

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

**ARAZİYE UYUM SAĞLAYABİLEN ÇOK KADEMELİ
SALINCAK MEKANİZMALI
BİR ARACIN TASARIMI**

Alper YONTAR

Tez Danışmanı : Öğr. Gör. Dr. Özgür KİLİT

Makine Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 625.02.00

Sunuş Tarihi : 04.02.2010

Bornova-İZMİR

2010

Alper YONTAR tarafından **Yüksek Lisans** Tezi olarak sunulan “**Araziye Uyum Sağlayabilen Çok Kademeli Salıncak Mekanizmalı Bir Aracın Tasarımı**” 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 04.02.2010 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı	: Öğr. Gör. Dr. Özgür KİLİT
Raportör Üye	: Doç. Dr. Hasan YILDIZ
Üye	: Yrd. Doç. Dr. Mehmet İ. Can DEDE

ÖZET**ARAZİ ARAZİYE UYUM SAĞLAYABİLEN ÇOK KADEMELİ
SALINCAK MEKANİZMALI BİR ARACIN TASARIMI**

YONTAR, Alper

Yüksek Lisans Tezi, Makine Mühendisliği Bölümü

Tez Yöneticisi: Öğr. Gör. Dr. Özgür KİLİT

Şubat 2010, 58 sayfa

Bu tez ile araziye uyumlu ve çok kademeli salıncak mekanizmalı bir aracın tasarımı, kararlılığı (stabilizesi) ve prototip üretimi incelenmiştir.

Dış-uzay gezen araştırmalarında kullanılan mobil robotların engebeli araziye uyumu için birçok tasarım bulunmaktadır. Bu tezde çok kademeli salıncak mekanizması incelenmiştir. Aracın yol eğrilerinin hesaplanması için arazi modellemeye gerek duymayan bir yöntem geliştirilmiştir. Kararlılık parametreleri belirlenmiş ve bilgisayarda bu değerler karşılaştırılmıştır.

Hesapları yapılan ve kararlılığı incelenen aracın prototip üretimi gerçekleştirilerek yapılan çalışmaların doğrulaması gerçekleştirilmiştir.

Anahtar sözcükler: Mobil robot, çok kademeli salıncak mekanizması, yol modelleme, kararlılık kriteri.

ABSTRACT

DESIGN OF A ROUGH TERRAIN ADAPTIVE VEHICLE WITH MULTI-STAGE BOGIE MECHANISM

YONTAR, Alper

MSc in Mechanical Eng.

Supervisor: Dr. Özgür KİLİT

February 2010, 58 pages

In this thesis; design, stability and production of a terrain adaptive vehicle with multi-stage bogie mechanism has been studied.

There are many rover designs which are used for outer-space ground exploration missions. In this thesis, multi-stage bogie mechanism design has been studied. A novel method to model the rover path which doesn't require any surface modeling tool of rough terrain is represented. Two tilting angles – measured about both longitudinal and lateral axes of the rover body – are taken into consideration as stability criteria. Through the computer simulation, the kinematics model of the mobile robot moving on a randomly generated rough verified.

Prototype rover production has been done after the calculations and investigation of stability criteria.

Keywords: Mobile robots, Multi-stage bogie mechanism, Path modelling, Stability criteria.

TEŐEKKÜR

Bu alıŐma sűresince her konuda benden desteęini eksik etmeyen DanıŐmanım Őęr. Gűr. Dr. Őzgűr KİLİT'e, prototip űretimde maddi destekleri iin Mustafa KAYABAŐI'na, yine prototip űretiminde tecrűbesi ile bana yardımcı olan Mehmet ATASOY'a ve desteęi eksik olmayan Buket YILMAZ'a teŐekkűrű bir bor bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
ABSTRACTvii
TEŞEKKÜR	ix
ŞEKİLLER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xv
1.GİRİŞ	1
2.İLERLEME VE ARAZIYE UYUM	4
2.1 Mobil Robotlar	4
2.2 İlerleme	4
2.3 Tekerlek	4
2.4 Süspansiyon	5
2.5 Salıncak (Bogie) Mekanizması	5
2.6 Rocker-Bogie Mekanizması	7
3.ARAÇ MODELİ	9
4. YOL MODELLEME	10
5.ARACIN KİNEMATİĞİ	14

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
6. KARARLILIK (STABİLİTE)	19
7. PROTOTİP	23
7.1 Tasarım.....	23
7.2 Protoip Üretimi	23
8. SONUÇ	25
9. ÖNERİLER.....	26
KAYNAKLAR	27
EKLER.....	29
Ek 1 Matlab Kodları.....	29
Ek 2 İmalat Resimleri	47
ÖZGEÇMİŞ	58

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1 USSR ilk ay aracı Lunokhod 1	2
1.2 NASA'nın ilk Mars aracı Sojourner	2
1.3 NASA'nın Mars Aracı Spirit.....	3
2.1 Salıncak mekanizmasının araziye uyumu	6
2.2 İki ve Üç kademeli salıncak mekanizmalı mobil robot	6
2.3 Rocker-bogie mekanizmasının patenti	7
2.4 NASA Spirit Rocker-bogie mekanizması	7
3.1 Çok kademeli salıncak mekanizmalı Mars aracı	9
3.2 Salıncak mekanizma grubun engebeli araziye uyumu	9
4.1 Yol modelleme algoritmasının psuedocode'u (taslak kodu)	12
4.2 Verilen nokta bulutu ve hesaplanan yollar	13
5.1 Tekerlek merkez eksenlerinin eğrisinin hesaplanması	14
5.2 Bir salıncak mekanizmasının geometrik özellikleri	15
5.3 Tekerlek eksenlerinin pozisyonlarının belirlenmesi.....	16
5.4 Tekerlek eksenlerinin belirlenmesinde kullanılan algoritmanın taslak kodu.	17
5.5 Mafsal noktaları yörüngelerinin geometrik hesabı.....	18

ŞEKİLLER DİZİNİ (devamı)

<u>Şekil</u>	<u>Sayfa</u>
6.1. Normal vektörü ve açılar	19
6.2 Farklı kademelerden oluşan araçların uyum karşılaştırması.....	21
6.3 Kararlılık (Stabilite) parametreleri.....	22
7.1 Bağlantı elemanları olmadan prototip aracın 3B bilgisayar destekli çizimi	23
7.2. LIBRA'nın prototip fotoğrafı	24

SİMGELER VE KISALTMALAR DİZİNİ

<u>Simgeler</u>	<u>Açıklama</u>
z_i	i-inci nokta için verilen yükseklik datası.
p	ağırlık sayısı.
Z_j	j noktası için hesaplanan yükseklik değeri.
d_{ij}	i noktasından hesaplanan j noktasına olan uzaklık.
x_i, y_i	Verilen arazi noktaları.
x_c, y_c	Aracın o an bulunduğu konum koordinatları.
x_t, y_t	Aracın hedef konum koordinatları.
m	nokta sayısı.
num	Pay.
$denom$	Payda.
X_j, Y_j	Bulunması istenen noktalar.
r	Tekerlek yarıçapı.
d_1, d_2, \dots, d_{m-1}	Yol eğrisini oluşturan doğru parçaları.
$d'_1, d'_2, \dots, d'_{m-1}$	Yol eğrisinin paralel doğru parçaları.
$w_{c1}, w_{c2}, \dots, w_{cm}$	Tekerlek merkezleri eğrilerinin birleşme noktaları.
L	iki son mafsallı birleştiren doğrunun boyu.

SİMGELER VE KISALTMALAR DİZİNİ (devamı)

<u>Simgeler</u>	<u>Açıklama</u>
H	Salıncak yüksekliği (L'ye dik doğrunun boyu).
A, B ve C	Üç bağlantı noktasının konum vektörleri.
i, j ve k	Sırayla x, y ve z yönündeki birim vektörler.
N	Üst düzlemin normal vektörü.
γ	N ve k vektörleri arasındaki açı (kararlılık parametresi).
γ_{\max}	γ açısının maksimum değeri.
α ve β	Sırasıyla enine ve boyuna eksenlerin açısal değişimleri.
<u>Kısaltmalar</u>	
NASA	National Aeronautics and Space Administration.
MER	Mars Exploration Rover.
USSR	The Union of Soviet Socialist Republics (Rusya).
USTPO	The United States Patent and Trademark Office.
IDP	Invers Distance to a Power.
NURBS	Non-uniform rational B-spline.
DTM	Digital Terrain Modelling.

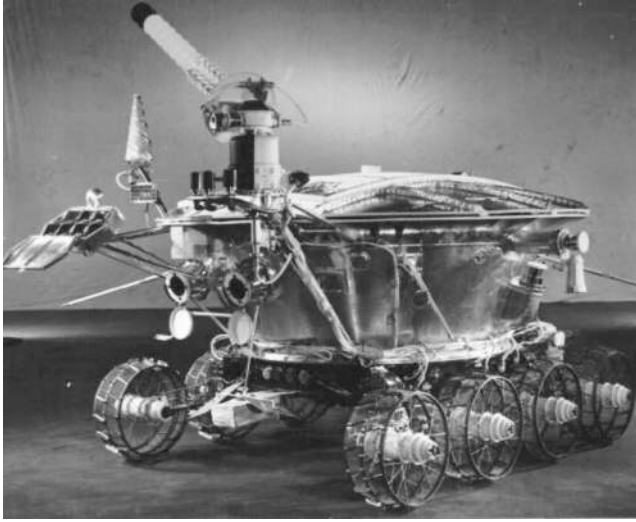
1. GİRİŞ

İnsanlık için tekerlek önemli bir keşiftir ve yüzyıllardır tekerlekli araçlar yolcu ve eşya taşımada kullanılmaktadırlar. Ancak bu araçlar, özel olarak tasarlandıkları yollar dışında fonksiyonlarını kaybetmektedirler. Bu nedenden dolayı birçok yol-dışı (off-road) araç tasarlanmıştır. Bu araçlar büyük tekerlekli, paletli ya da çok tekerlekli araçlar; tarımsal, buzlu, engebeli, kayalık ya da kum arazilerde kullanılan araçlar gibi çeşitli yol-dışı araçlar olarak örneklendirilebilir. Genel olarak bu araçların kişisel kullanım, tarım makineleri, iş makineleri, askeri araçlar ve özel kurtarma operasyonları gibi kullanım grupları bulunmaktadır. Bu örneklerden anlaşılacağı gibi yol-dışı araçların birçok farklı uygulamaları ve bunlara özgü tasarımları bulunmaktadır. Çünkü her arazinin geometrik yapısı ve yüzey özellikleri birçok değişiklikler göstermektedir. Günümüzde birçok araştırma kurumunda çok farklı yüzeyler üzerinde hareket etmeye uygun yol-dışı araç tasarımları üzerinde çalışılmaktadır.

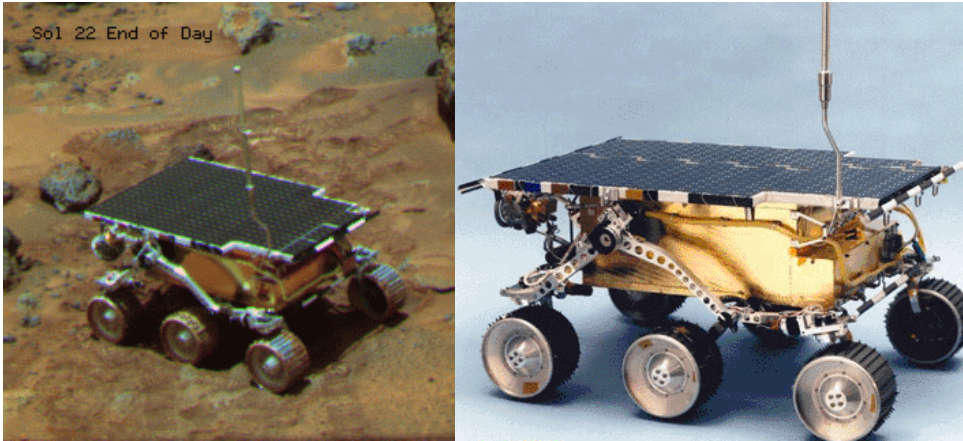
Yukarıda bahsedilen bu araçları “Araziye Uyum Sağlayabilen Araçlar” olarak tanımlayabiliriz. Araziye uyum sağlayabilen araçların ve mekanizmaların en çok kullanıldığı alan ise robotiktir. Özellikle Mars gibi Dünya dışı gezegenlerin keşif görevlerinde bu sistemler kullanılmaktadır.

İnsanlığın ilk günlerinden beri ilgisini çeken uzay, son yıllarda artık meraktan daha çok, önemli ve geniş bir bilim alanı olmuştur. Özellikle Dünya kaynaklarının sürekli büyüyen nüfusun gereksinmelerine yetmemesi, ortaya çıkabilecek büyük doğal afetlerin ya da nükleer kazalar sonucu insanlığın varlığına devam edebilmesi için bu araştırmaların önemi artmaktadır. Ancak uzay araştırmalarında insanlık keşif görevlerinin maliyetleri çok yüksektir. Gönderilecek ekibin dış-uzayda ihtiyaçlarının ve güvenliklerinin sağlanması, aynı zamanda ekibin psikolojik dengesinin korunması gibi daha birçok neden sayılabilir. İşte bu nedenlerden dolayı, insansız araştırmaların önemi ve uygulamaları her gün önem kazanmaktadır. Gezegen araştırmalarında tercih edilen mobil robot keşif görevleridir (Farritor, Hacot and Dubowsky, 1998). Amerikan Ulusal Uzay Ajansı NASA 1960'dan bu yana devam eden uzay araştırmaları ve çalışmaları sonucu

1997'de Mars'a Sojourner (Şekil 1.2) adlı mobil robotunu göndermiştir. 2004 yılında ise yine NASA'nın Mars Keşif Araçları, Spirit (Şekil 1.3) (MER1) ve Opportunity (MER2) Mars'ın farklı bölgelerine gönderilmişlerdir (NASA, 2004). Amerika'nın insansız araçlarına karşın USSR (Rusya) tarafından ilk kez 1970 yılında Ay'a gönderilen araç Lunokhod 1 (Şekil 1.1)'dir (Wikipedia, 2009).



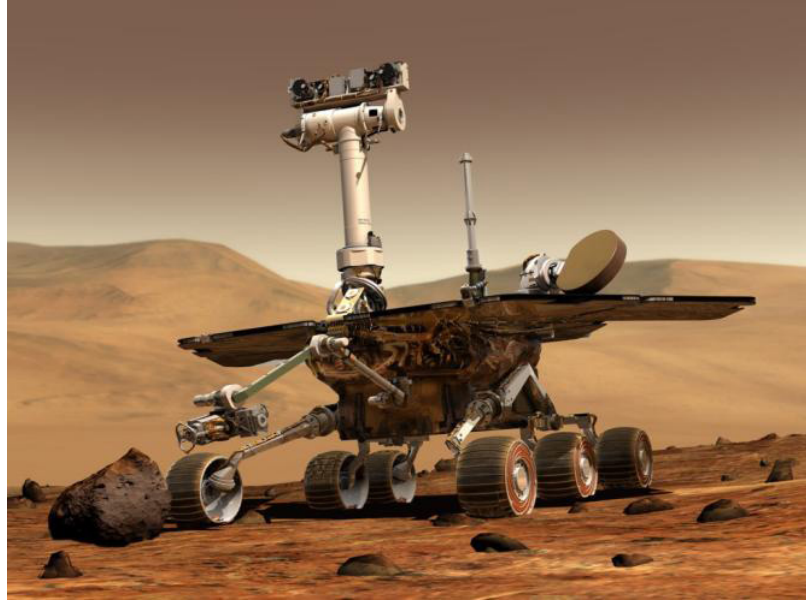
Şekil 1.1 USSR ilk ay aracı Lunokhod 1 (Wikipedia, 2009).



Şekil 1.2 NASA'nın ilk Mars aracı Sojourner (JPL-NASA, 2001).

Mobil robotların ve araçların Dünya'dan kontrolü (Mars için) yaklaşık 45 dakikalık bir gecikme ile gerçekleştirilebilmektedir (Jonathan, 1992). Bu nedenle otonom ya da yarı-otonom robotlar tercih edilmektedir. Gezegenin engebeli yüzeyinde, büyük zaman ve maliyetlerle hazırlanmış araçların devrilerek ya da sıkışarak çalışamaz duruma düşmesi tüm projeyi tehlikeye sokabilir. Araziye

uyum sağlayabilen keşif araçlarının kullanılmasının nedeni ve önemi burada yatmaktadır. Böyle bir durumun ne kadar gerçekçi bir risk faktörü olduğu, bu çalışma hazırlanırken Mars'ta hala görevini sürdüren Spirit aracının 6 yıllık çalışmasının ardından yumuşak kum yüzeye saplanarak kalması ile anlaşılmıştır. Spirit, Dünya'dan yapılan tüm müdahalelere rağmen şu ana kadar kurtarılamamıştır (NASA, 2009). Spirit örneğinde görüldüğü gibi, uzak bir gezene ya da başka bir uzay cismine gönderilen araçların, üzerinde görevlerini sürdürecekleri araziye uyumları ve arazi üzerindeki kararlılıkları büyük önem taşımaktadır. Aracın görevini tamamlayabilmesi ve görevini doğru olarak yapabilmesi için bu konu yaşamsal bir öneme sahiptir.



Şekil 1.3 NASA'nın Mars Aracı Spirit (MER-NASA (1), 2009).

Bu tez çalışmasında araziye uyum sağlayabilen tekerlekli ve çok kademeli bir aracın tasarımı gerçekleştirilecek ve stabilite değerleri belirlenerek prototip üretimi sonuçları incelenecektir.

2. İLERLEME VE ARAZİYE UYUM

2.1 Mobil Robotlar

Mobil robotlar, doğal ya da yapay engelleri bulunan bir araziye geçebilme kabiliyeti olan otonom sistemler olarak tanımlanabilir (Dudek and Jenkin, 2000). Mobil robotların ve araçların uygulamalarında en önemli kısmı dış-uzay yer araştırmalarıdır. Ancak uzay araştırmaları ve keşifleri dışında mobil robotlar, Dünya’da tehlikeli ve zor şartlar altında da (nükleer atık bölgeleri, mayınlı araziler, doğal afet bölgeleri, mağara araştırmaları, yüksek sıcaklığın olduğu alanlarda v.b) kullanılmaktadır. Bu değişik görevler ve araziler için tasarlanmış robotların ortak noktası, hepsinin araziye uyumlu araçlar olmalarıdır.

2.2 İlerleme

İlerleme, araç gövdesinin yaklaşık olarak araziye paralel olacak şekilde hareket etmesidir. Mobil robotların ve araçların, hareket ve ilerleme yetenekleri genel olarak aşağıda sıralanmış sistemlerle sağlanmaktadır:

Tekerlek (Wheel), Ayak (Leg), Palet (Track), Tekerlek-ayak (Wheeleg), Sürünme (Limbless).

Bu araçların hareketleri ve ilerlemeleri arazi yüzeyine temas halindeki sistemin bir hareketlendirici (genellikle motor) tarafından tahrik edilmesi ile sağlanır. Bazı araçlarda tüm hareket sistemleri ayrı ayrı tahrik edilirken bazıları da otomobillerde olduğu gibi tek bir merkezden de tahrik edilebilmektedir. Bu durum yine tasarlanan aracın ne amaçla kullanıldığına ve hareket sisteminin araziye uyum mekanizmasına göre değişmektedir.

2.3 Tekerlek

Tekerlek, araçlarda ve mobil robotlarda ilerlemenin sağlanması için kullanılan en önemli parçalardan biridir. Tekerleğin ilerlemeyi sağlaması için, bağlı olduğu mil ekseninde dönerek yer ile temas eden yüzeyinin sürtünmesini yenmesi gereklidir. Bir tekerleğin aşabileceği en büyük engel yüksekliği

tekerleğin yarıçapı kadardır. Tekerleğin bağlı olduğu mekanizma yardımıyla yarıçapından büyük yükseklikte basamak şeklinde engelleri aşması durumu tırmanma olarak tanımlanır. Bu tez çalışmasında basamak şeklinde engellerin aşılması yani tırmanma süreci incelenmeyecektir. Aracın doğada görülebilecek ve özel olarak düzenlenmemiş bir engebeli araziye uyumu incelenecektir.

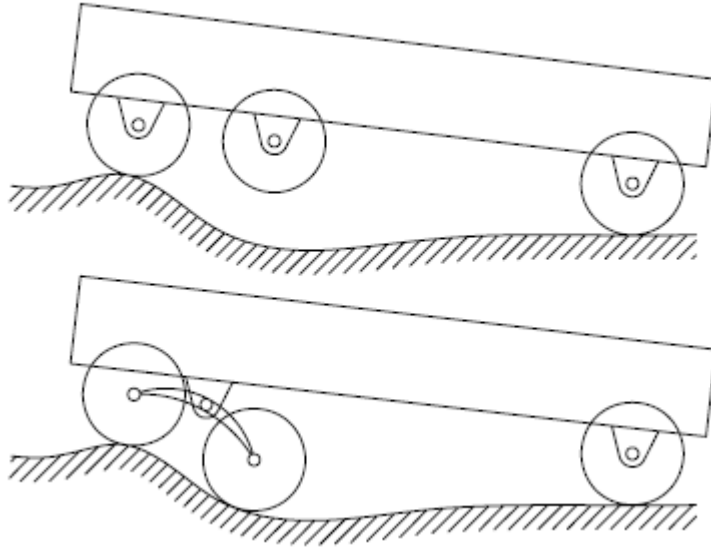
2.4 Süspansiyon

Araziye uyumlu araçların tüm tekerleklerinin araziye temas etmesi gerekmektedir. Bu nedenle değişik süspansiyon sistemleri kullanılır. Süspansiyon mekanizmaları sayesinde tasarım gerçekleştirilirken hesaplanan değerler ile aracın arazi üzerindeki göstereceği davranışların uyumu sağlanmış olur ve aracın görevini gerçekleştirirken beklenmeyen bir durumla karşılaşılması için sistem kontrol altında tutulur.

Birçok araç tasarımı için farklı birçok süspansiyon mekanizması bulunmaktadır. Tekerlekli araçlar için en çok bilinen ve uygulamada olan, otomobillerde de kullanılan mekanik yaylı süspansiyon mekanizmalarıdır. Ancak yaylar genellikle insan konforu için önemli olan titreşim kontrolü için kullanılır. Yaylar, hareket yeteneği (mobility) için önemlidirler, ancak bu durum 8m/s'den yüksek hızlara sahip araçlarda sürücünün konforu ve kontrolü için geçerlidir. Bu hızların altında yaylar hareket kabiliyetini engeller, çünkü her tekerleğin çarpışma sırasında yere uyguladığı kuvveti değiştirerek iletirler (Sandin, 2003). Yaylar yerine mobil robotlarda ve araçlarda kullanılanlar rocker, rocker-bogie ve bogie (salıncak) mekanizmaları kullanılmaktadır. Bunlar dışında benzer ve farklı birçok mekanizma bulunmakta ya da tasarlanmaktadır.

2.5 Salıncak (Bogie) Mekanizması

Salıncak mekanizmasının asıl amacı, araçla arazi yüzeyi arasındaki teması mümkün olduğunca sağlamaya çalışmaktır. Bu nedenle, tekerlekler araç gövdesine doğrudan bağlı değildir, Şekil 2.1'deki gibi serbest dönebilen mafsallı bir linkle gövdeye bağlanırlar.

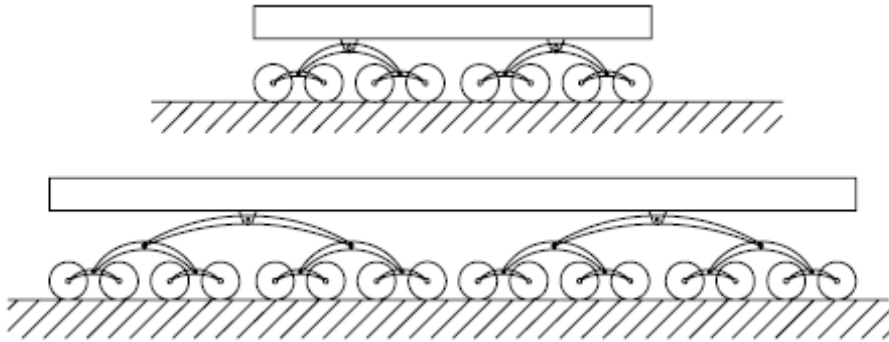


Şekil 2.1 Salıncak mekanizmasının araziye uyumu (Kilit, 2005).

Salıncak mekanizmaları ile Şekil 2.2’de görüldüğü gibi iki ve üç kademeli olarak gruplanarak çok kademeli mekanizmalar oluşturulabilir. Kilit (2005) tarafından önerilen bu çok kademeli sistem ile arazinin engebeli yapısının aracın ağırlık merkezinin yer çekimine paralel ekseninde hareket aralığını azaltarak zeminin araç üzerindeki etkisi düşürülebilir. Bu çok kademeli salıncak mekanizmasının avantajları ise;

1) Zeminin engebeli yapısının etkisini daha az araca aktarılması ve

2) Rocker-bogie mekanizmasına göre, aracın hem gidiş hem de tersi yönüne doğru olmak üzere her iki yöne de aynı özellikleri gösteren simetrik yapısıdır.

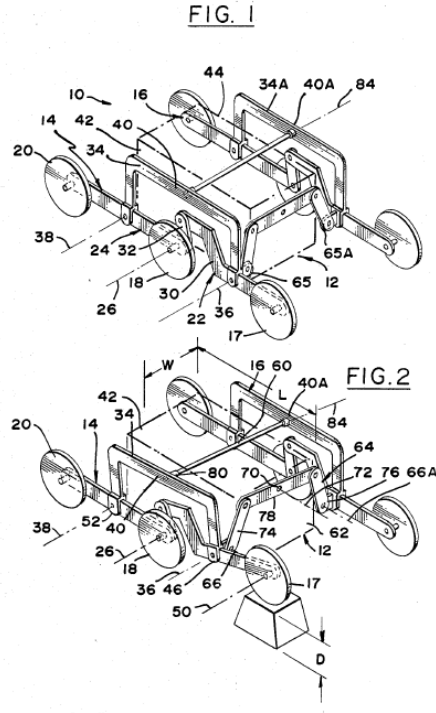


Şekil 2.2 İki ve Üç kademeli salıncak mekanizmalı mobil robot (Kilit, 2005).

2.6 Rocker-Bogie Mekanizması

Rocker-bogie mekanizması ilk olarak NASA tarafından Sojourner için geliştirilmiştir. Bu mekanizma için alınan patent (USTPO 4840394) Şekil 2.3’de görülebilir. NASA’nın Mars Keşif Aracı Spirit’te kullanılan rocker-bogie mekanizması ise Şekil 2.4’de yer almaktadır.

U.S. Patent Jun. 20, 1989 Sheet 1 of 2 4,840,394



Şekil 2.3 Rocker-bogie mekanizmasının patenti (USTPO, 1989).

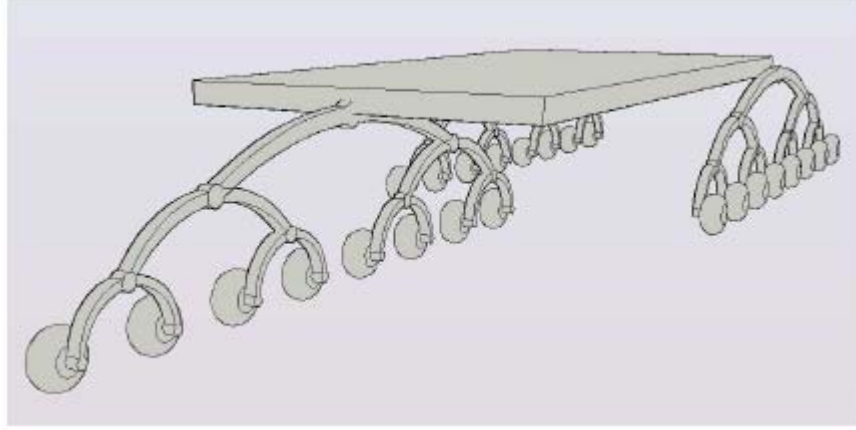


Şekil 2.4 NASA Spirit Rocker-bogie mekanizması (MER-NASA (2), 2009).

Rocker-bogie mekanizmasının avantajı her tekerlekteki yük neredeyse aynıdır. Farklı pozisyonlarında, diğer 4 tekerlekli yumuşak süspansiyonlara karşın tekerleklerin normal kuvvetleri eşit olarak dağılır (Sandin, 2003).

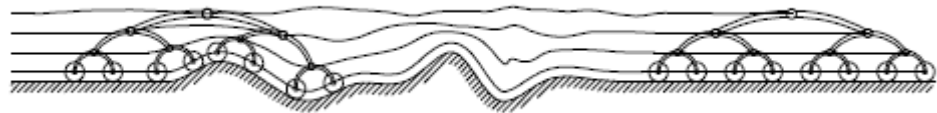
3. ARAÇ MODELİ

Bu tez çalışmasında Kilit (2005) tarafından önerilen çok kademeli salıncak mekanizmanın 3 boyutta araziye uyumu incelenmiştir.



Şekil 3.1 Çok kademeli salıncak mekanizmalı Mars aracı (Kilit, Yontar, 2009).

Önerilen çok kademeli salıncak süspansiyon sistemiyle aracın 3 boyutlu CAD çizimini Şekil 3.1’de görebilirsiniz. Mümkün olabildiğince çok araziye uyum için kademe sayısı üç olarak alınmıştır. Üç kademeli salıncak grubun araziye uyumunu Şekil 3.2’de görebilirsiniz. Şekil 3.2’de, üçüncü, ikinci ve birinci mafsal eğrileri sıra ile tekerlek eksenlerinden yukarıya doğru bulunmaktadır. Eğrilerin yerden yukarıya doğru giderek doğruya daha çok yaklaştığına dikkat edilmelidir. Salıncak süspansiyon mekanizması doğası gereği salınımlar üzerinde daha çok bir yumuşatma etkisi göstermektedir. Çünkü salıncak mekanizmasının orta mafsalının aldığı yol her iki uçtaki mafsalın aldığı yolun ortalaması olmaktadır. Daha fazla kademe, bu etkiyi arttırarak arazi düzensizliğinin araç gövdesine daha az iletilmesine neden olmaktadır.



Şekil 3.2 Salıncak mekanizma grubun engebeli araziye uyumu (Kilit, Yontar, 2009).

4. YOL MODELLEME

Kilit (2005), 2B’da yol modellemesi ve bu yol üzerinde aracın kararlılığının incelenmesi için bir model geliştirmiştir. Bu modelde yol, spline fonksiyonları ile tanımlanan parametrelere göre, yine bilgisayar programı yardımı ile salıncak mekanizmasına uygun bir yapıda oluşturulmaktadır. Modelleme sırasında, tekerlek yarıçapına göre oskülatör çember (osculation circle) kontrolü yapılarak, yolun tüm tekerleklerle öngörüldüğü gibi tek noktadan sürekli temas etmesi sağlanmıştır (Kilit 2005).

Bu bölümde, test arazisinden toplanan dağınık topografik data noktalarının aracın yörünge hesabında kullanılabilmesi amacıyla bir algoritma geliştirilmiştir. Aracın daha önceden belirlenmiş ilk konumdan hedef konumuna kadar, geçeceği yolun yörüngesinin hesaplanmasında Invers Distance to a Power (IDP) yönteminin değiştirilmiş bir versiyonu kullanılmıştır. Bu yöntem, bir araziye ait saçılmış nokta bulutu biçiminde verilmiş topografik data noktalarının ağırlıklı ortalamasını alarak istenilen herhangi bir noktanın yüksekliğini hesaplar. Bu yaklaşım arazi yüzeyini modellemek için NURBS (non-uniform rational bi-spline) gibi karmaşık yüzey modelleme araçlarının kullanım gerekliliğini ortadan kaldırır.

IDP yöntemi, yer yüzeyi topografisinin Dijital Arazi Modeli (DTM: Digital Terrain Modelling) gösteriminde çoğunlukla kullanılır. Bazı data toplama yöntemlerinin kullanımı ile, araziden dağınık yükseklik data noktaları toplanır. IDP yöntemi ağırlıklı ortalama (interpolatörü) düzelticisidir ve toplanan datayı düzenli bir kafes yapı ya da ızgaralama (gridding) gösterimine dönüştürür. Bugün, nerdeyse her modern haritalama yazılımı verilen arazinin kafes yapısını ve haritasını oluşturmak için bu yöntemi kullanır. Bu tez çalışmasında getirilen önemli bir özgün bir katkı, günümüzde yükselti nokta bulutu olarak verilen bir arazinin kafes yapısının çıkartılmasında ve haritalanmasında standart olarak kullanılan IDP yönteminin düzenlenerek bir arazi üzerindeki robot yörüngesinin bulunmasında kullanılmıştır. Burada üzerinde önemle durulması gereken bir başka nokta da geliştirilen bu algoritma ile tüm arazinin değil yalnızca robotun izlediği yolun üzerinde noktaların modellenmesidir. Bu da oldukça hızlı bir

hesaplama yapılabilmesi sonucunu getirmektedir. IDP yönteminde kullanılan temel denklem aşağıdadır.

$$Z_j = \frac{\sum_{i=1}^n \frac{z_i}{d_{ij}^p}}{\sum_{i=1}^n \frac{1}{d_{ij}^p}} \quad (4.1)$$

$(i = 1, 2, \dots, n), (j = 1, 2, \dots, m)$

z_i : i-inci nokta için verilen yükseklik datası ($i = 1, 2, \dots, n$)

p : ağırlık sayısı

d_{ij} : i noktasından hesaplanan j noktasına olan uzaklık

Z_j : j noktası için hesaplanan yükseklik değeri

Ağırlık sayısı büyüdükçe daha uzak x_i, y_i verilen arazi noktalarının bulunması istenen X_j, Y_j noktaları üzerindeki etkisi önemli bir şekilde azalmaktadır. Ağırlık sayısının etkisi arazi formu üzerinde etkilidir ve uygun sonuç alınıncaya kadar değişik sayılar denenmelidir. Bu çalışmada p değeri 2 alınmıştır. Verilen ve hesaplanan noktalar arasındaki uzaklık d_{ij} dikkatlice göz önüne alınmalıdır. Hesaplanan X_j, Y_j noktalarından biri x_i, y_i noktalarından biri ile çakışırsa d_{ij} sıfır olacaktır ve Z_j yükseklik değeri ∞/∞ (veya tanımsız) olur. Bu durum gerçekleştiğinde program durur ve “sıfıra bölme” hatası verir. Bu mesajı yok etmek için, d_{ij} değeri döngü içinde kontrol edilmelidir. Sıfıra eşit olduğunda z_i değeri Z_j değerine eşitlenmelidir. Bu yaklaşımın fiziksel anlamı, aranan değer zaten ölçülmüştür ve interpolasyona gerek yoktur. Şekil 4.1’de yol modelleme algoritmasının pseudocode’u (taslak kodu) yer almaktadır. Burada, x_i, y_i, z_i verilen n adet data noktalarının koordinatları, x_c, y_c ve x_t, y_t noktaları aracın o an ve hedef konum koordinatları, m o an bulunduğu ve ulaşacağı hedef

konumları arasındaki eşit aralıklı noktaların sayısı, p ağırlık sayısı, num ve $denom$ ise Denklem 4.1'in pay ve payda değişkenleridir.

```

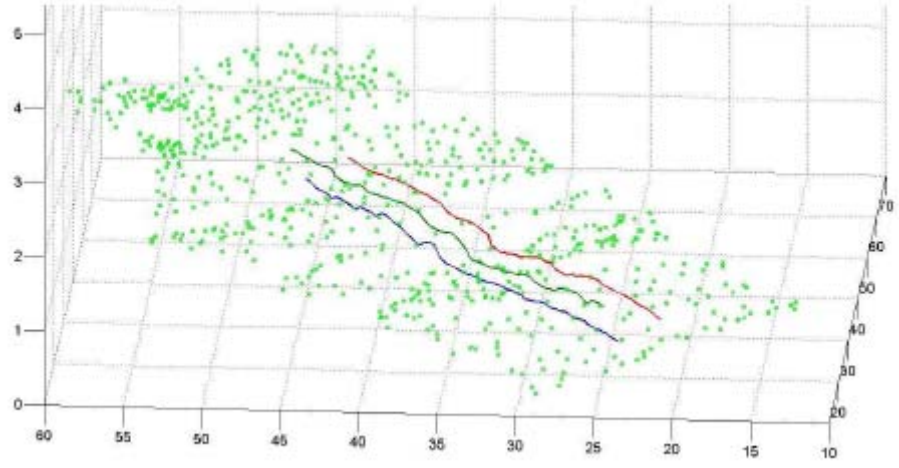
1. read  $x_i, y_i, z_i (i = 1, \dots, n)$ ,  $x_c, y_c, x_t, y_t, m, p$ 
2.  $num = 0, denom = 0, j = 1$ 
3. if  $j > m$  then goto 10
4.  $i = 1$ 
5. if  $i > n$  then  $Z_j = num/denom, j = j + 1, goto 3$ 
6.  $d = \sqrt{(X_j - x_i)^2 + (Y_j - y_i)^2}$ 
7. if  $d = 0$  then  $Z_j = z_i, j = j + 1, goto 3$ 
8.  $num = num + z_i/d^p, denom = denom + 1/d^p$ 
9.  $i = i + 1, goto 5$ 
10. write  $X_j, Y_j, Z_j (j = 1, \dots, m), END$ 

```

Şekil 4.1 Yol modelleme algoritmasının psuedocode'u (taslak kodu) (Kilit, Yontar, 2009).

Şekil 4.1'de verilen yol modelleme algoritması, rasgele oluşturulan bir arazi üzerinde test edilmiştir ve oluşturulan yollar Şekil 4.2'de gösterilmiştir. x_c, y_c ve x_t, y_t noktaları arasındaki doğru $m-1$ parçaya bölünmüştür ve p ağırlık sayısı 2 alınmıştır. Şekil 4.2'de görülen üç eğri, sırasıyla sağ, orta ve sol salıncak gruplarını göstermektedir. Bunun için algoritma üç tekerlek zinciri için üç kere çalıştırılmış ve sonuçlar birleştirilmiştir. Bu Şekil 4.2'de tekrar dikkat edilmelidir ki aracın üç tekerlek grubunun aldığı yolların hesaplanmasında da tüm arazinin ızgaralanması yerine yalnızca yolun bulunduğu yerlerdeki noktalar modellenmiş ve algoritma hızlandırılmıştır.

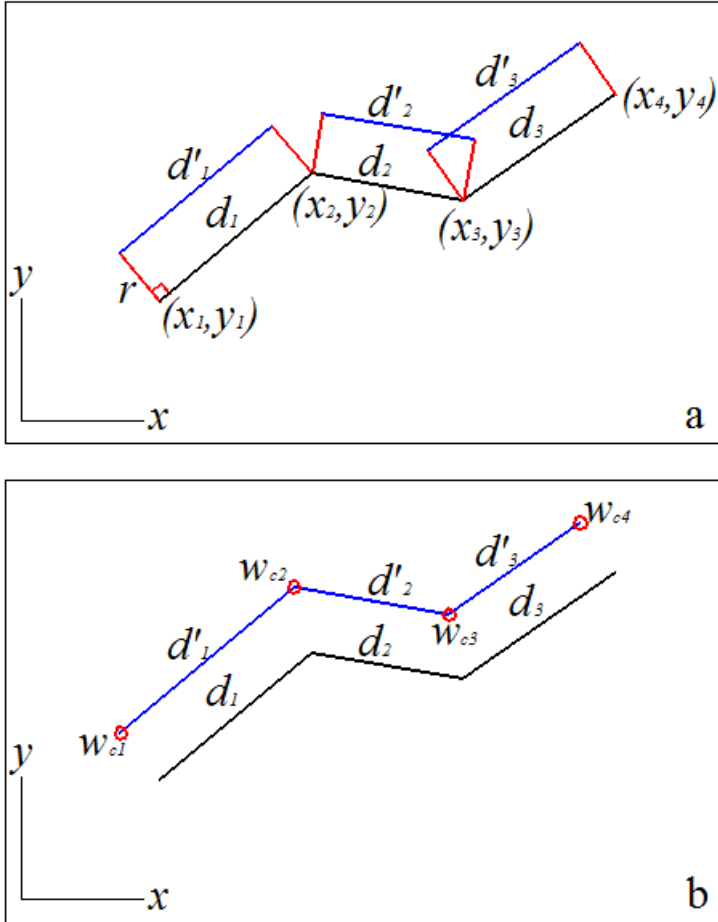
Hesaplamalar için MATLAB programının kodları EK 1'de bulunabilir.



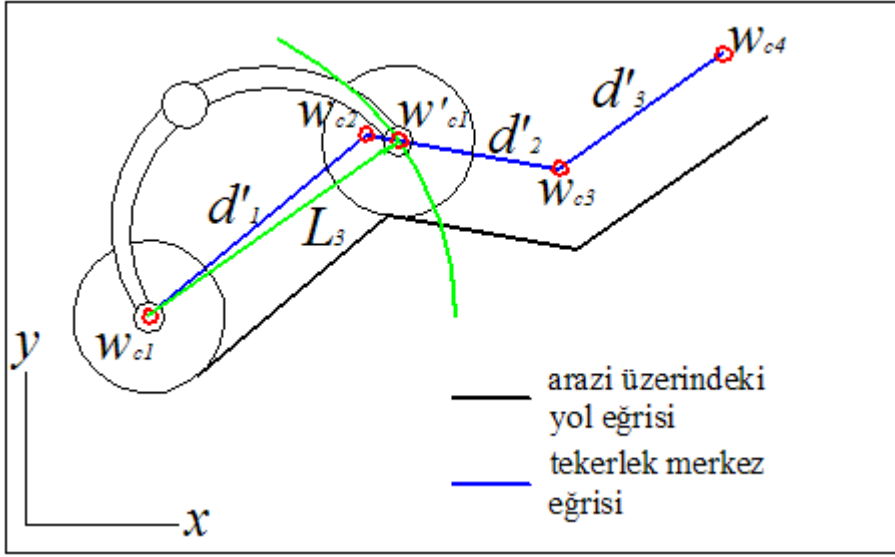
Şekil 4.2 Verilen nokta bulutu ve hesaplanan yollar (Kilit, Yontar, 2009).

5. ARACIN KİNEMATİĞİ

Yüksek hareket kabiliyetli araçların 3 boyutlu kinematik analizi oldukça karmaşıktır (Tarokh, McDermott, Hayati and Hung, 1999). Hesaplamaları kolaylaştırabilmek için sağ, sol ve orta salıncak gruplarının her zaman düşey olduğu kabul edilerek problem basitleştirilmiştir. Bu yaklaşım mafsal bağlantılarının 2 boyutta hesaplanabilmesini mümkün kılmıştır. Bir önceki bölümde üç tekerlek grubunun arazi üzerinde aldığı yollar yatayda eşit aralıklı ama farklı yüksekliklere sahip noktalar olarak hesaplanmıştır. Bu m adet noktayı doğrularla birleştirirsek, aracın $m-1$ adet doğru parçasından oluşan yolunu verir (Şekil 5.1a ve Şekil 5.1b). Daha yüksek bir m sayısı verilmesiyle giderek daha hassas ve uygun yumuşaklıkta eğri elde edilebilir, ancak bu sayının yükselmesiyle bilgisayarın hesaplama zamanı da uzar. Bu yüzden kullanıcı deneme yanılma ile uygun bir m değeri seçmelidir. Yol eğrisini oluşturan doğru parçaları Şekil 5.1a'daki gibi d_1, d_2, \dots, d_{m-1} olarak adlandırılmıştır.



Şekil 5.1 Tekerlek merkez eksenlerinin eğrisinin hesaplanması.



Şekil 5.3 Tekerlek eksenlerinin pozisyonlarının belirlenmesi.

İlk olarak orta salıncak grubunun ilk tekerleği, bu grup için belirlenmiş yol üzerinde yerleştirilir, diğer tekerleğin merkezi, ilk tekerlek eksenini merkezli ve L yarıçaplı çember ile tekerlek eksenini eğrisinin kestirilmesi ile bulunur ve salıncığın orta (üst) mafsalının yeri belirlenir. Bu işlem için önce sırayla iki tekerlek arasındaki $L_3 = L$ uzaklığı, tekerlek merkez noktaları arasındaki uzaklıkla karşılaştırılır. Bu L yarıçaplı çemberin kestiği yer, ikinci tekerlek merkezinin yerini göstermektedir. Şekil 5.3'de ve Şekil 5.4'de bu işlemin ayrıntılarını görebilirsiniz. Şekil 5.4'deki pseudocode'da (taslak kodda) kullanılan denklemleri aşağıda görebilirsiniz.

$$l_{ji} = \sqrt{(w_{ci_x} - w_{cj_x})^2 + (w_{ci_y} - w_{cj_y})^2} \quad (i=1,2,\dots,m-1), (j=1,2,\dots,m-1) \quad (5.1)$$

Doğru ile çemberin kesiminden çıkacak çözüm ile ikinci tekerlek merkezini bulunması için gereken denklemler aşağıdadır.

$$y = m_i x + b_i \quad (5.2)$$

$$(x - w_{ci_x})^2 + (y - w_{ci_y})^2 = L_3^2 \quad (5.3)$$

Denklem 5.2 ve Denklem 5.3'ün ortak çözümü

$$m_i = \frac{w_{ci+1y} - w_{ciy}}{w_{ci+1x} - w_{cix}}, b_i = \frac{w_{ciy}(w_{ci+1x} - w_{cix}) - w_{cix}(w_{ci+1y} - w_{ciy})}{w_{ci+1x} - w_{cix}} \text{ için yapılırsa;}$$

$$f(x) = S_1x^2 + S_2x + S_3 \quad (5.4)$$

$$S_1 = (1 + m_i), S_2 = 2(-b_iw_{cix} + b_im_i - w_{cix}), S_3 = (b_i - w_{ciy})^2 - L_3^2 + w_{cix}^2 \text{ olmak}$$

üzere elde edilir. Bulunan Denklem 5.4'deki $f(x)$ 'in ikiye bölme (bisection) yöntemi ile çözümünden gelecek kök Şekil 5.4'deki pseudocode'da (ham kodda) aranan ikinci tekerlek ekseninin koordinatlarını verecektir.

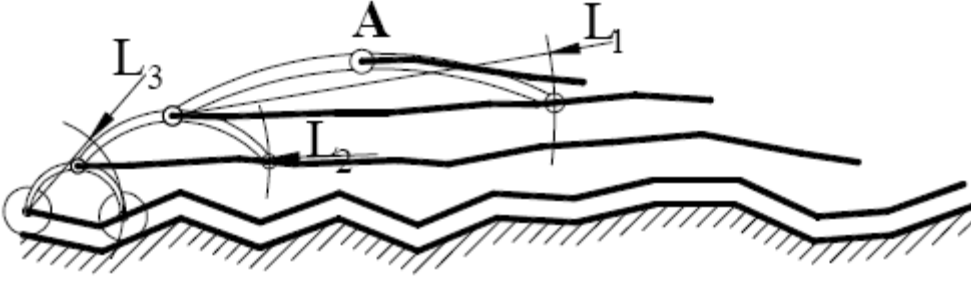
```

1. read  $w_{qj}, w_{cj} (j=1, 2, \dots, m)$ 
2.  $j=1$ 
3.  $i=1$ 
4. if  $L_3 = l_{ji}$  then  $w'_{cj} = w_{ci+1}$  else goto 5
5. if  $L_3 > l_{ji}$  then goto 8
6. if  $L_3 < l_{ji}$  then  $w'_{cj} = (\text{solve } f(x) \text{ with bisection})$  else goto 9
7.  $j = j+1$ 
8.  $i = i+1$ , goto 3
9. write  $w'_{cj} (j=1, 2, \dots, m)$ , END

```

Şekil 5.4 Tekerlek eksenlerinin belirlenmesinde kullanılan algoritmanın taslak kodu.

Yukarıda tanımlanan işlemler ile ilk kademe salıncak mekanizmasının da eğrisi elde edilir. Aynı yöntemi kullanarak, tüm mafsal noktalarının, Şekil 5.5'de gösterildiği gibi, pozisyonları kolaylıkla bulunabilir. En üst mafsal **A** araç gövdesine bağlantı mafsalıdır. Benzer şekilde, diğer arka salıncak gruplarının da **B** ve **C** olarak bağlantı mafsal eğrileri bulunur. Bu üç konum vektörü **A**, **B** ve **C**; araç gövdesinin yalpalama açılarının hesaplanmasında kullanılacaktır.

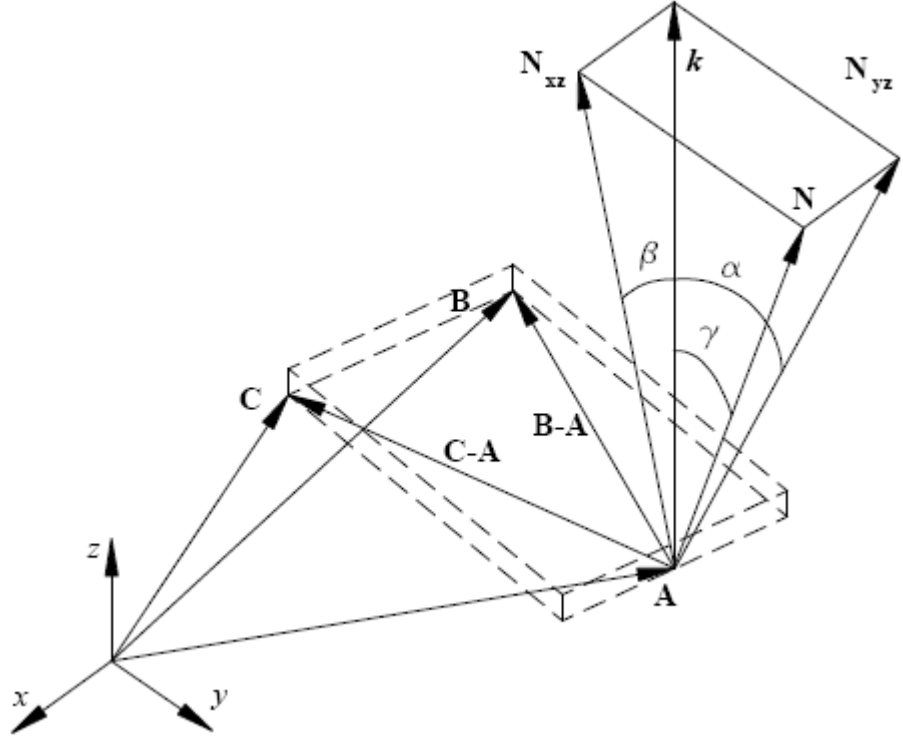


Şekil 5.5 Mafsal noktaları yörüngelerinin geometrik hesabı (Kilit, Yontar, 2009).

Hesaplamalar için MATLAB programının kodları EK 1’de bulunabilir.

6. KARARLILIK (STABİLİTE)

Mars araçları oldukça düşük hızlarından (ortalama 0.01m/s) dolayı yarı-statik (quasi-static) modellerle tasarlanırlar. Bu yüzden dinamik etkiler göz ardı edilebilir ve araç gövdesinin yalpalaması tek başına aracın kararlılık performansı için seçilebilir.



Şekil 6.1 Normal vektörü ve açılar (Kilit, Yontar, 2009).

Üç bağlantı noktasının konum vektörleri \mathbf{A} , \mathbf{B} ve \mathbf{C} Şekil 6.1'de gösterilmiştir. Burada \mathbf{i} , \mathbf{j} ve \mathbf{k} sırayla x , y ve z yönündeki birim vektörlerdir. Katı bir gövde 3 boyutlu uzayda üç konum vektörü ile tanımlanabilir. $\mathbf{B}-\mathbf{A}$ ve $\mathbf{C}-\mathbf{A}$ vektörleri beraberce araç gövdesinin üst düzlemini göstermektedir. Bu üst düzlemin normal vektörü \mathbf{N} aşağıdaki gibi hesaplanabilir.

$$\mathbf{N} = (\mathbf{B} - \mathbf{A}) \times (\mathbf{C} - \mathbf{A}). \quad (6.1)$$

$A=(A_x, A_y, A_z)^T$, $B=(B_x, B_y, B_z)^T$ ve $C=(C_x, C_y, C_z)^T$ alınarak normal vektörü N aşağıdaki gibi bulunabilir:

$$N = \begin{pmatrix} (B_y - A_y)(C_z - A_z) - (B_z - A_z)(C_y - A_y) \\ (B_z - A_z)(C_x - A_x) - (B_x - A_x)(C_z - A_z) \\ (B_x - A_x)(C_y - A_y) - (B_y - A_y)(C_x - A_x) \end{pmatrix}. \quad (6.2)$$

Aracın yalpalaması, N ve k vektörleri arasındaki açı γ ile ölçülür. Bu yüzden γ kararlılığın doğrudan ölçümüdür. Yüksek kararlılığa sahip araçlar için γ olabildiğince küçük olmalıdır. Kararlılık açısı aşağıdaki gibi hesaplanmıştır:

$$\gamma = \cos^{-1} \left(\frac{k \cdot N}{|N|} \right). \quad (6.3)$$

Denkleme bileşenleri koyarsak sonuçta

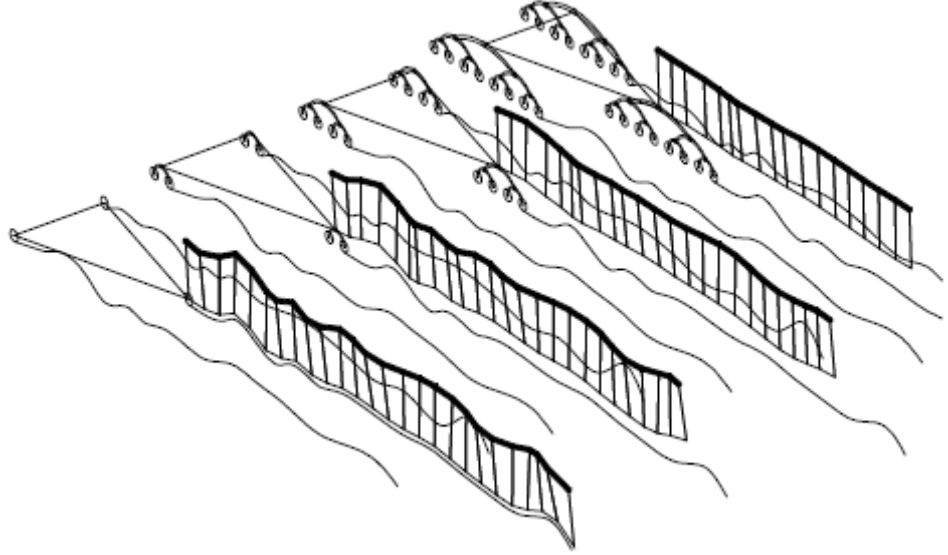
$$\gamma = \cos^{-1} \left(\frac{N_z}{\sqrt{N_x^2 + N_y^2 + N_z^2}} \right) < \gamma_{\max}. \quad (6.4)$$

Elde edilir.

γ açısı $[0, \gamma_{\max}]$ aralığında kalırsa, araç için “kararlıdır” denir. Eğer belli bir yol için hesaplanan γ değerinin, γ_{\max} değerini aştığı bulunuyorsa, önceden belirlenen bu yol değiştirilmelidir. Bu durum γ ’nın engebeli arazi araçlarının yol planlama görevlerinde belirleyici bir kriter olabileceğini göstermektedir. Bazı durumlarda enine ve boyuna eksenlerin açısal değişimlerinin de ayrı ayrı belirlenmesi gerekebilir. Bu değişimler α ve β olarak Şekil 6.3’de gösterilmişlerdir. $N_{yx} = N_y \mathbf{j} + N_z \mathbf{k}$ ve $N_{yz} = N_x \mathbf{i} + N_z \mathbf{k}$ olduğu göz önüne alındığında, α ve β açıları şu şekilde bulunabilir:

$$\alpha = \cos^{-1} \left(N_z / \sqrt{N_y^2 + N_z^2} \right) < \alpha_{\max}$$

$$\beta = \cos^{-1} \left(N_z / \sqrt{N_x^2 + N_z^2} \right) < \beta_{\max}$$
(6.5)



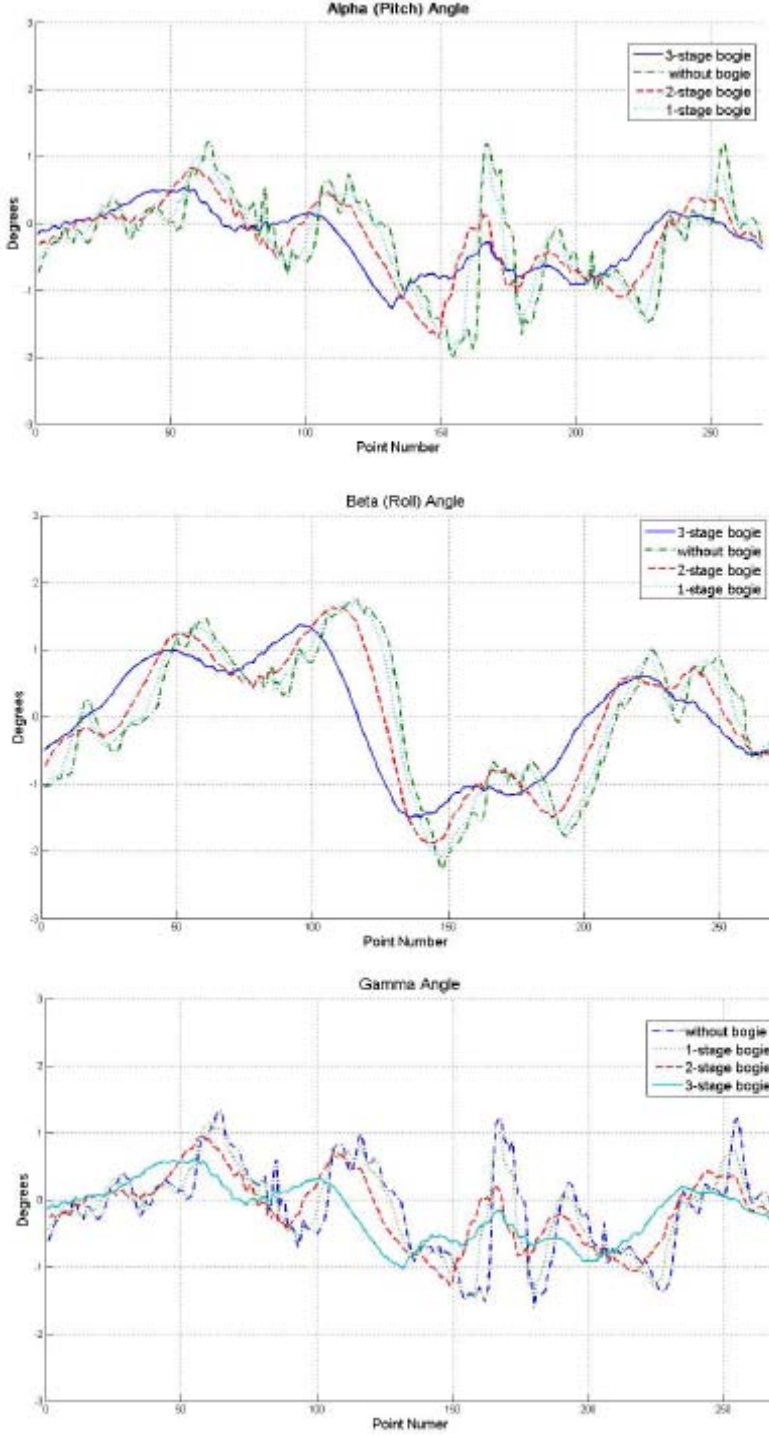
Şekil 6.2 Farklı kademelerden oluşan araçların uyum karşılaştırması (Kilit, Yontar, 2009).

Şekil 6.2’de \mathbf{N} normal vektörünün değişik durumlar için çizimi bulunmaktadır. Araç dört farklı durum için, yani 0 (salıncak mekanizması olmadan), 1, 2 ve 3 kademli olarak, aynı yol üzerinde hareket ettirilmiştir. Şekil 6.2’de dikkat edilmesi gereken salıncak mekanizması olmadan aracın normal vektörü \mathbf{N} ’nin ne kadar dalgalı bir yapıda olduğu ve 1-2-3 kademeli durumlarda nasıl giderek daha düz bir yapıya ulaştığıdır. Şekil 6.3’de bu durum daha ayrıntılı çizilmiştir. Bu dört durum α, β (enine ve boyuna yalpalama açıları) ve γ (normal vektörün düşey eksene göre yaptığı yalpalama açısı) açılarının grafikleri ile karşılaştırılmıştır. Aracın kademe sayısı arttıkça kararlılık verimi (stabilite performansı) artmaktadır.

Şekil 6.2’de elde edilen eğrilerin bir özelliği daha dikkati çekmektedir. Kademe sayılarının artması sonucu elde edilen eğriler sönümlenmelerinin yanı sıra giderek geriye (sola) doğru kaymaktadır. Bunun nedeni bir tekerin yerine bir salıncak mekanizmasının gelmesiyle $L/2$ kadar önde ve geride birer teker gelmesidir. Dolayısı ile yoldaki herhangi bir düzensizlik, salıncak

mekanizmasının mafsalını daha erken etkilemektedir. Bu nedenle kademe sayısı arttıkça eğri bir miktar geriye doğru kaymakta ve daha yassı hale gelmektedir.

Hesaplamalar için MATLAB programının kodları EK 1’de bulunabilir.

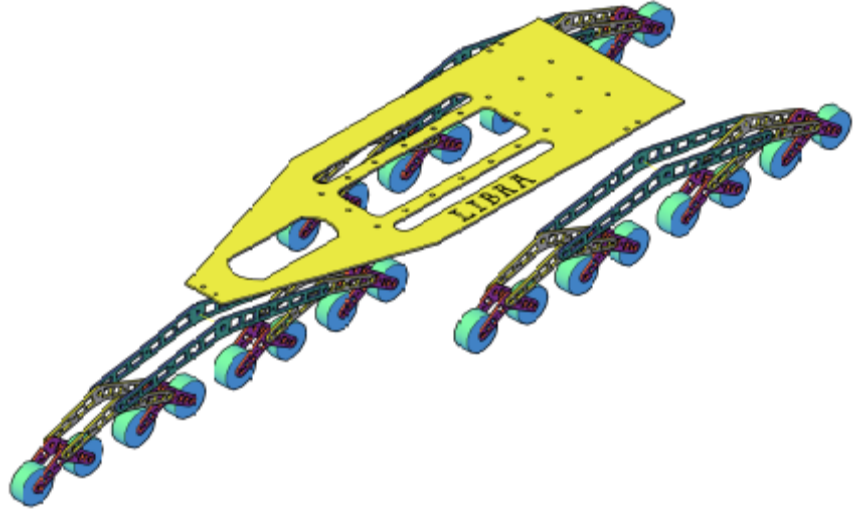


Şekil 6.3 Kararlılık (Stabilite) parametreleri (Kilit, Yontar, 2009).

7. PROTOTİP

7.1 Tasarım

Yapılan hesaplamalar sonucu araziye uyum sağlayabilen çok kademeli salıncak mekanizmalı aracın tasarımı gerçekleştirilmiştir. Prototip için yapılan tasarımın ölçüleri EK 2’de bulunabilir. Yapılan tasarım sonucu ortaya çıkan aracın 3 boyutlu katı modeli, bağlantı elemanları olmadan, Şekil 7.1’de görülebilir. Tasarım gerçekleştirilirken aracın tahrik sistemi ve dümen (direksiyon) sistemi üretim için öngörülmemiştir.



Şekil 7.1 Bağlantı elemanları olmadan prototip aracın 3B bilgisayar destekli çizimi.

Tasarımı yapılan araca salıncak mekanizmasından öykünülerek özgün bir isim koyulmaya karar verilmiş ve araca LIBRA adı verilmiştir.

7.2 Prototip Üretimi

Araziye uyum sağlayabilen çok kademeli salıncak mekanizmalı araç LIBRA'nın tasarımdan çıkartılan imalat resimleri ile prototip üretimi gerçekleştirilmiştir. İmalat resimleri EK 2’de bulunabilir. Prototip maliyetini ve üretilebilirliğini koruyabilmek için AISI1008 (St37) tipi kolay işlenebilir çelik malzeme seçilmiştir. Tekerlekler sanayi tipi poliamid malzemeden standart

tekerleklerdir. Prototip üretimi tamamlanmış aracın tekerleklerine araziye daha iyi tutunabilmesi için plastik-kauçuk malzemeden lastikler yapıştırılmıştır.

Şekil 7.2’de görüldüğü gibi prototip LIBRA engebeli araziye uyum sağlayarak üzerinden bulunduğu arazi geometrisinin düzensizliklerini sönümleyerek araç gövdesine iletmektedir ve yol düzensizlikleri artan kademe sayısı ile bağlantılı olarak araç gövdesi tarafından daha az “hissedilmektedir”.



Şekil 7.2 LIBRA'nın prototip fotoğrafı

8. SONUÇ

Bu çalışmada yeni bir üç kademeli salıncak mekanizması tanımlanmıştır. Bu mekanizmanın bir araca (özellikle de bir dış-uzay gezegen araştırma aracına) uygulanması incelenmiştir. Aracın test edilmesi için, oluşturulan ya da girilen arazi verisine göre yol modellemesi için yeni bir yöntem sunulmuştur. Bu yöntem ile aracı arazi üzerinde bilgisayar yardımı ile test etmek için arazinin tümüyle modellenmesine gerek kalmamaktadır. MATLAB'de hazırlanan program yardımıyla yol modellemesi, bu yol üzerinde yalpalama açılarının kararlılık kriteri olarak hesaplanması ve kullanılması için yeni bir yöntem geliştirilmiştir. Bu yöntem ile hesaplanan değerler karşılaştırılarak yöntemin uygulaması gösterilmiştir.

Yapılan tüm çalışmalar sonucu elde edilen sonuçlardan çok kademeli salıncak mekanizmalı bir aracın araziye uyumunda ve kararlılık performansında gelişme sağlandığı görülmektedir. Elde edilen sonuçlarla, önerilen mekanizmanın 3 boyutlu olarak bir araca uygulamasının prototip üretimi gerçekleştirilmiş ve yapılan çalışmanın doğruluğu gösterilmiştir.

9. ÖNERİLER

Bu tez çalışmasında çok kademeli salıncak mekanizmasının 3 boyutta uygulaması incelenmiş ve bu inceleme bir araç üzerinde hem bilgisayar hem de prototip olarak uygulamada görülmüştür. Bundan sonra yapılabilecek çalışmaların ana başlıklarını şöyle sıralayabiliriz:

- Aracın tahrik sistemi: Aracın sahip olduğu salıncak mekanizması aktif kontrol gerektirmeden doğası gereği dengeyi sağlayarak araziye uyumu ve bununla birlikte kararlılığı sağlamaktadır. Ancak bu mekanizmanın dezavantajı olarak görülebilecek tekerlek sayısının fazlalığı için, (örneğin) tek bir motor ve her tekerleğin farklı hızlarda dönmesini sağlayacak bir tahrik sistemi geliştirilebilir.
- Aracın dümenleme (direksiyon) sistemi: Tahrik sisteminde olduğu gibi tekerlek sayısının fazlalığının dezavantaj olmaması ve pasif kontrol avantajının korunması amaçlanarak, aracın dümen sistemi üzerinde çalışılabilir. Sürtünmeleri minimum değerlerde tutacak stabil bir dönme için gerekli olan mekanizmanın tasarımı gerçekleştirilebilir.
- Tekerlek ve kademe geometrik ölçülerinin optimizasyonu: Farklı yüksekliklerde ve hatta değişken yüksekliklerde salıncaklar ile optimum değerler için çalışma gerçekleştirilerek arazi yapısına göre tasarım parametreleri geliştirilebilir.

KAYNAKLAR DİZİNİ

- Dudek, G., Jenkin, M.**, 2000, Computational Principles of Mobile Robotics, Cambridge, United Kingdom.
- Farritor, S., Hcaot, H., Dubowsky, S.**, 1998, Physics Based Planning For Planetary Exploration, IEEE International Conference on Robotic and Automation.
- Jonathan, Steuer**, 1992, Defining Virtual Reality: Dimensions Determining Telepresence, Journal of Communication, 42(4), s73-93.
- JPL-NASA**, 2001, “Sojourner”, <http://www2.jpl.nasa.gov/basics/sj.html> (Erişim tarihi: 02 Ocak 2010).
- Kilit, Ö.**, 2005, Application of a Multi-Stage Bogie Suspension System to the Planet Exploration Vehicles, Proc. of 2nd International Conference on "Recent Advances in Space Technologies", İstanbul, Turkey, s304-308.
- Kilit, Ö., Yontar, A.**, 2009, Stability of a New Mars Rover with Multi-stage Bogie Mechanism, Proc. of 4nd International Conference on "Recent Advances in Space Technologies", İstanbul, Turkey, s145-148.
- MER-NASA (1)**, 2009, “Spirit”, http://www.nasa.gov/mission_pages/mer/freespirit.html (Erişim tarihi: 02 Ocak 2010).
- MER-NASA (2)**, 2009, “Spirit Rocker-bogie Mechanism”, <http://www.nasa.gov/externalflash/mer20090129/index.html> (Erişim tarihi: 02 Ocak 2010).
- NASA**, 2004, National Aeronautics and Space Administration (NASA) Mars Exploration Rover Landings Press Kit, USA.
- NASA**, 2009, “NASA's Mars Rover has Uncertain Future as Sixth Anniversary Nears”, http://www.nasa.gov/mission_pages/mer/news/mer2009/mer20091231.html (Erişim tarihi: 02 Ocak 2010).
- Sandin, P.E.**, 2003, Robot Mechanisms and Mechanical Devices Illustrated, McGraw Hill, New York.
- Tarokh, M., McDermott, G., Hayati, S., Hung, J.**, 1999, Kinematic Modeling of a High Mobility Mars Rover, Proc. IEEE Int. Conf. Robotics and Automation, Detroit, MI, s992-998.

KAYNAKLAR DİZİNİ (devamı)

USPTO, 1989, Articulated Suspension System, The United States Patent and Trademark Office, USA, Patent No: 4840394.

Wikipedia, 2009, “Lonokhod 1”, http://tr.wikipedia.org/wiki/Lunokhod_1
(Erişim tarihi: 02 Ocak 2010).

EKLER

EK-1 MATLAB KODLARI

Yol modellemesinde ve kararlılığın analizinde kullanılan algoritmaların kod haline getirilmesi ile oluşturulan programın MATLAB programı kodlarını aşağıda bulunmaktadır.

```

clear
%Start-----Input parameters
part=300; %input number of
parts for path %input initial
XI o=20. 0; YI o=25. 0; %input target
XTo=55. 0; YTo=47. 0; %input DTP control
R=3. 0; %input wheel
di ameter %input stage1
r0=0. 20; %input stage2
radius, offset length %input stage3
r1=0. 60; %input vehicle
length %input vehicle
r2=1. 20; %points of terrain,
length (x value=A1: Cxxx)
r3=2. 40; %points of terrain,
length (x value=A1: Cxxx)
%Fi ni sh-----Input parameters

al fa1=atan((YTo-YI o)/(XTo-XI o));
teta=-al fa1;
Rot=[cos(teta) -si n(teta) 0;
      si n(teta)  cos(teta) 0;
      0 0 1];

reRot=[cos(al fa1) -si n(al fa1) 0;
        si n(al fa1)  cos(al fa1) 0;
        0 0 1];

for i=1: length(Poi nts) %rotation of
Poi nts
    rotPoi ntsT=Rot*[Poi nts(i, 1); Poi nts(i, 2); Poi nts(i, 3)];
    rotPoi nts(i, 1)=rotPoi ntsT(1, 1);
    rotPoi nts(i, 2)=rotPoi ntsT(2, 1);
    rotPoi nts(i, 3)=rotPoi ntsT(3, 1);
end

fi gure(1)
hold all
grid on
e=1: length(Poi nts);
pl ot3(rotPoi nts(e, 1), rotPoi nts(e, 2), rotPoi nts(e, 3), 'gx')

%Start-----1. Chai n Cal cul ati on

```

```

I Pos_1=[XI o; YI o; 0];
rotI Pos_1=Rot*I Pos_1;
TPos_1=[XT o; YT o; 0];
rotTPos_1=Rot*TPos_1;

rotI Pos_1(1, 1)=rotI Pos_1(1, 1)-(vI 1);
rotTPos_1(1, 1)=rotTPos_1(1, 1)-(vI 1);
rotI Pos_1(2, 1)=rotI Pos_1(2, 1)+(vI 2/2);
rotTPos_1(2, 1)=rotTPos_1(2, 1)+(vI 2/2);

I Pos2_1=reRot*rotI Pos_1;
TPos2_1=reRot*rotTPos_1;

XI_1=I Pos2_1(1, 1);
YI_1=I Pos2_1(2, 1);
XT_1=TPos2_1(1, 1);
YT_1=TPos2_1(2, 1);

Path_1(1, 1)=XI_1; Path_1(1, 2)=YI_1;
for a=2:part
    Path_1(a, 1)=((XT_1-XI_1)/part)+Path_1(a-1, 1);
    Path_1(a, 2)=((YT_1-YI_1)/part)+Path_1(a-1, 2);
end
Path_1(part+1, 1)=XT_1; Path_1(part+1, 2)=YT_1;

for j=1:part+1
    p=0;
    for i=1:length(Poi nts) %l ength(Poi nts)
        points read from xls file
        d=(((Path_1(j, 1)-Poi nts(i, 1))^2)+((Path_1(j, 2)-
Poi nts(i, 2))^2));
        if d<=(R^2)
            p=p+1;
            tmp(p, 1)=Poi nts(i, 1);
            tmp(p, 2)=Poi nts(i, 2);
            tmp(p, 3)=Poi nts(i, 3);
            d2(p)=d;
        else end
    end
    z1=0; z2=0;
    for q=1:p
        z1=z1+(tmp(q, 3)/d2(q));
        z2=z2+(1.0/d2(q));
    end
    Path_1(j, 3)=z1/z2;
end

%offset of path

for i=1:part+1 %rotation of path
    points
        rotPathT_1=Rot*[Path_1(i, 1); Path_1(i, 2); Path_1(i, 3)];
        rotPath_1(i, 1)=rotPathT_1(1, 1);
        rotPath_1(i, 2)=rotPathT_1(2, 1);
        rotPath_1(i, 3)=rotPathT_1(3, 1);
end

%offset of lines

(rotPath)
for i=1:part
    m_1(i)=(rotPath_1(i+1, 3)-rotPath_1(i, 3))/(rotPath_1(i+1, 1)-
rotPath_1(i, 1));
    b_1(i)=((rotPath_1(i, 3)*(rotPath_1(i+1, 1)-rotPath_1(i, 1)))-
(rotPath_1(i, 1)*(rotPath_1(i+1, 3)-
rotPath_1(i, 3))))/(rotPath_1(i+1, 1)-rotPath_1(i, 1));
    al fa_1(i)=atan(m_1(i));
end

j=0;
for i=1:part

```

```

for k=0: 1
    if al fa_1(i)==0
        offsetrotPath_1(i+j+k, 1)=rotPath_1(i+k, 1);
    else if al fa_1(i)>0
        offsetrotPath_1(i+j+k, 1)=rotPath_1(i+k, 1)-
r0*sin(abs(al fa_1(i)));
    else
offsetrotPath_1(i+j+k, 1)=rotPath_1(i+k, 1)+r0*sin(abs(al fa_1(i)));
        end
    end
    if al fa_1(i)==0
        offsetrotPath_1(i+j+k, 3)=rotPath_1(i+k, 3)+r0;
    else
offsetrotPath_1(i+j+k, 3)=rotPath_1(i+k, 3)+r0*cos(al fa_1(i));
        end
        offsetrotPath_1(i+j+k, 2)=rotPath_1(i+k, 2);
    end
    j=j+1;
end

for i=1: part
    offsetm_1(i)=(offsetrotPath_1(2*i, 3)-offsetrotPath_1(2*i-
1, 3))/(offsetrotPath_1(2*i, 1)-offsetrotPath_1(2*i-1, 1));
    offsetb_1(i)=((offsetrotPath_1(2*i-
1, 3)*(offsetrotPath_1(2*i, 1)-offsetrotPath_1(2*i-1, 1)))-
(offsetrotPath_1(2*i-1, 1)*(offsetrotPath_1(2*i, 3)-
offsetrotPath_1(2*i-1, 3))))/(offsetrotPath_1(2*i, 1)-
offsetrotPath_1(2*i-1, 1));
end

orPathPoi nts_1(1, 1)=offsetrotPath_1(1, 1);
orPathPoi nts_1(1, 2)=rotPath_1(1, 2);
orPathPoi nts_1(1, 3)=offsetrotPath_1(1, 3);
for i=2: part
    orPathPoi nts_1(i, 1)=(-offsetb_1(i-
1)+offsetb_1(i))/(offsetm_1(i-1)-offsetm_1(i));

orPathPoi nts_1(i, 3)=offsetm_1(i)*orPathPoi nts_1(i, 1)+offsetb_1(i)
;
    orPathPoi nts_1(i, 2)=rotPath_1(i, 2);
end
orPathPoi nts_1(part+1, 1)=offsetrotPath_1(2*part, 1);
orPathPoi nts_1(part+1, 2)=rotPath_1(part+1, 2);
orPathPoi nts_1(part+1, 3)=offsetrotPath_1(2*part, 3);

                                                                    %stage-1

for i=1: part
    for j=1: part
        l1=((orPathPoi nts_1(j, 1)-
orPathPoi nts_1(i, 1))^2+(orPathPoi nts_1(j, 3)-
orPathPoi nts_1(i, 3))^2)^0.5;
        l2=((orPathPoi nts_1(j+1, 1)-
orPathPoi nts_1(i, 1))^2+(orPathPoi nts_1(j+1, 3)-
orPathPoi nts_1(i, 3))^2)^0.5;
        if (l2>=r1 & l1<r1)
            mw_1(j)=(orPathPoi nts_1(j+1, 3)-
orPathPoi nts_1(j, 3))/(orPathPoi nts_1(j+1, 1)-orPathPoi nts_1(j, 1));
            bw_1(j)=((orPathPoi nts_1(j, 3)*(orPathPoi nts_1(j+1, 1)-
orPathPoi nts_1(j, 1)))-
(orPathPoi nts_1(j, 1)*(orPathPoi nts_1(j+1, 3)-
orPathPoi nts_1(j, 3))))/(orPathPoi nts_1(j+1, 1)-
orPathPoi nts_1(j, 1));

            a=orPathPoi nts_1(j, 1);
            b=orPathPoi nts_1(j+1, 1);
            fa=(a-orPathPoi nts_1(i, 1))^2+(a*mw_1(j)+bw_1(j)-
orPathPoi nts_1(i, 3))^2-r1^2;

```

```

        fb=(b-orPathPoints_1(i, 1))^2+(b*mw_1(j)+bw_1(j)-
orPathPoints_1(i, 3))^2-r1^2;

        while(abs(b-a)>0.00001)
            c=(a+b)/2;
            fc=(c-orPathPoints_1(i, 1))^2+(c*mw_1(j)+bw_1(j)-
orPathPoints_1(i, 3))^2-r1^2;
            if (sign(fb)*sign(fc)<=0)
                a=c;
                fa=fc;
            else
                b=c;
                fb=fc;
            end
        end
        wc_1(i, 1)=c;
        wc_1(i, 3)=mw_1(j)*c+bw_1(j);
        ta=(wc_1(i, 3)-orPathPoints_1(i, 3))/(wc_1(i, 1)-
orPathPoints_1(i, 1));
        k1=r1/4;

                                                %stage-1 joint
center
        s1c_1(i, 1)=((orPathPoints_1(i, 1)+wc_1(i, 1))/2)-
        (sign(atan(ta))*k1*sin(abs(atan(ta))));

s1c_1(i, 3)=((orPathPoints_1(i, 3)+wc_1(i, 3))/2)+k1*cos(abs(atan(ta
))) );
        s1c_1(i, 2)=rotPath_1(i, 2);
        else end
    end
end

                                                %stage-2
k=length(s1c_1)
for i=1:k-1
    for j=1:k-1
        l1=((s1c_1(j, 1)-s1c_1(i, 1))^2+(s1c_1(j, 3)-
s1c_1(i, 3))^2)^0.5;
        l2=((s1c_1(j+1, 1)-s1c_1(i, 1))^2+(s1c_1(j+1, 3)-
s1c_1(i, 3))^2)^0.5;
        if (l2>=r2 & l1<r2)
            mw_1(j)=(s1c_1(j+1, 3)-s1c_1(j, 3))/(s1c_1(j+1, 1)-
s1c_1(j, 1));
            bw_1(j)=((s1c_1(j, 3)*(s1c_1(j+1, 1)-s1c_1(j, 1)))-
(s1c_1(j, 1)*(s1c_1(j+1, 3)-s1c_1(j, 3)))/(s1c_1(j+1, 1)-
s1c_1(j, 1));

            a=s1c_1(j, 1);
            b=s1c_1(j+1, 1);
            fa=(a-s1c_1(i, 1))^2+(a*mw_1(j)+bw_1(j)-s1c_1(i, 3))^2-
r2^2;
            fb=(b-s1c_1(i, 1))^2+(b*mw_1(j)+bw_1(j)-s1c_1(i, 3))^2-
r2^2;

            while(abs(b-a)>0.00001)
                c=(a+b)/2;
                fc=(c-s1c_1(i, 1))^2+(c*mw_1(j)+bw_1(j)-
s1c_1(i, 3))^2-r2^2;
                if (sign(fb)*sign(fc)<=0)
                    a=c;
                    fa=fc;
                else
                    b=c;
                    fb=fc;
                end
            end
            wc_1(i, 1)=c;
            wc_1(i, 3)=mw_1(j)*c+bw_1(j);
            ta=(wc_1(i, 3)-s1c_1(i, 3))/(wc_1(i, 1)-s1c_1(i, 1));

```



```

        k2=r2/4;
        %stage-2 joint
        center
            s2c_1(i, 1)=((s1c_1(i, 1)+wc_1(i, 1))/2)-
            (sign(atan(ta))*k2*sin(abs(atan(ta))));
        s2c_1(i, 3)=((s1c_1(i, 3)+wc_1(i, 3))/2)+k2*cos(abs(atan(ta)));
        s2c_1(i, 2)=rotPath_1(i, 2);
        else end
        end
    end

        %stage-3
        l=length(s2c_1)
        for i=1:l-1
            for j=1:l-1
                l1=((s2c_1(j, 1)-s2c_1(i, 1))^2+(s2c_1(j, 3)-
                s2c_1(i, 3))^2)^0.5;
                l2=((s2c_1(j+1, 1)-s2c_1(i, 1))^2+(s2c_1(j+1, 3)-
                s2c_1(i, 3))^2)^0.5;
                if (l2>=r3 & l1<r3)
                    mw_1(j)=(s2c_1(j+1, 3)-s2c_1(j, 3))/(s2c_1(j+1, 1)-
                    s2c_1(j, 1));
                    bw_1(j)=((s2c_1(j, 3)*(s2c_1(j+1, 1)-s2c_1(j, 1)))-
                    (s2c_1(j, 1)*(s2c_1(j+1, 3)-s2c_1(j, 3))))/(s2c_1(j+1, 1)-
                    s2c_1(j, 1));

                    a=s2c_1(j, 1);
                    b=s2c_1(j+1, 1);
                    %bi section
                    fa=(a-s2c_1(i, 1))^2+(a*mw_1(j)+bw_1(j)-s2c_1(i, 3))^2-
                    r3^2;
                    fb=(b-s2c_1(i, 1))^2+(b*mw_1(j)+bw_1(j)-s2c_1(i, 3))^2-
                    r3^2;

                    while(abs(b-a)>0.00001)
                        c=(a+b)/2;
                        fc=(c-s2c_1(i, 1))^2+(c*mw_1(j)+bw_1(j)-
                        s2c_1(i, 3))^2-r3^2;
                        if (sign(fb)*sign(fc)<=0)
                            a=c;
                            fa=fc;
                        else
                            b=c;
                            fb=fc;
                        end
                    end
                    end
                    wc_1(i, 1)=c;
                    wc_1(i, 3)=mw_1(j)*c+bw_1(j);
                    ta=(wc_1(i, 3)-s2c_1(i, 3))/(wc_1(i, 1)-s2c_1(i, 1));
                    k3=r3/4;

                    %stage-3 joint
                    center
                        s3c_1(i, 1)=((s2c_1(i, 1)+wc_1(i, 1))/2)-
                        (sign(atan(ta))*k3*sin(abs(atan(ta))));
                    s3c_1(i, 3)=((s2c_1(i, 3)+wc_1(i, 3))/2)+k3*cos(abs(atan(ta)));
                    s3c_1(i, 2)=rotPath_1(i, 2);
                    else end
                    end
                end
            end

        %m=length(s3c_1)
        %for i=1:m
        %re-rotation of
        %points
        %    rot_s3c=reRot*[s3c_1(i, 1); s3c_1(i, 2); s3c_1(i, 3)];
        %    vj1(i, 1)=rot_s3c(1, 1);
        %    vj1(i, 2)=rot_s3c(2, 1);
        %    vj1(i, 3)=rot_s3c(3, 1);
        %end
    end
end

```

```

%xlswrite('ran_data.xls', s3c_1, 2)

for j = 1: length(s3c_1)
    p=0;
    for i = 1: length(rotPoints)
        d = (((s3c_1(j, 1) - rotPoints(i, 1))^2) + ((s3c_1(j, 2) -
rotPoints(i, 2))^2));
        if d <= (R^2)
            p = p + 1;
            tmp_1(p, 1) = rotPoints(i, 1);
            tmp_1(p, 2) = rotPoints(i, 2);
            tmp_1(p, 3) = rotPoints(i, 3);
            d2(p) = d;
        else end
    end
    z1 = 0;    z2 = 0;
    for q = 1: p
        z1 = z1 + (tmp_1(q, 3) / d2(q));
        z2 = z2 + (1.0 / d2(q));
    end
    s3cPath_1(j, 1) = s3c_1(j, 1);
    s3cPath_1(j, 2) = s3c_1(j, 2);
    s3cPath_1(j, 3) = (z1 / z2) * r0 + k1 + k2 + k3;
end

%xlswrite('ran_data.xls', s3cPath_1, 5)

figure(1)
hold all
grid on
%c = 1: length(s3c_1);
plot3(s3c_1(c, 1), s3c_1(c, 2), s3c_1(c, 3))
%c = 1: length(s3cPath_1);
plot3(s3cPath_1(c, 1), s3cPath_1(c, 2), s3cPath_1(c, 3))
c = 1: length(rotPath_1);
plot3(rotPath_1(c, 1), rotPath_1(c, 2), rotPath_1(c, 3))

%figure(2)
%hold all
%grid on
%c = 1: length(rotPath_1);    plot(rotPath_1(c, 1), rotPath_1(c, 3))
%c = 1: length(s1c_1);      plot(s1c_1(c, 1), s1c_1(c, 3))
%c = 1: length(s2c_1);      plot(s2c_1(c, 1), s2c_1(c, 3))
%c = 1: length(s3c_1);      plot(s3c_1(c, 1), s3c_1(c, 3))
%c = 1: length(s3cPath_1);   plot(s3cPath_1(c, 1), s3cPath_1(c, 3))
%axis([20 75 0 4])
%Finish-----1. Chain Calculation

%Start-----2. Chain Calculation
XI_2 = XIo;
YI_2 = YIo;
XT_2 = XTo;
YT_2 = YTo;

Path_2(1, 1) = XI_2; Path_2(1, 2) = YI_2;
for a = 2: part
    Path_2(a, 1) = ((XT_2 - XI_2) / part) + Path_2(a - 1, 1);
    Path_2(a, 2) = ((YT_2 - YI_2) / part) + Path_2(a - 1, 2);
end
Path_2(part + 1, 1) = XT_2; Path_2(part + 1, 2) = YT_2;

for j = 1: part + 1
    p = 0;
    for i = 1: length(Points)           %length(Points)
points read from xls file
        d = (((Path_2(j, 1) - Points(i, 1))^2) + ((Path_2(j, 2) -
Points(i, 2))^2));
        if d <= (R^2)
            p = p + 1;

```

```

        tmp(p, 1)=Points(i, 1);
        tmp(p, 2)=Points(i, 2);
        tmp(p, 3)=Points(i, 3);
        d2(p)=d;
    else end
end
z1=0;    z2=0;
for q=1:p
    z1=z1+(tmp(q, 3)/d2(q));
    z2=z2+(1. 0/d2(q));
end
Path_2(j, 3)=z1/z2;
end

%offset of path

for i=1: part+1                                %rotation of path
points
    rotPathT_2=Rot*[Path_2(i, 1); Path_2(i, 2); Path_2(i, 3)];
    rotPath_2(i, 1)=rotPathT_2(1, 1);
    rotPath_2(i, 2)=rotPathT_2(2, 1);
    rotPath_2(i, 3)=rotPathT_2(3, 1);
end

                                                                    %offset of lines
(rotPath)
for i=1: part
    m_2(i)=(rotPath_2(i+1, 3)-rotPath_2(i, 3))/(rotPath_2(i+1, 1)-
rotPath_2(i, 1));
    b_2(i)=((rotPath_2(i, 3)*(rotPath_2(i+1, 1)-rotPath_2(i, 1)))-
(rotPath_2(i, 1)*(rotPath_2(i+1, 3)-
rotPath_2(i, 3)))/(rotPath_2(i+1, 1)-rotPath_2(i, 1)));
    alfa_2(i)=atan(m_2(i));
end

j=0;
for i=1: part
    for k=0: 1
        if alfa_2(i)==0
            offsetrotPath_2(i+j+k, 1)=rotPath_2(i+k, 1);
        else if alfa_2(i)>0
            offsetrotPath_2(i+j+k, 1)=rotPath_2(i+k, 1)-
r0*sina(abs(alfa_2(i)));
        else
            offsetrotPath_2(i+j+k, 1)=rotPath_2(i+k, 1)+r0*sina(abs(alfa_2(i)));
        end
    end
    if alfa_2(i)==0
        offsetrotPath_2(i+j+k, 3)=rotPath_2(i+k, 3)+r0;
    else
        offsetrotPath_2(i+j+k, 3)=rotPath_2(i+k, 3)+r0*cos(alfa_2(i));
    end
    offsetrotPath_2(i+j+k, 2)=rotPath_2(i+k, 2);
end
j=j+1;
end

for i=1: part
    offsetm_2(i)=(offsetrotPath_2(2*i, 3)-offsetrotPath_2(2*i-
1, 3))/(offsetrotPath_2(2*i, 1)-offsetrotPath_2(2*i-1, 1));
    offsetb_2(i)=((offsetrotPath_2(2*i-
1, 3)*(offsetrotPath_2(2*i, 1)-offsetrotPath_2(2*i-1, 1)))-
(offsetrotPath_2(2*i-1, 1)*(offsetrotPath_2(2*i, 3)-
offsetrotPath_2(2*i-1, 3))))/(offsetrotPath_2(2*i, 1)-
offsetrotPath_2(2*i-1, 1));
end

```

```

orPathPoints_2(1, 1)=offsetrotPath_2(1, 1);
orPathPoints_2(1, 2)=rotPath_2(1, 2);
orPathPoints_2(1, 3)=offsetrotPath_2(1, 3);
for i =2: part
    orPathPoints_2(i, 1)=(-offsetb_2(i -
1)+offsetb_2(i))/(offsetm_2(i -1)-offsetm_2(i));

orPathPoints_2(i, 3)=offsetm_2(i)*orPathPoints_2(i, 1)+offsetb_2(i)
;
    orPathPoints_2(i, 2)=rotPath_2(i, 2);
end
orPathPoints_2(part+1, 1)=offsetrotPath_2(2*part, 1);
orPathPoints_2(part+1, 2)=rotPath_2(part+1, 2);
orPathPoints_2(part+1, 3)=offsetrotPath_2(2*part, 3);

                                                                 %stage-1
for i =1: part
    for j =1: part
        l1=((orPathPoints_2(j, 1)-
orPathPoints_2(i, 1))^2+(orPathPoints_2(j, 3)-
orPathPoints_2(i, 3))^2)^0.5;
        l2=((orPathPoints_2(j+1, 1)-
orPathPoints_2(i, 1))^2+(orPathPoints_2(j+1, 3)-
orPathPoints_2(i, 3))^2)^0.5;
        if (l2>=r1 & l1<r1)
            mw_2(j)=(orPathPoints_2(j+1, 3)-
orPathPoints_2(j, 3))/(orPathPoints_2(j+1, 1)-orPathPoints_2(j, 1));
            bw_2(j)=((orPathPoints_2(j, 3)*(orPathPoints_2(j+1, 1)-
orPathPoints_2(j, 1)))-
(orPathPoints_2(j, 1)*(orPathPoints_2(j+1, 3)-
orPathPoints_2(j, 3))))/(orPathPoints_2(j+1, 1)-
orPathPoints_2(j, 1));

            a=orPathPoints_2(j, 1);           %bi secti on
            b=orPathPoints_2(j+1, 1);
            fa=(a-orPathPoints_2(i, 1))^2+(a*mw_2(j)+bw_2(j)-
orPathPoints_2(i, 3))^2-r1^2;
            fb=(b-orPathPoints_2(i, 1))^2+(b*mw_2(j)+bw_2(j)-
orPathPoints_2(i, 3))^2-r1^2;

            while(abs(b-a)>0.00001)
                c=(a+b)/2;
                fc=(c-orPathPoints_2(i, 1))^2+(c*mw_2(j)+bw_2(j)-
orPathPoints_2(i, 3))^2-r1^2;
                if (si gn(fb)*si gn(fc)<=0)
                    a=c;
                    fa=fc;
                else
                    b=c;
                    fb=fc;
                end
            end
            wc_2(i, 1)=c;
            wc_2(i, 3)=mw_2(j)*c+bw_2(j);
            ta=(wc_2(i, 3)-orPathPoints_2(i, 3))/(wc_2(i, 1)-
orPathPoints_2(i, 1));
            k1=r1/4;

                                                                 %stage-1 joi nt
center
        s1c_2(i, 1)=((orPathPoints_2(i, 1)+wc_2(i, 1))/2)-
(si gn(atan(ta))*k1*si n(abs(atan(ta)))));
s1c_2(i, 3)=((orPathPoints_2(i, 3)+wc_2(i, 3))/2)+k1*cos(abs(atan(ta)
)));
        s1c_2(i, 2)=rotPath_2(i, 2);
        else end
    end
end
end

```

```

                                                                    %stage-2
k=length(s1c_2)
for i=1:k-1
    for j=1:k-1
        l1=((s1c_2(j,1)-s1c_2(i,1))^2+(s1c_2(j,3)-
s1c_2(i,3))^2)^0.5;
        l2=((s1c_2(j+1,1)-s1c_2(i,1))^2+(s1c_2(j+1,3)-
s1c_2(i,3))^2)^0.5;
        if (l2>=r2 & l1<r2)
            mw_2(j)=(s1c_2(j+1,3)-s1c_2(j,3))/(s1c_2(j+1,1)-
s1c_2(j,1));
            bw_2(j)=((s1c_2(j,3)*(s1c_2(j+1,1)-s1c_2(j,1)))-
(s1c_2(j,1)*(s1c_2(j+1,3)-s1c_2(j,3))))/(s1c_2(j+1,1)-
s1c_2(j,1));

            a=s1c_2(j,1);                %bi section
            b=s1c_2(j+1,1);
            fa=(a-s1c_2(i,1))^2+(a*mw_2(j)+bw_2(j)-s1c_2(i,3))^2-
r2^2;
            fb=(b-s1c_2(i,1))^2+(b*mw_2(j)+bw_2(j)-s1c_2(i,3))^2-
r2^2;

            while(abs(b-a)>0.00001)
                c=(a+b)/2;
                fc=(c-s1c_2(i,1))^2+(c*mw_2(j)+bw_2(j)-
s1c_2(i,3))^2-r2^2;
                if (sign(fb)*sign(fc)<=0)
                    a=c;
                    fa=fc;
                else
                    b=c;
                    fb=fc;
                end
            end
            wc_2(i,1)=c;
            wc_2(i,3)=mw_2(j)*c+bw_2(j);
            ta=(wc_2(i,3)-s1c_2(i,3))/(wc_2(i,1)-s1c_2(i,1));
            k2=r2/4;

                                                                    %stage-2 joint
center
        s2c_2(i,1)=((s1c_2(i,1)+wc_2(i,1))/2)-
        (sign(atan(ta))*k2*sin(abs(atan(ta))));
        s2c_2(i,3)=((s1c_2(i,3)+wc_2(i,3))/2)+k2*cos(abs(atan(ta)));
        s2c_2(i,2)=rotPath_2(i,2);
        else end
    end
end

                                                                    %stage-3
l=length(s2c_2)
for i=1:l-1
    for j=1:l-1
        l1=((s2c_2(j,1)-s2c_2(i,1))^2+(s2c_2(j,3)-
s2c_2(i,3))^2)^0.5;
        l2=((s2c_2(j+1,1)-s2c_2(i,1))^2+(s2c_2(j+1,3)-
s2c_2(i,3))^2)^0.5;
        if (l2>=r3 & l1<r3)
            mw_2(j)=(s2c_2(j+1,3)-s2c_2(j,3))/(s2c_2(j+1,1)-
s2c_2(j,1));
            bw_2(j)=((s2c_2(j,3)*(s2c_2(j+1,1)-s2c_2(j,1)))-
(s2c_2(j,1)*(s2c_2(j+1,3)-s2c_2(j,3))))/(s2c_2(j+1,1)-
s2c_2(j,1));

            a=s2c_2(j,1);                %bi section
            b=s2c_2(j+1,1);
            fa=(a-s2c_2(i,1))^2+(a*mw_2(j)+bw_2(j)-s2c_2(i,3))^2-
r3^2;

```

```

fb=(b-s2c_2(i,1))^2+(b*mw_2(j)+bw_2(j)-s2c_2(i,3))^2-
r3^2;

while(abs(b-a)>0.00001)
    c=(a+b)/2;
    fc=(c-s2c_2(i,1))^2+(c*mw_2(j)+bw_2(j)-
s2c_2(i,3))^2-r3^2;
    if(sign(fb)*sign(fc)<=0)
        a=c;
        fa=fc;
    else
        b=c;
        fb=fc;
    end
end
wc_2(i,1)=c;
wc_2(i,3)=mw_2(j)*c+bw_2(j);
ta=(wc_2(i,3)-s2c_2(i,3))/(wc_2(i,1)-s2c_2(i,1));
k3=r3/4;
%stage-3 joint
center
s3c_2(i,1)=((s2c_2(i,1)+wc_2(i,1))/2)-
(sign(atan(ta))*k3*sin(abs(atan(ta))));
s3c_2(i,3)=((s2c_2(i,3)+wc_2(i,3))/2)+k3*cos(abs(atan(ta)));
s3c_2(i,2)=rotPath_2(i,2);
else end
end
end

%m=length(s3c)
%for i=1:m %re-rotation of
points
% rot_s3c=reRot*[s3c(i,1); s3c(i,2); s3c(i,3)];
% vj(i,1)=rot_s3c(1,1);
% vj(i,2)=rot_s3c(2,1);
% vj(i,3)=rot_s3c(3,1);
%end

%xlswrite('ran_data.xls',s3c_2,3)

for j=1:length(s3c_2)
    p=0;
    for i=1:length(rotPoints)
        d=(((s3c_2(j,1)-rotPoints(i,1))^2)+((s3c_2(j,2)-
rotPoints(i,2))^2));
        if d<=(R^2)
            p=p+1;
            tmp_2(p,1)=rotPoints(i,1);
            tmp_2(p,2)=rotPoints(i,2);
            tmp_2(p,3)=rotPoints(i,3);
            d2(p)=d;
        else end
    end
    z1=0; z2=0;
    for q=1:p
        z1=z1+(tmp_2(q,3)/d2(q));
        z2=z2+(1.0/d2(q));
    end
    s3cPath_2(j,1)=s3c_2(j,1);
    s3cPath_2(j,2)=s3c_2(j,2);
    s3cPath_2(j,3)=(z1/z2)+r0+k1+k2+k3;
end

%xlswrite('ran_data.xls',s3cPath_2,6)

figure(1)
hold all
grid on

```

```

%c=1: length(s3c_2);
plot3(s3c_2(c, 1), s3c_2(c, 2), s3c_2(c, 3))
%c=1: length(s3cPath_2);
plot3(s3cPath_2(c, 1), s3cPath_2(c, 2), s3cPath_2(c, 3))
c=1: length(rotPath_2);
plot3(rotPath_2(c, 1), rotPath_2(c, 2), rotPath_2(c, 3))

%figure(3)
%hold all
%grid on
%c=1: length(rotPath_2);      plot(rotPath_2(c, 1), rotPath_2(c, 3))
%c=1: length(s1c_2);        plot(s1c_2(c, 1), s1c_2(c, 3))
%c=1: length(s2c_2);        plot(s2c_2(c, 1), s2c_2(c, 3))
%c=1: length(s3c_2);        plot(s3c_2(c, 1), s3c_2(c, 3))
%c=1: length(s3cPath_2);    plot(s3cPath_2(c, 1), s3cPath_2(c, 3))
%axis([20 75 0 4])
%Fi ni sh-----2. Chain Calculati on

%Start-----3. Chain Calculati on
I Pos_3=[XI o; YI o; 0];
rotI Pos_3=Rot*I Pos_3;
T Pos_3=[XT o; YT o; 0];
rotT Pos_3=Rot*T Pos_3;
vl 2
rotI Pos_3(1, 1)=rotI Pos_3(1, 1)-(vl 1);
rotT Pos_3(1, 1)=rotT Pos_3(1, 1)-(vl 1);
rotI Pos_3(2, 1)=rotI Pos_3(2, 1)-(vl 2/2);
rotT Pos_3(2, 1)=rotT Pos_3(2, 1)-(vl 2/2);

I Pos2_3=reRot*rotI Pos_3;
T Pos2_3=reRot*rotT Pos_3;

XI _3=I Pos2_3(1, 1);
YI _3=I Pos2_3(2, 1);
XT _3=T Pos2_3(1, 1);
YT _3=T Pos2_3(2, 1);

Path_3(1, 1)=XI _3; Path_3(1, 2)=YI _3;
for a=2: part
    Path_3(a, 1)=((XT _3-XI _3)/part)+Path_3(a-1, 1);
    Path_3(a, 2)=((YT _3-YI _3)/part)+Path_3(a-1, 2);
end
Path_3(part+1, 1)=XT _3; Path_3(part+1, 2)=YT _3;

for j=1: part+1
    p=0;
    for i=1: length(Poi nts)          %l ength(Poi nts)
        poi nts read from xls file
        d=((Path_3(j, 1)-Poi nts(i, 1))^2)+((Path_3(j, 2)-
        Poi nts(i, 2))^2);
        i f d<=(R^2)
            p=p+1;
            tmp(p, 1)=Poi nts(i, 1);
            tmp(p, 2)=Poi nts(i, 2);
            tmp(p, 3)=Poi nts(i, 3);
            d2(p)=d;
        el se end
    end
    z1=0;      z2=0;
    for q=1: p
        z1=z1+(tmp(q, 3)/d2(q));
        z2=z2+(1. 0/d2(q));
    end
    Path_3(j, 3)=z1/z2;
end

%offset of path

for i=1: part+1          %rotati on of path
poi nts

```

```

    rotPathT_3=Rot*[Path_3(i, 1); Path_3(i, 2); Path_3(i, 3)];
    rotPath_3(i, 1)=rotPathT_3(1, 1);
    rotPath_3(i, 2)=rotPathT_3(2, 1);
    rotPath_3(i, 3)=rotPathT_3(3, 1);
end

%for i=1:length(Poi nts)                                %rotation of
Poi nts
%   rotPoi ntsT=Rot*[Poi nts(i, 1); Poi nts(i, 2); Poi nts(i, 3)];
%   rotPoi nts(i, 1)=rotPoi ntsT(1, 1);
%   rotPoi nts(i, 2)=rotPoi ntsT(2, 1);
%   rotPoi nts(i, 3)=rotPoi ntsT(3, 1);
%end

                                                                    %offset of lines
(rotPath)
for i=1:part
    m_3(i)=(rotPath_3(i+1, 3)-rotPath_3(i, 3))/(rotPath_3(i+1, 1)-
rotPath_3(i, 1));
    b_3(i)=((rotPath_3(i, 3)*(rotPath_3(i+1, 1)-rotPath_3(i, 1)))-
(rotPath_3(i, 1)*(rotPath_3(i+1, 3)-
rotPath_3(i, 3))))/(rotPath_3(i+1, 1)-rotPath_3(i, 1));
    al fa_3(i)=atan(m_3(i));
end

j=0;
for i=1:part
    for k=0:1
        if al fa_3(i)==0
            offsetrotPath_3(i+j+k, 1)=rotPath_3(i+k, 1);
        else if al fa_3(i)>0
            offsetrotPath_3(i+j+k, 1)=rotPath_3(i+k, 1)-
r0*s i n(abs(al fa_3(i)));
        else
            offsetrotPath_3(i+j+k, 1)=rotPath_3(i+k, 1)+r0*s i n(abs(al fa_3(i)));
        end
        if al fa_3(i)==0
            offsetrotPath_3(i+j+k, 3)=rotPath_3(i+k, 3)+r0;
        else
            offsetrotPath_3(i+j+k, 3)=rotPath_3(i+k, 3)+r0*cos(al fa_3(i));
        end
        offsetrotPath_3(i+j+k, 2)=rotPath_3(i+k, 2);
    end
    j=j+1;
end

for i=1:part
    offsetm_3(i)=(offsetrotPath_3(2*i, 3)-offsetrotPath_3(2*i -
1, 3))/(offsetrotPath_3(2*i, 1)-offsetrotPath_3(2*i -1, 1));
    offsetb_3(i)=((offsetrotPath_3(2*i -
1, 3)*(offsetrotPath_3(2*i, 1)-offsetrotPath_3(2*i -1, 1)))-
(offsetrotPath_3(2*i -1, 1)*(offsetrotPath_3(2*i, 3)-
offsetrotPath_3(2*i -1, 3))))/(offsetrotPath_3(2*i, 1)-
offsetrotPath_3(2*i -1, 1));
end

orPathPoi nts_3(1, 1)=offsetrotPath_3(1, 1);
orPathPoi nts_3(1, 2)=rotPath_3(1, 2);
orPathPoi nts_3(1, 3)=offsetrotPath_3(1, 3);
for i=2:part
    orPathPoi nts_3(i, 1)=(-offsetb_3(i -
1)+offsetb_3(i))/(offsetm_3(i -1)-offsetm_3(i));
orPathPoi nts_3(i, 3)=offsetm_3(i)*orPathPoi nts_3(i, 1)+offsetb_3(i)
;
    orPathPoi nts_3(i, 2)=rotPath_3(i, 2);

```



```

end
orPathPoi nts_3(part+1, 1)=offsetrotPath_3(2*part, 1);
orPathPoi nts_3(part+1, 2)=rotPath_3(part+1, 2);
orPathPoi nts_3(part+1, 3)=offsetrotPath_3(2*part, 3);

                                                                                               %stage-1
for i=1: part
    for j=1: part
        l1=((orPathPoi nts_3(j, 1)-
orPathPoi nts_3(i, 1))^2+(orPathPoi nts_3(j, 3)-
orPathPoi nts_3(i, 3))^2)^0.5;
        l2=((orPathPoi nts_3(j+1, 1)-
orPathPoi nts_3(i, 1))^2+(orPathPoi nts_3(j+1, 3)-
orPathPoi nts_3(i, 3))^2)^0.5;
        if (l2>=r1 & l1<r1)
            mw_3(j)=(orPathPoi nts_3(j+1, 3)-
orPathPoi nts_3(j, 3))/(orPathPoi nts_3(j+1, 1)-orPathPoi nts_3(j, 1));
            bw_3(j)=((orPathPoi nts_3(j, 3)*(orPathPoi nts_3(j+1, 1)-
orPathPoi nts_3(j, 1)))-
(orPathPoi nts_3(j, 1)*(orPathPoi nts_3(j+1, 3)-
orPathPoi nts_3(j, 3))))/(orPathPoi nts_3(j+1, 1)-
orPathPoi nts_3(j, 1));

            a=orPathPoi nts_3(j, 1);                               %bi secti on
            b=orPathPoi nts_3(j+1, 1);
            fa=(a-orPathPoi nts_3(i, 1))^2+(a*mw_3(j)+bw_3(j)-
orPathPoi nts_3(i, 3))^2-r1^2;
            fb=(b-orPathPoi nts_3(i, 1))^2+(b*mw_3(j)+bw_3(j)-
orPathPoi nts_3(i, 3))^2-r1^2;

            whi l e(abs(b-a)>0.00001)
                c=(a+b)/2;
                fc=(c-orPathPoi nts_3(i, 1))^2+(c*mw_3(j)+bw_3(j)-
orPathPoi nts_3(i, 3))^2-r1^2;
                if (si gn(fb)*si gn(fc)<=0)
                    a=c;
                    fa=fc;
                el se
                    b=c;
                    fb=fc;
                end
            end
            wc_3(i, 1)=c;
            wc_3(i, 3)=mw_3(j)*c+bw_3(j);
            ta=(wc_3(i, 3)-orPathPoi nts_3(i, 3))/(wc_3(i, 1)-
orPathPoi nts_3(i, 1));
            k1=r1/4;

                                                                                               %stage-1 joi nt
center
        s1c_3(i, 1)=((orPathPoi nts_3(i, 1)+wc_3(i, 1))/2)-
        (si gn(atan(ta))*k1*si n(abs(atan(ta))));

        s1c_3(i, 3)=((orPathPoi nts_3(i, 3)+wc_3(i, 3))/2)+k1*cos(abs(atan(ta)
        ));
        s1c_3(i, 2)=rotPath_3(i, 2);
        el se end
    end
end

                                                                                               %stage-2
k=length(s1c_3)
for i=1: k-1
    for j=1: k-1
        l1=((s1c_3(j, 1)-s1c_3(i, 1))^2+(s1c_3(j, 3)-
s1c_3(i, 3))^2)^0.5;
        l2=((s1c_3(j+1, 1)-s1c_3(i, 1))^2+(s1c_3(j+1, 3)-
s1c_3(i, 3))^2)^0.5;
        if (l2>=r2 & l1<r2)

```

```

mw_3(j)=(s1c_3(j+1,3)-s1c_3(j,3))/(s1c_3(j+1,1)-
s1c_3(j,1));
bw_3(j)=((s1c_3(j,3)*(s1c_3(j+1,1)-s1c_3(j,1)))-
(s1c_3(j,1)*(s1c_3(j+1,3)-s1c_3(j,3))))/(s1c_3(j+1,1)-
s1c_3(j,1));

a=s1c_3(j,1); %bi secti on
b=s1c_3(j+1,1);
fa=(a-s1c_3(i,1))^2+(a*mw_3(j)+bw_3(j)-s1c_3(i,3))^2-
r2^2;
fb=(b-s1c_3(i,1))^2+(b*mw_3(j)+bw_3(j)-s1c_3(i,3))^2-
r2^2;

while(abs(b-a)>0.00001)
c=(a+b)/2;
fc=(c-s1c_3(i,1))^2+(c*mw_3(j)+bw_3(j)-
s1c_3(i,3))^2-r2^2;
if(sign(fb)*sign(fc)<=0)
a=c;
fa=fc;
else
b=c;
fb=fc;
end
end
wc_3(i,1)=c;
wc_3(i,3)=mw_3(j)*c+bw_3(j);
ta=(wc_3(i,3)-s1c_3(i,3))/(wc_3(i,1)-s1c_3(i,1));
k2=r2/4;

%stage-2 joint
center
s2c_3(i,1)=((s1c_3(i,1)+wc_3(i,1))/2)-
(sign(atan(ta))*k2*sin(abs(atan(ta)))));
s2c_3(i,3)=((s1c_3(i,3)+wc_3(i,3))/2)+k2*cos(abs(atan(ta)));
s2c_3(i,2)=rotPath_3(i,2);
else end
end
end

%stage-3
l=length(s2c_3)
for i=1:l-1
for j=1:l-1
l1=((s2c_3(j,1)-s2c_3(i,1))^2+(s2c_3(j,3)-
s2c_3(i,3))^2)^0.5;
l2=((s2c_3(j+1,1)-s2c_3(i,1))^2+(s2c_3(j+1,3)-
s2c_3(i,3))^2)^0.5;
if(l2>=r3 & l1<r3)
mw_3(j)=(s2c_3(j+1,3)-s2c_3(j,3))/(s2c_3(j+1,1)-
s2c_3(j,1));
bw_3(j)=((s2c_3(j,3)*(s2c_3(j+1,1)-s2c_3(j,1)))-
(s2c_3(j,1)*(s2c_3(j+1,3)-s2c_3(j,3))))/(s2c_3(j+1,1)-
s2c_3(j,1));

a=s2c_3(j,1); %bi secti on
b=s2c_3(j+1,1);
fa=(a-s2c_3(i,1))^2+(a*mw_3(j)+bw_3(j)-s2c_3(i,3))^2-
r3^2;
fb=(b-s2c_3(i,1))^2+(b*mw_3(j)+bw_3(j)-s2c_3(i,3))^2-
r3^2;

while(abs(b-a)>0.00001)
c=(a+b)/2;
fc=(c-s2c_3(i,1))^2+(c*mw_3(j)+bw_3(j)-
s2c_3(i,3))^2-r3^2;
if(sign(fb)*sign(fc)<=0)
a=c;
fa=fc;
else

```

```

        b=c;
        fb=fc;
    end
end
    wc_3(i, 1)=c;
    wc_3(i, 3)=mw_3(j)*c+bw_3(j);
    ta=(wc_3(i, 3)-s2c_3(i, 3))/(wc_3(i, 1)-s2c_3(i, 1));
    k3=r3/4;
%stage-3 joint
center
    s3c_3(i, 1)=((s2c_3(i, 1)+wc_3(i, 1))/2)-
    (sin(atan(ta))*k3*sin(abs(atan(ta))));
s3c_3(i, 3)=((s2c_3(i, 3)+wc_3(i, 3))/2)+k3*cos(abs(atan(ta)));
    s3c_3(i, 2)=rotPath_3(i, 2);
else end
end
end

%m=length(s3c)
%for i=1:m %re-rotation of
points
%    rot_s3c=reRot*[s3c(i, 1); s3c(i, 2); s3c(i, 3)];
%    vj(i, 1)=rot_s3c(1, 1);
%    vj(i, 2)=rot_s3c(2, 1);
%    vj(i, 3)=rot_s3c(3, 1);
%end

%xlswrite('ran_data.xls', s3c_3, 4)

for j=1:length(s3c_3)
    p=0;
    for i=1:length(rotPoints)
        d=(((s3c_3(j, 1)-rotPoints(i, 1))^2)+((s3c_3(j, 2)-
rotPoints(i, 2))^2));
        if d<=(R^2)
            p=p+1;
            tmp_3(p, 1)=rotPoints(i, 1);
            tmp_3(p, 2)=rotPoints(i, 2);
            tmp_3(p, 3)=rotPoints(i, 3);
            d2(p)=d;
        else end
    end
    z1=0; z2=0;
    for q=1:p
        z1=z1+(tmp_3(q, 3)/d2(q));
        z2=z2+(1.0/d2(q));
    end
    s3cPath_3(j, 1)=s3c_3(j, 1);
    s3cPath_3(j, 2)=s3c_3(j, 2);
    s3cPath_3(j, 3)=(z1/z2)+r0+k1+k2+k3;
end

%xlswrite('ran_data.xls', s3cPath_3, 7)

figure(1)
hold all
grid on
%c=1:length(s3c_3);
plot3(s3c_3(c, 1), s3c_3(c, 2), s3c_3(c, 3))
%c=1:length(s3cPath_3);
plot3(s3cPath_3(c, 1), s3cPath_3(c, 2), s3cPath_3(c, 3))
c=1:length(rotPath_3);
plot3(rotPath_3(c, 1), rotPath_3(c, 2), rotPath_3(c, 3))
axis([20 90 2 18 0 10])

%figure(4)
%hold all
%grid on

```

```

%c=1:length(rotPath_3);    plot(rotPath_3(c,1),rotPath_3(c,3))
%c=1:length(s1c_3);       plot(s1c_3(c,1),s1c_3(c,3))
%c=1:length(s2c_3);       plot(s2c_3(c,1),s2c_3(c,3))
%c=1:length(s3c_3);       plot(s3c_3(c,1),s3c_3(c,3))
%c=1:length(s3cPath_3);   plot(s3cPath_3(c,1),s3cPath_3(c,3))
%axis([20 75 0 4])
%Fi ni sh-----3. Chai n Cal cul ati on

figure(7)
hold all
grid on
c=1:length(Path_1);       plot3(Path_1(c,1),Path_1(c,2),Path_1(c,3))
c=1:length(Path_2);       plot3(Path_2(c,1),Path_2(c,2),Path_2(c,3))
c=1:length(Path_3);       plot3(Path_3(c,1),Path_3(c,2),Path_3(c,3))
e=1:length(Poi nts);
plot3(Poi nts(e,1),Poi nts(e,2),Poi nts(e,3),'gx')
axis([10 70 10 60 0 10])

%Start-----Stabi l i ty Cal cul ati on
temp100=length(s3c_1);
temp101=length(s3c_2);
temp102=length(s3c_3);
z=min(temp100,temp101);
zz=min(z,temp102);

for i=1:zz
    for j=1:3
        aa(i,j)=s3c_1(i,j)-s3c_2(i,j);
        aaa(i,j)=orPathPoi nts_1(i,j)-orPathPoi nts_2(i,j);
        ss(i,j)=s2c_1(i,j)-s2c_2(i,j);
        gg(i,j)=s1c_1(i,j)-s1c_2(i,j);
    end
end
for i=1:zz
    for j=1:3
        bb(i,j)=s3c_3(i,j)-s3c_2(i,j);
        bbb(i,j)=orPathPoi nts_3(i,j)-orPathPoi nts_2(i,j);
        sss(i,j)=s2c_3(i,j)-s2c_2(i,j);
        ggg(i,j)=s1c_3(i,j)-s1c_2(i,j);
    end
end
for i=1:zz
    cc(i,1)=aa(i,2)*bb(i,3)-aa(i,3)*bb(i,2);
    cc(i,2)=aa(i,3)*bb(i,1)-aa(i,1)*bb(i,3);
    cc(i,3)=aa(i,1)*bb(i,2)-aa(i,2)*bb(i,1);
    lc(i)=((cc(i,1))^2+(cc(i,2))^2+(cc(i,3))^2)^0.5;

    ccc(i,1)=aaa(i,2)*bbb(i,3)-aaa(i,3)*bbb(i,2);
    ccc(i,2)=aaa(i,3)*bbb(i,1)-aaa(i,1)*bbb(i,3);
    ccc(i,3)=aaa(i,1)*bbb(i,2)-aaa(i,2)*bbb(i,1);
    lcc(i)=((ccc(i,1))^2+(ccc(i,2))^2+(ccc(i,3))^2)^0.5;

    tt(i,1)=ss(i,2)*sss(i,3)-ss(i,3)*sss(i,2);
    tt(i,2)=ss(i,3)*sss(i,1)-ss(i,1)*sss(i,3);
    tt(i,3)=ss(i,1)*sss(i,2)-ss(i,2)*sss(i,1);
    lt(i)=((tt(i,1))^2+(tt(i,2))^2+(tt(i,3))^2)^0.5;

    hh(i,1)=gg(i,2)*ggg(i,3)-gg(i,3)*ggg(i,2);
    hh(i,2)=gg(i,3)*ggg(i,1)-gg(i,1)*ggg(i,3);
    hh(i,3)=gg(i,1)*ggg(i,2)-gg(i,2)*ggg(i,1);
    lh(i)=((hh(i,1))^2+(hh(i,2))^2+(hh(i,3))^2)^0.5;

    for j=1:3
        dd(i,j)=cc(i,j)/lc(i);
        ddd(i,j)=ccc(i,j)/lcc(i);
        uu(i,j)=tt(i,j)/lt(i);
        jj(i,j)=hh(i,j)/lh(i);
    end

    s_alfa(i)=atan(dd(i,2)/dd(i,3))*180/3.14;

```

```

n_al fa(i)=atan(ddd(i, 2)/ddd(i, 3))*180/3. 14;
s_2_al fa(i)=atan(uu(i, 2)/uu(i, 3))*180/3. 14;
s_1_al fa(i)=atan(jj (i, 2)/jj (i, 3))*180/3. 14;

s_beta(i)=atan(dd(i, 1)/dd(i, 3))*180/3. 14;
n_beta(i)=atan(ddd(i, 1)/ddd(i, 3))*180/3. 14;
s_2_beta(i)=atan(uu(i, 1)/uu(i, 3))*180/3. 14;
s_1_beta(i)=atan(jj (i, 1)/jj (i, 3))*180/3. 14;

s_gama(i)=asin((dd(i, 2)^2+dd(i, 1)^2)^0. 5)*180/3. 14;
n_gama(i)=asin((ddd(i, 2)^2+ddd(i, 1)^2)^0. 5)*180/3. 14;
s_2_gama(i)=asin((uu(i, 2)^2+uu(i, 1)^2)^0. 5)*180/3. 14;
s_1_gama(i)=asin((jj (i, 2)^2+jj (i, 1)^2)^0. 5)*180/3. 14;

al fa_c(i)=(90-s_al fa(i))*3. 14/180;
beta_c(i)=(90-s_beta(i))*3. 14/180;

```

```

gama_c(i)=atan(((tan(al fa_c(i))^2+tan(beta_c(i))^2)^0. 5)/(tan(al f
a_c(i))*tan(beta_c(i))))*180/3. 14
end

```

```

figure(5) %al fa
hold all
grid on
c=1: zz; plot(c, s_al fa(c)-5)
c=1: zz; plot(c, n_al fa(c)-5)
c=1: zz; plot(c, s_2_al fa(c)-5)
c=1: zz; plot(c, s_1_al fa(c)-5)
axis([0 zz -3 3])

```

```

figure(6) %beta
hold all
grid on
c=1: zz; plot(c, s_beta(c))
c=1: zz; plot(c, n_beta(c))
c=1: zz; plot(c, s_2_beta(c))
c=1: zz; plot(c, s_1_beta(c))
axis([0 zz -3 3])

```

```

figure(8) %gama
hold all
grid on
c=1: zz; plot(c, n_gama(c)-5)
c=1: zz; plot(c, s_1_gama(c)-5)
c=1: zz; plot(c, s_2_gama(c)-5)
c=1: zz; plot(c, s_gama(c)-5)
axis([0 zz -3 3])

```

```

figure(9) %gama control
hold all
grid on
c=1: zz; plot(c, gama_c(c))
c=1: zz; plot(c, s_gama(c))
%axis([0 zz -3 3])

```

%Fi ni sh-----Stabi l i ty Cal cul ati on

```

%m=length(s3c_1)
%for i=1: m %re-rotation of
poi nts
% rot_s3c=reRot*[s3c_1(i, 1); s3c_1(i, 2); s3c_1(i, 3)];
% vj 1(i, 1)=rot_s3c(1, 1);
% vj 1(i, 2)=rot_s3c(2, 1);
% vj 1(i, 3)=rot_s3c(3, 1);
%end

```

```

%m=length(s3c_2)
%for i=1: m %re-rotation of
poi nts
% rot_s3c=reRot*[s3c_2(i, 1); s3c_2(i, 2); s3c_2(i, 3)];

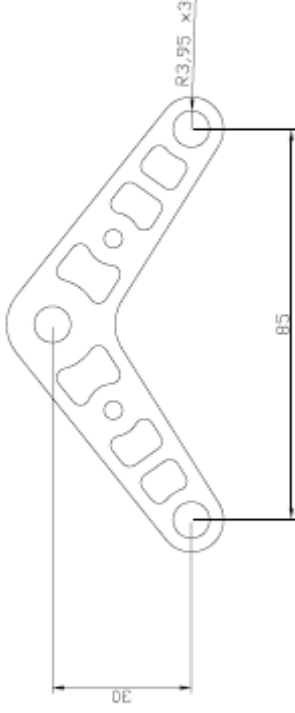
```

```
% vj 2(i, 1)=rot_s3c(1, 1);  
% vj 2(i, 2)=rot_s3c(2, 1);  
% vj 2(i, 3)=rot_s3c(3, 1);  
%end
```

```
clear
```


EK-2 İMALAT RESİMLERİ

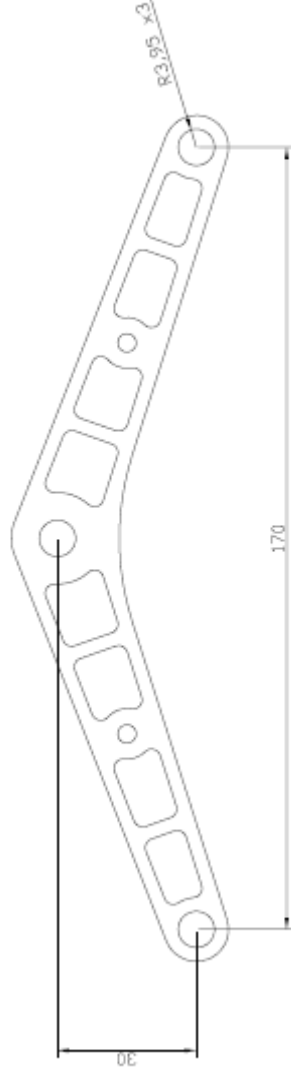
Libra'nın prototip üretimi için çıkarılmış imalat resimleri aşağıda görülebilir.



Notlar:


- 1) Parça kalınlığı 4,00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 24 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü almayınız.

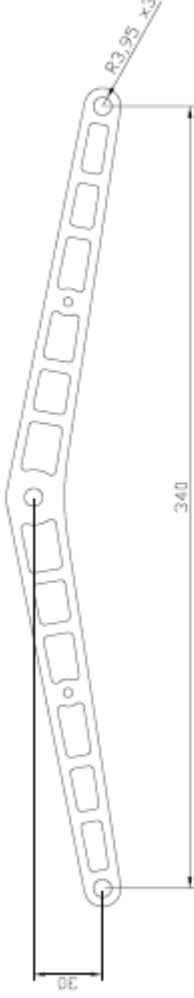
Çizen / Tarih	Resim No	LIBRA-P01-İKS-00	
Alper Yontar / 05.12.09	Parça Adı	1. Kademe Salmcak	
Onay	Malzeme	E.Ü. FBE. Makine Müh. ABD. Lisansüstü Tezi	Ölçek
Özgür KİLİT	St37 Saç		1:1 (A4)



Notlar:


- 1) Parça kalınlığı 4.00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 12 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü almayınız.

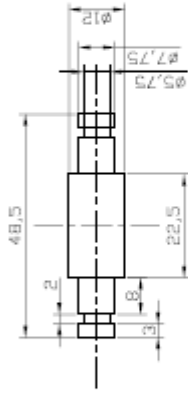
Çizen / Tarih	Resim No	LIBRA-P01-2KS-00	
Alper Yontar / 05.12.09	Parça Adı	2. Kademe Salıncak	
Onay	Malzeme	E.Ü. FBE. Makine Müh. ABD. Lisansüstü Tezi	Ölçek 1:1 (A4)
Özgür KİLİT	S137 Saç		



Notlar:


- 1) Parça kalınlığı 4.00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 6 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü almayınız.

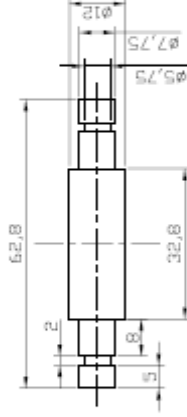
Çizen / Tarih Alper Yontar / 05.12.09	Resim No	LIBRA-P01-3KS-00	
	Parça Adı	3. Kademe Salıncak	
Onay	Malzeme	E.Ü. FBE, Makine Müh. ABD, Lisansüstü Tezi	Ölçek 1:2 (A4)
Özgür KİLİT	St37 Saç		



Notlar:


- 1) 12 adet üretilecek.
- 2) Resim üzerinden ölçü almayınız.

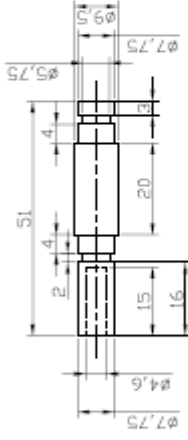
Çizen / Tarih	Resim No	LIBRA-P01-KM1-00	
Alper Yontar / 05.12.09	Parça Adı	1.-2. Kademe Mili	
Onay	Malzeme	E.Ü. FBE. Makine Müh. ABD. Lisansüstü Tezi	Ölçek
Özgür KİLİT	St37		1:1 (A4)



Notlar:

- 1) 6 adet üretilecek.
- 2) Resim üzerinden ölçü almayınız.

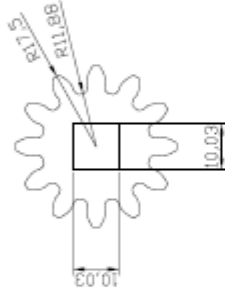
Çizen / Tarih	Resim No	LIBRA-P01-KM2-00	
Alper Yontar / 05.12.09	Parça Adı	2.-3. Kademe Mili	
Onay	Malzeme	E.Ü. FBE. Makine Müh. ABD. Lisansüstü Tezi	Ölçek 1:1 (A4)
Özgür KİLİT	St37		



Notlar:


- 1) 24 adet üretilecek.
- 2) Resim üzerinden ölçü almayınız.

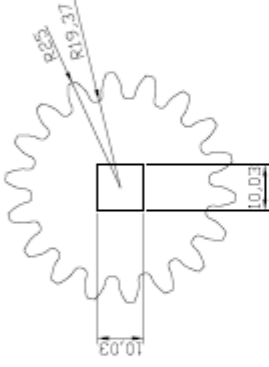
Çizen / Tarih	Resim No	LIBRA-P01-TM-00	
Alper Yontar / 05.12.09	Parça Adı	Tekerlek Mili	
Onay	Malzeme	E.Ü. FBE. Makine Müh. ABD. Lisansüstü Tezi	Ölçek
Özgür KİLİT	S137		1:1 (A4)



Notlar:


- 1) Parça kalınlığı 4.00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 24 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü alınmayız.

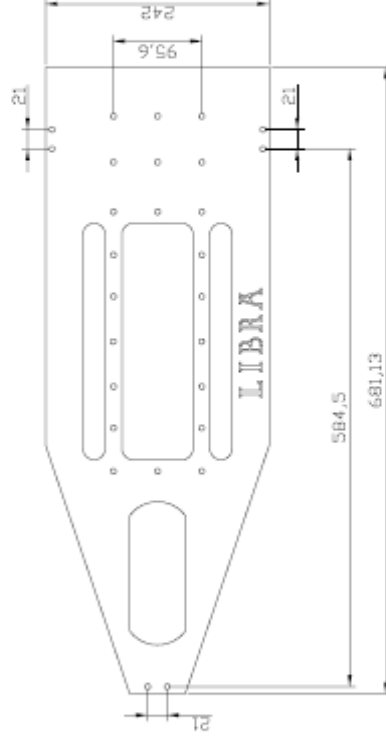
Çizen / Tarih Alper Yontar / 05.12.09	Resim No LIBRA-P01-MD-00	Motor Dişlisi		
		Parça Adı	Malzeme	
Onay		St37 Saç	E.Ü. FBE, Makine Müh. ABD, Lisansüstü Tezi	
Özgür KİLİT				



Notlar:

- 1) Parça kalınlığı 4.00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 24 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü almayınız.

Çizen / Tarih Alper Yontar / 05.12.09	Resim No	LIBRA-P01-TD-00		
	Parça Adı	Tahrik Dişlisi		
Onay	Malzeme	E.Ü. FBE, Makine Müh. ABD, Lisansüstü Tezi		Ölçek 1:1 (A4)
Özgür KİLT	St37 Saç			



Notlar:

- 1) Parça kalınlığı 4,00 mm'dir.
- 2) Parça lazer kesim ile CAD resmine göre 1 adet kesilecektir.
- 3) Resimde kritik ölçüler gösterilmiştir. Resim üzerinden ölçü almamız.

Çizen / Tarih	Resim No	LIBRA-P01-AG-00	
Alper Yontar / 05.12.09	Parça Adı	Araç Gövdesi	
Onay	Malzeme	E.Ü. FBE, Makine Müh. ABD. Lisansüstü Tezi	Ölçek 1:5 (A4)
Özgür KİLİT	St37 Saç		

ÖZGEÇMİŞ

Alper Yontar 1983 yılında İzmir’de doğmuştur. İzmir Atatürk Lisesi’ni bitirdiği 2001 yılında Ege Üniversitesi Mühendislik Fakültesi Makine Mühendisliği Bölümü’nü kazanmıştır. 2008 yılının Ocak ayında Makine Mühendisliği’ni tamamlamış ve aynı yılın Şubat ayında Ege Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı’nda yüksek lisansına başlamıştır. Yüksek lisans çalışmaları sırasında bu tezin de konusunu oluşturan araziye uyum sağlayabilen çok kademeli salınacak mekanizmalı bir aracın kararlılığının incelendiği bir bildiriye Danışmanı Öğr. Gör. Dr. Özgür Kilit ile İstanbul’da uluslararası bir konferans olan “Recent Advances in Space Technologies (RAST) 2009” konferansında sunmuştur.

Bildirinin tam tanımı: Kilit, Ö., Yontar, A., 2009, Stability of a New Mars Rover with Multi-stage Bogie Mechanism, Proc. of 4nd International Conference on "Recent Advances in Space Technologies", İstanbul, Turkey, s145-148.