğrisinin (a) shear hareketindeki (b) doğal hareketindeki Mapple çıktısı

4. ZAMAN SKALASINDA GEOMETRİK YAPILAR

4.1 Zaman Skalası ile İlgili Temel Bilgiler

Sürekli analizi ve ayrık analizi başarılı bir biçimde birleştirebilen zaman skalasında analiz, ilk olarak Stefan Hilger (1988) tarafından doktora tezinde ele alınmıştır. Bu bölümde zaman skalasında tanımlı bir fonksiyon için Δ türev kavramı ile ilgili bazı temel bilgiler verilmiştir.

Tanım 4.1.1: Reel sayıların boştan farklı kapalı bir alt kümesine zaman skalası denir ve T ile gösterilir (Bohner et al., 1966; 2001).

Tanım 4.1.2: Her $t \in T$, $t < \max T$ için $\sigma(t) = \inf\{s: s \in T, s > t\}$ ile tanımlanan $\sigma: T \rightarrow T$ operatörüne “ileriye sıçrama” (forward jump) operatörü adı verilir. T kapalı olduğundan, her $t \in T$ için $\sigma(t) \in T$ olur.

Her $t \in T$, $t > \min T$ için $\rho(t) = \sup\{s: s \in T, s < t\}$ ile tanımlanan $\rho: T \rightarrow T$ operatörüne “geriye sıçrama” (backward jump) operatörü adı verilir. T kapalı olduğundan her $t \in T$ için $\rho(t) \in T$ olur.

$\sigma(t) = t$ ise $t \in T$ ye “sağ yoğun” (right dense) nokta ve $\sigma(t) > t$ ise sağ yayılmış (right scattered) nokta denir.

$\rho(t) = t$ ise $t \in T$ ye “sol yoğun” (left dense) nokta ve $\rho(t) < t$ ise sol yayılmış (left scattered) nokta denir (Agarwal et al. , 1999; Bohner et al., 2001).

Tanım 4.1.3: $f: T \rightarrow IR$ fonksiyonu ve $t \in T^k$ noktası verilsin. Eğer sonlu bir $a \in IR$ sayısı için $\varepsilon > 0$ verildiğinde t noktasının

$$|f(\sigma(t)) - f(s) - a(\sigma(t) - s)| \leq \varepsilon |\sigma(t) - s|, \forall s \in U$$

olacak şekilde bir U komşuluğu varsa, f fonksiyonuna t noktasında Δ - türevlenebilir denir. Bu eşitsizlikteki a reel sayısına da f fonksiyonunun t noktasındaki Δ türevi denir ve $a = f^\Delta(t)$ yazılarak gösterilir. Başka bir deyişle,

$$f^\Delta(t) = \lim_{s \rightarrow t} \frac{f(\sigma(t)) - f(s)}{\sigma(t) - s}$$

yazılabilir.

Teorem 4.1.1: $f, g: T \rightarrow IR$ fonksiyonları ve $t \in T^k$ noktasında Δ -türevlenebilir ise,

i) $f + g$ fonksiyonu da $t \in T^k$ noktasında Δ -türevlenebilirdir ve

$$(f + g)^\Delta(t) = f^\Delta(t) + g^\Delta(t).$$

ii) Her k sabiti için kf fonksiyonu da $t \in T^k$ noktasında Δ -türevlenebilirdir ve

$$(kf)^\Delta(t) = kf^\Delta(t).$$

iii) fg fonksiyonu da $t \in T^k$ noktasında Δ -türevlenebilirdir ve

$$(fg)^\Delta(t) = f^\Delta(t)g(t) + f(\sigma(t))g^\Delta(t) = f(t)g^\Delta(t) + f^\Delta(t)g(\sigma(t)).$$

iv) $g(t)g(\sigma(t)) \neq 0$ ise, bu halde $\frac{f}{g}$ fonksiyonu da $t \in T^k$ noktasında Δ -türevlenebilirdir ve

$$\left(\frac{f}{g}\right)^\Delta(t) = \frac{f^\Delta(t)g(t) - f(t)g^\Delta(t)}{g(t)g(\sigma(t))}.$$

Tanım 4.1.4: $t_0 \in T$ olsun. Verilen her $\varepsilon > 0$ ve her $t \in U(t_0)$ için,

$$|f(t) - f(t_0)| < \varepsilon$$

olacak şekilde bir $U(t_0)$ komşuluğu bulunabiliyorsa, $f: T \rightarrow IR$ fonksiyonu $t = t_0$ noktasında “**süreklidir**” denir (Bohner et al., 2001).

Tanım 4.1.5: $f: T_1 \times T_2 \rightarrow IR$ sürekli fonksiyonu $(t_0, s_0) \in T_1 \times T_2$ noktasındaki

$V(t_0, s_0)$ komşuluğunda $\frac{\partial f(t, s)}{\Delta_1 t}$ kısmi türevine ve $\sigma_1((t_0), s_0)$ noktasını içeren

(t_0, s_0) noktasının $V^{\sigma_1}(t_0, s_0)$ komşuluğunda $\frac{\partial f(t, s)}{\Delta_2 t}$ kısmi türevine sahip olsun.

(t_0, s_0) noktasında bu türevler sürekli ise f fonksiyonu σ_1 -tamamen delta diferensiyellenebilirdir denir. Bu fonksiyonların oluşturduğu küme $C_{\sigma_1}^\Delta$ sembolü ile gösterilir. Ayrıca fonksiyonunun tanım kümesi sadece $I \in T$ den oluşuyorsa diferensiyellenebilen fonksiyonların kümesi $C^\Delta(I)$ ile gösterilir, (Bohner et al., 2003).

4.2 Zaman Skalasında Türev Dönüşümü ve Doğal Konneksiyon

Bu bölümde son zamanlarda kullanımı yaygınlaşan zaman skalası teorisi kullanılarak, herhangi bir T zaman skalasında tanımlı bir eğrinin vektörel alanının özelliklerini inceleyeceğiz. Ayrıca T zaman skalasında türev dönüşümünü ve doğal konneksiyonu tanımlayıp bunlarla ilgili özellikleri ispatlayacağız.

Tanım 4.2.1: I aralığı T zaman skalasında bir açık aralık ve $\alpha: I \subset T \rightarrow \mathbb{R}^2$ $\alpha(t) = (\alpha_1(t), \alpha_2(t))$ diferansiyellenebilir bir eğri olsun. Varsayalım ki Z , α üzerinde bir vektör alanı olsun. Eğri üzerinde Z nin $\alpha(t)$ noktasındaki değeri $Z(\alpha(t))$ dir. Vektör alanı tanımını düşünürsek

$$Z(\alpha(t)) \in T_{\alpha(t)}(\mathbb{R}^2)$$

şeklinde yazılabilir. $Z(\alpha(t))$ tanjant vektörünü gösterim kolaylığı açısından kısaca $Z(t)$ ile göstereceğiz. Her $t \in I$ için

$$Z(t) = \sum_{i=1}^2 z_i(t) \frac{\partial}{\partial x_i} (\alpha(t))$$

vektör alanı $Z \in C^A(I)$ delta diferansiyellenebilir olsun. Böylece $i = 1, 2$ için her z_i fonksiyonu $I \subset T$ den \mathbb{R} ye delta diferansiyellenebilir ve ayrıca sürekli fonksiyonlardır. $t, s \in I$ alınmasıyla $p = \alpha(s)$, $q = \alpha(\sigma(t))$ dir. Burada p noktasında $Z(\sigma(t)) \in T_q(\mathbb{R}^2)$ teğet vektörüne eşit olan sadece tek bir vektör vardır. Böylece

$$Z(\sigma(t)) - Z(s)$$

vektörünü tanımlayabiliriz ve burada $Z(s)$, p noktasındaki bir tanjant vektörü olarak düşünülebilir.

Tanım 4.2.2: $\alpha \in \mathbb{R}^2$, Δ -diferansiyellenebilir bir eğri olsun ve

$$\lim_{s \rightarrow t} \frac{Z(\sigma(t)) - Z(s)}{\sigma(t) - s}$$

vektörü Z vektör alanının $\alpha(t)$ noktasındaki vektör alanının delta türevi olarak adlandırılır ve $Z^\Delta(t)$ ile gösterilir.

Teorem 4.2.1: Z , α eğrisi üzerinde bir vektör alanı ise Z^Δ , α üzerinde bir vektör alanıdır.

İspat.

$$\begin{aligned} \frac{Z(\sigma(t)) - Z(s)}{\sigma(t) - s} &= \frac{1}{\sigma(t) - s} \sum_{i=1}^2 [z_i(\sigma(t)) - z_i(s)] \frac{\partial}{\partial x_i}(p) \\ &= \sum_{i=1}^2 \frac{z_i(\sigma(t)) - z_i(s)}{\sigma(t) - s} \frac{\partial}{\partial x_i}(p) \end{aligned}$$

Burada eğer $\sigma(s) \rightarrow t$ limitini alırsak

$$Z^\Delta(t) = \sum_{i=1}^2 z_i^\Delta(t) \frac{\partial}{\partial x_i}(p)$$

denklemini elde ederiz. Sonuç olarak Z^Δ , α eğrisi üzerinde bir vektör alanıdır.

Tanım 4.2.3: α eğrisi üzerinde bir Z delta diferansiyellenebilir vektör alanı verilsin. Eğer $t \in I \subset \mathbb{T}$ için $Z^\Delta(t) = 0$ yani $Z^\Delta = 0$ verilirse Z vektör alanına α eğrisi üzerinde sabit vektör alanı denir.

Teorem 4.2.2: Y ve Z Δ -diferansiyellenebilir α eğrisi üzerinde delta diferansiyellenebilir bir vektör alanı ve I, \mathbb{T} zaman skalası üzerinde bir açık aralık olsun. Eğer $f : I \subset \mathbb{T} \rightarrow \mathbb{R}$ bir delta diferansiyellenebilir bir fonksiyon ise

1. $(Y + Z)^\Delta = Y^\Delta + Z^\Delta$,
2. $(f.Y)^\Delta = f^\Delta.Y + f^\sigma.Y^\Delta = f.Y^\Delta + f^\Delta.Y^\sigma$,
3. $\langle Y, Z \rangle^\Delta = \langle Y^\Delta, Z \rangle + \langle Y^\sigma, Z^\Delta \rangle = \langle Y, Z^\Delta \rangle + \langle Y^\Delta, Z^\sigma \rangle$

denklemlerini elde ederiz.

İspat. $Y(t)$ yi ve $Z(t)$ vektör alanları

$$Y(t) = \sum_{i=1}^2 y_i(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)}, \quad Z(t) = \sum_{i=1}^2 z_i(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)}, \quad t \in I$$

olsunlar.

1. Vektör alanlarının toplamından

$$(Y + Z)(t) = Y(t) + Z(t) = \sum_{i=1}^2 (y_i(t) + z_i(t)) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)}$$

denklemini elde ederiz. Böylece $(Y + Z)$ Δ -fonksiyonunun türevi

$$(Y + Z)^\Delta(t) = \sum_{i=1}^2 (y_i(t) + z_i(t))^\Delta \frac{\partial}{\partial x_i} \Big|_{\alpha(t)}$$

olur. y_i ve z_i , I den \mathbb{R} ye Δ -diferansiyellenebilir fonksiyonlar olduğundan

$$\begin{aligned} &= \sum_{i=1}^2 y_i^\Delta(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} + \sum_{i=1}^2 z_i^\Delta(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} \\ &= Y^\Delta(t) + Z^\Delta(t) \\ &= (Y^\Delta + Z^\Delta)(t) \end{aligned}$$

olur.

2. Vektör alanının özelliklerinden dolayı

$$\begin{aligned} (fY)(t) &= f(t)Y(t) \\ &= f(t) \sum_{i=1}^2 y_i(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} \\ &= \sum_{i=1}^2 f(t) y_i(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} \\ &= \sum_{i=1}^2 (fy_i)(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} \end{aligned}$$

dır. Böylece (fY) fonksiyonunun Δ -türevi

$$\begin{aligned}
(fY)^\Delta(t) &= \sum_{i=1}^2 (fy_i)^\Delta(t) \frac{\partial}{\partial x_i} \Big|_{\alpha(t)} \\
&= f^\Delta(t)Y(t) + f(\sigma(t))Y^\Delta(t) \\
&= (f^\Delta Y)(t) + (f^\sigma Y^\Delta)(t) \\
&= (f^\Delta Y + f^\sigma Y^\Delta)(t)
\end{aligned}$$

dır. Böylece $(fY)^\Delta = f^\Delta Y + f^\sigma Y^\Delta$ olur.

3. İç çarpımın tanımından

$$\langle Y, Z \rangle = \sum_{i=1}^2 y_i z_i$$

dır. Böylece $\langle Y, Z \rangle$ iç çarpımının Δ türevi

$$\langle Y, Z \rangle^\Delta = \left(\sum_{i=1}^2 y_i z_i \right)^\Delta = \sum_{i=1}^2 (y_i z_i)^\Delta = \langle Y^\Delta, Z \rangle + \langle Y^\sigma, Z^\Delta \rangle$$

bulunur.

Tanım 4.2.4: $\Lambda^n = \mathbb{T}_1 \times \dots \times \mathbb{T}_n$ nin açık bir altkümesi U ve olmak üzere

$f = (f_1, f_2, \dots, f_m)$ bir $f : U \rightarrow \Lambda^m$, Δ -diferansiyellenebilir fonksiyon olsun.

$v_p \in T_p(U)$ için $T_p(\Lambda^n)$ uzayından $T_{f(p)}(\Lambda^m)$ uzayına f_{Δ^*p} dönüşümü

$$f_{\Delta^*p}(v_p) = \left(\frac{\partial f_1(P)}{\Delta_1 v}, \dots, \frac{\partial f_m(P)}{\Delta_m v} \right) = \left(\frac{\partial f_1}{\Delta_1 v}, \dots, \frac{\partial f_m}{\Delta_m v} \right)_{f(P)}$$

denklemleri ile tanımlanır ve f_{Δ^*p} fonksiyonu $P \in U$ noktasındaki f fonksiyonunun

Δ -türev dönüşümü olarak tanımlanır. U nun her P noktasına karşılık gelen f_{Δ^*p}

lineer dönüşümü zaman skalasında f fonksiyonunun türev dönüşümü olarak

adlandırılır. $v_p \in T_p(U)$ için $f_{\Delta^*p}(v_p)$ yerine gösterim olarak genellikle $f_{\Delta^*}(v_p)$

kullanılır. $T_p(\Lambda^n)$ ve $T_{f(p)}(\Lambda^m)$ uzaylarının doğal bazları sırayla

$$\left\{ \frac{\partial}{\Delta_1 x_1}(P), \dots, \frac{\partial}{\Delta_n x_n}(P) \right\} \text{ ve } \left\{ \frac{\partial}{\Delta_1 y_1}(f(P)), \dots, \frac{\partial}{\Delta_m y_m}(f(P)) \right\}$$

şeklindedir, (Kuşak and Çalışkan 2008).

Tanım 4.2.5: Λ^2 α uzayında V ve W iki vektör alanı olsun. W vektör alanının V vektör alanına göre

$$(\Delta_V W)(p) = \Delta_{V(p)} W$$

kovaryant türevinin oluşturduğu

$$\begin{aligned} \Delta : T(\Lambda^2) \times T(\Lambda^2) &\rightarrow T(\Lambda^2) \\ (V, W) &\rightarrow \Delta_V W \end{aligned}$$

operatörüne “delta doğal konneksiyonu” denir.

Sonuç 4.2.1: Vektör alanı

$$W = \sum_{i=1}^2 w_i \frac{\partial}{\partial x_i}$$

olarak alındığında delta doğal konneksiyonu

$$\Delta_V W = \sum_{i=1}^2 \frac{\partial w_i}{\Delta V} \frac{\partial w_i}{\partial x_i}$$

denklemini ile gösterilir.

Sonuç 4.2.2: V vektör alanı, zaman skalasında α eğrisi üzerindeki α^Δ delta teğet vektör alanı olarak düşünüldüğünde doğal konneksiyon

$$\Delta_{\alpha^\Delta(t)} f = \sum_{i=1}^2 \frac{\partial f_i}{\Delta \alpha^\Delta(t)} \frac{\partial}{\partial x_i}$$

denklemini ile ifade edilir.

Teorem 4.2.3: Λ^2 üzerindeki vektör alanları V, W, Y, Z olsun ve her $a, b \in IR$ için $f, g \in C_{\sigma_1}^\Delta$ olmak üzere delta doğal konneksiyonun bazı özelliklerinin aşağıdaki şekilde sağlandığı görülmektedir:

$$1) \Delta_V (fY + gZ) = f \Delta_V Y + g \Delta_V Z$$

$$2) \Delta_{fV+gW}Y = f\Delta_VY + g\Delta_WY$$

$$3) \Delta_V(fY) = \frac{\partial f}{\Delta V}Y(\sigma_1(t^0), s^0) + f(\sigma_1(t^0), \sigma_2(s^0))\Delta_VY + \left\{ -\mu_1V_1 \frac{\partial f}{\Delta_1V_1} + \mu_2V_1 \frac{\partial f(\sigma_1(t^0), s^0)}{\Delta_2V_2} \right\} \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1V_1} \frac{\partial}{\partial x_i}$$

$$4) \Delta_V \langle Y, Z \rangle = \langle \Delta_VY, Z \rangle + \langle \Delta_VZ, Y \rangle - \mu_1V_1 \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1V_1} \frac{\partial z_i}{\Delta_1V_1} \frac{\partial}{\partial x_i} - \mu_2V_1 \sum_{i=1}^2 \frac{\partial z_i}{\Delta_1V_1} \frac{\partial y_i(\sigma_1(t^0), s^0)}{\Delta_2V_2} \frac{\partial}{\partial x_i}$$

İspat: Bu teoremin ispatında vektör alanlarının özellikleri için (Barret, 1966) ve (Aktan et al.,2009) kaynaklarından yararlanılmıştır.

1) Y ve Z vektör alanları

$$Y = \sum_{i=1}^2 y_i \frac{\partial}{\partial x_i}, \quad Z = \sum_{i=1}^2 z_i \frac{\partial}{\partial x_i}$$

denklemleri ile ifade edilir. Vektör alanlarının ve kovaryant türevin özelliklerinden dolayı

$$fY + gZ = \sum_{i=1}^2 (fy_i + gz_i) \frac{\partial}{\partial x_i}$$

denklemi yazılır ve elde edilen vektör alanına ait delta doğal konneksiyonu

$$\begin{aligned} \Delta_V(fY + gZ) &= \sum_{i=1}^2 \frac{\partial(fy_i + gz_i)}{\Delta V} \frac{\partial}{\partial x_i} \\ &= f \sum_{i=1}^2 \frac{\partial y_i}{\Delta V} \frac{\partial}{\partial x_i} + g \sum_{i=1}^2 \frac{\partial z_i}{\Delta V} \frac{\partial}{\partial x_i} \\ &= f\Delta_VY + g\Delta_VZ \end{aligned}$$

olarak bulunur.

2) Vektör alanlarının tanımından dolayı Y vektör alanı

$$Y = \sum_{i=1}^2 y_i \frac{\partial}{\partial x_i}$$

ile ifade edilir. Vektör alanlarının ve kovaryant türevin özelliklerinden

$$\begin{aligned}
\Delta_{(fV+gW)}Y &= \sum_{i=1}^2 \frac{\partial y_i}{\Delta(fV+gW)} \frac{\partial}{\partial x_i} \\
&= f \sum_{i=1}^2 \frac{\partial y_i}{\Delta V} \frac{\partial}{\partial x_i} + g \sum_{i=1}^2 \frac{\partial y_i}{\Delta W} \frac{\partial}{\partial x_i} \\
&= f \Delta_V Y + g \Delta_W Y
\end{aligned}$$

ispatı elde edilir.

3) Benzer biçimde

$$fY = \sum_{i=1}^2 (fy_i) \frac{\partial}{\partial x_i}$$

vektör alanı için kovaryant türev alındığında

$$\begin{aligned}
\Delta_V(fY) &= \frac{\partial f}{\Delta V} \sum_{i=1}^2 y_i(\sigma_1(t^0), s^0) \frac{\partial}{\partial x_i} + f(\sigma_1(t^0), \sigma_2(s^0)) \sum_{i=1}^2 \frac{\partial y_i}{\Delta V} \frac{\partial}{\partial x_i} \\
&\quad - \mu_1(t^0) V_1 \frac{\partial f}{\Delta_1 t} \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1 t} \frac{\partial}{\partial x_i} + \mu_2(s^0) V_1 \frac{\partial f(\sigma_1(t^0), s^0)}{\Delta_2 s} \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1 t} \frac{\partial}{\partial x_i} \\
&= \frac{\partial f}{\Delta V} Y(\sigma_1(t^0), s^0) + f(\sigma_1(t^0), \sigma_2(s^0)) \Delta_V Y \\
&\quad + \left\{ -\mu_1(t^0) V_1 \frac{\partial f}{\Delta_1 t} + \mu_2(s^0) V_1 \frac{\partial f(\sigma_1(t^0), s^0)}{\Delta_2 s} \right\} \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1 t} \frac{\partial}{\partial x_i}
\end{aligned}$$

verilen özelliğin ispatlandığı görülmektedir.

4) İç çarpımın oluşturduğu vektör alanı

$$\langle Y, Z \rangle = \sum_{i=1}^2 y_i z_i$$

denklemleri ile gösterilir. İç çarpım vektör alanının V vektör alanı yönündeki konneksiyonu

$$\begin{aligned}
\Delta_V(Y, Z) &= \sum_{i=1}^2 \frac{\partial y_i}{\Delta V} \frac{\partial}{\partial x_i} z_i(\sigma_1(t^0), s^0) + \sum_{i=1}^2 \frac{\partial z_i}{\Delta V} \frac{\partial}{\partial x_i} y_i(\sigma_1(t^0), \sigma_2(s^0)) \\
&\quad - \mu_1 V_1 \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1 V_1} \frac{\partial z_i}{\Delta_1 V_1} \frac{\partial}{\partial x_i} - \mu_2 V_1 \sum_{i=1}^2 \frac{\partial z_i}{\Delta_1 V_1} \frac{\partial y_i(\sigma_1(t^0), s^0)}{\Delta_2 V_2} \frac{\partial}{\partial x_i} \\
&= \langle \Delta_V Y, Z \rangle + \langle \Delta_V Z, Y \rangle - \mu_1 V_1 \sum_{i=1}^2 \frac{\partial y_i}{\Delta_1 V_1} \frac{\partial z_i}{\Delta_1 V_1} \frac{\partial}{\partial x_i} - \mu_2 V_1 \sum_{i=1}^2 \frac{\partial z_i}{\Delta_1 V_1} \frac{\partial y_i(\sigma_1(t^0), s^0)}{\Delta_2 V_2} \frac{\partial}{\partial x_i}
\end{aligned}$$

olacak şekilde bulunur, (Kuşak and Çalışkan, 2011a).

5 DUAL UZAY

5.1 Temel Kavramlar

Tanım 5.1.1: Birimi 1 olan deęişimli bir halka H olsun. H üzerinde bir S abel grubu ile ařağıdaki özelliklere sahip olan S üzerinde ki bir

$$H \times S \rightarrow S$$

$$(a, \alpha) \rightarrow a\alpha$$

dıř iřlemine “**modül**” denir. $a, b \in H$ ve $\alpha, \beta \in S$ olmak üzere

$$M_1: a(\alpha + \beta) = a\alpha + a\beta ;$$

$$M_2: (a + b)\alpha = a\alpha + b\alpha ;$$

$$M_3: (ab)\alpha = a(b\alpha) ;$$

$$M_4: 1 \cdot \alpha = \alpha .$$

Teorem 5.1.2: $\mathbb{D}^3 = \{(A_1, A_2, A_3) : A_1, A_2, A_3 \in \mathbb{D}\}$ kümesi verilsin. $(\mathbb{D}^3, +)$ sistemi \mathbb{D} üzerinde bir modüldür ve “ \mathbb{D} -**Modül**” olarak adlandırılır.

Tanım 5.1.3: \mathbb{D} - *modül* ’ün elemanları olan sıralı dual üçlülere “**dual vektörler**” denir.

Teorem 5.1.4: $\vec{a}, \vec{a}^* \in \mathbb{R}^3$ (\mathbb{R}^3 üç boyutlu reel vektör uzayını göstermektedir.) olmak üzere \mathbb{D} - *modül* de her bir \vec{A} dual vektörü $\vec{A} = \vec{a} + \varepsilon \vec{a}^*$, [$\varepsilon = (0,1) \in \mathbb{D}$] şeklinde yazılabilir.

Tanım 5.1.5: $\vec{A} = \vec{a} + \varepsilon \vec{a}^*, \vec{B} = \vec{b} + \varepsilon \vec{b}^* \in \mathbb{D}$ - *modül* dual vektörlerinin “**iç çarpımı**”

$$f: \mathbb{D}^3 \times \mathbb{D}^3 \rightarrow \mathbb{D}^3, \quad f(\vec{A}, \vec{B}) = \langle \vec{A}, \vec{B} \rangle = \langle \vec{a} + \varepsilon \vec{a}^*, \vec{b} + \varepsilon \vec{b}^* \rangle$$

olarak tanımlanır.

Tanım 5.1.6: $\|\vec{A}\| = (\langle \vec{A}, \vec{A} \rangle)^{1/2} = \left(\|\vec{a}\|, \varepsilon \frac{\langle \vec{a}, \vec{a}^* \rangle}{\|\vec{a}\|} \right), \vec{a} \neq \vec{0}$, dual sayısına $\vec{A} = \vec{a} + \varepsilon \vec{a}^*$ dual vektörünün “**normu**” denir.

Tanım 5.1.7: Normu reel birime karşılık gelen $(1,0)$ dual sayısı olan dual vektöre “**birim dual vektör**” denir.

Tanım 5.1.8: $\{\vec{X} = \vec{x} + \varepsilon \vec{x}^* \mid \|\vec{x}\| = (1,0); \vec{x}, \vec{x}^* \in \mathbb{R}^3\}$ cümlesine $\mathbb{D} - \text{modül}$ ’de “**birim dual küre**” denir.

Teorem 5.1.1(E. Study): $\vec{A} \neq (\vec{0}, \vec{a}) \in \mathbb{D} - \text{modül}$ olmak üzere $\mathbb{D} - \text{modül}$ ’de denklemi $\|\vec{A}\| = (1,0)$ olan birim dual kürenin dual noktaları, \mathbb{R}^3 de ki yönlü doğrulara birebir karşılık gelir.

Tanım 5.1.9: $\vec{A} = \vec{a} + \varepsilon \vec{a}^* \in \mathbb{D} - \text{modül}$ olmak üzere $\vec{U} = \frac{\vec{A}}{\|\vec{A}\|}$ birim dual vektörüne \vec{A} vektörünün “**ekseni**” denir.

Tanım 5.1.10: $k = \frac{\langle \vec{a}, \vec{a}^* \rangle}{\|\vec{a}\|^2}$ reel sayısına $\vec{A} = \vec{a} + \varepsilon \vec{a}^*$ dual vektörünün “**adımı**” veya “**yükselişi**” denir.

- (i) $k = \text{sonlu bir sayı}$ ise $\vec{a} \neq \vec{0}$ ve $\vec{a}^* \neq \vec{0}$ dir ve \vec{A} dual vektörüne “**has dual vektör**” veya “**vida**” denir.
- (ii) $k = 0 \Rightarrow \vec{A} = a\vec{U}$ dir. Bu halde \vec{A} dual vektörü \vec{U} eksenine ile “**çakışık bir doğru**” gösterir.
- (iii) $k = \infty \Rightarrow \vec{a} = \vec{0}$ ise \vec{A} dual vektörü bir “**sırf dual**” vektördür.

Tanım 5.1.11: $\langle \vec{A}, \vec{B} \rangle = \cos \phi = \cos \varphi + \varepsilon \varphi^*$ olmak üzere $\phi = \varphi + \varepsilon \varphi^*$ dual sayısına \vec{A} ve \vec{B} birim vektörleri arasındaki “**dual açı**” denir.

Tanım 5.1.12: $\vec{A}, \vec{B} \in \mathbb{D} - \text{modül}$ dual vektörlerinin “**dış çarpımı**”

$$\wedge : \mathbb{D}^3 \times \mathbb{D}^3 \rightarrow \mathbb{D}^3$$

$$\vec{A} \wedge \vec{B} = \vec{a} \wedge \vec{b} + \varepsilon (\vec{a} \wedge \vec{b}^* + \vec{a}^* \wedge \vec{b}) = \|\vec{A}\| \|\vec{B}\| \sin \phi \vec{N}$$

olarak tanımlanır, (Yang, 1963).

5.2 Robotik Hareketler

Robotik, fiziksel aktivite ve karar verme gibi uygulamalarla bir görevi yürüterek insanların yerini alabilecek makinalarla ilgili çalışmaları içerir. Robotik, geleneksel mühendislik sınırlarını keşiştiren yeni bir modern teknoloji alanıdır. Robotların karmaşıklığını ve uygulama alanlarını anlamak elektrik-elektronik mühendisliği, makina mühendisliği, endüstri mühendisliği, bilgisayar mühendisliği, matematik alanlarında geniş bir bilgi ağı gerektirmektedir. Bu bölümde robotu oluşturan parçalar kısaca incelenecektir (Kayhan, 2003; Paul 1981; Sciavicco et al. 1996 ; Şuhubi, 1988).

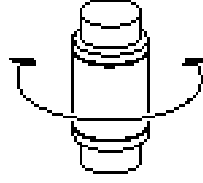
Tanım 5.2.1: Robot, bir dizi verilen görev çerçevesinde çeşitli programlanmış hareketler ile materyalleri, parçaları, aletleri veya özel donanımları hareket ettirmek için tasarlanmış programlanabilir çok işlevli manipülatördür (Robot Institute of America-RIA).

Tanım 5.2.2: Robot dört ana kısımdan meydana gelir :

1. Bir mekanik yapı yada eklemlerle birbirine bağlanmış sıralı katı cisimlerden(uzuvlardan) oluşan manipülatör; manipülatör, serbestliği sağlayan bir koldan, el becerisi sağlayan bir bilekten ve robotun yapması gereken görevi tamamlayan sonlandırıcıdan oluşmaktadır.
2. Eklemlerin hareketlenmesiyle manipülatörün hareketini sağlayan hareketlendiriciler
3. Manipülatörün veya çevrenin durumunu gözleyen algılayıcılar
4. Manipülatör hareketini kontrol eden ve yöneten bir kontrol sistemi (Kayhan, 2003; Paul 1981; Sciavicco et al. 1996 ; Şuhubi, 1988).

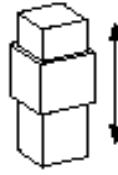
Tanım 5.2.3 (Eklem Yapıları): Eklemler manipülatörlerde hareketi sağlayan mekanizmalardır ve yapılarına göre ikiye ayrılırlar (Kayhan, 2003; Paul 1981; Sciavicco et al. 1996 ; Şuhubi, 1988):

1. **Döner Eklemler:** Mentşeşeye benzer ve iki uzuv arasında dönme hareketini yapar, (Bkz Şekil 5.1).



Şekil 5.1 Döner tip eklem (Kayhan , 2003).

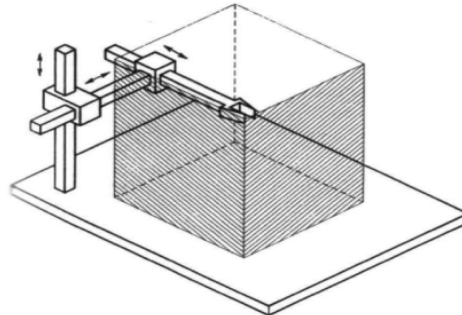
2. **Kayar Eklemeler:** İki uzuv arasında doğrusal hareketini yapar, (Bkz Şekil 5.2).



Şekil 5.2 Kayar tip eklem (Kayhan , 2003).

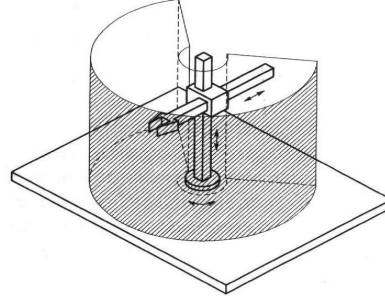
Tanım 5.2.4 (Manipülörlerin Sınıflandırılması): Manipülörler çalışma uzaylarına göre sınıflandırılırlar. Aşağıda manipülör yapıları ve bunların çalışma uzayları görülmektedir (Kayhan, 2003; Paul 1981; Sciavicco et al. 1996; Şuhubi, 1988).

1. **Kartezyen manipülör:** Bu tip bir manipülör üç tane kayar tip eklem ile elde edilir. Mekanik yönden çok sağlamdır fakat çalışma uzayındaki hareket yeteneği bakımından zayıftır. Bu tip manipülörler çok büyük boyutlarda ve ağırlıklarda nesnelere hareket ettirmek ve taşımak için idealdir. Kartezyen manipülörlerde eklemleri hareket ettiren motorlar çoğunlukla elektrik motorlarıdır, (Bkz Şekil 5.3).



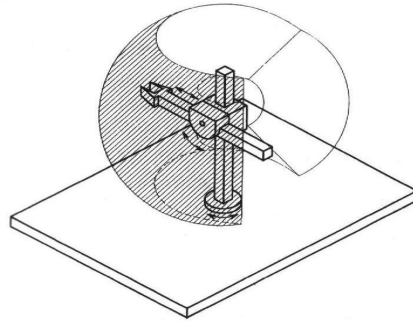
Şekil 5.3 Kartezyen manipülör (Kayhan, 2003).

- 2. Silindirik manipülatör:** Bu tip bir manipülatör bir tane döner ve iki tane kayar tip eklem ile elde edilir. Bu tip manipülatörler de mekanik yönden sağlamdır fakat bilek konum doğruluğu yatay harekete bağlı olarak azalır. Benzer şekilde büyük boyutlu nesnelerin taşınmasında kullanılırlar. Bu tip manipülatörlerde hidrolik motorları tercih edilir, (Bkz Şekil 5.4).



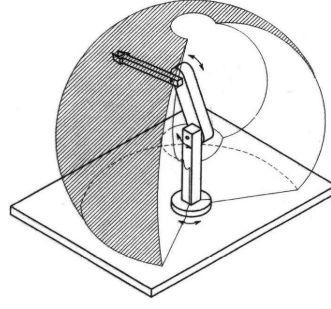
Şekil 5.4 Silindirik manipülatör (Kayhan , 2003).

- 3. Küresel manipülatör:** Bu tip bir manipülatör iki tane döner ve bir tane kayar tip eklem ile elde edilir. Bu tip manipülatörler mekanik yönden diğer iki tipten daha zayıf, mekanik yapı yönünden daha karmaşıktır. Çoğunlukla makina montajlarında kullanılırlar. Bu tip manipülatörlerde elektrik motorları tercih edilir, (Bkz Şekil 5.5).



Şekil 5.5 Küresel manipülatör (Kayhan , 2003).

- 4. Eklemlİ-İnsan Kolu manipülatör:** İnsan kol yapısı esas alındığı için bu isim verilmiştir. Bu tip manipülatörler tüm eklemleri döner olduğundan çalışma uzaylarında en yetenekli manipülatörlerdir. Endüstriyel uygulamalarda geniş kullanım alanına sahiptirler. Bu tip manipülatörlerde elektrik motorları tercih edilir, (Bkz Şekil 5.6).



Şekil 5.6 Eklemlİ-İnsan kolu manipülatör (Kayhan , 2003).

5.3 Dual Kuaterniyon ve Dual Uzayların Robotik Bir Uygulaması

Robotik hareketlerde katı cisim dönmeleri veya birim reel vektörlerin döndürülmesi sıkça kullanılmaktadır. Ancak, kinematik analizlerinde hesaplamaların karmaşıklığından dolayı artık robotik hareketlerde kuaternionlar kullanılmaya başlanmıştır. Reel uzayda kuaternionlar dönmeyi, dual kuaterniyonlar hem dönme hem ötelemeye yardımcı olmaktadır. Ayrıca dual vektörler ile kolayca bir vida ya da bir eksen tanımlayabiliriz. Mekanizmaların incelenmesinde, son yıllarda, dual kuaternionlar ve vida hareketleri kullanılarak yararlı çalışmalar yapılmıştır.

Biz bu çalışmamızda çok iyi bilinen reel küresel dirsek (wrist) hareketinde oluşan dönmeler için bilinen dizisel ve modüler metodları dual uzaya taşıdık. Böylece reelde sadece birim vektörleri döndürürken artık vidaları veya eksenleri dual kinematikteki dual küresel mekanizmaların üzerinde döndürebileceğiz. Bu çalışma ile ilk defa dual küresel mekanizmalara geçilerek reelde birim vektörlerle yaptığımız mekanik olayları dual uzayda vidalarla gerçekleştirebileceğimizi göreceğiz. Ayrıca reel küresel dirsek hareketinde mafsalların hareketiyle çizilen yörüngeler birer eğri iken dual küresel dirsek hareketinde mafsalların çizdiği yörüngeler regle yüzeyi olmaktadır. Bu durumun farklı mekanik çalışmalara bir baz oluşturabileceğini düşünüyoruz.

Son günlerde fizik, kimya, robotik, mekanik, elektronik gibi birçok bilim dalında kullanım pratikliği açısından kuaternionlar yaygınca kullanılmaya başlanmıştır. $q = a + b\vec{i} + c\vec{j} + d\vec{k}$ kuaternionun reel bir vektörü yada bir katı cismin matris formundaki noktalarını döndürebilmektedir. Normalde döndürme

matrislerinin kompleks yapısının oluşturduğu zorluk kuaternionun basit yapısı ile ortadan kalkmaktadır. Ayrıca küresel dirsek hareketinin kinematik hesaplamaları için kuaternionları kullanarak yeni bir metod geliştirmiştir, (Veldkamp, 1976). Bu çalışmalarda katı cisimlerin yada birim reel vektörlerin döndürülmesini sağlayan kuaternion operatörü, dönme ve ötelemeyi aynı anda uygulama konusunda yetersiz kalmaktadır. Bu gibi durumlarda A, B, C, D dual sayılar olmak üzere $Q = A + Bi + Cj + Dk$ dual kuaternionunun kullanımı tercih edilmektedir. Özellikle uzay kinematiğinde ve robotikte dual vektörlerin yada axodeların dual kuaternionlarla döndürülüp ötelenmesi hem işleyiş açısından hem pratik açısından kuaternionlara göre daha kullanışlıdır. Veldkamp ve Yang, dual sayı matrislerini ve dual kuaterniyonlarını ani uzay kinematiğinde kullanmıştır, (Agrawal , 1985). Çalışmamızda robotik ve mekanik çalışmalarda iyi bilinen küresel dirsek hareketini dual uzay üzerinde incelemeye çalışacağız. Kuaternionlarla birim vektörleri döndürürken çalışmamızda doğruların ve vidaların döndürülmesi için dual kuaternionları kullanacağız. Böylece dual küresel dirsek hareketini oluşturmuş olacağız. Dual uzaya geçmemizdeki bir avantajın da reel küresel dirsek hareketlerinde mekanizmanın oluşturduğu yörünge bir eğri olmasına rağmen dual küresel dirsek hareketinde mekanizmanın oluşturduğu yörünge bir regle yüzey olduğu görülmektedir.

5.3.1 İki dual vektörün çarpımı ile dual kuaterniyon elde etme

Bir dual kuaternion

$$Q = d + a\vec{i} + b\vec{j} + c\vec{k} + d^*\varepsilon + a^*\varepsilon\vec{i} + b^*\varepsilon\vec{j} + c^*\varepsilon\vec{k} \quad (5.3.1)$$

gibi $+1, \vec{i}, \vec{j}, \vec{k}, \varepsilon, \varepsilon\vec{i}, \varepsilon\vec{j}, \varepsilon\vec{k}$ biçimindeki temel elemanlar cinsinden tanımlanabilir. Burada $d, a, b, c, d^*, a^*, b^*, c^* \in \mathbb{R}$ ve

$$i^2 = j^2 = k^2 = ijk = -1, \quad \varepsilon^2 = 0$$

Şartları sağlanmaktadır. (5.3.1) denkleminde verilen dual kuaternionu

$$D = d + \varepsilon d^*, A = a + \varepsilon a^*, B = b + \varepsilon b^*, C = c + \varepsilon c^*$$

dual vektörlerin birleşimi şeklinde aşağıdaki gibi yazabiliriz:

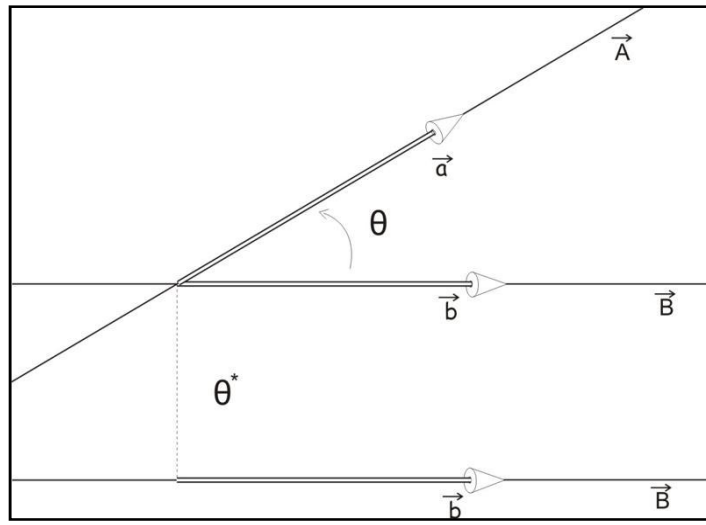
$$Q = D + A\vec{i} + B\vec{j} + C\vec{k}$$

Ayrıca bir dual kuaternionu iki dual vektörün dual kuaternion çarpımı olarak da tanımlayabiliriz. Yani aynı zamanda Q dual kuaternionunu

$$Q = (A_1\vec{i} + B_1\vec{j} + C_1\vec{k}).(A_2\vec{i} + B_2\vec{j} + C_2\vec{k})$$

$$= -(A_1A_2 + B_1B_2 + C_1C_2) + \vec{i}(B_1C_2 - C_1B_2) + \vec{j}(C_1A_2 - A_1C_2) + \vec{k}(A_1B_2 - B_1A_2)$$

olarak tanımlayabiliriz. \vec{R}_1, \vec{R}_2 ve \vec{R} dual vektörlerin birim uzunlukta olduğunu düşünelim. θ, \vec{R}_1 ve \vec{R}_2 birim vektörleri arasındaki dönme açısı ve θ^* ise uzaklığı gösterebilir. O halde $\Theta = \theta + \varepsilon\theta^*$ dual açısı tanımlanabilir, (Bkz Şekil 5.7).



Şekil 5.7 Dual birim vektörler arasındaki dual açı, (Yang et al., 1964).

\vec{R}_3 vektörü \vec{R}_1 ve \vec{R}_2 vektörüne dik olan bir doğru ise \vec{R}_1 ve \vec{R}_2 dual vektörlerinin dual kuaternion çarpımı sonucu bir birim dual kuaternion oluşur:

$$Q = \vec{R}_1 \times \vec{R}_2 = (A_1\vec{i} + B_1\vec{j} + C_1\vec{k}).(A_2\vec{i} + B_2\vec{j} + C_2\vec{k})$$

$$= -\langle \vec{R}_1, \vec{R}_2 \rangle + \vec{R}_1 \wedge \vec{R}_2$$

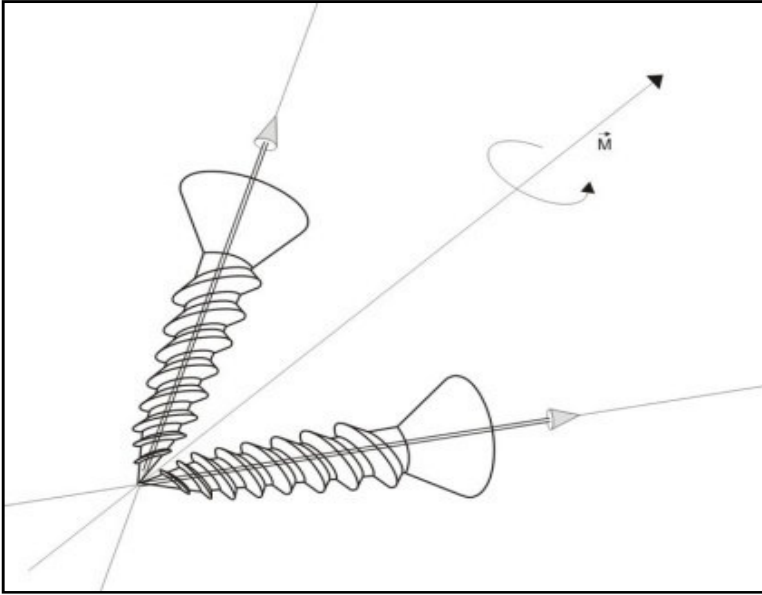
$$= |\vec{R}_1||\vec{R}_2| \cos \Theta + |\vec{R}_1||\vec{R}_2| \sin \Theta \vec{M}$$

$$= -\cos \Theta + \vec{M} \sin \Theta, \quad \vec{M} = \vec{m} + \varepsilon\vec{m}^*.$$

Burada \vec{R}_3 vektörü \vec{M} birim virda eksenini ile çakışmıştır yani $\vec{R}_3 = \vec{M}$ dir. O halde bir birim dual kuaternionu $Q = -\cos \Theta + \vec{M} \sin \Theta$ olarak göstermenin mümkün olduğunu görürüz.

5.3.2 Bir vidayı dual kuaterniyon yardımıyla döndürme

Birim reel $q = a + b\vec{i} + c\vec{j} + d\vec{k}$ kuaternionu, reel uzayda bir vektörün belirli bir açı ile bir eksen etrafında dönmesini sağlamaktadır. Kuaternionların mühendislikte; mekanik ve özellikle robotik çalışmalarda yaygın olarak kullanıldığı görülmektedir. Ancak dönüştürmek istediğimiz yapı bir vektör değil de bir vida ya da eksen olsaydı reel kuaternionlar bu yapıların dönüştürülmesinde yetersiz kalacaktı. Robotik hareketlerde vidaların ya da eksenlerin aynı anda hem döndürme hem öteleme hareketlerini içeren dönüşümü birim dual kuaternionlarla yapılmaktadır. Birim dual kuaternion $Q = -\cos \Theta + \vec{M} \sin \Theta$ denklemi ile verilir. Burada $\Theta = \theta + \varepsilon\theta^*$ dual açı ve $\vec{M} = \vec{m} + \varepsilon\vec{m}^*$ vida eksenini göstermektedir, (Bkz Şekil 5.8). Q dual kuaternionun tersi $Q^{-1} = -\cos \Theta - \vec{M} \sin \Theta$ olsun. $\vec{R} = \vec{r} + \varepsilon\vec{r}^*$ vidasının Q dual kuaternionu ile dönüştürülmesini $Q(\vec{R})Q^{-1}$ çarpımı ile gösterebiliriz. Ancak bu gösterim ile \vec{R} vidasını iki kez Θ dual açısı ile hareket ettirmiş oluyoruz. Bu çarpımı kullanarak bir kez vida hareketi yapabilmek için Q dual kuaternionu $Q = -\cos \frac{\Theta}{2} + \vec{M} \sin \frac{\Theta}{2}$ olarak alalım.



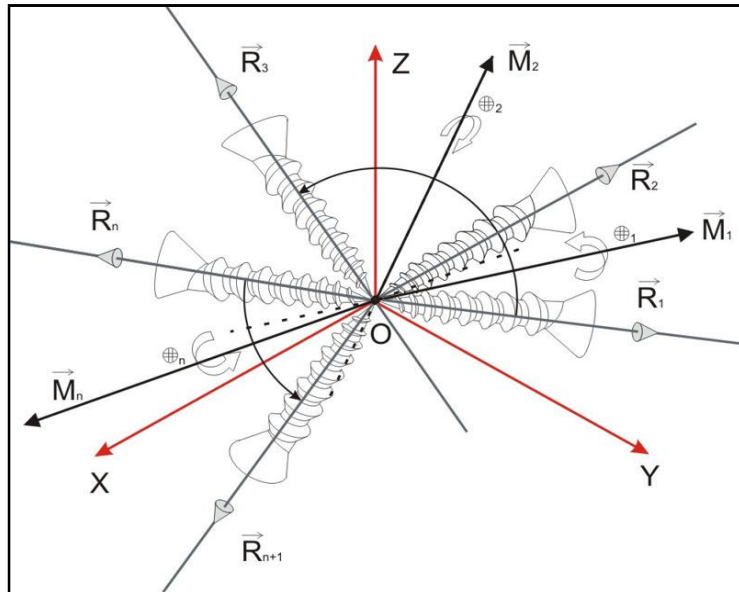
Şekil 5.8 Bir vidayı bir eksen etrafında döndürme.

Dual Θ açısı vidaları hem döndürüp hem de ötelemektedir. Ancak çalışmamız dual küre bilek hareketleri ile ilgili olduğu için dual kuaternionda öteleme açısını $\theta^* = 0$ olarak alarak, sadece dönme operatörünü içeren dual kuaternionu kullanacağız. O halde $\Theta = \theta$ olup $Q = -\cos\frac{\theta}{2} + \vec{M} \sin\frac{\theta}{2}$ kuaternionu ile vidaya ya da eksene dönme hareketi uygulayacağız. Aynı zamanda kolaylık açısından \vec{R} vidasını birim olarak eksen haline getirelim.

5.3.3 Bir dual kuaterniyon ile belirlenen vida dönmelerinin dizisel metod ile hareketleri

Vida ya da eksenlerin dizisel metod ile dual kuaternion yardımıyla döndürülebilmesi için $i = 1, 2, \dots, n$ olmak üzere $\vec{R} = \vec{r} + \varepsilon \vec{r}^*$ ekseninin Θ_i dual açıları ile \vec{M}_i eksenleri etrafında dönmesi gerekir. O halde dizisel metotta her \vec{R}_i vidası \vec{R}_{i+1} vidasına dönüştürülmektedir. Yani dizisel metod $\vec{R}_{i+1} = Q_i(R_i)Q_i^{-1}$ ile tanımlanır, (Bkz Şekil 5.9). Genelleştirilmiş olarak \vec{R}_1 vidasına dizisel olarak Q dual kuaternionunun uygulanışı aşağıda verilen denklem gibidir:

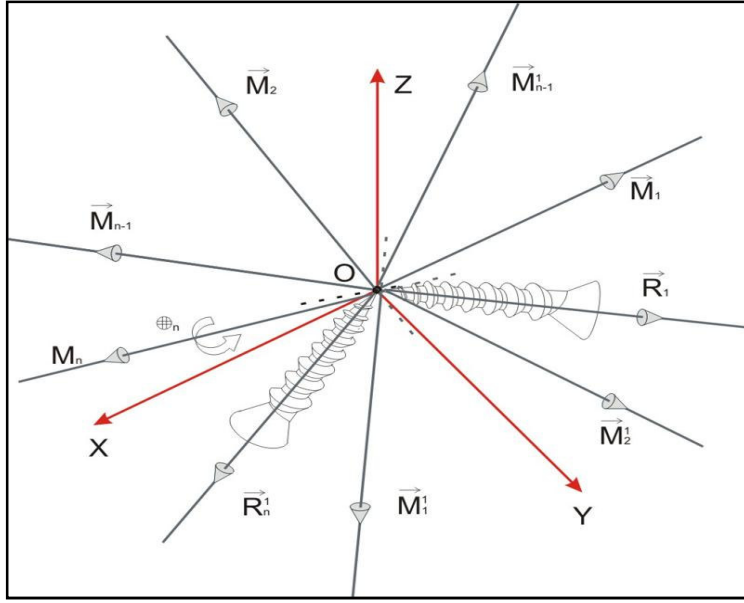
$$\vec{R}_{n+1} = Q_n Q_{n-1} \dots Q_2 Q_1 (\vec{R}_1) Q_1^{-1} Q_2^{-1} Q_3^{-1} \dots Q_{n-1}^{-1} Q_n^{-1}.$$



Şekil 5.9 Bir vidanın dizisel metod ile hareketi.

5.3.4 Bir dual kuaterniyonun dönme hareketi ile tanımlanan modüler metod

Robotik hareketlerde vida hareketlerinin gerçekleşmesini sağlayan dual kuaternionlar oldukça yaygın bir şekilde kullanılmaktadır. Örneğin bir fabrikada otomobil parçalarının montajının yapıldığını düşünelim. Her bir parçaya dönme ve öteleme dönüşümünü tatbik edebilmek için dönüşümün parçalara belirli bir sırada uygulanması gerekir. Bunun için dual kuaternionu uygulama amacımıza göre dizisel metodun yanında modüler metodu da kullanabiliriz.



Şekil 5.10 Bir vidanın modüler metod ile hareketi.

Çalışmamızda dual küresel bilek hareketini ele alacağımız için dual kuaternionu dönme operatörü olarak aldık. Dual küresel bilek hareketinde ki modüler metodu şöyle tanımlarız: \vec{R}_1 vidası ve \vec{M}_i eksenleri, \vec{M}_n eksenleri etrafında Θ_n kadar döndürülsünler. Dönüşmüş olan yeni vidayı \vec{R}_1^1 ve yeni eksenini \vec{M}_1^1 ile gösterelim. Oluşan yeni vida ve eksenleri tekrar \vec{M}_n eksenini etrafında Θ_n kadar döndürelim, (Bkz Şekil 5.10). Bu işleme \vec{R}_1^n vidasını elde edinceye kadar devam edelim. Yani dual kuaternion $i = 1, 2, \dots, n$ olmak üzere

$$Q_i^* = -\cos \frac{\Theta_k}{2} + \hat{m}_k^{i-1} \sin \frac{\Theta_k}{2} \quad , \quad k = p + 1 - i$$

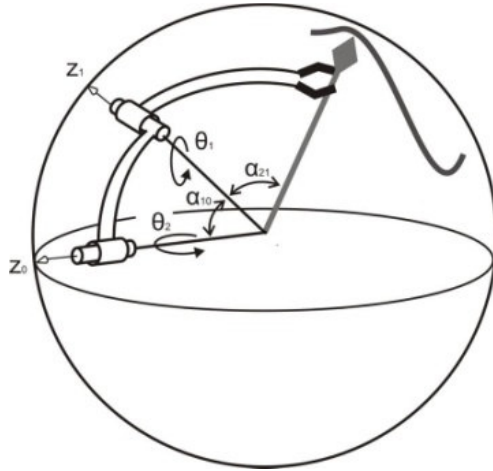
olarak tanımlayalım. \vec{R}_1 vidasını Θ_k dual açısı kadar Q_i dual kuaternionu ile döndürürsek $\vec{R}_1^i = Q_i^*(\vec{R}_1)Q_i^{*-1}$ vidaları elde edilecektir.

Aynı şekilde \vec{M}_i eksenlerinin Θ_k açısı ile Q_i dual kuaternionu ile döndürürsek $\vec{M}_j^i = Q_i^* M_j^{i-1} Q_i^{*-1}$ eksenlerini elde ederiz. Böylece dual kuaternionun dönme hareketi ile tanımlanan dual küresel dirsek hareketleri için modüler metodu aşağıdaki gibi üretmiş oluruz:

$$\vec{R}_1^n = Q_n^* Q_{n-1}^* \dots Q_2^* Q_1^* (R_1) Q_1^{*-1} Q_2^{*-1} \dots Q_{n-1}^{*-1} Q_n^{*-1}.$$

5.3.5 Dual küresel mekanizmalarda kuaterniyonlarla tanımlanan dirsek hareketi

Bilindiği gibi küresel mekanizmalarda küresel dirsek hareketleri makine mühendisliğinde yaygın bir şekilde kullanılmaktadır. Reel uzaydaki küre hareketinde birim vektörlerin reel açılarla döndürülmesi mümkündür. Küresel mekanizmasında reel küre üzerinde küresel dirsek hareketinin çizdiği yörünge küresel eğrilerden oluşmaktadır. Bu sonuç mekanik anlamda bir çok uygulamaya yol açmaktadır.



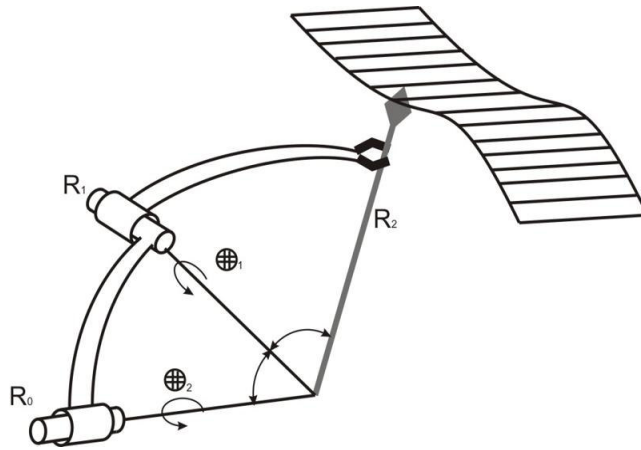
Şekil 5.11 Küresel mekanizmanın çizdiği küresel eğri.

Biz çalışmamızda doğruların dönüştürülmesini kullandığımız için reel uzaydaki kuaternionlarla çalışamayız. Bu yüzden dual uzayda birim dual vektör bir doğru (eksen) belirttiği için bir vidanın dual kuaternion yardımıyla hareketini ele almak zorundayız. Dual vektörlerin dizisel ya da modüler yöntem yardımıyla dönüştürülmesi sonucu dual küre oluşmaktadır. Dolayısıyla dual küresel mekanizmalarda dual kuaternionlarla tanımlanan dirsek hareketi tanımlanmış olur. Şekil 5.11 deki gibi 2 DOF dual küresel manipulatörünün verildiğini farz edelim.

Burada dizisel ya da modüler metodun kullanılmasıyla robot kolunun geldiği son durum incelenecektir. İlk olarak dizisel metodun kullanılmasıyla birinci dönmeden sonra robot kolunun konumunu inceleyelim. \vec{R}_2 vidasını Θ_1 dual açısı ile \vec{R}_1 etrafında ve sonra Θ_2 açısı ile \vec{R}_0 vidası etrafında döndürerek son konumunu belirleyelim. $\vec{R}_2' = Q_1(\vec{R}_1)Q_1^{-1}$ öyle ki burada $Q_1 = -\cos\frac{\Theta_1}{2} + \vec{R}_1 \sin\frac{\Theta_1}{2}$ şeklindedir, sonra $\vec{R}_2^2 = Q_2(\vec{R}_2)Q_2^{-1}$ olur. Burada $Q_2 = -\cos\frac{\Theta_2}{2} + \vec{R}_0 \sin\frac{\Theta_2}{2}$ dir. Bu iki birleştirirsek dizisel metodun robot kolunda oluşturduğu son konum $\vec{R}_2^2 = Q_2Q_1(\vec{R}_2)Q_1^{-1}Q_2^{-1}$ olur.

5.3.6 Dual küresel dirsek hareketleri sonucu oluşan yörünge yüzeyleri olarak regle yüzeyleri

Reel Küresel Dirsek hareketlerinde kuaternion yardımı ile döndürülen her bir robot kolunun ya da mafsallın hareket esnasında reel küre üzerinde çizmiş olduğu bir küresel eğri vardır. Bu yörünge eğrilerinin mühendislikte mekanizmalardaki kinematik analiz için yaygın bir şekilde kullanıldığını biliyoruz. Dual kinematik uzaya geçince kinematikteki elemanlar farklı bir anlama sahip oluyorlar. Çalışmamızda görüyoruz ki robot kolları ya da mafsallar dual küre üzerinde bir eğri çizmektedir. Dual küre üzerinde bir nokta bir doğruya karşılık geleceği gibi bir eğri regle uzaya karşılık gelir. O yüzden dual küresel dirsek hareketlerinde dual kuaternion yardımı ile döndürülen her eksenin yani vidanın dual küre üzerinde çizdiği dual yörüngeler regle yüzeyleri olur, (Bkz Şekil 5.12).



Şekil 5.12 Dual küresel mekanizmanın çizdiği regle yüzey.

Bir bakıma \vec{R} vidasına uygulanan $Q = -\cos\frac{\Theta}{2} + M \sin\frac{\Theta}{2}$ dual kuaternionu için $\vec{R}' = Q(\vec{R})Q^{-1}$ vidası oluşur. Dual küresel dirsek hareketinin her t anı için düşünersek $\vec{R}'(t) = Q(\vec{R}(t))Q^{-1}$ denklemi bir regle yüzeyinin denklemini verir. Böylece kinematikte regle yüzeylerinin imalatı için farklı bir bakış açısı elde ediyoruz. Eğer kullanılmak istenen metot modüler metot ise mafsalsın son konumunu bulmak için \vec{R}_1 ve \vec{R}_2 birim dual vidaları Θ_2 dual açısı ile \vec{R}_0 etrafında $Q_1 = -\cos\frac{\Theta_2}{2} + \vec{R}_0 \sin\frac{\Theta_2}{2}$ dual kuaternionu ile döndürdüğümüzde

$$\vec{R}_1^{1*} = Q_1^*(\vec{R}_1)\vec{R}_1^{*-1}, \quad \vec{R}_2^{1*} = Q_1^*(\vec{R}_2)\vec{R}_1^{*-1}$$

birim dual vidaları elde edilir. 2 DOF olduğu için tekrar \vec{R}_2^{1*} vidasını, Θ_1 dual açısı ile tanımlanan $Q_2 = -\cos\frac{\Theta_1}{2} + \vec{R}_1^{1*} \sin\frac{\Theta_1}{2}$ dual kuaternionu ile gönderdiğimizde son konumdaki doğruları elde ederiz:

$$R_2^{2*} = Q_2^*(\vec{R}_2^{1*})Q_2^{*-1}.$$

Buradan görüldüğü gibi robot kolunun son konumunu belirlemek için dizisel veya modüler metot kullanarak farklı yöntemlerle konum tespiti yapabiliriz. Benzer çalışmayı 3 DOF lu küresel mekanizmalara uygulamak mümkündür. Aynı sonuçlar genelleştirilmiş hali ile bulunur.

5.3.7 Sonuç

Bu çalışmadan görülüyor ki robot kollarının konumunun tespit edilmesi için kullanılan dizisel ve modüler yöntemlerin dual kinematik uzaylarında da çalışılabilirdiği görülmektedir. Regle yüzeyler robotikte ve BDGT da önemli bir rol oynamaktadır. Birçok sanayi makinelerinde kullanılan küresel dirsek hareketleri ile küresel eğriler üretilirken, çalışmamızda dual küresel dirsek hareketleri ile regle yüzeyler imal etmiş oluyoruz. Böylece çalışmamızın, dual küresel dirsek hareketinde oluşan yörüngelerin mekanik, robotik uygulamalarda ve BDGT da yapılan çalışmalara geometrik bazlı bir köprü olabileceği düşüncesi ortaya çıkmaktadır, (Kusak and Çalışkan, 2011b).

6. GALİLE DÜZLEMİ

6.1 Galile Düzlemi Temel Kavramları

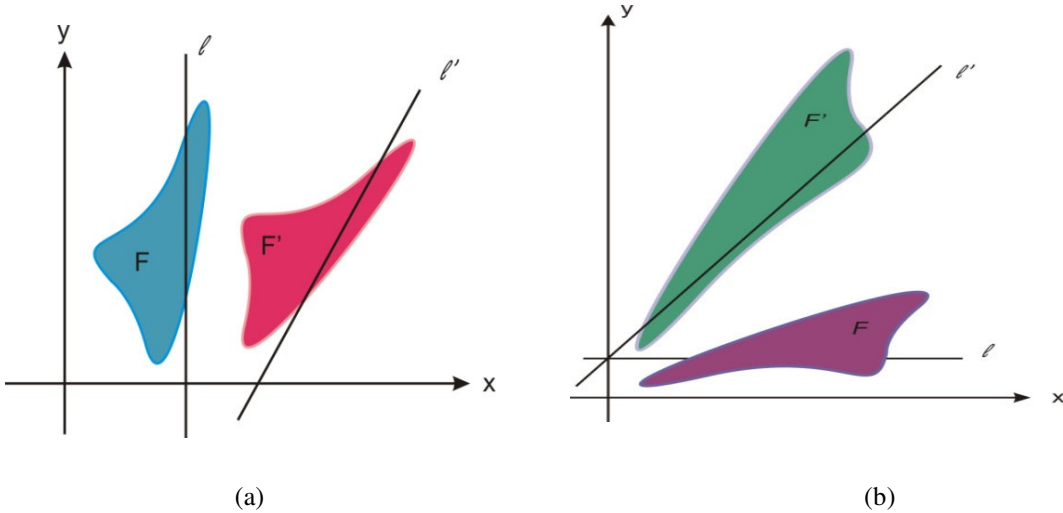
Tanım 6.1.1: Galile geometrisinde

$$x' = x + a \quad y' = vx + y + b \quad (6.1.1)$$

denklemleri ile verilen dönüşüme iki boyutlu Galile düzleminin “**Galile dönüşümü**” denir.

Tanım 6.1.2: Galile dönüşümü; $(x' = x, y' = vx + y)$ “**shear**” ve $(x' = x + a, y' = y + b)$ “**öteleme**” gibi iki hareketin birleşiminden oluşmaktadır.

Bu bölümde Galile dönüşümü ile belirtilen hareketin xoy düzlem geometrisi incelenecektir. (6.1.1) dönüşümü altında invaryant kalan xoy düzlemindeki şekillerin özellikleri ele alınacaktır. Doğal olmayan bu geometride şekillerin bu özellikleri geometrik öneme sahiptir. Ayrıca bu çalışmada Galile relativite prensipleri ile mekanik bağlantının dışında kalarak sadece dönüşümün geometrik yapısı incelenecektir. Bu durumda geometrik özellikler bir boyutlu kinematiğin özelliklerini belirler. Galile geometrisinin temel içeriklerini incelemeden önce (6.1.1) hareketinin temel özelliklerini listelemek yararlı olacaktır.



Şekil 6.1 Galile dönüşümünün etkisi (a) Öklid düzleminde l dik bir doğru ve l' herhangi bir doğru (b) l doğrusunun ve l' doğrusunun Galile düzlemindeki karşılığı (Yaglom, 1979)

Doğrular, paralel doğrular, doğru parçalarının alanları sadece Öklid geometrisinde değil aynı zamanda Galile geometrisinde de önemlidir. Şekil 7.1 de görüldüğü

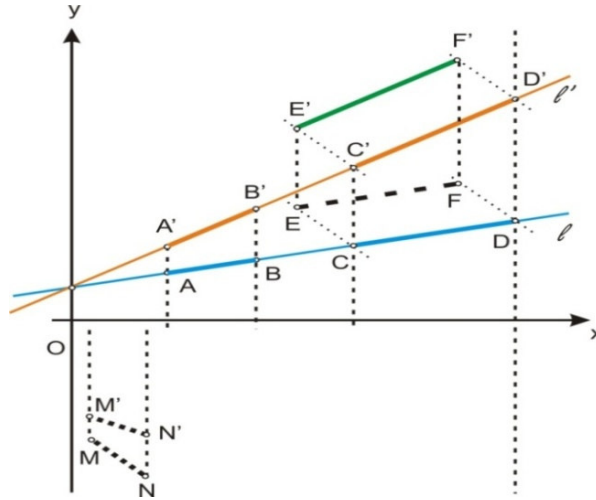
gibi Galile dönüşümü ile y eksenine paralel olan doğrular yine y eksenine paralel olan doğrulara dönüşür. Öklid geometrisinde “ y -eksenine paralel doğru” ifadesinin hiçbir geometrik önemi yok iken Galile geometrisinde y eksenine paralel doğruların diğer doğrulara göre çok daha önemli ve farklı bir rolü vardır.

Tanım 6.1.3: Galile geometrisinde y -eksenine paralel olmayan doğrulara “**sıradan doğrular**” denir. Bu yüzden Galile geometrisinde herhangi bir doğru denince y -eksenine paralel olmayan doğrular akla gelmektedir.

Tanım 6.1.4: Galile geometrisinde y -eksenine paralel olan doğrulara ise “**özel doğrular**” denir.

Teorem 6.1.1: (İnvaryant Özelliği) Galile dönüşümü hem öteleme hem de shearing dönüşümleri altında invaryant kalır. Ohalde Galile dönüşümü

- 1) Doğruları doğrulara
- 2) Paralel doğruları paralel doğrulara
- 3) Doğru parçalarını doğru parçalarına
- 4) Şekilleri aynı alanlı şekillere dönüştürür (Bkz. Şekil 6.2).



Şekil 6.2 Galile dönüşümünün doğru parçalarına etkisi (Yaglom, 1979).

Örnek 6.1.1: $y = 2x + 4$ doğrusunu sağlayan iki $(0,4)$ ve $(-2,0)$ noktasının reel galile matris dönüşümü altındaki görüntüsünü $v = 2$ $a = 1$ $b = 3$ değerleri için hesaplatırıp PASCAL ve GSP programları yardımıyla algoritmasını buldurup

Grafik görüntülerini çizdirelim. Burada v sabiti shearing, a ve b sabitleri x ve y eksenindeki ötelemeyi göstermektedir (Bkz. Şekil 6.3 ve 6.4) (Bkz Ek 4).

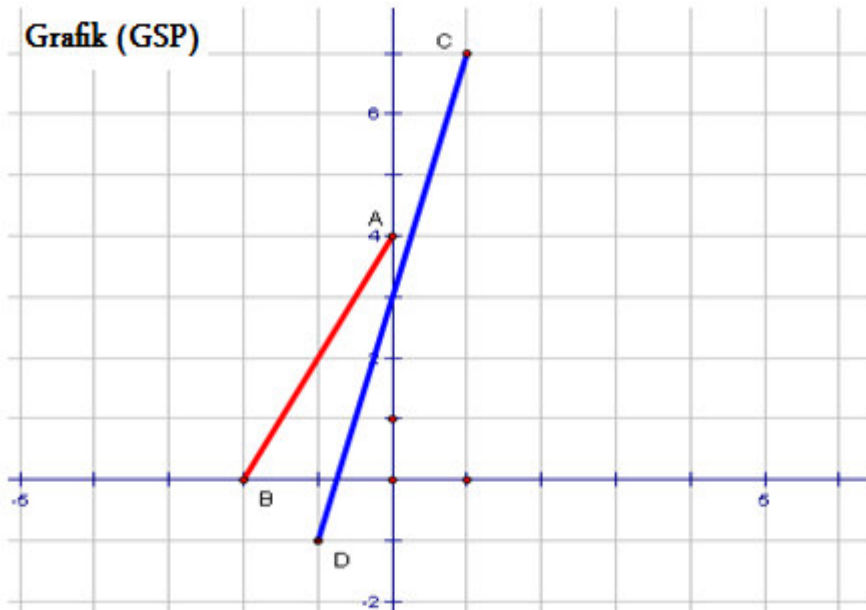
Ekran Görüntüsü(PASCAL):

```

C:\TP\BIN\TURBO.EXE
sırasıyla v, a ve b sayılarını giriniz
2
1
3
Orijinal noktalar
(0,4)
(-2,0)
yeni noktalar
(1,7)
(-1,-1)

```

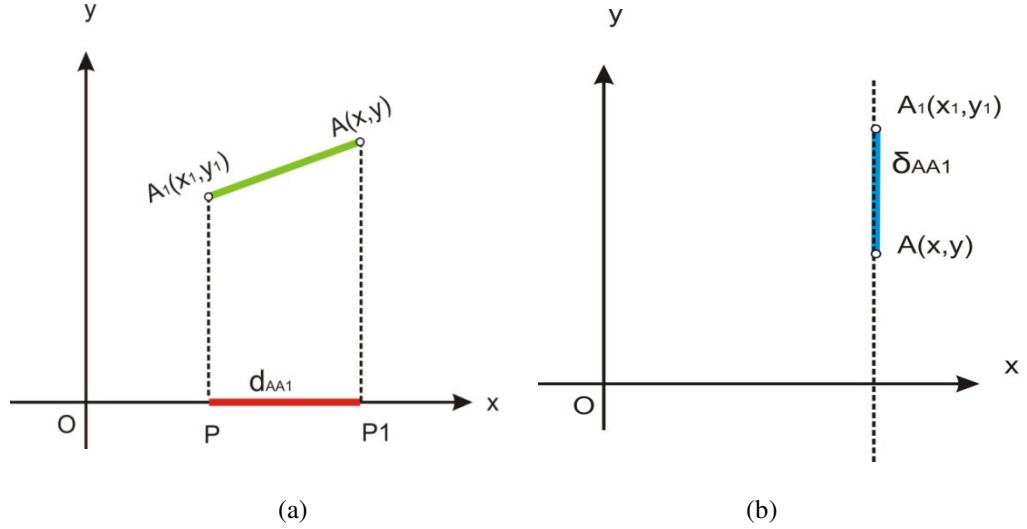
Şekil 6.3 Galile dönüşümü için PASCAL programında bir örnek (Bkz Ek 4) (Yaglom, 1979).



Şekil 6.4 Öklid düzlemindeki AB doğru parçasının Galile düzlemindeki CD doğru

parçasına dönüştüren programın GSP de Grafik çıktısı.

Tanım 6.1.5: Galile düzleminde alınan $A(x, y)$ ve $A_1(x_1, y_1)$ noktaları için $d_{AA_1} = x_1 - x$ formülü Galile geometrisinde “iki nokta arasındaki uzaklığı” olarak tanımlanır. Bu uzaklık AA_1 parçasının x eksenine üzerindeki izdüşümü olan PP_1 nin uzunluğuna eşittir, (Bkz. Şekil 6.5a).



Şekil 6.5 Galile düzleminde (a) uzaklık (b) özel uzaklık (Yaglom, 1979).

Örnek 6.1.2: Galile düzleminde $A(1, -2)$ ve $A_1(3, 6)$ noktalarını alalım. Bu noktalar arasındaki Öklidyen anlamdaki uzaklık ile Galile anlamındaki uzaklığını bulalım. Öklid uzaklığı

$$d_{AA_1} = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} = \sqrt{(3 - 1)^2 + (6 - (-2))^2} = \sqrt{68}$$

ve Galile uzaklığı $d_{AA_1} = x_1 - x = 3 - 1 = 2$ olarak bulunur.

Uyarı 6.1.1: Dikkat edelim ki burada uzaklıklar negatif olabilir yani $d_{A_1A} = -d_{AA_1}$ olduğu kolayca görülebilir.

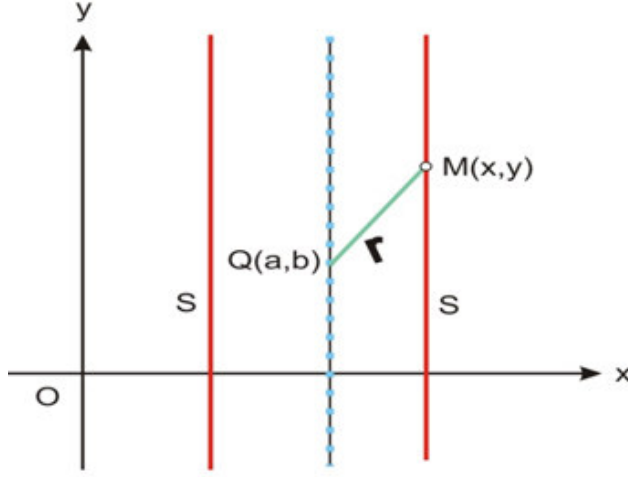
Uyarı 6.1.2: A noktasının x koordinatının shearing dönüşümü altındaki görüntüsü $x' = x + a$ formülü ile verildiğinden A_1 ve A noktalarının apsisi farkı olan $x_1 - x$ farkı da shearing dönüşümü altında değişmez (invariant) kalır.

Tanım 6.1.5: Eğer A ve A_1 arasında d_{AA_1} uzaklığı sıfır ise yani apsisi çakışık ($x_1 = x$) ise o halde noktalar aynı özel doğru üzerinde bulunurlar. Aynı özel doğru üzerinde bulunan noktalar arasındaki $\delta_{AA_1} = y_1 - y$ uzaklığına Galile düzleminde “**özel uzaklık**” denir.

Örnek 6.1.3: Galile düzleminde özel doğru üzerinde bulunan $A(1, 5)$ ve $A_1(1, 2)$ noktaları için özel uzaklık $\delta_{AA_1} = y_1 - y = 2 - 5 = -3$ olarak bulunur. Görüldüğü üzere Galile uzayında özel uzaklık negatif olabilmektedir.

Teorem 6.1.2: Galile dönüşümünde uzaklık aynı kalır.

Tanım 6.1.6: Sabit bir Q noktasından sabit r uzaklıktaki $M(x,y)$ noktalar kümesine “Galile çemberi” denir. Galile çemberini $d_{QM} = r$ formülü ile gösterebiliriz (Bkz. Şekil 6.6).



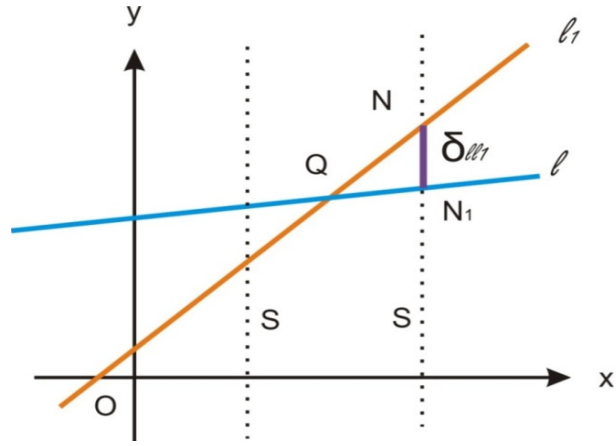
Şekil 6.6 Galile çemberi (Yaglom, 1979).

Tanım 6.1.6 : Q ve M noktaları arasındaki Galile uzaklığı d_{QM} olmak üzere $d_{QM}^2 = r^2$ denklemini ile “Galile çemberi” tanımlanır.

Ohalde $(x - a)^2 = r^2$ denklemini $p = -a$ ve $q = a^2 - r^2$ olmak üzere $x^2 + 2px + q = 0$ biçiminde düzenleyebiliriz. S çemberi Q merkezi r yarıçapı ile iki özel doğru üzerindeki noktalarını içerir. Bunların Q merkezinden S çemberi üzerindeki noktalara olan Öklid uzaklığına r , yarıçap denir. $r = 0$ olduğu durumda bu iki özel doğru çakışır. Şekil 6.6 dan da görüldüğü gibi S çemberinin merkezi sonsuz sayıda olabilir.

Sonuç: Galile çemberinde merkez değiştiğinde çember değişmez aynı kalır.

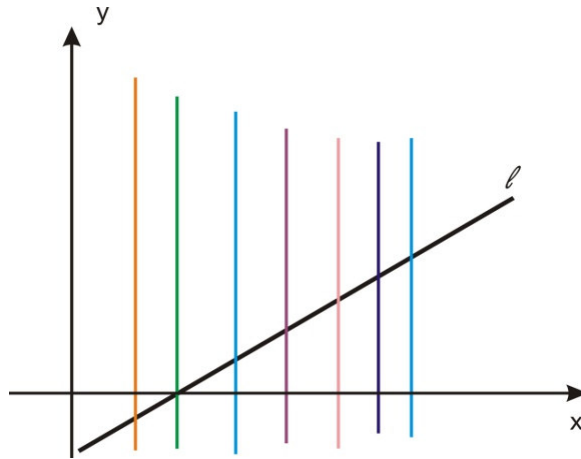
Tanım 6.1.7: $l_1 = y_1 = k_1x + s_1$ ve $l = y = kx + s$ olmak üzere Galile düzleminde kesişen iki doğru alınsın. Bu doğruların eğimleri $m_1 = k_1$ ve $m = k$ dir. Bu iki doğrunun m özel doğrusunu kestiği doğru parçasına “Galile açısı” denir. Galile açısı ayrıca bir özel uzaklık olduğu için δ_{l_1} ile gösterilir. Aynı zamanda $\delta_{l_1} = k_1 - k$ olarak da verilebilir (Bkz Şekil 6.7).



Şekil 6.7 Galile açısı (Yaglom, 1979).

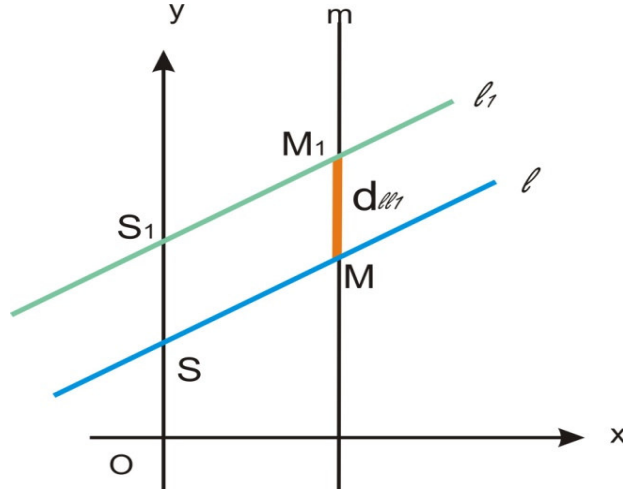
Örnek 6.1.4: Galile düzleminde $l_1 = y_1 = 3x + 1$ ve $l = y = 2x + 5$ olmak üzere iki doğru alalım. Bu doğruların eğimleri $m_1 = 3$ ve $m = 2$ dir. Galile açısı doğruların eğimlerinin farkına eşittir yani $\delta_{ll_1} = k_1 - k = 3 - 2 = 1$ dir.

Tanım 6.1.8: Galile düzleminde alınan $l_1 = y_1 = k_1x + s_1$ ve $l = y = kx + s$ doğruları arasındaki Galile açısı $\delta_{ll_1} = 0$ ise “**Galile düzleminde iki doğru birbirine paraleldir**” denir. Paralel durumunda herhangi bir özel doğruya ait olan d_{ll_1} uzaklığı oluşur ve $d_{ll_1} = s_1 - s$ şeklinde tanımlanır (Bkz Şekil 6.8).



Şekil 6.8 Galile düzleminde özel doğrular (Yaglom, 1979).

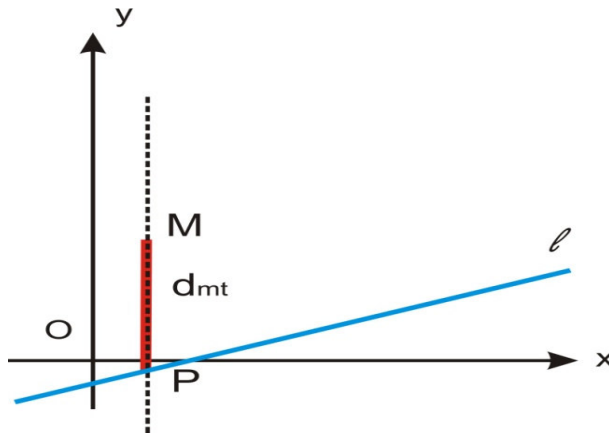
Tanım 6.1.9: Galile düzleminde $l_1 = y_1 = k_1x + s_1$ ve $l = y = kx + s$ iki paralel doğru olsun. Bu paralel doğruların belirttiği $d_{ll_1} = s_1 - s$ denklemi “**iki doğru arasındaki uzaklığı**” verir (Bkz. Şekil 6.9).



Şekil 6.9 Paralel iki doğru arasındaki uzaklık (Yaglom, 1979).

Örnek 6.1.5: Galile düzleminde $l_1 = y_1 = 2x + 4$ ve $l = y = 2x + 7$ iki paralel doğru alalım. Bu paralel doğruların belirttiği Galile anlamında iki doğru arasındaki uzaklık doğruların y eksenini kestiği noktaların ordinatlarının farkından oluşur yani $d_{ll_1} = s_1 - s = 4 - 7 = -3$ dir. Paralel iki doğru arasındaki uzaklığın Galile düzleminde negatif olabileceği görülmektedir.

Tanım 6.1.9: Galile düzleminde alınan $l = y = kx + s$ doğrusunu kesen y eksenine paralel olan tüm özel doğrular “ l doğrusuna Galile anlamında diktir” denir (Bkz. Şekil 6.10) (Yaglom, 1979).



Şekil 6.10 Galile düzleminde diklik (Yaglom, 1979).

6.2 Galile Düzleminde Dönüşümler

Tanım6.2.1: İki boyutlu Galile dönüşümünü reel anlamda homojen koordinatlardaki matris dönüşümleri

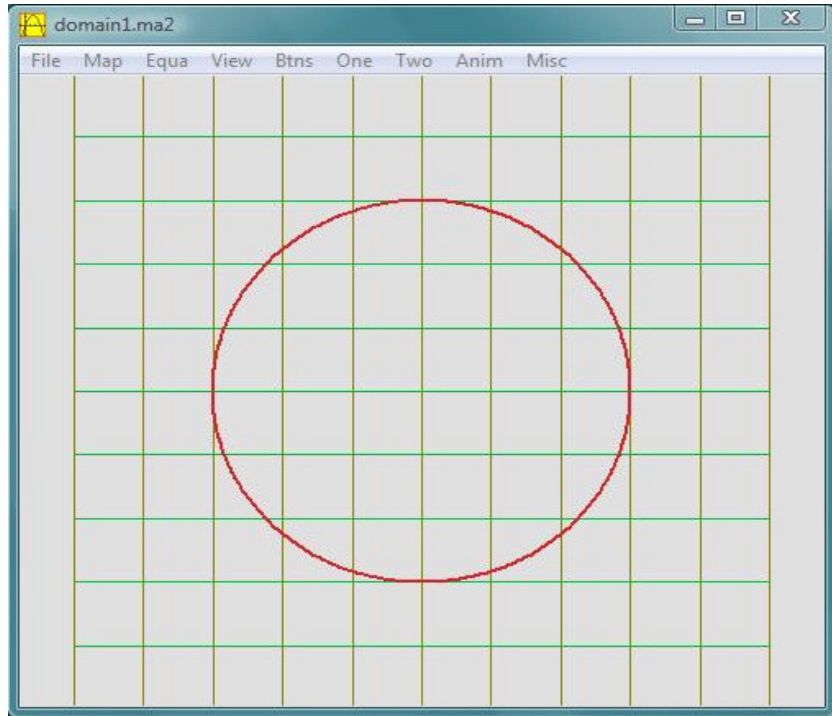
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a \\ v & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

matrisi ile ifade edilebilir (Tütüncü, 2009).

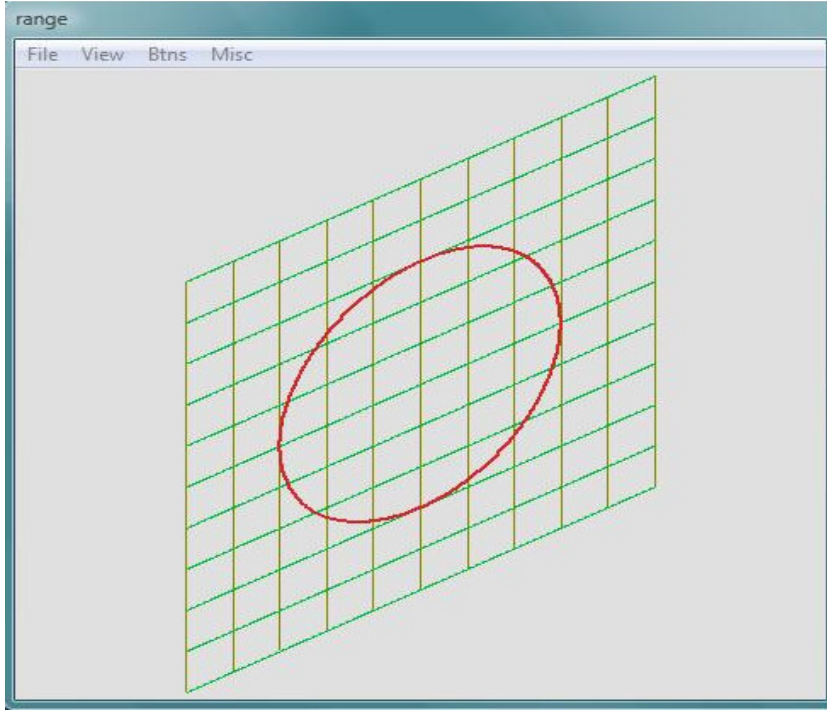
Tanım6.2.2: Galile dönüşümünü shearing matrisi $[S]_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ v & 1 \end{bmatrix}$ ve öteleme matrisi $[T]_{2 \times 1} = \begin{bmatrix} a \\ b \end{bmatrix}$ ile verilebilir. Ohalde Galile düzleminde verilen $P(x, y)$ noktasını $P'(x', y')$ noktasına dönüştüren Galile dönüşümünü $[P']_{2 \times 1} = [S]_{2 \times 2}[P]_{2 \times 1} + [T]_{2 \times 1}$ matris formunda yazabiliriz.

Şimdi de Galile dönüşümü ile ilgili bilgisayar uygulamasını vererek bir çemberin Galile dönüşümü altındaki görüntüsünü alalım:

Örnek 6.2.1: $(3\cos t, 3\sin t)$ çemberinin $v=0.5$ shear katsayısı ile verilen Galile dönüşümü altındaki görüntüsünü **Winplot** programı ile çizdirelim:



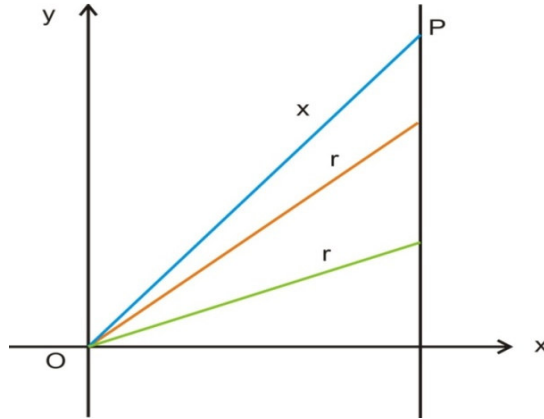
Şekil 6.11 Çemberin Öklid Düzlemindeki **Winplot** program görüntüsü



Şekil 6.12 Çemberin Galile Düzlemindeki **Winplot** program görüntüsü

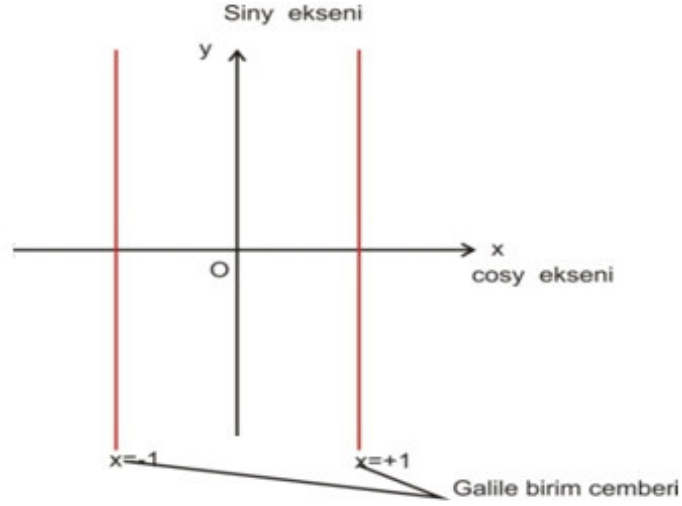
6.3 Galile Düzleminde Trigonometri

Tanım 6.3.1: Galile düzleminde sabit bir M noktasından 1 br uzaklıkta bulunan noktaların geometrik yerine “**birim Galile çemberi**” denir ve $S_G^1 = \{(1, y) \mid y \in \mathbb{R}\}$ ile gösterilir (Yaglom, 1979)(Bkz Şek.6.13) .



Şekil 6.13 Birim Galile çemberi (Yaglom, 1979)

Tanım 6.3.2: Galile birim çemberinde x apsisini üzerinde bulunduran eksene $\cos g$ ekseni, y ordinatını kapsayan eksene de **$\sin g$ ekseni** denir (Bkz Şek.6.14).



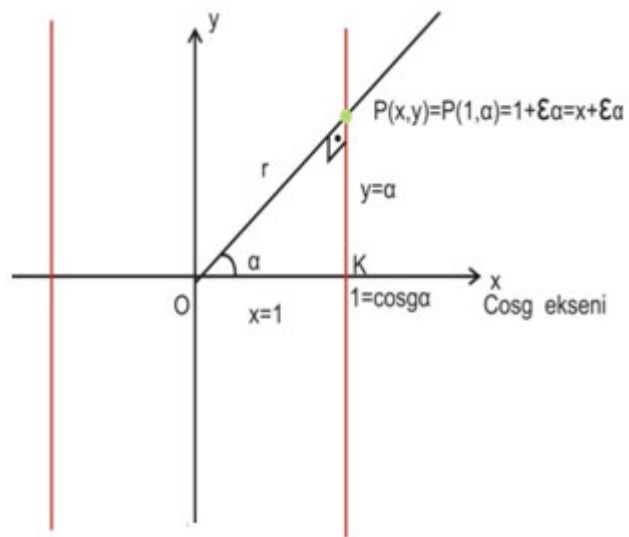
Şekil 6.14 Galile trigonometrik fonksiyonları (Yaglom, 1979).

Galile düzlemindeki açı Öklidyen anlamındaki açıdan farklı olduğu için Galile düzleminde trigonometrik bağıntılar farklılık gösterir. Şimdi de Galile düzlemindeki trigonometrik fonksiyonların tanımlarını verelim.

Tanım 6.3.5 Galile birim çemberi üzerinde herhangi bir nokta $P(x, y)$ ise bu noktadan x eksenine indirilen dikmenin ayağına, yani $\cos g\alpha = \text{izd}_x(P(x, y))$ ile tanımlı

$$\cos g : \mathbb{R} \rightarrow \{-1, +1\}, \alpha \rightarrow \cos g\alpha = 1$$

fonksiyonuna α açısının “galile kosinüsü” denir (Bkz Şek.6.15).

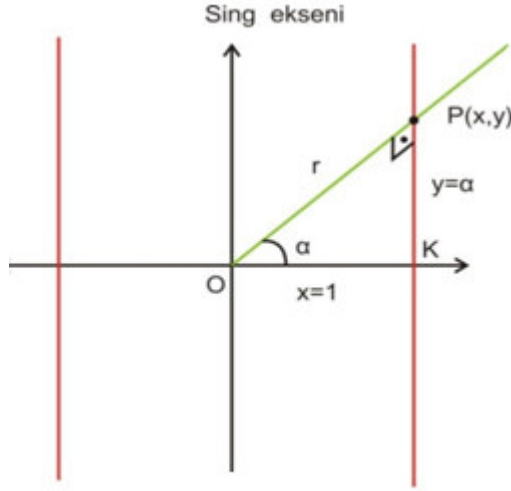


Şekil 6.15 Galile kosinüs fonksiyonu

Tanım 6.3.6: Galile birim çemberi üzerindeki bir $P(x, y)$ noktasının ordinatına, yani $\sin g \alpha = \text{izd}_y(P(x, y))$ ile tanımlı

$$\sin g : \mathbb{R} \rightarrow \mathbb{R}, \alpha \rightarrow \sin g \alpha = \alpha$$

fonksiyonuna “ α sayısının Galile sinüsü” denir. Bu, bir özdeşlik fonksiyonudur (Bkz Şek.6.16).



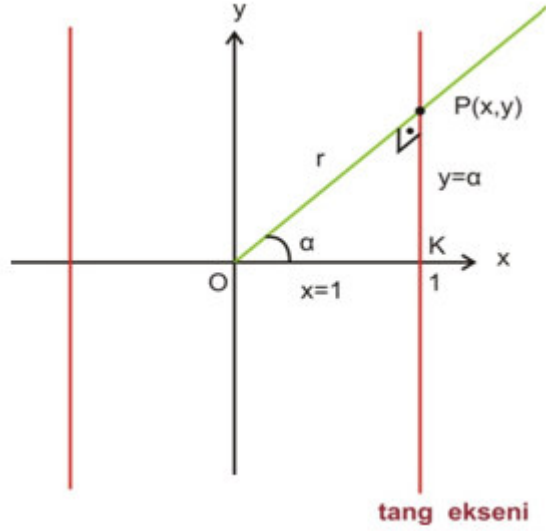
Şekil 6.16 Galile sinüs fonksiyonu

Tanım 6.3.7: $x = 1$ doğrusuna Galile düzleminde “**tanjant eksen**” denir.

Tanım 6.3.8: Galile birim çemberi üzerindeki bir $P(x, y)$ için bu noktayı orijine birleştiren $[OP]$ doğru parçasının tanjant eksenini kestiği noktanın ordinatına “ α açısının Galile tanjantı” denir. Diğer bir deyişle tanjant fonksiyonu

$$\tan g : \mathbb{R} \rightarrow \mathbb{R}, \alpha \rightarrow \tan g \alpha = \alpha$$

biçiminde tanımlı olup özdeşlik fonksiyonudur. Yani, $\sin g \alpha = \tan g \alpha = \alpha = I$ dir (Bkz Şek.6.17).



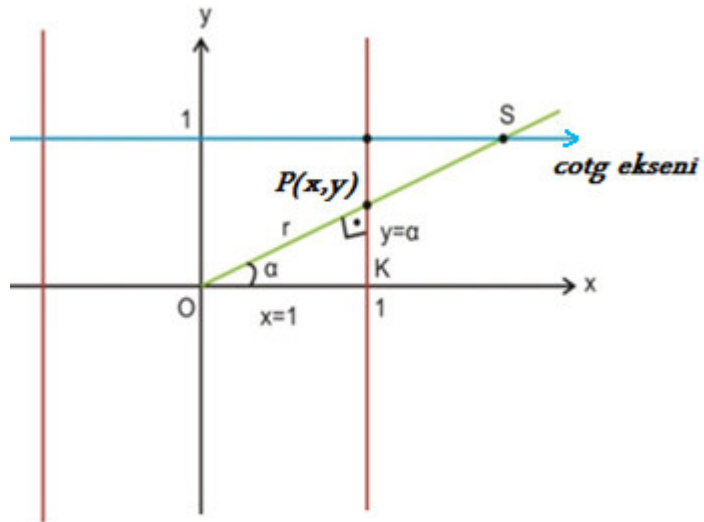
Şekil 6.17 Galile tanjant fonksiyonu

Tanım 6.3.9: $y = 1$ doğrusuna Galile düzleminde kottanjant Galile eksen denir.

Tanım 6.3.10: Galile birim çemberi üzerinde herhangi bir nokta $P(x,y)$ olsun. Bu noktanın apsisinin ordinatına oranına “ α açısının Galile kottanjantı” denir ve

$$\cot g : \mathbb{R} \rightarrow \mathbb{R}, \quad \alpha \rightarrow \cot g \alpha = \frac{1}{\alpha} \quad \alpha \neq (0,0)$$

şeklinde ifade edilir (Bkz Şek.6.18).



Şekil 6.18 Galile kottanjant fonksiyonu

Sonuç 6.3.1:

1) $\cos g\alpha = 1$

2) $\sin g\alpha = \alpha$

3) $\tan g\alpha = \alpha$

4) $\cot g\alpha = \frac{1}{\alpha}$

Özellikler 6.3.1:

1) $\tan g\alpha = \frac{\sin g\alpha}{\cos g\alpha}$

2) $\cot g\alpha = \frac{\cos g\alpha}{\sin g\alpha}$

3) $\tan g\alpha \cdot \cot g\alpha = 1$

4) $\tan g\alpha = \frac{1}{\cot g\alpha}$

5) $\cot g\alpha = \frac{1}{\tan g\alpha}$

6) $\sec g\alpha = \frac{1}{\cos g\alpha}$

7) $\csc g\alpha = \frac{1}{\sin g\alpha}$

Sonuç 6.3.2: Öklid düzlemindeki trigonometriden farklı olarak Galile düzlem trigonometrisinde

$$\sin g\alpha = \tan g\alpha$$

$$\sec g\alpha = \cos g\alpha$$

$$\cot g\alpha = \csc g\alpha$$

eşitlikleri geçerlidir.

Görüldüğü üzere, Öklid düzlemindeki trigonometrik değerler ile Galile düzlemindeki trigonometrik değerler farklı yapıdadır. Öklid düzlemindeki bazı açılarının trigonometrik değerleri aşağıdaki gibidir.

radyan	sin	cos	tan	cot
0	0	0	0	tanımsız
$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{3}$	$\sqrt{3}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	1	1
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{2}$	$\sqrt{3}$	$\frac{\sqrt{3}}{3}$
$\frac{\pi}{2}$	1	0	tanımsız	0

Çizelge 6.1 Öklid uzayında bazı trigonometrik değerler

Galile düzlemindeki bazı açılarının trigonometrik değerleri ise aşağıdaki gibidir:

radyan	sing	cosg	tang	cotg
0	0	1	0	tanımsız
$\frac{\pi}{6}$	$\frac{\pi}{6}$	1	$\frac{\pi}{6}$	$\frac{6}{\pi}$
$\frac{\pi}{4}$	$\frac{\pi}{4}$	1	$\frac{\pi}{4}$	$\frac{4}{\pi}$
$\frac{\pi}{3}$	$\frac{\pi}{3}$	1	$\frac{\pi}{3}$	$\frac{3}{\pi}$
$\frac{\pi}{2}$	$\frac{\pi}{2}$	1	$\frac{\pi}{2}$	$\frac{2}{\pi}$

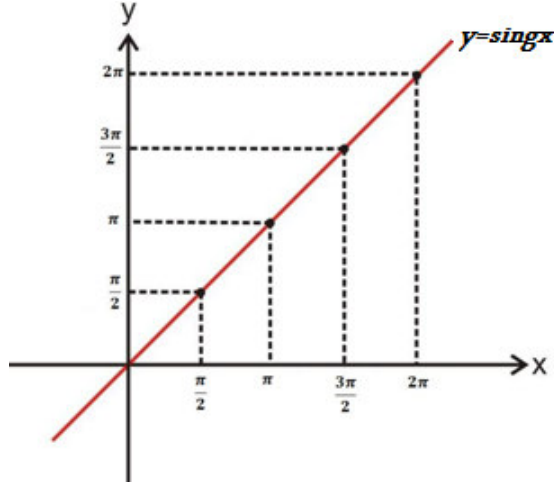
Çizelge 6.2 Galile düzleminde bazı trigonometrik değerler

Çizelge 6.1 ve 6.2 den görüldüğü üzere Öklid düzlemindeki trigonometrik değerler Galile düzleminde farklı anlamlara sahip olurlar. Şimdi de Galile düzlemindeki trigonometrik fonksiyonların grafiklerini ele alalım.

1) $y = I = \sin gx = x$ özdeşlik fonksiyonu:

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
sin gx	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π

Çizelge 6.3 Galile uzayında sing fonksiyonunun trigonometrik değerleri

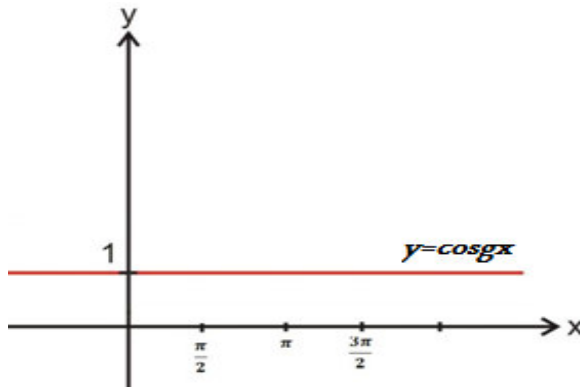


Şekil 6.19 sing fonksiyonunun grafiği

2) $y = \cos gx = 1$ fonksiyonu:

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
cos gx	1	1	1	1	1

Çizelge 6.4 Galile uzayında cosg fonksiyonunun trigonometrik değerleri

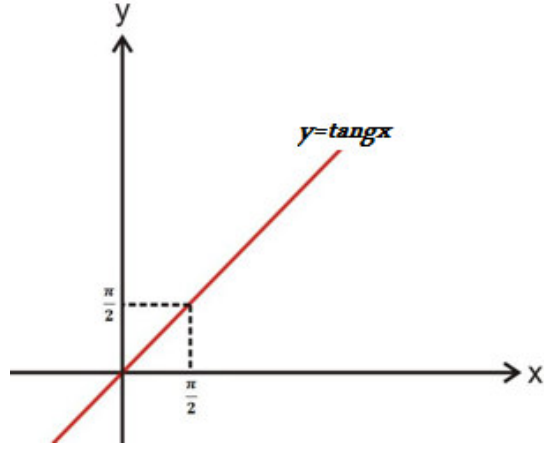


Şekil 6.20 Galile cosg fonksiyonunun grafiği

3) $y = I = \tan gx = x$ özdeşlik fonksiyonu:

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
tan gx	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π

Çizelge 6.5 Galile uzayında tang fonksiyonunun trigonometrik değerleri

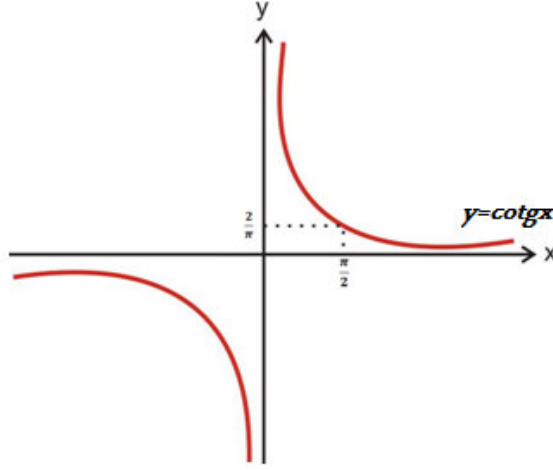


Şekil 6.21 Galile tang fonksiyonunun grafiği

4) $y = \cot gx = \frac{1}{x}$ fonksiyonu:

x	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$
cot gx	<i>tanımsız</i>	$\frac{2}{\pi}$	$\frac{1}{\pi}$	$\frac{2}{3\pi}$

Çizelge 6.6 cotg fonksiyonunun trigonometrik değerleri



Şekil 6.22 Galile cotg fonksiyonunun grafiği

Galile düzleminin trigonometrik eşitlikleri aşağıdaki gibi verilmiştir.

Teorem 6.3.1: Galile düzleminde verilen iki açı için

$$\sin g(\alpha + \beta) = \sin g\alpha \cdot \cos g\beta + \cos g\alpha \cdot \sin g\beta$$

$$\cos g(\alpha + \beta) = \cos g\alpha \cdot \cos g\beta - \sin g\alpha \cdot \sin g\beta$$

formülleri “**toplam ve fark formülleri**” olarak adlandırılır (Yaglom, 1979).

Sonuç 6.3.3: Galile düzlemindeki toplam ve fark formülleri kısaca aşağıdaki eşitliklerle ifade edilebilir:

$$\sin g(\alpha \mp \beta) = \alpha \mp \beta, \quad \cos g(\alpha \mp \beta) = 1$$

$$\tan g(\alpha \mp \beta) = \alpha \mp \beta, \quad \cot g(\alpha \mp \beta) = \frac{1}{\alpha \mp \beta}$$

Teorem 6.3.2: Galile düzlemindeki trigonometride

$$\sin g2\alpha = 2 \sin g\alpha \cdot \cos g\alpha, \quad \cos g2\alpha = 2 \cos g\alpha^2 - 1$$

formülleri “**Galile yarım açı formülleri**” olarak verilir (Yaglom, 1979).

Teorem 6.3.3: Galile düzlemindeki trigonometride

$$\sin gx \mp \sin gy = 2 \sin g \frac{x \mp y}{2} \cdot \cos g \frac{x \pm y}{2}$$

formülleri “**Galile dönüşüm formülleri**” olarak verilir (Yaglom, 1979).

Teorem 6.3.4: Galile düzleminde

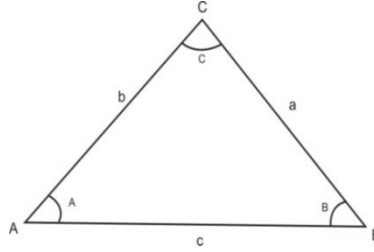
$$\frac{a}{A} = \frac{b}{B} = \frac{c}{C} = 2R$$

formülü “**Galile üçgeninde sinüs teoremi**” olarak verilir (Bkz. Şekil 6.23) (Yaglom, 1979).

Teorem 6.3.5: Galile düzleminde

$$a^2 = b^2 + c^2 - 2bc$$

formülü “**Galile üçgeninde kosinüs teoremi**” olarak adlandırılır (Bkz. Şekil 6.23) (Yaglom, 1979).



Şekil 6.23 Galile düzleminde bir üçgenin kenar uzunlukları ve açıları (Kolar et al., 2005)

Teorem 6.3.6: Galile düzleminde

$$\frac{b+c}{b-c} = \frac{B+C}{B-C}$$

formülü “**Galile üçgeninde tanjant teoremi**” olarak adlandırılır (Bkz. Şekil 6.23).

İspat: $\frac{b+c}{b-c} = \frac{\tan g \frac{B+C}{2}}{\tan g \frac{B-C}{2}}$ eşitliğinden dolayı $\frac{b+c}{b-c} = \frac{\frac{B+C}{2}}{\frac{B-C}{2}}$ ve $\frac{B+C}{B-C}$ dir.

Teorem 6.3.7: Galile düzleminde

$$A(\widehat{ABC}) = \frac{1}{2} a \cdot b \cdot C = \frac{1}{2} b \cdot c \cdot A = \frac{1}{2} a \cdot c \cdot B$$

formülü “**Galile üçgeninde alan formülü**” olarak adlandırılır (Bkz. Şekil 23) (Yaglom, 1979).

7. GENEL SONUÇLAR

Bu çalışmada;

- 1) Geometrik uzayların incelemesini sağlayacak olan dual sayı sistemi ve Clifford cebirlerinden kuaterniyon, dual kuaterniyon yapıları hakkında temel bilgiler verilmiştir.
- 2) Öklid uzayı ile ilgili bazı ön bilgiler verilmiştir.
- 3) Öklid uzayı ile mühendislik arasında bir ilişki kurabilen araç-tekerlek problemini konu alan mekanik bir problem geometrik açıdan incelenmiştir.
- 4) Öklid uzayında shear dönüşümü yardımı ile düzlemlerdeki deformasyonlar düşünülerek düzlemsel kinematiğe farklı bir bakış getirilmiştir.
- 5) Öklid uzayında sürekli olmayan geometrik yapılar için zaman skalasında vektör alanları ve konneksiyonlar üzerinde çalışılmıştır.
- 6) Dual uzaylar ve robotik hareketler hakkında temel kavramlar verilmiştir.
- 7) Robotikteki modüler ve dizisel metod vida hareketleri için ortaya konulmuştur.
- 8) Galile düzlemi ile ilgili temel bilgiler verilmiştir.
- 9) Galile düzleminde trigonometrik fonksiyonlar daha detaylı ele alınarak düzenlenmiştir.

Daha sonraki çalışmalarımızda Öklid uzayı, dual uzay, robotik hareketler, zaman skalası konusunda yapılan çalışmaların genişletilmesi düşünülebilir. Galile düzlemi ve uzayı ile ilgili yeni yapıları araştırıp ele aldığımız diğer uzaylar ve yeni uzaylarla Galile uzayının ilişkileri incelenebilir.

KAYNAKLAR DİZİNİ

Agarwal, P. and Bohner M., 1999, “Basic Calculus on Time Scales and Some of its Applications” Resultate der Mathematik, 35 (1-2), 3-22 pp.

Agrawal, P., 1985, Hamilton Operators and Dual number-quaternions in Spatial Kinematics Mech. Mach. Theory 22, 569-575 pp.

Aktan, N., Sarıkaya, M., Ilarslan, K. and Yıldırım, H., 2009, Directional Nabla-derivative and Curves on n-dimensional Time Scale, Acta Appl.Math. 105, 45-63 pp.

Alizade, R.I., Gezgin, E. and Kilit, Ö., 2005, A New Method in Computational Kinematics of a Spherical Wrist Motion Through Quaternions.

Barett, O.N., 1966, Elementary Geometry Differential, Academic Press, London.

Blaschke, W., 1949, Diferensiyel Geometri Dersleri, (Çeviri Erim, K.), Şirketi Mürettibiye Matbaası.

Bohner, M. and Guseinov, G.Sh., 2004, Partial differentiation on time scales, Dynamic Systems and Applications, Vol. 13, No:3-4, 351-379 pp.

Bohner, M. and Peterson, A., 2001, Dynamic Equations on Time Scales An Introduction with Applications, Birkhauser Boston.

David, R. and Alan, A.,1990, Mathematical Elements for Computer Graphics, McGraw-Hill.

Divjak, B., 1997, Geometrija Pseudogalilejevih Prostora, Ph. D. thesis, Un. of Zagreb.

Dönmez, A., 1999, Karmaşık Fonksiyonlar Kuramı, Beta Yayıncılık, İstanbul, Türkiye.

KAYNAKLAR DİZİNİ (devam)

- Goetz, A.**, 1970, Introduction to Differential Geometry, Notre Dame, Indiana.
- Gray, A.**, 1997, Osculating Circles to Plane Curves, Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd ed. Boca Raton, FL: CRC Press, 111-115 pp.
- Hacısalıhoğlu, H.H.**, 1983; Hareket Geometrisi ve Kuarterniyonlar Teorisi, Gazi Ün. Fen Edb. Fak. Yayınları, Ankara.
- Hamilton, R.**, 1844; On a Theory of Quaternions By William Rowan Hamilton, British Association Report, Part II, 2 p.
- Heinrich, G.**, 1963, Differential Geometry, McGraw-Hill, New York.
- Hillman, A.P., Alexanderson, G.L. and Newton, M. E.**, Complex Numbers and Trigonometry, <http://www.thiel.edu/mathproject/Cnat>, (Erişim Tarihi:10Mart 2008).
- Kayhan, G.**, 2003, Robot Kinematığının Esasları, Yüksek Lisans Semineri , Ondokuz Mayıs Üniversitesi Elektrik-Elektronik Müh. Bölümü.
- Kilit, Ö.**, 2005, Application of a Multi-stage Bogie Suspension System to the Planet Exploration Vehicles, Proceedings of 2nd International Conference on “Recent Advances in Space Technologies”, Turkey.
- Kolar, S.R., Kolar Z.B., Volenec, V. and Beban J.**, 2005, Metrical Relations in a Standard Triangle in an Isotropic Plane, Math. Com. 10, 149-157 pp.
- Köse, Ö.**, 1974, Çizgiler Uzayında Yörünge Yüzeyleri. Dok. Tezi Erzurum Atatürk Ün.

KAYNAKLAR DİZİNİ (devam)

Kuşak, H. and Çalışkan, A., 2011 a, “The Delta Nature Connection on Time Scale”, Journal of Mathematical Analysis and Applications; Vol. 375, Issue 1, 323-330.

Kuşak, H. and Çalışkan, A., 2011 b, “About Dual Spherical Wrist Motion and its Trajectory Surface as a Ruled Surface”, Mathematical and Computational Applications, Vol.16, No.1, 309-316 pp.

Kuşak, H. and Çalışkan, A., 2009, “Geometric Design of the Rover Wheels in 3D-Space”, Hadronic Journal 32, 433-444 pp.

Kuşak, H. and Çalışkan, A., 2008, "Applications of Vector Field and Derivative Mapping on Timescale" Hadronic Journal: 6 Vol. 31, 617-633 pp.

Müller, H. R., 1963, Kinematik Dersleri, A.Ü. Fen Fak. Yayınları.

Patel N., Ellery A., Welch C. and Curley A., 2002, Preliminary Analysis of Mobility and Suspension Systems for a Mars Micro Rover., 2nd World Space Congress, IAC-02-U.2.08, Houston, TX, London, U.K.

Paul, R.P., 1981, Robot Manipulators:Mathematics, Programming and Control, MIT Press.

Rogers, D.F. and Adams, J.A.,1989, Matemactical Elements For Comp.Graphics.

Sciavicco, L. and Siciliano, B., 1996, Modeling and Control of Robot Manipulators, McGraw-Hill International Publications.

KAYNAKLAR DİZİNİ (devam)

Shoemake, K., 1987, Animating Rotation with Quaternion Calculus, ACM SIGGRAPH, Course Notes10, Computer Animation: 3–D Motion, Specification, and Control.

Sipus, Z.M. and Divjak, B., 2003, Minding Isometries of Ruled Surfaces in Pseudo-Galilean Space, Journal of Geometry, 77, 35 – 47 pp.

Şhubi, E.S., 1988, Rijid Cisimler Dinamiği , İ.T.Ü. Fen-Edebiyat Fakültesi Matbaası.

Thianwiboon, M., Sangveraphunsiri, V. and Chanchaen, R., 2001, Rocker Bogie Suspension Performance, Proceeding of the 11.th International Pacific Conference on Automotive Engineering, University Bangkok, Thailand.

Tütüncü, E.E., 2009, Galile Uzaylarında Hareketlerin Geometrisi, Doktora Tezi, Ankara Üniversitesi.

Veldkamp, G.R., 1976, On the Use of Dual Numbers, Vectors and Matrices in Instantaneous, Spatial Kinematics, Mech. Mach. Theory 11, 141-156 pp.

Yaglom, I.M., 1979, A Simple Non-Euclidean Geometry and its Physical Basis. New York- Heidelberg- Berlin.

Yang, A.T and Freudenstein, F., 1964, Application of Dual Number Quaternion Algebra to the Analysis of Spatial Mechanism , Transactions of the ASNE , 300-308 pp.

Yang, A.T., 1963, Application of Quaternion Algebra and Dual Numbers to the Analysis of spatial Mechanism, Columbia University, Eng. Sc.D.,Ph.D. Dissertation.

ÖZGEÇMİŞ

Hatice KUŞAK 1981 yılı İzmir doğumlu olup, İzmir’de 1987-1995 yılları arasında Batı Çimento İlköğ., ve Bornova Ortaokulundan mezun olmuştur. 1995-1999 yılları arasında İzmir Kız Lisesinde Yabancı Dil Ağırlıklı olarak liseyi tamamlamıştır. Ege Üniversitesi lisans programından 2003 yılında mezun olmuştur. 2003-2005 yılları arasında Geometri Anabilimdalında yüksek lisansını tamamlamıştır. Doktorasına 2005 yılında Geometri Anabilimdalında başlamış, lisansüstü eğitimi süresince çeşitli özel eğitim kurumlarında matematik öğretmeni olarak çalışmıştır.

Akademik Çalışmalar

Diploma Çalışması, 2003, “Bilgisayar Grafikleri için Temel Matematik Bilgileri”, EGE Üniversitesi Fen Fakültesi, Danışman: Prof. Dr. Ali ÇALIŞKAN.

Yüksek Lisans Tezi, 2003-2005: “Dönüşümlerin Geometrik Elemanlara Etkisi”, EGE Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilimdalı, Danışman: Yrd. Doç. Dr. Bahadır TANTAY.

Doktora Tezi, 2005- 2011: “Geometrik Uzaylar ve Uygulamaları”, EGE Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilimdalı, Danışman: Prof. Dr. Ali ÇALIŞKAN.

Projeler (Doktora Tez Projesi): “Hiperkompleks Sayılar ve Doğrular Geometrisine Uygulamaları”, E.Ü.Bilimsel Araştırma Projeleri (BAP).

Uluslararası hakemli dergilerde yayınlanan makaleler

Kuşak, H. and Çalışkan, A., 2011; “The Delta Nature Connection on Time Scale”, Journal of Mathematical Analysis and Applications; Vol. 375, Issue 1, 323-330 (SCI, A sınıfı makale).

Kuşak, H. and Çalışkan, A., 2011, “About Dual Spherical Wrist Motion and its Trajectory Surface as a Ruled Surface”, Mathematical and Computational Applications, Vol.16, No.1, 309-316 (SCI, C sınıfı makale).

Kuşak, H. and Çalışkan, A., “Geometric Design of the Rover Wheels in 3D-Space”, Hadronic Journal 32, 433-444,2009, (Uluslararası makale).

Kuşak, H. and Çalışkan, A., "Applications of Vector Field and Derivative Mapping on Timescale" Hadronic Journal: 6 Vol. 31, PP. 617-633 , 2008, (Uluslararası makale).

Bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler ve posterler

Kuşak, H. and Çalışkan, A., “About Dual Spherical Wrist Motion and its Trajectory Surface as a Ruled Surface”, 1st International Symposium on Computer in Science &Engineering, June 3-5, 2010, (Uluslararası Bildiri).

Şahiner, B., Çalışkan, A. and Kuşak, H., “Dual De Casteljau Algorithm in Drawing Prism and Pyramid”, 1st International Symposium on Computer in Science &Engineering, June 3-5, 2010, (Uluslararası Bildiri).

Şahiner, B., Çalışkan, A. and Kuşak, H., “Drawing Platonic Solids With Dual Drawing Method”, 1st International Symposium on Computer in Science &Engineering, June 3-5, 2010, (Uluslararası Poster).

Gülsoy, F., Kuşak, H. and Çalışkan, A., “Calculating The Bending Energy of The K-Flex and Twaron Fabric by The Holditch Area Theorem”, 1st International Symposium on Computer in Science &Engineering, June 3-5, 2010, (Uluslararası Poster).

Gülsoy, F., Kuşak, H. and Çalışkan, A., “Kflex ve Twaron kumaşlarının bending enerjisinin Hesaplanması”, İYT Proje Pazarı, 2008, (Ulusal Poster).

Gülsoy, F., Kuşak, H., Karahan, M. and Çalışkan, A., “Thin Plate Spline yöntemiyle K-Flex ve Twaron kumaşları üzerinde bending enerjisinin incelenmesi”, 6. Matematik Sempozyumu (TOBB ETÜ Konferans Salonları Söğütözü Ankara), 29 Kasım-01 Kasım 2007, (Ulusal Bildiri).

EKLER

Ek 1 Kuaterniyon Cebri Hesap Makinesi Programının Kodları

Ek 2 Shearing Kinematik ile İlgili Mapple Kodları

Ek 3 Shear Dönüşümü Olmadan Normal Kinematığın Mapple Kodları

Ek 4 Galile Dönüşümlerinin Bilgisayar Uygulamalarının Program Kodları

Ek1: Kuaterniyon Cebri Hesap Makinesi Programının Kodları

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<PROCESS.H>
#define N 10

int i, j, k, n, c, ac, qal, aci, na;
double qcarpma[N], dmatrisi[N][N],
vstun[N][N], S[N][N], qa[N][N];
double rteta, rlteta, dteta, nrmluq[N], norm1, dogrlt[N], son;
double nd[N];
const double pi=3.14159;

/***** Fonksiyonların içinde kullanılan fonksiyonlar *****/

void qoku() {
    printf("quaternion sayısını gir:");
    scanf("%i", &n);
    return;
}

//.....

void sncuyari() {
    printf("\nSONUÇ İÇİN ENTER TUŞUNA
BASIN.....:\n\n"); getch();
    return;
}

//.....dönme matrisi.....
void dmtrs() {
    dmatrisi[1][1]=qa[1][0]*qa[1][0]+qa[1][1]*qa[1][1]-
qa[1][2]*qa[1][2]-qa[1][3]*qa[1][3];
    dmatrisi[2][2]=qa[1][0]*qa[1][0]-
qa[1][1]*qa[1][1]+qa[1][2]*qa[1][2]-qa[1][3]*qa[1][3];
    dmatrisi[3][3]=qa[1][0]*qa[1][0]-qa[1][1]*qa[1][1]-
qa[1][2]*qa[1][2]+qa[1][3]*qa[1][3];
    dmatrisi[1][2]=(2*qa[1][1]*qa[1][2])-(2*qa[1][0]*qa[1][3]);
    dmatrisi[2][1]=(2*qa[1][1]*qa[1][2])+(2*qa[1][0]*qa[1][3]);
```

```

dmatrisi[3][1]=(2*qa[1][1]*qa[1][3])-(2*qa[1][0]*qa[1][2]);
dmatrisi[1][3]=(2*qa[1][1]*qa[1][3])+(2*qa[1][0]*qa[1][2]);
dmatrisi[3][2]=(2*qa[1][2]*qa[1][3])+(2*qa[1][0]*qa[1][1]);
dmatrisi[2][3]=(2*qa[1][2]*qa[1][3])-(2*qa[1][0]*qa[1][1]);
printf("\n\nd"nme\a matrisi.....:\n\n");
for(i=1;i<4;i++){
    for(j=1;j<4;j++){
        printf("%.1f\t",dmatrisi[i][j]);
        printf("\n\n");
    }
return;
}

//.....
//(3*3) l k dönme dönüşüm ve (3*1) lik vektörün matris
çarpım

void matcrp(){
    for ( i=1;i<4;i++)
        for(j=1;j<2;j++) S[i][j]=0;
    for ( i=1;i<4;i++){
        for(j=1;j<2;j++)
            for(k=1;k<4;k++)
                S[i][j]+= dmatrisi[i][k] * vstun[k][j];
    }
    printf("döndürülmüş vektör.....:\n\n");
    printf("dv= ( %.1f , %.1f , %.1f
)\n\n",S[1][1],S[2][1],S[3][1]);
    getch();
    return;
}

//.....
void sonuc(){
    printf("\ndöndürülecek vektör.....:\n\n");
    printf(" v= ( %.1f , %.1f , %.1f )
\n\n",vstun[1][1],vstun[2][1],vstun[3][1]);
    return;
}

//dönme uygulanacak 3D lu vektörün girişi
void vektor(){
    int ve;

```

```

printf("\n\n döndürülecek vektörü girin\n\n");
for (i=1;i<4;i++) {
    printf(" v[%i]=",i);
    scanf("%i",&ve);
    vstun[i][1]=(double)ve;
}
sonuc();
return;
}
//.....
void acibul(){
    rlteta=acos(aci);
    rteta=2*rlteta;
    dteta=(double)(rteta/pi)*180;
    printf("\n.....Bu kuaternionun dönme
açısı .....:\n");
    printf("\n          radyan olarak: %.1f \n ",rlteta);
    printf("\n          nderece olarak: %.1f \n ",dteta);
}

//..... Çarpma Matrisi .....//

void carpmtrs(){
    qcarpma[0]=qa[1][0]*qa[2][0]-qa[1][1]*qa[2][1]-
qa[1][2]*qa[2][2]-qa[1][3]*qa[2][3];

    qcarpma[1]=qa[1][0]*qa[2][1]+qa[1][1]*qa[2][0]+qa[1][2]*qa[2][
3]-qa[1][3]*qa[2][2];

    qcarpma[2]=qa[1][0]*qa[2][2]+qa[1][2]*qa[2][0]+qa[1][3]*qa[2][
1]-qa[1][1]*qa[2][3];

    qcarpma[3]=qa[1][0]*qa[2][3]+qa[1][3]*qa[2][0]+qa[1][1]*qa[2][
2]-qa[1][2]*qa[2][1];
    return;
}

//***** bir kuaterniyon girişi *****/

void qqgir(){
    if (c==0) qoku();
}

```

```

        else goto label3;
label3:
    for (i=1;i<n+1;i++) {
        for(j=0;j<4;j++){
            printf(" q[%i][e%i]=",i,j);
            scanf("%i",&qal);
            qa[i][j]=qal;
        }
        printf("q[%i]= %.2f e0 + %.2f e1 + %.2f e2 + %.2f e3
\n",i,qa[i][0],
            qa[i][1],qa[i][2],qa[i][3]);
    }
    return;
}

/***** iki kuaterniyonun toplam *****/

void qtopla(){
    double qtoplams[N];
    printf("\n\n.....TOPLAMA işlemiYapılacaktır.....\n");
    qoku();
    qqgir();
    for (i=0;i<4;i++){
        qtoplams[i]=0;
        for(j=1;j<n+1;j++)
            qtoplams[i]+=qa[j][i];
    }
    sncuyari();
    printf( "\n\n qtoplama= %.1f e0 + %.1f e1 + %.1f e2 + %.1f e3
\n " ,qtoplams[0],qtoplams[1],qtoplams[2],qtoplams[3]);
    return ;
}

/***** iki quaternionun fark *****/

void qfark(){
    double qfark[N];
    n=2;
    printf ("\n\n.....ÇIKARMA işlemi
Yapılacaktır!.....\n");
    printf("\n\n q[1]-q[2] bulabilmek için önce q[1] sonra
q[2] yi giriniz! \n\n");

```

```

    qqgir();
    for (i=0;i<4;i++)
        qfark[i]=qa[1][i]-qa[2][i];
    sncuyari();
    printf(" q[1]-q[2]= %.1f e0 + %.1f e1 + %.1f e2 + %.1f e3
\n " ,qfark[0],qfark[1],qfark[2],qfark[3]);
return ;
}
/***** Bir kuaterniyonun skalerle çarpılması *****/
void qscarp(){
    int s;
    double sq[N];
    printf ("\n\n...SKALERLE ÇIKARMA işlemi
Yapılacaktır!.....\n");
    printf("\nscalari gir:");
    scanf("%i",&s);
    printf("\n q yu girin:\n\n");
    n=1;
    qqgir();
    for (i=1;i<2;i++)
        for (j=0;j<4;j++)
            sq[j]=s*qa[i][j];
    printf("qscarpim= %.1f e0 + %.1f e1 + %.1f e2 + %.1f e3 \n"
,sq[0],sq[1],sq[2],sq[3]);

return;
}
/***** bir quaternionun eşleniği*****/

void qeslenik(){
    int qal;
    double vq[N];
    printf ("\n\n.....EŞLENİK ALMA işlemi
Yapılacaktır!.....\n");
    printf("\n eşleniği alınacak kuaternionu giriniz! \n\n");
    n=1;
    qqgir();
    for (int i=1;i<2;i++)
        for (j=1;j<4;j++)
            vq[j]=-qa[i][j];
    sncuyari();
}

```

```

    printf( " Kq= %.1f e0 + %.1f e1 + %.1f e2 + %.1f e3 \n "
,qa[1][0],vq[1],vq[2],vq[3]);
return ;
}

```

```

/***** bir quaternionun normu *****/
void qnorm(){
    float qkaretop,qnrm;
    printf ("\n\n.....NORM ALMA işlemi
Yapılacaktır!.....\n");
    printf("\n normu alınacak kuaternionu giriniz! \n\n");
    n=1;
    qqgir();
    qkaretop=0;
    for(i=1;i<2;i++)
        for (j=0;j<4;j++)
            qkaretop+=qa[i][j]*qa[i][j];
    qnrm=sqrt(qkaretop);
    sncuyari();
    printf( " norm=karekök(%.2f) yani norm= %.2f \n "
, qkaretop, qnrm);
    return ;
}

```

```

/***** iki quaternionun quaternion çarpım *****/

void qcarp(){
    printf ("\n\n.....ÇARPMA işlemi Yapılacaktır!.....\n");
    printf("\n q[1]*q[2] bulabilmek in "nce q[1] sonra q[2] yi
giriniz! \n\n");
    n=2;
    qqgir();
    carpmtrs();
    sncuyari();
    printf( " q[1]*q[2]= %.2f e0 + %.2f e1 + %.2f e2 + %.2f e3
\n ",
            qcarpma[0],qcarpma[1],qcarpma[2],qcarpma[3]);
    return ;
}

```

```

/***** bir quaternionun tersi *****/
void qters() {
    printf ("\n\n.....TERS ALMA işlemi
Yapılacaktır!.....\n");
    float qkaretop, qnrm, vq[N], ksq;
    n=1;
    printf("\n tersi alınacak kuaternionu giriniz! \n\n");
    qggir();
    qkaretop=0;
    for (i=0;i<2;i++)
        for (j=0;j<4;j++)
            qkaretop+=qa[i][j]*qa[i][j];
    qnrm=sqrt(qkaretop);
    if (qnrm!=0){
        ksq=qa[1][0]/qnrm;
        for (i=1;i<4;i++)
            vq[i]=(-qa[1][i])/qnrm;
        sncuyari();
        printf("qters= %.2f e0 + %.2f e1 + %.2f e2 + %.2f e3
\n", ksq, vq[1], vq[2], vq[3]);
    }
    else
        printf("norm sıfırken ters alınamaz!");
    return ;
}

/***** vektörel quaternionun üssünü
alma*****/
void qus() {
    int a;
    float k2, k3, qus[N];
    float t, c, k1;
    n=1;
    printf ("\n\n.....KUVVET ALMA işlemi
Yapılacaktır!.....\n");
    printf("\n.....Kuvvet Alma için vektörel q yu
girin.....\n\n");
    qggir();
    qa[1][0]=0;
    k1=0-qa[1][1]*qa[1][1]-qa[1][2]*qa[1][2]-qa[1][3]*qa[1][3];
    printf("\n\nkağ nc dereceden us alınsın?.....:");
}

```

```

scanf("%i",&a);
if ((a%2)==0){
    c= (a/2);
    k3=pow(k1,c);
    printf("\n\n q^n de n=%i ss n çift olarak
girdiniz!\n",a);
    qus[0]=k3;
    qus[1]=0;
    qus[2]=0;
    qus[3]=0;
    sncuyari();
    printf("q s= %.2f e0 + %.2f e1 + %.2f e2 + %.2f e3
\n",qus[0],qus[1],qus[2],qus[3]);
}
else
{
    t=(a-1)/2;
    k2= pow(k1,t);
    printf("\n\n q^n de n=%i ss n tek olarak
girdiniz!\n",a);
    qus[0]=0;
    qus[1]=k2*qa[1][1];
    qus[2]=k2*qa[1][2];
    qus[3]=k2*qa[1][3];
    sncuyari();
    printf("q s= %.2f e0 + %.2f e1 + %.2f e2 + %.2f e3
\n",qus[0],qus[1],qus[2],qus[3]);
}
return ;
}
/***** açı verilerek döndürme *****/
void qdonmeaci(){

    printf ("\n\n...Açı Girilerek Döndürme
Yapılacaktır!...\n\n");
    printf("\n dönme açısını giriniz...: ");
    scanf("%i",&aci);
    //.....derece olan aciyi radyan
yapalım.....
    rteta=(aci*pi)/180;
    son=sin(rteta);

```

```

printf("\n %i dereceli a# %.2f radyand r.....sin s
: %.2f \n\n", aci, rteta, son);
//.....döndürme doğrultusunu
girelim.....
printf("\n dönme doğrultusunu\ a girin.....:\n\n");
for(j=1; j<4; j++){
printf(" \n nor[e%i]=", j);
scanf("%i", &na);
nd[j]=(double)na;
}
printf("\n n= ( %.2f , %.2f , %.2f )
\n\n", nd[1], nd[2], nd[3]);
//.....q kuaternionun
oluşturulması .....
qa[1][0]=cos(rteta/2);
qa[1][1]=nd[1]*sin(rteta/2);
qa[1][2]=nd[2]*sin(rteta/2);
qa[1][3]=nd[3]*sin(rteta/2);
dmtrs();
getch();
vektor();
getch();
matcrp();
return ;
}
//***** bir kuaterniyondan elde edilen
döndürme*****/
//***** matrisi ile verilen vektörün döndürülmesi****//
void qdonme(){
printf("\n.....bir quaternion
girin.....!\n\n");
n=1;
qqgir();
getch();
norm1=sqrt(qa[1][0]*qa[1][0]+qa[1][1]*qa[1][1]+qa[1][2]*qa[1][2]+qa[1][3]*qa[1][3]);
printf("\nqnorm .....:%.2f", norm1);
if(norm1==1){
//.....girilen q nun dönme açısını bulur.....:
printf("\n\n kuaternion birimdir normlamaya gerek yok
!\n\n");
}
}

```

```

    aci=qa[1][0];
    acibul();
    son=sin(rteta);
    //.....girilen q nun döndürme
doğrultusu.....:
    /* for(i=1;i<4;i++)
    dogrilt[i]=(double)(qa[1][i]/son);
    //HATA VAR .....!
    printf("\n\n.....d"nme
doğrultusu.....:\n\n");
    printf("%.2f , %.2f , %.2f
) ",dogrilt[1],dogrilt[2],dogrilt[3]);
    */
    getch();
}
else{
    printf("\n\n.....kuaternion birim değil!.....\n\n");
    // q nun normu al n p d"nme aþ s ve doğrultusu bulunur
    for(i=0;i<4;i++)
    nrmluq[i]=qa[1][i]/norm1;
    printf(" \n\n normluq= %.1f e0 + %.1f e1 + %.1f e2 + %.1f
e3 \n",nrmluq[0],nrmluq[1],nrmluq[2],nrmluq[3]);
    //normlanmış q nun dönme açısını bulur
    aci=nrmluq[0];
    acibul();
    //normlanmış q nun döndürme doğrultusu:
    for(i=1;i<4;i++)
        if (rteta==0)
            printf("\n 0 a bölme hatası olabilir");
            printf("\n.....sin(2*%.1f)=%.1f.....",
r1teta,sin(rteta));
            // dogrilt[i]=(double)((qa[1][i])/son); //HATA
VAR.....!
            // printf("\n\n normlu q nun dönme doğrultusu: (%.1f ,
%.1f , %.1f )\n ",dogrilt[1],dogrilt[2],dogrilt[3]);
            //normlu q yu yine qa ya at yal m
            for(i=0;i<4;i++)
                qa[1][i]=nrmluq[i];
            getch();
}
dmters();

```

```

vektor();
sncuyari();
matcrp();
return;
}
//***** SEÇİMLER *****//
void secenekler(){
    printf("Kuaternion cebrinde hangi işlemi yapmak
istiyorsunuz?\n\n ");
    printf("Bir kuaternionun cebirsel tanımı.....: 0 \n
");
    printf("n tane kuaternionun toplam .....: 1 \n
");
    printf("iki kuaternionun fark .....: 2 \n
");
    printf("Bir kuaternionun eşleniği.....: 3 \n
");
    printf("Bir kuaternionun skaler çarpım .....: 4 \n
");
    printf("Bir kuaternionun normu.....: 5 \n
");
    printf("Bir kuaternionun tersi.....: 6 \n
");
    printf("iki q nun kuaternionik çarpım .....: 7 \n
");
    printf("Bir vektörel q nun ussu.....: 8 \n
");
    printf("Açı verilerek döndürme.....: 9 \n
");
    printf("q verilerek döndürme.....: 10
\n ");
    printf("\n seçeneklerinden birini girin!");
return;
}
//***** ANA PROGRAM *****//
main(){
int y,status;
labell1:
clrscr();
//secimgraf();
secenekler();

```

```
printf("\n\n se#iminiz  :");
scanf("%i", &c);
label2:
switch (c){
    case 0:
        qqgir();
        break;
    case 1:
        qtopla();
        break;
    case 2:
        qfark();
        break;
    case 3:
        qeslenik();
        break;
    case 4:
        qscarp();
        break;
    case 5:
        qnorm();
        break;
    case 6:
        qters();
        break;
    case 7:
        qcarp();
        break;
    case 8:
        qus();
        break;
    case 9:
        qdonmeaci();
        break;
    case 10:
        qdonme();
        break;
}
printf("\a\n\n devam etmek istiyor musunuz
(evet=1)/(hay r=0) :");
scanf("%i", &y);
```

```
if (y==1){
    goto label1;
    goto label2;
}
else if(y==0)
{
    printf("\n\nProgramdan çıkmak için ENTER tuşuna
basınız");
    exit(status - '0');
}
return 0;
getch();
}
```

Ek 2: Shearing Kinematik ile İlgili Mapple Kodları

1. \vec{u} öteleme vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);  
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);  
> plot([u1,u2, t=-1..1]);
```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

2. \vec{u} öteleme vektörünün türev vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);  
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);  
> u1t:=diff(u1,t);  
> u2t:=diff(u2,t);  
> plot([u1t,u2t, t=-1..1]);
```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

$$u1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$u2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

3. \vec{x} vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);  
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);  
> a:=2;  
> b:=1;  
> x1:=u1+a;  
> x2:=u2+b;  
> plot([x1,x2, t=-1..1]);
```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

$$a := 2$$

$$b := 1$$

$$x1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t) + 2$$

$$x2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1$$

4. \vec{v}_r bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);
```

```
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);
```

```
> u1t:=diff(u1,t);
```

```
> u2t:=diff(u2,t);
```

```
> x1t:=u1t;
```

```
> x2t:=u2t;
```

```
> plot([x1t,x2t, t=-1..1]);
```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

$$u1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$u2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$x1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$x2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

5. \vec{v}_f bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);
```

```
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);
```

```
> a:=2;
```

```

> b:=1;
> x1:=u1+a;
> x2:=u2+b;
> u1t:=diff(u1,t);
> u2t:=diff(u2,t);
> x1t:=u1t;
> x2t:=u2t;
> vf:=(-u1t-u2/sqrt(1+t^2)+x2/sqrt(1+t^2),-t/(1+t^2)*u2-u2t-t*x2/(1+t^2));
> plot([vf, t=-1..1]);

```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

$$a := 2$$

$$b := 1$$

$$x1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t) + 2$$

$$x2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1$$

$$u1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$u2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$x1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$x2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$\begin{aligned}
vf := & 3 \cos(t) - 2 t \sin(t) + 2 \sin(t) + 6 t \cos(t) - t^2 \cos(t) - 2 t^2 \sin(t) \\
& + \frac{(-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1}{\sqrt{1 + t^2}}, -2 \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}} \\
& - (\sin(t) - t \cos(t)) \sqrt{1 + t^2} - \frac{t((-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1)}{1 + t^2}
\end{aligned}$$

6. \vec{v}_a bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```

> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);

```

```

> a:=2;
> b:=1;
> x1:=u1+a;
> x2:=u2+b;
> u1t:=diff(u1,t);
> u2t:=diff(u2,t);
> x1t:=u1t;
> x2t:=u2t;
> vr:=(x1t,x2t);
> vf:=(-u1t-u2/sqrt(1+t^2)+x2/sqrt(1+t^2),-t/(1+t^2)*u2-u2t-t*x2/(1+t^2));
> va:=vr+vf;
> plot([va, t=-1..1]);

```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2}$$

$$a := 2$$

$$b := 1$$

$$x1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t) + 2$$

$$x2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1$$

$$u1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$u2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$x1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$x2t := (\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$vr := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t),$$

$$(\sin(t) - t \cos(t)) \sqrt{1 + t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$vf := 3 \cos(t) - 2 t \sin(t) + 2 \sin(t) + 6 t \cos(t) - t^2 \cos(t) - 2 t^2 \sin(t)$$

$$+ \frac{(-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1}{\sqrt{1 + t^2}}, -2 \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1 + t^2}}$$

$$- (\sin(t) - t \cos(t)) \sqrt{1 + t^2} - \frac{t ((-t \sin(t) - 2 \cos(t)) \sqrt{1 + t^2} + 1)}{1 + t^2}$$

$$va := 2 \cos(t) + t \sin(t) + \frac{(-t \sin(t) - 2 \cos(t)) \sqrt{1+t^2} + 1}{\sqrt{1+t^2}},$$

$$- \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1+t^2}} - \frac{t((-t \sin(t) - 2 \cos(t)) \sqrt{1+t^2} + 1)}{1+t^2}$$

7. Pol eğrisinin Mapple kodları:

```
> u1:=-t*cos(t)-2*t*sin(t)+t^2*sin(t)-2*t^2*cos(t);
```

```
> u2:=(-t*sin(t)-2*cos(t))*sqrt(1+t^2);
```

```
> a:=2;
```

```
> b:=1;
```

```
> x1:=u1+a;
```

```
> x2:=u2+b;
```

```
> u1t:=diff(u1,t);
```

```
> u2t:=diff(u2,t);
```

```
> p:=(x1,sqrt(1+t^2))*u1t+u2;
```

```
> plot([p, t=-1..1]);
```

$$u1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t)$$

$$u2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1+t^2}$$

$$a := 2$$

$$b := 1$$

$$x1 := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t) + 2$$

$$x2 := (-t \sin(t) - 2 \cos(t)) \sqrt{1+t^2} + 1$$

$$u1t := -\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t)$$

$$u2t := (\sin(t) - t \cos(t)) \sqrt{1+t^2} + \frac{(-t \sin(t) - 2 \cos(t)) t}{\sqrt{1+t^2}}$$

$$p := -t \cos(t) - 2 t \sin(t) + t^2 \sin(t) - 2 t^2 \cos(t) + 2,$$

$$\sqrt{1+t^2} (-\cos(t) + 3 t \sin(t) - 2 \sin(t) - 6 t \cos(t) + t^2 \cos(t) + 2 t^2 \sin(t))$$

$$+ (-t \sin(t) - 2 \cos(t)) \sqrt{1+t^2}$$

Ek 3: Shear Dönüşümü Olmadan Normal Kinematığın Mapple

Kodları

1. \vec{u} öteleme vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t;  
> u2:=2*t;  
> plot([u1,u2, t=-1..1]);  
u1 := -t  
u2 := 2 t
```

2. \vec{u} öteleme vektörünün türev vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları

```
> u1:=-t;  
> u2:=2*t;  
> u1t:=diff(u1,t);  
> u2t:=diff(u2,t);  
> plot([u1,u2, t=-1..1]);  
u1 := -t  
u2 := 2 t  
u1t := -1  
u2t := 2
```

3. \vec{x} vektörünün hareket boyunca çizdiği eğrisinin Mapple kodları:

```
> u1:=-t;  
> u2:=2*t;  
> a:=2;  
> b:=1;  
> x1:=u1+a;  
> x2:=u2+b;  
> plot([x1,x2, t=-1..1]);  
u1 := -t  
u2 := 2 t  
a := 2  
b := 1  
x1 := -t + 2  
x2 := 2 t + 1
```

4. \vec{v}_r bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple

kodları:

```
> u1:=-t;  
> u2:=2*t;  
> u1t:=diff(u1,t);  
> u2t:=diff(u2,t);  
> x1t:=u1t;  
> x2t:=u2t;  
> vr:=(x1t,x2t);  
> plot([vr, t=-1..1]);
```

u1 := -t

u2 := 2 t

u1t := -1

u2t := 2

x1t := -1

x2t := 2

vr := -1, 2

5. \vec{v}_f bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple

kodları:

```
> u1:=-t;  
> u2:=2*t;  
> a:=2;  
> b:=1;  
> x1:=u1+a;  
> x2:=u2+b;  
> vf:=(1+2*t+x2,-2+t+x1);  
> plot([vf, t=-1..1]);
```

u1 := -t u2 := 2 t

a := 2

b := 1

x1 := -t + 2

x2 := 2 t + 1

vf := 2 + 4 t, 0

6. \vec{v}_a bağıl hız vektörünün hareket boyunca çizdiği eğrisinin Mapple

kodları:

```
> x1:=u1+a;  
> x2:=u2+b;  
> vr:=(diff(x1,t),diff(x2,t));  
> vf:=(1+2*t+x2,-2+t+x1);  
> va:=vf+vr;  
> plot([va, t=-1..1]);
```

$$u1 := -t$$

$$u2 := 2 t$$

$$a := 2$$

$$b := 1$$

$$x1 := -t + 2$$

$$x2 := 2 t + 1$$

$$vr := -1, 2$$

$$vf := 2 + 4 t, 0$$

$$va := 1 + 4 t, 2$$

7. Pol eğrisinin Mapple kodları:

```
> P:=(2-t,-1-2*t);  
> plot([P, t=-1..1]);  
 $P := -t + 2, -1 - 2 t$ 
```

Ek4: Galile Dönüşümlerinin Bilgisayar Uygulamalarının Program Kodları

```
uses crt;

var

x,y,a,b,v,i,j:integer;

son:array[1..2,1..20] of integer;

cev,x1,y1:integer;

begin

clrscr;

randomize;

writeln('sırasıyla v, a ve b say lar n giriniz');

readln(v,a,b);

x:=0;

y:=2*x+4;

repeat

x1:=-2;

if x1=x then cev:=1;

until cev<>1;

y1:=2*x1+4;

son[1,1]:=x+a;

son[1,2]:=x*v+y+b;

son[2,1]:=x1+a;

son[2,2]:=x1*v+y1+b;
```

```
writeln('Orijinal noktalar');  
  
writeln('(x,',y,')');  
  
writeln('(x1,',y1,')');  
  
writeln('yeni noktalar');  
  
for j:=1 to 2 do  
  
    writeln('(son[j,1],',son[j,2],')');  
  
readln;  
  
end.
```