

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**PARALEL MEKANİZMALARIN ÇALIŞMA UZAYI ANALİZİNDE  
KULLANILAN YAPAY SİNİR AĞI YAPISININ GENETİK  
ALGORİTMALAR İLE BELİRLENMESİ**

**YÜKSEK LİSANS TEZİ  
Zeynep EKİCİOĞLU**

**Anabilim Dalı : Disiplinlerarası Anabilim Dalı**

**Programı : Mekatronik Mühendisliği**

**HAZİRAN 2011**





**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**PARALEL MEKANİZMALARIN ÇALIŞMA UZAYI ANALİZİNDE  
KULLANILAN YAPAY SİNİR AĞI YAPISININ GENETİK  
ALGORİTMALAR İLE BELİRLENMESİ**

**YÜKSEK LİSANS TEZİ  
Zeynep EKİCİOĞLU  
(518081033)**

**Tezin Enstitüye Verildiği Tarih : 06 Mayıs 2011  
Tezin Savunulduğu Tarih : 14 Haziran 2011**

**Tez Danışmanı : Prof. Dr. İbrahim ÖZKOL (İTÜ)  
Eş Danışman : Yard. Doç. Dr. Vasfi E. ÖMÜRLÜ  
(YTÜ)  
Diğer Jüri Üyeleri : Prof. Dr. Metin O. KAYA (İTÜ)  
Doç. Dr. Erol UZAL (İÜ)  
Yard. Doç. Dr. Hacı İ. KESER (İTÜ)**

**HAZİRAN 2011**

## ÖNSÖZ

Paralel mekanizmalar uç organı ile tabanın birden fazla bağımsız bağ ile bağlandığı kapalı çevrim yapılarıdır. Hassas konumlandırma, yüksek yük taşıma kapasiteleri gibi avantajlarından dolayı son yıllarda paralel mekanizmalar üzerine oldukça fazla çalışma yapılmıştır. Stewart Platformu yapıları da 6 derbestlik dereceli özelleşmiş paralel mekanizmalardır. Bu çalışmada SPM yapılarının düz ve ters kinematik analizleri yapılmıştır. Ayrıca 6x3 SPM yapısının çalışma uzayının analizini yüksek doğrulukta yapabilecek YSA yapıları belirlenmiştir. Uygulanan bu belirleme yönteminde Genetik Algoritmalar kullanılmış ve yazılan programın paralel mekanizmaların çalışma uzayı analizleri için geliştirilmesi amaçlanmıştır. Bu doğrultuda 6x6 SPM'nin deneysel verileri üzerinden de programın doğruluğu test edilmiştir. Alınan sonuçlar değerlendirildiğinde yüksek doğruluk oranı ile istenilen hedefe ulaşılmıştır.

Bu çalışmayı yaptığım süre boyunca benden yardımını esirgemeyen danışmanım Prof. Dr. İbrahim Özkol'a, eş danışmanım Yard. Doç. Dr. Vasfi Emre Ömürlü'ye, yine yardımlarından dolayı Dr. Hüseyin Alp'e ve Dr. Utku Büyüksahin'e, aynı zamanda her zaman yanımda olan maddi, manevi desteklerini hep hissettiğim sevgili aileme teşekkürü bir borç bilirim.

Mayıs 2011

Zeynep Ekicioğlu



## İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	v
KISALTMALAR .....	vii
ÇİZELGE LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	xi
ÖZET.....	xiii
SUMMARY.....	xv
<b>1. GİRİŞ .....</b>	<b>1</b>
1.1 Paralel Mekanizmalar .....	1
1.2 Paralel Mekanizmaların Tarihi Gelişimi .....	2
1.3 Stewart Platformu.....	6
1.3.1 Stewart platformu ile ilgili yapılan çalışmalar .....	9
<b>2. KİNEMATİK ANALİZ .....</b>	<b>15</b>
2.1 Ters Kinematik Analiz.....	15
2.1.1 3x3 Stewart platformunun ters kinematik analizi .....	15
2.2 Düz Kinematik Analiz .....	17
2.3 3x3 Stewart platformunun düz kinematik çözümünde Bezout metodunun kullanılması.....	18
<b>3. YAPAY SİNİR AĞLARI.....</b>	<b>25</b>
3.1 YSA'nın Tanımı ve Literatür İncelemesi .....	25
3.2 Biyolojik Sinir Hücreleri ve Yapay Sinir Modeli.....	26
3.3 YSA'nın Çalışma Sistemi ve Bileşenleri.....	28
3.3.1 Yapay sinir ağı yapısının çeşitleri .....	29
3.3.1.1 Tek katmanlı ileri beslemeli yapay sinir ağı yapısı .....	29
3.3.1.2 Çok katmanlı ileri beslemeli yapay sinir ağı yapısı.....	29
3.3.1.3 Geri beslemeli yapay sinir ağı yapısı.....	30
3.3.2 Yapay sinir ağı transfer fonksiyonları .....	30
3.3.3 Yapay sinir ağı eğitme algoritmaları ve öğrenme kuralları .....	31
<b>4. GENETİK ALGORİTMALAR.....</b>	<b>35</b>
4.1 Genetik Algoritmalar Tanımı.....	35
4.2 Genetik Algoritmaların Literatür İncelemesi.....	36
4.3 Genetik Algoritmada Kodlama .....	38
4.4 Genetik Operatörler .....	38
4.4.1 Seçim operatörü.....	39
4.4.2 Çaprazlama (cross-over) operatörü .....	40
4.4.3 Mutasyon operatörü .....	41
<b>5. GENETİK ALGORİTMALAR İLE YAPAY SİNİR AĞLARININ BİRLİKTE KULLANIMI .....</b>	<b>43</b>
5.1 GA ve YSA'nın Birlikte Kullanımının Avantajları .....	44

5.2 GA ile YSA'nın Birlikte Kullanıldığı Çalışmalar .....	45
<b>6. SİMÜLASYONLAR ve SONUÇLARI .....</b>	<b>47</b>
6.1 Genetik Algoritma Uygulaması Örneği Olarak Gezgin Satıcı Probleminin Çözümlemesi .....	48
6.1.1 Problemin tanımı.....	48
6.1.2 Bireylerin kodlanması ve kullanılan yöntem.....	48
6.1.3 Gezgin satıcı probleminde genetik algoritma uygulaması .....	50
6.1.4 GSP sonuçları .....	52
6.2 6x3 SPM'nin Çalışma Uzayı Analizi için Kullanılacak YSA Yapısının Genetik Algoritmalar ile Belirlenmesi.....	53
6.2.1 Kromozom yapısı ve kodlanması.....	54
6.2.2 Nesil büyüklüğünün belirlenmesi .....	55
6.2.3 Uygunluk değerinin hesaplanması .....	56
6.2.4 Seçim yöntemi .....	57
6.2.5 Yeni nesil üretimi.....	57
6.2.6 YSA destekli genetik algoritmanın çalışma mantığı.....	57
6.2.7 YSA destekli GA programı kullanılarak 6x3 Stewart Platformu üzerine yapılan uygulama sonuçları.....	59
6.3 Deneysel 6x6 SPM'nin Çalışma Uzayı Analizi için Kullanılacak YSA Yapısının Genetik Algoritmalar ile Belirlenmesi.....	63
6.3.1 YSA destekli GA programı kullanılarak deneysel 6x6 Stewart Platformu dataları üzerine yapılan uygulama sonuçları .....	65
<b>7. SONUÇLAR VE ÖNERİLER.....</b>	<b>67</b>
<b>KAYNAKLAR.....</b>	<b>69</b>
<b>EKLER .....</b>	<b>77</b>



## **KISALTMALAR**

<b>GA</b>	: Genetik Algoritmalar
<b>SPM</b>	: Stewart Platform Mekanizması
<b>YSA</b>	: Yapay Sinir Ağları
<b>UPS</b>	: Üniuersal-Prizmatik-Küresel Mafsallı



## ÇİZELGE LİSTESİ

### Sayfa

Çizelge 5.1 : Gizli katmandaki sınır sayısını belirleme metodları. ....	45
Çizelge 6.1 : Oluşturulan ağ yapılarının başarı yüzdeleri.....	61
Çizelge 6.2 : Program ile oluşturulan ağ yapılarından örnekler ve başarı oranları ....	65



## ŞEKİL LİSTESİ

### Sayfa

Şekil 1.1 : 1931 yılında patenti alınmış uzaysal paralel mekanizma.....	2
Şekil 1.2 : 1942 yılında patenti alınmış ilk endüstriyel paralel robot.....	3
Şekil 1.3 : Orjinal Gough Platformu (a) ve 2000 yılına kadar kullanılmış bir lastik testi cihazı (b).....	3
Şekil 1.4 : Mühendis Cappel tarafından 1960'larda geliştirilen simülatör.....	4
Şekil 1.5 : 3 serbestlik dereceli düzlemsel paralel mekanizma örneği.....	5
Şekil 1.6 : Üçgen şeklindeki hareketli platformlu 3 serbestlik dereceli düzlemsel paralel mekanizma.....	5
Şekil 1.7 : Lufthansa havayollarına ait bir simülatör (solda) ve pitch hareketini yaparken bir simülatör görüntüsü.....	6
Şekil 1.8 : Stewart'ın tasarladığı paralel mekanizma yapısı.....	7
Şekil 1.9 : 1965 yılında Stewart tarafından yayınlanan makaledeki uçuş simülatörü olarak tasarlanan SPM resmi.....	8
Şekil 1.10 : 6-SPS Stewart platformu (a) ve 6-UPS Stewart platformu (b).....	8
Şekil 1.11 : Klasik (a) ve (b) modifiye edilmiş (MSP) Stewart Platformu yapısı.....	9
Şekil 1.12 : Esnek paralel mekanizma örneği (a) ve Minyatür Hexapod resmi (b)....	10
Şekil 1.13 : a- 6 eksenli SPM tabanlı kuvvet/tork sensörü.b- Ön gerilimli SPM tabanlı sensörün prototip resmi.....	11
Şekil 1.14 : Paralel mekanizmaların haptik sistemlerde kullanımı.....	12
Şekil 1.15 : Rehabilitasyon için kullanılan platform.....	12
Şekil 1.16 : Forklift eğitim simülatörü.....	13
Şekil 2.1 : Kinematik analizi yapılan 3x3 lük Stewart Platformunun temsili gösterimi.....	16
Şekil 2.2 : Düz kinematik analiz için kabul edilen geometrik model.....	18
Şekil 2.3 : Kabul edilen geometrik yapının açık hali.....	19
Şekil 3.1: Biyolojik sinir ağı yapısı.....	27
Şekil 3.2: Yapay sinir ağı modeli.....	27
Şekil 3.3: Tek katmanlı ileri beslemeli(a) ve ileri beslemeli çok katmanlı ağ(b).....	29
Şekil 3.4: Doğrusal (a) ve Basmak Eşik Fonksiyonu (b).....	30
Şekil 3.5: Sigmoid (a) ve Tanjant Hiperbolik Fonksiyonu(b).....	31
Şekil 4.1 : Tek noktalı çaprazlama.....	40
Şekil 4.2 : Tamsayı kodlama için mutasyon örneği.....	41
Şekil 4.3 : İkili kodlama için mutasyon örneği.....	41
Şekil 6.1 : Bireylerin oluşturulma mantığı (kaynak [91]'dan uyarlanmıştır).....	49
Şekil 6.2 : GSP çözümünde kullanılan GA programının çalışma şekli.....	51
Şekil 6.3 : (a)Rastgele hesaplanan tur ve (b) Optimizasyonu yapılmış tur örneği.....	53
Şekil 6.4 : Kromozomların kodlanma şekli.....	55
Şekil 6.5 : Yazılan programın blok şeması.....	56
Şekil 6.6 : GA programının akış şeması.....	58
Şekil 6.7 : Çalışma uzayı içindeki üst platformun konumları.....	59
Şekil 6.8 : Çalışma uzayı dışındaki üst platform konumları.....	60

<b>Şekil 6.9</b> : Çalışma uzayı içindeki ve dışındaki dataların birlikte gösterimi .....	61
<b>Şekil 6.10</b> : 3432'lik yapıyı veren kromozomun ait ağırlık ve bias değerleri. ....	62
<b>Şekil 6.11</b> : Deneysel sitemin çalışma uzayı içerisindeki dataları .....	63
<b>Şekil 6.12</b> : Deneysel sitemin çalışma uzayı dışındaki dataları.....	64

# PARALEL MEKANİZMALARIN ÇALIŞMA UZAYI ANALİZİNİNDE KULLANILAN YAPAY SİNİR AĞI YAPISININ GENETİK ALGORİTMA İLE BELİRLENMESİ

## ÖZET

Yüksek yük taşıma kapasitesi hassas konumlandırma gibi özelliklere sahip paralel mekanizmalar son yıllarda birçok alanda kullanılmaktadırlar. Stewart Platform Mekanizması (SPM) da 6 serbestlik dereceli ve üzerine birçok araştırma yapılmış bir paralel mekanizmadır.

Bu çalışmada paralel mekanizmalar ve Stewart Platformları üzerine yapılmış araştırmalar incelenmiş ve literatür taranmıştır. Ardından 3x3'lük bir SPM'nin ters ve düz kinematik analizleri yapılmıştır.

Paralel mekanizmaların en büyük dezavantajlarından birisi küçük çalışma uzayına sahip olmalarıdır. Bu tez çalışmasında, 6x3'lük SPM'nin çalışma uzayı analizini yapacak yapay sinir ağları (YSA) yapısının belirlenmesi üzerine çalışılmıştır. Analiz için kullanılacak YSA yapısı, genetik algoritma tabanlı bir program ile belirlenmekte ve verilen datalar karşısında dizayn edilen YSA yapısının uygunluğu hesaplanmaktadır. Geliştirilen programın doğruluğunun genelleştirilmesi için deneysel 6x6 Stewart Platformu'nun verileri ile de çalışılmıştır. Çıkan sonuçlar oldukça yüksek doğrulukta ve tatmin edici olmuştur.

Sonuç itibari ile YSA kullanılarak bir probleme çözüm getirilmek istendiğinde ağ yapısı ve ağırlık, bias gibi değişkenlerin değerleri ağın başarısı üzerinde çok etkili olmaktadır. Bu nedenle deneme yanılma yolu ile bulunan bu değerlerin ve uygun yapının hesabı, çok fazla zaman kaybına neden olmaktadır. Sözü edilen kaybı engellemek ve tek bir program ile hangi yapının en uygun olduğunu tespit etmek yazılan GA programı ile mümkün kılınmıştır.





## **DETERMINATION OF THE NEURAL NETWORK STRUCTURE USED IN PARALLEL MECHANISM'S WORKSPACE ANALYSES BY GENETIC ALGORITHMS**

### **SUMMARY**

Due to features such as precise positioning and high load carrying capacity, parallel mechanisms are used in many areas in recent years. Also Stewart Platform Mechanism (SPM) is a type of parallel mechanism has 6 degrees of freedom and many studies have been done about it.

In this study, literature which is about the SPM was searched and parallel mechanisms were investigated, examined. Afterwards inverse and forward kinematics analyses of 3x3 Stewart Platform Mechanism were performed.

Small workspace of SPMs are the worst disadvantages of this mechanisms. In this thesis studies, determination of the neural network structure which analyses SPM's workspace was studied. Structure of YSA which is used for analyzing of workspace is determined by a genetic algorithm based programme and according to the given data, convenience of the YSA structure can be calculated. In addition to, data of the experimental 6x6 SPM were applied to the developed programme for generalization of programme's accuracy. The results were very high precision and had been very satisfactory.

As a result if NN is used to solve any problem, the network structure of NNs and weights, biases, are very effective on the success of the network. Therefore calculation of these values and appropriate structure by trial and error, causes too much loss time. To prevent the loss time and determine the which structure is the most suitable by just using one programme, are made possible with developed GA programme.



# 1. GİRİŞ

## 1.1 Paralel Mekanizmalar

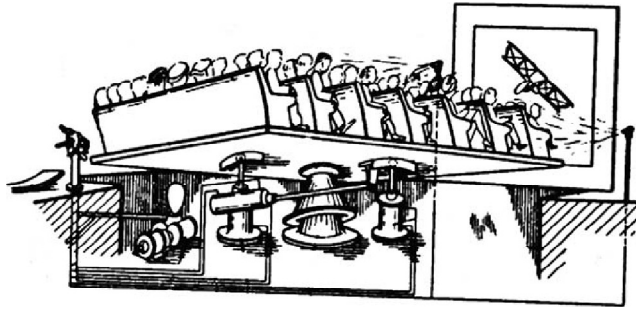
İnsan gücünün yerine mekanik sistemlerin gücünden yararlanma düşüncesi sanayide makineleşmenin ve teknolojinin gelişmesinin temelinde yatan nedendir. Bu düşünce, insanlığı kendi kolunun yapısına benzeyen seri robotların bulmasına yöneltmiştir. Seri robotların hızlı manevra yetenekleri ve geniş çalışma uzayı gibi avantajları bulunmaktadır. Ancak kol yapısı nedeniyle düşük yük taşıma kapasitesine sahiptir. Ayrıca büyük bir çalışma uzayına sahip olmasına rağmen konumlandırma hassasiyetleri oldukça düşüktür. Kısacası tıpkı insan kolunun avantajlarına ve dezavantajlarına sahiptir diyebiliriz. Bu nedenle yüksek yük taşıma kapasitesi, hassas konumlandırma ve iyi bir dinamik performans istenen yerlerde kullanılmak üzere geliştirilen alternatif mekanizmalar, paralel mekanizmalardır. Paralel mekanizmaların çalışma mantığını biyolojik dünya da görebiliriz, örneğin birden fazla paralel ayakları olan hayvanlar, iki ayağı olan insanlardan daha kararlı hareket ederler yada insanların ağır bir yükü kaldırmak için iki elini birden kullandıklarını görmekteyiz. Başka bir örnek ise yazma gibi hassas bir iş için 3 parmak paralel olarak kullanılmaktadır. Bu örneklerden anlaşıldığı gibi paralel mekanizmalar daha hassas ve yüksek kapasiteli yük taşınması gereken yerlerde kullanılmaktadır [1] .

Paralel mekanizmalar uç organı ile tabanın birden fazla bağımsız bağ ile bağlandığı kapalı çevrim yapılarıdır. Paralel mekanizmalara, paralel robotlarda denebilmektedir. Paralel robotlar sabit bir platform ve ona bacak denilen yapılarla bağlı olan uç organından (yani hareketli platform) oluşmaktadır. Bacakların platformlara bağlanmasında genelde prizmatik, küresel ve universal mafsallar kullanılmaktadır. Bu nedenle linkler sadece, pozisyon hassasiyetini arttıran, çekme veya sıkışmaları hissedebilir. Bacaklardaki motorlar prizmatik eklemler ile sabit platforma bağlanır ki böylece bacakların ağırlıkları da mekanizmaya eklenmemiş olur ve platform biraz daha hafifletilmiş olur. Hareketli platformun aynı anda birden fazla yere gidebilmesi, paralel manipülatörlerin yapısal rijitlik prensibine sahip olmalarını sağlamaktadır. Tüm bu özellikler sonucu paralel manipülatörler geniş bir hareket kabiliyetine

sahiptirler. Bu tip yapıların en kötü özelliği, sınırlı çalışma uzayı alanlarıdır, çünkü bacaklar birbiriye çarpışabilmektedir ve her bir eklem kendi mekanik kısıtlamaları vardır. Paralel robotların diğer bir dezavantajı ise tekil noktalarda yapısal kararlılıklarını kaybetmeleridir (Robot sonlu veya sonsuz serbestlik derecesi kazanır ve kontrol edilemez, yada titreşime girebilir). Kullanılan motorlar sabit platformun üzerine yüklerini bindirdikleri için ağırlıktan tasarruf edilir ve paralel robotlar diğer geleneksel mafsallı robotlara göre daha hızlı hareket edebilmektedirler. Aynı zamanda paralel robotların diğer robotlara göre daha fazla yük taşıma özelliğinin altında, hareketli platformun birden fazla bacakla desteklenmesi bulunmaktadır. Burada kısaca bahsettiğimiz paralel robotların avantaj ve dezavantajlarından ‘Seri ve Paralel Robotların Karşılaştırması’ kısmında daha fazla bahsedeceğiz.

## 1.2 Paralel Mekanizmaların Tarihi Gelişimi

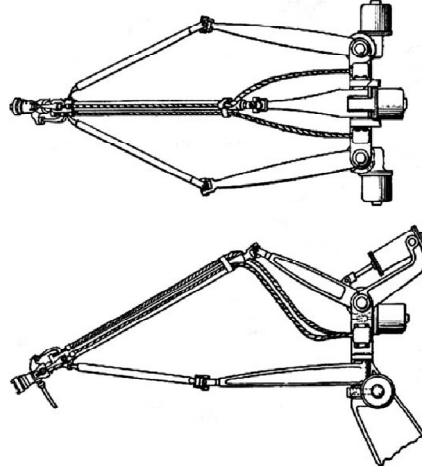
Son yıllarda endüstrinin ve akademik çalışmaların vazgeçilmez bir parçası haline gelen paralel yapıların kökeni 1800’lü yılların sonunda Maxwell (1890) tarafından yayınlanan ilk teorik makaleye dayanmaktadır [2]. Bazı kaynaklara göre ise 1928 yılında James E. Gwinnet tarafından eğlence konseptiyle yapılan Şekil 1.1’deki mekanizma, ilk uzaysal paralel mekanizma örneği olarak kabul edilmektedir [3].



Şekil 1.1 : 1931 yılında patenti alınmış uzaysal paralel mekanizma

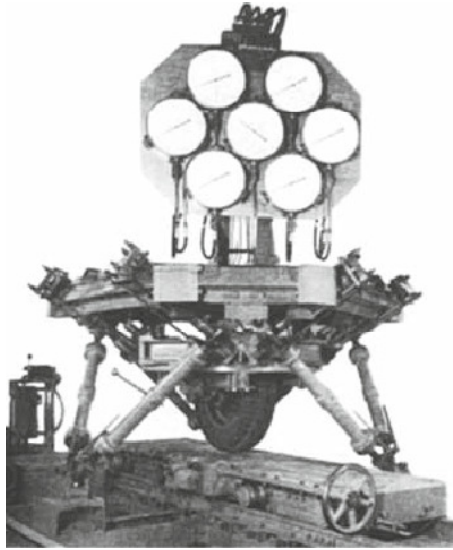
İlk paralel mekanizma çalışmalarından bir tanesi de 1942 yılında Willard L. V. Pollard tarafından patenti alınan, araba boyamak için tasarlanan ve ilk endüstriyel robot olduğu iddia edilen paralel mekanizmadır [4].

Şekil 1.2’de görülen bu mekanizma üç bacaklı olup beş serbestlik derecesine sahiptir, bunlardan üç tanesi aletin kafa pozisyonlaması için diğer iki tanesi de yönelimi içindir.



**Şekil 1.2 :** 1942 yılında patenti alınmış ilk endüstriyel paralel robot.

Daha sonra, 1962 yılında Dr. Eric Gough and Whiteall lastik testi için bir paralel manipülatör icat etmişler ve kullanmışlardır [5]. Bu sistemin ilk hali ve 2000 yılına kadar kullanılan bir lastik test cihazının resmi Şekil 1.3’de görülmektedir. Bu mekanizmanın yapılması endüstriyel ve akademik alandaki paralel mekanizma çalışmalarında büyük gelişmelere olanak sağlamıştır.



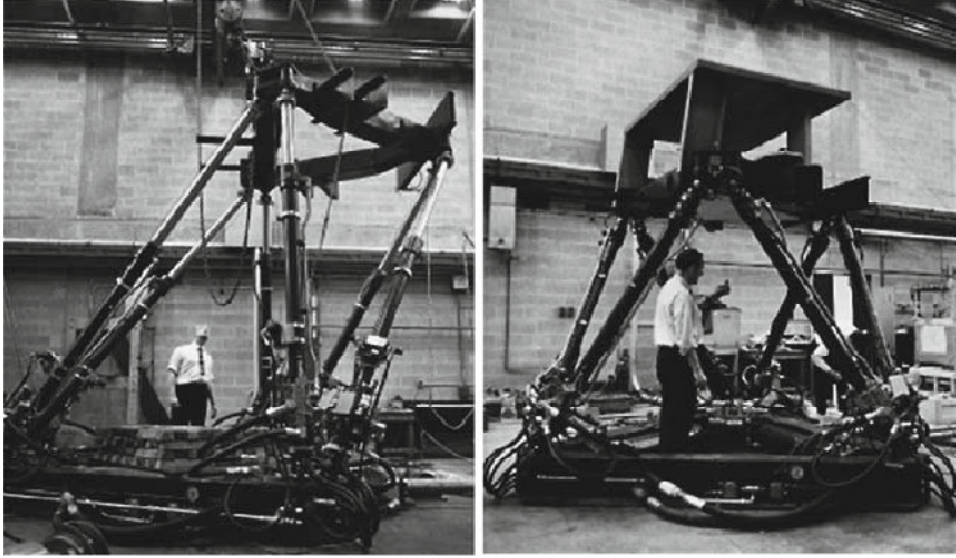
(a)



(b)

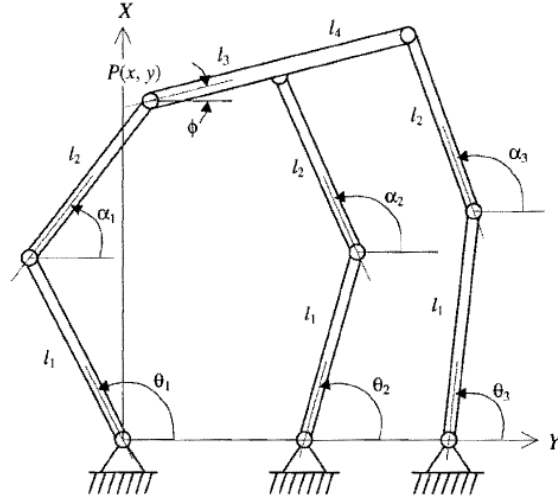
**Şekil 1.3 :** Orjinal Gough Platformu (a) ve 2000 yılına kadar kullanılmış bir lastik testi cihazı (b).

1965 yılında ise Stewart başka bir alanda, ilk defa uçuş simülatörü olarak 6 serbestlik dereceli paralel manipülatörün kullanıldığı bir makale yayınlamıştır [6]. Stewart yaptığı bu çalışma ile Gough mekanizmasının literatürdeki ününü oldukça artırmış, endüstride kullanımını yaygınlaştırmış ve akademik açıdan da bu mekanizmaların gelişimi için bir dönüm noktası teşkil etmiştir. Paralel mekanizmaların tarihinde önemli bir başka isim ise mühendis Klaus Cappel'dir. 1967 yılında hareket simülatörü olarak bir hexapod yapısı kullanmış ve yapıyı gerçekleştirerek endüstride kabulünü sağlamıştır. (bkz. Şekil 1.4)



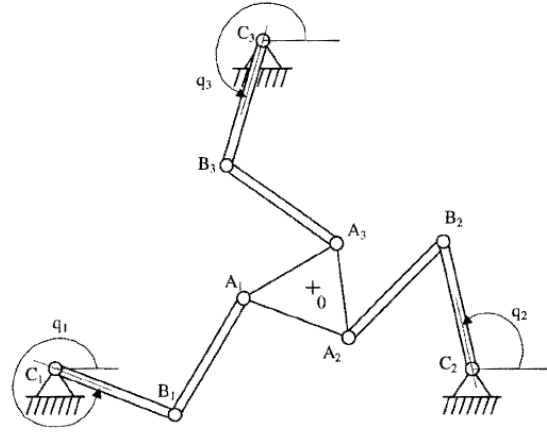
**Şekil 1.4 :** Mühendis Cappel tarafından 1960'larda geliştirilen simülatör.

Anlatılan manipülatörlerin hepsinde lineer motorlar kullanılmıştır, ancak 1983 yılında Hunt dönel motorlar ile yeni bir paralel manipülatör yapısı sunmuştur [7]. Son yıllarda da birçok araştırmacı farklı paralel kinematik yapılar önermişlerdir, bazıları yalnız lineer motorlar ile bazıları yalnız döner motorlar ile bir kısım ise lineer ve döner motorların bir arada kullanılmasıyla farklı yapıda paralel mekanizmalar oluşturmuşlardır. Paralel mekanizmalardaki bu gelişme ivmesiyle beraber araştırmacılar farklı serbestlik dereceli mekanizmalar da önermeye başlamışlardır. Üç serbestlik derecesine sahip, 3 döner eklemden oluşan şekil 1.5' de ki sistemde iki bağımsız eklem hareketli platform ile sabit platformu bağlamaktadır. Gosselin ve Wang'ın [8] geliştirdiği bu yapı, düzlemsel paralel mekanizmalar sınıfında yer almaktadır.



**Şekil 1.5 :** 3 serbestlik dereceli düzlemsel paralel mekanizma örneği.

Yapılan paralel mekanizmalardaki çalışma uzayı analizleri sırasında sistemin geometrik yapısının da analiz üzerinde etkili olabileceği düşünülmüş ve Kumar'ın [9] yaptığı gibi üst platformun şekli üçgen olarak tasarlanmıştır. Üzerine çalışılan yapı ve üst platformun geometrisi Şekil 1.6'da görülmektedir.



**Şekil 1.6 :** Üçgen şeklindeki hareketli platformlu 3 serbestlik dereceli düzlemsel paralel mekanizma.



(a)



(b)

**Şekil 1.7 :** Lufthansa havayollarına ait bir simülâtör (a) ve pitch hareketini yaparken bir simülâtör görüntüsü (b).

Yukarıdaki şekil 1.7’de ise uçuş simülâtörlerinin son halleri ve pitch hareketini yaparkenki simülâtör resmi görülmektedir.

### 1.3 Stewart Platformu

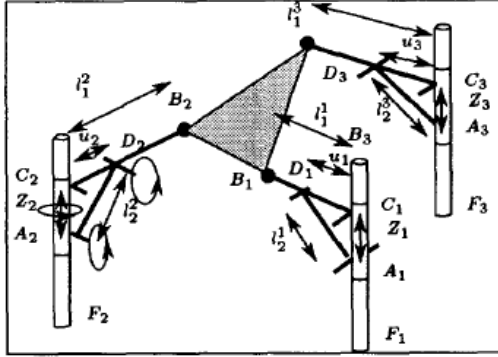
Stewart Platformu 6 serbestlik derecesine sahip bir paralel mekanizma türüdür. Zamanla paralel mekanizmaların en çok kullanıldığı alanlardan birisinin uçuş simülâtörleri olmasıyla ve bu alanda da kullanan ilk kişinin D. Stewart olmasıyla, genel olarak bu paralel yapılara ‘Stewart Platformu’ adı verilmiştir. Bu mekanizmanın uçuş simülâtörü olarak kullanılmasındaki amaç, pilotların uçuş eğitimi zamanlarını azaltmak ve gerçek bir uçaktaki öteleme ve dönme hareketlerini pilotlara hissettirebilmektir. Teknolojinin gelişimiyle uçuş simülâtörlerinde ki doğruluk ve hassasiyet de artmıştır. Böylece pilotlar tehlikeli durumlarda ne yapmaları gerektiğini bu eğitim simülâtörlerinde öğrenebilmektedirler. Bu nedenle uçuş simülâtörleri pilotların eğitiminde oldukça önemli bir yere sahiptir, dolayısıyla Stewart platformunun en çok kullanıldığı alanların başında havacılık eğitimleri gelmektedir. Ancak zaman içerisinde araştırmacılar uçuş simülâtörü olarak önerilen paralel mekanizmaları birçok değişik uygulama alanında da kullanılabileceğinin farkına varmışlardır.

Endüstri ve ekonomideki değişimlerle ilgili olarak paralel mekanizmalarda gelişmektedirler ve son zamanlarda teknolojiye birçok alanda yüksek kararlılığa ve hassasiyete ihtiyaç duyulmaktadır. Küçük bir çalışma alanına sahip olmalarına



rağmen montaj, paketleme, tutma ve konumlandırma gibi işlemlerde paralel robotlar seri robotlara göre daha fazla tercih edilmeye başlanmıştır [10]. Havacılık sektörü dışında kullanılan diğer alanların başında da; tıp alanı, uydu konumlandırma, oyun simülatörleri ve petrol platformları gelmektedir.

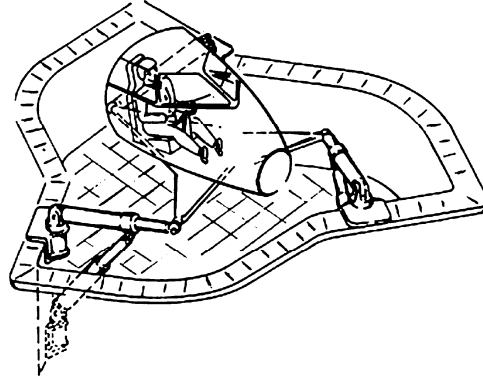
1965 yılında Stewart'ın tanımladığı mekanizma Şekil 1.8'de görülmektedir. Bakılacak olursa Stewart'ın tasarladığı mekanizma günümüzde anılandan biraz farklıdır. Şekildeki mekanizmanın üçgen hareketli platformu yere özdeş mekanizmalar ile bağlanmıştır. Bu mekanizma sabit dik eklemlere iki noktadan bağlıdır ve kirişler birbirlerine döner mafsallarla bağlıdır, kirişlerden bir tanesi de 'Ball and Socket' eklemiyle  $B_i$  noktasından hareketli platforma bağlıdır. Herbir kirişteki lineer motorlar kiriş boylarının değişmesini sağlar. Altı kirişin, hareketli platformun herhangi bir duruşuna karşılık gelen tek bir uzunlukları vardır.



Şekil 1.8 : Stewart'ın tasarladığı paralel mekanizma yapısı.

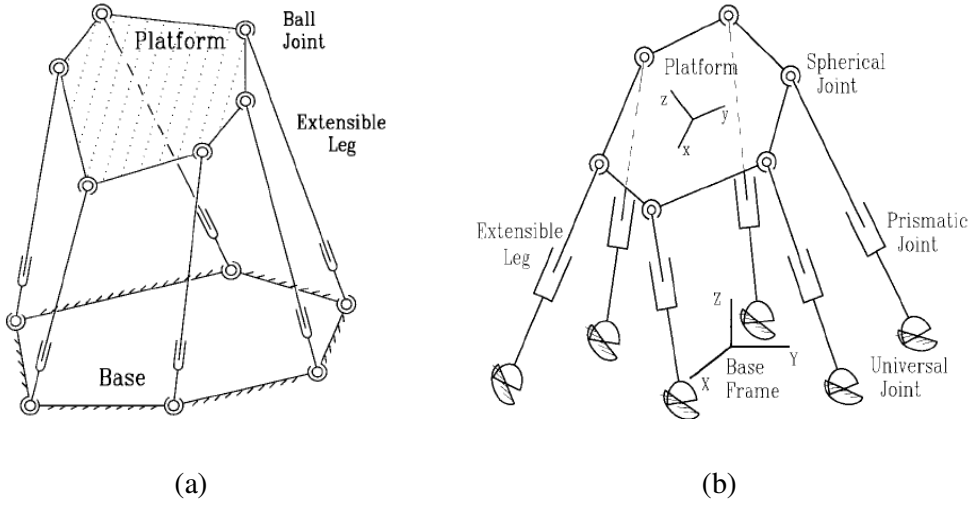
Literatürde ki bazı kaynaklarda tasarlanan ilk Stewart Platformu yapısının, iki rijid platformun 6 değişebilen uzunluktaki bacaklarla bağlanmış şekli gösterilmektedir. Ancak genel olarak 'doğru' Stewart Platformu yapısının yani Stewart'ın ilk tasarladığı yapının Şekil 1.8'deki yapı olduğu kanısı yaygındır [11]. Yine aynı makale de anlatılan ve uçuş simülatörü olarak SPM yapısının nasıl kullanılacağını gösteren resim Şekil 1.9'da görüldüğü gibidir.

Stewart Platform yapısının günümüzdeki haline gelişi ise bu konuda çok değerli çalışmalara imza atmış olan Gough'un, altı bacağın da birbirine paralel çalışmasını önermesiyle olmuştur.



**Şekil 1.9 :** 1965 yılında Stewart tarafından yayınlanan makaledeki uçuş simülatörü olarak tasarlanan SPM resmi.

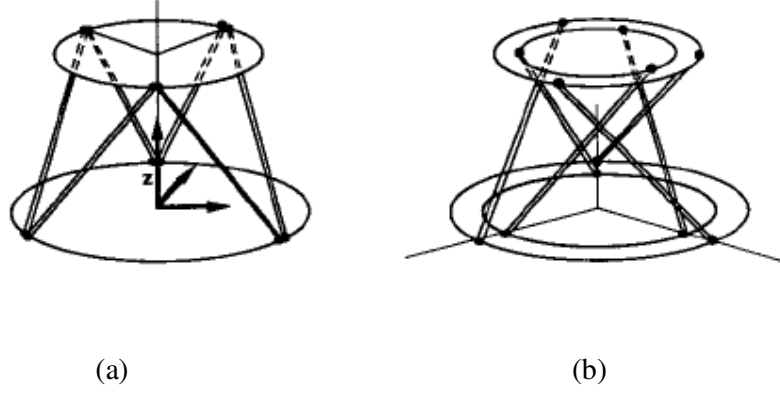
Eklemler yapıları, alt ve üst platformların geometrik yapıları bakımından da farklı Stewart Platform mekanizmalarından bahsedilmiştir. Örneğin Şekil 1.10'da soldaki dizaynda bacaklar küresel-prizmatik- küresel eklemler ile bağlı olduğundan 6-SPS (spherical-prismatic-spherical) Stewart Platformu olarak isimlendirilmiştir, sağ tarafta görülen mekanizma yapısına ise yine eklem yapısından hareketle 6-UPS (universal-prismatic-spherice) Stewart Platformu denilmektedir [12].



**Şekil 1.10 :** 6-SPS Stewart platformu (a) ve 6-UPS Stewart platformu (b).

### 1.3.1 Stewart platformu ile ilgili yapılan çalışmalar

SPM'nin temel sorunlarından birisi, bacakların yönelimleri manipülatörün z ekseninden fazla uzakta olamadığı için bacakların çarpışarak birbirlerini engellemelerinin söz konusu olmasıdır. Sabit bir kuvvet uygulandığında her bir bacak tarafından hareketli platforma iletilir ve bu bacak eksenini üzerinden iletilmek zorundadır. Bu nedenle z eksenindeki güç kapasitesi, x ve y düzlemlerine göre oldukça fazladır, aynı zamanda z eksenindeki tork kapasitesi de sınırlıdır. Bazı araştırmacılar çalışmalarında SPM'nin modifiye edilmiş bir halinden bahsetmişlerdir ve bu değiştirilmiş mekanizma yapısına MSP (Modified Stewart Platform) adını vermişlerdir.

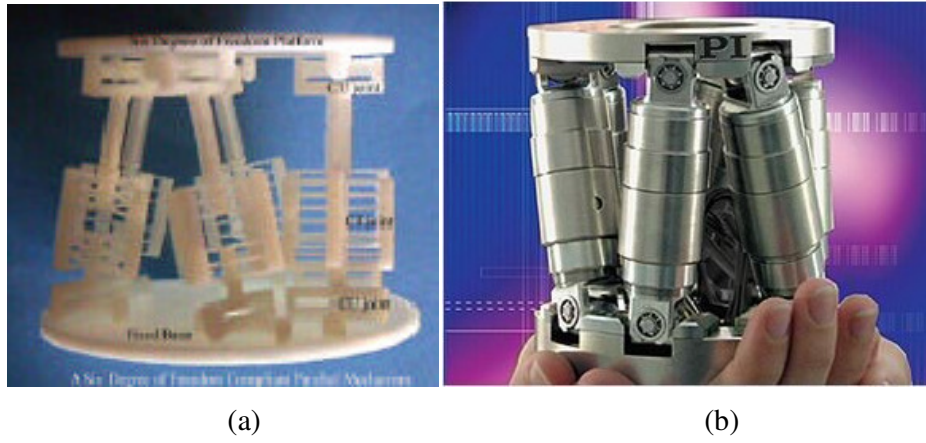


**Şekil 1.11** : Klasik (a) ve (b) modifiye edilmiş (MSP) Stewart Platformu yapısı.

MSP'nin dizaynında 6 bacağın 3 tanesi sabit platformun üzerinde bulunan iç dairenin üzerinde, bu bacakların diğer uçları ise hareketli platformun üzerinde bulunan dairelerden dış dairenin üzerinde konumlandırılmıştır. Bu yapı sayesinde bacakların birbirini engellemeden kesişmesi sağlanmaktadır. Ayrıca bu tasarlanan sistem ile x, y düzlemindeki yük taşıma kapasitesinin ve z eksenindeki tork kapasitesinin artırılacağından bahsedilmiştir [13].

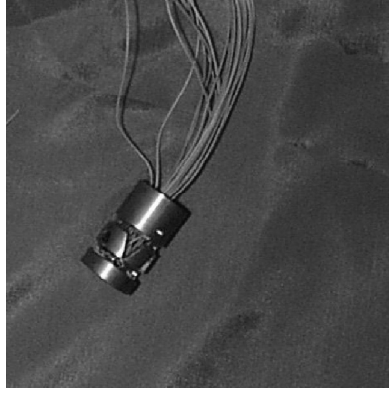
Birçok çalışma da gösteriyor ki doğal yapısı gereği paralel mekanizmalar seri mekanizmalara göre daha kararlı yapıya sahiptirler. Birçok uygulamada yüksek hassasiyetle 3 boyutlu pozisyonlama ve işleme yeteneğine gerek duyulmaktadır. Örnek verilecek olursa, yarı iletken malzeme üretiminde kullanılacak aletlerde, mikro elektronik aygıtlarda hata oranının yok denecek kadar az olması gerekmektedir. Ancak bütün mekanik sistemlerin bu konudaki en büyük dezavantajı,

sahip oldukları eklemlerin üretimi sırasındaki üretim ve histerisis hatalarının mekanizmada ters etkilere neden olmasıdır. Bu nedenle klasik eklem yapılarında yüksek hassasiyet elde edebilmek için büyük mücadeleler verilmiştir. Bu sınırlamanın üstesinden gelmek için tasarımcılar başka bir yol daha bulmuşlardır, esnek eklemler kullanmak, Moon ve Kota da esnek bir paralel mekanizma meydana getirmişler ve bu mekanizmanın farklı bir metot ile analizini yapmışlardır [14]. Esnek bir SPM örneği resmi Şekil 1.12a'da görülmektedir. Ayrıca günümüzde hassas konumlandırma gereken alanlarda kullanmak için 1.12b'deki gibi oldukça küçük boyutlarda SPM'lerin de üretimi yapılmaktadır (Azonanotechnology Co.)

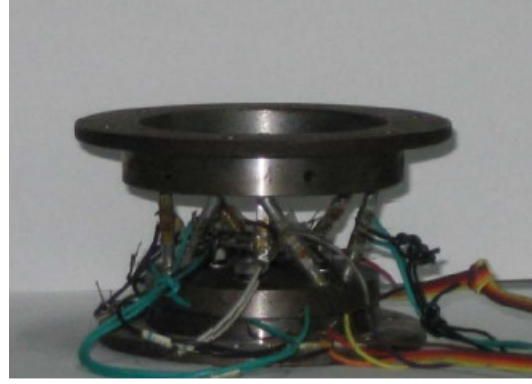


**Şekil 1.12 :** Esnek paralel mekanizma örneği (a) ve Minyatür Hexapod resmi (b)

Son zamanlarda SPM'lerin çalışıldığı bir başka alanda sensör teknolojileridir. SPM tabanlı 6 eksenli kuvvet/tork sensörleri üzerine yapılan araştırmalar mevcut olmakla birlikte, bu sensörlerin bazı avantaj ve dezavantajlarından söz edilmektedir. Yüksek kararlılığa sahip bu yapıların düşük hassasiyete sahip olmaları aşılmaya çalışılan sorunların başında gelmektedir. Konuyla ilgili olarak kullanılan platformun ölçüleri ve performans değerlendirme kriterleri üzerine çalışılmıştır. Çapı 42 mm, uzunluğu 20 mm. boyutlarında olan bir SPM tabanlı kuvvet/tork sensör resmi Şekil 1.13a'da görülmektedir [15]. Bu konu üzerine yapılan başka bir çalışma da ise öngerilimli SPM tabanlı kuvvet/tork sensöründen bahsedilmektedir ve şekil 1.13b'de bu sensörün prototipinin resmi bulunmaktadır [16].



(a)

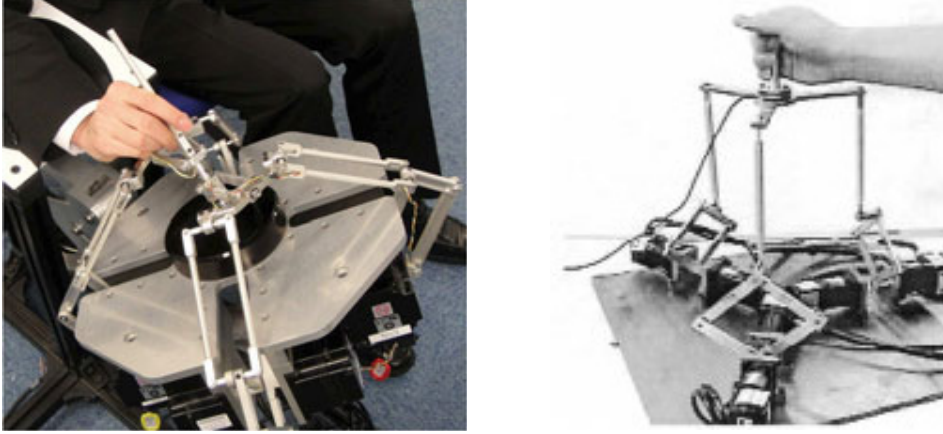


(b)

**Şekil 1.13 :** (a) 6 eksenli SPM tabanlı kuvvet/tork sensörü, (b) Ön gerilimli SPM tabanlı sensörün prototip resmi.

Paralel mekanizmaların kullanıldığı ve tercih edildiği başka bir alanda bilgisayar teknolojisinde ki hızlı gelişmeyle beraber önem kazanan sanal gerçeklik (Virtual Reality) uygulamalarıdır. Bu uygulama alanlarının başında mühendislik, tıbbi operasyonlar, teleoperasyonlar ve eğlence sektörü gelmektedir. Sanal gerçeklik uygulamalarında kullanıcıya gerçek zamanlı olarak dokunma hissini vermek için haptik mekanizmalar kullanılmaktadır. Birçok araştırmacı farklı yapıda haptik sistemi önermiştir; exoskeleton tipi, tool tipi ve seri robot tipi başlıcalarıdır. Ancak geniş bant aralığı, güvenilirliği, az yer kaplaması nedeniyle bu tipler içinden en çok 'tool tipi' kullanılmaktadır ve bu tür haptik sistemlerde paralel mekanizmalar oldukça fazla kullanılmaktadırlar. Tercih edilme sebepleri ise düşük atalet, yüksek kararlılık, kesin çözünürlük olarak sıralanabilir. Ancak şimdiye kadarki yapılan uygulamalar gösteriyor ki sadece paralel mekanizma kullanılan sistemlerde bazı sorunlarla karşılaşmaktadır, bunların başlıca nedeni olarak paralel mekanizmaların dar bir çalışma uzayına sahip olmaları ve düz kinematik çözümlerinin zor olması gösterilebilir. Bu nedenle haptik sistemlerin oluşturulmasında seri ve paralel mekanizmaların ortak kullanılması yönünde eğilim artmıştır. Böylece çalışma uzayı genişlemiş ve düz kinematik kolay bir şekilde çözülebilmektedir. Son yıllarda bazı araştırmacılar yeni bir tool tipi haptik sistemde kullanılmak üzere bir paralel mekanizma tasarlamışlardır. Bu yeni tasarımda daha iyi bir dinamik yapı için motorlar sabit platforma bağlanmıştır ve mekanizma RRR tipi küresel mafsallardan oluşturulduğu için daha geniş bir çalışma uzayına sahip olmuştur [17-18]. İlerleyen yıllarda ise haptik sistemler üzerine yapılan çalışmalar oldukça hızla yol almış ve

paralel mekanizmaların avantajlarına sahip ancak daha geniş dairesel çalışma alanı imkanı sağlayan sekiz motorlu haptik sistem tasarlanmıştır [19]. Şekil 1.14’de haptik sistemler de paralel mekanizmaların kullanımına dair örnek resimler bulunmaktadır.

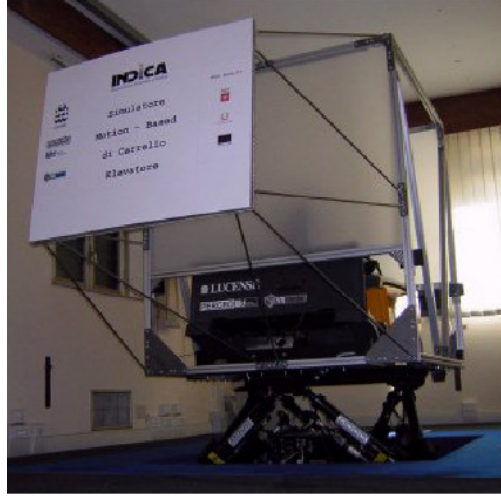


**Şekil 1.14 :** Paralel mekanizmaların haptik sistemlerde kullanımı

İki kompakt SPM’nin birarada kullanılarak yürüme simülasyonu yaptırabilecek bir prototipi fizik tedavi alanında kullanılmaktadır. Bu uygulama da yukarıda bahsettiğimiz haptik sistemlere bir örnek teşkil etmektedir, sistemde kişiye sanal yer hissi verilmekte ve kişinin yürümesi ile SPM hareket etmektedir. Görüldüğü gibi bu da paralel mekanizmaların kullanıldığı başka bir alana daha işaret etmektedir [20].



**Şekil 1.15 :** Rehabilitasyon için kullanılan platform.



**Şekil 1.16 :** Forklift eğitim simülatorü.

Araba kullanmak ve uçuş eğitimleri başından beri kişisel tecrübe ile pekişen uygulamalardır. Havacılıkla ilgili birçok uygulamada simülatorler kullanıcılara büyük bir tecrübe kazandırmakta ve riske girmeden tehlike anını yaşatabilmektedir. Aynı amaç doğrultusunda araştırmacılar bu kez sundukları sistem ile simülator teknolojisini, forklift kullanımı eğitimlerine de getirmektedirler. Çünkü sürücüler yeterli eğitim almadıklarında bazı kaza ve yaralanmalar meydana gelebilmektedir. Bu durumu ortadan kaldırmak için bir simülator tasarlanmıştır. Araç tarafından bütün fiziksel etkiler kullanıcıya iletilmekte ve aynı zamanda sürücü tıpkı gerçek bir forklift sürüyormuş gibi pedal ve diğer araçlarla harekete yön verebilmektedir. Böylece gerçek hayata geçmeden yeterli eğitim alınarak kazaların önüne geçilmesi amaçlanmaktadır [21]. Şekil 1.16’da bu farklı kullanım alanına örnek bir sistem görülmektedir.

Yapılan birçok araştırma göstermiştir ki paralel robotların ve seri robotların sunduğu farklı imkanlar ve kolaylıklar vardır. Günümüzde kullanılan robotların çoğunluğunun açık kinematik zincir yapısında olduğu görülmesine rağmen (Seri Manipulatorler), kapalı kinematik zincir yapısına sahip manipulatorlerin de seri manipulatorlere göre bazı avantajları vardır. Bunlar şu şekilde sıralanabilir:

- 1) Yük birçok paralel bacak tarafından taşındığı için, yüksek ağırlık taşıma kapasitesi
- 2) Eklem hatalarının ardarda toplanmaması nedeniyle, yüksek hassasiyet
- 3) Kapalı kinematik yapısı nedeniyle, yüksek rijitliğe sahip olması

- 4) Yüksek pozisyonlama hassasiyeti
- 5) Motorların sabit platform üzerine bindirilmiş olması nedeniyle, ağırlıkta hafifleme olması
- 6) Ters kinematik denklemlerinin tek çözümlerinin olması

Paralel manipülatörlerin dezavantajları olarak ise; küçük çalışma uzayı, tekillikler ve karmaşık ileri kinematik çözümleri gösterilebilir [22].

Robot teknolojisindeki gelişmeler ve ileri uzaysal mekanizma teorisindeki ilerlemeler ışığında paralel robotlar, robotik teknolojinin önemli bir dalı olmuştur. Bununla beraber birçok akademisyen mekanizmanın boyut analizi, dinamik, kinematik, çalışma uzayı ve tekillikleri konusunda çalışmalarda bulunmuştur ki bunlar genelde Gough-Stewart platformu baz alınarak yapılan çalışmalardır. Ancak bazı özel çalışmalarda seri ve paralel robotların dezavantajlarından kaçınmak için hibrit yani her iki mekanizma tipinin bir arada kullanıldığı sistemler kullanılmaktadır. Seri sistemlerde bütün yük alt eksenlerin üzerine bindiği için özellikle büyük yük taşıyan sistemlerde alt eksenlerin dinamik davranışı çok kötü olmaktadır. Ayrıca seri mekanizmalarda, her bir eklemdaki hataların birbirine eklenerek artması uç işlevcinin doğruluğunun düşük olmasına yol açar. Bu kusurlardan dolayı seri mekanizmanın yapısı sistem performansını etkiler.

Seri, paralel ve hibrit mekanizma yapıları arasında, en çok mimari yapı çeşitliliği sunan mekanizmalar paralel sistemlerdir. Bu bahsedilen yapısal olasılıklardaki fazlalığı Stewart Platformu'nun literatür geçmişinde de görmek mümkün, şöyle ki; 3-3, 3-6, 4-4, 4-6, 6-6 yapılarındaki SPM'ler araştırmalar da önerilmiştir [23,24,25,26,27]. Yapısal olasılıklardaki çeşitlilikte paralel mekanizmaların artlarından bir tanesidir.



## **2. KİNEMATİK ANALİZ**

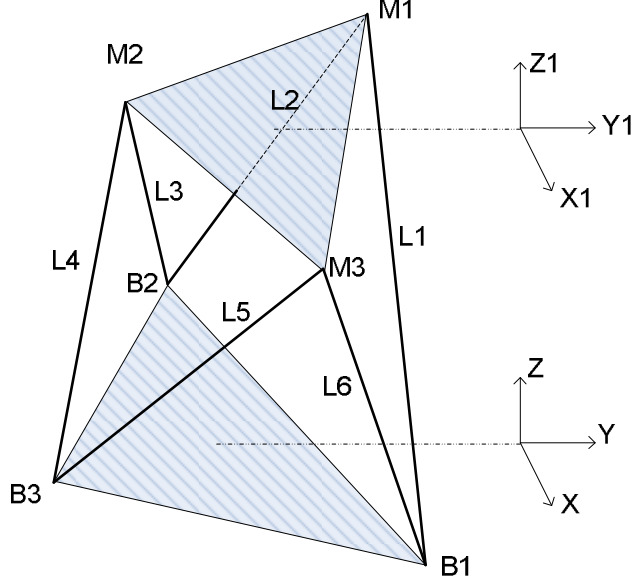
Seri ve paralel robot ayırt edilmeksizin, mekanizmaların hareketlerinin planlanması, kontrolü, dinamiği için herşeyden önce kinematik analizlerinin yapılması gerekmektedir. Ters kinematik analiz; paralel mekanizmalarda üst platformun konumu bilinirken bacak boylarının bulunması, seri mekanizmalar için ise uç işlevcisinin konumundan bacakların açılarının ve konumlarının bulunmasıdır. Düz kinematik analiz kısmında ise ters kinematiğin izlediği yolun tam tersinden gitmek, yani bacak uzunluklarından üst platformun ya da uç işlevcisinin açısal ve doğrusal konumunun bulunması işlemi gerçekleştirilmektedir. Ters kinematik paralel mekanizmalar için kolay iken seri mekanizmalar için ağır işlem yükleri gerektirmektedir. Düz kinematik hesabı ise paralel mekanizmalar da tek bir sonuç vermemesi nedeniyle uğraştırıcı, seri mekanizmalar için daha kolaydır.

### **2.1 Ters Kinematik Analiz**

Paralel mekanizmalar için ters kinematik analiz demek; doğrusal ve açısal konumu bilinene üst platformun pozisyonundan yola çıkarak buna karşılık gelen bacak boylarının bulunması demektir. Burada sabit platformun merkezinde bir referans eksen takımı olduğu kabul edilmiş, alt platform ve üst platform köşe noktalarının bu eksen takımına göre konumları bulunmaya çalışılmıştır. Daha sonra kullanılan yöntemde; bu konumların birbirinden çıkarılmasıyla ve elde edilen vektörün normunun alınmasıyla bacak boylarının bulunması sağlanmıştır.

#### **2.1.1 3x3 Stewart platformunun ters kinematik analizi**

Bu çalışmada, elimizde bulunan deneysel sistem 3x3 lük bir Stewart platform mekanizması olduğu için Şekil 2.1’de temsili olarak yapısı gösterilen mekanizmanın ters kinematik analizi yapılmıştır.



**Şekil 2.1 :** Kinematik analizi yapılan 3x3 lük Stewart Platformunun temsili gösterimi.

Daha öncede bahsedildiği gibi öncelikle üst platformun köşe noktalarının, sabit platformdaki referans eksen takımına göre konumları bulunmakta ve üst köşe noktalarının konumlarından alt platformun köşe noktaları çıkarılarak, bulunan vektörün normunun alınmasıyla bacak boylarına ulaşılmaktadır.

İlk olarak üst platformun kendi eksen takımına göre bilinen köşe noktalarının konumları, R dönme matrisi ve t öteleme vektörü kullanılarak sabit platformdaki eksen takımına göre ifade edilmiştir. Üst platformun x ekseninde  $\alpha$ , y ekseninde  $\beta$ , z ekseninde  $\gamma$  açıları ile döndüğü kabul edilmiş ve R rotasyon matrisi yazılmıştır.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & -\sin \alpha & \cos \beta \end{bmatrix} \quad (2.1)$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Herbir eksendeki dönme matrisleri denklem (2.1) de yazılmıştır. Toplam dönme matrisi ise aşağıdaki denklem (2.2) de görülmektedir.

$$R = \begin{bmatrix} \cos \gamma \cos \beta & -\sin \gamma \cos \beta & \sin \beta \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & \cos \alpha \cos \gamma - \sin \gamma \sin \beta \sin \alpha & -\sin \alpha \cos \beta \\ -\cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \sin \beta \cos \alpha \sin \gamma + \sin \alpha \cos \gamma & \cos \beta \cos \gamma \end{bmatrix} \quad (2.2)$$

Üst platformun köşe noktalarının üst platform eksen takımına göre konumları p'ler ile, alt platform eksen takımına göre konumları P'ler ile ifade edilmiştir.

$$P_{M_1} = p_{M_1} R + t \quad (\text{M2 ve M3 köşeleri için de aynısı yazılabilir.}) \quad (2.3)$$

Artık üst platformun ve alt platformun köşe noktalarının koordinatları referans eksen takımına göre bilindiğine göre, iki konum arasındaki fark vektörünün normu alınarak bacak boyları bulunabilir.

$$L_1 = \sqrt{[P_{M_1}(x) - B1(x)]^2 + [P_{M_1}(y) - B1(y)]^2 + [P_{M_1}(z) - B1(z)]^2} \quad (2.4)$$

Sistemin tüm bacak boyları ( $L_1 \dots L_6$ ) yukarıda verilen (2.4) nolu denklemden bulunmaktadır. Ek A.4'de yazılan ters kinematik analiz programının MATLAB kodları bulunmaktadır.

## 2.2 Düz Kinematik Analiz

Düz kinematik analizde amaç verilen bacak boylarından hareket ederek, üst platformun alt platformdaki eksen takımına göre yaptığı öteleme miktarlarının ve dönme açılarının bulunmasıdır. Ancak paralel robotlar için ileri kinematiğin ters kinematik kadar hızlı çözülemeyeceğini daha öncede belirtmiştik. Bu nedenle ilk olarak düz kinematik denklemlerin oluşturulması, bunun için de gerekli geometrik kabullerin yapılması ve daha sonrada bu karmaşık denklemlerin çözülmesi gerekmektedir. Tek bir sonucu olmayan bu denklemlerin çözümünde kullanmak ve bulunan sonuçlardan doğru olanı bulabilmek için farklı metodlar denenmiştir. Araştırmacılardan Sadjadian [28] bunları genel olarak üç başlık altında sıralamıştır;

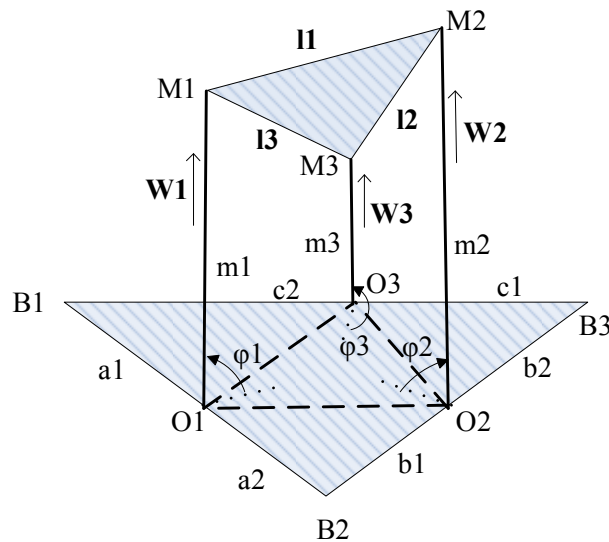
sayısal (nümerik), analitik yöntemler ve kapalı zincir yapısı çözümleri olmak üzere ayrılmışlardır.

Literatürde birçok değişik yöntem kullanılarak düz kinematik analizi yapılmıştır. Wan [29] ve Nyugen [30], Jakobiyen matrisi kullanarak, Tsai [31] ve Zhang [32], farklı geometrik yaklaşımlarından kapalı çevrim çözümleri hesaplayarak, Lee [33], Sylvester matrisi oluşturup eleme yaparak, Geng [34] ise çok katlı YSA yapısı kullanarak, düz kinematik çözümünü hesaplamışlardır. Bunlara ek olarak Jakobovic [35], düz kinematik çözümü için Powell, Hooke-Jeeves ve NR optimizasyon metodlarını kullanmıştır.

Bu çalışmada düz kinematik denklemlerin çözümü için cebirsel bir metod olan Bezout Teoremi kullanılmıştır.

### 2.3 3x3 Stewart platformunun düz kinematik çözümünde Bezout metodunun kullanılması

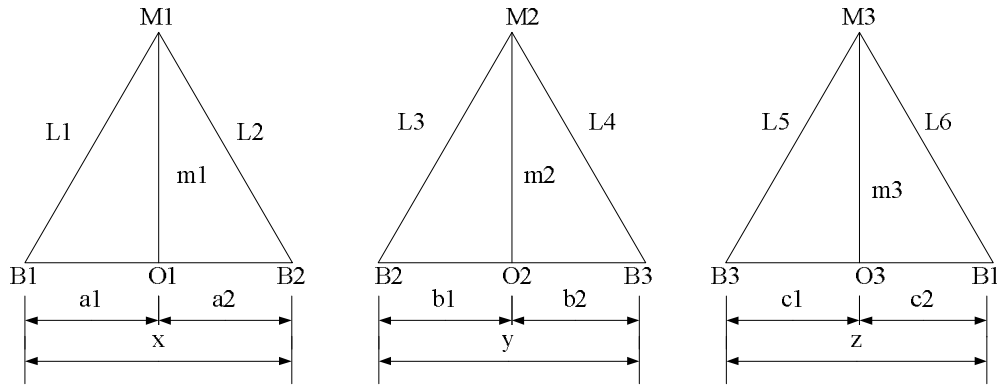
Öncelikle sistemin denklemlerinin oluşturulabilmesi için, 3x3 lük SPM'nin geometrik eşdeğeri çıkarılmış ve gerekli değişkenler belirlenerek denklemler yazılmıştır. Bu kabul edilen geometrik model ve değişkenleri Şekil 2.2'de görülmektedir.



Şekil 2.2 : Düz kinematik analiz için kabul edilen geometrik model.

M1, M2, M3 noktalarını bacakların üst platform bağlantı noktaları olarak, B1, B2, B3 noktalarını da bacakların alt platforma bağlandıkları noktalar olarak ifade edebiliriz.

Şekil 2.3'de ise geometrik yapının açık hali gösterilmiştir. Bu çizimde M1 noktasının, B1 merkezli ve L1 yarıçaplı küre ile B2 merkezli L2 yarıçaplı kürenin kesişiminden meydana gelen çemberin üzerinde olduğu düşünülmektedir. Bu çemberin merkezi O1 ve yarıçapı m1 olarak gösterilmektedir. Aynı mantık ile M2 noktası, B2 merkezli ve L3 yarıçaplı küre ile B3 merkezli L4 yarıçaplı kürenin kesişiminde ve son olarak M3 noktası, B3 merkezli ve L5 yarıçaplı küre ile B1 merkezli L6 yarıçaplı kürenin kesişiminde bulunmaktadır. Sırası ile çemberin merkezleri O2, O3 ve yarıçapları m2, m3 olarak gösterilmiştir.



**Şekil 2.3 :** Kabul edilen geometrik yapının açık hali

Sistemin yukarıda gösterilen geometrik eşdeğeri oluşturulduktan sonra, değişkenlerin bilinenler cinsinden yazılmasına geçilmiştir. Geometrik formüllerden yararlanılarak aşağıdaki (2.5) ve (2.6) denklemleri yazılmıştır.

$$\begin{aligned}
 a_1 &= \frac{x^2 + L_1^2 - L_2^2}{2x} & a_2 &= x - a_1 \\
 b_1 &= \frac{y^2 + L_3^2 - L_4^2}{2y} & b_2 &= y - b_1 \\
 c_1 &= \frac{z^2 + L_5^2 - L_6^2}{2z} & c_2 &= z - c_1
 \end{aligned}
 \tag{2.5}$$

$$\begin{aligned}
m_1 &= (L_1^2 - a_1^2)^{1/2} \\
m_2 &= (L_3^2 - b_1^2)^{1/2} \\
m_3 &= (L_5^2 - c_1^2)^{1/2}
\end{aligned} \tag{2.6}$$

Denklemlerin çözümlenebilmesi için  $O_1$ ,  $O_2$  ve  $O_3$  noktalarının konum vektörleri tanımlanmıştır ve bu noktaların tanımlandığı koordinat sistemi alt platformun koordinat sistemi ile çakışık olarak kabul edilmiş ve böylece aşağıdaki (2.7) numaralı denklem sistemi elde edilmiştir.

$$\begin{aligned}
P_{O_1} &= P_{B_1} + \frac{a_1(P_{B_2} - P_{B_1})}{|P_{B_2} - P_{B_1}|} \\
P_{O_2} &= P_{B_2} + \frac{b_1(P_{B_3} - P_{B_2})}{|P_{B_3} - P_{B_2}|} \\
P_{O_3} &= P_{B_3} + \frac{c_1(P_{B_1} - P_{B_3})}{|P_{B_1} - P_{B_3}|}
\end{aligned} \tag{2.7}$$

Sabit platformun koordinat merkezinden  $B_1B_2$ ,  $B_2B_3$  ve  $B_3B_1$  kenarlarına dik çizilen doğruların sabit platformun x eksenini ile yaptığı açılar  $\beta$  açıları olarak tanımlanmakta ve (2.8) numaralı denklemler ile ifade edilmektedir.

$$\begin{aligned}
\beta_1 &= \sin^{-1} \left[ \frac{\{(\overline{P_{B_2}} - \overline{P_{B_1}}) \times k\} \cdot j}{|(\overline{P_{B_2}} - \overline{P_{B_1}}) \times k|} \right] \\
\beta_2 &= \sin^{-1} \left[ \frac{\{(\overline{P_{B_3}} - \overline{P_{B_2}}) \times k\} \cdot j}{|(\overline{P_{B_3}} - \overline{P_{B_2}}) \times k|} \right] \\
\beta_3 &= \sin^{-1} \left[ \frac{\{(\overline{P_{B_1}} - \overline{P_{B_3}}) \times k\} \cdot j}{|(\overline{P_{B_1}} - \overline{P_{B_3}}) \times k|} \right]
\end{aligned} \tag{2.8}$$

Şekil 2.2’de görülen  $w_1$ ,  $w_2$ ,  $w_3$  vektörleri sırasıyla  $O_1M_1$ ,  $O_2M_2$ ,  $O_3M_3$  doğruları üzerindeki birim vektörlerdir. Aşağıdaki denklem takımıyla ifade edilmiştir.

$$\begin{aligned}
w_1 &= \cos \beta_1 \cos \varphi_1 i + \sin \beta_1 \cos \varphi_1 j + \sin \varphi_1 k \\
w_2 &= \cos \beta_2 \cos \varphi_2 i + \sin \beta_2 \cos \varphi_2 j + \sin \varphi_2 k
\end{aligned} \tag{2.9}$$

$$w_3 = \cos \beta_3 \cos \varphi_3 i + \sin \beta_3 \cos \varphi_3 j + \sin \varphi_3 k$$

Bu aşamada  $M_1, M_2, M_3$  noktalarının pozisyonları şu vektörler ile gösterilebilir;

$$P_{M_1} = P_{O_1} + m_1 w_1$$

$$P_{M_2} = P_{O_2} + m_2 w_2 \quad (2.10)$$

$$P_{M_3} = P_{O_3} + m_3 w_3$$

Tabiki üst platformun köşe noktaları arasındaki uzaklık, sistemin rijit olmasından dolayı sabit olmalıdır ve aşağıda gösterilen denklemler sağlanmalıdır.

$$|P_{M_1} - P_{M_2}|^2 = l_1^2$$

$$|P_{M_2} - P_{M_3}|^2 = l_2^2 \quad (2.11)$$

$$|P_{M_3} - P_{M_1}|^2 = l_3^2$$

Düz kinematik analiz çözümlenmesinin bu kısmına kadar, henüz sistemin geometrik eşdeğeri oluşturulmuş ve geometrik denklemlerden yararlanılarak bilinmeyenler bilinen değerler cinsinden yazılmaya çalışılmıştır.

Bu kabuller ve belirlemeler yapıldıktan sonra düz kinematik çözüm yöntemimiz olan Bezout metodunun asıl uygulamasına geçilebilir. Bezout metodunu birden fazla değişkene bağlı olan denklem sistemlerinin çözümünde değişken sayısını azaltarak, sistemin çözümüne ulaşılmasını sağlayan cebirsel bir yöntem olarak tanımlayabiliriz. Bezout'un uygulanması mantığında Bezout matrisi oluşturularak değişken sayısının bir azaltılması vardır. Aynı zamanda değişken sayısı bir azalırken polinomun derecesi iki katına çıkmaktadır. Tek değişkene düşürülen denklemlerin köklerinden yola çıkılarak diğer bilinmeyenler hesaplanır.

Bezout yöntemini uygulayacağımız denklem sistemini, (2.10) numaralı denklemleri (2.11) denklemlerinde yerine koyarak bulabiliriz.

$$l_1 = ((O_1)_x - (O_2)_x + a_1 \cos \beta_1 \cos \varphi_1 - a_2 \cos \beta_2 \cos \varphi_2)^2 + ((O_1)_y - (O_2)_y + a_1 \sin \beta_1 \cos \varphi_1 - a_2 \sin \beta_2 \cos \varphi_2)^2 + (a_1 \sin \varphi_1 - a_2 \sin \varphi_2)^2$$

$$l_2 = ((O_2)_x - (O_3)_x + b_1 \cos \beta_2 \cos \varphi_2 - b_2 \cos \beta_3 \cos \varphi_3)^2 + ((O_2)_y - (O_3)_y + b_1 \sin \beta_2 \cos \varphi_2 - b_2 \sin \beta_3 \cos \varphi_3)^2 + (b_1 \sin \varphi_2 - b_2 \sin \varphi_3)^2 \quad (2.12)$$

$$l_3 = ((O_3)_x - (O_1)_x + c_1 \cos \beta_3 \cos \varphi_3 - c_2 \cos \beta_1 \cos \varphi_1)^2 + ((O_3)_y - (O_1)_y + c_1 \sin \beta_3 \cos \varphi_3 - c_2 \sin \beta_1 \cos \varphi_1)^2 + (c_1 \sin \varphi_3 - c_2 \sin \varphi_1)^2$$

Denklemlerin karmaşıklığını gidermek için sadeleştirme yapılırsa;

$$\begin{aligned} V_1 \cos \varphi_1 + V_2 \cos \varphi_2 + V_3 \cos \varphi_1 \cos \varphi_2 + V_4 \sin \varphi_1 \sin \varphi_2 + V_5 &= 0 \\ Y_1 \cos \varphi_2 + Y_2 \cos \varphi_3 + Y_3 \cos \varphi_2 \cos \varphi_3 + Y_4 \sin \varphi_2 \sin \varphi_3 + Y_5 &= 0 \end{aligned} \quad (2.13)$$

$$Z_1 \cos \varphi_3 + Z_2 \cos \varphi_1 + Z_3 \cos \varphi_3 \cos \varphi_1 + Z_4 \sin \varphi_3 \sin \varphi_1 + Z_5 = 0$$

Yukarıda ki (2.13) numaralı denklemler elde edilebilir,  $V_i$ ,  $Y_i$ ,  $Z_i$  ( $i:1\dots5$ ) katsayılarının açık halleri ek-A1 de mevcuttur.

Bu denklemlerden muhtemel  $\varphi_i$  ( $i:1, 2, 3$ ) açılarını bulmamız gerekmektedir. Tam burada  $x_i = \tan(\varphi_i/2)$  dönüşümünü uygulayarak elde edeceğimiz yeni denklem sisteminden Bezout Matrisini oluşturabilir, bulacağımız denklem köklerinden de ters dönüşümle tekrar  $\varphi_i$  açılarına dönebiliriz.

$$\begin{aligned} (K_1 x_1^2 + K_2) x_2^2 + (K_3 x_1) x_2 + (K_4 x_1^2 + K_5) &= 0 \\ (L_1 x_3^2 + L_4) x_2^2 + (L_3 x_3) x_2 + (L_2 x_3^2 + L_5) &= 0 \\ (M_1 x_1^2 + M_4) x_3^2 + (M_3 x_1) x_3 + (M_2 x_1^2 + M_5) &= 0 \end{aligned} \quad (2.14)$$

$K_i$ ,  $L_i$ ,  $M_i$  ( $i:1\dots5$ ) katsayılarının açık halleri ek-A2 de mevcuttur.

Bezout Metodu ile yukarıdaki denklemlerden  $x_2$  elimine edilerek Bezout matrisi oluşturulursa;

$$\begin{vmatrix} \begin{vmatrix} K_1 x_1^2 + K_2 & K_4 x_1^2 + K_5 \\ L_1 x_3^2 + L_4 & L_2 x_3^2 + L_5 \end{vmatrix} & \begin{vmatrix} L_3 x_3 & K_3 x_1 \\ L_1 x_3^2 + L_4 & K_1 x_1^2 + K_2 \end{vmatrix} \\ \begin{vmatrix} K_3 x_1 & K_4 x_1^2 + K_5 \\ L_3 x_3 & L_2 x_3^2 + L_5 \end{vmatrix} & \begin{vmatrix} K_1 x_1^2 + K_2 & K_4 x_1^2 + K_5 \\ L_1 x_3^2 + L_4 & L_2 x_3^2 + L_5 \end{vmatrix} \end{vmatrix} \quad (2.15)$$

matrisi elde edilir. Buradaki matrisin determinantı hesaplanır ve katsayılar kullanılarak sadeleştirme işlemi yapılırsa aşağıdaki denklem elde edilir.

$$P_1 x_3^4 + P_2 x_3^3 + P_3 x_3^2 + P_4 x_3 + P_5 = 0 \quad (2.16)$$

Buradaki  $P_i$ 'ler mekanizmanın geometrik yapısından gelen sabitlerden ve  $x_1$  değişkeninden oluşmaktadır, aşağıda açık şekilleri görülebilir.



$$\begin{aligned}
P_1 &= R_1 x_1^4 + R_2 x_1^2 + R_3 \\
P_2 &= R_4 x_1^3 + R_5 x_1 \\
P_3 &= R_6 x_1^4 + R_7 x_1^2 + R_8 \\
P_4 &= R_9 x_1^3 + R_{10} x_1 \\
P_5 &= R_{11} x_1^4 + R_{12} x_1^2 + R_{13}
\end{aligned} \tag{2.17}$$

$R_i$ ' lerin açık halleri ise ek-A3 de verilmiştir.

Bizim Bezout Metodu ile yapmak istediğimiz denklemin derecesi artsa bile bilinmeyen sayısını teke indirmektir. Bu nedenle yine aynı mantık ile  $x_3$  değişkenini elimine edecek matrisi şu şekilde yazabiliriz;

$$\begin{vmatrix}
P_1 S_3 - P_3 S_1 & P_2 S_3 - P_4 S_1 - P_3 S_2 & -P_5 S_1 - P_4 S_2 & -P_5 S_2 \\
P_1 S_2 - P_2 S_1 & P_1 S_3 - P_3 S_1 & -P_4 S_1 & -P_5 S_1 \\
S_1 & S_2 & S_3 & 0 \\
0 & S_1 & S_2 & S_3
\end{vmatrix} \tag{2.18}$$

Buradaki  $S_i$ ' ler (2.16) denklemdeki  $x_3$ ' ün katsayılarıdır ve aşağıdaki gibi gösterilebilir.

$$\begin{aligned}
S_1 &= M_1 x_1^2 + M_4 \\
S_2 &= M_3 x_1 \\
S_3 &= M_2 x_1^2 + M_5
\end{aligned} \tag{2.19}$$

Yukarıdaki (2.18) numaralı denklemdeki matris sadece  $x_1$ ' e bağlıdır, 16. dereceden denklemler çözülerek  $x_1$ ' in değerleri bulunur. Sistemin mekanik yapısının elvermeyeceği negatif ve imajiner değerler göz ardı edilir. Bulunan diğer her bir  $x_1$  değerleri için iki tane  $x_2$  ve 2 tane  $x_3$  değeri hesaplanır. Bulunan  $x_2$  ve  $x_3$  değerlerinden (2.14) denklemini sağlayan çift bizim çözüm kümemizdir. Buradan tekrar ters dönüşümle  $\varphi_i$  açılarına geçilir, bundan sonra üst platformun köşe noktalarının konumunun hesabı için bilinmesi gereken  $w_i$  vektörleri denklem (2.9) dan hesaplanabilir. Son hesaplamanın ardından denklem (2.10) da bilinmeyen bir ifade kalmadığı için, üst platformun sabit eksen takımına göre konumları  $P_i$ ' ler artık (2.20) denklemden kolayca hesaplanabilmektedir.

$$G = \frac{M1+M2+M3}{3} \quad (2.20)$$

Sonuç olarak düz kinematiğin ana amacı olan üst platformun köşe noktaları ve buradan hareketle üst platformun ağırlık merkezi hesabına ulaşılmış olmaktadır.

Yapılan çalışma da 3x3 SPM'nin düz kinematik analizinin Bezout eliminasyon metodu kullanılarak yapılması üzerine çalışılmıştır. Kodlar MATLAB programında yazılmış ve girilen bacak boylarına göre üst platformun ağırlık merkezinin konumu hesaplanmıştır.

### 3. YAPAY SİNİR AĞLARI

Genel bir tanımla, öğrenme yeteneği açısından insan beyni ile benzerliği olan, deneysel bilgileri biriktiren ve bunları kullanan paralel bir işlemci olarak tanımlanabilirler [36]. Yapay sinir ağları öğrendikleri bilgileri aynı zamanda genelleyebildikleri için çok miktarda karmaşık problemin çözümünü de mümkün kılmışlardır. Klasik yöntemlerde bilgiler kurullarla açıklanır ve ilişkilendirilir. Ancak yapay sinir ağlarında verilen örnekler ile ağın öğrenmesi sağlanır ve ağ kendi kurallarını oluşturur [37]. 1980'lerin sonundan itibaren yükselişe geçmiş bir konu olan yapay sinir ağları, birçok araştırmacı tarafından temel prensiplerinin geliştirilmesi ve yeni algoritmaların üretilmesi için çalışılmıştır [38]. Son yıllarda yapılan araştırmalarda iyi sonuç verdikleri alanların başında; doğrusal olmayan kontrol problemleri, bilgi işleme ve görüntü tanıma problemleri örnek verilebilir. En çok kullanılan ve bilinen ağ yapısı ileri beslemeli (feedforward) olmasına rağmen, geri beslemeli (recurrent) ağ yapısı da literatür de mevcuttur.

#### 3.1 YSA'nın Tanımı ve Literatür İncelemesi

İlk olarak 1943 yılında McCulloch ve Pitts [39] adındaki araştırmacılar kendilerinin nörolojiden anladıklarına ve basit mantık kurallarına dayanan yapay sinir ağı modellerini geliştirmişlerdir. Diğer bir gelişme ise 1954 yılında Minsky [40] tarafından yazılan bir doktora tezidir. Bu çalışmalarda sinir ağının öğrenmesi üzerine durulmuş, genelleme kısmına pek fazla girilmemiştir. Yapay sinir ağları konusunda ki büyük gelişmelerden birtanesi 1958 yılında Rosenblatt [41] tarafından kaydedildi. Araştırmacı ilk defa üç katmanlı bir yapıdan ve yapının öğrenme özelliğinden bahsetmiştir. Yine 1960 senesinde Widrow ve Hoff [42] tarafından yapılan, öğrenme yöntemiyle o güne kadar yapılanlardan farklı olan bir çalışma kayda geçmiştir. Bu çalışmada öğrenme yöntemi olarak en küçük ortalama kareler (LMS- Least Mean Squares) kullanılmıştır. Ancak bu yılın sonlarına doğru Minsky ve Peper [43] tarafından yayınlanan ve yapay sinir ağlarındaki gelişmelerin verimsiz olduğunu açıklayan bir kitap yayınladılar. Bunun da etkisi ile 1970'ler boyunca bu konuda

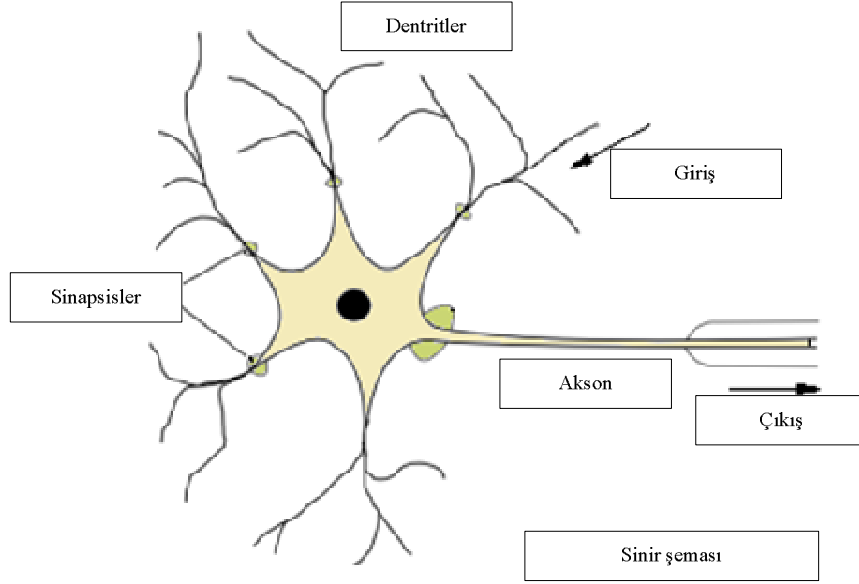
yapılan arařtırmalarda büyük bir düşüş yařanmıřtır, ta ki 1980'in ilk yıllarında fizikçi Hopfield [42] tarafından bu konu tekrar canlandırılıncaya kadar. Arařtırmacı bu çalışmasında yeni bir öğrenme kuralı geliřtirmiřtir. Şöyle ki, sinir ağı bağlantılarındaki ağırlıklarda deęişiklik yapılacaksa bu deęişim önceden belirlenir. Yani veri alıř veriřinde bulunan nöronlar aynı iřaretli ise bağlantı ağırlığı öğrenme oranı kadar artırılır, aksi halde öğrenme oranı kadar azaltılır. Bu konunun dönüm noktalarından olan başka bir büyük geliřme ise, hatanın geri yayılımına (backpropagation) dayanan bir öğrenme algoritmasının geliřtirilmesidir. Bu yöntem ile daha önceden yapay sinir ağılarını sınırlayan tek katmanlı yapılardan sıyrılınmıř, çok katmanlı yapılarla çalışmaya olanak saęlanmıřtır. Bu dönemde Rumelhart'ın [45] kitabının da etkisiyle, birçok arařtırmacı ilginç hesaplamalar gerçekleřtirmiřlerdir. Arařtırmaların bu kadar olumlu geliřmesine, birde bilgisayar teknolojisinde ki ilerleme eklenince yapay sinir ağı çalışmalarında adeta patlama yařanmıřtır. Günümüzde ise artık birçok alanda yapay sinir ağıları başarılı bir şekilde kullanılmaktadır. Bu farklı alanların bazıları ařaęıda sıralanmıřtır.

- Robot ve doęrusal olmayan kontrol sistemleri
- Kalite kontrolü
- Görüntü iřleme uygulamaları
- Finans alanı
- Tıp alanı
- Askeri alanlar (özellikle radar sistemleri)
- Sosyolojik arařtırmalar
- İstatistik ve ekonomi
- Bilgi güvenlik sistemleri

### **3.2 Biyolojik Sinir Hücresi ve Yapay Sinir Modeli**

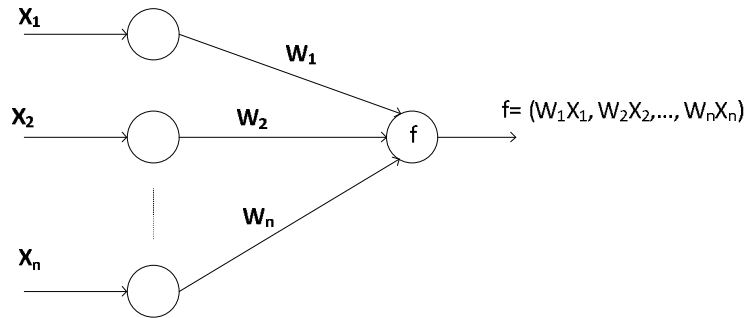
Yapay sinir ağılarının oluşmasında esin kaynağı olan biyolojik sinir ağı yapısını Şekil 3.1'de görebiliriz. Kısaca bu yapıdan bahsederseniz, “dendrit” adı verilen lifli yapılardan sinir hücre sine dış dünyadan sinyaller gelmekte ve “akson denilen” yapıdan ise dış dünyaya bilgi gönderilmektedir. Gelen ve giden bilgilerin iřlenmesi kısmı ise hücrenin çekirdek kısmında gerçekleşmektedir. Nöronlar birbirleri ile birinin aksonuyla dięerinin dendritinin iletiřime geçmesi sayesinde haberleřirler ve bu ince bağlantı aralıklarına “snaps” denmektedir. Bir akson aynı anda birden fazla

dentrit ile bağlantıya geçebilmektedir. İnsan beyinde yaklaşık  $10^{10}$  nöron vardır ve hücre başına bağlantı sayısı  $10^4$  ile ifade edilmektedir. Ancak insan sinir sisteminin bu karmaşık yapısına rağmen oldukça hızlı ve doğru kararlar vermektedir. Günümüzdeki bilgisayarların hiçbiri bu kadar hızlı ve doğru cevaplar vermemesinin en büyük nedeni, insan beyinde daha düşük hızlı ancak çok sayıda paralel işlemcinin olmasıdır.



**Şekil 3.1:** Biyolojik sinir ağı yapısı

Beynin bu kadar kusursuz çalışmasından ve çok karmaşık durumlarda bile doğru sonuçlar üretebilmesinden etkilenilerek, zor problemlerin çözümü için insan beyine benzer bir yapının modellenmesine gidilmiştir. İlk olarak insan beyinin en büyük avantajı olan paralellik özelliği dikkate alınmıştır.



**Şekil 3.2:** Yapay sinir ağı modeli.

Şekil 3.2’den görülebileceği gibi yapay sinir hücreleri paralel olarak düşünülmüştür. Her bir yapay nöron, biyolojik sistemdeki çekirdeğe karşılık gelmektedir. Ayrıca her bir nöronun giriş ve çıkış vardır, tıpkı dentrit ve aksonlar gibi. Sapsler ise yapay sinir ağlarındaki ağırlık değerlerine karşılık gelmekte ve bir önceki nöron ile olan ilişkisi hakkında bilgi vermektedir. Çıkış nöronunda yer alan “f” ise nörondaki çıkış hesaplayan fonksiyondur.

### **3.3 YSA’nın Çalışma Sistemi ve Bileşenleri**

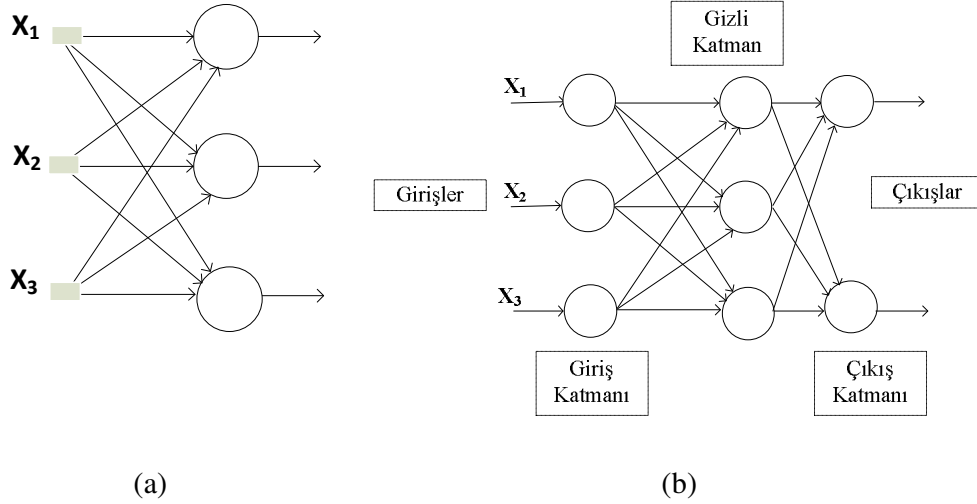
İnsan beyni gibi çalışan, öğrenen ve genelleme yapabilen yapılar oluşturulmak amacıyla ortaya çıkan yapay sinir ağlarının bu özelliklerini ve başarısını belirleyen birkaç kriterden söz edilebilir. Bu bölümde yapay sinir ağlarının performansını etkileyebilecek ağ yapısı, aktivasyon fonksiyonu ve eğitim algoritmaları gibi bileşenlerden bahsedilecektir. Yapay sinir ağları öğrenme ve genelleme olaylarını, girişten aldıkları sürekli verilerle kendi ağırlıklarını değiştirerek yapmaya çalışırlar. Yani öğrenme denilen olay, girdi/çıkış verilerinin arasında tanımlanan öğrenme algoritmasına ve aktivasyon fonksiyonuna göre ağırlık değerlerinin ayarlanmasıyla olur. Bu ağırlık değerleri bir süre sonra belli bir değere yakınsamaya başladığında öğrenme süreci tamamlanmış demektir ve bundan sonraki süreç olan genelleme sürecine geçilebilmektedir. Genelleme sürecinde ise ağın eğitim sırasında öğrenmediği giriş bilgilerine göre bir çıkış oluşturmasıdır. Eğer ağ doğru eğitilmiş ve optimum ağırlık değerleri bulunmuş ise yapılan genellemelerde oldukça yüksek doğruluk elde edilebilmektedir. Ancak yapılan çalışmalar göstermiştir ki ağın eğitimi sırasında eğitim datalarının sınıflandırılması ve miktarı eğitim verimi açısından büyük önem arz etmektedir. Eğer öğrenme için verilen bilgi miktarı yetersiz ise ağ öğrenemez ve genellemede büyük hatalı sonuçlar alabiliriz. Öğrenme verisinin fazla olması da istenmeyen bir durumdur. Şöyle ki ağa gereğinden fazla eğitim bilgisi verilirse, bu defa ağ ezberleme yoluna gidecektir. Ezberleme olayı, öğrenme verilerindeki hatanın azalması ancak test (genelleme) verilerinde ki hatanın artmasıyla anlaşılabilir [46].

### 3.3.1 Yapay sinir ağı yapısının çeşitleri

Genel olarak başlıca üç tane yapay sinir ağı yapısından bahsedebiliriz. Bunlar; tek katmanlı ileri beslemeli yapı, çok katmanlı ileri beslemeli yapı ve geri beslemeli yapılar olarak adlandırılmaktadır [35].

#### 3.3.1.1 Tek katmanlı ileri beslemeli yapay sinir ağı yapısı

Sinir ağı yapılarının en basit halidir. Bir tane çıkış katmanı vardır sadece ve buraya verilen değişken sayıdaki giriş verilerine göre bir çıkış değeri üretir. Çalışma mantığı eşik fonksiyonuna dayanır, genelde sınıflandırma problemlerinde kullanılır. Şekil 3.3a'da yapısı görülmektedir.



Şekil 3.3: Tek katmanlı ileri beslemeli(a) ve (b) ileri beslemeli çok katmanlı ağ.

#### 3.3.1.2 Çok katmanlı ileri beslemeli yapay sinir ağı yapısı

Bir giriş katmanına, bir veya birden fazla gizli katmana ve bir de çıkış katmanına sahip yapıdır. Bu ağların ortaya çıkmasındaki asıl sebep, doğrusal olmayan fonksiyonların tek katmanlı ağlar ile çözümünün mümkün olmayışındır. Bu nedenle daha öncede bahsedildiği gibi araştırmacılardan Rosenblatt [40], böyle bir ağ yapısı denemiş ve bu konuda oldukça büyük bir ilerleme kaydetmiştir. Bu ağ yapısı ile doğrusal olmayan problemlerin çözümü mümkünleşmiş, sınıflandırma, tanıma ve genelleme yapılması istenen problemler için büyük bir gelişme kaydedilmiştir. Örnek birçok katmanlı ileri beslemeli yapı Şekil 3.3b'de görülmektedir. Günümüzde kullanılan en yaygın ve genel yapay sinir ağı mimarisi bu şekildedir.

### 3.3.1.3 Geri beslemeli yapay sinir ağı yapısı

Bu ağ yapısında ağa verilen girdiye karşılık olarak elde edilen çıkış verisi olması gerekenle karşılaştırılır ve elde edilen fark hata olarak adlandırılır. Tüm hatalar ağa geri beslenir ve hatanın azaltılmasına çalışılır. Ancak bu geri besleme durumu ağın eğitimini oldukça yavaşlatmaktadır ve bu dezavantaj nedeniyle geri beslemeli ağların kullanımı ileri beslemeli ağlar kadar yaygın olarak görülmemektedir. Bu yapıdaki ağların da tek katmanlı veya çok katmanlı çeşitleri mevcuttur.

### 3.3.2 Yapay sinir ağı transfer fonksiyonları

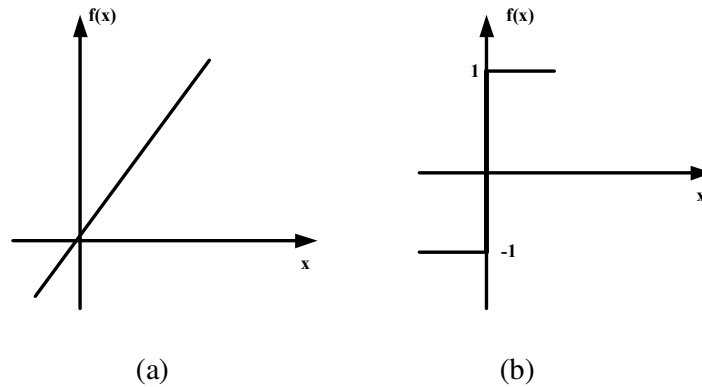
Bir yapay sinir ağı hücresinin net girdilerinden çıkışını hesaplayabilmek için belirlenmesi gereken fonksiyondur. Yaygın olarak kullanılan dört çeşidinden aşağıda bahsedilmiş ve fonksiyonları verilmiştir.

Şekil 3.4a'da 'Doğrusal Fonksiyon' görülmektedir. Doğrusal olan problemlerin çözümünde kullanılabilir. Aşağıda bulunan formülünde ki A, sabit bir sayıdır ve doğrunun eksenler ile yaptığı açığı belirlemektedir.

$$f(x) = A \cdot x \quad (3.1)$$

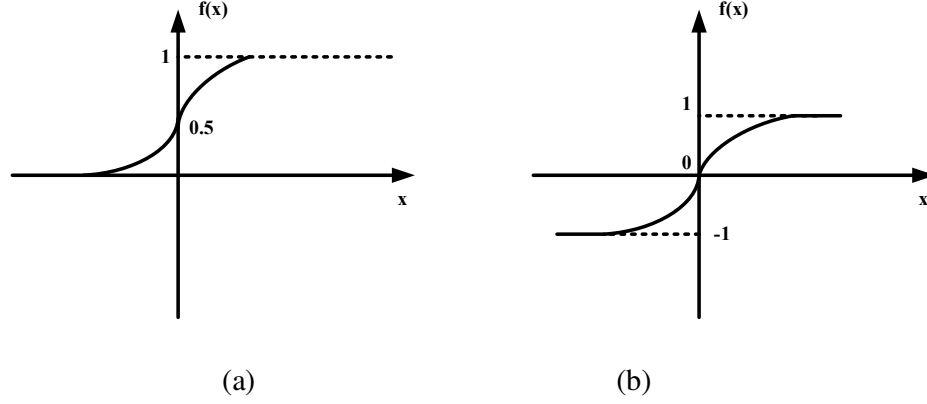
Şekil 3.4b'de 'Basamak Eşik Fonksiyonu' görülmektedir. Girdilerin pozitif olması durumunda +1, diğer tüm durumlarda -1 değerini verir.

$$\begin{aligned} x > 0 \text{ ise, } f(x) &= 1 \\ \text{aksi halde, } f(x) &= -1 \end{aligned} \quad (3.2)$$



Şekil 3.4: Doğrusal (a) ve Basamak Eşik Fonksiyonu (b)





**Şekil 3.5:** Sigmoid (a) ve Tanjant Hiperbolik Fonksiyonu(b)

Şekil 3.5a'da 'Sigmoid Fonksiyonu' görülmektedir. Doğrusal olmayan, sınırlı, monoton artan bir fonksiyondur. Şu şekilde formüle edilmiştir;

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

Son olarak Şekil 3.5'in b şikkında, 'Tanjant Hiperbolik Fonksiyonu' görülmektedir. Aşağıda doğrusal olmayan bu fonksiyonun formülü verilmiştir.

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.4)$$

Bu çalışmada optimizasyonu yapılan yapay sinir ağı yapısında kullanılan transfer fonksiyonu tanjant hiperboliktir.

### 3.3.3 Yapay sinir ağı eğitime algoritmaları ve öğrenme kuralları

Sinir ağının performansını belirleyen en büyük etkenlerden bir tanesi de çözülmesi gereken soruna göre eğitim algoritmasının belirlenmesi hususudur. Genel olarak eğitime algoritmaları üç kısımda sınıflandırılmıştır.

- Destekli eğitim; genelde optimizasyon problemlerinde kullanılmıştır. Bu algorithmada sisteme giriş verilerinin yanında bunlara destek olarak işaretçi girişi de yapılmaktadır.
- Desteksiz eğitim; bu tip sistemlerde, ağın öğrenmesi için destekleyici bir şey yapılmaz. Sadece giriş verileri kullanılarak, ağın çıkıştaki verilere göre bir öğrenme gerçekleştirmesi beklenir.
- Denetimli eğitim; ağın öğrendiğinden emin olana kadar bir destek mevcuttur. Ancak ağırlıkların sabitlendiği durumda, denetim kalkar [47].

Kısacası eğitime algoritmasının belirlenmesindeki amaç kullanılacak öğrenme kuralının ağa nasıl adapte edilmesi gerektiğinin bulunmasıdır. Şimdi de bu öğrenme kurallarından kısaca bahsedilecektir. Araştırmalara konu olan birçok öğrenme kuralı mevcut olmakla birlikte, sadece başlıcaları şu şekildedir;

**Hebb Kuralı:** 1949 yılında Kanadalı bilim adamı Hebb'in "The Organisation of the Behaviour" adlı kitabında, ağdaki hücrelerin arasındaki bağlantı sayısı değiştirildiğinde ağın öğrenebileceğinden bahsedilmiştir [48]. Bu bilgi kullanılan birçok öğrenme algoritmasının temelini oluşturmaktadır [49].

**Adaline Öğrenme Kuralı:** Açılımı "Adaptive Linear Element" olan kural 1959 yılında Widrow ve Hoff tarafından geliştirilmiştir [50]. Delta kuralı olarak da bilinmektedir ve küçük kareler algoritmasını (Least Mean Square) kullanarak öğrenmeyi gerçekleştirir. Bu yapının birden fazlasının bir arada kullanılmasına ise Madaline denilmektedir ve genelde iki katmandan oluşmaktadır. Her katmanda farklı sayıda Adaline ünitesi vardır.

**Kohonen Kuralı:** Bu kural öğrenme metodu olarak denetimsiz öğrenmeye ve kendi kendini organize eden model (Self Organizing Map) yapısına dayanmaktadır [51]. Teorik yapısı tam olarak bitmediği için pek fazla kullanım alanı bulamamıştır.

**Geriye Yayılım (Back Propagation) Kuralı:** Elde edilen çıkış verileri ile olması gereken veriler karşılaştırılarak bir hata değeri elde edilir ve bu hata değerleri ağa geri beslenir. Hataları azaltmak için ağırlık ve bağlantı değerleri değiştirilir. Ta ki istenilen veya minimum hata değerine ulaşıncaya kadar bu işleme devam edilir. Günümüzde özellikle tahmin gerektiren alanlarda oldukça fazla kullanılan bir algoritmadır.

Bu tez çalışmasında doğa esinli hesaplama yöntemi diyebileceğimiz GA ve sezgisel hesaplama yöntemi olan YSA bir arada kullanılmıştır. Bu nedenle bu bölümde YSA'ların literatür çalışmalarından, hesaplama mantığından ve kurallarından genel olarak bahsedilmiştir.

Yapılan çalışmalarda 3 girişli ve 2 çıkışlı bir yapay sinir ağının GA ile tasarımı aynı zamanda eğitimi yapılmıştır. Girişler, SPM'nin üst platformunun köşe noktalarını, çıkışlar ise bu noktaların çalışma uzayı içerisinde olup olmadığı bilgisini tutmaktadırlar. Kullanılan GA yöntemi ile sadece bu bilgileri bilinen ve giriş-çıkış dataları bulunan bir probleme sinir ağı yapısı tasarlanmış ve ağ eğitimine gerek

kalmadan optimum ağırlık ve bias değerleri GA ile bulunmuştur. Yapılan çalışmanın ve kodun daha ayrıntılı anlatımı simülasyonlar ve sonuçlar bölümünde bulunmaktadır.



## 4. GENETİK ALGORİTMALAR

### 4.1 Genetik Algoritmalar Tanımı

Genetik algoritmalara “İnsanlığın doğadan esinlenerek geliştirmiş olduğu ve genelde optimizasyon problemlerinin çözümünde kullandığı bir yöntemdir” diyebiliriz. Bu bölümde Genetik Algoritmaların tanımı ve literatür incelemesinin yanı sıra Genetik Algoritmaları oluşturan temel kavramlar, operatörler, genetik algoritmaların genel akış şeması ve Genetik Algoritma mantığını oluşturan bileşenlerden bahsedilecektir.

Genetik algoritmalar da ilk olarak çözüm adayı olabilecek bireylerden oluşan bir popülasyon oluşturulur, popülasyon denilen bu çözüm kümesi başlangıçta rastgele oluşturulur. Popülasyonu oluşturan bireyler kromozom olarak da adlandırılmaktadır, kromozomlar dizi şeklinde kodlanmış ise her bir elemanına gen denilmektedir. Problemin olası çözümlerinin nasıl ifade edileceği bu kromozom tasarımı ile belirlenir.

Genetik algoritmaların çalışma mantığını ve sonuçlarını çok fazla etkileyen ve her probleme özgü bir ‘Uygunluk Fonksiyonu’ belirlenmesi kısmı, bu yöntemin en önemli kısımlarındandır. Popülasyonu oluşturan her birey için uygunluk değeri hesaplanır ve diğer bireyler ile karşılaştırılır. Uygunluk değeri yüksek olan bireyler bir sonraki nesile aktarılır, doğal seleksiyon mantığı ile kötüler elenir. Popülasyonda ki çeşitliliği artırmak içinse çaprazlama (crossover) ve mutasyon gibi işlemler bireylere uygulanır. Böylece sürekli yeni genotipe sahip bireyler popülasyona katılmış olur ve yeni bireylerin popülasyona katılmasıyla arama uzayının genişlemesi sağlanmış olur. Programda uygulanacak mutasyon ve çaprazlama olasılıkları araştırmacı tarafından belirlenir. Bu işlemler dizisi birçok kez tekrarlanır ve sonuçta yeni bir nesil oluşturulur. Birçok iterasyon yapıldıktan sonra yeni nesillerde bir yada daha fazla yüksek uygunluk değerine sahip bireyler elde edilebilir. Algoritmanın durma kriteri sağlandı ise program durdurulur ve en iyi uygunluk değerine sahip birey problemin çözümü olarak kabul edilir. Durdurma kriteri olarak; nesil sayısı yada belli bir uygunluk değerine ulaşıldığında durmak gibi koşullar olabilir.

Uygunluk deęerinin hata oranı ile belirlenmesi durumunda ise uygunluęu dūřuk olan en iyi bireydir demek daha doęru olacaktır.

Ancak genetik algoritmanın bařlangıç popūlasyonunun tamamen rastgele seřilmesi nedeniyle aynı problem iin aynı algoritma her alıřtırıldıęında birbirine yakın ama farklı sonular elde edebilir. Bu durum Genetik Algoritmaların iřlem sūresinde ki kısıklık ve matematiksel iřlemlerle uęrařmadan özūme gidebilme kolaylıęı dūřünüldūęünde göz ardı edilebilir bir özellięidir.

Genetik algoritmaların dięer özüm yöntemlerinden farklarını řu řekilde açıklayabiliriz;

- Genetik algoritmalar bir özüm ile deęil, özüm kümesi ile alıřır. Olası özümlerden oluřan bu küme iyi özümler ierdięi gibi kötü özümlerde ierebilir.
- Klasik algoritmalar arama sırasında yerel bir en iyi özūme takılabilir, ancak genetik algoritmalar özüm aramasını geniř bir alanda yaptıęı iin yerel özümlere takılmadan genel özümü bulabilir.
- Klasik yöntemlerde adım adım iyi özūme gidilir, ancak Genetik Algoritmalarda bunun bir garantisi yoktur. Yani bařlangıtaki bir nesilde, sonraki nesildeki bireylere göre daha iyi bir özüm bulunmuř olabilir.
- Genetik algoritmalarda problemin matematiksel yapısı ile ilgilenilmez, uygunluk fonksiyonun ve özümün kodalanmıř řeklinin doęru oluřturulması algoritmanın düzgün alıřması iin yeterlidir. eřitli matematiksel iřlemlerden uzak olduęu iin program cevap sūresi dięer yöntemlere göre oldukça kısadır [52].

## 4.2 Genetik Algoritmaların Literatür İncelemesi

Genetik algoritmalar, Charles Darwin tarafından ortaya atılan Evrim Teorisi'nin kuralları taklit edilerek, bunların programlama mantıęına dōnūřtürölmesiyle elde edilmiřtir. Bu mantıęı bilgisayar ortamına ilk kez 1970 yılında Michingan Üniversitesi'nden John Holland [53] adında bir arařtırmacı tařımıřtır ve 1975 yılında bu konuyla ilgili bir kitap yayınlamıřtır. İlerleyen yıllarda bu konu ile ilgili birok makale ve alıřma yayınlanmaya devam etmiřtir. 1985 yılında ise inřaat mühendisi David E. Goldberg [54], Holland'ın doktora öęrencisi olarak tezini genetik

algoritmalar konusunda yapmış ve konunun daha ilgi çeken bir noktaya gelmesini sağlamıştır. Günümüzde birçok alanda genetik algoritmalar kullanılmaya başlanmıştır ve uygulama alanları da genişlemektedir. Elektronik devre tasarımı, iş çizelgeleme, yapay sinir ağı dizaynı, robot dizaynı ve kontrolü, ekonomik modelleme, gezgin satıcı problemi gibi alanlar bunlara örnek verilebilir [55].

Genetik algoritmaların Yapay Sinir Ağları ile birlikte kullanımı da oldukça yaygınlaşmaya başlamıştır, öyleki 1992 yılında ‘Combinations of Genetic Algorithms and Neural Network’ konulu bir çalıştay düzenlenmiştir. Yine aynı yıl Guha ve arkadaşları [56] ‘Genetik Algoritmanın Yapay Sinir Ağı ile Sentezi’ adı altında bir patent almışlardır. Yapay Sinir Ağları ve Genetik Algoritmaların beraber kullanıldığı birkaç çalışma örneği vermek gerekirse, [57]’de bir nükleer santralin giriş değişkenlerinin seçiminde genetik algoritma kullanılmıştır. Bu çalışmada nükleer sistemin çalışmasına etkeyen 22 farklı giriş değişkeni mevcuttur ve farklı durum senaryoları için farklı değişkenler etki etmektedir. Her senaryo için etkiyen giriş değişkenlerinin sayısının bulunması oluşturulacak yapay sinir ağlarının yapısı için oldukça önem arz etmektedir. Ayrıca araştırmacılar yapay sinir ağlarının eğitim sürelerini kısaltabilmek için, sinir ağlarının ağırlık değerlerini genetik algoritma kullanarak optimize etmeye çalışmışlardır [58]. Bazı çalışmalarda ise yapay sinir ağlarını eğitmek için oldukça yaygın olarak kullanılan ‘Back Propagation’ yöntemi yerine genetik algoritmalar kullanılmış ve sonuçlar karşılaştırılmıştır [59, 60].

Bir yapay zeka oluşturma prensibi diyebileceğimiz Bulanık Mantık (Fuzzy Logic) uygulamaları da GA ile kombine edilerek kullanılmıştır. Bulanık mantık kontrolcüsünün kural tabanlarının genetik ile oluşturulduğu ve mobil robota uygulandığı bir çalışma [61] bu alana örnek verilebilir. Robot uygulamalarında genetik algoritmalarla yararlanılarak çalışılan bir başka konu ise manipülatörlerin yol planlaması (path planning) problemleridir [62].

Genetik algoritmalar devre tasarım veya iş çizelgeleme optimizasyonunun yanında nümerik fonksiyon optimizasyonlarında da kullanımı araştırılmıştır [63]. Özel amaçlar için bilgisayar programı geliştirilmesinde de kullanıldığı görülmektedir, örnek olarak veri madenciliği [64] yada sıralama ağlarının geliştirilmesi için yapılan çalışmalar örnek olarak gösterilebilir [65].

Tıp ve biyoloji alanında genetik algoritmaların kullanıldığı görülmektedir [66],[67]. Uydu konumlandırma çalışmalarında da genetik algoritmalar denenmiştir [68].

### **4.3 Genetik Algoritmada Kodlama**

Genetik algoritmaların temel özelliklerden bir tanesi kodlanmış bireyler ile çalışmasıdır. Probleme özgü bilgiler kromozomlarda kodlanır, problemdeki değişkenlerin sıralanmış haline kromozom denebilir. Her bir değişkene de gen adı verilmektedir. Ancak kodlama dediğimiz olayın farklı yapılaş şekilleri vardır; ikili sistemde (binary encoding) yapılabileceği gibi, gerçel sayılarla (floating-point encoding) yada tamsayılarla (integer encoding) veya karakter kodlaması (character encoding) ile yapılabilir.

Yapay sinir ağları ve genetik algoritmalar birarada kullanılırken kodlama için genelde şu üç yöntem kullanılır [69];

- Direkt kodlama, örneğin sinir ağları arasındaki bağlantılar direkt matris olarak kodlanmasıdır. Ancak bu yöntemin dezavantajı, yapıdaki sinir sayısının karesi kadar uzunluğunda kromozomlar oluşturulması gerektirmesidir [70].
- Parametre kodlama, bir parametre listesinin kodlanmasıdır. Bu liste katman sayısı, katmanların büyüklüğü ve katmanların birbirine nasıl bağlandığı gibi parametrelerden oluşabilir. Bu metod ile büyük sinir ağı yapılarını, kısa kromozomlar ile ifade edebiliriz [71].
- Gramatik kodlama, yukarıda bahsedilen direkt kodlamada ki uzun kromozomlar yerine daha kısa kromozom yapılarını oluşturabilmeyi hedeflemiştir. Yani bazı klasik problemlerin çözümü için oluşturulan sinir ağı yapısının bulunmasında genel bir çözüm yöntemi geliştirmeye çalışılmıştır [72].

### **4.4 Genetik Operatörler**

Genetik algoritmaların temel üç tane operatörü vardır; seçim (selection), çaprazlama (crossover), ve mutasyon (mutation).



#### 4.4.1 Seçim operatörü

Bu operatörün amacı, iyi bireylerin uygunlukları ile orantılı olarak bir sonraki nesile aktarılmasını sağlamaktır. Aslında mantıklı bir şekilde bu operatörün anlamı irdelendiğinde Darwin'in evrim teorisine dayandığı görülebilir. Şöyle ki; doğadaki değişen şartlara ve çevre koşullarına adapte olabilen bireyler hayatta kalırken , uyum sağlayamayan bireyler elenirler. Bu operatörün amacı da, genetik programında kullanılan bireylerden uygunluk değeri yüksek olanların hayatta kalmasını sağlayıp, uygunluğu küçük bireylerin yok olmasını sağlayabilmektir.

Bu işlem ile yüksek uygunluk değerinde ki bireyler bir sonraki nesil için kopyalanır. Çözüm aramasının yönlendirilmesinde, global cevabın bulunabilmesinde bu operatör oldukça önemli bir role sahiptir.

Literatürde birçok seçim yöntemi mevcuttur, bunlardan bazıları ve en popülerleri şu şekildedir;

Rulet tekerleği yöntemi (roulette wheel selection): Bu yöntem uygunluk değeri çok az olan bir elemana bile seçilme şansı az da olsa tanımaktadır. Tekerlek isminin nedeni ise, her bireyin uygunluk değerine göre bir daire üzerindeki dilimlere yerleştirilmesidir. Büyük uygunluğa sahip bireylerin dilimleri de büyüktür. Daha sonra sıfır ile toplam uygunluk değeri arasında rastgele bir sayı seçilir ve popülasyondaki ilk kromozomdan başlanarak uygunluk değerleri toplanmaya başlanır. Ne zaman toplanan uygunluk değeri rastgele seçilen değerden büyük olursa, orada durulur ve bulunan kromozom bir sonraki nesle aktarılır. Yani tekerlekte dilimi büyük olanın bir sonraki nesle aktarılma olasılığı yüksektir, ancak küçük dilimli bireylerin de aktarılma olasılığı yok değildir.

Turnuva yöntemi: Bu yöntemde rastgele sayıda bireyler seçilir ve bu bireyler arasında bir turnuva yapılır. En iyi uygunluğa sahip birey bulunur ve bu işlem iki kere tekrar edildiğinde çaprazlama için ebeveynler elde edilmiş olunur. Ebeveynlerden yola çıkılarak çaprazlama ve mutasyon işlemlerinden geçirilen bireyler bir sonraki nesle aktarılır. Turnuva yönteminin avantajı kötü bireylerin asla seçilemeyecek olmalarıdır, ayrıca iyi bireylerin kendi arasında kısır döngüye girmesi ise diğer genetik operatörler ile önlenmektedir.

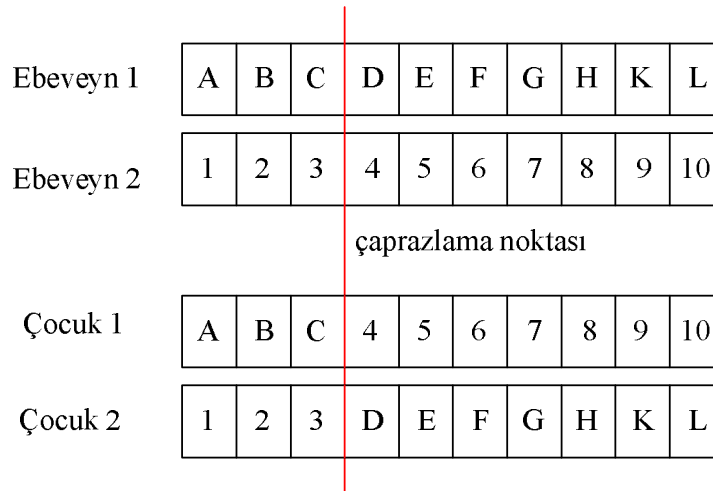
Elitizm yöntemi: Bu yöntemde en iyi kromozom yada birkaç en iyi kromozom uygunluk değerine bakılarak bir sonraki nesle aktarılır. Yeni popülasyonun geri kalan bireyleri ise klasik yöntemler ile oluşturulur.

#### 4.4.2 Çaprazlama (cross-over) operatörü

Popülasyonu oluşturan iki birey üzerinde yapılan bu işlem, neslin çeşitliliğini sağlamak ve tek tip gen yapısına sıkışıp kalmasını engellemeyi amaçlamaktadır. Her zaman iyi sonuç vermeyi garanti etmemekle birlikte, verilen kriterlere göre en mükemmel sonuç bulunmaya çalışılır. Genetik algoritmalar da uygunluk değerine göre seçilen en iyi bireyler arasında yapılan bu işlem, ebeveynlerinden daha iyi özellikte ki bireyleri bulmayı amaçlar.

İkili veya tamsayı kodlama şekilleri için farklı türde çaprazlama operatörleri mevcuttur. İkili sisteme göre kodlanmış bireyler için uygulanabilecek çaprazlama türleri;

- Tek noktalı çaprazlama, şekil 4.1’de gösterilmiştir.
- Çift noktalı çaprazlama, aşağıdaki şekilden farklı olarak çaprazlama noktasının birden fazla noktadan oluşmasıdır.
- Çok noktalı çaprazlama, bu tür çaprazlama da çaprazlama noktası ikiden fazladır.



**Şekil 4.1** : Tek noktalı çaprazlama

Tamsayı şeklinde kodlanmış bireyler için kullanılabilen çaprazlama türleri;

- Sıralı çaprazlama (Order crossover-OX)

- Sıralamaya dayalı çaprazlama (Order based crossover-OBX)
- Kısmi eşleştirmeli çaprazlama (Partially matched crossover-PMX)
- Devirli çaprazlama (Cycle crossover-CX)
- Düzenli çaprazlama (Uniform crossover-UX) [73]

#### 4.4.3 Mutasyon operatörü

Doğal yaşamda görülen kromozomların yapısının ve hatta kromozomdaki gen sayısının değişmesini, yapay sistemlere uygulanmasını sağlayan operatördür. Ancak yapay sistemlerdeki mutasyonlarda gen sayısı değişmez, ikili kodlanmış bir bireyde ki rastgele belirlenecek herhangi bir gen '1' iken '0' ya da '0' iken '1' yapılır. Tamsayı kodlamalar için kullanılan diğer bir mutasyon yöntemi ise kromozomda ki herhangi iki genin yer değiştirmesi olayıdır. Nasıl uygulanır ise uygulansın mutasyon oranının çok fazla olması, çaprazlama işleminden sonra uygulandığı için bu işlemde elde edilmiş iyi bireylerin kaybına yol açabilmektedir. Bu nedenle genelde düşük mutasyon oranları tercih edilmektedir. Şekil 4.2'de ikili ve Şekil 4.3'de tamsayı olarak kodlanmış bireylere uygulanan mutasyon işlemleri görülmektedir.

Ebeveyn	1	2	3	4	5	6	7	8	9	10
Çocuk	1	2	8	4	5	6	7	3	9	10

**Şekil 4.2 :** Tamsayı kodlama için mutasyon örneği

Ebeveyn	1	0	0	0	1	1	0	1	0	1
Çocuk	1	0	0	0	0	1	0	1	0	1

**Şekil 4.3 :** İkili kodlama için mutasyon örneği

Bu işlemin asıl amacı; çözüm uzayı içerisinde, kromozomlarda küçük değişiklikler yaparak, küçük adımlarla arama yapmaktır. Yani arama uzayında çözüm olarak sinanmamış noktaların bulunmasında diğer operatörlere yardım etmektedir [74].



## 5. GENETİK ALGORİTMALAR İLE YAPAY SİNİR AĞLARININ BİRLİKTE KULLANIMI

Doğadan ilham alarak çalışan ancak tamamen farklı iki teknik olan Genetik Algoritmalar (GA) ve Yapay Sinir Ağları (YSA) yöntemleri son yıllarda bilgisayar bilimi ve mühendislik ile ilgilenenlerin oldukça dikkatini çekmiştir. Bu konularda yapılan çalışmalar ve kullanılan alanlar günümüzde oldukça geniş bir alanı kapsamaktadır. Ancak son zamanlarda yapılan çalışmalara bakılırsa, aynı yerden ilham alan ancak tamamen farklı çalışan bu tekniklerin, bir arada kullanılmasının da literatürde çok yaygın olduğu görülmektedir. Böylece daha fazla sorunun çözülmesi ve problemlerin çözüm sürelerinin düşürülmesi mümkün olmuştur. Bu iki tekniğin birlikte kullanılmasını;

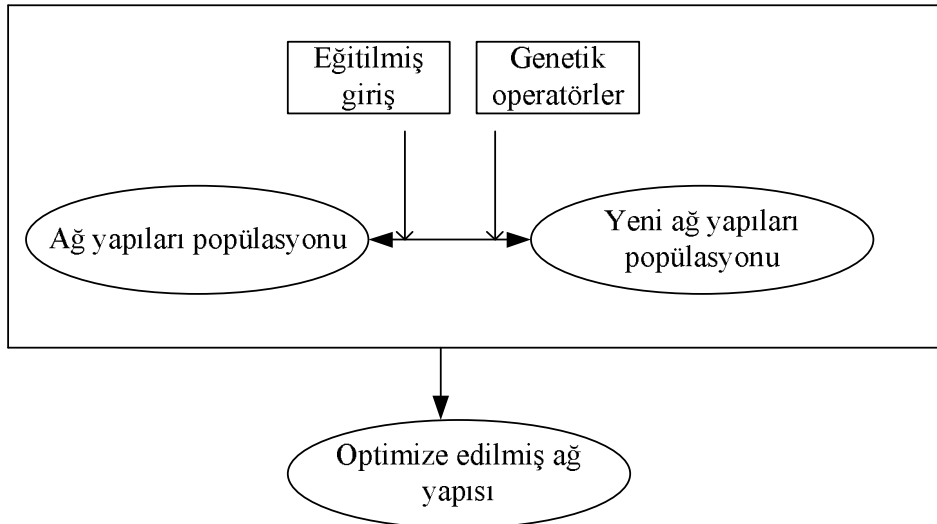
1. Destekleyici Kombinasyonlar,
2. İşbirliği Kombinasyonları

olarak ikiye ayırabiliriz [75].

Bazı araştırmacılar genetik algoritmaları, yapay sinir ağına yardımcı olarak kullanırken bazıları ise tam tersi yapay sinir ağlarını genetik algoritmalara yardımcı olarak kullanmışlardır. Bu konuda yapılan yayınlara bakılarak, genetik algoritmaların en fazla yapay sinir ağlarının mimarisinin ve topolojisinin belirlenmesinde kullanıldığı söylenebilir. Mimariden kasıt, yapay sinir ağlarındaki katman sayıları, her bir katmandaki sinir hücresi sayısı ve bu hücrelerin birbirine nasıl bağlandığı konularıdır. Bunlarla beraber birçok çalışmada sinir hücrelerinin optimum ağırlık değerleri de GA ile bulunmuştur. Tasarlanan sinir ağı yapısı, giriş datalarını öğrenebilmeli ve eğitim aşmasından sonra giriş dataları arasında olmayan bilgiler için de genelleme yapabilmelidir. Ancak buradaki önemli nokta belirtilen istekleri yerine getirirken minimum katman ve sinir sayısı niteliğine bağlı kalmalıdır.

## 5.1 GA ve YSA' nın Birlikte Kullanımının Avantajları

Peki yapay sinir ağlarının mimarisinin genetik ile belirlenmesi bize ne gibi avantajlar sağlamaktadır? Yapılan genellemelerde hata oranının düşmesi, ağın öğrenme süresinin kısılması, katman sayısının ve sinir sayısının yeterli minimuma indirilmesi gibi avantajlar söylenebilir. Eğer yapıdaki sinir ağı ve gizli katman sayısı fazla olursa ağ doğru genellemeler yapabilir ancak eğitim süresi oldukça uzun sürer ve büyük yapılarda maliyet fazladır. Tam aksini düşündüğümüzde, ağ yapısı az sinir hücresi ve yetersiz katmandan oluşmuşsa bu kez de ağın öğrenmesi kafi gelmemekte ve genelleme yaparken hata oranı yüksek çıkabilmektedir. Kısaca anlatılan bu nedenlerden dolayı yapay sinir ağlarında, sistemin verimliliği ve doğruluğu açısından ağın mimarisi oldukça önemli bir yer tutmaktadır. Şekil 5.1'de GA kullanılarak YSA yapısının belirlenmesinde izlenen örnek bir işlem şeması gösterilmiştir [76].



**Şekil 5.1** : GA ile YSA yapısının dizayn edilmesi

Aslında ağın yapısını oluşturan gizli katmanların sayısını belirlemede genetik algoritmalar dışında kullanılan başka yöntemlerden de söz edilmektedir. Aşağıdaki tabloda bu yöntemlerin sınıflandırılması bulunmaktadır [77].

Çizelge 5.1'de bahsedilen yapıcı metod ile kastedilen az sayıda sinir ağı ile başlayıp sayıyı artırarak performansı gözleyip optimum sayıyı bulma yöntemidir. Yıkıcı metod ile tabir edilen metod ise tam tersine çok sayıda sinir hücresi ile başlayıp performans gözlenerek sayının azaltılmasıdır.

### Çizelge 5.1 : Gizli katmandaki sinir sayısını belirleme metodları.

(Kaynak [72]'den uyarlanmıştır)

Gizli katman sayısı belirleme metodları		Metodun türü
Deneysel metodlar		Direkt
İstatiksel kriterlere dayanan metodlar		Dolaylı
Ontogenetik metodlar	Yapıcı	Direkt
	Yıkıcı	Direkt
	Karışık	Direkt
	Genetik algoritmalara dayanan	Direkt

### 5.2 GA ile YSA'nın Birlikte Kullanıldığı Çalışmalar

Yarı iletken malzeme üretiminde geleneksel metodların yanı sıra yapay sinir ağları yöntemi de kullanılmış ve doğruluk ile genelleme açısından diğerlerine üstünlük sağladığı belirlenmiştir. Ancak yapay sinir ağı modelini geliştirirken öğrenme katsayısı, momentum faktörü, eğitim toleransı ve gizli katmanlarda ki sinir sayıları gibi parametrelerin optimum değerlerinin bulunması gerekmektedir. Silikon dioksit filmin yapım aşamasında da kullanılan yapay sinir ağı modeli, genetik ile optimize edilmiştir. Nitekim genetik ile optimize edilen yapay sinir ağları modelinde eğitim hatasının %10, tahmin hatasının ise %66 oranında düştüğü ifade edilmektedir [78]. Yine başka bir yarı iletken olan çinko oksit de son yıllarda ucuzluğu, sağlığa zararlı olmayışı nedeni ile diğer maddelere alternatif olmuş ve çokca çalışılmaya başlanmıştır. Çinko oksit film üretiminde de sinir ağları ve genetik algoritmalar birarada kullanılmıştır [79].

Bu konu başlığına örnek verilebilecek bir başka alan ise, Bulanık Kontrolör (Fuzzy Logic Controller) tasarımında, yapay sinir ağlarının ve genetik algoritmaların beraber kullanılmasıdır. Örneğin Thakur ve arkadaşlarının yaptığı çalışmada, araç merkezleme probleminde kullanılacak optimum bulanık mantık kontrolcüsünün dizaynı yapılmıştır [80]. Yapılan araştırmada, yapay sinir ağlarının MATLAB Toolbox'dan eğitilmesi durumundaki sonuçlar ile MATLAB de yazılan genetik algoritmalar ile eğitilmesi sonucunda çıkan sonuçlar karşılaştırılmıştır. Görülmüştür ki, genetik algoritmalar ile yapılan eğitimde daha az döngü yapılmış yapılmış ve hata

oranı da daha düşmüştür. Endüstride de yapay sinir ağlarının ve genetik algoritmaların kombine edilerek bazı problemlere çözüm bulunduğu görülmektedir. Sanayide kullanılan transformatörlerde ki demir harcamasını optimum hale getirmek için iki yöntem beraber kullanılmış ve transformatör üretim maliyeti düşürülmüştür [81]. Yine yüksek hızlı ağlarda ki aktif sıra yönetimi algoritmalarının geliştirilmesinde yapay sinir ağı ve genetik algoritmalara dayanan yöntemlerin kullanıldığı görülebilir, aynı çalışma da akıllı PID diye tabir edilen kontrolcünün katsayıları da sinir ağları-genetik algoritma ortak çalışmasıyla bulunmuştur [82].

Genetik algoritma uygulanan yapay sinir ağları, uçak motorlarında ki aşınma tahmini için de kullanılmış ve ağırlık bilgileri genetik ile hesaplanmış sinir ağı modelinin diğer yöntemlere göre üstünlük sağladığı ispatlanmıştır [83]. Bir başka çalışma alanı olan lojistik alanında da bu iki yöntemin birlikte kullanıldığı ve başarılı sonuçlar elde edildiği söylenebilir. Li ve arkadaşları [84] lojistik tahmin alanında genetik algoritma kullanılmış ve kullanılmamış iki ağ yapısı üzerine çalışmış ve sonuçta genetik optimizasyonu yapılmış ağın daha doğru sonuçlar verdiğini kanıtlamışlardır.



## 6. SİMÜLASYONLAR ve SONUÇLARI

Bu tez çalışmasında; Stewart Platformu'nun ters ve düz kinematik analizleri, yapay sinir ağları ve genetik algoritmalar üzerine detaylı olarak çalışılmış ve 6x3'lük Stewart Platformu'nun çalışma uzayı analizini yapacak yapay sinir ağı yapısı genetik algoritmalar ile belirlenmiş ve bu ağı eğitimi yapılmıştır. Uygulanan bu yöntem ile yapay sinir ağlarının eğitiminde ve yapısının belirlenmesinde ki zaman kaybını engellemek ve tek bir program ile probleme en uygun sinir ağının ağırlıkları, bias değerleri, gizli katman ve nöron sayıları dahil bulunması amaçlanmaktadır.

İlk olarak yazılan genetik algoritma programının doğru çalıştığını ispatlamak için genetik algoritma yöntemiyle ile çözülen çok genel bir problem olan 'Gezgin Satıcı Problemi (Traveling Salesman Problem)' çözümlenmiştir.

Tasarlanan genetik program mantığının doğru çalıştığı GSP (Gezgin Satıcı Problemi) ile görüldükten sonra, tezin asıl amacı olan kısma geçilmiştir. Bu kısımda, giriş datalarını SPM'nin üst platformunun konumlarının oluşturduğu, çıkışta ise bu konumların çalışma uzayı içerisinde yada dışarısında olduğu bilgisini veren optimum yapay sinir ağı yapısının bulunması amaçlanmıştır. Yani optimum tasarımının, ağırlık ve bias değerlerinin bulunması amaçlanan bu ağı giriş ve çıkışında ki nöron sayıları sırasıyla 3 ve 2 dir. Çünkü 6x3'lük SPM'nin üst platform köşe noktalarından hesaplanan üst platformun merkezinin x,y ve z eksenindeki değerleri sinir ağına giriş olarak verilmekte, çıkışta ise her verilen üçerli datalara göre üst platformun çalışma uzayı içinde yada dışında olduğu bilgisi tutulmaktadır. Bu sistemin sadece klasik YSA yöntemi ile çözümlenmesi daha önceden yapılmış bir çalışmadır [46]. Ancak bizim burada getirmek istediğimiz yenilik ise deneme yanılma yoluyla bulunan YSA gizli katmanlarındaki nöron sayılarının, GA ile en iyisinin bulunmaya çalışılmasıdır. Aynı zamanda ağırlık ve bias değerleri de GA ile üretilerek ve sonradan yazılan algoritma içerisinde iyileştirilmesi yapılarak, sinir ağı yapısına en uygun hale getirilmektedir. Yani buarada yapılan çalışmada çözümlenmesi istenen probleme göre giriş ve çıkış nöron sayısı belirlenen bir sinir ağı yapısının gizli katmanlardaki sinir sayılarının belirlenmesi ve bu yapılara en uygun ağırlık, bias değerlerinin

hesaplanması işlemleri gerçekleştirilmiştir. Böylece giriş ve çıkış bilgilerini belirlediğimiz bir problemi YSA ile çözmek istediğimizde tek yapılması gereken programın gerekli yerlerinde istenen giriş ve çıkış dosyalarını programa vermektir. Oluşturulan algoritma ile çözmek istediğimiz probleme en uygun sinir ağı yapısı ve bununla birlikte ağdaki nöronların ağırlık ve bias değerleri sonuç olarak verilecektir.

## **6.1 Genetik Algoritma Uygulaması Örneği Olarak Gezgin Satıcı Probleminin Çözümlemesi**

### **6.1.1 Problemin tanımı**

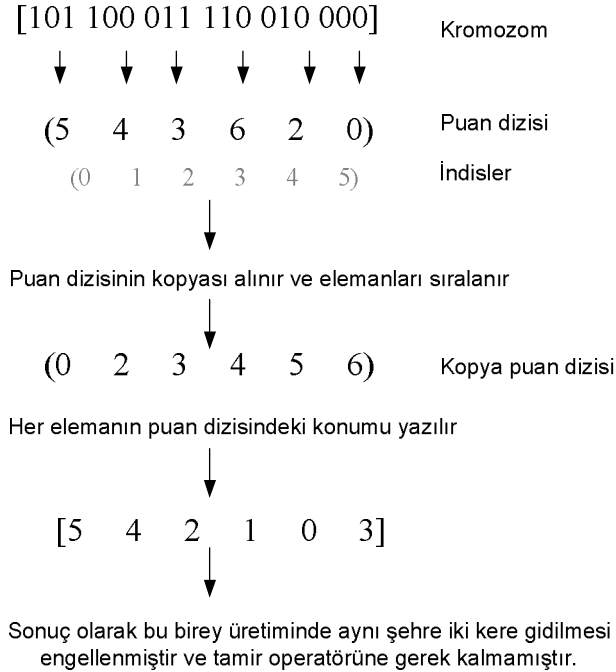
Genel anlamda elinde N tane şehrin yazıldığı bir liste olan satıcının, bu listedeki tüm şehirlere uğramak kaydıyla başladığı şehre en kısa yoldan dönmesi problemidir. Gezgin satıcı probleminin çözümü aslında geniş bir konuya hitap etmektedir. Çünkü birçok farklı problemin çözümünde bu metod kullanılmaktadır. Bunlara birkaç örnek vermek gerekirse; araç rotalama problemlerinde [85], bir makine için iş sıralaması yapılmasında [86], bir depodaki malzemelerin taşınması probleminde [87], kristal yapılarının analizinde [88], diziler halindeki verilerin sınıflandırılmasında kullanılmaktadır [89].

1985 yılında Brady'nin [90] genetik algoritmaları gezgin satıcı problemi üzerinde ilk kez uygulamasıyla, bu konudaki çalışmalar artmaya başlamıştır.

### **6.1.2 Bireylerin kodlanması ve kullanılan yöntem**

Bu çalışmada GSP için çok genel bir kodlama yöntemi olan rota genetik gösterimi kullanılmıştır. Bu gösterimde N tane şehri yada düğüm noktası olan bir problem için bir birey (kromozom) N tane genden oluşmaktadır. Ancak bu tarz gösterimde bireyler oluşturulurken aynı genden, aynı birey içinde birkaç tane olması durumu söz konusu olabilmektedir. Bu istenmeyen ve düzeltilmesi gereken bir durumdur. Tamir fonksiyonları adı verilen alt programlar bu şekildeki sorunların çözülmesi için kullanılan bir yöntemdir. Ancak bu çalışmada bir tamir fonksiyonu yerine, araştırmacılardan Aytekin'in [91] yaptığı çalışma baz alınarak her şehre karşılık ikili permütasyon bir değer atamak yerine, bireyi oluşturan her bir gene bir puan değeri atanmıştır. Kromozomu oluşturan her genin kromozom içerisinde bir değeri olmakta ve daha sonra bu değerler belirli bir kurala göre sıralanmaktadır. Böylece düzgün bir tur anlamına gelen doğru bireyler oluşturulabilir.

Örneğin 6 şehirli bir GSP problemini ele alalım, her bir şehir de 3 bit ile ifade edilebileceği için (8 şehirli bir problem olsaydı 4 bit ile ifade etmek gerekcekti), bu problemde 18 (6x3) genli kromozomlar üretilmesi gerekmektedir. Rastgele [101 100 011 110 010 000] bireyinin üretildiğini varsayalım. Öncelikle bu bireydeki her üç bitin onluk sistemde karşılık geldiği değerler hesaplanır ve bunlara puan adı verilir. Her üç bite karşılık gelen bu puanlar bir diziye atılır ve daha sonra bu puan dizisi kopyalanarak, elemanları küçükten büyüğe doğru sıralanır. Bu sıralanmış kopya dizisindeki her elemana karşılık gelen, puan dizisindeki indisleri yazılır. Mesela kopya puan dizisindeki ilk eleman 0' ın, puan dizisindeki indisi (konumu) 5'e denk gelmektedir. Yani sırasıyla bütün elemanların konumları yazılarak sağlıklı bir tur oluşturulmaktadır. Bu indislerin oluşturduğu dizi artık şehirleri temsil etmektedir ve bu yöntemle hatalı yada eksik bir birey üretme yada çaprazlama, mutasyon gibi işlemlerden sonra hatalı birey oluşturma sorunu ortadan kalkmaktadır. Bu kodlama mantığının aşamaları Şekil 6.1'de açıklanmıştır.



**Şekil 6.1 :** Bireylerin oluşturulma mantığı (kaynak [91]'dan uyarlanmıştır)

Kullanılan bu yöntemin en büyük avantajı tamir fonksiyonuna, yada hatalı birey elde etmemek için oluşturulması gerekli olabilecek özel çaprazlama ve mutasyon

operatörlerine ihtiyaç duyulmamasıdır. Ancak bu yöntemin kullanıldığı problemde ki şehir sayısı arttıkça kromozomun gen sayısı üssel olarak artmaktadır ve buda hesaplama süresi açısından bir dezavantaj yaratabilmektedir.

Peki ilk oluşturulan kromozomda birden fazla onluk sistemde aynı sayıya denk gelen üçerli bitler var ise, puan dizisinde ki değerleri nasıl sıralanacak?

Burada da şu şekilde bir yöntem geliştirilmiştir; konum indisi küçük olan gen, son üretilen bireyde öncelikli yazılır. Örneğin [111 000 101 111 110 001] kromozomunu ele alırsak, burada görüldüğü gibi onluk sistemdeki değeri 7 ye karşılık gelen iki tane birey vardır. Bu durumda kopyalanmış ve sıralanmış puan dizisi şu şekilde olur,

[0 1 5 6 7 7 ]. Bu dizinin konum indislerine bakılarak oluşturulan son kromozom ise budur; [1 5 2 4 0 3]. Dikkat edilirse 0. indisli elemanın konum değeri, diğer aynı puana sahip elemanın konum değerinden önce yazılmıştır.

### **6.1.3 Gezin satıcı probleminde genetik algoritma uygulaması**

Bu tez çalışmasında asıl yapılmak istenen, paralel mekanizmaların çalışma uzayı analizinde uygulanan yapay sinir ağı modeli yapısının genetik algoritmalar desteği ile belirlenmesi ve eğitilmesidir, çalışmaya ön hazırlık sağlaması açısından tez kapsamında 11 şehirden oluşan GSP çözümlenmiştir.

Yukarıda anlatılan kodlama tekniği kullanıldığı için her şehrin ikilik sistemde karşılığı dört bit ile ifade edilmiştir, böylece her birey 44(11x4) genden oluşmaktadır. Başlangıç popülasyonunun ise 50 adet bireyden meydana gelmesi öngörülmüştür.

Yazılan program algoritma olarak şu adımları takip etmektedir:

Adım1: [Popülasyonu oluşturma] Her biri 44 genden oluşan, 50 adet bireyin rastgele olarak üretilmesidir.

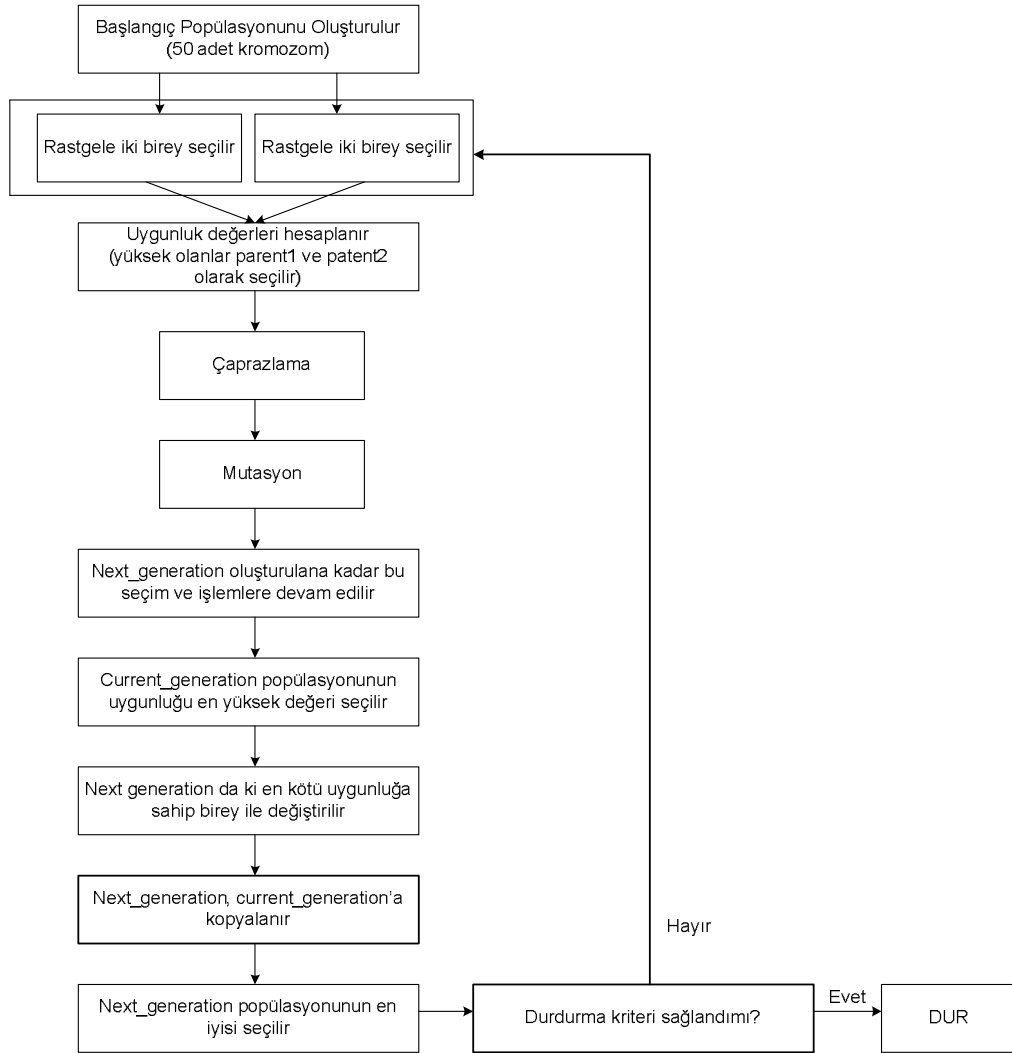
Adım 2 : [İkişerli seçim] Popülasyondaki rastgele iki bireyin seçilip, uygunluk değeri yüksek olanın tutulması, ve bu işlemin iki kere tekrar edilmesi sonucu iki tane kısmen seçilmiş ebeveyn bireyin elde edilmesidir.

Adım 3: [Çaprazlama] Seçilen ikişerli bireyler arasında, belli bir oran ile rastgele seçilen tek noktadan çaprazlama işleminin yapılmasıdır.

Adım 4: [Mutasyon] Seçilen ikişerli bireylerin %1 ihtimalle mutasyona uğraması sağlanmaktadır. Yani bireyin rastgele seçilen biti 1 ise 0, 0 ise 1 yapılmaktadır.

Adım 5: [Yeni popülasyonun oluşturulması] Seçilen ebeveynlerin çaprazlama ve mutasyon adımlarından geçtikten sonra, yeni nesle kopyalanmasıdır. Denk gelmeyen ve çaprazlamaya yada mutasyona uğramayan birey de aynen sonraki nesle kopyalanır.

Adım 6: [En iyi bireyin aktarılması] Mevcut popülasyondaki uygunluğu en yüksek birey ile yeni popülasyondaki uygunluğu en kötü birey belirlenir ve en iyi birey yeni jenerasyonda en kötü bireyin yerine yazılır. Böylece geçmiş nesilde ki en iyi birey kaybedilmemiş olur.



**Şekil 6.2 :** GSP çözümünde kullanılan GA programının çalışma şekli.

Adım 7: [Yeni neslin mevcut nesil yapılması] Henüz programın durdurma kriteri sağlanamamış ise yukarıda anlatıldığı gibi oluşturulan yeni nesil, aynı adımlardan geçirilmek için mevcut popülasyona kopyalanır.

Adım 8: [Durdurma] Programda belirlenen kriter sağlandıysa mevcut en iyi birey sonuç olarak döner. Bizim programımızda iterasyon sayısı durdurma kriteri olarak kullanılmıştır. Eğer üst üste 10000 iterasyonda aynı bireyi en iyi olarak döndürüyorsa, program durur ve en iyi birey sonuç olarak kabul edilir.

Geliştirilen programın çalışma mantığı aşağıdaki Şekil 6.2’de görsel olarak anlatılmıştır.

#### **6.1.4 GSP sonuçları**

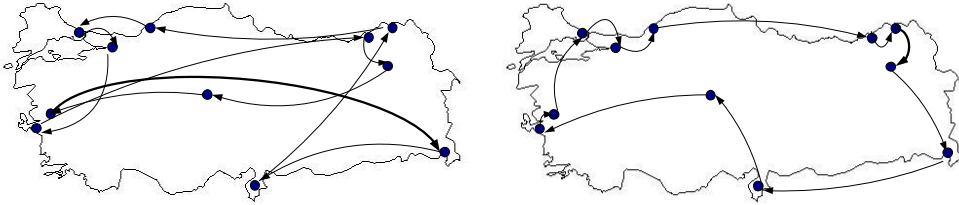
C programlama dili kullanılarak yukarıda açıklanan mantığa ve kodlama yöntemlerine uyularak yazılan programın örnek bir çözümü açıklanacaktır.

Program çalıştırıldığında başlangıç popülasyonundaki en iyi birey ekrana yazdırılır. Daha sonra en iyi bireyi tuttuğumuz değişkenin değeri her değiştiğinde yani her yeni en iyi hesaplandığında ekrana yazdırılır. Aynı zamanda en iyi bireylerin önerdikleri turlarda katedilen toplam mesafeler bireylerin yanında gösterilmektedir.

Saticımızın rastgele seçilen bir numaralı ilden başlayarak 11 ili gezip tekrar başladığı yere dönmesini hesaplayacak programımız çalıştırılmış ve ilk popülasyondaki en iyi birey ekrana, [1100 1101 0101 1001 1001 0001 0100 1111 0001 1010 0001] olarak yazılmıştır. Bu kromozomun verdiği tur izlenirse katedilecek toplam yolun 9314 km olduğu hesaplanmış ve bu bilgide ekrana yazdırılmıştır. Bu hesaplama göre çizilen rota görsel açıdan Şekil 6.3a’da verilmiştir. Rastgele oluşturulan bu kromozomda görüldüğü gibi üçer ve ikişer olmak üzere aynı dörtlü gen topluluğundan bulunmaktadır. Bu hata yukarıda anlatılan kodlama algoritması ile düzeltilmiştir. Ayrıca şekle bakınca bu kromozomun bize önerdiği turun hiçte mantıklı olmadığını anlayabiliriz, bu yüzden programda en kısa mesafeyi veren ve tüm şehirlerden geçtiği kontrol edilen kromozom bulunmaya çalışılmıştır. Yaklaşık 16760 iterasyondan sonra, Şekil 6.3b’deki görseli veren, [0010 1001 1100 1100 1001 0111 1101 1011 1000 1110 1110] kromozomu elde edilmiştir. Optimum yol olarak bulunan bu turdaki toplam katedilen yol 5244 km olarak hesaplanmıştır. Tezde yapılacak asıl amaca ön hazırlık sağlanması ve örnek bir GA kodunun çalışma

mantığının anlaşılması için genetik algoritmaların çok yaygın olarak kullanıldığı genel bir problem olan GSP için bir çözüm programı yazılmıştır.

Bu programda amaç çok kapsamlı bir GSP çözümü yapmak olmadığı için, 11 ili kapsayan bir tur optimizasyonu yapılmıştır.



(a)

(b)

**Şekil 6.3 :** (a)Rastgele hesaplanan tur ve (b) Optimizasyonu yapılmış tur örneği.

Ayrıca bu programda ilk başlanılacak il rastgele seçilmektedir, ancak istendiği takdirde ilk başlangıç ilinin hep aynı verilmesi de küçük bir düzenleme ile mümkündür. Çıkan sonuçlardan da görülmüştür ki, tasarlanan algoritma çalışmakta ve doğru sonucu bulmaktadır. Programda durdurma kriteri olarak; en iyi değer 10000 iterasyon boyunca aynı bulunması kullanılmıştır. Yani her jenerasyonda birtane en iyi bulmaktadır, eğer bu birey 10000 iterasyon boyunca hep en iyi kalabilirse bizim çözümümüz olarak kabul edilmekte ve sonuç olarak yazdırılmaktadır.

## **6.2 6x3 SPM'nin Çalışma Uzayı Analizi için Kullanılacak YSA Yapısının**

### **Genetik Algoritmalar ile Belirlenmesi**

Genetik Algoritma programı oluşturulurken şu şekilde bir işlem sırası ile ilerlenmiştir;

- 1- Kromozom yapısının ve kodlama şeklinin belirlenmesi
- 2- Nesil büyüklüğünün belirlenmesi
- 3- Uygunluk değeri hesaplama yolunun belirlenmesi
- 4- Seçim yapılırken izlenecek yöntemin belirlenmesi
- 5- Yeni nesil üretiminin ve kullanılacak operatörlerin şekillendirilmesi.

Kısaca maddeler halinde bahsedilen basamakların ayrıntıları ve bizim problemimize göre özelleştirilmiş halleri ileriki sayfalarda açıklanmıştır.

### 6.2.1 Kromozom yapısı ve kodlanması

Programda kullanılan ve birey olarak adlandırabileceğimiz her kromozom 83 genden oluşmaktadır. Gen denilen birimler; gizli katmandaki sinir sayılarını, sınırlar arasındaki her bağlantının ağırlık bilgilerini ve sınırlardaki bias değerleri bilgilerini tutmaktadırlar.

Genetik algoritmalarda iki şekilde kromozomların kodlanması mümkündür. Bunlar ikilik sistemdeki sayılarla ve gerçek sayılarla kodlama olarak ayrılmaktadır. Bizim kromozomlarımızda kodladığımız ağırlık ve bias değerleri sadece tamsayılardan oluşmadığı için ve de programa fazladan işlem yükü getireceği için ikilik kodlama tercih edilmemiştir. Kromozomlarımız gerçek sayılar ile kodlanmaktadır. Oluşturulan kromozomun ilk iki geni (0 ve 1 indisli genleri) sırasıyla ilk gizli katmandaki ve ikinci gizli katmandaki sinir sayılarının bilgisini tutmaktadır ve 1 ile 6 arasında rastgele değerler atanarak oluşturulmaktadır. Kromozomun üçüncü (yani 2 indisli) geninden yirminci genine kadar olan kısımda, giriş katmanı ile ilk gizli katman arasındaki ağırlık bilgileri tutulmaktadır. Bu gen aralığının hesabı ise şu şekilde yapılmaktadır;

x: input sayısı, 3

y: ilk gizli katmandaki sinir sayısı, maksimum 6

birinci ağırlık değerleri için kullanılan gen sayısı maksimum:  $(3 \times 6) = 18$  olabilir.

Daha sonra yapılan kodlama ile ağırlık değerleri için ayrılan genlerin kaçının kullanılacağı kaçına sıfır atanacağı ilk gizli katmandaki sinir sayısına göre belirlenmektedir. Örneğin girişimizin 3 olduğu bilinmekte ve ilk gizli katmandaki sinir sayımızın 5 olarak atandığını varsayalım, bizim giriş ile ilk gizli katmandaki ağırlık değerlerimiz 3 sütunlu 5 satırlı bir matrisde tutulmalıdır. Yani birinci ağırlık değerleri için ayrılan 18 genlik alanın 15 genlik kısmına rastgele ağırlık değerleri yerleştirilmekte geri kalan 3 genlik kısma ise 0 yerleştirilmektedir.

Her kromozomun 21. ve 56. genleri ise ikinci ağırlık değerleri için ayrılmıştır. Yine yukarıda anlatılan mantık ile bu genlerin kullanım oranları ilk gizli katmandaki sinir sayısına ve ikinci gizli katmandaki sinir sayısına bağlı olarak değişmektedir.



Kromozom kodlamasındaki 57. ve 68. genler arası üçüncü ağırlık değerlerine ayrılmıştır ve doluluk oranları ikinci gizli katman ile çıkış sinir sayısına bağlıdır. Ağırlıkların kodlanmasının ardından, sıra bias bilgilerinin kromozomlarda tutulmasına gelmiştir. Birinci bias değerleri 69. – 74. genler arasında, ikinci bias değerleri 75. – 80. genler arasında ve üçüncü bias değerleri 81 ve 82 numaralı genlerde tutulmaktadır. Son bias değerinin sayısı çıkış kamanına bağlı olduğu için ve bizimde çıkışımız sabit ve 2 olduğu için bu şekilde kromoza kodlanmıştır. Tüm bu YSA'nın yapısal bilgilerinin kodlanması bittikten sonra, kromozomda son bir gen de o bireyin uygunluk değerinin tutulması için oluşturulmuştur. Kullanılan kromozom yapısı Şekil 6.4'de verilmiştir.

H1	H2	W1	W2	W3	B1	B2	B3	U
0	1	2...19	20...55	56...67	68...73	74...79	80,81	82

**Şekil 6.4 :** Kromozomların kodlanma şekli.

H1: ilk gizli katmandaki sinir sayısının tutulduğu gen

H2: ikinci gizli katmandaki sinir sayısının tutulduğu gen

W1: giriş ve birinci gizli katman arasındaki ağırlık değerlerinin tutulduğu gen dizisi

W2: birinci ve ikinci gizli katman arasındaki ağırlık değerlerinin tutulduğu gen dizisi

W3: ikinci gizli katman ve çıkış katmanı arasındaki ağırlık değerlerinin tutulduğu gen dizisi

B1: birinci gizli katmandaki her sinirin bias değerlerinin tutulduğu gen dizisi

B2: ikinci gizli katmandaki her sinirin bias değerlerinin tutulduğu gen dizisi

B3: çıkış katmandaki her sinirin bias değerlerinin tutulduğu gen dizisi

U: kromozomun uygunluk değerinin tutulduğu gen.

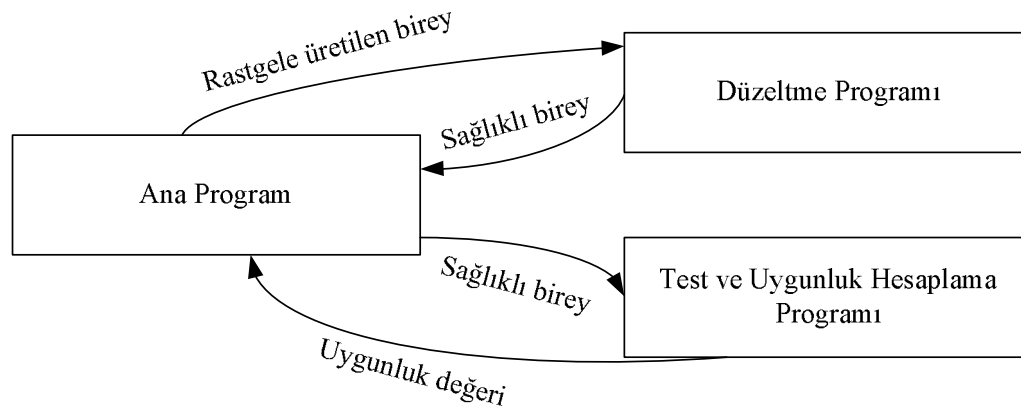
### 6.2.2 Nesil büyüklüğünün belirlenmesi

Algoritmada ilk oluşturulan popülasyonu ve daha sonra üzerinde işlem yapılacak popülasyonları oluşturan birey sayıları oldukça önemlidir. Çünkü neslimizi oluşturan birey sayısını az seçersek, topluluğumuzu oluşturan gen çeşitliliği az olacağı için programımız global minimumlar yerine yerel minimumlarda takılabilir. Birey sayımız çok fazla olduğunda ise işlem süresi ve yükü artmaktadır. Bu nedenle süre

kısıtı göz önüne alınarak, mümkün olan en büyük nesil sayısı seçilmelidir. Deneme yanılma ile bir nesildeki uygun birey sayısı belirlenebilir. Bizim algoritmamızda popülasyon büyüklüğü 10 olarak alınmıştır ve başlangıçta yukarıda anlatılan şekilde kromozomlara rastgele değerler verilerek oluşturulmuştur.

### 6.2.3 Uygunluk değerinin hesaplanması

Bu kısımda popülasyonumuzu oluşturan bireylerin bizim bulmak istediğimiz çözüme ne kadar yakın olduğu hesaplanır. Uygunluk değeri yüksek olan bireyin çözüm değerine daha yakın olması nedeniyle bir sonraki nesle aktarılma olasılığı daha yüksektir. Bu çalışmada 6x3'lük SPM'nin çalışma uzayı analizini yapan YSA yapısının belirlenmesi, ve her rastgele gelen YSA yapısı için de gerekli ağırlık ve bias değerlerinin hesaplanması amaçlanmıştır. Bu doğrultuda GA'da üretilen her kromozom için başka bir alt foksiyonda yapay sinir ağları eğitilir ve kromozomdaki ağırlık ile bias değerlerinin, kromzomun ilk iki geninde verilen yapıya uygun olarak hesaplanması sağlanır. Burada amaç rastgele oluşturulan bireylerin daha sağlıklı bireyler haline getirilmesidir. Daha sağlıklı hale getirilen bireyler, başka bir alt programa gönderilir ve burada taşıdıkları yapının performansına göre uygunluk değeri alırlar. Şöyleki bir bireye kodlanmış YSA yapısının verilen bazı datalara göre ne kadar doğru sonuç verdiği bakılmış yani giriş datalarının ne kadarının çalışma uzayı içinde veya dışında olduğunu doğru bildiği hesaplanmıştır. Doğru bilinen nokta sayısı bizim uygunluk değerimizi ifade etmektedir ve bu değer kromozomumuzun son genine yazılmaktadır.



Şekil 6.5 : Yazılan programın blok şeması.

Her popülasyondaki en büyük uygunluğa sahip birey sürekli bir sonraki nesle aktarılır ve durdurma kriteri sağlanana kadar en iyisi bulma işlemine devam edilir. Yukarıda anlatılan ana program ve alt programlar arasındaki bilgi alışverişi Şekil 6.5’de gösterilmiştir.

#### **6.2.4 Seçim yöntemi**

Bu işlem ile yeni oluşturulacak neslin bireylerinin seçimi yapılmaktadır. Önemli bir adımdır çünkü mantıklı şekilde yapılmaz ise bulunan iyi bireylerin kaybı da söz konusu olabilir. Geliştirilen programda, bireylerin uygunluk değerleri belirlenip son genlerine yazıldıktan sonra rastgele iki birey seçilir. Bu iki bireyden uygunluğu yüksek olan birinci ebeveyn olur, ikinci ebeveyn de aynı şekilde seçilir. Bu ebeveyn bireylerden genetik operatörler yardımıyla yeni bireyler üretilir.

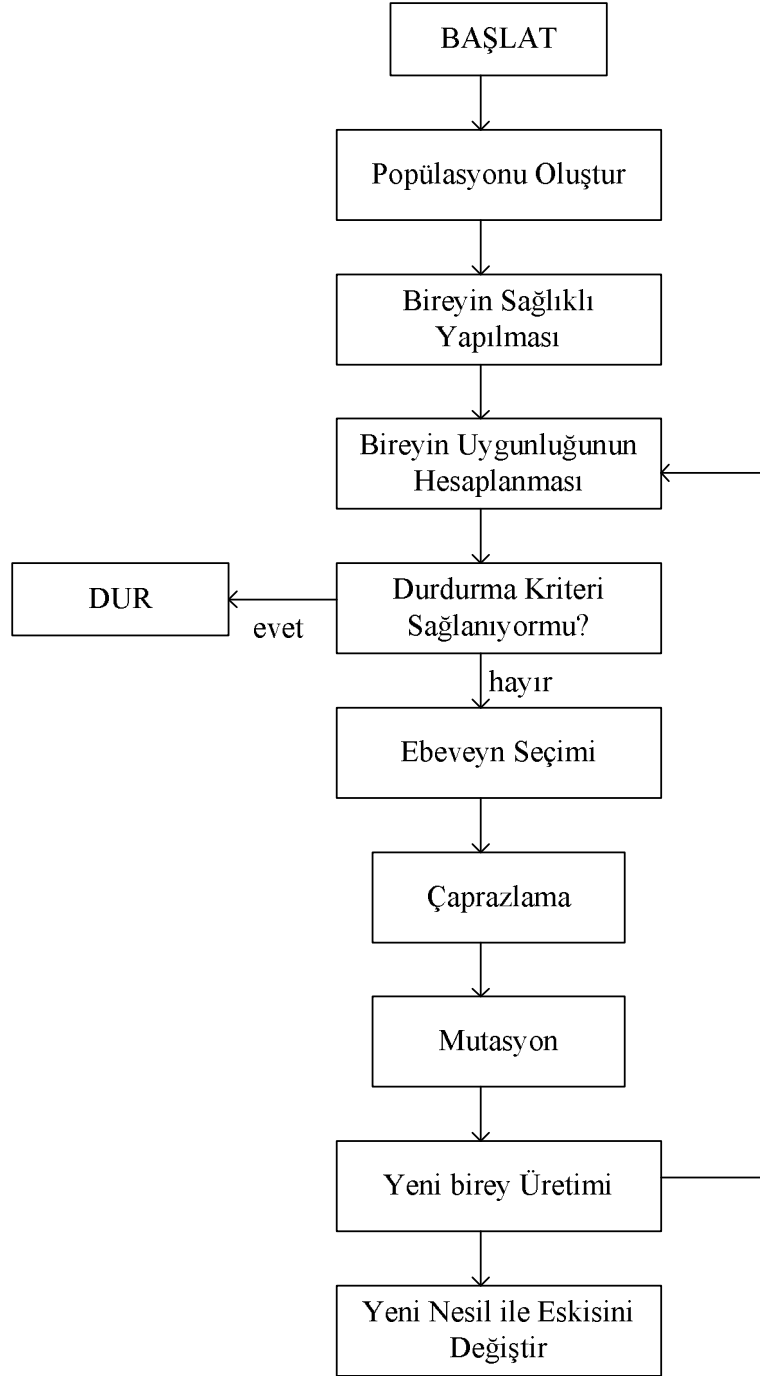
#### **6.2.5 Yeni nesil üretimi**

Mevcut nesilde mümkün olduğunca iyileştirme yapılarak bir sonraki nesli oluşturma işlemine denilmektedir. Kurulan algoritmada uygunluk değerleri belirlenmiş olan bireyler arasından rastgele iki tanesi seçilir, uygunluğu yüksek olan birinci ebeveyn yapılır, ikinci ebeveyn ise aynı şekilde seçilir. Daha sonra bu iki birey arasında rastgele seçilen bir noktadan çaprazlama işlemi yapılır ve iki farklı bireyler oluşturulur. Bu işlemden sonra başka bir genetik operatör olan mutasyon işlemi %1 olasılıkla rastgele bir noktada olacak şekilde bireylere uygulanır. Bu işlemlerin uygulandığı bireylerin bozulma ihtimaline karşı, tekrardan düzeltme fonksiyonuna gönderilirler. Düzeltilmiş yeni bireyler uygunlukları hesaplanmak üzere uygunluk programına gönderilir ve son genlerine uygunluk değerleri yazılmış şekilde yeni nesile kopyalanırlar. Bu döngü popülasyonda olması gereken birey sayısına ulaşıncaya kadar devam eder. Buradan sonra yeni nesil içindeki bireylerin en kötüsü bulunur, en kötüsü yerine mevcut nesildeki en iyi birey yazılır. Bir adım sonra ise bu yeni nesil, mevcut nesle kopyalanır ve aynı işlemler bu nesil içinde gerçekleştirilir.

#### **6.2.6 YSA destekli genetik algoritmanın çalışma mantığı**

Aslında bu çalışma sisteminin klasik bir GA hesaplama algoritmasından tek farkı bireylerin sağlıklı hale getirilmesi için yapay sinir ağlarından destek alınmasıdır denilebilir. Diğer prosedürler açısından program aynı mantık ile devam etmektedir.

Yani ilk olarak popülasyon oluşturulur, uygunluk hesaplanır, seçme işlemi yapılır ve istenilen değere ulaşana kadar yeni nesil üretilmesine devam edilir. Şekil 6.6'da programın genel akış şeması gösterilmiştir.

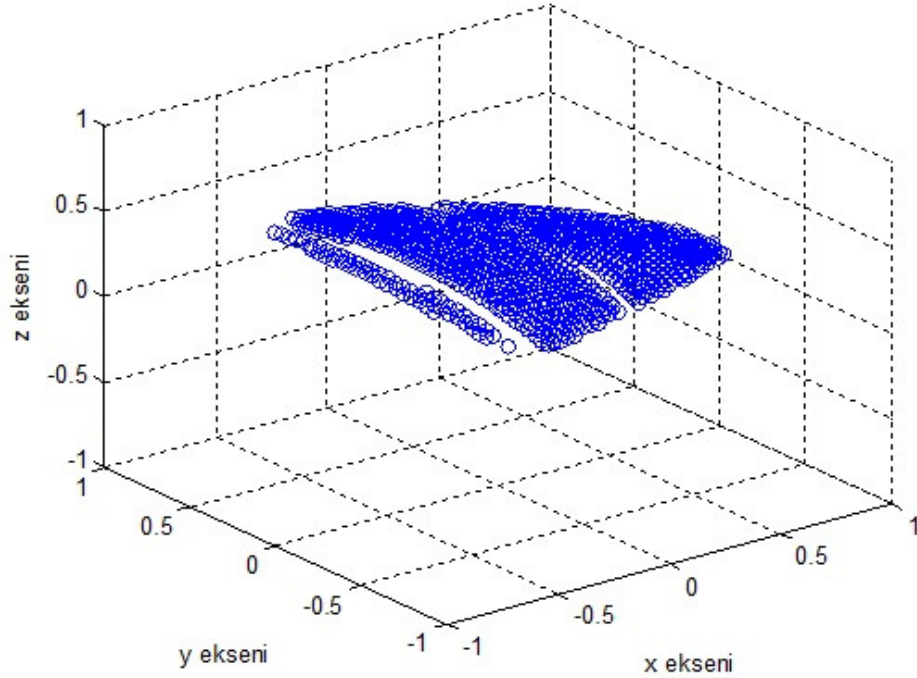


Şekil 6.6 : GA programının akış şeması.

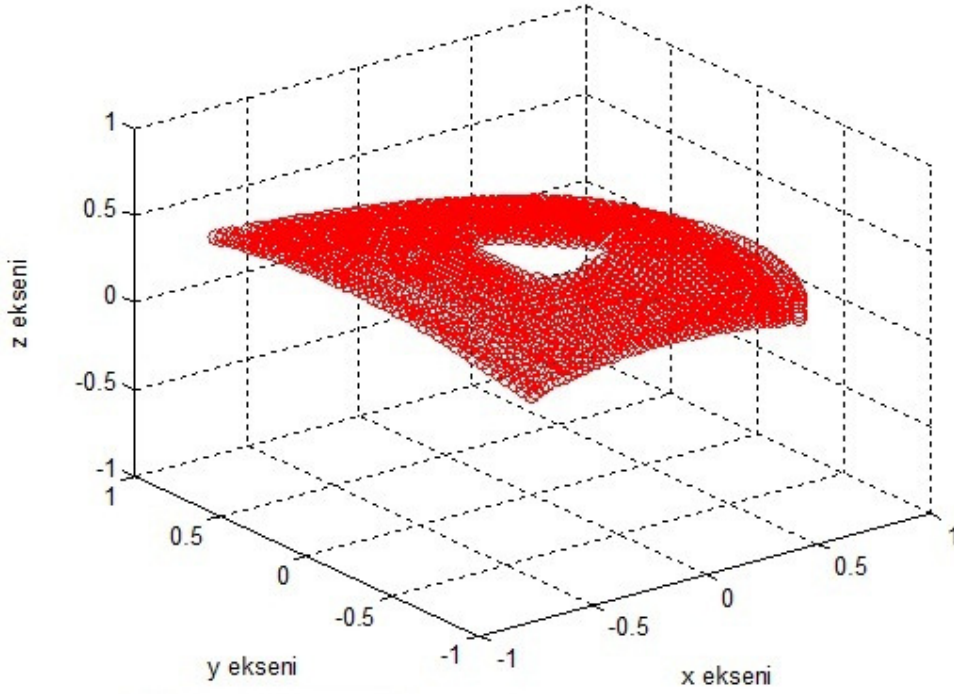
### 6.2.7 YSA destekli GA programı kullanılarak 6x3 Stewart Platformu üzerine yapılan uygulama sonuçları

Bu bölümde daha önceden klasik sinir ağırları ile yapılan 6x3'lük SPM'nin çalışma uzayı analizinin, YSA destekli GA programı ile daha pratik, kolay ve yüksek doğrulukta yapılması amaçlanmıştır. GA ile belirlenen farklı YSA yapılarının başarı tabloları verilmiştir ve kullanılan yöntemin avantajı, dezavantajı tartışılmıştır. Ayrıca programın performansının artırılabilmesi için yapılabileceklerden de bahsedilmiştir.

Yukarıda anlatılan YSA destekli GA algoritması, C dilinde yazılmış ve 6x3'lük SPM yapısının dataları üzerinden simülasyonu yapılmıştır. Stewart platformunun üst platformunun konumlarından oluşan bu datalara göre yaptığımız program ile, bu noktaların çalışma uzayı içinde mi, dışında mı olduğuna karar verilecektir. En çok doğru kararı veren, yapı bizim GA ile ürettiğimiz en iyi birey olarak kabul edilmekte ve sonuç olarak döndürülmektedir. SPM'nin çalışma uzayının içerisinde ve dışında bulunan, bizim üzerinden doğrulama yaptığımız dataların grafiksel gösterimi Şekil 6.7 ve Şekil 6.8'de verilmiştir. Çalışma uzayı içerisindeki veri sayısı toplam olarak 11220 tane, çalışma uzayı dışındaki data sayısı da toplamda 36891 tanedir.

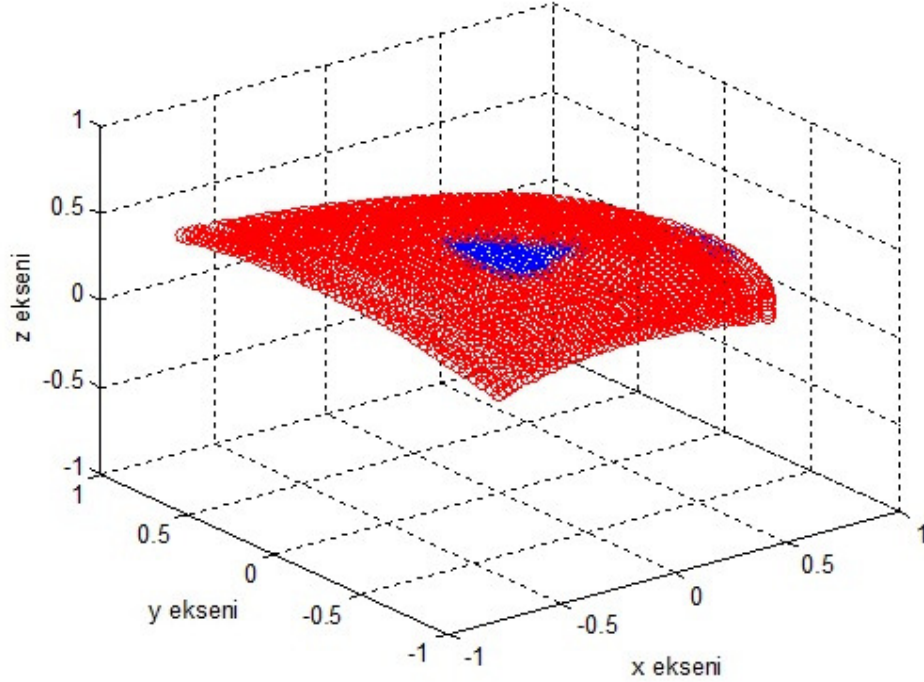


Şekil 6.7 : Çalışma uzayı içindeki üst platformun konumları.



**Şekil 6.8 :** Çalışma uzayı dışındaki üst platform konumları.

Yukarıda gösterilen veriler bizim programımızın giriş değerlerini göstermektedir. Bu değerlerin çalışma uzayının içinde yada dışında olduğuna karar verecek olan YSA yapısının, yine YSA destekli bir GA programı ile belirlenmesi üzerine çalışılmıştır. Çizelge 6.1’de GA programın verdiği örnek birkaç birey verilmiştir. Her bireyin başarısı yerini doğru bildiği nokta sayısı ile ölçülmektedir. İlk satırdaki kromozomun verdiği sonuçlara bakılarak, çalışma uzayı içindeki tüm noktalar için oldukça iyi düzeyde, %68.2 ile doğru bildiğini söyleyebiliriz. Aynı kromozomun çalışma uzayı dışındaki veriler içinse tüm datalar içerisinde %61.6’lık bir yüzdeyle doğru sonuç verdiği görülmektedir. Birbirine bu kadar geçmiş çalışma uzayı içindeki ve dışındaki dataların doğru bulunmasındaki bu yüzdeler oldukça yeterlidir. Şekil 6.9 ‘da çalışma uzayı içi ve dışındaki verilerin yakınlığı açıkça görülmektedir. Verilen doğruluk yüzdesi, bu kromozom üzerinde kodlanmış; ilk gizli katmandaki sinir sayısı, ikinci gizli katmandaki sinir sayısı, tüm bağlantılardaki ağırlık değerleri, tüm sinirlerin bias değerleri gibi verilere göre tasarlanan bir YSA ağına aittir denilebilir. Çünkü biz kromozom yapımızda bu doğruluğu verecek bir ağın tüm yapısal değerlerini tutmaktayız.



**Şekil 6.9 :** Çalışma uzayı içindeki ve dışındaki dataların birlikte gösterimi

İkinci bireyin ise çalışma uzayı içindeki dataların bilinmesinde gösterdiği performans çok iyi seviyededir. İkinci birey çalışma uzayı içindeki datalar için %93.7' lik doğru bilme oranına, çalışma uzayı dışındaki veriler için %45.4'lük doğru tespit oranına sahiptir. Çalışma uzayı içindeki ve dışındaki datalar arasındaki bu büyük doğrulama farkı bu bireyin en iyi birey olarak seçilmesini engelleyebilir.

**Çizelge 6.1 :** Oluşturulan ağ yapılarının başarı yüzdeleri.

İlk Gizli Katmandaki Sinir Sayısı	İkinci Gizli Katmandaki Sinir Sayısı	Çalışma Uzayı İçerisindeki Datalar için Bilinen Nokta Sayısı	Çalışma Uzayı Dışındaki Datalar için Bilinen Nokta Sayısı
4	3	7655 (%68.2)	22701 (%61.6)
2	5	10522 (%93.7)	16773 (%45.4)
4	6	7567 (%67.4)	19278 (%52.2)
6	4	8443 (%75.2)	21108 (%57.2)
6	5	8473 (%75.5)	21291 (%57.7)

Ancak bir önceki sayfada bulunan Çizelge 6.1’de dikkat edilmesi gereken asıl nokta oluşturulan ağ yapılarının başarıları sadece çalışma uzayı içindeki yada dışındaki verilere göre değerlendirilmemelidir. Yani yapay sinir ağının başarısı sadece belli bir alandaki başarısına göre değil, verilen dataların geneline verdiği doğru sonuca göre değerlendirilmelidir. Bu nedenle hem çalışma uzayı içerisindeki hem çalışma uzayı dışındaki veriler için birbirine yakın doğruluk yüzdeleri gösteren ilk kromozom bu yapılar içerisindeki en iyi bireydir denilebilir. Sonuç olarak bu verilere bakılarak giriş/çıkış sayıları belli olan ve elimizde dataları bulunan 6x3 SPM’nin çalışma uzayı analizinde kullanılabilir en uygun YSA yapısının 3432’lük yapı olduğu bulunmuştur. Yazılan programda kromozomlar ile gizli katmanlardaki nöron sayılarının yanında ağırlık ve bias değerleride kodlandığı için programın sonunda en uygun bulduğumuz yapının tüm değerleri hesaplanmış olunur. Aşağıda Şekil 6.9’da birinci gizli katmandaki sinir sayısının 4, ikinci gizli katmandaki sinir sayısının 3 olarak kodlandığı kromozomun diğer genlerindeki program ile optimize edilmiş ağırlık ve bias değerleri gösterilmiştir.

$$\text{Weight1} = \begin{bmatrix} -7.262926 & 5.041522 & -7.681049 \\ -4.673687 & 1.978702 & -4.290550 \\ 3.613929 & -7.775501 & 3.775796 \\ 1.767706 & -2.640655 & 3.957522 \end{bmatrix}$$

$$\text{Weight2} = \begin{bmatrix} 4.094160 & 3.870780 & 1.060114 & -0.451826 \\ 2.221840 & 0.530486 & 4.944385 & 0.497366 \\ 6.877147 & 1.021785 & -1.867306 & -2.217171 \end{bmatrix}$$

$$\text{Weight3} = \begin{bmatrix} -0.576150 & 1.475573 & 1.916572 \\ -2.602312 & -1.475342 & 1.259211 \end{bmatrix}$$

$$\text{Bias1} = [ 4.239500 \quad -1.066500 \quad 4.239500 \quad 2.700000 ]$$

$$\text{Bias2} = [ 2.221840 \quad 0.530486 \quad 4.944385 ]$$

$$\text{Bias3} = [ -1.338158 \quad 1.341078 ]$$

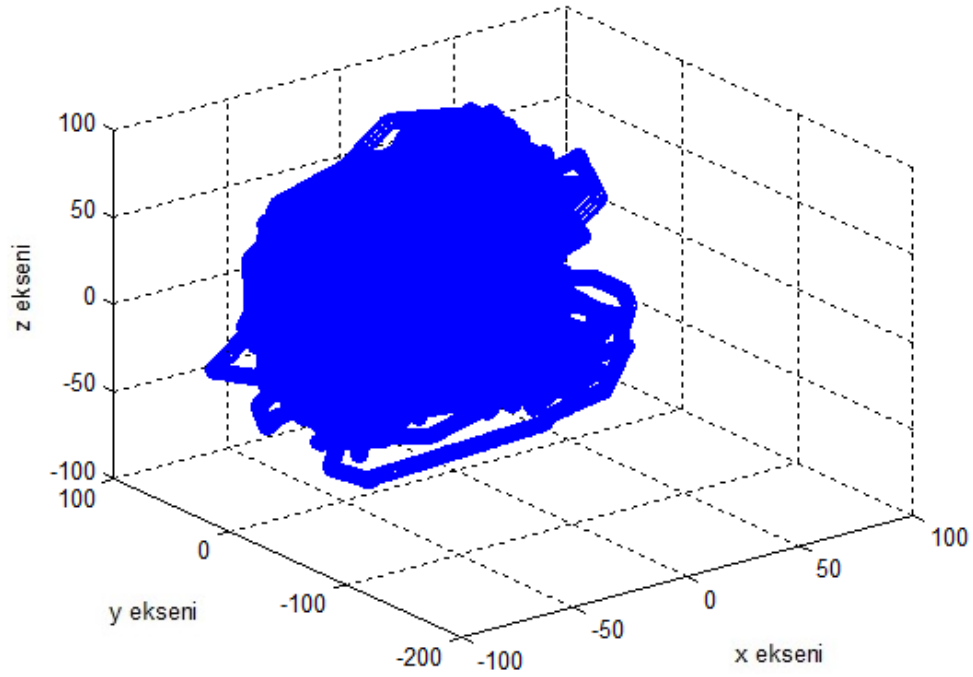
**Şekil 6.10 :** 3432’lik yapıyı veren kromozomun ait ağırlık ve bias değerleri.



Sonuç olarak Şekil 6.7 ve Şekil 6.8'deki grafiklerden görüldüğü gibi çalışma uzayının içindeki ve çalışma uzayının dışındaki giriş dataları çok iç içe girmiş şekilde olduğu için, elde edilen ağların başarı oranları oldukça yeterlidir. Böylece kaynak [46]'da aynı dataların işlendiği ve deneme yanılma yoluyla yapısının 3432'lik bulunduğu YSA sisteminin, yazılan yeni bir algoritma ile doğrulaması yapılmıştır. Buradan hareketle yazılan YSA tabanlı GA programının gerçekten iyi bireyler ürettiği ve amaçlanan hedefe ulaşıldığı sonucuna varabiliriz.

### 6.3 Deneysel 6x6 SPM'nin Çalışma Uzayı Analizi için Kullanılacak YSA Yapısının Genetik Algoritmalar ile Belirlenmesi

Bu kısımda, yukarıda bölüm 6.2'de ayrıntılarıyla anlatılan YSA destekli GA programının doğruluğunun pekiştirilmesi için deneysel datalar üzerinden hareket edilerek bu datalara uygun yapay sinir ağı yapısı hesaplaması yapılmıştır.

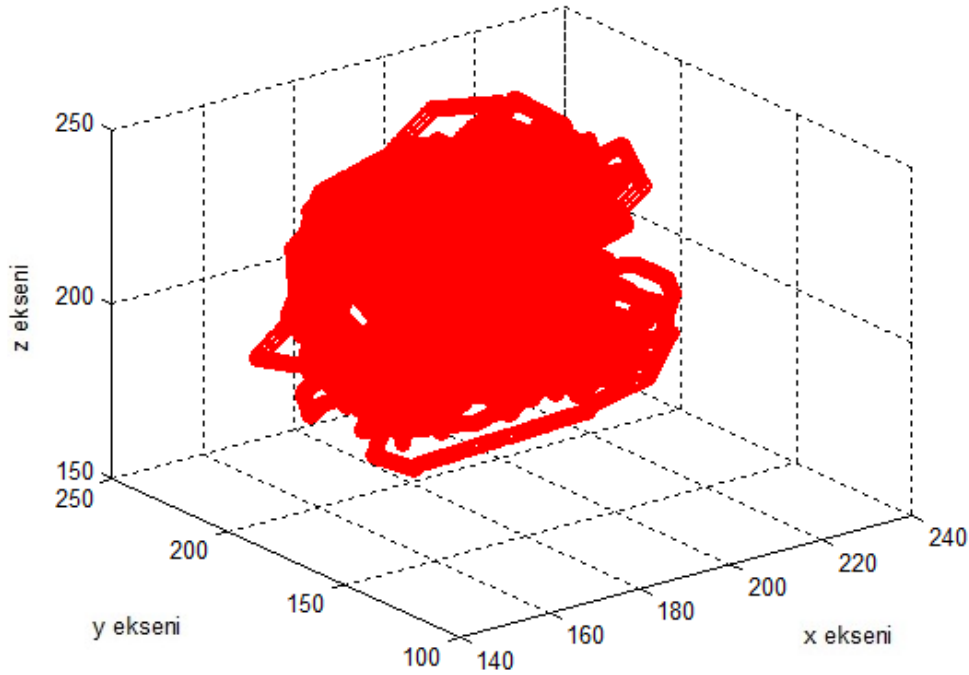


Şekil 6.11 : Deneysel sistemin çalışma uzayı içerisindeki dataları

Kullanılan datalar gerçek hayatta sinemalarda simülatör olarak kullanılan 6x6'lık Stewart Platformunun konum bilgilerinden oluşmaktadır.

Programın çalışma mantığına uygun olarak, çalışma uzayının içinde olan dataların yanında çalışma uzayı dışını temsil eden datalar da gerekmektedir. Doğal olarak gerçek bir sistemden çalışma uzayının dışındaki dataları elde etmemiz mümkün olmamaktadır. Bu nedenle platformun gidebileceği noktalara belli öteleme değerleri eklenmiştir, böylece çalışma uzayına ait olmayan ve programımızın doğruluğunun tespitinde kullanacağımız bilgiler elde edilmiştir.

Deneysel sistemden alınan ve programda kullanılan çalışma uzayı içerisindeki noktalar Şekil 6.11’ de görülmektedir. Bu grafik aslında noktalar kümesinden oluşmaktadır. Ancak sinema salonunda kullanılan ve film efektine göre hareket eden 6x6 Stewart Platformuna ait olan bu bilgilerde, aynı noktadan ve yakın noktalardan birkaç defa geçildiği için grafikte noktaların birleşmiş hali görülmektedir. Ayrıca buradaki toplam nokta sayısı 84447 gibi yüksek miktarlarda olduğu için, sonuçta bu şekilde bir grafik oluşmuştur.



**Şekil 6.12 :** Deneysel sistemin çalışma uzayı dışındaki dataları

Şekil 6.12’de ise deneysel sistemin çalışma uzayı içindeki datalarının ötelenmesi ile elde edilmiş, çalışma uzayı dışındaki dataların grafiksel hali görülmektedir. Burada yine yukarıda anlatıldığı gibi noktaların çakışması nedeni ile noktalar kümesi bir

yüzeş oluşturmuş gibi görünmektedir. Çalışma uzayı dışının tanımlanmasında ise 80000 data kullanılmıştır.

YSA destekli GA programı kullanılarak deneşel 6x6 Stewart Platformu dataları üzerine yapılan uygulama sonuçları

Teş çalışmasının amacını oluşturan, Genetik Algoritmalar ile çeşitli problemlerin çözümünde kullanılacak uygun Yapay Sinir Ağları yapısını bulan programın geliştirilmesi doğrultusunda program iki farklı probleme uygulanmıştır. Yukarıda bahsedilen deneşel olmayan 6x3 SPM'ye ait çalışma uzayı bilgilerine uygulanan örnekten sonra, bu kısımda deneşel bir 6x6'lık SPM'nin dataları üzerinden geliştirilen programın doğruluęu araştırılmıştır. Çizelge 6.2'de programın ürettięi birkaç sonuç ve bulunan en iyi sonuç gösterilmiştir.

**Çizelge 6.2 :** Program ile oluşturulan ağ yapılarından örnekler ve başarı oranları

İlk Gizli Katmandaki Sinir Sayısı	İkinci Gizli Katmandaki Sinir Sayısı	Çalışma Uzayı İçerisindeki Datalar için Bilinen Nokta Sayısı	Çalışma Uzayı Dışındaki Datalar için Bilinen Nokta Sayısı	Ortalama Hata (toplam hata / nokta say.)
6	4	80913	79887	0.118
6	3	78549	80000	0.128
6	5	79171	80000	0.132
4	5	77379	80000	0.160
5	5	79753	80000	0.125
5	5	78811	80000	0.141
6	6	77109	80000	0.187

Yukarıdaki çizelgeden görüldüğü gibi, çalışma uzayının dışındaki verilerde doğruluk oranı %100'lere kadar çıkmıştır. Bunun nedeni gerçek sistemde çalışma uzayı dışındaki dataların ancak öteleme ile üretilmesidir. Çünkü bu bireye göre, doğru bulunan nokta sayısı en fazla ve ek olarak tüm giriş dataları kullanılarak hesaplanan ortalama hata oranı 0.118 ile en düşük değerdedir. Genetik algoritma uygunluk değeri hesaplama mantığımızı doğru bulunan nokta sayısı değerine göre oluşturmuş olsak bile uygunluk değerini ortalama hata değerinin küçüklüğüne de endekslenmiş olsaydık, program sonuç çıktısı olarak bize ilk değeri gönderecektir.

Durdurma kriteri olarak ise en iyi bireyin belirlenen iterasyon sayısı boyunca deęişmeyip aynı kalması kullanılmaktadır.

## 7. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, son yıllarda birçok alanda yaygın olarak kullanılmaya başlanılan 6 serbestlik dereceli Stewart Platform Mekanizması ile ilgili çalışılmıştır. İlk olarak paralel mekanizmalar ve SPM'ler için yapılan literatür araştırması Bölüm 1'de anlatılmıştır. Bölüm 2'de ise 3x3'lük SPM'nin ters ve düz kinematik analizleri MATLAB programı kullanılarak yapılmıştır. Programların kodları ekler kısmında verilmiştir. Düz kinematik analiz için yöntem olarak Bezout Metodu seçilmiş ve denklemleri oluşturulmuştur.

Bölüm 3'de ise yapay sinir ağları ile ilgili literatür araştırması yapılmış, YSA'nın çalışma mantığı araştırılmış ve kullanılan algoritma çeşitleri, ağ yapıları incelenmiştir. Bu bölümün koyulmasının asıl amacı, genetik programlamada yardımcı yöntem olarak YSA'nın kullanılacak olması ve GA ile yapılacak hesaplamaların sonucunda 6x3'lük SPM'nin çalışma uzayı analizini yapan uygun bir YSA yapısının belirlenecek olmasıdır. Takip eden bölümde genetik algoritmaların tarihi gelişimi, çalışma mantığı, kullanılan operatörler, yöntemin avantajları, dezavantajları gibi başlıklar incelenmiştir.

Bu çalışmada YSA ve GA birlikte kullanıldığı için Bölüm 5'de bu iki yöntemin hangi uygulamalarda ve nasıl birarada kullanıldığına dair bir literatür araştırılması yapılmıştır.

Son kısımda ise, GA programlaması mantığının anlaşılması ve yazılan programın başka bir problemde doğruluğunun test edilmesi açısından, genel bir problem olan gezgin satıcı problemi çözülmüştür. Genetik algoritmada çok önemli olan kromozom kodlamasının ayrıntıları, düzeltme fonksiyonunun çalışma mantığı, yazılan programın seçim ve yeni nesil üretme algoritması, programın durma kriteri bu bölümde açık olarak anlatılmıştır. 11 şehirden oluşan örnek bir tur planlaması da çözümlenerek GSP için yapılan çalışmalar sonlandırılmıştır. Yine simülasyonlar kısmında bir GA programı geliştirilerek, 6x3'lük SPM'ye ait olan datalar üzerinden çalışan YSA'nın yapısı belirlenmiştir. Burada kullanılan GA programının akış şeması ve oluşturulan programın çalışma mantığı geniş bir şekilde açıklanmıştır. Bu

programdan alınan sonuçlara göre 6x3'lük SPM'nin çalışma uzayı analizinde kullanılacak en uygun yapının 3432 yapısı olduğu bulunmuş ve aynı zamanda bu yapı ile kullanılacak ağırlık ve bias değerleri hesaplanmıştır. Bulunan ağırlık doğru çalışma oranı; çalışma uzayı içerisindeki veriler için %68.2 , çalışma uzayı dışındaki veriler için %61.6 olduğu görülmüştür. Farklı yapıdaki YSA'ların verdiği doğruluk oranları Çizelge 6.1'de verilmiştir, bu tabloda bazı yapıların çalışma uzayı içerisindeki veriler için başarı oranlarının %90'lara kadar çıktığı görülmektedir. Ancak bizim için en iyi yapı sadece çalışma uzayı içindeki yada sadece dışındaki veriler için doğru çalışan değildir. Bu nedenle çalışma uzayı içi ve dışı için birbirine yakın ve yüksek doğruluk oranları veren 3432'lik yapının en optimum yapı olduğu düşünülmüştür.

Yine Bölüm 6 başlığı altında, geliştirilmiş olan YSA ve GA hibrit programı mantığı ile deneysel bir 6x6'lık SPM'nin çalışma uzayı verileri üzerinden çalışmalar yapılmıştır. Elde edilen sonuçların doğruluk yüzdesi çalışma uzayı dışındaki veriler için %100'e yakın olurken, çalışma uzayı içindeki veriler için %90'lara tekabül etmiştir. Alınan bu sonuçlar ile birlikte C tabanlı geliştirilen GA ve YSA hibrit programının doğruluğu kanıtlanmıştır.

Paralel bir mekanizma olan SPM'nin farklı yapıdaki örnekleri üzerine yapılan bu çalışmaya, başka mekanizmaların farklı problemlerinin geliştirilen program kullanılarak çözümlenmesi eklenebilir. Böylece bu metodun doğruluğu pekiştirilmiş ve program genelleştirilmiş olur.

## KAYNAKLAR

- [1] **Dasgupta, B., Mruthyunyaja, T. S.**, 1998. Force redundancy in parallel manipulators: theoretical and practical issues, *Mechanism and Machine Theory*, vol: 33, Issue 6, s.s: 727-742.
- [2] **Clerc, J. P., Tol, U.A., Wiens, G.J.**, 2002. Deburring Using A Micro/Macro Parallel Kinematic Machine, *Proc. Conference on Recent Advances in Robotics*.
- [3] **Zabalza, I., Ros, J., Gil, J. J., Pintor, J. M., Jimenez, J. M.**, 2002. “TRI-SCOTT. A MICABO like 6-DOF Quasi-Decoupled Parallel Manipulator, *Proc. of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*.
- [4] **Pollard, W. L.**, 1942, Position Controlling Apparatus, US Patent No: 2.286.571, June 16.
- [5] **Gough, V. E., Whitehall, S. G.**, 1962. Universal tyre test machine, *Proc. 9th International Technical Congress, F.I.S.I.T.A.*, s: 177.
- [6] **Stewart, D.**, 1965. A Platform with Six Degrees of Freedom, In *Proceedings of the institution of mechanical engineers*, Cilt: 80, s: 371-386.
- [7] **Hunt, K. H.**, 1983. Structural kinematics of in-parallel-actuated robot-arms, *Proc. ASME J. Mech. Transm. Autom. Des.*, cilt: 105, s:705–712.
- [8] **Gosselin C. M., Wang J.**, 1997. Singularity Loci of Planar Parallel Manipulators with Revolute Actuators, *Robotics and Autonomous Systems*, cilt:21, s.s: 377-398.
- [9] **Kumar V.**, 1992. Characterization of Workspace of Parallel Manipulators, *ASME Journal of Mechanical Design*, Sept., cilt: 114, s.s: 368-375.
- [10] **Callegari, M., Tarantini, M.**, 2003. Kinematic Analysis of a Novel Translational Platform, *J. Mech. Des.*, June, vol:125, issue 2, 308 (8 pages).
- [11] **Lazard, D., Merlet, J. P.**, 1994. The (true) Stewart platform has 12 configurations, *Proc. IEEE Int. Conf. on Robotics and Automation*, vol:3, ss: 2160 – 2165.
- [12] **Zhang, D.**, 2010. *Parallel Robotic Machine Tools*, Springer Science+Business Media, LLC.
- [13] **Stoughton, R. S., Arai, T.**, 1993. A modified Stewart platform manipulator with improved dexterity, *Robotics and Automation, IEEE Transactions on*, Apr, Vol.: 9 Issue:2.
- [14] **Moon, Y., Kota, S.**, 2002. Design of compliant parallel kinematic machines, *Proc. of DETC'02*, s.s: 1-7, September 29-October 2.

- [15] **Chen G. L., Wang S.**, 2007. Analysis of Sensitivity for Six-Axis Force/Torque Sensor Based on Stewart Platform, Int. Conf. on Mechatronics and Automation, August5-8, Harbin China.
- [16] **Yao J., Hou Y., Lu L., Zhao Y.**, 2006. Analysis of a Pre-stressed Six-component Force/Torque Sensor Based on Stewart Platform, Int. Conf. on Robotics and Biomimetics, Dec. 17-20, Kunming, China.
- [17] **Yoon, J., Ryu, J.**, 2001. Design, Fabrication, and Evaluation of a New Haptic Device Using a Parallel Mechanism, IEEE/ASME Trans. on Mechatronics, vol. 6, no 3, s.s 221-233, september.
- [18] **Yoon, J., Ryu, J.**, 2000. Design and Analysis of a New Haptic Device Using a Parallel Mechanism, Proc. of the IEEE/RSI Int. Conf. on Intelligent Robots and Systems, s.s 949-954.
- [19] **Arata J., Ikedo N., Fujimoto H.**, 2010"Haptic Device using a newly developed Redundant Parallel Mechanism," IEEE Trans. on Robotics.
- [20] **Boian, R.F., Bouzit, M., Burdea, G.C., Lewis, J. and Deutsch, J.E.**, 2005. Dual Stewart Platform Mobility Simulator, Proc. of the 2005 IEEE 9th Int. Conf. on Rehabilitation Robotics, s.s 550-555, June 28 - July 1, Chicago, IL, USA.
- [21] **Bergamasco, M., Perotti, S., Avizzano, C. A., Angerilli, M., Carrozzino, M., and Ruffaldi, E.**, 2005. Fork-Lift Truck Simulator For Training in Industrial Environment, Emerging Technologies and Factory Automation, 10th IEEE Conference on, vol. 1, s.s 689-693.
- [22] **Dasgupta, B., Mruthyunjaya, T. S.**, 2000. The Stewart platform manipulator: a review, Elsevier Mechanism and Machine Theory 35, s.s: 15-40.
- [23] **Chen, N. X., Song, S. M.**, 1994. Direct position analysis of the 4-6 Stewart Platform, Journal Mechanical Design, March, vol: 116, Issue 1, s.s: 61 (6 sayfa).
- [24] **Lee, T. Y., Shim, J. K.**, 2001. Forward kinematics of the general 6–6 Stewart platform using algebraic elimination, vol: 36, Issue: 9, s.s: 1073 – 1085.
- [25] **Song, S. K., Kwon, D. S.**, 2001. Efficient Formulation Approach for the Forward Kinematics of the 3-6 Stewart-Gough Platform, proceedings of International Conference on Intelligent Robots and Systems, s.s: 1688-1693, USA, Oct.29 – Nov03.
- [26] **Lin, W., Griffis, M., Duffy, J.**, 1992. Forward Displacement Analyses of the 4-4 Stewart Platforms, Journal of Mechanical Design, Vol: 114, Issue 3,s.s: 444 - 450.
- [27] **Yıldız, İ., Ömürlü, V. E., Ekicioglu, Z., Güney, A.**, 2010. Serbestlik Dereceli Paralel Mekanizmadaki İleri Kinematik Analiz Yöntemleri, Türkiye Otomatik Kontrol Kongresi.
- [28] **H. Sadjadian, H.D. Taghirad ve A. Fatehi**, “Neural Networks Approaches for Computing the Forward Kinematics of a Redundant Parallel Manipulator”, Proc. International Journal of Computational Intelligence, Cilt: 2, Sayı:1, s: 40-47, 2005.



- [29] **Y. Wan ve S. Wang**, “Kinematic Analysis and Simulation System Realization of Stewart Platform Manipulator”, Proc. The Fourth International Conference on Control and Automation, Canada, 2003.
- [30] **C.C. Nyugen, Z.L. Zhou, S.S. Antrazi**, 1991. “Efficient Computation of Forward Kinematics and Jacobian Matrix of a Stewart Platform-Based Manipulator”, Proc. IEEE Southeastcon, s: 880–884.
- [31] **L.W. Tsai, G.C. Walsh ve R.E. Stamper**, “Kinematics of a Novel Three DOF Translational Platform”, Proc. International Conference on Robotics and Automation, Minnesota, 1996.
- [32] **C. Zhang, S.M. Song**, “Forward Kinematics of a Class of Parallel (Stewart) Platforms with Closed-Form Solutions”, Proc. International Conference on Robotics and Automation, 1991.
- [33] **T.Y. Lee, J.K. Shim** “ Elimination-Based Solution Method for the Forward Kinematics of the General Stewart-Gough Platform,” Proc. F.C. Park C.C. Iurascu, editor, Computational Kinematics, s: 259-267, 2001.
- [34] **Z. Geng, L. Haynes**, “Neural Network Solution for the Forward Kinematics Problem of a Stewart Platform”, Proc. International Conference on Robotics and Automation, California, s: 2650-2655, 1991.
- [35] **D. Jakobovic ve L. Jelenkovic**, “Kinematic Evaluation and Forward Kinematic Problem for Stewart Platform Based Manipulators”, Proc. International Conference on Computational Cybernetics, 2003.
- [36] **S. Haykin**, “Neural Networks”, Prentice Hall, 1998.
- [37] **R. Rojas**, “Neural Networks, A Systematic Introduction”, Springer, Germany, 1996.
- [38] **C. M. Bishop**, “Neural Networks and Their Applications”, Rev. Sci. Instrum 65(6), june 1994.
- [39] **W. S. McCullogh, ve Pitts, W.** “A Logical Calculus of the Ideas Immanent in Nervous Activity”, Bulletin of Mathematical Biophysics, 5, s.s: 115-133, 1943.
- [40] **M. L. Minsky**, “Theory of Neural-Analog Reinforcement Systems and its Applications to the Brain-Model Problem”, PhD thesis Princeton University.
- [41] **F. Rosenblatt**, “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms”, Washington, D.C.: Spartan Books, 1961.
- [42] **B. Widrow, M.E. Hoff**, “Adaptive switching circuits”, WESCON Convention Record Part IV, s.s: 96-104, 1960.
- [43] **M. L. Minsky, S. A. Papert**, “Perceptrons An Introduction to Computational Geometry”, MIT press, 1959.
- [44] **J. J. Hopfield**, “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”, Proc. Natl. Acad. Sci., USA, vol:79, s.s: 2554 -2558, Nisan, 1982.

- [45] **D. E. Rumelhart, G. E. Hinton, R. S. Williams**, “Parallel Distributed Processing: Explonations in the Microstructure of Cognition”, MIT Press, Cambridge, vol.:2, 1986.
- [46] **H. Alp**, “Sezgisel Yöntemlerle Paralel Mekanizmaların Çalışma Uzayı Analizi”, Doktora Tezi, İTÜ, Ocak, 2007.
- [47] **U. Özveren**, “PEM Yakıt Hücrelerinin Yapay Sinir Ağları ile Modellenmesi”, Yüksek Lisans Tezi, YTÜ, 2006.
- [48] **D. O., Hebb**, “The Organization of Behaviour”, Wiley, New York, 1949.
- [49] **U., Özveren**, “PEM Yakıt Hücrelerinin Yapay Sinir Ağları ile Modellenmesi”, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, 2006.
- [50] **B., Widrow**, M. E. Hoff, “Adaptive Switching Circuits”, IRE WESCON Conv. Rec., Part 4, s.s: 96-104, 1960.
- [51] **T. Kohonen**, “The Self-organizing Map”, Proc. IEEE 78, s.s: 1464-1480, 1990.
- [52] **Gen, M., Cheng, R.**, 1997. Genetic Algorithms and Engineering Design, John Wiley & Sons, Inc, NY.
- [53] **Holland, J. H.**, 1975. Adaptation in Natural and Artificial Systems. MIT Press.
- [54] **Goldberg, D. E.**, 1989. Genetic Algorithms in search, optimization and machine learning, Addison-Wesley.
- [55] **Mitchell, M., Forrest, S.**, 1993. Genetic Algorithms and Artifical Life.
- [56] **Guha, A., Harp, S. A., Samad, T.**, 1992. US Patent Number: 5.140.530, Aug 18.
- [57] **Guo, Z., Uhrig, R. E.**, 1992. Using Genetic Algorithms to Select Inputs for Neural Networks,s:223-234,COGANN.
- [58] **Montana, D. J.**, 1995. Intelligent Hybrid Systems, s: 85-104, Wiley.
- [59] **Gupta, J. N. D., Sexton, R. S.**, 1999. Comparing Backpropagation with a Genetic Algorithm for Neural Network Training, Omega, vol: 27, s: 679-684.
- [60] **Potter, M. A.**, 1992. A Genetic Cascade Correlation Learning Algorithm, COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, s:123-133. IEEE Computer Society Press.
- [61] **Leitch D., Probert P.**, 1994. Genetic Algorithms for the Development of Fuzzy Controllers for Mobile Robots, Advances in Fuzzy Logic, Neural Networks and Genetic Algorithms, IEEE Workshop, Nagoya Japan, August.
- [62] **Zalzala A. M. S., Ang M. C., Chen M., Rana A. S., Wang Q.**, 1997. Evolutionary algorithms for robotic systems: principles and implementations,Genetic Algorithms in engineering systems, Zalzala A. M. S., Fleming P. J. (editors), vol:55, chapter 8.
- [63] **De Jong, K. A.**, 1992. Are genetic algorithms function optimizers?, R. Männer and B. Manderick, Editors, Parallel Problem Solving from Nature, cilt: 2, North-Holland, Amsterdam, s:3–13.

- [64] **Freitas, A. A.**, 1997. A Genetic Programming Framework for Two Data Mining Tasks: Classification and Generalized Rule Induction, Genetic Programming Proceedings of 2nd Annual Conference, s: 96-101.
- [65] **Koza, J. R., BennettIII, F. H., Huddings, J. L., Bade, J. L., Keane, M. A., Andre, D.**, 1998, Evolving Sorting Networks Using Genetic Programming and the Rapidly Reconfigurable Xilinx 6216 field-programmable Gate Array, 31. Asilomar Conference on Signals, Systems and Computers, cilt: 1, s: 404-410.
- [66] **Wang, D., Li, X.**, 2009. GAPK: Genetic Algorithms with Prior Knowledge for Motif Discovery in DNA Sequences, CEC'09 Proceedings of the Eleventh conference on Congress on Evolutionary Computation, s.s. : 277-284.
- [67] **Congdon, C. B., Aman, J. C., Nava, G. M., Gaskins, H. R., Mattingly, C. J.**, 2008. An Evaluation of Information Content as a Metric for the Inference of Putative Conserved Noncoding Regions in DNA Sequences Using a Genetic Algorithms Approach, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol:5, no:1,s.s.:1-14.
- [68] **Ferringer, M. P., Spencer, D. B., Reed, P.**, 2009. Many-objective reconfiguration of operational satellite constellations with the large-cluster epsilon non-dominated sorting genetic algorithm-II, CEC'09 Proceedings of the Eleventh conference on Congress on Evolutionary Computation, s.s. : 340-349.
- [69] **Gruau, F.**, 1994. Neural network synthesis using cellular encoding and the genetic algorithm, Phd Thesis, Université Claude Bernard Lyon 1.
- [70] **Whitley, D., Stakweather, T., and Bogart, C.**, 1990. Genetic algorithms and neural networks, optimizing connection and connectivity. Parallel Computing, vol:14, s.s: 347-361.
- [71] **Harp, S., Samad, T., Guha, A.**, 1989, Toward the genetic synthesis of neural Networks, in D. J. Schaffer, editor, 3rd Intern. Conf. on Genetic Algorithms, s.s: 360-369.
- [72] **Kitano, H.**, 1990. Designing neural networks using genetic algorithms with graph generation system, Complex Syst., vol. 4, no. 4, s.s: 461 – 476.
- [73] **Söke, A.**, 2003. Genetik algoritma ve benzetilmiş tavlama ile iki boyutlu giyotinsiz kesme problemlerine olasılıksal yaklaşım, YL Tezi, Kocaeli Üniversitesi.
- [74] **Beasley, D., Bull, D.R, Martin, R.R.**, 1993. An overview of Genetic Algorithms: Part 2. Research Topics , University Computing, cilt: 15(4), s.s: 170-181, UK.
- [75] **Schaffer, J. D., Whitley, D., Eshelman, L. J.**, 1992. Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art. In Proceedings of COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, IEEE.

- [76] **Dasgupta D., McGregor R. D.**, 1992. Designing Application-Specific Neural Networks using the Structured Genetic Algorithm, Combinations of Genetic Algorithms and Neural Networks, COGANN-92
- [77] **Ileana, I., Rotar, C., Incze, A.**, 2004. The Optimization of Feed Forward Neural Networks Structure Using Genetic Algorithms. Proc. Int. Conf. on Theory and Applications of Mathematics and Informatics-ICTAMI, Greece, s.s: 223-234.
- [78] **Han, S., May, G. S.**, 1996. Optimization of Neural Network Structure and Learning Parameters Using Genetic Algorithms. In Proceedings of the Eighth IEEE International Conference on Tools with Artificial Intelligence, pages 200–206, Toulouse, France, 16.-19. November.
- [79] **Ko, Y., Moon, P., Kim, C. E., Ham, M., Myoung, J., Yun, I.**, 2009. Modeling and optimization of the growth rate for ZnO thin films using neural networks and genetic algorithms, Expert Systems with Applications, Volume 36, Issue 2, Part 2, March, Elsevier, Pages 4061-4066.
- [80] **Thakur R., Singh V. K., Singh M. P.**, 2010. Evolutionary Design Of Fuzzy Logic Controllers With The Techniques Artificial Neural Network And Genetic Algorithm For Cart-Pole Problem. Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on.
- [81] **Doulamis, N.D., Doulamis, A.D.**, 2002. Optimal Distribution Transformers Assembly Using An Adaptable Neural Network –Genetic Algorithm Scheme, vol:5.
- [82] **Laisheng X., Zhengxia W., Xiaohong P.**, 2009. Research on Congestion Control Model and Algorithm for High-Speed Network Based on Genetic Neural Network and Intelligent PID.
- [83] **Xufeng J., Changying G., Yuan Z., Jianbo W.**, 2010. Application of Genetic Algorithm-Based Artificial Neural Network in Prediction of Aircraft Engine Wear. Digital Manufacturing and Automation (ICDMA), 2010 International Conference on, ss: 260-262.
- [84] **Li Y. J., Sun X. Q., Zhang L. M.**, 2010. Logistics Forecasting Technology by RBF Neural Network Trained by Genetic Algorithm. s.s: V7-513 - V7-516.
- [85] **Kim D. O.**, 2007. Ant system for a Multi-vehicle Routing Problem. Int. Conf. On Convergence Information Technology, s.s:691-696.
- [86] **Gilmore P., Gomory R.**, 1964. Sequencing a one-state-variable machine: A solvable case of the traveling salesman problem. Operations Research, 12:655–679.
- [87] **Ratliff H., Rosenthal A.**, 1981. Order-picking in a rectangular warehouse: A solvable case for the traveling salesman problem. PDRC Report Series No. 81-10. Georgia Institute of Technology, Atlanta, Georgia.
- [88] **Bland R. E., Shallcross D. F.**, 1987. Large traveling salesman problem arising from experiments in x-ray crystallography: a preliminary report on computation. Technical Report No. 730, School of OR/IE, Cornell University, Ithaca, New York.

- [89] **Koç İ. O.**, 2007. Gezin Satıcı Problemi için Çok Popülasyonlu Paralel Bir Genetik Algoritma Tasarımı, geliştirilmesi ve Analizi, Doktora Tezi, Eskişehir Osmangazi Üniversitesi.
- [90] **Brady, R. M.**, 1985. Optimization strategies gleaned from biological evolution, Nature, 317, 804-806.
- [91] **Aytekin M. A., Kalaycı T. E.**, 2010. Gezin Satıcı Probleminin İkili Kodlanmış Genetik Algoritmalarla Çözümünde Yeni Bir Yaklaşım, Akademik Bilişim 2010 (AB'10).



## EKLER

**EK A.1 :**  $V_i, Y_i, Z_i$  katsayılarının açık halleri

$$V_1 = 2m_1 \cos \beta_1 \{(O_1)_x - (O_2)_x\} + 2m_1 \sin \beta_1 \{(O_1)_y - (O_2)_y\}$$

$$V_2 = -2m_2 \cos \beta_2 \{(O_1)_x - (O_2)_x\} + 2m_2 \sin \beta_2 \{(O_1)_y - (O_2)_y\}$$

$$V_3 = -2m_1 m_2 \cos(\beta_1 - \beta_2)$$

$$V_4 = -2m_1 m_2$$

$$V_5 = \{(O_1)_x - (O_2)_x\}^2 + \{(O_1)_y - (O_2)_y\}^2 + m_1^2 + m_2^2 - l_1^2$$

$$Y_1 = 2m_2 \cos \beta_2 \{(O_2)_x - (O_3)_x\} + 2m_2 \sin \beta_2 \{(O_2)_y - (O_3)_y\}$$

$$Y_2 = -2m_3 \cos \beta_3 \{(O_2)_x - (O_3)_x\} - 2m_3 \sin \beta_3 \{(O_2)_y - (O_3)_y\}$$

$$Y_3 = -2m_2 m_3 \cos(\beta_2 - \beta_3)$$

$$Y_4 = -2m_2 m_3$$

$$Y_5 = \{(O_2)_x - (O_3)_x\}^2 + \{(O_2)_y - (O_3)_y\}^2 + m_2^2 + m_3^2 - l_2^2$$

$$Z_1 = 2m_3 \cos \beta_3 \{(O_3)_x - (O_1)_x\} + 2m_3 \sin \beta_3 \{(O_3)_y - (O_1)_y\}$$

$$Z_2 = -2m_1 \cos \beta_1 \{(O_3)_x - (O_1)_x\} - 2m_1 \sin \beta_1 \{(O_3)_y - (O_1)_y\}$$

$$Z_3 = -2m_3 m_1 \cos(\beta_3 - \beta_1)$$

$$Z_4 = -2m_3 m_1$$

$$Z_5 = \{(O_3)_x - (O_1)_x\}^2 + \{(O_3)_y - (O_1)_y\}^2 + m_3^2 + m_1^2 - l_3^2$$

**EK A.2 :**  $K_i, L_i, M_i$  katsayılarının açık halleri

$$K_1 = -V_1 - V_2 + V_3 + V_5$$

$$K_2 = V_1 - V_2 - V_3 + V_5$$

$$K_3 = 4V_4$$

$$K_4 = -V_1 + V_2 - V_3 + V_5$$

$$K_5 = V_1 + V_2 + V_3 + V_5$$

$$L_1 = -Y_1 - Y_2 + Y_3 + Y_5$$

$$L_2 = Y_1 - Y_2 - Y_3 + Y_5$$

$$L_3 = 4Y_4$$

$$L_4 = -Y_1 + Y_2 - Y_3 + Y_5$$

$$L_5 = Y_1 + Y_2 + Y_3 + Y_5$$

$$M_1 = -Z_1 - Z_2 + Z_3 + Z_5$$

$$M_2 = Z_1 - Z_2 - Z_3 + Z_5$$

$$M_3 = 4Z_4$$

$$M_4 = -Z_1 + Z_2 - Z_3 + Z_5$$

$$M_5 = Z_1 + Z_2 + Z_3 + Z_5$$



**EK A.3 :**  $R_i$  katsayılarının açık halleri

$$R_1 = (L_2K_1 - K_4L_1)^2$$

$$R_2 = 2(L_2K_1 - K_4L_1)(K_2L_2 - K_5L_1) + L_1L_2K_3^2$$

$$R_3 = (K_2L_2 - K_5L_1)^2$$

$$R_4 = -K_1K_3L_2L_3 - L_1L_3 - L_1L_3K_3K_4$$

$$R_5 = -K_2K_3L_2L_3 - L_1L_3K_3K_5$$

$$R_6 = 2(L_2K_1 - L_1K_4)(L_5K_1 - L_4K_4) + L_3^2K_1K_4$$

$$R_7 = 2(L_2K_1 - L_1K_4)(K_2L_5 - K_5L_4) + 2(L_2K_2 - K_5L_4)(K_1L_5 - K_4L_4) + L_3^2K_1K_5 + L_3^2K_2K_4 + K_3^2L_1L_5 + K_3^2L_2L_4$$

$$R_8 = 2(K_2L_2 - K_5L_1)(K_2L_5 - K_5L_4) + L_3^2K_2K_5$$

$$R_9 = -L_3L_4K_3K_4 - L_3L_5K_1K_3$$

$$R_{10} = -L_3L_4K_3K_5 - L_3L_5K_2K_3$$

$$R_{11} = (L_5K_1 - L_4K_4)^2$$

$$R_{12} = 2(K_1L_5 - L_4K_4)(K_2L_5 - K_5L_4) + L_4L_5K_3^2$$

$$R_{13} = (K_2L_5 - K_5L_4)^2$$

#### EK A.4 : 3x3 SPM ters kinematik analiz kodları

```
alfa = pi/10;
beta = pi/15;
gama = pi/12;
z_first = 300;
x_max = 500;
y_max = 500;
delta_z = 5;
z_enable = true;
Leg_Lengths = [0 0 0 0 0 0];
leg_min=325;
leg_max=360;
point_count=0;
M=10000;
N=6;
A=zeros(M,N);

%R=[(cos(beta)*cos(gama)) (cos(gama)*sin(alfa)*sin(beta)-sin(gama)*cos(alfa))
(cos(gama)*sin(beta)*cos(alfa)+sin(gama)*sin(alfa));
    (sin(gama)*cos(beta)) (sin(alfa)*sin(beta)*sin(gama)+cos(gama)*cos(alfa))
(sin(gama)*sin(beta)*cos(alfa)-cos(gama)*sin(alfa));
    %(-sin(beta)) (cos(beta)*sin(alfa)) (cos(beta)*cos(alfa))];

R=[1 0 0; 0 1 0; 0 0 1];
B1=[150;0;0];
B2=[-75;129.9;0];
B3=[-75;-129.9;0];
z = z_first;

while z_enable==true
    for i=1:1:x_max
        for j=1:1:y_max
```

```

t=[i; j; z];
p1=[-10;-10;0];
p2=[10;-10;0];
p3=[10;10;0];

P1=R*p1+t;
P2=R*p2+t;
P3=R*p3+t;

leg1_vector= P1-B1;
leg2_vector= P1-B2;
leg3_vector= P2-B2;
leg4_vector= P2-B3;
leg5_vector= P3-B3;
leg6_vector= P3-B1;

leg_lengths=[norm(leg1_vector) norm(leg2_vector) norm(leg3_vector)
norm(leg4_vector) norm(leg5_vector) norm(leg6_vector)];
count=0;
for a=1:1:6
    if leg_lengths(a)<leg_max && leg_lengths(a)>leg_min
        count=count+1;
    end %if
end

if count==6
    point_count=point_count+1
    A(point_count,1)=i;
    A(point_count,2)=j;
    A(point_count,3)=z;
    A(point_count,4)=leg_lengths(1);
    A(point_count,5)=leg_lengths(2);
    A(point_count,6)=leg_lengths(3);
end

```

```

A(point_count,7)=leg_lengths(4);
A(point_count,8)=leg_lengths(5);
A(point_count,9)=leg_lengths(6);

dosya=fopen('D:\veri.txt','a+');
fprintf(dosya,'\n\n >>%d- [%d %d %d %.3f %.3f %.3f %.3f %.3f
%.3f]\n',point_count,i,j,z,leg_lengths(1),leg_lengths(2),leg_lengths(3),leg_lengths(4)
,leg_lengths(5),leg_lengths(6));
fclose(dosya);
end

end

end
%z nin olabilirligini kontrol et
z=z+delta_z;
t=[0; 0; z];
p1=[30;0;0];
p2=[-30;0;0];
p3=[0;30;0];

P1=R*p1+t;
P2=R*p2+t;
P3=R*p3+t;

leg1_vector= P1-B1;
leg2_vector= P1-B2;
leg3_vector= P2-B2;
leg4_vector= P2-B3;
leg5_vector= P3-B3;
leg6_vector= P3-B1;

```

```
leg_lengths=[norm(leg1_vector) norm(leg2_vector) norm(leg3_vector)
norm(leg4_vector) norm(leg5_vector) norm(leg6_vector)];
control=0;
for a=1:1:6
    if leg_lengths(a)<leg_max && leg_lengths(a)>leg_min
        control=1;
    else
        control=0;
        break;
    end %if
end
if control==0
    z_enable=false;
end
end
```

**EK A.5 :** 3x3 SPM düz kinematik analiz kodları

```
clear
```

```
i=[1;0;0];
```

```
j=[0;1;0];
```

```
k=[0;0;1];
```

```
syms x1;
```

```
syms x2;
```

```
syms x3;
```

```
syms x4;
```

```
% Sabit platformun ilk konumları
```

```
B30=0.175;
```

```
B31=0;
```

```
B32=0;
```

```
B10=-0.0875;
```

```
B11=0.15155;
```

```
B12=0;
```

```
B20=-0.0875;
```

```
B21=-0.15155;
```

```
B22=0;
```

```
B1=[ B10; B11;B12];
```

```
B2=[B20;B21;B22];
```

```
B3=[ B30; B31; B32];
```

```
L1=0.41;
```

```
L2=0.42;
```

```
L3=0.43;
```

```
L4=0.44;
```

```
L5=0.43;
```

```
L6=0.42;
```

%Üst platformun kenar uzunlukları

l1=0.260;

l2=0.260;

l3=0.260;

%Alt platformun kenar uzunlukları(geometrik ifadelerden)

x=norm(B1-B3);

y=norm(B2-B3);

z=norm(B2-B1);

c1=(z^2+L5^2-L6^2)/(2\*z);

c2=z-c1;

m3=(L5^2-c1^2)^0.5;

b2=(y^2+L4^2-L3^2)/(2\*y);

b1=y-b2;

m2=(L4^2-b2^2)^0.5;

a2=(x^2+L2^2-L1^2)/(2\*x);

a1=x-a2;

m1=(L2^2-a2^2)^0.5;

% O noktalarının konumları

PO3=[-0.150;0;0];

PO2=[0.075;-0.1299;0];

PO1=[0.075;0.1299;0];

% Beta açılarının hesaplanması

v1 = cross((B2-B1), k);

n1=dot(v1,j);

beta1=asin(n1/norm(v1));

v2=cross((B3-B2), k);

n2=dot(v2,j);

beta2=asin(n2/norm(v2));

v3=cross((B1-B3), k);

$$n3=\text{dot}(v3,i);$$

$$\text{beta3}=\text{acos}(n3/\text{norm}(v3));$$

%Düz kinematik çözümü

$$V1=2*m3*\cos(\text{beta1})*(B30+(c1*(B10-B30)/\text{norm}(B1-B3))-B30-(b2*(B20-B30)/\text{norm}(B2-B3)))+2*m3*\sin(\text{beta1})*(B31+(c1*(B11-B31)/\text{norm}(B1-B1))-B31-(b2*(B21-B31)/\text{norm}(B2-B3)));$$

$$V2=-2*m2*\cos(\text{beta2})*(B30+(c1*(B10-B30)/\text{norm}(B1-B3))-B30-(b2*(B20-B30)/\text{norm}(B2-B3)))-2*m2*\sin(\text{beta2})*(B31+(c1*(B11-B31)/\text{norm}(B1-B3))-B31-(b2*(B21-B31)/\text{norm}(B2-B3)));$$

$$V3=-2*m3*m2*\cos(\text{beta1}-\text{beta2});$$

$$V4=-2*m3*m2;$$

$$V5=(B30+(c1*(B10-B30)/\text{norm}(B1-B3))-B30-(b2*(B20-B30)/\text{norm}(B2-B3)))^2+(B31+(c1*(B11-B31)/\text{norm}(B1-B3))-B31-(b2*(B21-B31)/\text{norm}(B2-B3)))^2+m3^2+m2^2-l1^2;$$

$$Y1=2*m2*\cos(\text{beta2})*(B30+(b2*(B20-B30)/\text{norm}(B2-B3))-B20-(a2*(B10-B20)/\text{norm}(B1-B2)))+2*m2*\sin(\text{beta2})*(B31+(b2*(B21-B31)/\text{norm}(B2-B3))-B21-(a2*(B11-B21)/\text{norm}(B1-B2)));$$

$$Y2=-2*m1*\cos(\text{beta3})*(B30+(b2*(B20-B30)/\text{norm}(B2-B3))-B20-(a2*(B10-B20)/\text{norm}(B1-B2)))-2*m1*\sin(\text{beta3})*(B31+(b2*(B21-B31)/\text{norm}(B2-B3))-B21-(a2*(B11-B21)/\text{norm}(B1-B2)));$$

$$Y3=-2*m2*m1*\cos(\text{beta2}-\text{beta3});$$

$$Y4=-2*m2*m1;$$

$$Y5=(B30+(b2*(B20-B30)/\text{norm}(B2-B3))-B20-(a2*(B10-B20)/\text{norm}(B1-B2)))^2+(B31+(b2*(B21-B31)/\text{norm}(B2-B3))-B21-(a2*(B11-B21)/\text{norm}(B1-B2)))^2+m2^2+m1^2-l2^2;$$

$$Z1=2*m1*\cos(\text{beta3})*(B20+(a2*(B10-B20)/\text{norm}(B1-B2))-B30-(c1*(B10-B30)/\text{norm}(B1-B3)))+2*m1*\sin(\text{beta3})*(B21+(a2*(B11-B21)/\text{norm}(B1-B2))-B31-(c1*(B11-B31)/\text{norm}(B1-B3)));$$

$$Z2=-2*m3*\cos(\text{beta1})*(B20+(a2*(B10-B20)/\text{norm}(B1-B2))-B30-(c1*(B10-B30)/\text{norm}(B1-B3)))-2*m3*\sin(\text{beta1})*(B21+(a2*(B11-B21)/\text{norm}(B1-B2))-B31-(c1*(B11-B31)/\text{norm}(B1-B3)));$$



$$Z3=-2*m3*m1*cos(beta3-beta1);$$

$$Z4=-2*m3*m1;$$

$$Z5=(B20+(a2*(B10-B20)/norm(B1-B2))-B30-(c1*(B10-B30)/norm(B1-B3)))^2+(B21+(a2*(B11-B21)/norm(B1-B2))-B31-(c1*(B11-B31)/norm(B1-B3)))^2+m3^2+m1^2-l3^2;$$

$$K1=-V1-V2+V3+V5;$$

$$K2=V1-V2-V3+V5;$$

$$K3=4*V4;$$

$$K4=-V1+V2-V3+V5;$$

$$K5=V1+V2+V3+V5;$$

$$L1=-Y1-Y2+Y3+Y5;$$

$$L2=Y1-Y2-Y3+Y5;$$

$$L3=4*Y4;$$

$$L4=-Y1+Y2-Y3+Y5;$$

$$L5=Y1+Y2+Y3+Y5;$$

$$M1=-Z1-Z2+Z3+Z5;$$

$$M2=Z1-Z2-Z3+Z5;$$

$$M3=4*Z4;$$

$$M4=-Z1+Z2-Z3+Z5;$$

$$M5=Z1+Z2+Z3+Z5;$$

% Bezout matrisinin oluşturulması

$$\text{Bezout1\_11}=[(K1*x1^2+K2) (K4*x1^2+K5); (L1*x3^2+L4) (L2*x3^2+L5)];$$

% x2 elimine edilmesi için yazılan bezout

$$\text{determinant1\_11}=\det(\text{Bezout1\_11});$$

$$\text{Bezout1\_12}=[(L3*x3) (K3*x1); (L1*x3^2+L4) (K1*x1^2+K2)];$$

$$\text{determinant1\_12}=\det(\text{Bezout1\_12});$$

$$\text{Bezout1\_21}=[(K3*x1) (K4*x1^2+K5); (L3*x3) (L2*x3^2+L5)];$$

$$\text{determinant1\_21}=\det(\text{Bezout1\_21});$$

```

Bezout1_22=[(K1*x1^2+K2) (K4*x1^2+K5); (L1*x3^2+L4) (L2*x3^2+L5)];
determinant1_22=det(Bezout1_22);
Bezout1=[determinant1_11 determinant1_12; determinant1_21 determinant1_22];
f1=det(Bezout1);

```

%f1 i hesaplamak için

```

R1=((L2*K1)-(K4*L1))^2;
R2= 2*((L2*K1)-(K4*L1))*((K2*L2)-(K5*L1))+(L1*L2*(K3^2));
R3=((K2*L2)-(K5*L1))^2;
R4=-((K1*K3*L2*L3)-(L1*L3*K3*K4));
R5=-((K2*K3*L2*L3)-(L1*L3*K3*K5));
R6=2*(L2*K1-L1*K4)*(L5*K1-L4*K4)+((L3^2)*K1*K4);
R7=2*((L2*K1)-(L1*K4))*((K2*L5)-(K5*L4))+2*(L2*K2-K5*L1)*(K1*L5-
K4*L4)+(L3^2*K1*K5)+(L3^2*K2*K4)+(K3^2*L1*L5)+(K3^2*L2*L4);
R8=2*(K2*L2-K5*L1)*(K2*L5-K5*L4)+L3^2*K2*K5;
R9=-((L3*L4*K3*K4)-(L3*L5*K1*K3));
R10=-((L3*L4*K3*K5)-(L3*L5*K2*K3));
R11=(L5*K1-L4*K4)^2;
R12=2*(K1*L5-L4*K4)*(K2*L5-K5*L4)+(L4*L5*K3^2);
R13= (K2*L5-K5*L4)^2;

```

```

P1=(R1*x1^4)+(R2*x1^2)+R3;
P2=(R4*x1^3)+R5*x1;
P3=(R6*x1^4)+(R7*x1^2)+R8;
P4=(R9*x1^3)+(R10*x1);
P5=(R11*x1^4)+(R12*x1^2)+R13;

```

```

S1=(M1*(x1^2))+M4;
S2=M3*x1;
S3=(M2*(x1^2))+M5;

```

% x3 elimine edilmesi için yazılan bezout

```

Bezout2=[(P1*S3-P3*S1) (P2*S3-P4*S1-P3*S2) (-P5*S1-P4*S2) (P5*S2); (P1*S2-
P2*S1) (P1*S3-P3*S1) (-P4*S1) (-P5*S1); (S1) (S2) (S3) 0; 0 (S1) (S2) (S3)];
f2=det(Bezout2);

```

```

%x1 in deęerleri

```

```

ex1=double(solve(f2,x1));

```

```

%x1 in imajiner ve negatif deęerleri elimine ediliyor.

```

```

yer=0;
dizi=zeros(1,16);
for t=1:1:16
    kok = ex1(t,1);
    img = imag(kok);
    sayi=real(kok);
    if (sayi>0 && img==0) || (img>-1 && img<1 && sayi>0)
        yer=yer+1;
        dizi(1,yer)=sayi;
    end
end
x1_uygun=zeros(1,yer);

```

```

for t=1:1:yer
    x1_uygun(1,t)=dizi(1,t);
end

```

```

% x3 deęerlerinin bulunması

```

```

yer1=0;
for t=1:1:yer
    x1=x1_uygun(1,t);
    z=((M1*x1^2)+M4)*x3^2+(M3*x1)*x3+(M2*x1^2+M5);
    ex3=double(solve(z,x3));

```

```

for q=1:1:2
    kok= ex3(q,1);
    img = imag(kok);
    sayi=real(kok);
    if (sayi>0 && img==0) || (img>-1 && img<1 && sayi>0)
        yer1=yer1+1;
        dizi(1,yer1)=sayi;
    end
end
end
end
x3_uygun=zeros(1,yer1);

for t=1:1:yer1
    x3_uygun(1,t)=dizi(1,t);
end

%x2 değerlerinin bulunması
yer2=0;
for t=1:1:yer
    x1=x1_uygun(1,t);
    g=(K1*x1^2+K2)*x2^2+(K3*x1)*x2+(K4*x1^2+K5);
    ex2=double(solve(g,x2));
    for q=1:1:2
        kok= ex2(q,1);
        img = imag(kok);
        sayi=real(kok);
        if (sayi>0 && img==0) || (img>-1 && img<1 && sayi>0)
            yer2=yer2+1;
            dizi(1,yer2)=sayi;
        end
    end
end
end
end

```

```

x2_uygun=zeros(1,yer2);
for t=1:1:yer2
    x2_uygun(1,t)=dizi(1,t);
end

```

% x2 ve x3 un yerine konulup saglama yapılması

```

h=zeros(1,(yer1*yer2));
yer3=0;
temp=100;
for t=1:1:yer1
    for q=1:1:yer2
        x2=x2_uygun(1,q);
        x3=x3_uygun(1,t);
        yer3=yer3+1;
        h(1,yer3)=(L3*x3^2+L6)*x2^2+(L5*x3)*x2+(L4*x3^2+L1);
        if h(1,yer3)<temp && h(1,yer3)>0
            temp = h(1,yer3);
            x2_indis=q;
            x3_indis=t;
        end
        if h(1,yer3)>(temp*-1) && h(1,yer3)<0
            temp = h(1,yer3)*-1;
            x2_indis=q;
            x3_indis=t;
        end
    end
end
end

```

```

x3=x3_uygun(1,x3_indis);
z_temp=100;
for t=1:1:yer
    x1=x1_uygun(1,t);
    z=((M1*x1^2)+M4)*x3^2+(M3*x1)*x3+(M2*x1^2+M5);

```

```
if abs(z)<z_temp;
```

```
x1_indis=t;
```

```
z_temp=abs(z);
```

```
end
```

```
end
```

```
cozum=zeros(1,3);
```

```
cozum(1,1)=x1_uygun(1,x1_indis);
```

```
cozum(1,2)=x2_uygun(1,x2_indis);
```

```
cozum(1,3)=x3_uygun(1,x3_indis);
```

```
fi_1=2*atan(cozum(1,1));
```

```
fi_2=2*atan(cozum(1,2));
```

```
fi_3=2*atan(cozum(1,3));
```

```
% omegalar
```

```
w1=(cos(beta1)*cos(fi_1)*i)+(sin(beta1)*j)+sin(fi_1)*k;
```

```
w2=(cos(beta2)*cos(fi_2)*i)+(sin(beta2)*j)+sin(fi_2)*k;
```

```
w3=(cos(beta3)*cos(fi_3)*i)+(sin(beta3)*j)+sin(fi_3)*k;
```

```
%p'ler
```

```
M1=PO1+m1*w1;
```

```
M2=PO2+m2*w2;
```

```
M3=PO3+m3*w3;
```

```
%ağırlık merkezi
```

```
G=(M1+M2+M3)/3
```

```
p1=[0.05;0.0866;0];
```

```
p2=[0.05;-0.0866;0];
```

```
p3=[-0.1;0;0];
```

```
syms alpha;
```

```
syms beta;
```

```
syms gama;
```

```
R=[cos(gama)*cos(beta) -sin(gama)*cos(beta) sin(beta);
```

```
sin(alpha)*sin(beta)*cos(gama)+cos(alpha)*sin(gama) cos(alpha)*cos(gama)-
```

```
sin(gama)*sin(beta)*sin(alpha) -sin(alpha)*cos(beta); -
```

```
cos(alpha)*sin(beta)*cos(gama)+sin(alpha)*sin(gama)
```

```
sin(beta)*cos(alpha)*sin(gama)+sin(alpha)*cos(gama) cos(beta)*cos(alpha)];
```

```
x=R*p1;
```

```
y=x+G;
```

```
R11=0.05*cos(gama)*cos(beta)-0.0866*sin(gama)+0.433;
```

```
R12=0.05*sin(alpha)*sin(beta)*cos(gama)+0.05*cos(alpha)*sin(gama)+0.0866*cos(alpha)*cos(gama)-0.0866*sin(gama)*sin(beta)*sin(alpha)+0.0316;
```

```
R13=-
```

```
0.05*cos(alpha)*sin(beta)*cos(gama)+0.05*sin(alpha)*sin(gama)+0.0866*cos(alpha)*sin(beta)*sin(gama)+0.0866*sin(alpha)*cos(gama)+0.1571;
```

```
eq1=R11-M1(1,1);
```

```
eq2=R12-M1(2,1);
```

```
eq3=R13-M1(3,1);
```





## **ÖZGEÇMİŞ**

**Ad Soyad: Zeynep Ekiciođlu**

**Dođum Yeri ve Tarihi: 1986 - Kırřehir**

**Adres: Yıldız Teknik Üniversitesi Beřiktař / İstanbul**

**Lisans Üniversitesi: Kocaeli Üniversitesi Mekatronik Müh.**

### **Yayın Listesi:**

- Yıldız İ., Ömürlü V.E., **Ekiciođlu Z.**, Güney A., 6 Serbestlik Dereceli Paralel Mekanizmadaki İleri Kinematik Analiz Yöntemleri, TOK, 2010.