T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

UZAYSAL PARALEL ROBOTLARDA ARDIŞIK KUADRATİK PROGRAMLAMA TABANLI ÇOK PARAMETRELİ KİNEMATİK TASARIM OPTİMİZASYONU

CENK ERYILMAZ

DOKTORA TEZİ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI
MAKİNE TEORİSİ VE KONTROL PROGRAMI

DANIŞMAN
PROF. DR. VASFİ EMRE ÖMÜRLÜ

İSTANBUL, 2019

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

UZAYSAL PARALEL ROBOTLARDA ARDIŞIK KUADRATİK PROGRAMLAMA TABANLI ÇOK PARAMETRELİ KİNEMATİK TASARIM OPTİMİZASYONU

Cenk ERYILMAZ tarafından hazırlanan tez çalışması 24/05/2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Vasfi Emre ÖMÜRLÜ

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. Vasfi Emre ÖMÜRLÜ

Yıldız Teknik Üniversitesi _____

Prof. Dr. İbrahim ÖZKOL

İstanbul Teknik Üniversitesi _____

Doç. Dr. Akın DELİBAŞI

Yıldız Teknik Üniversitesi _____

Doç. Dr. Semih SEZER

Yıldız Teknik Üniversitesi _____

Doç. Dr. Erkan KAPLANOĞLU

Marmara Üniversitesi _____

ÖNSÖZ

Bu çalışmada 3x3 yapısal mimariye sahip bir Stewart Platformu'nun hareket kabiliyeti ve çalışma uzayı açısından optimizasyonu amacıyla çeşitli algoritmalar geliştirilerek yeni bir yöntem önerilmiştir. Yöntemde farklı alt fonksiyonlar kullanan genel bir optimizasyon algoritması ile kinematik parametrelerin farklı kombinasyonları optimize edilerek sonuçlar hareket kabiliyeti ve çalışma uzayı açısından değerlendirilmiştir. Parametrelerin optimizasyonu için ardışık kuadratik programlama ve iç noktalar yöntemi kullanılarak sonuçlar karşılaştırılmıştır. Optimizasyon hesaplamaları Matlab Optimization Toolbox ile gerçekleştirilmiştir. Optimizasyonu etkileyen faktörlerin neler olduğu ve etkileri tartışılmıştır.

Bu çalışmayı gerçekleştirirken yanımda olan aileme, eşime, gerçekleştirmemde beni cesaretlendiren destek olan, bilgi ve tecrübelerini paylaşan, danışmanım Sayın Prof. Dr. Vasfi Emre Ömürlü'ye, yardımlarından ötürü çok teşekkür ederim.

Mayıs, 2019

Cenk ERYILMAZ

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ

l_i	Bacak uzunlukları
{P}	Hareketli platformun koordinat sistemi
{B}	Sabit platformun koordinat sistemi
x	Hareketli platformun X ekseni
y	Hareketli platformun Y ekseni
z	Hareketli platformun Z ekseni
α	Hareketli platformun X ekseni etrafındaki dönme açısı
β	Hareketli platformun Y ekseni etrafındaki dönme açısı
γ	Hareketli platformun Z ekseni etrafındaki dönme açısı
J	Jakobiyen matrisi
Z	Jakobiyen matrisinin koşul sayısı
O_p	Hareketli platformun ağırlık merkezi
O_b	Sabit platformun ağırlık merkezi
λ_i	(X_p, O_pP_i) arasındaki açı
Λ_i	(X_b, O_bB_i) arasındaki açı
r_b	Sabit platform ve bacakları birbirine bağlayan mafsalların merkezinden geçen dairenin yarıçapı
r_p	Hareketli platform ve bacakları birbirine bağlayan mafsalların merkezinden geçen dairenin yarıçapı
θ_b	Sabit platform üzerindeki bacakların bağlantı noktaları arasındaki açı
θ_p	Hareketli platform üzerindeki bacakların bağlantı noktaları arasındaki açı
R	Rotasyon matrisi

KISALTMA LİSTESİ

CRS	Controlled Random Search
DOF	Degree Of Freedom
GA	Genetic Algoritm
GCI	Global Condition Index
GSI	Global Stiffness Index
LCI	Local Condition Index
PRS	Paralel Robot Sistemi
PSO	Particle Swarm Optimization
RPR	Rotary Prismatic Rotary
RPS	Rotary Prismatic Spherical
SP	Stewart Platform
SPM	Stewart Platform Mekanizması
SQP	Sequential Quadratic Programming
UPS	Universal Prismatic Spherical
UPU	Universal Prismatic Universal

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ÖZET

UZAYSAL PARALEL ROBOTLARDA ARDIŞIK KUADRATİK PROGRAMLAMA TABANLI ÇOK PARAMETRELİ KİNEMATİK TASARIM OPTİMİZASYONU

Cenk ERYILMAZ

Makine Mühendisliği Makine Teorisi ve Kontrol Programı

Doktora Tezi

Tez Danışmanı: Prof. Dr. Vasfi Emre ÖMÜRLÜ

Farklı yapısal konfigürasyondaki paralel robotik sistemler (PRS) ile ilgili birçok araştırma vardır. En popüler olanı bu tezde de kullandığımız iyi bilinen Stewart Platform (SP) mekanizmasıdır. Bir paralel robot sistemi tasarlarken tekil konfigüsasyonlardan kaçınıp erişilebilir çalışma uzayını maksimize etmek için kinematik parametreleri belirlemek çok önemlidir.

İlgili literatürlerde GCI/LCI (Global koşul /Yerel koşul) indeksi mekanizmanın becerikliliği hakkında fikir verirken genellikle en çok kullanılan performans indeksleridir. Genetik algoritma (GA), kontrollü rastgele araştırma (CRS), parçacık sürü optimizasyonu (PSO) gibi optimizasyon metotları literatürde mevcuttur.

Biz bu tezde farklı olarak, 6-DOF 3x3 UPU SP'nin kinematik parametrelerini optimize etmek, maksimum çalışma uzayını minimum koşul sayılarıyla sağlamak için ardışık kuadratik programlama (SQP) metodunu kullandık. Sonuçları iç noktalar yöntemiyle karşılaştırdık. Sabit ve hareketli platformun yarı çapları ile birlikte bacak uzunlukları çok amaçlı optimizasyon probleminin optimize edilecek parametreleri olarak seçilmiştir. Böylelikle optimizasyon farklı aşamalarda her seviyede optimize edilerek kinematik parametre sayısı arttırılarak gerçekleştirilmiştir. Geliştirmiş olduğumuz optimizasyon algoritmaları farklı sayıdaki kinematik parametreler için kullanılabilir olması açısından çalışmaya özgün bir değer katmıştır. Sonuç olarak üç grup kinematik parametreyi aynı anda SQP tekniğini kullarak optimize ederek ve sonuçları iç noktalar yöntemiyle karşılaştırarak PRS için en iyi sonuçları sunduk.

Anahtar Kelimeler: Optimizasyon, ardışık kuadratik programlama (SQP), Stewart Platform, kinematik parametreler, koşul sayısı

ABSTRACT

Multi-parameter Kinematic Optimization of Spatial Parallel Robots by Sequential Quadratic Programming

Cenk ERYILMAZ

Department of Mechanical Engineering

Phd. Thesis

Advisor: Prof. Dr. Vasfi Emre ÖMÜRLÜ

There are many researches about parallel robotic systems (PRS) with various structural configurations. The most popular one is well known as Stewart Platform Mechanism (SP) which is the current subject in this thesis. While designing a PRS, determining kinematic parameters is very important in order to maximize reachable workspace while avoiding singular configurations.

In related literature, GCI/LCI (Global Condition Index/Local Condition Index) are commonly used performance indexes which give an idea about dexterity of a mechanism. Optimization methods like Genetic Algorithm (GA), Controlled Random Search (CRS), Particle Swarm Optimization (PSO) are present.

In this research differently, we used Sequential Quadratic Programming (SQP) method to optimize kinematic parameters of a 6-DOF 3x3 UPU SP in order to reach maximum workspace satisfying small condition numbers. We compared the results with interior point methods. The radius of mobile and base platforms, and also leg lengths are chosen to be kinematic parameters to be optimized as a multi-objective optimization problem. Thus, optimization is performed at different stages increasing the number of optimized kinematic parameters at each level. The optimization algorithms that we have developed have added value to the study because they can be used for a different number of kinematic parameters. Finally, optimizing three groups of kinematic parameters at once by using SQP technique and comparing with interior point method presented the best results for the PRS.

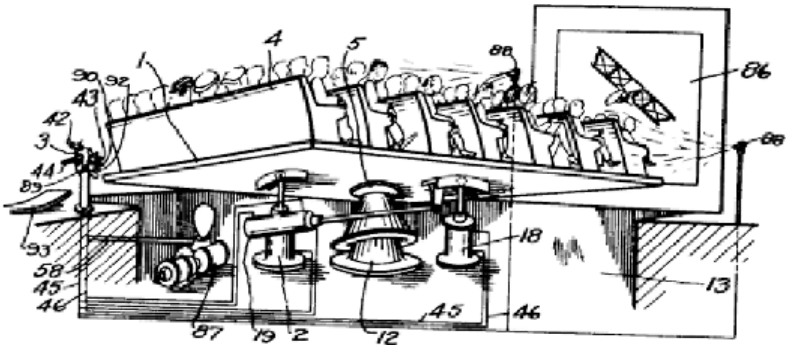
Keywords: Optimization, Sequential Quadratic Programming, Stewart Platform, Kinematic Parameters, Condition Number

BÖLÜM 1

GİRİŞ

Uzaydaki katı bir cisim farklı biçimlerde öteleme veya dönme hareketi yapabilir. Bu o cismin serbestlik derecesi olarak isimlendirilir. Üç boyutlu uzayda katı bir cismin serbestlik derecesi 6'yı aşamaz.

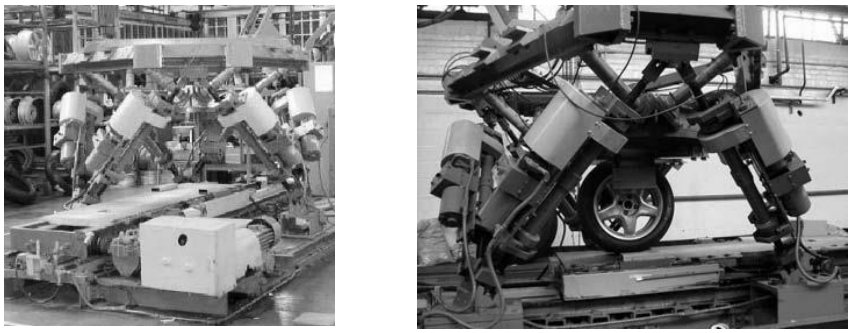
Her bir robot (kolu) için tanımlanan bu kola, bir eklem ile bağlı katı cisim sayısına bağlantı derecesi denir. Örneğin seri robotlarda her bir kolun bağlantı derecesi en fazla 2'dir. Taban ve (end effector) uç işlemci'nin bağlantı derecesi 1'dir. Böyle bir zincir, açık çevrim kinematik zincir diye isimlendirilir. Kapalı çevrim kinematik zincir, kollardan birinin (taban değil) bağlantı derecesi 3'e eşit veya daha büyük olur ise oluşur. Kapalı çevrim kinematikle ilgili ilk teorik problemler 1645'in başlarında Cristopher Wren, 1813'de Cauchy, 1867'de Lebesque, 1897'de Bricard tarafından ele alınmıştır [1].



Şekil 1.1 1928 'de J.E. Gwinnet tarafından önerilen küresel mekanizma [1]

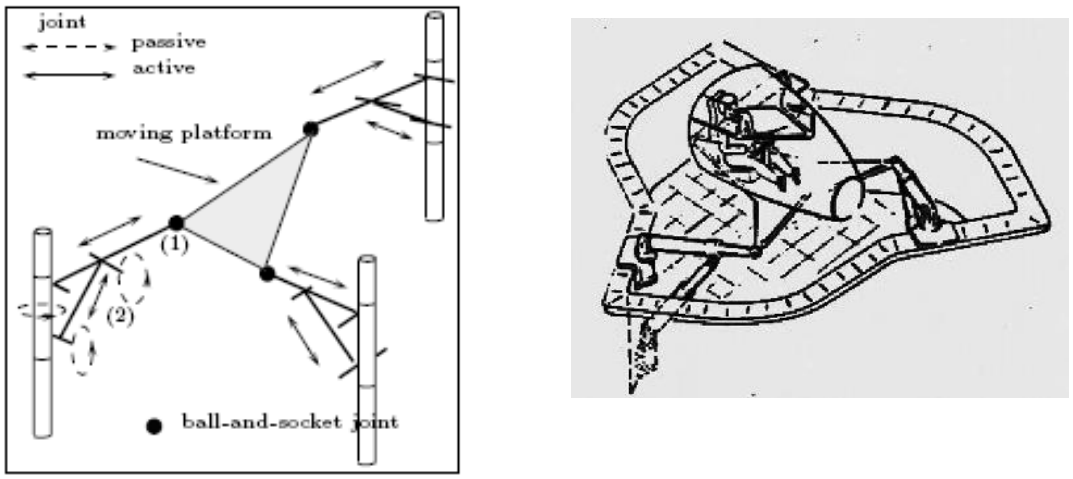
1947'de Gough, kapalı çevrim kinematik yapılı mekanizmaların temel prensiplerini ortaya koyar [1]. 1955'te lastiklerin aşınma ve yırtılma testlerinde kullanılmak üzere hareketli bir platform yapar [1]. Hareketli kısım altıgendir. Bütün köşeleri küresel

mafsalla kolların bir ucuna bağlıdır. Kolların diğer ucu ise kardan kavraması ile tabana bağlıdır. Bu sistem 2000 yılına kadar kullanılmış daha sonra müzeye konulmuştur.



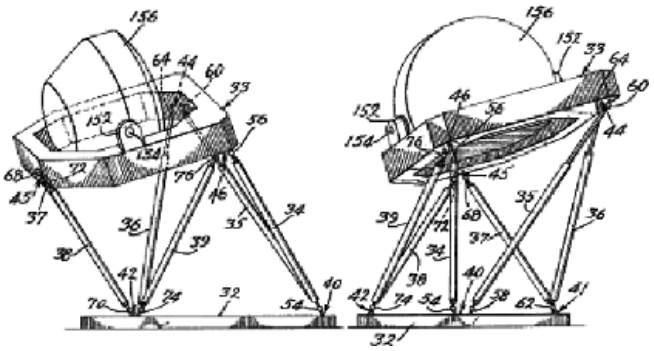
Şekil 1.2 Dunlop firmasında kullanılan Gough platformunun son prototibi [1]

1960'lar da havacılık endüstrisindeki gelişmelerle, pilot eğitimlerindeki maliyetlerin artışı, kullanılan malzemelerin uçmadan yüksek yüklerde yüksek dinamik koşullarda test edilmesi için test yapabilecek simülatör platformları gereksinimi, araştırmacıları uçak simülatörü için mekanizma arayışına itmiştir.



Şekil 1.3 1965'te Stewart 'ın uçak simulatörü için önerdiği mekanizma [1]

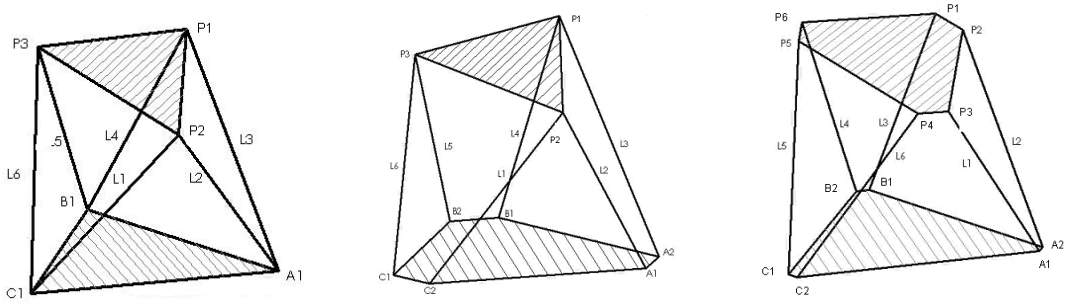
Şekil 1.4' te, Sikorsky helikopterleri için tasarlanmış 6 serbestlik dereceli uçuş simülatörü tasarlanmış ve 1967'de patenti alınmıştır. Bugünlerde halen uçuş simülatörleri için Gough ve Cappel'in platform yapısının prensipleri kullanılmaktadır.



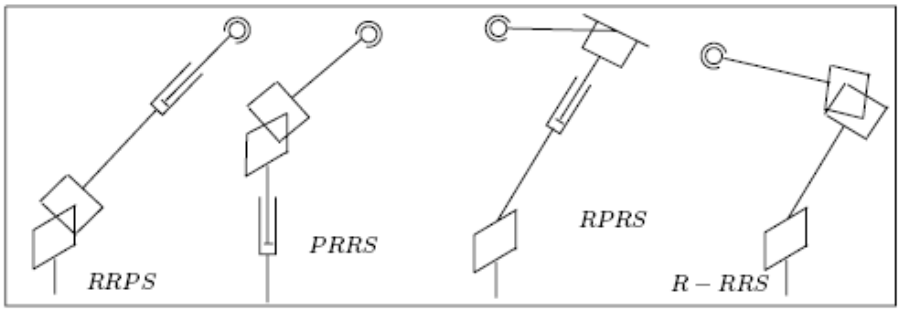
Şekil 1.4 K. Cappel (1967 de) in aldığı patentin bir çizimi [2]

Paralel robotlar uçuş simülatörü dışında da çeşitli uygulamalarda kullanılmaktadır. Örneğin L.V Pollard otomatik sprey boyama için paralel robot tasarlamıştır. Ancak bir prototipini üretmemiştir. Oğlu 1940'da patentini almıştır [3].

Seri robotların eksikliklerinden ötürü çeşitli araştırmacılar yeni robot yapıları geliştirmeyi denediler. Örneğin Minsky 1972'de [4], Hunt 1978'de çeşitli yapılar önerdiler [5]. Ancak bu yapılar tamamen pratik zekaya dayanan belirli bir metodolojisi olmayan yapılardı. Belirli serbestlik dereceli paralel robotların olası yapılarını belirlemek için çeşitli metodlar ortaya konmuştur. Bu yapısal sentez yaklaşımlarının başlıcaları grafik teorisi, grup teorisi, vida teorisidir. Fakat bu alanda önemli sorulardan biri robotun serbestlik derecesinden farklı olarak kinematik performansı ile ilgili sentez yaklaşımının varlığıdır. Paralel robotların genel anlamda performansları boyutsal olarak ölçülendirilmelerine bağlıdır. Yapısal mimarisi uygun fakat ölçüleri iyi hesaplanmamış bir robot kötü performans gösterebilir. Sonuç olarak yapısal sentezi ölçüsel sentezden ayıramayız. Son kullanıcı için ayrıca cevabı bulunulması gereken bir soru da, ''uygulama için kullanılacak olan uygun yapı hangisidir? '' Sorusudur.



Şekil 1.5 Çeşitli SPM mimarileri 3x3 SPM, 6x3 SPM, 6x6 SPM [6]



Şekil 1.6 Çeşitli Kinematik Zincirler [1]

RRPS tipinde bir kinematik zincir, kardan kavraması (universal joint), prizmatik mafsal ve küresel mafsaldan oluşur. RPRS tipi döner mafsal, üzerine prizmatik mafsal, takibinde döner mafsal (revolute joint), ve küresel mafsaldan oluşur (ball-and-socket joint). PRRS tipi zincir ise prizmatik mafsal, kardan kavraması ve küresel mafsaldan oluşur. Son olarak R-RRS tipi döner mafsal, kardan kavraması ve takibinde küresel mafsaldan oluşmaktadır. Bu tip kinematik zincirler 6 serbestlik dereceli parallel robotların bacak tasarımında sıklıkla kullanılan yapılardır.

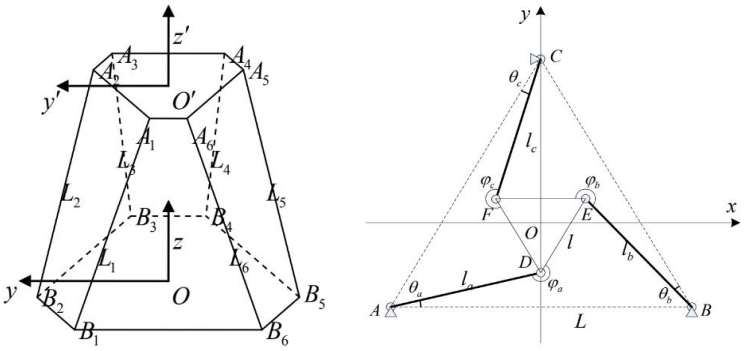
Paralel robotların kinematiğiyle ilgili yapılan çalışmalar incelendiğinde, çok çeşitli yapılardaki farklı mekanizmalar için yapılmış birçok çalışmayla karşılaşılmıştır. Bu çalışmaların genel amacı parallel robotların kinematiğini etkileyen parametrelerin optimize edilmesi üzerinedir. Bazı çalışmalarda optimizasyon tek bir amaç için yapılırken, bazılarında birden fazla, çok amaçlı optimizasyon gerçekleştirilmiştir. Dolayısıyla kullanılan optimizasyon yöntemleri ve optimize edilen parametreler ile birlikte optimizasyondan beklenen sonuçlar çeşitlilik göstermektedir.

1.1 Literatür Özeti

Paralel robotlar, mimarilerinden dolayı seri robotlara göre daha yüksek doğal frekansa, katılığa, çalışma hızına, nominal yüklemeye ve hassasiyete sahiptir. Fakat çalışma uzayları daha küçüktür. Paralel robotların kullanım alanları geniş olmasına rağmen, kinematik hesaplamalarının karmaşıklığı ve yapısal mimarilerinin çeşitliliği sebebiyle çalışma uzaylarını belirlemede kullanılabilecek açık bir formülasyon yoktur. Fakat bir çok yaklaşım vardır; tasarım aşamasında mutlaka çalışma uzayları hesaplanmalıdır [7].

Literatürde incelenen çalışmalardan ilkinde bilinen Stewart platformunun çalışma uzayı hacmi 2, 3 ve 6 boyutlu olarak sürü (swarm) optimizasyon yaklaşımından türetilmiş; firefly algoritması ile oluşturulmuştur. Yöntem iki aşamadan oluşmaktadır. Birinci aşamada, çalışma uzayının içinde olduğu saptanan birkaç koordinat noktası ile boşluk doldurma yaklaşımıyla çalışma uzayı kabaca haritalanmıştır. İkinci aşama da ise çalışma uzayının sınırları daha belirgin hale getirilmiştir [8]. Yöntem cebirsel, nümerik araştırmalara dayanan bir yöntemdir. Farklı bir çalışmada, Stewart platformu'nun çalışma uzayının belirlenmesinde geometrik bir yöntem önerilmiştir. Her bacağın kendisine ait çalışma uzayı tanımlanıp, çalışma uzayının hacmini bulmak için altı tane kürenin geometrik kesişimleri hesaplanmıştır. Sonuç olarak geometrik yöntemlerin cebirsel yöntemlerden hesaplama olarak daha az meşakkatli olduğu ortaya çıkarılmıştır [9]. Bu iki araştırmada da Stewart platformu'nun çalışma uzayı sabit oryantasyonda olduğu kabul edilerek hesaplanmıştır. Bununla birlikte çalışma uzayındaki tekil noktalar göz ardı edilmiştir. Tekil noktalar paralel robotların hareket kabiliyetlerini olumsuz yönde etkilediklerinden çalışma uzayı içinde istenmezler. Bu yüzdendir ki, bazı araştırmalarda tekil noktalar dikkate alınıp bu noktaları içermeyen çalışma uzayına sahip paralel robotlar tasarlanmaya çalışılmıştır.

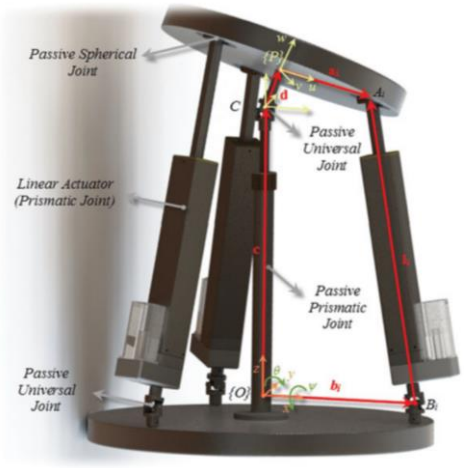
Bu araştırmalardan birinde, benzer-şekillilik (homeomorphism) teoremleri ve sonlu bozulma prensipleri kullanılarak tekil noktasız çalışma uzayı nasıl elde edilir açıklanmıştır. İki tip merkezi simetrik paralel robot (3-RPR ve 6-SPS yapısındaki) için, tekil noktasız çalışma uzayı amaçlanarak kısıt koşullar tanımlanmıştır [10].



Şekil 1.7 Paralel robot (6-SPS) ve düzlemsel paralel robot (3-RPR) [10]

Benzer bir çalışmada, Stewart platformunun yapısından daha farklı bir yapıda olan bir paralel robotun, (bir öteleme iki dönel serbestlik dereceli üç bacaklı paralel robot)

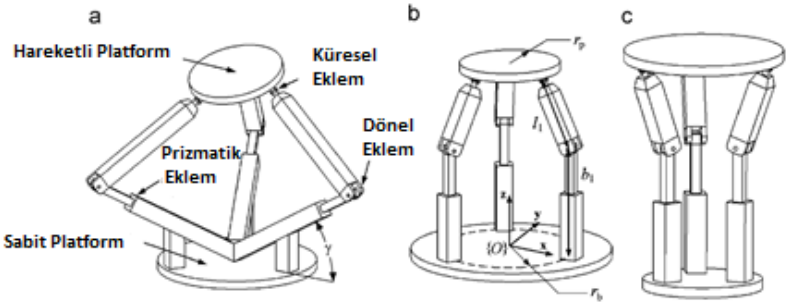
maksimum kartezyen çalışma uzayı araştırılmıştır. Kartezyen çalışma uzayı, küboid olacak şekilde kabul edilip, robotun hareket kabiliyeti de dikkate alınarak, çalışma uzayının maksimum x, y, z ölçüleri belirlenmiştir. Maksimum tekil değerler ve yerel koşul indeksleri dikkate alınarak tekilliklerden kaçınıp en hızlı şekilde pozisyonlama ile maksimum kartezyen çalışma uzayına sahip olabilmek için bir manipülatör tasarlanmıştır [11], [12].



Şekil 1.8 Üç bacaklı paralel robot yapısı [12]

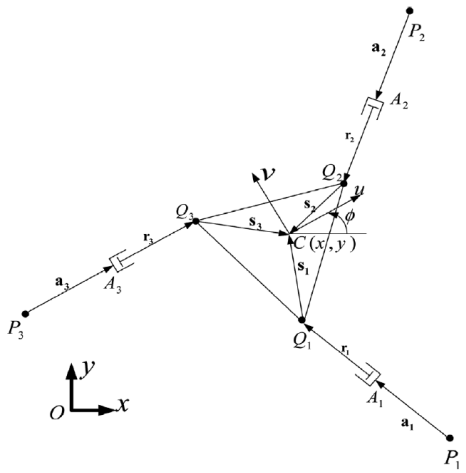
Çalışma uzayının etkinliği manipülatörün (Tricept paralel) çalışma uzayının her yerindeki becerisine göre nitelendirilmiştir. Performans indeksi, jakobiyen matrisinin tekil değerlerine bağlıdır. Jakobiyen elemanlarını uzunluk birimine bölerek, boyutsal olarak homojen yeni bir jakobiyen matrisi elde edilmiştir. Optimal tasarım problemi, kısıtlı doğrusal olmayan bu problem, genetik algoritma yöntemi ile çözülmüştür [12]. Tekillik jakobiyen matrisiyle direk olarak ilişkili olduğu için jakobiyen matrisinin doğru analiz edilmesi önemlidir. Bazı çalışmalarda jakobiyen matrisinden elde edilen performans indeksinin daha doğru yorumlanabilmesi için farklı yöntemler ile jakobiyen matrisi homojenleştirilmeye çalışılmıştır. Bunun sebebi matrisi oluşturan elemanların ifade ettikleri değerlerin farklı (öteleme veya açısal) nitelikte olmasıdır. Benzer bir araştırmada 3-PRS paralel manipülatörün üç farklı varyasyonu için, yapısal parametrelerin tekillikten uzak (dexterous) çalışma uzayı hacminin genişliğine etkisi her bir model için araştırılmış; yapısal parametreler optimize edilerek çalışma uzayı maksimum boyutlara getirilmeye çalışılmıştır.

Tekil değerler ve koşul sayısıyla çalışılarak boyut olarak homojen olan bir kare jakobiyen matrisi geliştirilmiştir [13]. Bir önceki araştırma ile arasındaki en belirgin fark jakobiyen matrisinin homojenleştirme yöntemidir.



Şekil 1.9 Paralel manipülatör modelleri (3-PRS) [13]

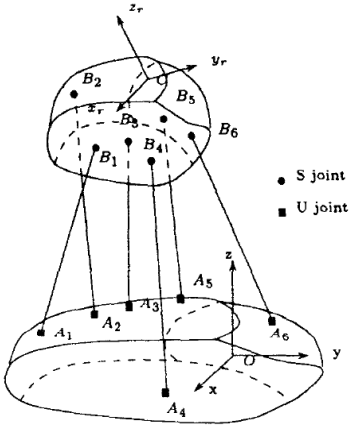
İhtiyaç duyulan çalışma uzayı hacminin farklı formlarda olması muhtemeldir. Daha önce tekillikten uzak küboid çalışma uzayı elde etmek için yapılan araştırmaya [12] benzer bir araştırma, tekillik içermeyen silindirik bir çalışma uzayı için yapılmıştır. Silindir eksenine en yakın tekil noktayı bulmak için, parçacık sürü optimizasyonu (PSO) yöntemi kullanılmıştır. Sonuç olarak geliştirilen algoritmanın 3-RPR düzlemsel paralel manipülatöre uygulanmasıyla verimlilik artarak daha geniş bir tekilliksiz çalışma uzayı elde edilmiştir [14].



Şekil 1.10 Düzlemsel paralel manipülatörün geometrik modeli (3-RPR) [14]

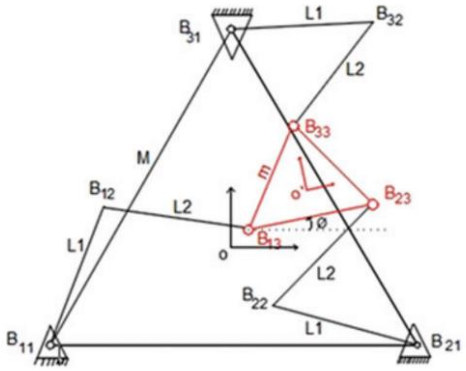
Paralel manipülatörlerin tasarımında öne çıkan konulardan biri optimize edilecek tasarım parametrelerinin hangi amaç için optimize edildiğidir. Bir başka araştırma da çalışma uzayı yine önceden sınırlandırılmış Stewart platformu için, en iyi doğrulukta verilen yüke göre minimum eklem kuvveti gerektirecek optimal bir paralel

manipulatorün tasarımı için çalışılmıştır. Arzu edilen çalışma uzayı indirgenmiş tasarım parametreleri kümesi ile tanımlanmış; parametrelerin optimal değerleri nümerik araştırmalar neticesinde belirlenmiştir [15].



Şekil 1.11 Paralel manipulator (6-DOF) [15]

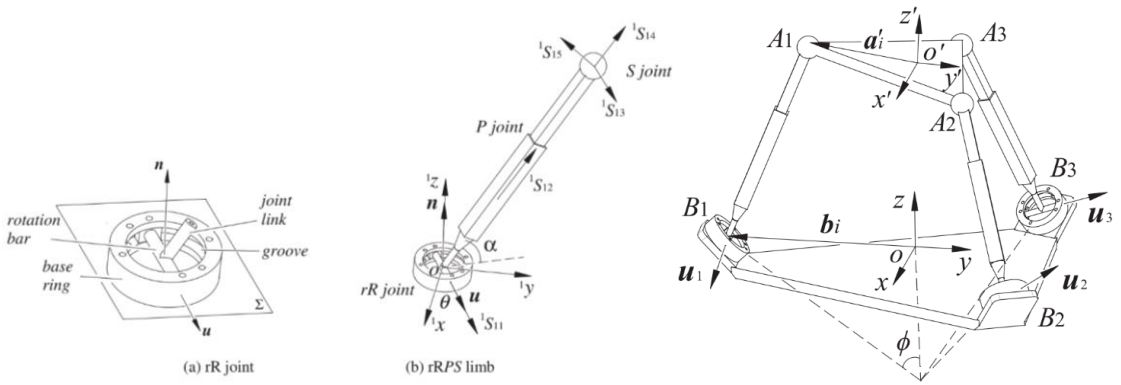
Paralel manipülatörlerin serbestlik dereceleri ne kadar düşük olursa optimizasyonları da o kadar kolaylaşır. Bununla birlikte serbestlik dereceleri artığında ise zorlaşır.



Şekil 1.12 Düzlemsel paralel manipülator 3-DOF (3-RRR) [16]

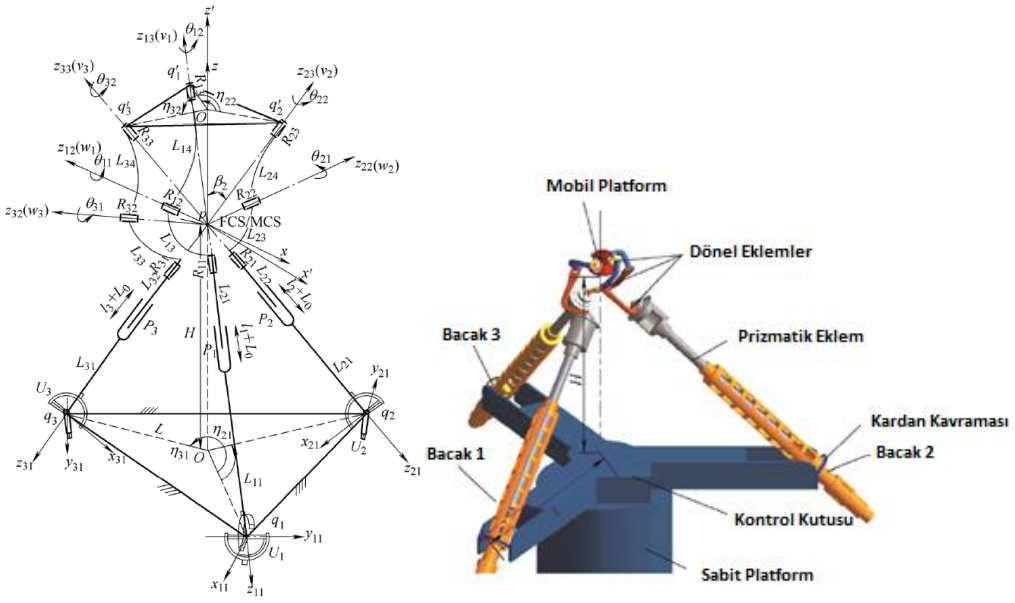
Diğer bir çalışmada üç serbestlik dereceli düzlemsel bir manipulatorün çalışma uzayı şekli geometrik parametrelerin bir fonksiyonu olarak tanımlanabilmiş ve kapalı döngüsel alan ifadeleri, üç döner eklemli (3-RRR) sabit oryantasyonlu bir çalışma alanı için türetilmiştir. İfadelerin, hareketli platformun farklı oryantasyonları için de uygulanabilir olduğu gösterilmiştir. Bununla beraber, belirlenen bir çalışma uzayı için düzlemsel (3-RRR) manipülatörün tasarımında geometrik yaklaşımla uygulanan bir optimizasyon prosedürü önerilmiştir [16].

Paralel robotların çalışma uzayını etkileyen en önemli etkenlerden biri bacakların hareketini sağlayan mafsal tipleridir. Genelde yaygın olarak kullanılan mafsal (eklem) tipleri küresel, prizmatik, dönel eklemlerdir. Bazı çalışmalarda özel tasarım eklem tiplerinin çalışma uzayı üzerine etkileri araştırılmıştır. Bunlardan birine örnek olarak konfigüre edilebilir dönel eklem ile oluşturulmuş bir paralel mekanizma yapısını verebiliriz. Bu çalışmada maksimum tekil noktasız çalışma uzayı ve kinematik performansa dayalı kriterler esas alınarak, 3rRPS metamorfik paralel mekanizmanın temel parametreleri eklem limitlerine göre optimize edilmiştir. Değişken tasarım öncelikleri ağırlıklandırılarak hedef fonksiyonu birleştirilip iki topolojili olarak sunulmuştur [17].



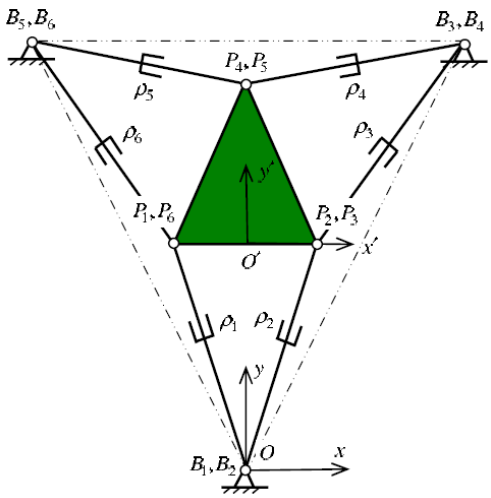
Şekil 1.13 rR Eklem, rRPS bacak ve 3rRPS metamorfik paralel mekanizma [17]

Paralel robotlarda genellikle bir çıkış değişkeninin değeri bütün giriş değişkenlerinin değerleriyle ilişkilidir. Eğer bir çıkış değişkeni bütün giriş değişkenleri yerine bir kaç giriş değişkeni ile ilişkiliyse bu tip manipülatörler ayrıştırılmış manipülatörler olarak isimlendirilir. Literatürde bir çok aynı işlevi gerçekleştirecek fakat farklı yapıdaki paralel robotların verimlerinin arttırılıp maliyetlerinin düşürülmesine yönelik araştırmalar mevcuttur. Bu amaçla yapılan çalışmalardan biri 3 bacaklı 6 serbestlik dereceli bir paralel manipulatorün (TLPM) belirli bir çalışma uzayı için optimal tasarımı üzerinedir. Öteleme ve dönel hareketlerin ayrıştırılması yapısal olarak farklılığını ortaya koymuştur. Optimal tasarımda boyutsal parametrelerin minimum olması üzerine çalışılmış; minimum değerlerin belirlenebilmesi için bir algoritma geliştirilmiştir. Bu metodla uygun kinematik parametreler belirlenip verimlilik arttırılmıştır. Büyük çalışma uzayı için küçük mekanik boyutlarda hafif bir robot tasarlanmıştır [18].



Şekil 1.14 TLPM'in kinematik diyagramı ve 3-D mimarisi [18]

Başka bir çalışmada Gough–Stewart platformunun minimal basitleştirilmiş simetrik halinin optimal mimarisi belirlenmeye çalışılmış; geometrik parametrelerinin, tekillikten muaf çalışma uzayı üzerine etkileri, incelenmiştir. Tekillikten muaf çalışma uzayını belirlemek için analitik bir algoritma geliştirilmiştir.



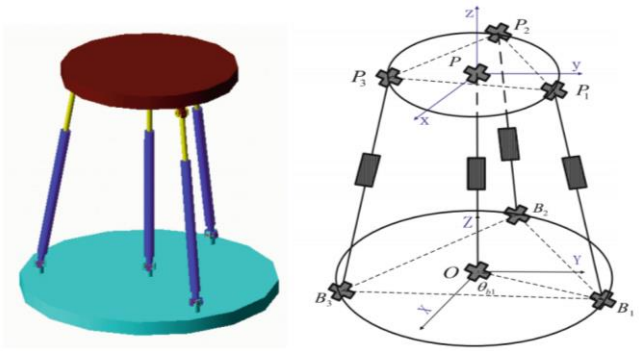
Şekil 1.15 MSSM mimarisi (üst görünüş) [19]

Yapılan analizlerin neticesinde: (1) Benzer ikiz kenar üçgen taban ve platform için ve eşkenar üçgen taban ve platform için optimal yapı bir tane olduğu belirtilmiştir. Platform ve taban için ölçü oranının 1/2 olması gerektiği; ve (2) Eğer taban ve platform

benzer üçgenler değilse, global optimal yapıyı belirlemenin zor olduğu belirtilmiştir. Sadece yaklaşık optimal mimari elde edilebilir sonucuna varılmıştır [19].

Tekillik içermeyen maksimum çalışma uzayı hacmine sahip olması arzu edilen paralel robotlar üzerine yapılmış bir çok çalışma incelenmiştir. Literatürdeki her bir çalışma kendine özgü bir özelliği barındırmakta, farklı bir konsepte dayanmaktadır. Farklı olarak incelenen diğer bir araştırmada Stewart platformunun beceriklilik indeksi ve çalışma uzayının şeklinin düzgün geometrilerde olması (dikdörtgen veya küre) dikkate alınmıştır. Optimal tasarım problemi, maksimum etkili düzgün çalışma uzayını sağlayacak manipülatör boyutlarını bulmak için, formülize edilmiştir. Problemi çözmek için CRS (controlled random search) tekniği kullanılmıştır [20]. CRS tekniğinin güvenilir olduğu sonucuna ulaşılmıştır.

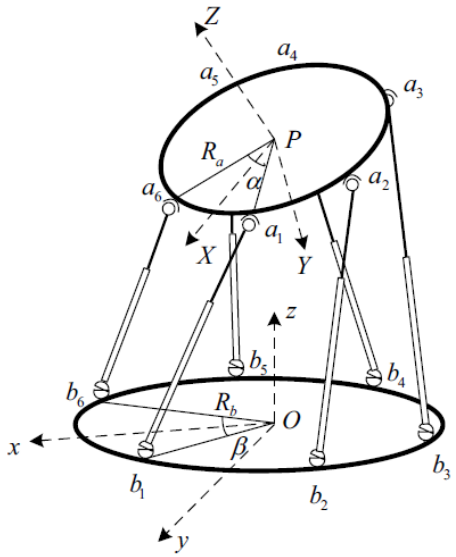
Çok amaçlı optimizasyonlarda birden fazla hedef amaçlanmıştır. İlgili paralel robotun genel olarak birden fazla özelliği için optimal çözüm elde edilmeye çalışılmıştır. Bu çalışmalara örnek olarak; 3UPU-UPU paralel mekanizma global rijitliği ve iyi koşullandırılmış çalışma uzayı dikkate alınarak optimize edilmiştir. Mekanizmanın global koşul indeksi Monte Carlo metodu kullanarak türetilmiştir. Global rijitliği hesaplanırken kriter olarak rijitlik matrisinin diagonal elemanlarının toplamı kullanılmıştır [21].



Şekil 1.16 3UPU-UPU mekanizması [21]

Paralel robotların hareket kabiliyetlerinin ihtiyaç duyulan çalışma uzayı içerisinde her noktada birbirine benzer olması arzu edilir. Bu özellik bir çalışmada GGI (global gradyen indeks) değeri ile ifade edilmiştir. Stewart platform için hedef fonksiyonlar GCI, GPI, GGI (global condition indeks, global payload indeks, global gradyen indeks) ile ilişkili

olarak belirlenerek ihtiyaç duyulan çalışma uzayı için hesaplanmıştır. Hedef fonksiyonlar gerekli çalışma uzayındaki becerikliliği arttırmak için eş zamanlı olarak optimize edilmiştir [22].



Şekil 1.17 Şematik gösterim (6-UPS paralel manipülatör) [23]

Çok amaçlı optimizasyonların tek amaçlı optimizasyonlardan farkı, optimizasyonun çözümünün tek bir sonucunun olmayışıdır. Optimizasyon sonucu paralel robotun bir veya birkaç özelliği iyileştirilirken bir başka özelliği kötüleştirilebilmektedir.

Daha önceden de bahsedildiği üzere Jakobiyen matrisinin homojenleştirilmesi için farklı yöntemler ile literatürde karşılaşılmıştı. Bu çalışmalardan birinde 6 serbestlik dereceli (UPS) paralel manipülatörün yapısal parametrelerinin boyutlandırılması metodolojisi üzerine çalışılmıştır. Jakobiyen matrisi, karakteristik uzunluk adı verilen bir değer hesaplanarak matrisin bütün elemanları bu değere bölünmek suretiyle matris homojenleştirilmeye çalışılmıştır. Bazı performans kriterleri (çalışma uzayı, kinetostatik performanslar, rijitlik ve dinamik beceriklilik (dexterity) eş zamanlı dikkate alınarak, çok amaçlı optimizasyon probleminin denklemi elde edilmiştir. Bu problem SPEA-II genetik algoritma metodu kullanılarak çözülmeye çalışılmıştır [23].

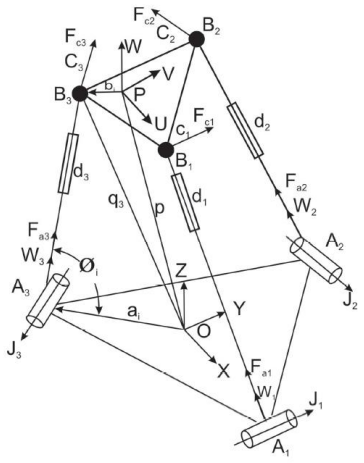
Paralel robotlarda performans kriterlerinden biri de rijitliktir. Robotun rijitliğinin iyi olması yüksek hızlarda çalışmasına imkan verir. Çok amaçlı optimizasyon yapılan çalışmalarda rijitlik indeksi ile sıklıkla karşılaşılmıştır. Bu çalışmaların ilkin de 6-DOF bir paralel manipülatörün performans kriterlerinden olan çalışma uzayı, koşul sayısı ve

rijitlik eş zamanlı olarak optimize edilmeye çalışılmıştır. Rijitlik indeksinin tekil noktalarda değişken olmasından ötürü paralel manipülatörün rijitliğini ölçmek için yeni bir performans indeksi (rijitliğin geometrik ortalaması) önerilmiştir. Optimizasyon için PSO (particle swarm optimization) yöntemi kullanılmıştır. Yönteme yeni bir alt döngü eklenerek yakınsamasının iyileştirilmesi hedeflenmiştir [24].

Literatürde incelenen paralel robotlar üzerine yapılmış çok amaçlı optimizasyon çalışmalarının bazılarında kabul gören diğer bir performans kriteri de aktüvatörlerde oluşan kuvvetlerdir. Taşınacak yükün maksimize edilmesi ve aynı zamanda da çalışma uzayı hacminin azalmaması amaçlanmıştır. [25], [26].

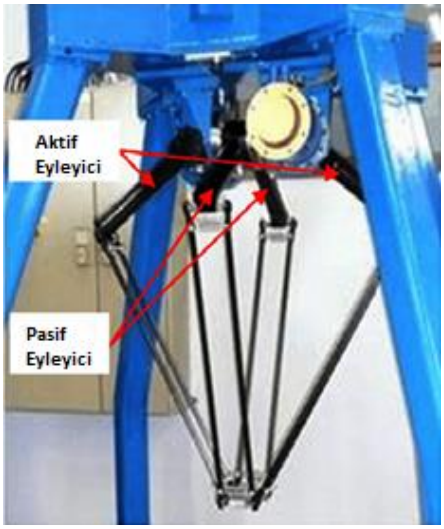
Paralel robotların optimal tasarımı genel olarak üç şekilde gerçekleştirilir. Bunlardan ilki analize dayalı optimizasyon metodlarıdır. Performans indeksleri analiz edilip sonuca göre tasarım parametreleri değiştirilir. İkincisi algoritma tabanlı optimizasyon yöntemleridir. Bir veya birden fazla performans indeksleri belirlenerek hedef fonksiyon kompleks nonlineer algoritmalar ile çözülerek optimizasyon gerçekleştirilir. Son olarak ise performans haritaları çizilerek optimizasyon gerçekleştirilebilir. Literatürde bu yöntem ile Stewart platformunun GCI (global condition index) haritası çıkarılarak belirli bir küresel çalışma uzayı için optimizasyonu gerçekleştirilmiştir. Performans haritaları metodunun diğer metodlara göre optimizasyonda dikkate alınan performans hedeflerindeki değişikliklere karşı daha kolay adapte olabileceği belirtilmiştir [27].

Algoritma tabanlı optimizasyon çalışmalarından bir diğerinde, 3-RPS tipi uzaysal paralel manipülatörün çok amaçlı optimal kinematik tasarımı gerçekleştirilmiştir. Hedef fonksiyon belirlenirken GCI (global conditioning index), GSI (global stiffness index) ve çalışma uzayı hacmi dikkate alınmıştır. Eş zamanlı olarak beceriklilik (dexterity) geliştirilmiş; çalışma uzayının hacmini artırmak için belirlenen hedef fonksiyon optimize edilmiştir. Kontrol seçiciliği çok amaçlı evrimsel algoritmaya dayanan, doğru optimal pareto listesini bulmak için, genetik algoritma uygulanmıştır. Manipülatörün çalışma uzayı nümerik araştırmalarla bulunmuştur [28].



Şekil 1.18 3-RPS uzaysal manipülatör [28]

Literatürde önceki çalışmalardan farklı olarak, paralel robotlarda boyutsal sentez için uygulanan iki yaklaşımı, "tek amaçlı optimizasyon (bir performans kriterli) ile çok amaçlı (birkaç performans kriterli) optimizasyonu", karşılaştırmışlardır. İki serbestlik dereceli paralel robot iki yaklaşım için örneklenmiştir. Optimizasyonda SQP ve Genetik algoritma NSGA-II yöntemleri kullanılmıştır. Sonuç olarak çok amaçlı optimizasyonun tek amaçlı optimizasyondan daha faydalı olduğu gösterilmiştir [29].



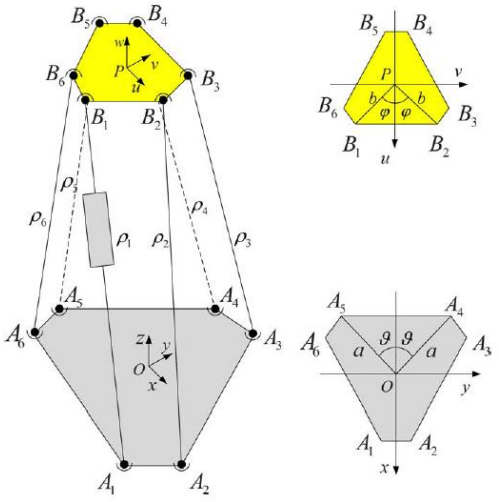
Şekil 1.19 Par 2 robot prototipi [29]

Paralel robot tasarımlarının birbirleriyle karşılaştırıldığı, birbirlerine karşı üstünlüklerinin ifade edildiği bir çok çalışmayla karşılaşılmıştır. Bu çalışmalardan birinde 3 DOF bir paralel manipülatör ile 6 DOF Stewart Platform'unun tasarımı için iki farklı optimizasyon yöntemi kullanılmıştır. İlk yöntemin hedef fonksiyonu, manipülatörün

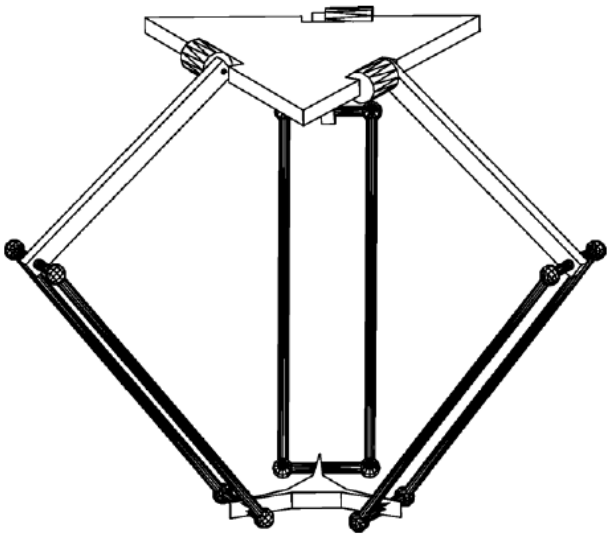
toplam çalışma uzayı hacmini maksimize eder. İkinci yöntem, GCI'i maksimize ederek iyi koşullandırılmış toplam çalışma uzayı hacmini optimize eder. Sonuçlar karşılaştırılarak ikinci tasarımın daha optimal bir tasarım olduğu belirtilmiştir. Optimizasyon metodu olarak Monte Carlo yöntemi kullanılmıştır [30].

Literatürde paralel robotların optimizasyonu için algoritma tabanlı yapılan çalışmalarda kullanılan algoritmaların birbirleriyle karşılaştırıldığı bir çok araştırma ile karşılaşılmıştır. Bu çalışmalardan birinde, 6-dof bir parallel manipülatün kinematik tasarımı analiz edilmiştir. İlk olarak ters kinematik jakobiyen matrisinin koşul sayısı hedef fonksiyon olarak düşünülmüş ve genetik algoritma metodu kullanılarak optimizasyon problemi çözülmüştür. İkinci bir yaklaşım olarak analitik hedef fonksiyonunun sinir ağları modeli geliştirilmiş ve hedef fonksiyon olarak tanımlanarak genetik algoritma ile optimizasyon gerçekleştirilmiştir. Sinir-genetik algoritmanın maksimum beceriklilik için optimal sonuca hesaplama yükünü azaltarak yaklaşabildiği gösterilmiştir [31].

Benzer şekilde optimizasyonda kullanılan algoritmaların, (SQP), (CRS), (GA), (DE), (PSO) karşılaştırıldığı bir çalışmada problemin çözümündeki performansları delta robot ve Stewart platformu üzerinde incelenmiştir. Sonuç olarak basit tasarım problemlerinde (SQP) metodunun çok başlangıç noktasıyla etkili olduğu, (DE) ve (PSO) metodlarının bütün tasarım problemleri için etkili olduğu, (CRS) metodunun ise başlangıçta iyi başlagıç noktaları bulmak için kullanılabileceği sonucuna varılmıştır [32].



Şekil 1.20 Gough-Stewart platformu'nun şematik gösterimi [32]



Şekil 1.21 Dönel Delta Robotun Yapısal Mimarisi [32]

Farklı amaçlar için kullanılan paralel robotların kinematik tasarım parametrelerinin optimize edildiği bir çok araştırma ile karşılaşılmıştır. Bu amaçlardan biri de aktif vibrasyon izolasyonudur. 6-dof Stewart platformunun tasarım parametreleri genetik algoritma eğitimli sinir ağları yöntemiyle, SIMULINK modeli geliştirilerek optimize edilmiştir. Tasarım parametrelerinin vibrasyon izolasyonu üzerine etkileri incelenmiştir. Eklem bağlantı noktalarının olabildiğince birbirine yakın olmasının faydalı olabileceği sonucuna varılmıştır [33].

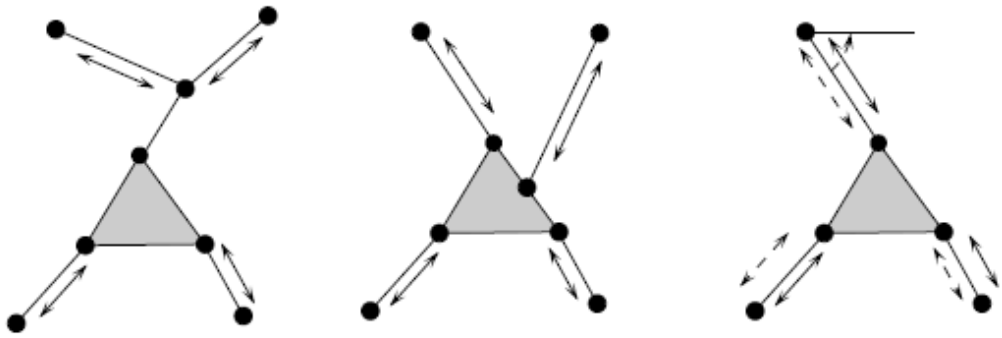
Paralel robotların çalışma uzayı hacimlerinin önemi olduğu kadar bu hacmin içinde gösterdikleri dinamik performansları da çok önemlidir. Dolayısı ile tasarım aşamasında servo-motor seçimi bu açıdan büyük önem arz etmektedir. Bu konu üzerine yapılan bir araştırmada Stewart platformunun dinamik performansını iyileştirmek için ataleti ayrıştırarak, boyutsal parametrelerin atalet indeksi üzerine etkileri tartışılıp boyutsal parametreler optimize edilmiştir. Atalet indeksi atalet matrisinin en büyük öz değerinin oransal ifadesi olarak tanımlanmıştır. Atalet indeksinin platform çaplarından daha çok eklem noktaları arasındaki açıdan etkilendiği belirtilmiş; olabildiğince bu açıların yük ataletini azaltması için minimum yapılması önerilmiştir [34].

Paralel robotların performansları çeşitli performans indeksleri kullanılarak değerlendirilmektedir. Bu amaçla bir çalışmada, paralel manipülatörün lokal beceriklilik

için analitik denklemleri belirlenmiştir. İzotropik konfigürasyonunu elde etmek için parametrik ilişkiler türetilmiştir. Bağıl beceriklilik konsepti üzerinde durulmuştur [35]. Benzer şekilde, 6-DOF bir parallel manipulatorün kinematik analizi yapılarak rijitlik ve beceriklilik için hedef foksiyonlar türetilmiştir. Bu fonksiyonları genetik algoritma ve yapay sinir ağları metodlarını kullanılarak optimize edilmiştir [36].

Boyutsal olarak homojen olmayan jakobiyen matrisine sahip paralel manipülatörlerin kinematik hassasiyetlerini belirlemek için yeni indeksler önerilmiştir. Maksimum dönüş hassasiyetini ve maksimum noktasal yer değiştirme hassasiyetini belirlemek için bir yaklaşım geliştirilmiştir [37]. Benzer şekilde eklem hızlarının optimal seviyelerde olmasının paralel robotların kontrol edilebilirliliği açısından olumlu olacağı düşünülürek yeni bir performans indeksi önerilmiştir. Bu indeks eklem hızlarının şiddetini yansıtan "hız dönüşüm faktörü" olarak tanımlanmıştır. Daha sonra global hız dönüşüm faktörünü minimize edecek manipulator geometrisini bulmak için optimal tasarım problemi formülize edilerek genetik algoritma metoduyla problem çözülmüştür. Stewart-Gough platformunun tasarımı optimal eklem hızlarına dayanarak gerçekleştirilmiştir [38].

Paralel robotların seri robotlara göre kısıtlı olan çalışma uzaylarını ve hareket becerilerini (dexterity) geliştirmek için farklı yapıda mekanizmalar geliştirilerek bir arayışa girilmiş; bu sebeple (redundant) fazla eklemli yapıda paralel robotlar tasarlanmıştır.



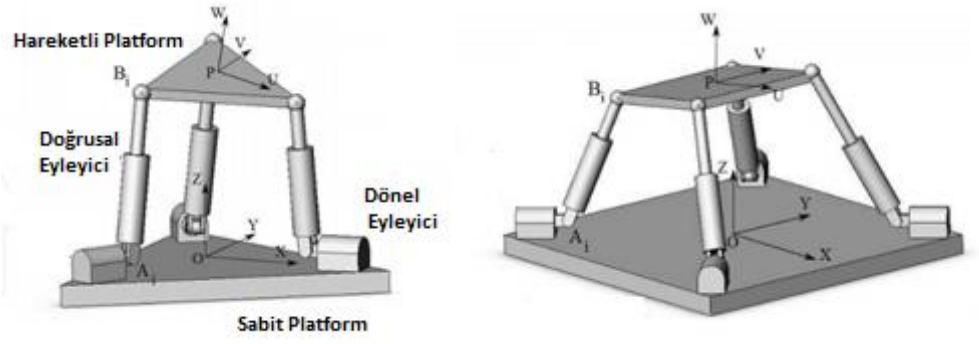
Şekil 1.22 Soldan sağa, kinematik fazlalığı, eyleyici fazlalığı ve ölçüm fazlalığı [1]

1. Kinematik fazlalık: Arzu edilen serbestlik derecesini sağlayacak yeterli ayak sayısından en az bi tane daha fazla ayağa sahip olunması durumudur. Çalışma uzayını genişletmek için yapılır.

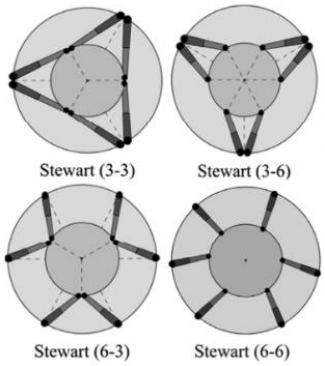
2. Eyleyici fazlalığı: Uç efektöre fazladan ayak bağlama durumudur. Tekilliklerden (Singularity) kaçınmak için yapılır.

3. Ölçüm fazlalığı: Sensör sayısının hareket eden eklem sayısından fazla olması durumudur. Bu fazlalık ileri kinematik problemi çözme de ve robotun kalibrasyonunda, pozisyonlama hatalarını azaltmada yardımcı olur.

İncelenen fazla eklem ile ilgili yapılan çalışmaların ilkinde yedekli 4 bacaklı ve yedekli olmayan 3 bacaklı paralel mekanizmalar Gough-Stewart platformlarının bilinen tanınmış 4 mimarisi ile karşılaştırılmıştır. Önerilen 3 bacaklı ve 4 bacaklı mekanizmaların ters kinematiklerinin kapalı formlu bir çözüme sahip oldukları gösterilmiştir. Jakobiyen matrisleri belirlenmiştir. Tasarım açısından bakıldığında, Stewart platformlarındaki pasif kardan mafsallarının aktif mafsallar ile değiştirilmesiyle, bacakların 6'dan 3'e veya 4'e düşürülebileceği saptanmıştır. Böylelikle mekanizmanın daha hafif olacağı belirtilmiştir. Yedekliğin, yedekli olmayan paralel manipülatörün yeteneğini ve performansını geliştirdiklerini gösterilmiştir. Fazla eklemin, kinematik tekilliklerden kaçınma, çalışma alanını artırma, beceri, yönetilebilme ve hassasiyet gibi performans indekslerini geliştirme gibi paralel manipülatörler için bazı avantajlar getirdikleri belirtilmiştir. Son olarak, fazla eklemin hemen hemen tüm paralel mekanizmalarda yaygın olan tekil noktaları kaldırmak için kilit bir seçim olduğuna karar verilmiştir [39].



Şekil 1.23 Soldan sağa fazlalıksız mekanizma, fazlalıklı mekanizma [39]

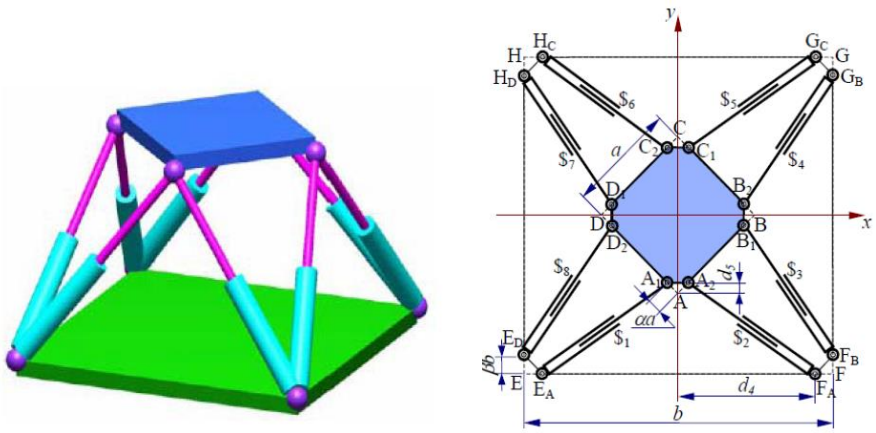


Şekil 1.24 Stewart platform şematikleri [39]

Paralel robotlardaki fazlalık konsepti bazı çalışmalarda iki başlık altında sınıflandırılıp incelenmiştir. Bu başlıklar aktüatörlerin sayısının fazlalığı ve manipülatörün serbestlik derecesinin ihtiyaç olandan fazlalığıdır. Sonuç olarak fazlalığın başlıca avantajları çalışma uzayını genişletmesi, tekilliklerden kaçınılması ve eklem torklarının dağılımının iyileşmesinin olduğu belirtilmiştir. Fazlalığın olmasının avantajları ve dez avantajları kontrol, kalibrasyon gibi farklı açılardan incelenmiştir [40].

Son olarak literatürde incelenmiş olan çalışma, performans ölçütü olarak farklı bir kriterin dikkate alındığı çalışmalardan biri olan 8-8 bir paralel manipulator için gerçekleştirilmiştir. Kalite indeksi manipülatörün merkezi simetrik konfigürasyona olan yakınlığını ifade etmektedir. Değeri 0 ile 1 arasında değişmektedir. Bir'e ne kadar yakınsa bu değer manipulator tasarımı açısından daha iyi olacağı belirtilmiştir. Tasarımcılara yardımcı olması amacıyla manipülatörün hareketli platformu ile sabit platformunun boyutları birbirlerine göre kalite indeksi dikkate alınarak ölçeklendirilmiştir. Aynı zamanda platformun çalışma uzayının merkez konumunun ideal olması için sabit ve hareketli platformun eklem merkezlerinin yerleri belirlenmiştir [41].

Literatürde incelenmiş olan çalışmalarda farklı bir çok sayıda performans indeksi tanımlanıp, değerlendirilerek farklı optimizasyon metodları ile farklı yapılardaki paralel robotların optimize edilmeye çalışıldığı görülmüştür.



Şekil 1.25 Fazlalıklı 8-8 paralel manipülatör [41]

1.2 Tezin Amacı

Farklı yapıdaki paralel robot konfigürasyonları için literatürde bir çok çalışma mevcuttur. Bu parallel robot konfigürasyonlarından biri olan 6-DOF (6 serbestlik dereceli) 3x3 UPU yapıdaki Stewart platformunun kinematik parametrelerinin çok amaçlı olarak optimize edilmesi amaçlanmıştır. Kinematik parametrelerin değerleri tekilliklerden kaçınırken maksimum çalışma uzayına erişebilmek noktasında çok önemlidir. Bunu gerçekleştirebilmek için optimizasyon metodu olarak Sequential Quadratic Programming (SQP) metodu seçilmiştir. Hareketli platformun yarıçapı ile sabit platformun yarıçapı ve bacak uzunlukları optimize edilmek istenen kinematik parametrelerdir. Optimizasyon kademeli olarak parameter sayısı arttırılmak suretiyle farklı hedef fonksiyonları için gerçekleştirilmiştir.

1.3 Hipotez

Uniform yapıda tekillikleri minimize edilmiş hedef çalışma uzayı amaçlanarak 6-DOF (6 serbestlik dereceli) 3x3 UPU yapısındaki Stewart platformunun geometrisi SQP Sequential Quadratic Programming optimizasyon metodu ile belirlenmiştir. Belirlenen kinematik parametrelerle optimal yapı elde edilip minimum tekillik koşuluyla elde edilen uniform çalışma uzayındaki bütün koordinatlara erişim sağlanmıştır.

BÖLÜM 2

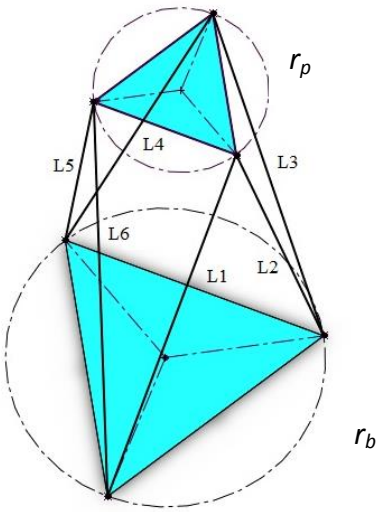
PARALEL ROBOT SİSTEMİ

Literatürde birçok (PRS) paralel robot sistemi mevcut olup farklı amaçlar için günümüzde kullanılmaktadır. Bu bölümde tez çalışmasında kullanılan paralel robot sisteminden bahsedilecektir.

2.1 Paralel Robot Mekanizması

Bir parallel robot sistemi (PRS) hareketli ve sabit platform arasında parallel bağlı bacaklardan oluşur. PRS için bazı sınıflandırma metotları mevcuttur a) Hareketli ve sabit platformdaki bacak bağlantılarının eklem tipine göre b) Bacakların sayısına göre. Özellikle doğrusal olarak hareket eden hareketli ve sabit platforma bağlı 6 bacak bütünüyle uzaysal hareketi yani 3 eksen öteleme ve 3 eksen dönme hareketini sağlar. Bu paralel robotik sistem (PRS) Stewart Platformu olarak isimlendirilir, [22] [28] [42]. Üst ve alt platformlardaki bağlantı sayısına göre sınıflandırılır. 3x3 SP mekanizmasında alt ve üst platform kenarlarında 3 bacak bağlantısı vardır. 6 bacaktan her ikisi birlikte merkezlenmiş mafsalı paylaşır. Bu tezde Şekil 2.1'de gösterilen 3x3 PRS yapısı kullanılmıştır.

Bir SP, çalışma uzayını ve kinematik davranışları etkileyen 48 tasarım parametresine sahiptir [15]. Bu parametreler alt ve üst platformdaki eklemlerin (mafsal) uzaysal koordinatları (2x18) ve bacakların maksimum ve minimum uzunlukları (2x6) dır. Eklemlerin merkezlerini bir daire üzerinde gerçekleştirirsek, üst ve alt platformun eklem merkezlerinden geçen dairelerin yarı çapları r_p *ve* r_b olacak şekilde, bu parametre sayısını düşürmemizi sağlar. Bu tezde optimizasyon parametrelerimiz bu iki yarıçap ve bacakların uzunluğu (l_i) dur.



Şekil 2.1 Tezde kullanılan 3x3 Stewart Platform'un (SP) yapısı

2.2 Paralel Robot Sisteminin Geometrik Parametreleri

Bacakların uzunlukları (l_i), hareketli platformun oryantasyonuna ve arzu edilen pozisyona göre değişir. Hareketli platformun oryantasyonuna ve hedeflenen pozisyona, **x** (*x, y, z, α, β, γ*), göre ters kinematik denklemler kullanılarak bacak uzunlukları hesaplanabilmektedir.

Bu durumda altı tane bilinmeyen vardır; l_i (i=1….6). Şekil 2.2 SP'nin geometrik yapısını anlamak için bize yardım eder. İki koordinat takımı, {P} ve {B}, tanımlanmıştır. Her biri hareketli ve sabit platformla ilişkilendirilmiştir. {P} koordinat takımının merkezi hareketli platform ve bacaklar arasındaki eklemlerin (mafsal) oluşturduğu çemberin merkezidir. Z_p ekseni yukarı yönlü düşey eksendir; ve X_p ekseni, (O_PP_6-O_P P_1) açısının açı ortayıdır. {B} koordinat takımının merkezi sabit platform ve bacaklar arasındaki eklemlerin (mafsal) oluşturduğu çemberin merkezidir. Z_b ekseni yukarı yönlü düşey eksendir; ve X_b ekseni (O_bB_6-O_bB_1) açısının açı ortayıdır.

Bu iki koordinat takımı sabit (B_1.., B_6) ve hareketli platformdaki (P_1.., P_6) bağlantı noktalarının pozisyonlarını belirlemek için bize yardımcı olur. (O_pP_2, O_pP_3) açısı θ_p ile ifade edilmiştir. (O_pP_1, O_pP_3), (O_pP_3, O_pP_5) ve (O_pP_5, O_pP_1) açıları herbiri birbirine eşit ve 120° dir. Benzer şekilde, (O_bB_1, O_bB_2) açısı θ_b ile ifade edilmiştir. (O_bB_1, O_bB_3), (O_bB_3, O_bB_5) ve (O_bB_5, O_bB_1) açıları da birbirine eşit ve 120° dir.

Bacak uzunlukları (l_i) aşağıdaki denklemler ile hesaplanabilir;

i =1,2,…,6. (2.1)

(X_p, O_pP_i) açısı λ_i ile ; ve (X_b, O_bB_i) açısı Λ_i ile ifade edilmiştir. Dolayısıyla

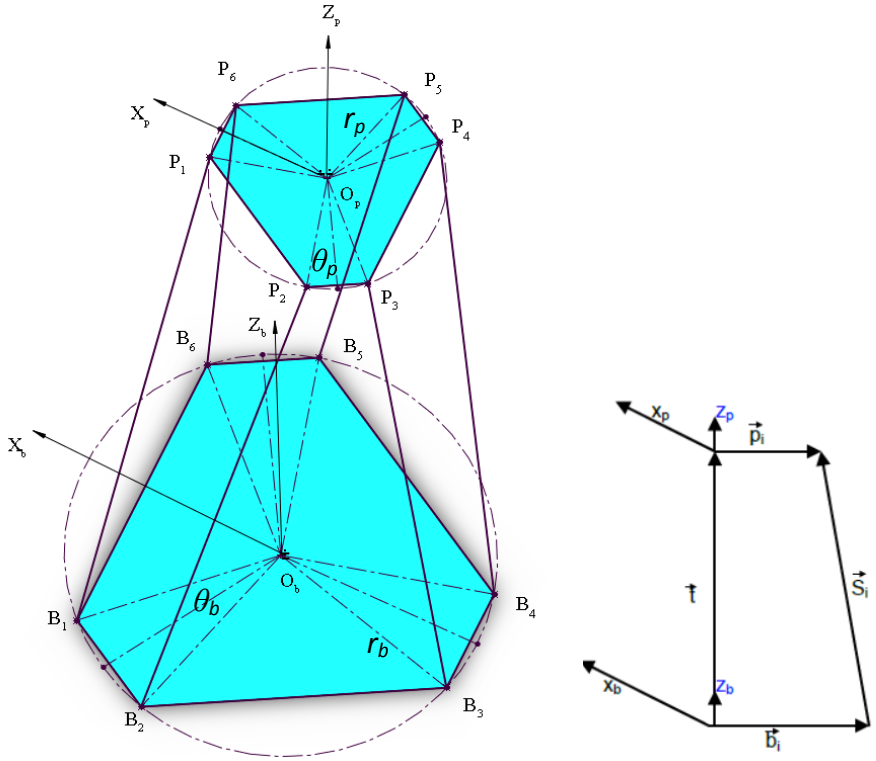
$$\Lambda_i = 60i - \theta_b/2 \quad , \lambda_i = 60i - \theta_p/2 \, , \quad i=1,3,5 \tag{2.2}$$

$$\Lambda_i = \Lambda_{i-1} + \theta_b \quad , \lambda_i = \lambda_{i-1} + \theta_p \qquad i=2,4,6 \tag{2.3}$$

$$^B\boldsymbol{S}_i = -\,^B\boldsymbol{b}_i + {}^B\boldsymbol{t} + {}^B\boldsymbol{p}_i \tag{2.4}$$

$$^B\boldsymbol{t} = \{x,y,z\}^T \tag{2.5}$$

(5) no'lu eşitlik {P} koordinat sisteminin {B}' ye göre öteleme pozisyonunu verir.



Şekil 2.2 SPM 'nın koordinat sistemleri, {P} ve {B}

$$^B\vec{\boldsymbol{p}}_i = {}^B_P\mathbf{R}\,{}^P\vec{\boldsymbol{p}}_i \, . \tag{2.6}$$

(2.6) eşitliği {P} Hareketli platformun merkezinin {B} sabit platformun koordinat sistemine göre dönmesini verir.

$$^{P}\boldsymbol{p}_i = \{p_{ix}, p_{iy}, p_{iz}\}^T = \{r_p.\cos(\lambda_i), r_p.\sin(\lambda_i), 0\}^T \quad (2.7)$$

(2.7) eşitliği hareketli platformun eklem pozisyonlarını P_i , {P} koordinat sistemine göre tanımlar.

$$^{B}\boldsymbol{b}_i = \{b_{ix}, b_{iy}, b_{iz}\}^T = \{r_b.\cos(\Lambda_i), r_b.\sin(\Lambda_i), 0\}^T \quad (2.8)$$

(2.8) eşitliği sabit platformun eklem pozisyonlarını B_i , {B} koordinat sistemine göre tanımlar.

$C_\alpha = \cos(\alpha)$, $S_\alpha = \sin(\alpha)$, $C_\beta = \cos(\beta)$, $S_\beta = \sin(\beta)$, $C_\gamma = \cos(\gamma)$, $S_\gamma = \sin(\gamma)$

$$^{B}_{P}\boldsymbol{R} = \begin{bmatrix} C_\alpha C_\beta & C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma & S_\gamma S_\alpha + C_\alpha S_\beta C_\gamma \\ S_\alpha C_\beta & C_\alpha C_\gamma + S_\alpha S_\beta S_\gamma & C_\gamma S_\beta S_\alpha - S_\gamma C_\alpha \\ -S_\beta & C_\beta S_\gamma & C_\beta C_\gamma \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.9)$$

α açısı {P} koordinat sisteminin {B} koordinat sistemine göre Z_b etrafındaki dönme açısıdır; β açısı {P} koordinat sisteminin {B} koordinat sistemine göre Y_b etrafındaki dönme açısıdır; γ açısı ise {P} koordinat sisteminin {B} koordinat sistemine göre X_b etrafındaki dönme açısıdır.

Sonuç olarak,

$$^{B}\boldsymbol{p}_i = \begin{bmatrix} C_\alpha C_\beta . r_p C(\lambda_i) + \left(C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma\right). r_p . S(\lambda_i) \\ S_\alpha C_\beta . r_p C(\lambda_i) + \left(S_\alpha S_\beta S_\gamma + C_\alpha C_\gamma\right). r_p . S(\lambda_i) \\ -S_\beta . r_p C(\lambda_i) + C_\beta S_\gamma . r_p . S(\lambda_i) \end{bmatrix} \quad (2.10)$$

Bacak uzunlukları aşağıdaki gibi hesaplanabilir:

$$l_i = \sqrt{(S_{ix})^2 + (S_{iy})^2 + (S_{iz})^2} \quad (2.11)$$

Eşitlik (2.4) kullanılarak bacak uzunlukları aşağıdaki gibi elde edilir,

$$l_i^2 = x^2 + y^2 + z^2 + {r_p}^2 + {r_b}^2 + 2\left(r_{11}p_{ix} + r_{12}p_{iy}\right)(x - b_{ix}) \quad (2.12)$$
$$+2\left(r_{21}p_{ix} + r_{22}p_{iy}\right)\left(y - b_{iy}\right) + 2\left(r_{31}p_{ix} + r_{32}p_{iy}\right)(z) - 2(xb_{ix} + yb_{iy})$$

2.3 Jakobiyen Matrisi

Bir robotik sistemin Jakobiyen matrisi eklem uzayındaki hız, $\dot{l}_i$, ile dünya uzayındaki hız, $\dot{\boldsymbol{x}}$, arasındaki dönüşümü temsil eder [42].

Paralel bir manipulator için, bir vektör (l_i) olarak bacakların uzunluğu:

$$\mathbf{l} = (\mathrm{l}_1, \dots, \mathrm{l}_6)^T \tag{2.13}$$

Hareketli platformun konumu,

$$\mathbf{l} = \mathrm{g}(\mathbf{x}) \,, \mathbf{x} = (x, y, z, \alpha, \beta, \gamma)^T \tag{2.14}$$

Dolayısıyla, hareketli platformun pozisyonu (**x**) ile bacakların uzunluğu (l_i) arasındaki eşitlik şöyle ifade edilebilir:

$$\mathbf{l} = \mathrm{g}(\mathbf{x}) \tag{2.15}$$

Eğer eşitlik (2.15)'in türevi alınırsa:

$$\dot{\mathbf{I}} = \mathbf{J}\dot{\mathbf{x}} \tag{2.16}$$

$$\mathbf{J} \equiv \partial g / \partial x_i \tag{2.17}$$

$$\left[\dot{l}_1, \dots, \dot{l}_6\right]^T = \mathbf{J}\left[\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}\right]^T \tag{2.18}$$

$$\mathbf{J}_{ij} = \partial \mathbf{l}_i / \partial \mathbf{x}_j \tag{2.19}$$

Sonuç olarak, $\mathbf{J}$ mekanizmanın jakobiyen matrisi bu tezde yararlanılan PRS'nin

$$\mathbf{J} = [\, \mathbf{J}_{i1}, \mathbf{J}_{i2}, \mathbf{J}_{i3}, \mathbf{J}_{i4}, \mathbf{J}_{i5}, \mathbf{J}_{i6} \,] \,, i = (1,2,3,4,5,6) \tag{2.20}$$

performans indekslerinin hesaplanmasında kullanılmıştır.

2.4 Performans İndeksleri

2.4.1 Beceriklilik, Global ve Yerel Koşul İndeksi (Dexterity & GCI & LCI)

Beceriklilik paralel manipülatörün eklem hızlarının, hareketli platformun hızına dönüşme, aynı şekilde tam tersi şekilde hareketli platformun hızının eklem hızlarına dönüşme, kabiliyetinin bir göstergesidir. Paralel manipülatörün yapısına ve poziyonuna bağlı olarak değişir; ve Jakobiyen matrisinin koşul sayısı kullanılarak ölçülebilir.

Beceriklilik = Cond(J) . (2.21)

Cond (J) Jakobiyen matrisinin koşul sayısıdır; ve aşağıdaki gibi tanımlanır;

$$\text{Cond(J)} = \|J\| \, \|J^{-1}\| \tag{2.22}$$

$\| \, . \, \|$ ilgili vektör veya matrisin norm'unu ifade eder.

Jakobiyen matrisin koşul sayısını hesaplarken, literatürde kullanılan farklı tip normlar bulunmakta ve en çok kullanılanları Frobenius ve Spectral normlarıdır [1] . Bu normlar arasındaki fark onların hesaplanma metotlarıdır. Eğer Frobenius norm'u dikkate alacak olursak,

$$\|\mathrm{J}\| = \sqrt{\mathrm{tr}\,(\mathrm{J}^{\mathrm{T}}\mathrm{J})} \tag{2.23}$$

Bu durumda, frobenius normu, Jakobiyen matrisinin her bir elemanının ikinci dereceden toplamlarının kökleri olarak tanımlanır. GCI (Global Condition Index) bütün çalışma uzayı için hesaplanmış olan Cond(J)=LCI (Local Condition Index)'lerin ortalamasıdır.

Eğer Spektral norm dikkate alınacak olursa, beceriklilik indeksi şöyle tanımlanır,

$$\text{Cond(J)} = \frac{SV_{\max(J)}}{SV_{\min(J)}} \tag{2.24}$$

$\mathrm{SV}_{\max(\mathrm{J})}$ ve $\mathrm{SV}_{\min(\mathrm{J})}$ sırasıyla Jakobiyen matrisinin (J), maksimum ve minimum tekil değerlerini ifade eder. Cond (J) koşul sayısının değeri, direk olarak Jakobiyen matrisinin tekil değerlerine bağlıdır; ve bir ile pozitif sonsuz arasında bir değer alır. Koşul sayısı

Cond(J) 1'e eşit olduğu zaman, manipulator isotropiktir yani çalışma uzayının her noktasında beceriklilik konumdan bağımsız hali alır; ve Jakobiyen matrisindeki bütün tekil değerler eşit olur. Diğer taraftan koşul sayısı Cond(J) pozitif sonsuza eşit olduğu zaman, Jakobyen matrisi tekildir. Dolayısıyla koşul sayısı dikkate alındığında, *koşul sayısının değeri daha iyi beceriklilik (dexterity) için minimum olmalıdır*. Koşul sayısı robotik bir sistemin doğruluğu/becerikliliği ölçmek için bir indeks gibi kullanılır; ve bir noktanın tekilliğe uzak ya da yakın olduğu ile ilgili bilgi verir. Bununla birlikte, öteleme ve dönel serbestliği olan bir PRS de tekillik için tam olarak bir matematiksel yaklaşım belirlemek mümkün değildir. Bu sebeple koşul sayısının kullanımı sadece bir indeks olaraktır; tek bir sayı olarak avantajı PRS'in bütün kinematik davranışını açıklamak için kullanılır. Koşul sayısının kompleks tanımı sebebiyle, çok basit robotlar dışında biz koşul sayısının analitik formunu konum parametrelerinin bir fonksiyonu gibi hesaplayamayız. Bununla birlikte, güvenilir lineer cebir araçları koşul sayısını verilen bir uzaysal konum için nümerik olarak hesaplayabilir. Öteleme ve dönel serbestliğe sahip paralel robotların koşul sayısıyla ilgili önemli bir sorunu vardır. Öteleme hareketinin birimi dönel hareketinkinden farklı olduğu için, Jakobiyen matrisi homojen değildir [28]. Bu sebeple, dönel ve öteleme hareketi yapan PRS'lerin Jakobiyen matrislerinin normalizasyonu için araştırmalar vardır. Bazıları normalizasyon için kendi metotlarını belirlemişler, bazıları ise Jakobyen matrisinin her elemanının değerini matrisin derecesine bölmüşlerdir [13] [18] [37] [43].

2.4.2 Homojenlik (Uniformity)

Homojenlik PRS'ler için bir diğer global performans indeksidir; ve çalışma uzayında tekillik dağılımının derecesini ifade etmektedir. Homojenlik indeksinin hesaplanması eşitlik (2.25)'te verilmiştir. Homojenlik indeksinin değeri bir ile sonsuz arasında değişir. En homojen çalışma uzayının homojenlik indeks değeri bire eşittir.

$$U = \frac{LCI_{\max}}{LCI_{\min}} \tag{2.25}$$

2.5 Bu Araştırmaya Konu Olan Çalışma Uzayı

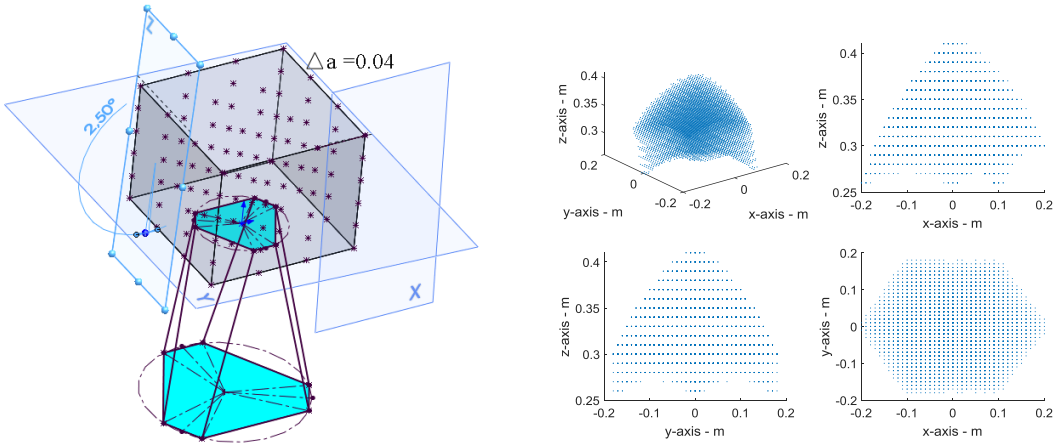
Bu tezde, performans değerlendirmesi ve optimizasyon için tanımladığımız hedef çalışma uzayı aşağıdaki gibidir.

P_x = X doğrultusundaki koordinatlar -0.06 m 'den +0.06 m ye 0.04 m düzenli artımlar ile

P_y = Y doğrultusundaki koordinatlar -0.06 m 'den +0.06 m ye 0.04 m düzenli artımlar ile

P_z = Z doğrultusundaki koordinatlar +0.3 m to +0.4 m in 0.02 m düzenli artımlar ile

α, β, γ = Düzlemlerin açısal değişimleri -5° 'den +5° 'ye 2.5° lik düzenli artımlar ile



Şekil 2.3 Optimize edilecek SP'nin çalışma uzayının ve genel çalışma uzayının hacimsel gösterimi

Dolayısıyla, toplamda 12000 uzaysal nokta koordinatlarına, 6-DOF PRS ile, erişebilmek için kinematik parametrelerin optimizasyonu gerçekleştirilecektir. Bunun için ilk olarak, tanımlanmış çalışma uzayına göre, yerel koşul sayıları *LCI Frobenius normunda* çalışma uzayı içindeki 12000 nokta için hesaplanmış; sonrasında, global koşul sayısı *GCI* hesaplanmıştır.

"Şekil 2.3" 3x3 SP için belirlediğimiz çalışma uzayını temsil etmektedir. Çalışma uzayının her bir ekseni için farklı "Δa" (artım) değerleri kullanılmıştır. Örneğin y ekseni için Δa = 0.04 m, z ekseni için 0.02 m'dir.

2.6 Ardışık Kuadratik Programlama (SQP) ile Optimizasyon

Optimizasyondaki öncelikli hedef becerikliliği ve platformun kullanılabilir çalışma uzayını, tasarım parametrelerini değiştirerek arttırmaktır. Tasarım parametrelerinin (r_b, r_p, l_i) optimizasyonu için bu tezde SQP metodu kullanılmıştır. Hedef fonksiyon tasarım parametrelerinin min. ve max. değerleri optimizasyon prosedüründen önce belirlenmiştir. Diğer bir kısıt ise *Z (çalışma uzayındaki noktaların* LCI *değerleri*) olarak belirlenmiştir. Bütün Z (LCI) değerleri önceden belirlenmiş çalışma uzayının (Şekil 2.3) her bir noktası için hesaplanmıştır.

2.6.1 Maliyet Fonksiyonu Ataması

Bu tezde, PRS'in SQP tekniğiyle optimizasyonu için Matlab Optimization Toolbox (fmincon) kullanılmıştır. Optimizasyon prosedürü maksimum beceriklilik ve erişilebilir çalışma uzayı için üç aşamada gerçekleştirilmiştir:

- Tek değişken optimizasyonu: r_p değerinin optimize edilmesi hedeflenmiş
- İki değişken optimizasyonu: r_p, *ve* r_b değerlerinin optimize edilmesi hedeflenmiş
- Sekiz değişken optimizasyonu: r_b, r_p, l_i değerlerinin optimize edilmesi hedeflenmiş

Bütün optimizasyon yaklaşımları için, bir alt maliyet fonksiyonu tanımlanmıştır;

- Tek değişken için alt maliyet fonksiyonu

$$f_1(x): r_p \cdot (Z-1)$$

- İki değişken için alt maliyet fonksiyonu

$$f_2(x): r_p \cdot r_b \cdot (Z-1)$$

- Üç grup değişkenin alt maliyet fonksiyonu

$$f_3(x): r_p \cdot r_b \cdot Ortalama\ (l_i)\ (Z-1)$$

Bütün durumlar için toplam maliyet fonksiyonu

$$g(x): \sum_i Alt\ maliyet\ f_i(x)$$

Maliyet fonksiyonu yukarıda tanımlandığı şekilde belirlenmiştir. Böylelikle minimum alt maliyet fonksiyonları f(x) değerlerinin toplamından oluşan minimum g(x) değerleri

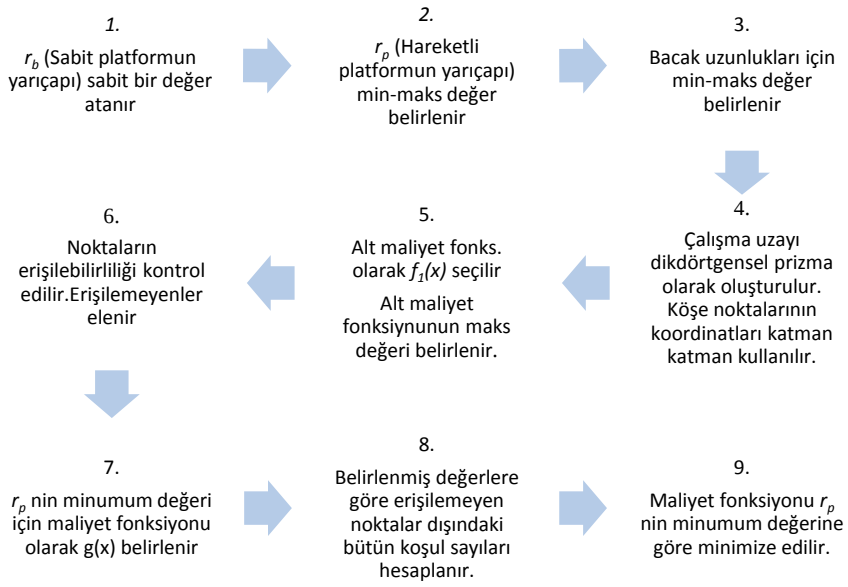
araştırılmıştır. Bunun anlamı minimum f(x) değerlerinin bulunması ihtiyaçtır. Minimum f(x) değerlerini araştırırken, iki kriter sunulmuştur. Bunlardar biri minimum *Z* ve minimum r_p ye ihtiyaç duyulması ve ikinci kriter erişilebilir çalışma uzayı olmasıdır. *Z* asla 1'den küçük olmadığı için algoritma buna göre kurulmuştur. Çalışma alanında hedeflenen konuma erişilemiyorsa, erişilemeyen değerlendirme noktasını ortadan kaldırmak ve ilgili *Z* değerini hesaplamamak için, *Z* değerine büyük bir değer atanır. Böylelikle minimum f(x)'i bulmak için yapılan gereksiz hesaplamalardan, pozisyona karşılık gelen değerden kaçınılır. Alt maliyet fonksiyonunu minimum yapan iki değişkenin r_p ve r_b minimumlarını araştırmak için, alt maliyet fonksiyonu $f_2(x)$'i belirlerken bu iki değişken çarpılır. Böylelikle maliyet fonksiyonu minimize edilirken değişkenler optimize edilir. Becerikliliğin göstergesi olan *Z* (koşul sayısı) değeri maksimum 1000 olacak şekilde kısıtlanmıştır. Bacak uzunlukları sınırlanmış; ve çalışma uzayı içerisindeki koordinatlara erişilip erişilemediği kontrol edilmiştir. Alt maliyet fonksiyonu f(x)'i minimum yapan üç değişkenin r_p, r_b, l_i minimumunu araştırırken, üç değişkeni r_p, r_b (ortalama) l_i birbirleriyle ve (Z-1) değeriyle çarpılarak alt maliyet fonksiyonu $f_3(x)$ olarak belirlenir. Böylece maksimum erişilebilir tekillikten uzak çalışma uzayı için parametreler optimize edilmiş olur.

2.6.2 Bir, İki ve Sekiz Değişkenli Optimizasyon Algoritmaları

Şekil 2.4 tek değişkenli optimizasyon prosedürünü sırasıyla anlatan şematik görünümdür. Tek değişkenli optimizasyon için algoritmada ilk olarak optimizasyonu gerçekleştirmek için diğer değişkenlerin değerleri sabitlenir; optimize edilecek olan değişkenin kısıtları yani minimum ve maksimum değerleri belirlenir. İkinci olarak, erişilmek istenen noktaların çalışma uzayında olup olmadığı, yani erişilebilir ya da erişilemez olup olmadığının anlaşılması için bacak uzunlukları sınırlandırılmıştır. Devamında hangi alt maliyet fonksiyonu için hesap yapılacağı; $f_1(x)$ olarak tek değişken için veya; $f_2(x)$ olarak iki değişken için belirlenir. Dolayısıyla tek değişken için $f_1(x)$ seçilir. Seçilmiş olan $f_i(x)$ için limit değer belirlenir. Sonrasında erişilmek istenen noktanın çalışma uzayında olup olmadığı kontrol edilir, erişilemez olanlar elenir. Böylece optimizasyon için gereksiz hesaplamalardan kaçınmış olunur. Devamında maliyet fonksiyonu $g(x)$ tanımlanır ve koşul sayısı *Z* erişilebilen her bir nokta için hesaplanır; ve

r_p nin optimal değerini *f(x)* 'lerin toplamını minimum yapan değer (veya *g(x)*) için araştırılır.

Benzer bir algoritma iki değişkenli, (r_p ve r_b) optimizasyon için ufak bir değişiklikle gerçekleştirilebilir. İki değişkeni optimize ederken; Şekil 2.4 'teki ilk adımda r_b değerini sabitlemek yerine r_b nin'de min-max değerleri belirlenir. İkinci değişiklik, beşinci adımda alt maliyet fonksiyonu olarak $f_2(x)$ seçilir. Benzer şekilde sekiz değişkenli optimizasyon için üç grup değişkenin birden minimum ve maksimum değerleri belirlenir. Alt maliyet fonksiyonu olarakta $f_3(x)$ kullanılır.



Şekil 2.4 Tek değişkenli optimizasyonda kullanılan algoritmanın adımları

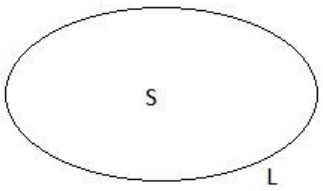
BÖLÜM 3

OPTİMİZASYON

3.1 Optimizasyonun Tarihi

Bir problemin, birden çok çözümü varsa doğal olarak herkes içlerinden en iyisini seçmeyi isteyecektir. Bununla birlikte, doğada birçok süreçte optimizasyonun kendiliğinden gerçekleştiği görülmektedir. Ünlü matematikçi Leonhard Euler "Doğada bir maksimum ve bir minimum anlamı taşımayan hiçbir şey yoktur" diyerek optimizasyonun önemini belirtmiştir. Optimal sözcüğünün kökü Latince "optimus" sözcüğünden gelmektedir; ve "en iyi" anlamına gelir. Optimal veya en iyi çözümü bulmak için minimum veya maksimum problemi çözmek şarttır. Bazı kaynaklarda maksimum (en büyük) ve minimum (en küçük) terimleri ekstremum (uç) ifadesiyle de tanımlanmış olabilir; bu sebeple ki maksimum ve minimum problemlere "ekstremal problemler" de denilir.

Optimal problemlerin geçmişi çok eskilere dayanır; ilk optimal problem ise: Düzlemde uzunluğu verilmiş kapalı eğriler içerisinden öyle birisini bulunuz ki, çevrelediği alan maksimum olsun. Bu problem "Dido problemi" olarak bilinir.



Şekil 3.1 Dido problemi

M.Ö. 5.yüzyıldan başlayarak Aristoteles, Archimedes, Euclid, Apollonius gibi bir çok ünlü bilim adamlarının eserlerinde optimizasyon problemlerine rastlanmaktadır. Rönesans döneminde, bilimsel çalışmaların yeniden hareketlendiği bir dönemde, minimum ve maksimum problemleri birçok bilim adamını bu alana çekmiştir. On yedinci yüzyılda optimal problemleri araştırma zorunluluğu doğadaki süreçlerin meydana çıkardığı problemler ile ilgiliydi. W.Snellius'un deneysel olarak bulduğu ışının yansıma kanununu açıklamak için dahi Fransız matematikçi P.Fermat tahminen 1660 yılında ekstremal ilkesini ortaya atmıştır. Bu ilke sonraları Fermat ilkesi olarak adlandırılmıştır. Buna göre ışın iki nokta arasındaki mesafeyi geçerken, geçiş süresini minimum kılacak bir yörünge izlemektedir. Fermat ilkesinden sonra "varyasyon ilkeleri", yani doğa kanunlarının ekstremal kanunlardan elde edilmesi temel bir yaklaşım olmuştur. Maksimum ve minimum problemleri çözmek için ilk genel yöntem P.Fermat tarafından 1629 yılında verilmiştir. Fermat bir P(x) polinomunun ekstremum noktalarının, (3.1) denkleminin kökü olduğunu ispat etmiştir.

$$\left.\frac{P(x+h)-P(x)}{h}\right|_{h=0} = 0 \qquad (3.1)$$

Yaklaşık 50 yıl sonra 1680' lerde Newton ve Gottfried Wilhelm von Leibniz ile bir fonksiyonun türevi kavramı matematiğe girmiştir. Yukarıdaki denklem $P'(x) = 0$ şeklinde çağdaş matematik dilinde yazılmaktadır. Bu yöntem matematik kitaplarına Fermat ilkesi olarak girmiştir. On sekizinci yüzyılda L.Euler ve J.Lagrange tarafından çok değişkenli fonksiyonların kısıtlamasız ve eşitlik kısıtlaması olan optimizasyon problemleri için çözüm yöntemleri önerilmiştir. J.Lagrange Çarpanları Yöntemi olarak adlandırılan bu yöntem, 20.yüzyılın 40'lı yıllarında eşitlik ve eşitsizlikler ile verilen kısıtlamalı optimizasyon problemleri için geliştirildi. 1930-1940 yıllarında Sovyetler birliğinde Leonid Vitalyevich Kantorovic (1912-1986), sonraları ABD'de George Dantzig (1914-2005) eşitlik ve eşitsizlik türünden lineer kısıtlamalarla verilmiş kümelerde lineer fonksiyonların minimum problemini çözmek için lineer programlama teorisini geliştirmişlerdir. Bu problemler için L.V.Kantorovich'in geliştirdiği yöntemin G.Dantzig'in simpleks yönteminden (1947) çok az farklı olduğu belirtilir. [44]

Karmaşık mühendislik optimizasyon problemlerinin çözümü için kullanılan genel yöntemlerin yanı sıra, son yıllarda kullanımı yaygınlaşan modern optimizasyon teknikleri olarak isimlendirilen, güçlü ve popüler metotlar ortaya çıkmıştır. Bu metotların, Genetik Algoritmalar, Benzetimli Tavlama, Parçacık Sürüsü Optimizasyonu, Karınca Koloni Optimizasyonu, Sinir Ağı Tabanlı Optimizasyon ve Bulanık Optimizasyonlar gibi çeşitleri vardır. Genetik Algoritmalar bilgisayarlı arama ve optimizasyon algoritmalarıdır. Doğal genetik ve doğal seleksiyonun mekaniğine dayanılarak oluşturulmuştur. Genetik algoritmalar ilk olarak John Holland tarafından 1975'te önerilmiştir. Benzetimli tavlama yöntemi eritme yoluyla erimiş metallerin soğutma sürecinin mekaniğine dayanır. Yöntem başlangıçta Kirkpatrick, Gelatt ve Vecchi tarafından geliştirildi. Parçacık sürüsü optimizasyon algoritması, toplumsal organizmaların davranışlarını taklit eder. Böceklerin bir koloni veya sürüsü (örneğin, karıncalar, termitler, arılar ve arılar), kuşların sürüsü ve bir balık yuvası. Algoritma ilk olarak Kennedy ve Eberhart tarafından 1995'te önerildi. Karınca koloni optimizasyonu, karınca kolonilerinin birlikte yuvalarından yiyecek kaynağına giden en kısa yolu bulabilmeleri davranışına dayanmaktadır. Yöntem ilk kez 1992 yılında Marco Dorigo tarafından geliştirildi. Sinir ağı yöntemleri, sinir sisteminin paralel işleme yeteneği sayesinde muazzam miktarda duyusal veri varlığında algılama problemlerini çözebileceği muazzam hesaplama gücüne dayanmaktadır. Yöntem, başlangıçta Hopfield ve Tank tarafından 1985 yılında optimizasyon için kullanılmıştır. Bulanık optimizasyon yöntemleri, tasarım verilerini içeren hedef fonksiyonları ve açıklayıcı olmayan biçimlerde belirtilen kısıtlamalarla ilgili optimizasyon problemlerini çözmek için geliştirilmiştir. Bulanık yaklaşımlar mühendislik tasarımında tek ve çok amaçlı optimizasyon için ilk kez Rao tarafından 1986'da önerilmiştir. Gelişen teknoloji ve yüksek hızlı dijital bilgisayarların ortaya çıkmasıyla birlikte optimizasyon prosedürlerinin uygulanması ve yeni optimizasyon metodları üzerine çalışmalar yapılmaya devam edilmektedir. [45]

3.2 Optimizasyon Probleminin Tanımı

Bir optimizasyon problemi veya matematiksel programlama problemi aşağıdaki gibi tanımlanır.

$g_j(\mathbf{X}) \leq 0\,,\ \ j = 1,2,\ldots.,m$
$l_j(\mathbf{X}) = 0\,,\ \ j = 1,2,\ldots.,p$ kısıtlarına göre

$$f(x) \text{ değerini minumum yapacak } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ . \\ x_n \end{Bmatrix} \text{ değerini bulunuz} \qquad (3.2)$$

X n boyutlu tasarım vektörü, f(**X**) hedef fonksiyonu, $g_j(\mathbf{X})$ ve $l_j(\mathbf{X})$ eşitsizlik ve eşitlik kısıtlarıdır. Değişken sayısı n, kısıt sayısı m ve/veya p ile ifade edilmiştir. (3.2) denklemleriyle kısıtlı bir optimizasyon problemi tanımlanmış olur. Bazı optimizasyon problemleri hiçbir kısıt içermez. Bu tip optimizasyon problemleri kısıtsız optimizasyon problemi olarak tanımlanır; ve sadece aşağıdaki gibi ifade edilir.

$$f(x) \text{ değerini minumum yapacak } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \cdot \\ . \\ x_n \end{Bmatrix} \text{ değerini bulunuz} \qquad (3.3)$$

3.2.1 Tasarım Vektörü

Herhangi bir mühendislik sistemi veya bileşeni, bir değerler kümesiyle tanımlanır; bunlardan bazıları tasarım süreci boyunca değişken olarak görülür. Genel olarak, belirli değerler genellikle başlangıçta sabitlenir ve bunlara önceden atanan parametreler denir. Diğer değerler tasarım sürecinde değişken gibi davranır ve bunlar tasarım veya karar değişkeni diye isimlendirilir ; x_i , i = 1,2,…,n. Toplu olarak tasarım vektörü şeklinde ifade edilir. Kısıtlar tasarım değişkenlerinin fiziksel sınırlarını ortaya koyar; kullanılabilirlik, üretilebilirlik ve taşınabilirlik, geometrik veya yan kısıtlamalar olarak da bilinir.

3.2.2 Tasarım Kısıtları

Pratikte birçok problemde tasarım değişkenleri gelişi güzel seçilemez; özellikle belirlenmiş fonksiyonları ve diğer gereklilikleri sağlamak zorundadır. Kabul edilebilir bir tasarım için uyulması gereken kısıtların bütünü tasarım kısıtları olarak isimlendirilir.

Sistemin davranışını veya performans limitlerini belirleyen kısıtlar, davranış veya fonksiyonel kısıtlar olarak isimlendirilir.

3.2.3 Hedef Fonksiyon

Geleneksel tasarım prosedürleri, sadece sorunun işlevsel ve diğer gereksinimlerini karşılayan, kabul edilebilir veya uygun bir tasarım bulmayı amaçlar. Genel olarak, birden fazla kabul edilebilir tasarım olacaktır; ve optimizasyonun amacı mevcut birçok kabul edilebilir tasarımdan en iyisini seçmektir. Böylece bir kriter, farklı alternatif kabul edilebilir tasarımları karşılaştırmak ve en iyisini seçmek için seçilmelidir. Optimize edilen tasarım kriteri, tasarım değişkenlerinin bir fonksiyonu olarak ifade edildiğinde, kriter veya hedef fonksiyon olarak tanımlanır. Hedef fonksiyonun seçimi problemin doğası tarafından yönetilir. Uzay ve hava araçlarında genellikle hedef fonksiyon olarak ağırlıkları yapısal tasarım problemi kabul edilerek minimizasyonu için çalışılır.

Bazı durumlarda eş zamanlı olarak sağlanması gereken birden fazla kriter olabilir. Örneğin bir dişli çifti minimum ağırlıkta tasarlandığı gibi belirli bir gücü maksimum verimlilikle aktarmak için tasarlanma zorunluluğu olabilir. Birden fazla hedef fonksiyonu içeren bir optimizasyon problemi *çok amaçlı programlama problemi* olarak da bilinir. Çok amaçlı optimizasyon problemlerinde amaçların çakışma olasılığı ortaya çıkar; bu sebeple optimizasyon probleminin çözümü için hedef fonksiyonların lineer kombinasyonlarıyla problemi ifade edecek tek bir hedef fonksiyon oluşturulur. Şöyle ki eğer $f_1(\mathbf{X})$ ve $f_2(\mathbf{X})$ iki ayrı hedef fonksiyonu tanımlıyorsa tüm hedef fonksiyonlar için tek bir hedef fonksiyon şu şekilde tanımlanır.

$$f(\mathbf{X}) = \alpha_1 f_1(\mathbf{X}) + \alpha_2 f_2(\mathbf{X}) \tag{3.4}$$

Denklem (3.4) deki sabitler α_1 ve α_2 bağıl olarak hedef fonksiyonların önemlerini ifade eder.

3.3 Optimizasyon Yöntemleri

Daha önceki kısımlarda optimizasyon problemlerinin klasik optimizasyon yöntemleri ve gelişen teknoloji ve bilgisayarların kullanılmaya başlamasıyla birlikte modern optimizasyon yöntemleri ile birlikte çözümlendiğinden bahsedilmişti. Optimizasyonda

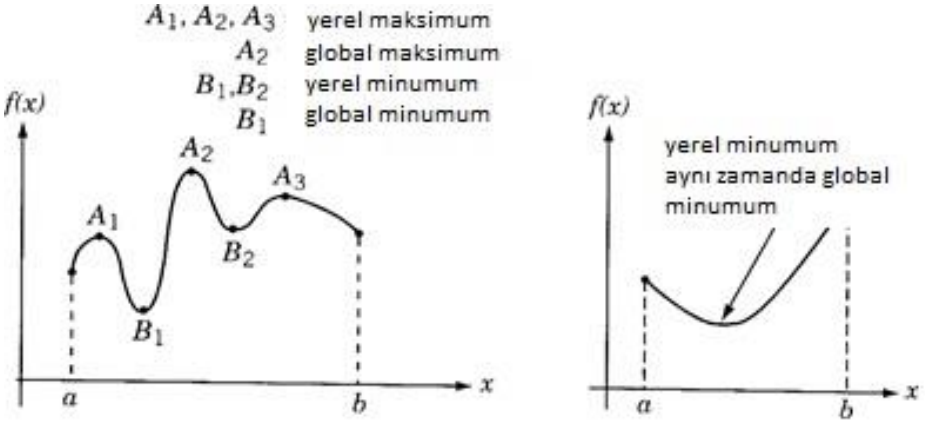
kullanılan modern yöntemlerden biri olan SQP yönteminin daha iyi anlaşılması için ilk olarak yöntemin temellerinin dayandığı klasik optimizasyon yöntemleri incelenecektir.

3.3.1 Klasik Optimizasyon Yöntemleri

Klasik optimizasyon metotları sürekli ve türevlenebilir fonksiyonların optimal çözümünü bulmak için kullanılır. Bu metotlar analitiktir ve diferansiyel hesap teknikleri kullanılır. Bazı pratik problemler sürekli ve türevlenemeyen fonksiyonlar içerdikleri için klasik optimizasyon tekniklerinin kullanımı pratik uygulamalarda sınırlıdır. Ancak nümerik optimizasyon metodlarının çoğunun geliştirilmesi için bir temel oluşturur. Bu bölümde tek değişkenli, çok değişkenli kısıtsız ve çok değişkenli eşitlik ve eşitsizlik kısıtlı bir fonksiyonun optimal çözümünün bulunması için gerek ve yeter koşulların neler olduğunu açıklayacağız.

3.3.1.1 Tek Değişkenli Optimizasyon

Yeterince küçük pozitif ve negatif bütün h değerleri için $f(x^*) \leq f(x^* + h)$ ise f(x) fonksiyonu $x = x^*$da yerel minimuma sahiptir. Benzer şekilde f(x) fonksiyonu eğer $f(x^*) \geq f(x^* + h)$ ise sıfır'a yeterince yakın bütün h değerleri için $x = x^*$ da yerel maximuma sahiptir. F(x) in tanımlı olduğu bölgedeki bütün x değerleri için sadece x^* a yakın olduğu noktalar değil, $f(x^*) \leq f(x)$ ise, $x = x^*$ da f(x) mutlak veya bölgesel minumuma sahiptir. Benzer olarak tanımlı bölgedeki bütün x değerleri için, $f(x) \geq f(x^*)$ ise, $x = x^*$ da f(x) mutlak veya bölgesel maksimuma sahiptir.



Şekil 3.2 x^* noktasında yerel ve bölgesel minimum (maksimum) noktalar

Tek değişkenli optimizasyon problemi; [a,b] aralığında f(x)'i minimize eden $x = x^*$ değerinin bulunmasıdır. Aşağıdaki iki teorem tek değişkenli bir fonksiyonun yerel minumumu için gerek ve yeter şartları verir. Şekil (3.2) da yerel ve bölgesel minimum (maksimum) noktalar gösterilmiştir.

Teorem $a \leq x \leq b$ aralığında tanımlı bir f(x) fonksiyonu $a \leq x^* \leq b$ olmak üzere $x = x^*$ da yerel minumuma sahipse $x = x^*$ da türevi $f'(x)$ var ve sonlu ise o zaman $f'(x^*) = 0$ dır.

İspat $f'(x^*) = \lim_{h\to\infty}\left(\frac{f(x^*+h)-f(x^*)}{h}\right)$ tanımlı ve var olduğu verilmiş x^* yerel minimum nokta olarak verildiğinden sıfıra yeterince yakın bütün h değerleri için,

$$f(x^*) \leq f(x^* + h) \tag{3.5}$$

olur. Bu yüzden

$$\frac{f(x^* + h) - f(x^*)}{h} \geq 0, h > 0 \; ise \tag{3.6}$$

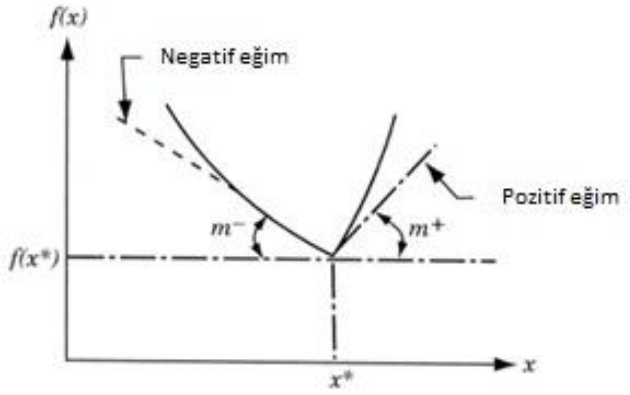
Ve

$$\frac{f(x^* + h) - f(x^*)}{h} \leq 0, h < 0 \; ise \tag{3.7}$$

yazılabilir.

Sonuç olarak h'ın sıfır'a yaklaşan pozitif değerleri için $f'(x^*) \geq 0$, h'ın sıfır'a yaklaşan negative değerleri için $f'(x^*) \leq 0$ dır. Bu iki durumu da sağlaması için $f'(x^*) = 0$ olması zorunludur. Bu teorem için aşağıdaki durumlar mevcuttur.

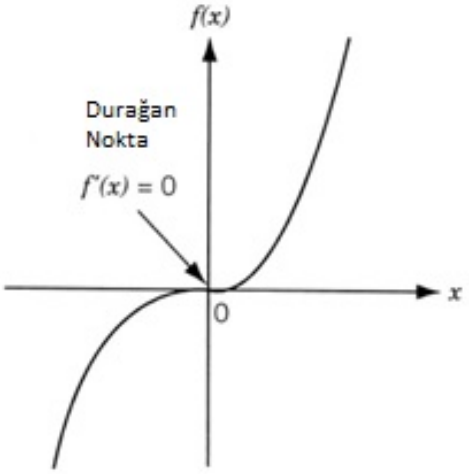
- Bu teorem x^* relative maximum olsa bile ispatlanabilir
- Bu teorem x^* noktasında bir türev yoksa bu nokta da bir maximum veya minimum oluşup oluşmayacağı hakkında birşey söyleyemez.
- "h" değeri 0'a negative veya pozitif değerlerden yaklaştıkça duruma göre $\lim_{h\to\infty}\left(\frac{f(x^*+h)-f(x^*)}{h}\right) = m^+ veya\, m^-$ olur. Burada m^+ ve m^- değerleri eşit olmadıkça $f'(x^*) = 0$ olmaz dolayısıyla türev olmaz; teorem uygulanamaz.



Şekil 3.3 x^* noktasında bir türev tanımsız iken

Teorem fonksiyonun tanım aralığının uç noktalarında bir minimum ve maksimum oluşursa ne olacağını söylemez. Bu durumda, $\lim_{h\to\infty}\left(\frac{f(x^*+h)-f(x^*)}{h}\right)$ türev yanlızca h'ın pozitif ve negative değerlerinde oluşacağından uç noktolarda türev tanımlı değildir.

Teorem fonksiyonun türevinin sıfır olduğu her noktada bir minimum veya maksimum oluşacağını söylemez. Yani bir noktada maximum veya minimum varsa o noktada fonksiyonun türevi kesinlikle sıfırdır ancak tersi durumda bu geçerli değildir. Örneğin şekil (3.4) da x=0 noktasında $f'(x^*)=0$ a eşit olmasına rağmen o noktada ne bir minimum ne de maksimum oluşmamıştır.



Şekil 3.4 Fonksiyonun durağan (büküm) noktası

$f'(x^*)=0$ şartını sağlayan noktanın minimum veya maksimum nokta olduğunu gösteren yeter şartlar aşağıdaki teoremlerle elde edilir.

Teorem Yeter Şartlar $f'(x^*) = 0, f''(x^*) = 0, \ldots. f^{n-1}(x^*) = 0, f^n(x^*) \neq 0$ ise

$f^n(x^*) > 0$ ve n çift ise $x = x^*$ da f(x) yerel minimum noktasına sahiptir.

$f^n(x^*) < 0$ ve n çift ise $x = x^*$ da f(x) yerel maximum noktasına sahiptir.

n tek ise $x = x^*$ da ne bir minimum ne de maximum noktası olur. Yani o nokta fonksiyonun büküm noktasıdır.

İspat fonksiyonun x^* civarında Taylor açılımı yazılırsa,

$$f(x^* + h) = f(x^*) + hf'(x^*) + \frac{h^2}{2!} f^n(x^*) +$$

$$+ \cdots + \frac{h^{n-1}}{(n-1)!} f^{n-1}(x^*) + \frac{h^n}{n!} f^n(x^* + h\theta), 0 < \theta \leq 1 \qquad (3.8)$$

$f'(x^*) = f''(x^*) = \cdots = f^{n-1}(x^*) = 0$ olduğundan (3.8) eşitliği $f(x^* + h) - f(x^*) = \frac{h^n}{n!} f^n(x^* + h\theta)$ olur. $f^n(x^*) \neq 0$ olduğunda x^* civarında öyle bir aralık bulunabilir ki buradaki her x için n. türev $f^n(x)$, $f^n(x^*)$ ile aynı işaretlidir. Bu yüzden bu aralığın her $x^* + h$ noktasında $f^n(x^* + h\theta)$, $f^n(x^*)$ ile aynı işaretli olur. "n" çift olduğunda $\frac{h^n}{n!}$, "h" ın işaretine bağlı olarak bazen pozitif, bazen negatif olur. Öyleyse x^* noktasında bu durumda ne bir maksimum ne bir minimum söz konusu olur; x^* noktası büküm noktası olur.

3.3.1.2 İki Değişkenli Optimizasyon

F(x_1, x_2) fonksiyonunun minimum (maksimum) noktası için gerek şartlar $\frac{\delta f}{\delta x_1} = 0$ ve $\frac{\delta f}{\delta x_2} = 0$ dır. Yeterli şartları elde etmek için h = $x - x^*$ olmak üzere f(x_1, x_2) nin (x_1, x_2) civarındaki Taylor açılımı göz önüne alınırsa

$$f(x^* + h) = f(x^*) + \sum_{i=1}^{2} h_i \frac{\delta f}{\delta x_i}(x^*) + \frac{1}{2} \sum_{i=1}^{2} \sum_{j=1}^{2} h_i h_j \frac{\delta^2 f}{\delta x_i \delta x_j}\Bigg|_{x = x^* + h\theta}$$

$$f(x^* + h) - f(x^*) = \frac{1}{2!}\left[h_1^2 \frac{\delta^2 f}{\delta x_1^2} + 2h_1 h_2 \frac{\delta^2 f}{\delta x_1 \delta x_2} + h_2^2 \frac{\delta^2 f}{\delta x_2^2}\right] \qquad (3.9)$$

$u = \frac{h_2}{h_1}$ olsun.

$$A = \left.\frac{\delta^2 f}{\delta x_1^2}\right|_{x=x^*+h\theta}, B = \left.\frac{\delta^2 f}{\delta x_1 \delta x_2}\right|_{x=x^*+h\theta}, C = \left.\frac{\delta^2 f}{\delta x_2^2}\right|_{x=x^*+h\theta}$$

$$f(x^* + h) - f(x^*) = \frac{h_1^2}{2!}[A + 2Bu + Cu^2] = \frac{h_1^2}{2!}Q(u) \quad (3.10)$$

$$f(x^* + h) - f(x^*) = \frac{h_1^2}{2!}Q(u) \quad (3.11)$$

$x^* = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ in yerel maksimum olması için her u için $Q(u) < 0$ olması gerekir; x^* ın yerel minimum olması için her u için $Q(u) > 0$ olması gerekir. $Q(u)$ fonksiyonunun reel kökleri varsa Q , u'nun seçilen değerlerine bağlı olarak ya pozitif ya negative ya da sıfır olur. $Q(u)$ reel kökleri yoksa Q , u'nun her değeri için ya pozitif ya da negative olur.

$Q(u)$ nun reel köklerinin olmadığı şartlar aşağıdadır:

$Q(u) = 0 = A + 2Bu + Cx^2$ olsun. $Q(u)$ nun kökleri $u = \frac{-B \pm \sqrt{B^2 - 4AC}}{C}$ olur. $B^2 - 4AC < 0$ ise $Q(u)$ eşlenik komleks olur; bu durumda her u için $Q(u)$ her zaman pozitif ya da her zaman negative olur. Bu durum bir extremumu gösterir. Eğer $B^2 - 4AC > 0$ ise u nun değerlerine bağlı olarak bazen pozitif bazen negative olabilir. Bu da eyer noktasını gösterir. $B^2 - 4AC = 0$ belirsizlik durumunu gösterir.

$B^2 - 4AC < 0$ olsun. A ve C aynı işaretli olur (değilse AC negative olacak) ve $B^2 - 4AC$ pozitif olur, bu da $B^2 - 4AC < 0$ ile çelişir.

Eğer A negative ve u = 0 ise o zaman $Q(u) = A < 0$ dır ve buradan $Q(u) < 0$ (her u için) olur. Bu da yerel maksimum demektir. $A < 0,\ A + C < 0$ olur.

Diğer taraftan $A > 0$ ise $A + C > 0$ ve her u için $Q(u) > 0$ olur. Bu da x^* noktasında yerel minimum olduğunu gösterir. Fonksiyon sürekli ve x^* ile $x^* + h\theta$ noktasında tüm türevler sürekli olduğundan bu noktadaki türevler aynı işaretli olurlar. Öyleyse

$$A_* = \left.\frac{\delta^2 f}{\delta x_1^2}\right|_{x^*}, B_* = \left.\frac{\delta^2 f}{\delta x_1 \delta x_2}\right|_{x^*}, C_* = \left.\frac{\delta^2 f}{\delta x_2^2}\right|_{x^*} \quad (3.12)$$

Olurlar. Sonuçlar özetlenirse:

- $B_*^2 - 4A_*C_* < 0$ ve $A_* + C_* < 0$ ise x^* da yerel maksimum vardır
- $B_*^2 - 4A_*C_* < 0$ ve $A_* + C_* > 0$ ise x^* da yerel minumum vardır
- $B_*^2 - 4A_*C_* > 0$, x^* da eyer noktası vardır.
- $B_*^2 - 4A_*C_* = 0$, x^* da belirsizlik söz konsudur.

3.3.1.3 Kısıtsız Çok Değişkenli Optimizasyon

Kısıtsız çok değişkenli bir fonksiyonun maksimum veya minumumu için gerek ve yeter şartlar aşağıdadır.

Tanım *fonksiyonun r. diferansiyeli* : f fonksiyonunun $\mathbf{X}^*$ noktasında $r > 1$ dereceli bütün kısmi türevleri var ve sürekli ise

$$d^r f(\mathrm{X}^*) = \sum_{i=1}^{n}\sum_{j=1}^{n} \ldots\ldots \sum_{k=1}^{n} h_i h_j \ldots . h_k \frac{\delta^r f(\mathrm{X}^*)}{\delta x_i \delta x_j \ldots \delta x_k} \tag{3.13}$$

Ifadesine X^* da *f* in *r.* diferansiyeli denir; *r* = 2 ve *n* = 3 durumunda

$$d^r f(\mathrm{X}^*) = d^2 f(x_1^*, x_2^*, x_3^*) = \sum_{i=1}^{3}\sum_{j=1}^{3} h_i\, h_j \frac{\delta^2 f(\mathrm{X}^*)}{\delta x_i \delta x_j}$$

$$= h_1^2 \frac{\delta^2 f}{\delta x_1^2}(\mathrm{X}^*) + h_2^2 \frac{\delta^2 f}{\delta x_2^2}(\mathrm{X}^*) + h_3^2 \frac{\delta^2 f}{\delta x_3^2}(\mathrm{X}^*) + 2h_1 h_2 \frac{\delta^2 f}{\delta x_1 \delta x_2}(\mathrm{X}^*)$$

$$= 2h_2 h_3 \frac{\delta^2 f}{\delta x_2 \delta x_3}(\mathrm{X}^*) + 2h_1 h_3 \frac{\delta^2 f}{\delta x_1 \delta x_3}(\mathrm{X}^*) \tag{3.14}$$

F(**X**) fonksiyonunun verilen bir (**X***) noktası civarında Taylor serisi açılımı yazılırsa:

$$f(\mathbf{X}) = f(\mathbf{X}^*) + df(\mathbf{X}^*) + \frac{1}{2!} d^2 f(\mathbf{X}^*) + \frac{1}{3!} d^3 f(\mathbf{X}^*)$$

$$+ \cdots + \frac{1}{N!} d^N f(\mathbf{X}^*) + R_N(\mathbf{X}^*, h) \tag{3.15}$$

son terim *kalan* diye isimlendirilir ve:

$$R_N(\mathbf{X}^*, h) = \frac{1}{(N+1)!} d^{N+1} f(\mathbf{X}^* + \theta h), 0 < \theta < 1 \; ve \; h = \mathbf{X} - \mathbf{X}^* \qquad (3.16)$$

Teorem Gereklilik Koşulu

Eğer *f*(**X**) , $\mathbf{X} = \mathbf{X}^*$,da maksimum veya minimum noktası varsa ve Eğer *f*(**X**) in birinci kısmi türevleri $\mathbf{X}^*$ da mevcut ise ,

$$\frac{\delta f}{\delta x_1}(\mathbf{X}^*) = \frac{\delta f}{\delta x_2}(\mathbf{X}^*) = \cdots = \frac{\delta f}{\delta x_n}(\mathbf{X}^*) = 0 \qquad (3.17)$$

İspat Birinci kısmi türevlerden biri, yani *k.* incisi , $\mathbf{X}^*$da sıfır olmasın , veya $\frac{\delta f}{\delta x_k}(\mathbf{X}^*) \neq 0$ olsun. Taylor açılımı ile:

$$f(\mathbf{X}^* + h) = f(\mathbf{X}^*) + \sum_{i=1}^{n} h_i \frac{\delta f}{\delta x_i}(\mathbf{X}^*) + R_1(\mathbf{X}^*, h) \qquad (3.18)$$

veya

$$f(\mathbf{X}^* + h) - f(\mathbf{X}^*) = h_k \frac{\delta f}{\delta x_k}(\mathbf{X}^*) + \frac{1}{2!} d^2 f(\mathbf{X}^* + \theta h), \; 0 < \theta < 1 \qquad (3.19)$$

yazılabilir; $d^2 f(\mathbf{X}^* + \theta h)$ de $\boldsymbol{h_i^2}$ çarpanı yer alacağından küçük h ler için h dereceden terimler h ın daha yüksek dereceli terimlerinden büyük olur. Bu nedenle $f(\mathbf{X}^* + h) - f(\mathbf{X}^*)$ ın işareti $h_k \frac{\delta f}{\delta x_k}(\mathbf{X}^*)$ ın işaretine bağlı olur. $h_k \frac{\delta f}{\delta x_k}(\mathbf{X}^*) > 0$ olduğunu varsayalım. O zaman $f(\mathbf{X}^* + h) - f(\mathbf{X}^*); h_k > 0$ için pozitif, $h_k < 0$ için de negatif olur.Bu $\mathbf{X}^*$ın ekstremum olamayacağını gösterir.Oysa $\mathbf{X}^*$ ekstremum nokta olarak alınmıştı. Aynı sonuç $\frac{\delta f}{\delta x_k}(\mathbf{X}^*) < 0$ olarak kabul edildiğinde de $\mathbf{X}^*$ın ekstremum olması için çeliştiği görülür. Öyleyse $\mathbf{X}^*$ bir ekstremumdur ve $\mathbf{X} = \mathbf{X}^*$ da , $\frac{\delta f}{\delta x_k} = 0$ olur.Bu $\mathbf{X}^*$ noktasında bütün kısmi türevlerin sıfır olması demektir.

Teorem Yeterlilik Koşulu

$\mathbf{X}^*$ durağan (büküm) noktasının *f(x)* fonksiyonunun bir ekstremum noktası olması için yeterli şart *f(x)* in $\mathbf{X}^*$ da hesaplanan ikinci kısmi türevlerin matrisinden oluşan Hessian Matrisi;

- Pozitif tanımlı olduğunda $\mathbf{X}^*$ yerel minimum noktası olur
- Negatif tanımlı olduğunda $\mathbf{X}^*$ yerel maksimum olur.

İspat Taylor teoreminden faydalanılarak $0 < \theta < 1$ olmak üzere,

$$f(\mathbf{X}^* + h) = f(\mathbf{X}^*) + \sum_{i=1}^{n} h_i \frac{\delta f}{\delta x_i}(\mathbf{X}^*) + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*+h\theta} \quad (3.20)$$

$\mathbf{X}^*$ durağan nokta olduğundan $\frac{\delta f}{\delta x_i} = 0, i = 1,2 \ldots n$

$$f(\mathbf{X}^* + h) - f(\mathbf{X}^*) = \frac{1}{2!}\sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*+h\theta}, 0 < \theta < 1 \quad (3.21)$$

Bu yüzden eşitliğin her iki tarafı da aynı işaretli olur. Çünkü $\frac{\delta^2 f(x)}{\delta x_i \delta x_j}$ ikinci kısmi türevi $\mathbf{X}^*$ komşuluğunda sürekli olacağından yeterince küçük bütün h lar için $\left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*+h\theta}$ ile $\left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*}$ işaretleri aynı olur. Böylelikle $Q = \sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*}$ pozitif ise $f(\mathbf{X}^* + h) - f(\mathbf{X}^*)$ pozitif olacak ve buradan da $\mathbf{X}^*$ yerel bir minimum nokta olacaktır. Bu Q ifadesi kareli bir formdur ve matris olarak $Q = h^T J h|_{\mathbf{X}=\mathbf{X}^*}$ şeklinde yazılabilir. Burada $J|_{\mathbf{X}=\mathbf{X}^*} = \left[\left.\frac{\delta^2 f}{\delta x_i \delta x_j}\right|_{\mathbf{X}=\mathbf{X}^*}\right]$ ikinci kısmi türevlerin matrisidir; ve f(**X**) in Hessian matrisi olarak isimlendirilir.

Matris cebirinden $h^T J h$ kareli formunun bütün h değerleri için pozitif olması $\mathbf{X}^*$ da J nin pozitif tanımlı olmasına bağlıdır. Bunun anlamı da $\mathbf{X}^*$ durağan noktasının yerel minimum olması için yeter şart aynı noktada hesaplanan Hesian matrisinin pozitif tanımlı olmasıdır. Bu da minimizasyon durumunu ispatlar. Benzer şekilde $\mathbf{X}^*$noktası yerel maksimum ise Hesian matrisi negatif tanımlı olacaktır.

Bir **A** matrisinin bütün özdeğerleri pozitif ise pozitif tanımlıdır. Aşağıdaki determinal eşitliği (3.22) sağlayan bütün λ değerleri pozitif olmalıdır. Benzer şekilde **A** matrisi negatif tanımlıysa bütün özdeğerleri negatif'dir.

$$|A - \lambda \mathbf{I}| = 0 \tag{3.22}$$

A matrisinin pozitif tanımlı olup olmadığını bulmak için başka bir yöntem daha vardır.

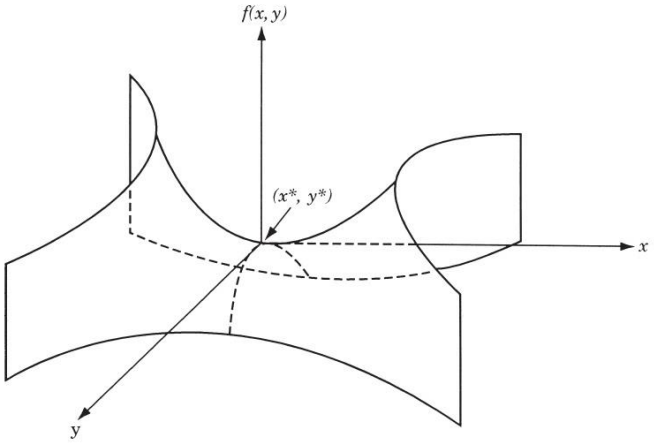
$$A = |a_{11}|, A_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, A_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix},$$

$$A_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} & . & a_{1n} \\ a_{21} & a_{22} & a_{23} & . & a_{2n} \\ a_{31} & a_{32} & a_{33} & . & a_{3n} \\ . & . & . & . & . \\ a_{n1} & a_{n2} & a_{n3} & . & a_{nn} \end{vmatrix} \tag{3.23}$$

A matrisi bütün A_1, A_2 ,… A_n determinantları pozitif olduğunda pozitif tanımlıdır. A_j nin işareti $(-1)^j$, i=1…n olmak üzere ise A matrisi negatif tanımlıdır. Bazıları pozitif bazıları sıfır ise A matrisi pozitif yarı tanımlı olarak isimlendirilir.

Eyer Noktası

İki değişkenli bir f(x,y) fonksiyonu için Hesian matrisi ne pozitif ne de negative tanımlı değil ise (x^*,y^*) noktasında ,ve $\frac{\delta f}{\delta x} = \frac{\delta f}{\delta y} = 0$ ise (x^*,y^*) noktası eyer noktası olarak isimlendirilir.



Şekil 3.5 Eyer noktası

3.3.1.4 Eşitlik Kısıtlı Çok Değişkenli Optimizasyon

Eşitlik kısıtlı sürekli fonksiyonların optimizasyonu,

$$\min f = f(x), \qquad g_j(\mathbf{X}) = 0, \qquad j = 1 \ldots . m$$

$$\mathbf{X} = (x_1, x_2 x_3 \ldots . x_n) \tag{3.24}$$

Bu problemde "m", "n" e eşit veya daha küçüktür. Bu problemin çözümü için çeşitli metodlar kullanılabilir. Bunlardan bazıları direk değişiklik, kısıtlı varyasyon ve Lagrange çarpanları metodlarıdır. Biz Lagrange çarpanları metodunu inceleyeceğiz.

Lagrange Çarpanları Metodu

Lagrange çarpanları metodunu başlangıçta basit olması için iki değişkenli bir kısıtlı bir problem için inceleyelim. Daha sonra m değişkenli n kısıtlı bir problem için genelleştirme yapacağız.

$$\min f(x_1, x_2)\ ,\ g(x_1, x_2) = 0\ için \tag{3.25}$$

Bu problemde extremum noktasının $\mathbf{X} = \mathbf{X}^*$ da oluşması için gereklilik koşulu aşağıdaki denklemdeki gibidir.

$$\min f(x_1, x_2)\ ,\ g(x_1, x_2) = 0\ için \tag{3.26}$$

$$\left(\frac{\delta f}{\delta x_1} - \frac{\delta f / x_2}{\delta g / x_2}\frac{\delta g}{\delta x_1}\right)\Bigg|_{(x_1^*, x_2^*)} = 0 \tag{3.27}$$

$$\lambda = -\left(\frac{\delta f / x_2}{\delta g / x_2}\right)\Bigg|_{(x_1^*, x_2^*)}, \qquad Lagrange\ çarpanı(\lambda) \tag{3.28}$$

$$\left(\frac{\delta f}{\delta x_1} + \lambda\frac{\delta g}{\delta x_1}\right)\Bigg|_{(x_1^*, x_2^*)} = 0 \tag{3.29}$$

$$\left(\frac{\delta f}{\delta x_2} + \lambda\frac{\delta g}{\delta x_2}\right)\Bigg|_{(x_1^*, x_2^*)} = 0 \tag{3.30}$$

Bu denklemlere ek olarak eşitlik kısıtı da ekstremum noktasında sağlanmalıdır.

$$g(x_1, x_2)|_{(x_1^*, x_2^*)} = 0 \tag{3.31}$$

Eşitlik (3.29) dan (3.31) e kadar olan kısım (x_1^*, x_2^*) noktalarının ektremum noktası olması için gereklilik koşullarını içerir. "λ" çarpanını belirleyebilmek için $\left(\frac{\delta f/x_2}{\delta g/x_2}\right)\Big|_{(x_1^*, x_2^*)}$ değerinin sıfırdan farklı bir değer olmak zorundadır. Çünkü dx_2 ifadesi (3.27) eşitliğinde dx_1 türünden ifade edilmiştir. Diğer taraftan eğer dx_1 terimini dx_2 türünden ifade etseydik $\left(\frac{\delta g}{\delta x_1}\right)\Big|_{(x_1^*, x_2^*)}$ ifadesinin sıfırdan farklı olma zorunluluğu doğacaktı. Dolayısıyla Lagrange çarpanlarıyla gereklilik koşulunun türetilmesinde ekstremum noktasında en azından $g(x_1, x_2)$ nin kısmi türevlerinden biri sıfırdan farklı olmalıdır. Eşitlik (3.29) dan (3.31) e kadar olan gereklilik koşulunu genel bir ifadeyle

Lagrange fonksiyonuyla gösterilirse:

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda g(x_1, x_2) \tag{3.32}$$

L yi üç değişkenli bir fonfsiyon gibi düşünerek ekstremumu için gereklilik koşulları aşağıdaki gibi ifade edilebilir.

$$\frac{\delta L}{\delta x_1}(x_1, x_2, \lambda) = \frac{\delta f}{\delta x_1}(x_1, x_2) + \lambda \frac{\delta g}{\delta x_1}(x_1, x_2) = 0$$

$$\frac{\delta L}{\delta x_2}(x_1, x_2, \lambda) = \frac{\delta f}{\delta x_2}(x_1, x_2) + \lambda \frac{\delta g}{\delta x_2}(x_1, x_2) = 0 \tag{3.33}$$

$$\frac{\delta L}{\delta \lambda}(x_1, x_2, \lambda) = g(x_1, x_2) = 0$$

Eşitlik (3.33) ile (3.29)'dan (3.31) e kadar olan eşitlikler aynı ifadelerdir.

Örnek Problem Lagrange çarpanları metodunu kullanarak $f(x, y) = kx^{-1}y^{-2}$fonksiyonunun $g(x, y) = x^2 + y^2 - a^2 = 0$ kısıtına göre optimal noktalarını bulun.

$$\frac{\delta L}{\delta x} = -kx^{-2}y^{-2} + 2x\lambda = 0 \tag{3.34}$$

$$\frac{\delta L}{\delta y} = -2kx^{-1}y^{-3} + 2y\lambda = 0 \tag{3.35}$$

$$\frac{\delta L}{\delta \lambda} = x^2 + y^2 - a^2 = 0 \tag{3.36}$$

Eşitlik (3.34) ve (3.35) kullanılarak eşitlik (3.37) elde edilir:

$$2\lambda = \frac{k}{x^3 y^2} = \frac{2k}{xy^4} \tag{3.37}$$

Dolayısıyla x ve y arasındaki ilişki $x^* = \left(\frac{1}{\sqrt{2}}\right) y^*$ olarak bulunur. (3.36) eşitliğinde bulunan sonuç yerine konularak $x^* = \left(a/\sqrt{3}\right)$, $y^* = \sqrt{2}\left(a/\sqrt{3}\right)$ olarak optimal sonuç bulunur.

Genelleştirilmiş gereklilik koşulu

(n değişekenli m eşitlik kısıtlı problem için)

$$\min f(\mathbf{X}) \text{ , } g_j(\mathbf{X}) = 0 \; için \; j = 1,2, \ldots . m \tag{3.38}$$

Böyle bir durumda Lagrange fonksiyonu,L,her bir kısıt $g_j(\mathbf{X})$ için bir tane λ_j Lagrange çarpanı tanımlanır.

$$\begin{aligned} L(x_1, x_2, \ldots, x_n, \lambda_1, \lambda_2 \ldots \lambda_m) \\ = f(\mathbf{X}) + \lambda_1 \, g_1(\mathbf{X}) + \lambda_2 \, g_2(\mathbf{X}) + \cdots \lambda_m \, g_m(\mathbf{X}) \end{aligned} \tag{3.39}$$

L 'yi n+m bilinmeyenli bir fonksiyon olarak düşünerek (3.38) verilen orjinal problemin çözümüyle ilişkili olarak ektremum noktasının olabilmesi için gereklilik koşulu aşağıdaki gibidir.

$$\frac{\delta L}{\delta x_i} = \frac{\delta f}{\delta x_i} + \sum_{j=1}^{m} \lambda_j \frac{\delta g_j}{\delta x_i} = 0, i = 1,2 \ldots . n \tag{3.40}$$

$$\frac{\delta L}{\delta \lambda_j} = g_j(\mathbf{X}) = 0, j = 1,2 \ldots . m \tag{3.41}$$

Eşitlik (3.40) ve (3.41) n+m bilinmeyenli ve n+m denklemi ifade etmektedir.Bu denklemlerin çözümü: $\mathbf{X}^* = \begin{Bmatrix} x_1^* \\ x_2^* \\ . \\ . \\ x_n^* \end{Bmatrix}$ ve $\boldsymbol{\lambda}^* = \begin{Bmatrix} \lambda_1^* \\ \lambda_2^* \\ . \\ . \\ \lambda_m^* \end{Bmatrix}$ $\mathbf{X}^*$ vektörü $f(\mathbf{X})$'in kısıtlı yerel minumumlarına karşılık gelir (yeterlilik koşulunu sağladığı düşünülerek); $\boldsymbol{\lambda}^*$ ise güvenilirlik bilgisine karşılık gelir.

Genelleştirilmiş yeterlilik koşulu

$f(\mathbf{X})$ fonksiyonunun $\mathbf{X}^*$ noktasında kısıtlı yerel minumumunun olması için yeterlilik koşulu aşağıdaki teoremle verilmiştir.

Teorem yeterlilik koşulu için quadratik Q aşağıda tanımlanmıştır.

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\delta^2 L}{\delta x_i \delta x_j}\, dx_i\, dx_j \qquad (3.42)$$

Şeklinde hesaplanır ve $\mathbf{X} = \mathbf{X}^*$ da d**X** in bütün değerleri kısıtların sağlanması için pozitif tanımlı olmalıdır.

İspat Eğer Q

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\delta^2 L}{\delta x_i \delta x_j}(\mathbf{X}^*, \boldsymbol{\lambda}^*)\, dx_i\, dx_j \qquad (3.43)$$

dx_i nin kabul edilebilir varyasyonlarının bütün seçenekleri için negatifse $\mathbf{X}^*$, $f(\mathbf{X})$'in kısıtlı maksimumu olacaktır. Kuadratik formda gösterilmiş olan eşitlik (3.42) nin $d\mathbf{X}$ bütün kabul edilebilir varyasyonlarında pozitif (negative) tanımlı olabilmesi için aşağıdaki determinal eşitlikte tanımlamış z_i polinomunun kökü pozitif ya da negative olmalıdır.

$$\begin{vmatrix} L_{11}-z & L_{12} & L_{13} & \dots & L_{1n} & g_{11} & g_{21} & \dots & g_{m1} \\ L_{21} & L_{22}-z & L_{23} & \dots & L_{2n} & g_{12} & g_{22} & \dots & g_{m2} \\ . & . & . & \dots & . & . & . & \dots & . \\ . & . & . & \dots & . & . & . & \dots & . \\ L_{n1} & L_{n2} & L_{n3} & \dots & L_{nn}-z & g_{1n} & g_{2n} & \dots & g_{mn} \\ g_{11} & g_{12} & g_{13} & \dots & g_{1n} & 0 & 0 & \dots & 0 \\ g_{21} & g_{22} & g_{23} & \dots & g_{2n} & 0 & 0 & \dots & 0 \\ . & . & . & \dots & . & . & . & \dots & . \\ g_{m1} & g_{m2} & g_{m3} & \dots & g_{mn} & 0 & 0 & \dots & 0 \end{vmatrix} = 0 \qquad (3.44)$$

$$L_{iJ=}\frac{\delta^2 L}{\delta x_i \delta x_j}(\mathbf{X}^*, \boldsymbol{\lambda}^*) \qquad (3.45)$$

$$g_{iJ=}\frac{\delta g_i}{\delta x_j}(\mathbf{X}^*) \qquad (3.46)$$

Eşitlik (3.44) ün genişlemesiyle z'de (n-m). dereceden bir polinom oluşturur. Eğer bu polinomun köklerinin bazıları pozitifken bazıları negatifse $\mathbf{X}^*$ noktası ekstremum nokta değildir.

Örnek Problem

Lagrange çarpanları metodunu kullanarak $f(x,y) = \pi x_1^2 x_2$ fonksiyonunun değerini $2\pi x_1^2 + 2\pi x_1 x_2 = 24\pi = A_0$ kısıtına göre maksimize edin.

Problemin Lagrange fonksiyonu:

$$L(x_1, x_2, \lambda) = \pi x_1^2 x_2 + \lambda(2\pi x_1^2 + 2\pi x_1 x_2 - A_0) \qquad (3.47)$$

Fonksiyonun maksimumu için gereklilik koşulları:

$$\frac{\delta L}{\delta x_1} = 2\pi x_1 x_2 + 4\pi\lambda x_1 + 2\pi\lambda x_2 = 0 \qquad (3.48)$$

$$\frac{\delta L}{\delta x_2} = \pi x_1^2 + 2\pi\lambda x_1 = 0 \qquad (3.49)$$

$$\frac{\delta L}{\delta \lambda} = 2\pi x_1^2 + 2\pi x_1 x_2 - A_0 = 0 \qquad (3.50)$$

Eşitlik (3.48) ve (3.49) kullanılarak:

$$\lambda = -\frac{x_1 x_2}{2x_1 + x_2} = -\frac{1}{2}x_1 \tag{3.51}$$

$$x_1 = \frac{1}{2}x_2 \tag{3.52}$$

Eşitlik (3.50) ve (3.52) kullanılarak:

$$x_1^* = \left(\frac{A_0}{6\pi}\right)^{1/2}, x_2^* = \left(\frac{2A_0}{3\pi}\right)^{1/2}, \lambda^* = -\left(\frac{A_0}{24\pi}\right)^{\frac{1}{2}}, f^* = \left(\frac{{A_0}^3}{54\pi}\right)^{1/2}$$

Olarak bulunur. Eğer$A_0 = 24\pi$ ise $x_1^* = 2, x_2^* = 4, \lambda^* = -1, f^* = 16\pi$ sonucuna ulaşılır.

Sonucun doğruluğundan emin olmak için yeterlilik koşulu olan eşitlik (3.44) ü uygulanırsa:

$$L_{11} = \left.\frac{\delta^2 L}{\delta {x_1}^2}\right|_{(\mathbf{X}^*,\lambda^*)} = 2\pi x_2^* + 4\pi\lambda^* = 4\pi$$

$$L_{12} = \left.\frac{\delta^2 L}{\delta x_1 \delta x_2}\right|_{(\boldsymbol{X}^*,\lambda^*)} = L_{21} = 2\pi x_1^* + 2\pi\lambda^* = 2\pi$$

$$L_{22} = \left.\frac{\delta^2 L}{\delta {x_2}^2}\right|_{(\mathbf{X}^*,\lambda^*)} = 0$$

$$g_{11} = \left.\frac{\delta g_1}{\delta x_1}\right|_{(\mathbf{X}^*,\lambda^*)} = 4\pi x_1^* + 2\pi x_2^* = 16\pi$$

$$g_{12} = \left.\frac{\delta g_1}{\delta x_2}\right|_{(\mathbf{X}^*,\lambda^*)} = 2\pi x_1^* = 4\pi$$

Böylece Eşitlik (3.44) şu şekli alır:

$$\begin{vmatrix} 4\pi - z & 2\pi & 16\pi \\ 2\pi & 0 - z & 4\pi \\ 16\pi & 4\pi & 0 \end{vmatrix} = 0;\ buradan\ 272\pi^2 z + 192\pi^3 = 0\ ve\ z = -\frac{12}{17}\pi$$

z negatif olduğu için $(x_1^*, x_2^*), f$ fonksiyonunun maksimumlarıdır.

Lagrange çarpanlarının yorumu

Lagrange çarpanlarının fiziksel olarak anlamını bulmak için aşağıdaki tek eşitlik kısıtı içeren optimizasyon problemin çözümünü incelenirse.

$$\min f(\mathbf{X}) \, , \tilde{g}(\mathbf{X}) = \; b \; veya \; g(\mathbf{X}) = b - \tilde{g}(\mathbf{X}) = \; 0 \tag{3.53}$$

Tek kısıtlı problemi için *b* bir sabittir. Bu problem için gerek sartlar yazılırsa

$$\frac{\delta f}{\delta x_i} + \lambda \frac{\delta g}{\delta x_i} = 0 \, , i = 1{,}2 \ldots . , n \; ve \; g = 0 \tag{3.54}$$

Bu denklemlerin çözümü $\mathbf{X}^*, \lambda^*, f^* = f(\mathbf{X}^*)$ olsun. Hedef fonksiyonun optimal değeri üzerinde kısıttaki küçük bir değişimin etkisi için $b - g(\mathbf{X})$ kısıtının diferansiyeli

$$db - d\tilde{g} = 0 \; veya \; db = d\tilde{g} = \sum_{i=1}^{n} \frac{\delta g}{\delta x_i} dx_i \tag{3.55}$$

Eşitlik (3.54) aşağıdaki gibi yazılırsa

$$\frac{\delta f}{\delta x_i} + \lambda \frac{\delta g}{\delta x_i} = \frac{\delta f}{\delta x_i} - \frac{\delta g}{\delta x_i} = 0 \;\; veya \tag{3.56}$$

$$\frac{\delta \tilde{g}}{\delta x_i} = \frac{\delta f / \delta x_i}{\lambda} \;\; i = 1{,}2 \ldots . , n \tag{3.57}$$

Eşitlik (3.57) eşitlik (3.55) ile birleştirilirse

$$db = \sum_{i=1}^{n} \frac{1}{\lambda} \frac{\delta f}{\delta x_i} dx_i = \frac{df}{\lambda} \tag{3.58}$$

Elde ederiz. Çünkü;

$$df = \sum_{i=1}^{n} \frac{\delta f}{\delta x_i} dx_i \tag{3.59}$$

Bu yüzden eşitlik (3.58) bizi şu sonuçlara ulaştırır;

$$\lambda = \frac{df}{db} \ , \lambda^* = \frac{df^*}{db} \ \ veya \tag{3.60}$$

$$df^* = \lambda^* db \tag{3.61}$$

Böylece λ^* f'nin b ye göre x^* daki duyarlılığını (değişim oranını) veya f^* içindeki marjinal değişimin veya f^* içindeki artımsal değişimin b'ye göre x^* daki duyarlılığı gösterdiğini bu eşitlikten söyleyebiliriz.

3.3.1.5 Eşitsizlik Kısıtlı Çok Değişkenli Optimizasyon

Eşitsizlik kısıtlı sürekli fonksiyonların optimizasyonu problemi:

$$\text{minimize edin } f = f(\mathbf{X}), kısıtına\ göre\ \ g_j(\mathbf{X}) \leq 0, \qquad j = 1 \ldots . m \tag{3.62}$$

biçimindedir.

Eşitsizlik kısıtı (3.62'de) eşitlik kısıtına negatif olmayan serbest değişken vasıtasıyla "$y_j{}^2$" dönüştürülebilir.

$$g_j(\mathbf{X}) + y_j{}^2 = 0, \qquad j = 1 \ldots . m \tag{3.63}$$

Bu haliyle problem şu şekli alır:

$$\min f(\mathbf{X}) , kısıtına\ göre\ \ G_j(\mathbf{X}, \mathbf{Y}) = g_j(\mathbf{X}) + y_j{}^2 = 0 \ \ j = 1,2,..m$$

$$\mathbf{Y} = \{y_1, y_2, \ldots y_m\}^T \ serbest\ değişken\ vektörü \tag{3.63}$$

Bu problem Lagrange çarpanları metoduyla rahatlıkla çözülebilir. Bunun için Lagrange fonksiyonunu yazacak olursak

$$L(\mathbf{X}, \mathbf{Y}, \boldsymbol{\lambda}) = f(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j G_j(\mathbf{X}, \mathbf{Y})$$
$$\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \ldots \lambda_m\}^T \ lagrange\ çarpanları\ vektörü \tag{3.64}$$

Lagrange fonksiyonunun durağan noktası aşağıdaki denklemlerin çözümüyle bulunabilir.

Gerek Şartlar

$$\frac{\delta L}{\delta x_i}(\mathbf{X},\mathbf{Y},\boldsymbol{\lambda}) = \frac{\delta f}{\delta x_i}(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j \frac{\delta g_j}{\delta x_i}(\mathbf{X}) = 0, \qquad i = 1,2 \ldots . n \tag{3.65}$$

$$\frac{\delta L}{\delta \lambda_j}(\mathbf{X},\mathbf{Y},\boldsymbol{\lambda}) = G_j(\mathbf{X},\mathbf{Y}) = g_j(\mathbf{X}) + {y_j}^2 = 0, \qquad j = 1,2 \ldots . m \tag{3.66}$$

$$\frac{\delta L}{\delta y_j}(\mathbf{X},\mathbf{Y},\boldsymbol{\lambda}) = 2\lambda_j y_j = 0, \qquad j = 1,2 \ldots . m \tag{3.67}$$

Eşitlik (3.65) ten (3.67) ye kadar (n+2m) bilinmeyenli $\mathbf{X},\mathbf{Y},\boldsymbol{\lambda}$ (n+2m) denklem verilmiştir. Bu denklemlerin çözümü optimal çözüm vektörü $\mathbf{X}^*$, lagrange çarpanları vektörü $\boldsymbol{\lambda}^*$ ve serbest değişken vektörü $\mathbf{Y}^*$ ı verir.

Eşitlik (3.66) $g_j(\mathbf{X}) \leq 0,\ j = 1,2,\ldots m$ kısıtını sağlarken, eşitlik (3.67) $\lambda_j = 0\ veya\ y_j = 0$ denklemlerini içerir. Eğer $\lambda_j = 0$ ise j'ninci kısıt aktif değildir ve ihmal edilebilir anlamına gelir. Diğer taraftan eğer $y_j = 0$ ise optimal noktada kısıt $(g_j = 0)$ aktiftir anlamına gelir. Kısıstların iki alt kümeye bölündüğünü düşünürsek $J_1\ ve\ J_2$ olarak $J_1 + J_2$ toplam kısıt kümesini ifade eder. J_1 optimal noktada aktif olan kısıtların indisinin göstergesi olsun; J_2 de aktif olmayan kısıtların.

Böylece $j \in J_1$, $y_j = 0$ (kısıtlar aktif), $j \in J_2$, $\lambda_j = 0$ (kısıtlar aktif değil) ve eşitlik (3.65) şu şekilde sadeleştirilebilir.

$$\frac{\delta f}{\delta x_i} + \sum_{j \in J_1} \lambda_j \frac{\delta g_j}{\delta x_i}(\mathbf{X}) = 0, \qquad i = 1,2 \ldots . n \tag{3.68}$$

Benzer şekilde eşitlik (3.66) şu şekilde yazılabilir:

$$g_j(\mathbf{X}) = 0, \qquad j \in J_1 \tag{3.69}$$

$$g_j(\mathbf{X}) + {y_j}^2 = 0, \qquad j \in J_2 \tag{3.70}$$

Eşitlik (3.68) den (3.70) e kadar $n + m$ bilinmeyenli $x_i(i = 1,2,\ldots n), \lambda_j(j \in J_1)\ ve\ y_j(j \in J_2)$ aktif kısıt sayısını ifade eden p den oluşan $n + p + (m - p) = n + m$

denklemler verilmiştir. İlk p kısıtın aktif olduğunu farzederek eşitlik (3.68) aşağıdaki gibi ifade edilebilir.

$$-\frac{\delta f}{\delta x_i} = \lambda_1 \frac{\delta g_1}{\delta x_i} + \lambda_2 \frac{\delta g_2}{\delta x_i} + \cdots + \lambda_p \frac{\delta g_p}{\delta x_i} = 0, \qquad i = 1,2 \ldots . n \tag{3.71}$$

Bu denklemler toplu şekilde yazılırsa:

$$-\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 + \cdots + \lambda_p \nabla g_p = 0, \qquad i = 1,2 \ldots . n \tag{3.72}$$

∇f ve ∇g hedef fonksiyonun gradyenti ve j. kısıt olarak sırasıyla:

$$\nabla f = \begin{Bmatrix} \frac{\delta f}{\delta x_1} \\ \frac{\delta f}{\delta x_2} \\ \vdots \\ \frac{\delta f}{\delta x_n} \end{Bmatrix} \; ve \; \nabla g = \begin{Bmatrix} \frac{\delta g_j}{\delta x_1} \\ \frac{\delta g_j}{\delta x_2} \\ \vdots \\ \frac{\delta g_j}{\delta x_n} \end{Bmatrix}$$

Eşitlik (3.72) hedef fonksiyonun gradyentinin negatifinin optimal noktada aktif kısıtların gradyentinin lineer kombinasyonu şeklinde yazılabileceğini gösterir. İlaveten minimizasyon problemi durumunda λ_j değerleri $(j \in J_1)$ pozitif olmak zorunda olduğunu gösterebiliriz. Basitçe açıklamak adına sadece iki kısıtın (p=2) optimal noktada aktif olduğunu farzedelim. Eşitlik (3.72) şu şekli alacaktır:

$$-\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 \tag{3.73}$$

S optimal noktada fizibıl yön olsun. Eşitliğin iki tarafını $\mathbf{S}^T$ ile çarparsak aşağıdaki eşitlik elde edilir.

$$-\mathbf{S}^T \nabla f = \lambda_1 \mathbf{S}^T \nabla g_1 + \lambda_2 \mathbf{S}^T \nabla g_2 \tag{3.74}$$

"T" transpozu ifade etmektedir. **S** fizıbıl yön olduğu için aşağıdaki ilişkileri sağlamalıdır.

$$\mathbf{S}^T \nabla g_1 < 0$$

$$\mathbf{S}^T \nabla g_2 < 0 \tag{3.75}$$

Böylelikle eğer $\lambda_1 > 0$ ve $\lambda_2 > 0$ ise $\mathbf{S}^T \nabla f$ niceliği her zaman pozitif olarak görülebilir. ∇f gradyent yönünü yani maksimum oranda fonksiyon değerinin artış yönünü gösterir. $\mathbf{S}^T \nabla f$ ise f in artışının **S** yönü boyunca bileşkesini ifade eder. Eğer $\mathbf{S}^T \nabla f > 0$ ise **S** yönü boyunca ilerledikçe fonksiyon değeri artar. Dolayısıyla eğer λ_1 ve λ_2 pozitif ise fizıbıl domain boyunca fonksiyon değerinin düşeceği hiç bir yön bulamayacağız. Çünkü (3.75) eşitliğini sağlayan nokta optimal olduğu farz edilir ve λ_1 ve λ_2 pozitif olmak zorundadır. Bu mantık iki kısıttan daha fazlasının aktif olduğu duruma da genişletilebilir. Benzer şekilde λ_j değerleri maksimizasyon problemleri için negatif olmak zorundadır.

Kuhn-Tucker Koşulları

Aşağıda gösterildiği gibi kısıtlı minimum noktada $\mathbf{X}^*$ sağlanan koşullar, açıklanabilir

$$\frac{\delta f}{\delta x_i} + \sum_{j \in J_1} \lambda_j \frac{\delta g_j}{\delta x_i}(\mathbf{X}) = 0, \qquad i = 1,2 \ldots . n \tag{3.76}$$

$$\lambda_j = 0, \ j \in J_1 \tag{3.77}$$

Bu koşullar $f(\mathbf{X})$ in lokal minumumda sağlanması gerekli koşullar *Kuhn-Tucker* koşulları olarak matematikçiler tarafından türetilmiştir. Bu koşullar genel olarak yerel minumum için yeterli değildir. Bununla birlikte *Kuhn-Tucker* koşulları *konveks programlama problemleri* olarak isimlendirilen bir diğer problem sınıfının global minumumu için gerekli ve yeter koşullardır. Eğer aktif kısıt kümesi bilinmiyorsa *Kuhn-Tucker* koşulları aşağıdaki gibi olacaktır:

$$\frac{\delta f}{\delta x_i} + \sum_{j=1}^{m} \lambda_j \frac{\delta g_j}{\delta x_i}(\mathbf{X}) = 0, \qquad i = 1,2 \ldots . n$$

$$\lambda_j g_j = 0, \quad j = 1,2, \ldots m \tag{3.78}$$

$$g_j \leq 0, \qquad j = 1,2, \ldots m$$

$$\lambda_j \geq 0, \qquad j = 1,2, \ldots m$$

Eğer problem maksimizasyon problemi veya kısıtlar $g_j \geq 0$, şeklinde ise eşitlik (3.78) deki λ_j nin değeri pozitif olmamak zorundadır. Diğer bir yandan eğer problem $g_j \geq 0$ formunda kısıtlı maksimizasyon problemi ise λ_j negatif olmamak zorundadır.

3.3.2 Ardışık Kuadratik Programlama Metodu (SQP)

SQP metodu yakın zamanda geliştirilmiş belki de en iyi optimizasyon metodu olabilir. Lagrange tabanlı bir yöntemdir [46].

Metodun teorik temeli şöyledir;

1) Newton'un metodunu kullanarak doğrusal olmayan denklemler kümesinin çözümü.
2) Eş zamanlı olarak doğrusal olmayan denklemlerin Kuhn-Tucker koşullarının kısıtlı optimizasyon probleminin Lagrangian ile türetilmesi

Denklemlerin türetilmesi

Sadece eşitlik kısıtlı doğrusal olmayan bir optimizasyon problemi düşünülürse :

f(X)'i minimize edecek X değerini , $h_k(X) = 0$, k = 1,2,…p koşuluna göre bulmak için ,

Eşitlik kısıtlı doğrusal olmayan bir optimizasyon probleminin Lagrange fonksiyonu L (**X**,λ);

$$L = f(\mathbf{X}) + \sum_{k=1}^{p}(\lambda_k h_k(\mathbf{X})) \qquad (3.79)$$

λ_k k. Eşitlik kısıtı için Lagrange çarpanı olarak isimlendirilir.

Kuhn-Tucker gereklilik koşulu:

$$\nabla L = 0 \text{ veya } \nabla f + \sum_{k=1}^{p} \lambda_k \nabla h_k = 0 \text{ veya } \nabla f + [A]^T \lambda = 0 \qquad (3.80)$$

$$h_k(\mathbf{X}) = 0 \text{ , k} = 1,2, \dots \dots \text{p} \qquad (3.81)$$

[A], n x p boyutunda bir matristir. k. sütun h_k fonksiyonunun gradyentini simgeler. Yukarıdaki denklemler (3.80) ve (3.81) içinde n+p adet bilinmeyen olan n+p adet doğrusal olmayan denklemleri gösterir. (X_i, i=1,...,n and λ_k, k=1,...,p). Doğrusal olmayan

bu denklemler Newton metodu kullanılarak çözülebilir. Kolaylık için, yukarıdaki denklemleri şu şekilde yazabilir.

$$\mathbf{F}(\mathbf{Y}) = 0 \tag{3.82}$$

$$\mathbf{F} = \begin{pmatrix} \nabla \mathrm{L} \\ h \end{pmatrix}_{(n+p)x1}, \mathbf{Y} = \begin{pmatrix} \mathrm{X} \\ \lambda \end{pmatrix}_{(n+p)x1}, \mathbf{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}_{(n+p)x1} \tag{3.83}$$

Newton metoduna göre, yukarıdaki denklemlerin çözümü:

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \Delta \mathbf{Y}_j \tag{3.84}$$

$$[\nabla F]_j^T \Delta \mathbf{Y}_j = -\mathbf{F}(\mathbf{Y}_j) \tag{3.85}$$

Y_j j.'ninci iterasyon başlangıcındaki çözüm ve ΔY_j , Y_j deki değişim, geliştirilmiş çözüm Y_{j+1} elde etmek için gerekli ve $[\nabla F]_j = [\nabla F(Y_j)]j$ (n+p) x (n+p) doğrusal olmayan denklemin Jacobian matrisi i. sütunu Y vektörüne ait fonksiyonun F_i (Y) gradyentini simgeler.(3.85) no'lu eşitlikte (3.82) ve (3.83) 'teki eşitlikler yerlerine konursa (3.86) no'lu eşitlikler elde edilir.

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H]^T & [0] \end{bmatrix}_j \begin{pmatrix} \nabla \mathrm{X} \\ \nabla \lambda \end{pmatrix}_j = -\begin{pmatrix} \nabla \mathrm{L} \\ h \end{pmatrix}_j \tag{3.86}$$

$$\Delta \mathbf{X}_j = \mathbf{X}_{j+1} - \mathbf{X}_j \tag{3.87}$$

$$\Delta \lambda_j = \lambda_{j+1} - \lambda_j \tag{3.88}$$

Eşitlik (2.8)'deki matrisin elemanı $[\nabla^2 L]_{nxn}$ Lagrange fonksiyonunun Hessian matrisi olarak isimlendirilir. (3.86) eşitliğindeki ilk denklem takımı (3.89) deki gibi ayrı yazılabilir.

$$[\nabla^2 L]_j \Delta \mathbf{X}_j + [H]_j \Delta \lambda_j = -\nabla L_j \tag{3.89}$$

Eşitlik (3.89)'de , $\Delta\lambda_j$ yerine (3.88) eşitliği ve ∇L_j yerine (3.80) eşitliği kullanılarak (3.90) deki gibi yazılabilir.

$$\left[\nabla^2 L\right]_j \Delta \mathbf{X}_j + [H]_j (\lambda_{j+1} - \lambda_j) = -\nabla f_j - [H]_j^T \lambda_j \tag{3.90}$$

ve eşitlik sadeleştirilirse (3.91)'teki gibi eşitlik elde edilir.

$$[\nabla^2 L]_j \Delta \mathbf{X}_j + [H]_j \lambda_{j+1} = -\nabla f_j \quad (3.91)$$

(3.91) ile (3.86) birleştirilirse (3.92)'deki eşitlik elde edilir.

$$\begin{bmatrix} [\nabla^2 L] & [H] \\ [H]^T & [0] \end{bmatrix}_j \begin{pmatrix} \nabla \mathrm{X}_j \\ \lambda_{j+1} \end{pmatrix} = -\begin{pmatrix} \nabla \mathrm{f}_j \\ h_j \end{pmatrix} \quad (3.92)$$

Yukarıdaki eşitlik (3.92), tasarım vektörü ΔX_j teki değişimi ve Lagrange çarpanlarının yeni değerlerini λ_{j+1} 'i bulmak için çözülebilir .Yukarıdaki eşitlikle (3.92) gösterilmiş iteratif proses yakınsama gerçekleşene kadar devam ettirilir. Şimdi (3.93) ve (3.94) daki eşitlikleri dikkate alarak hangi ΔX kuadratik hedef fonksiyonunu minimize eder bu problemin çözümü araştırılırsa:

$$Q = \nabla f^T \Delta \mathbf{X} + \frac{1}{2} \Delta \mathbf{X}^T [\nabla^2 L] \Delta \mathbf{X} \quad (3.93)$$

aşağıdaki lineer kısıt denklemlerine göre

$$h_k + \nabla h_k^T \Delta \mathbf{X} = 0 \text{ , k} = 1,2, \ldots \ldots \text{p veya h} + [H]^T \Delta \mathbf{X} = 0 \quad (3.94)$$

Yukarıdaki problemin Lagrange fonksiyonu

$$\tilde{L} = \nabla f^T \Delta \mathbf{X} + \frac{1}{2} \Delta \mathbf{X}^T [\nabla^2 L] \Delta \mathbf{X} + \sum_{k=1}^{p} \lambda_k (h_k + \nabla h_k^T \Delta \mathbf{X}) \quad (3.95)$$

λ_k k. Eşitlik kısıtı ile ilgili Lagrange çarpanıdır.

Kuhn-Tucker gereklilik koşulu şu hali alır :

$$\nabla f + [\nabla^2 L] \Delta \mathbf{X} + [H] \lambda = 0 \quad (3.96)$$

$$h_k + \nabla h_k^T \Delta \mathbf{X} = 0 \text{ , k} = 1,2, \ldots \ldots \text{p} \quad (3.97)$$

Yukarıdaki eşitlikler (3.96) ve (3.97) matris formunda aynı eşitlik (3.92)'te olduğu tanımlanabilir. Böylece orijinal problemi f(**X**)'i minimize edecek X değerini , h_k(**X**) = 0 , k = 1,2,…p koşuluna göre bulmak için , aşağıdaki denklem ile tanımlanan kuadratik programlama problemi iteratif olarak çözülebilir.

$$Q = \nabla f^T \Delta \mathbf{X} + \frac{1}{2} \Delta \mathbf{X}^T [\nabla^2 L] \Delta \mathbf{X} \qquad (3.98)$$

Orijinal probleme eşitsizlik kısıtları da eklendiği zaman yukarıdaki kuadratik programlama problemi denklemi şu hali alır:

$$g_j + \nabla g_j^T \Delta \mathbf{X} \leq 0\,, \quad j = 1,2,\ldots,m$$
$$h_k + \nabla h_k^T \Delta \mathbf{X} \leq 0\,, \quad k = 1,2,\ldots,p$$

kısıtlarına göre

$$Q = \nabla f^T \Delta \mathbf{X} + \tfrac{1}{2} \Delta \mathbf{X}^T [\nabla^2 L] \Delta \mathbf{X}$$

denklemini minimize eden **X** değerini bulun (3.99)

Lagrange fonksiyonu ile

$$\tilde{L} = f(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j \, g_j(\mathbf{X}) \sum_{k=1}^{p} \lambda_{m+k} + h_k(\mathbf{X}) \qquad (3.100)$$

Augmented Lagrange fonksiyonunun minumumunu içerdiğinden, ardışık kuadratik programlama metodu tahmini Lagrangian metodu olarakta bilinir.

Çözüm Prosedürü

Kısıtsız minimizasyondaki Newton metodun daki gibi, (3.99) eşitliğindeki çözüm vektörü ΔX, S araştırma yönündeki gibi işlem görerek, ve kuadratik programlama alt problemi (S tasarım vektörü şeklinde) yeniden ifade edilirse

$$\beta_j g_j(\mathbf{X}) + \nabla g_j(\mathbf{X})^T \mathbf{S} \leq 0\,, \quad j = 1,2,\ldots,m$$
$$\bar{\beta} h_k(\mathbf{X}) + \nabla h_k(\mathbf{X})^T \mathbf{S} = 0\,, \quad k = 1,2,\ldots,p$$

kısıtlarına göre

$$Q(\mathbf{S}) = \nabla f(\mathbf{X})^T \mathbf{S} + \tfrac{1}{2} \mathbf{S}^T [H] \mathbf{S}$$

denklemini minimize eden **S** değerini bulun (3.101)

[H] başlangıçta pozitif belirli birim matris olarak alınan ve sonraki iterasyonlarla denklemdeki Lagrange fonksiyonunun Hessian matrisine yakınsamak amacıyla güncellenen bir matristir. Eşitlik (3.101)'in Lagrange fonksiyonu ve β_j ve $\overline{\beta}$ uygun uzayı tamamen ayırmayan doğrusallaştırılmış kısıtları sağlamak için kullanılan sabitlerdir. Bu sabitlerin değerleri genel olarak

$$\bar{\beta} \approx 0.9\,;\ \beta_j = \begin{cases} 1 & \text{eğer } g_j(\mathbf{X}) \le 0 \\ \bar{\beta} & \text{eğer } g_j(\mathbf{X}) \ge 0 \end{cases} \tag{3.102}$$

Eşitlik (3.101)'in alt problemi bir kuadratik programlama problemidir ve bu yüzden kuadratik fonksiyonları minimize etmek için kullanılan metodlar çözüm için kullanılabilir. Alternatif olarak fonksiyonların gradyentini içeren diğer metodlar ile de kolaylıkla hesaplanabilir. Yukarıdaki problemin (3.101) çözümüyle ilişkili olduğu için Lagrange çarpanları şöyle hesaplanabilir :

$$\mathbf{G}_{nxp}\,\boldsymbol{\lambda}_{px1} = \mathbf{F}_{nx1} \tag{3.103}$$

$$\lambda = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{F} \tag{3.104}$$

$$G = \begin{bmatrix} \frac{\partial g_{j1}}{\partial x_1} & \cdots & \frac{\partial g_{jp}}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{j1}}{\partial x_n} & \cdots & \frac{\partial g_{jp}}{\partial x_n} \end{bmatrix}_{\mathrm{x}} \quad ,\lambda = \begin{Bmatrix} \lambda_{j1} \\ \lambda_{j2} \\ . \\ . \\ \lambda_{jp} \end{Bmatrix} \text{ ve } \mathrm{F} = \begin{Bmatrix} -\frac{\partial f}{\partial x_1} \\ -\frac{\partial f}{\partial x_2} \\ . \\ . \\ -\frac{\partial f}{\partial x_n} \end{Bmatrix}_{\mathrm{x}} \tag{3.105}$$

Araştırma yönü, **S**, (3.101) eşitliğindeki problemin çözümüyle bulunur. Tasarım vektörü güncellenir :

$$\mathbf{X}_{j+1} = \mathbf{X}_j + \alpha^*\mathbf{S} \tag{3.106}$$

α* , **S** araştırma yönü boyunca optimal adım uzunluğudur. Fonksiyon minimize edilerek bulunur (exterior penalty fonksiyonu yaklaşımı kullanılarak) :

$$= f(\mathbf{X}) + \sum_{j=1}^{m} \lambda_j \,(\max[0, g_j(\mathbf{X})]) \sum_{k=1}^{p} \lambda_{m+k} |h_k(\mathbf{X}|) \tag{3.107}$$

$$= \begin{cases} |\lambda_j|, & j = 1,2, \dots m+p \text{ ilk iterasyonda} \\ \max\left\{|\lambda_j|, \frac{1}{2}(\widetilde{\lambda_j}, |\lambda_j|)\right\} & \text{sonraki iterasyonlarda} \end{cases} \tag{3.108}$$

$$= \lambda_j \text{ önce ki iterasyondan} \tag{3.109}$$

Tek değişkenli adım uzunluğu α* herhangi bir tek değişkenli optimizasyon metodlarıyla bulunabilir. $\mathbf{X}_{j+1}$ (3.106) eşitliğinden bulunduğu zaman, gelecek iterasyonda kuadratik yakınsamayı geliştirmek için Hessian matrisi [H] eşitlik (3.101)'de güncellenir. Sonuç olarak modifiye edilmiş BFGS formülü aşağıda verildiği gibidir.

$$[\mathrm{H_{i+1}}] = [\mathrm{H_i}] - \frac{[\mathrm{H_i}]\mathrm{P_i}\mathrm{P_i}^{\mathrm{T}}[\mathrm{H_i}]}{\mathrm{P_i}^{\mathrm{T}}[\mathrm{H_i}]\mathrm{P_i}} + \frac{\gamma\gamma^{\mathrm{T}}}{\mathrm{P_i}^{\mathrm{T}}\mathrm{P_i}} \tag{3.111}$$

$$\mathbf{P}_i = \mathbf{X}_{i+1} - \mathbf{X}_i \tag{3.112}$$

$$\gamma = \theta\mathbf{Q}_i + (1 - \theta)[H_i]\mathbf{P}_i \tag{3.113}$$

$$\mathbf{Q}_i = \nabla_x \tilde{L}(\mathbf{X}_{i+1}, \lambda_{i+1}) - \nabla_x \tilde{L}(\mathbf{X}_i, \lambda_i) \tag{3.114}$$

$$\theta = \begin{cases} 1.0 \text{ eğer } \mathbf{P}_i^T\mathbf{Q}_i \geq 0.2\mathbf{P}_i^T[H_i]\mathbf{P}_i \\ \dfrac{0.8\mathbf{P}_i^T[H_i]\mathbf{P}_i}{\mathbf{P}_i^T[H_i]\mathbf{P}_i - \mathbf{P}_i^T\mathbf{Q}_i} \text{ eğer } \mathbf{P}_i^T\mathbf{Q}_i < 0.2\mathbf{P}_i^T[H_i]\mathbf{P}_i \end{cases} \tag{3.115}$$

Eşitlik (2.36)'teki 0.2 ve 0.8 sabitleri nümerik tecrübelere dayanılarak değiştirilebilir.

3.3.3 İç Noktalar Yöntemi (Bariyer Fonksiyonu Metodu)

Bariyer fonksiyonlar kısıtlı bir problemi kısıtsız bir probleme veya bir problem dizisine dönüştürmek için kullanılır. Bu fonksiyonlar, uygun bölgeden ayrılmaya karşı bir engel oluşturur. Eğer optimal çözüm uygulanabilir bölgenin sınırında meydana gelirse, prosedür iç kısımdan sınıra geçer.

Optimizasyon problemimizi tanımlayacak olursak, $f(x)$ fonksiyonunu $g(x) \leq 0$ kısıtına göre minimize etmek için $x \in X$, $g(x)$ bileşenleri $g_1(x), \ldots, g_m(x)$ olan bir vektör fonksiyonu iken f, $g_1(x), \ldots, g_m(x)$ fonksiyonları R^n 'de tanımlı, sürekli ve X, R^n 'de boş küme değildir.

Varsa eşitlik kısıtları X kümesine dahil edilmiştir. Alternatif olarak, afin eşitlik kısıtlamaları söz konusu olduğunda, diğerlerine göre bazı değişkenleri çözdükten sonra bunları kolayca eleyerek, problemin boyutunu azaltabiliriz. Bu davranışın gerekliliğinin sebebi, bariyer fonksiyonları yöntemi $\{x: g(x) < 0\}$ kümesinin boş olmamasını

gerektirir. Eğer eşitlik kısıtı $h(x) = 0$, eşitsizlik kısıtı $h(x) \leq 0$ ve $h(x) \geq 0$ şeklinde ifade edilseydi, bariyer fonksiyonu yöntemi mümkün olmayacaktı.

Bariyer problemi; Inf $\theta(\mu)$, $\mu > 0$ için $\theta(\mu) =$ inf $\{f(x) + \mu B(x): g(x) < 0, x\epsilon X\}$. şeklinde ifade edilir. Burada B, $\{x: g(x) < 0\}$ bölgesi üzerinde sürekli ve negatif olmayan bariyer fonksiyonudur ve bölgenin sınırına $\{x: g(x) < 0\}$ içten yaklaştırıldığı için fonksiyon ∞'a yaklaşır.

Bariyer fonksiyonları tanımlayacak olursak tipik bariyer fonksiyonu;

$$B(x) = \sum_{i=1}^{m} -1/g_i(x) \tag{3.116}$$

Yardımcı fonksiyon olarak $\mathrm{f(x)} + \mu\mathrm{B(x)}$ tanımlarsak, idealinde B fonksiyonunun $\{\mathrm{x}: g(\mathrm{x}) < 0\}$ bölgesinde sıfır ve sınırında ∞ değerini almasını arzu ederiz. Bu bizim $\{\mathrm{x}: g(\mathrm{x}) \leq 0\}$ bölgesinden ayrılmayacağımızı garanti altına alarak, minimizasyon probleminin iç noktadan başlamasını sağlar. Ancak bu süreksizlik herhangi bir hesaplama prosedürü için ciddi zorluklar çıkmasına sebep olur. Bu nedenle, B'nin bu ideal yapısı, B'nin $\{\mathrm{x}: \mathrm{g(x)} < 0\}$ bölgesi üzerinde pozitif ve sürekli olmasının ve sınırın iç kısımdan yaklaştığı zaman sonsuzluğa yaklaşmasının daha gerçekçi bir gerekliliği ile yer değiştirilir. Yukarıdaki fonksiyonda μ sıfıra yaklaşırken μB ideal bariyer fonksiyonuna yaklaşır. $\mu > 0$ iken $\theta(\mu) =$ inf $\{\mathrm{f(x)} + \mu\mathrm{B(x)}: g(\mathrm{x}) < 0, \mathrm{x}\epsilon\mathrm{X}\}$ hesaplarken $g(\mathrm{x}) < 0$ kısıtının varlığından dolayı orijinal problemi çözmekten daha kolay değildir. Ancak B'nin yapısının sonucu olarak optimizasyona $\mathrm{S} = \{\mathrm{x}: g(\mathrm{x}) < 0\} \cap \mathrm{X}$ bölgesindeki bir noktadan başlarsak ve $g(\mathrm{x}) < 0$ kısıtına aldırmazsak S bölgesinde bir optimal noktaya ulaşırız. $\{x: g(x) < 0\}$ sınırına S içinden yaklaşırken, B sonsuzluğa yaklaşır ve bu da S kümesinden ayrılmamızı önleyecektir.

$G(x_k) < 0$ ise ve bariyer fonksiyonu $G = \{\mathrm{x}: g(\mathrm{x}) < 0\}$ bölgesi sınırına kadar sonsuzluğa yaklaştığından, $g(\mathrm{x}) < 0$ sınırlaması göz ardı edilebilir. Elde edilen optimal nokta $x_{k+1} \in G$'nin elde edilmesini sağlayacak kısıtlanmamış bir optimizasyon tekniği kullanılmalıdır. Ancak, çoğunlukla araştırma yöntemi olarak ayrık basamaklar kullanır. Eğer sınıra yakınsak, bir adım, bariyer fonksiyonunun B değerinin büyük bir negatif sayı olduğu, uygulanabilir bölgenin dışında bir noktaya yol açabilir. Bu nedenle, problem

eğer uygulanabilirliği açıkça kontrol edilirse kısıtsız bir optimizasyon problemi olarak ele alınabilir. [47]

Uygulamada optimizasyon için oluşturduğumuz algoritmayla birlikte iç noktalar yöntemiyle matlab optimization toolbox yardımıyla optimizasyon hesaplamaları gerçekleştirilmiştir.

BÖLÜM 4

SONUÇ VE ÖNERİLER

4.1 Optimizasyon Sonuçları

İlk olarak 3x3 SP mekanizmasında hareketsiz platformun çapı sabit tutularak, hareketli platformun çapı SQP yöntemi aracılığı ile optimize edilip sonuçlar analiz edilmiştir. Sonrasında hareketli ve sabit platform çapları daha yüksek beceriklilik kabiliyetiyle ve daha çok erişilebilir nokta içeren çalışma uzayı için optimize edilmiştir. Seçilmiş aralıklarda çeşitli hareketli platform çap değerleri için

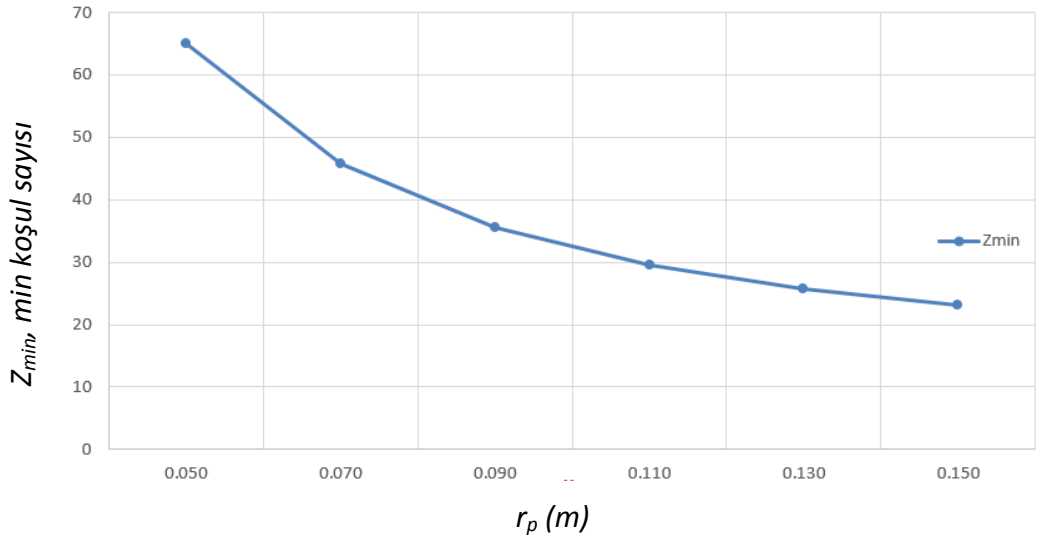
1) koşul sayıları değişimi,

2) sabit/hareketli platform çap değişimine göre minimum koşul sayısı,

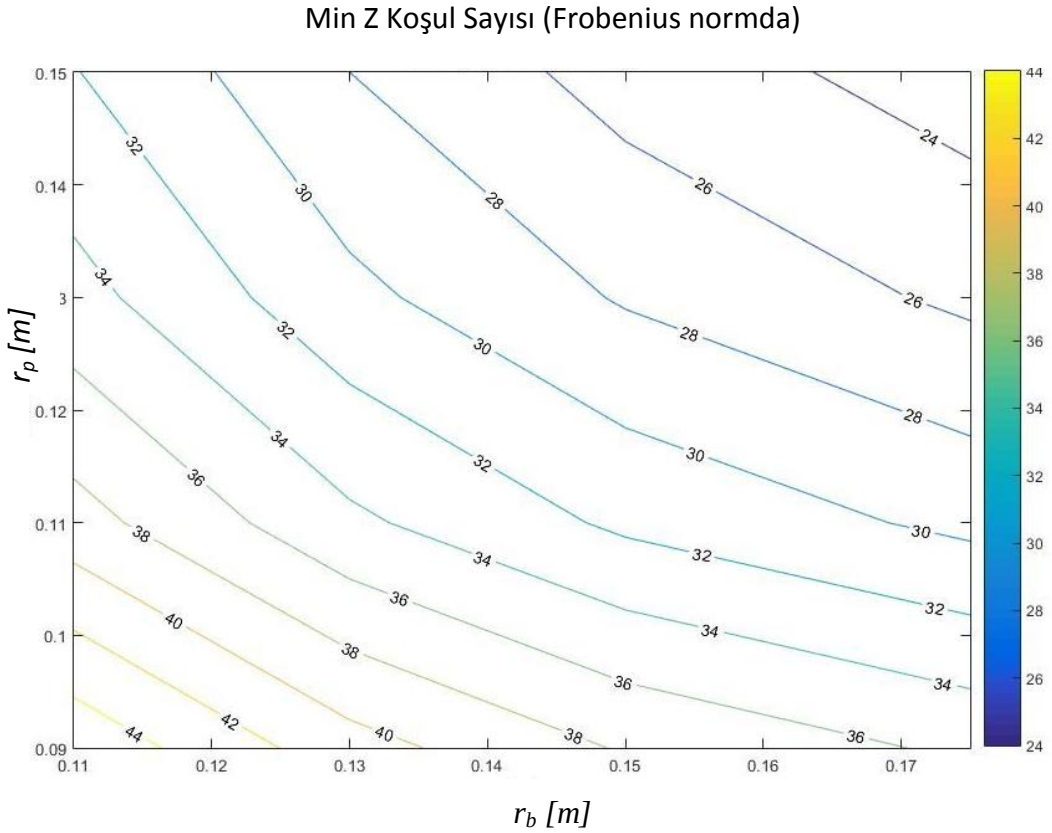
3) farklı sabit/hareketli platform çap kombinasyonlarına göre erişelebilir nokta sayısı

tartışılmıştır.

Şekil 4.1'de, yatay eksen farklı r_p değerlerini ifade etmekte, ve dikey eksen homojenleştirilmemiş Jakobyen matrisinden hesaplamalarla elde edilmiş minimum koşul sayısını ifade etmektedir. Hareketli platform yarıçapı r_p artarken düşük Z_{min} değerleri elde edilir. Düşük koşul sayılı erişilebilinen en efektif çalışma uzayı hacmi için optimal değer belirlenmeye çalışılmıştır. *Sonuç olarak, hareketli platformun çapı arttıkça minimum koşul sayısına götürdüğü gözlemlenmiştir*.



Şekil 4.1 Farklı r_p değerleri için Z_{min} (min koşul sayısı) varyasyonları r_b = *0.175[m].*



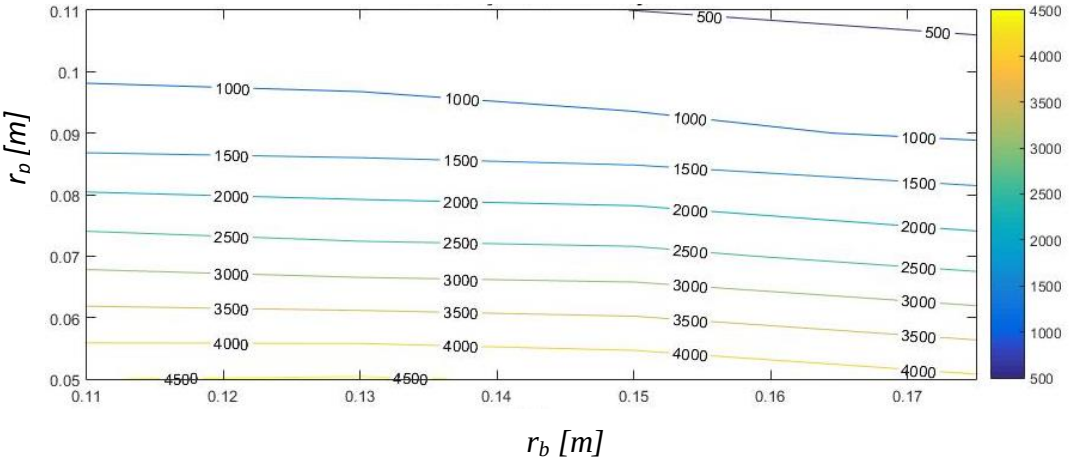
Şekil 4.2 Önceden tanımlanmış erişilebilir çalışma uzayındaki r_p ve r_b 'ye göre Z_{min}

Şekil 4.2'de, yatay eksen farklı r_b değerlerini, sol dikey eksen farklı r_p değerlerini ve sağ dikey eksen ise tanımlanmış erişilebilir çalışma uzayı için Frobenius normda hesaplanan koşul sayılarının değerlerinin kartografyasını ifade etmektedir. Şekil 4.2'de görüldüğü

gibi, *minimum koşul sayıları r_b ve r_p nin yüksek değerlerinde görülmüştür; bunun anlamı hareketli/sabit platform çapları arttıkça beceriklilik iyileşmektedir*.

Bununla birlikte, *minimum koşul sayıları elde etmek için sabit platform çapına göre hareketli platform çapında karşılaştırılabilir bir azalma olması gerektiği çizimden anlaşılmaktadır. Ayrıca sabit platformun çapı hareketli platform çapından çok daha yüksek olduğu zaman bu azalma oranı gittikçe düşmektedir*

Z < 1000 Çalışma uzayı içindeki erişilebilen nokta miktarı



Şekil 4.3 Erişilebilir nokta sayısı (Koşul sayısı < 1000)

Şekil 4.3'de, yatay eksen r_b değerlerini, sol dikey eksen farklı r_p değerlerini ve sağ dikey eksen ise çalışma uzayı içinde koşul sayısı 1000'in altında olan erişilebilen noktaların sayısının kartografyasını ifade etmektedir. Şekil 4.3'te gösterildiği üzere, r_p değerleri azalırken, r_b değerleri artmaktadır; ve erişilebilen nokta sayısı yükselmektedir. Z koşul sayısının üst limiti 1000 olarak kabul edilmiştir. Böylece, *düşük çaplı hareketli platform ve yüksek çaplı sabit platformlarda erişilebilen nokta sayısının yüksek olduğu sonucuna varılmıştır.*

Çizelge 4.1 Farklı kısıt aralıklarındaki hareketli platform optimal yarıçap değerleri

$0.050<r_p<0.180$ *$r_{p,opt}=0.075$*	*$0.070<r_p<0.180$* *r_p opt=0.075*	*$0.080<r_p<0.180$* *r_p opt=0.088*
$0.090<r_p<0.180$ *r_p opt=0.090*	*$0.100<r_p<0.180$* *r_p opt=0.100*	*$0.110<r_p<0.180$* *r_p opt=0.110*
$0.120<r_p<0.180$ *r_p opt=0.120*	*$0.090<r_p<0.250$* *r_p opt=0.090*	*$0.050<r_p<0.250$* *r_p opt=0.075*

Hareketli platform çapının optimizasyon süreci sırasında, Çizelge 4.1'de gösterildiği gibi çeşitli kısıt değerleriyle birlikte farklı başlangıç değerleri kullanılmıştır. Sabit platform yarıçapı r_b*=0.175[m]* iken, minimum limit r_p için *0.050 [m]* olarak alındığında ve bacak uzunlukları *0.300[m]* ve *0.450[m]* değerleri arasında olduğu varsayılarak optimal sonuç r_p için araştırılmış ve *0.075 [m]* sonucuna varılmıştır. Hareketli platform yarıçapı r_p için minimum limit olarak *0.080[m],* kabul edildiğinde optimal değeri r_p*=0.088 [m]* olarak bulunmuştur. Daha önce de belirtildiği üzere r_p için farklı başlangıç değerleri ile optimizasyon prosedürü işletilmiştir. Sonuç olarak r_p için önceden belirlediğimiz optimizasyon kısıtları değiştiği zaman, r_p için bulduğumuz optimal değerde değişir. Bunun sebebi*, farklı çalışma uzayı bölgeleri içinde farklı yerel minumumların olmasıdır.* Çizelge 4.1'deki elde edilmiş olan optimizasyon sonuçları analiz edildiğinde, hareketli platformun yarıçapı r_p için belirlenmiş olan en alt limit değerleri *0.050* ve *0.070[m]* iken optimal değer için taranan aralık *0.050-0.180[m] dir.* Hareketli platform yarıçapı r_p için bulunan optimal değer *0.075[m] dir.* Ne zaman ki alt limit değeri r_p *0.080[m]* olarak belirlenmiştir*,* optimal değeri yani yerel minimum değeri r_p *0.075[m]* olarak değişmiştir. Optimal nokta, r_p için, *0.075[m]* den *0.088[m]* değerine kaymıştır. Dokuz farklı optimizasyon kombinasyonundan sonra r_p için iki farklı yerel minimum ile karşılaşılmıştır; ve bu sonuçların değerleri *0.075[m] ile 0.088[m] dir.* Sonrasında, başlangıç değeri *0.090[m]* ve daha yüksek başlangıç değerleri ile optimal nokta bulmak için yapılan taramalarda, araştırma yapılan değer aralığının başlangıç değeri optimal nokta olarak bulunmuştur; *0.090[m], 0.100[m],0.110[m], vb*. Sonuç olarak, r_p için koşul sayısının "Z" üst limitinin 1000 ile sınırlandırıldığı, maksimum sayıda erişebileceğimiz noktayı sağlayan iki yerel minimum değeri bulunmuştur *(0.075[m] ve 0.088 [m])*. İki değişken için (r_b ve r_p) farklı başlangıç değerleri ile gerçekleştirilmiş optimizasyonun maliyet fonksiyonu değerleri ve optimizasyon sonuçları Çizelge 4.2 de gösterildiği gibidir. Örnek olarak, başlangıç değeri olarak sabit platform yarıçapı r_b için 0.125 [m] ve hareketli platform yarıçapı r_p için 0.070 [m] alındığında, maliyet fonksiyonunun değeri 4.8xE7, r_p 'nin optimal değeri 0.074 [m] ve r_b 'nin optimal değeri 0.129 [m] olarak bulunmuştur. Optimal değerlerin r_p ve r_b 'nin kısıtlarına bağlı olduğu açıktır.

Çizelge 4.2'yi oluştururken, belirlenen kısıtlar r_b için 0.125 [m] < r_b < 0.175 [m] ve r_p için, 0.070[m] < r_p <0.125 [m] 'dir. Düşük çaplı hareketli platform ile yüksek çaplı sabit

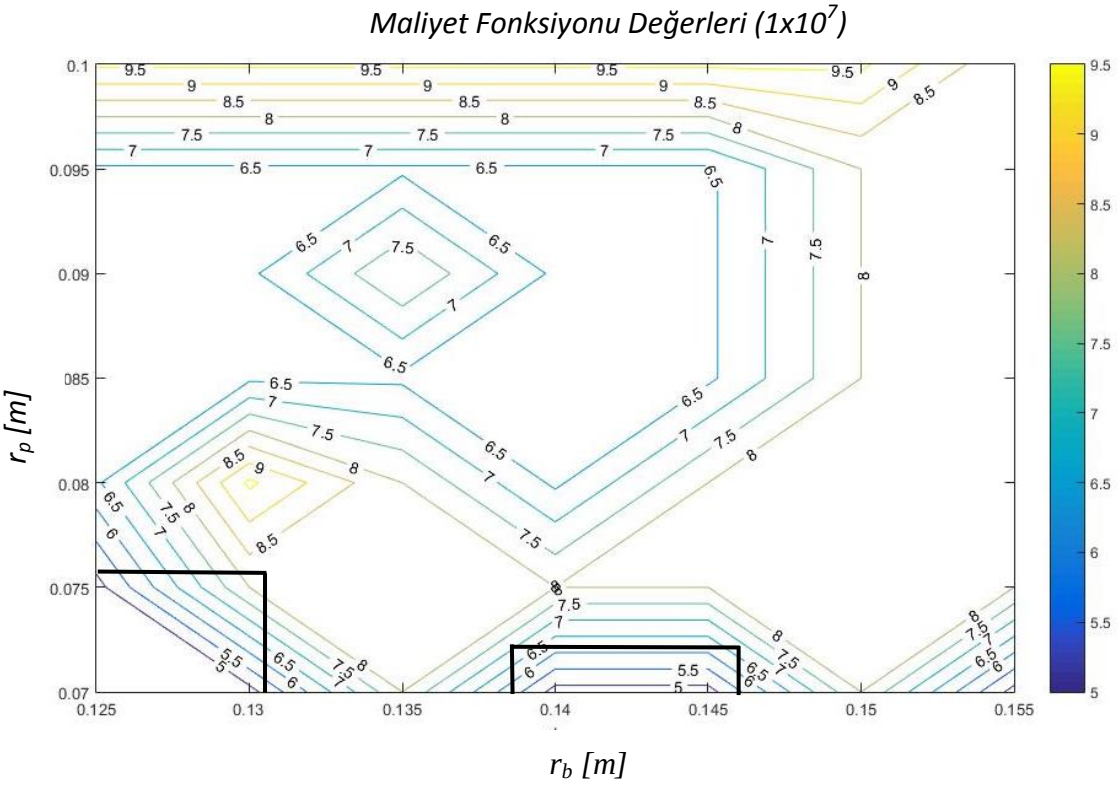
platform nispeten düşük değerli maliyet fonksiyonuna neden olur ve bu da daha iyi hareket kabiliyeti anlamına gelir.

Çizelge 4.2 Ardışık kuadratik programlama (SQP) ile optimize edilen hareketli ve sabit platform yarıçapları, $r_{p,opt}$, $r_{b,opt}$

Maliyet.F(x)		Sabit Platform Yarıçapı, r_b						
$r_{p,opt\ [m]}$ / $r_{b,opt\ [m]}$		0.125	0.130	0.135	0.140	0.145	0.150	0.155
Hareketli Platform Yarıçapı, r_p	0.070	4.8xE7	4.8xE7	8xE7	4.8xE7	4.8xE7	8xE7	4.8xE7
		0.074	0.075	0.082	0.075	0.076	0.090	0.076
		0.129	0.130	0.133	0.130	0.125	0.141	0.125
	0.075	4.8xE7	8xE7	8xE7	8xE7	8xE7	8xE7	8xE7
		0.077	0.075	0.081	0.086	0.087	0.081	0.093
		0.125	0.130	0.134	0.137	0.140	0.135	0.145
	0.080	6.4xE7	9.6xE7	8xE7	6.4xE7	8xE7	8xE7	8xE7
		0.095	0.102	0.084	0.080	0.088	0.093	0.094
		0.125	0.128	0.134	0.125	0.142	0.144	0.146
	0.085	6.4xE7	6.4xE7	6.4xE7	6.4xE7	6.4xE7	8xE7	8xE7
		0.085	0.085	0.085	0.085	0.085	0.093	0.098
		0.125	0.125	0.125	0.125	0.125	0.144	0.148
	0.090	6.4xE7	6.4xE7	8xE7	6.4xE7	6.4xE7	8xE7	6.4xE7
		0.090	0.090	0.084	0.090	0.090	0.096	0.090
		0.125	0.125	0.145	0.125	0.125	0.147	0.125
	0.095	6.4xE7	9.6xE7	6.4xE7	6.4xE7	6.4xE7	8xE7	8xE7
		0.095	0.095	0.095	0.095	0.095	0.097	0.099
		0.125	0.130	0.125	0.125	0.125	0.148	0.150
	0.100	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	8xE7
		0.100	0.100	0.109	0.112	0.100	0.112	0.102
		0.125	0.130	0.132	0.133	0.125	0.138	0.152

*E7= $1x10^7$

Şekil 4.4'te, minimum maliyet fonksiyonu değeri 4.8 x 10^7 olarak dikdörtgen içine alınmış bölgelerde elde edilmiştir. Bu bölgelerde optimize edilmiş r_b ve r_p değerleriyle Çizelge 4.2' de gösterilmiş değerlerle de uyumlu olarak maliyet fonksiyonu değerleri minumumdur.



Şekil 4.4 İki değişkenli (r_p, r_b) optimizasyon sonuçları

Optimal r_p değerleri minimum iken, maliyet fonksiyonu değerleri de minimumdur. Daha önceden bahsedildiği gibi, minimum maliyet fonksiyonu değeri daha iyi hareket kabiliyeti elde etmek için araştırılmıştır. Çizelge 4.2'de, minimum maliyet fonksiyonu değerini elde etmek için sadece birden fazla optimal ve alt optimal sonuçların bulunduğu gösterilmiştir. Örnek olarak $r_{p,opt}$, $r_{b,opt}$ = 0.074 [m], 0.129 [m] iken maliyet fonksiyonunun minimum değeri (4.8xE7) ve aynı değer, $r_{p,opt}$, $r_{b,opt}$ = 0.077 [m], 0.125 [m] iken de minimum maliyet fonksiyonu değeri olarak (4.8xE7) bulunmuştur. Daha iyi doğrulukta bir sonuç elde etmek için, çözüm bölgesindeki çözünürlüğü arttırarak daha detaylı bir araştırma ile Çizelge 4.3 oluşturulmuştur. Sonuçları yorumlayabilmek için maliyet fonksiyonu değerleri kullanılmıştır. Bununla birlikte, çalışma uzayının her tarafında koşul sayısı geniş bir spektruma sahiptir. Bu yüzden, koşul sayılarının ortalamasını farklı kısıtlara sahip farklı tasarım parametreleri için karşılaştırmak

anlamsızdır. Sonuç olarak, maliyet fonksiyonunu hesaplayarak (koşul sayılarının r_b ve r_p ile çarpımlarının toplamı) sonuç değerini dikkate almak optimal tasarım parametrelerini araştırırken daha verimlidir.

Çizelge 4.3 Hedef bölgenin kısıtlarının daraltılarak, ardışık kuadratik programlama yöntemi ile optimize edilen hareketli ve sabit platform yarıçapları, $r_{p,opt}$, $r_{b,opt}$

Maliyet.F(x)	Sabit Platform Yarıçapı, r_b				
$r_{p,opt\ [m]}$ $r_{b,opt\ [m]}$	0.125	0.126	0.127	0.128	0,129
0.060	3.2xE7	3.2xE7	3.2xE7	3.2xE7	3.2xE7
	0.068	0.072	0.071	0.071	0.068
	0.125	0.127	0.127	0.128	0.129
0.065	3.2xE7	3.2xE7	3.2xE7	3.2xE7	3.2xE7
	0.069	0.070	0.070	0.071	0.067
	0.129	0.128	0.128	0.128	0.129
0.066	3.2xE7	3.2xE7	3.2xE7	3.2xE7	3.2xE7
	0.068	0.070	0.071	0.071	0.068
	0.125	0.128	0.127	0.128	0.129
0.067	3.2xE7	3.2xE7	3.2xE7	3.2xE7	3.2xE7
	0.069	0.069	0.070	0.070	0.068
	0.129	0.128	0.127	0.128	0.129
0.068	3.2xE7	3.2xE7	3.2xE7	3.2xE7	4.8xE7
	0.068	0.070	0.070	0.070	0.075
	0.125	0.126	0.127	0.128	0.130
0.069	4.8xE7	3.2xE7	3.2xE7	3.2xE7	4.8xE7
	0.076	0.070	0.071	0.070	0.074
	0.125	0.126	0.127	0.128	0.129
0.070	4.8xE7	4.8xE7	3.2xE7	3.2xE7	4.8xE7
	0.074	0.074	0.071	0.070	0.075
	0.129	0.129	0.127	0.128	0.129

*E7= 1x10

Çizelge 4.3'teki minimum maliyet fonksiyonu değeri 3.2xE7 dir. Minumum maliyet fonksiyonu değeri ile birlikte elde edilen optimal ve alt optimal sonuçlar, $r_{p,opt}$, $r_{b,opt}$ = 0.072 [m], 0.127 [m]; $r_{p,opt}$, $r_{b,opt}$ = 0.071 [m], 0.127 [m] ; $r_{p,opt}$, $r_{b,opt}$ = 0.071 [m], 0.128 [m] dir.

Çizelge 4.4 İç noktalar yöntemi ile optimize edilen hareketli ve sabit platfor yarıçapları, $r_{p,opt}$, $r_{b,opt}$

Maliyet.F(x)		Sabit Platform Yarıçapı, r_b						
$r_{p,opt\ [m]}$ / $r_{b,opt\ [m]}$		0.125	0.130	0.135	0.140	0.145	0.150	0.155
Hareketli Platform Yarıçapı, r_p	0.070	4.8xE7	4.8xE7	3.2xE7	6.4 xE7	6.4 xE7	3.2xE7	3.2xE7
		0.076	0.088	0.072	0.077	0.077	0.070	0.072
		0.125	0.118	0.127	0.125	0.126	0.126	0.127
	0.075	6.4xE7	8xE7	6.4xE7	6.4xE7	6.4 xE7	4.8xE7	3.2xE7
		0.078	0.075	0.133	0.099	0.070	0.082	0.072
		0.126	0.130	0.108	0.125	0.170	0.122	0.127
	0.080	6.4xE7	6.4xE7	6.4xE7	3.2xE7	6.4 xE7	3.2xE7	6.4 xE7
		0.095	0.099	0.128	0.072	0.094	0.070	0.095
		0.125	0.125	0.111	0.127	0.125	0.127	0.125
	0.085	6.4xE7	6.4xE7	3.2xE7	6.4xE7	6.4xE7	4.8xE7	6.4 xE7
		0.097	0.099	0.072	0.089	0.078	0.077	0.098
		0.125	0.125	0.127	0.125	0.126	0.125	0.125
	0.090	6.4xE7	6.4xE7	9.6xE7	4.8xE7	4.8xE7	3.2xE7	6.4xE7
		0.098	0.096	0.111	0.093	0.078	0.072	0.082
		0.125	0.125	0.140	0.115	0.121	0.127	0.125
	0.095	6.4xE7	9.6xE7	6.4xE7	6.4xE7	4.8xE7	4.8xE7	6.4 xE7
		0.099	0.095	0.102	0.098	0.076	0.077	0.098
		0.125	0.130	0.124	0.125	0.126	0.125	0.125
	0.100	9.6xE7	4.8xE7	4.8xE7	6.4xE7	3.2xE7	9.6xE7	4.8xE7
		0.100	0.077	0.076	0.086	0.072	0.107	0.076
		0.125	0.125	0.125	0.128	0.127	0.135	0.125

Çizelge 4.3 ve 4.4 incelendiğinde optimal sonucun $r_{p,opt}$, $r_{b,opt}$ = 0.072 [m], 0.127 [m] olduğu iç yöntemler metoduyla da doğrulanmıştır. İç noktalar yöntemi, ardışık kuadratik programlama yöntemine göre optimizasyon başlangıç noktası değerleri eğer

yakınsama bölgesi içinde ise daha hızlı sonuca ulaştığı görülmüştür. Ancak alt optimal değerlerin bu yöntem ile tespitinin zor olduğu görülmüştür.

Çizelge 4.5 Hareketli/sabit platform yarıçapı, $r_{p,opt}$, $r_{b,opt}$ ve bacak uzunlukları, l_i'nin ardışık kuadratik programlama ile optimizasyonu

Maliyet.F(x)		Sabit Platform Yarıçapı, r_b						
$r_{p,opt\ [m]}$		0.125	0.130	0.135	0.140	0.145	0.150	0.155
$r_{b,opt\ [m]}$		Bacak Uzunluğu $l_i[m]$						
$l_{i,opt\ [m]}$		0.300	0.325	0.350	0.375	0.400	0.425	0.450
Hareketli Platform Yarıçapı, r_p	0.070	4.8xE7	4.8xE7	8xE7	8xE7	6.4xE7	-	4.8xE7
		0.076	0.075	0.070	0.091	0.094	-	0.077
		0.125	0.130	0.135	0.143	0.125	-	0.125
		0.300	0.325	0.350	0.375	0.375	-	0.388
	0.075	4.8xE7	4.8xE7	8xE7	4.8xE7	8xE7	6.4xE7	8xE7
		0.076	0.075	0.081	0.077	0.097	0.072	0.076
		0.125	0.125	0.134	0.125	0.150	0.172	0.133
		0.300	0.358	0.352	0.373	0.375	0.375	0.375
	0.080	4.8xE7	8xE7	8xE7	6.4xE7	8xE7	4.8xE7	-
		0.070	0.087	0.080	0.080	0.080	0.070	-
		0.125	0.142	0.135	0.125	0.133	0.125	-
		0.322	0.359	0.350	0.375	0.375	0.322	-
	0.085	6.4xE7	6.4xE7	6.4xE7	8xE7	6.4xE7	8xE7	6.4xE7
		0.085	0.085	0.085	0.085	0.085	0.096	0.085
		0.125	0.125	0.125	0.140	0.125	0.147	0.125
		0.300	0.300	0.367	0.375	0.300	0.375	0.300
	0.090	6.4xE7	6.4xE7	6.4xE7	6.4xE7	6.4xE7	8xE7	6.4xE7
		0.090	0.090	0.090	0.090	0.098	0.091	0.090
		0.125	0.125	0.127	0.125	0.125	0.150	0.125
		0.300	0.358	0.353	0.375	0.375	0.375	0.300
	0.095	6.4xE7	9.6xE7	6.4xE7	9.6xE7	8xE7	6.4xE7	8xE7
		0.095	0.095	0.095	0.110	0.095	0.095	0.095
		0.125	0.130	0.125	0.143	0.150	0.126	0.149
		0.300	0.325	0.357	0.375	0.375	0.388	0.375
	0.100	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	8xE7
		0.100	0.100	0.100	0.100	0.100	0.118	0.094
		0.125	0.130	0.135	0.140	0.145	0.150	0.145
		0.300	0.325	0.350	0.375	0.400	0.375	0.386

Farklı başlangıç değerleriyle gerçekleştirilmiş ardışık kuadratik programlama tabanlı sekiz değişkenli, r_b, r_p, l_i optimizasyonunun maliyet fonksiyonlarının değerleri ile optimal ve alt optimal sonuçlar Çizelge 4.5 ve 4.7'de gösterildiği gibidir. Örneğin, sabit platform yarıçapı, r_b nin başlangıç değeri 0.125 [m], r_p nin başlangıç değeri 0.070 [m] ve l_i 'nin başlangıç değeri 0.300 [m] iken, maliyet fonksiyonu değeri 4.8xE7'e eşittir. Optimal değerler $r_{p\ opt}$ için 0.076 [m], $r_{b\ opt}$ 0.125 [m] ve $l_{i\ opt}$ 0.300 [m] olarak bulunmuştur. Parametrelerin optimal değerleri kısıtlarına bağlıdır. Çizelge 4.5'ü oluştururken kullanılan kısıtlar Çizelge 4.6'te gösterilmiştir.

Çizelge 4.6 PRS için üç grup değişkenin optimizasyon kısıtları

$r_{b\ [m]}$	$r_{p\ [m]}$	l_i
$0.125\ [\mathrm{m}] < r_b < 0.175$	$0.070\ [\mathrm{m}] < r_p < 0.125\ [\mathrm{m}]$	$0.300\ [\mathrm{m}] < l_i < 0.450\ [\mathrm{m}]$

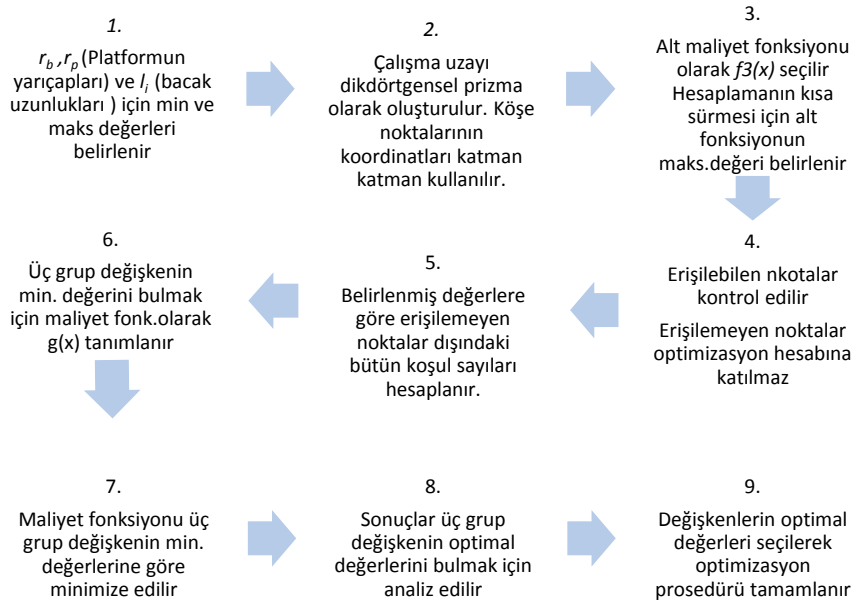
Çizelge 4.5'te görüldüğü üzere minimum maliyet fonksiyonu için birden fazla optimal ve alt optimal değerler bulunmaktadır. Örneğin, $r_{p,opt}$, $r_{b,opt}$, $l_{i,opt}$ = 0.076 [m], 0.125 [m] , 0.300 [m] değerlerine eşitken minimum maliyet fonksiyonu (4.8xE7) olarak bulunmuştur. Benzer şekilde, $r_{p,opt}$, $r_{b,opt}$, $l_{i,opt}$ = 0.077 [m], 0.125 [m] , 0.388 [m] değerlerine eşitken minimum maliyet fonksiyonu değeri yine (4.8xE7)'dir. Daha iyi doğrulukta bir sonuç elde etmek için Çizelge 4.5'teki çözüm alanı daha detaylı şekilde incelenmiştir. Bu sebeple Çizelge 4.7 oluşturulmuştur. Çizelge 4.7'de görüldüğü üzere, minimum maliyet fonksiyonu değeri olarak 3.2xE7 sonucuna erişilmiştir. Çizelge 4.7'ye göre kinematik parametrelerin optimal ve alt optimal değerleri $r_{p,opt}$, $r_{b,opt}$, $l_{i,opt}$ = 0.071 [m],0.127 [m] ,0.350 [m], $r_{p,opt}$, $r_{b,opt}$, $l_{i,opt}$ = 0.072 [m],0.127 [m] ,0.374 [m] olarak elde edilmiştir. Ardışık kuadratik programlama yöntemiyle elde ettiğimiz sonuçlardan oluşan Çizelge 4.7 iç noktalar yöntemi ile gerçekleştirilen optimizasyon hesaplamaları neticesinde elde edilen sonuçları yansıttığımız Çizelge 4.8 ile birbirine çok benzerdir. Optimizasyon sonuçları optimize edilen parametrelerin başlangıç değerleri ve optimizasyon toleranslarına göre değişebilir. Bu sebeple elde edilen optimizasyon sonuçları matematiksel hesaplamalar ile kontrol edilerek sonuçlar ile karşılaştırılmıştır.

Çizelge 4.7 Hedef bölgenin kısıtlarını daraltarak üç grup değişkenin, hareketli/sabit platform yarıçapı, $r_{p,opt}$, $r_{b,opt}$ ve bacak uzunlukları, l_i'nin ardışık kuadratik programlama yöntemi ile optimizasyonu

Maliyet.F(x)		Sabit Platform Yarıçapı,r_b				
$r_{p,opt\ [m]}$		0.125	0.126	0.127	0.128	0.129
$r_{b,opt\ [m]}$		Bacak Uzunluğu $l_{i,[m]}$				
$l_{i,opt\ [m]}$		0.300	0.325	0.350	0.375	0.400
Hareketli Platform Yarıçapı	0.070	4.8xE7	4.8xE7	3.2xE7	3.2xE7	4.8xE7
		0.076	0.075	0.071	0.072	0.076
				0.125	0.126	0.129
		0.300	0.323	0.349	0.374	0.391
	0.071	4.8xE7	4.8xE7	3.2xE7	4.8xE7	4.8xE7
		0.075	0.074	0.071	0.074	0.076
		0.125	0.126	0.127	0.128	0.129
		0.300	0.324	0.350	0.373	0.398
	0.072	4.8xE7	4.8xE7	-	4.8xE7	4.8xE7
		0.076	0.087	-	0.073	0.077
		0.125	0.120	-	0.128	0.125
		0.300	0.330	-	0.375	0.385
	0.074	4.8xE7	-	6.4xE7	6.4 xE7	4.8xE7
		0.076	-	0.091	0.087	0.075
		0.125	-	0.126	0.128	0.130
		0.300	-	0.333	0.375	0.399
	0.076	4.8xE7	4.8xE7	6.4xE7	4.8xE7	6.4xE7
		0.077	0.076	0.076	0.077	0.084
		0.125	0.125	0.127	0.125	0.128
		0.300	0.300	0.350	0.374	0.385
	0.078	6.4xE7	6.4xE7	6.4xE7	6.4xE7	6.4xE7
		0.078	0.078	0.078	0.081	0.094
		0.125	0.126	0.125	0.129	0.125
		0.300	0.325	0.300	0.375	0.355
	0.080	4.8xE7	6.4xE7	6.4xE7	6.4xE7	4.8xE7
		0.070	0.080	0.094	0.090	0.076
		0.125	0.126	0.126	0.126	0.125
		0.322	0.325	0.335	0.375	0.335

Çizelge 4.8 Hareketli/sabit platform yarıçapı, $r_{p,opt}$, $r_{b,opt}$ ve bacak uzunlukları, l_i'nin iç noktalar yöntemi ile optimizasyonu

Maliyet.F(x)		Sabit Platform Yarıçapı, r_b						
$r_{p,opt\ [m]}$		0.125	0.130	0.135	0.140	0.145	0.150	0.155
$r_{b,opt\ [m]}$		Bacak Uzunluğu $l_i[m]$						
$l_{i,opt\ [m]}$		0.300	0.325	0.350	0.375	0.400	0.425	0.450
Hareketli Platform Yarıçapı, r_p	0.070	4.8xE7	4.8xE7	8xE7	8xE7	4.8xE7	8xE7	6.4xE7
		0.070	0.088	0.070	0.090	0.077	0.070	0.083
		0.125	0.119	0.135	0.142	0.125	0.150	0.125
		0.300	0.329	0.350	0.375	0.393	0.425	0.388
	0.075	4.8xE7	3.2xE7	4.8xE7	4.8xE7	8xE7	3.2xE7	8xE7
		0.075	0.071	0.076	0.077	0.075	0.070	0.075
		0.125	0.127	0.125	0.125	0.145	0.126	0.155
		0.300	0.318	0.346	0.300	0.400	0.308	0.450
	0.080	9.6xE7	9.6xE7	8xE7	4.8xE7	8xE7	4.8xE7	8xE7
		0.107	0.080	0.080	0.076	0.080	0.070	0.080
		0.131	0.130	0.135	0.129	0.145	0.125	0.155
		0.317	0.325	0.350	0.347	0.400	0.300	0.450
	0.085	9.6xE7	3.2xE7	6.4xE7	8xE7	4.8xE7	3.2xE7	6.4xE7
		0.100	0.070	0.088	0.085	0.075	0.071	0.083
		0.130	0.126	0.125	0.140	0.129	0.127	0.125
		0.345	0.303	0.300	0.375	0.309	0.307	0.370
	0.090	8xE7	4.8xE7	6.4xE7	6.4xE7	6.4xE7	8xE7	6.4xE7
		0.070	0.089	0.096	0.098	0.096	0.095	0.101
		0.138	0.118	0.126	0.125	0.126	0.149	0.124
		0.356	0.354	0.361	0.358	0.363	0.409	0.300
	0.095	6.4xE7	9.6xE7	6.4xE7	3.2xE7	9.6xE7	6.4xE7	8xE7
		0.097	0.095	0.082	0.072	0.095	0.096	0.103
		0.125	0.130	0.125	0.127	0.145	0.125	0.154
		0.357	0.325	0.310	0.363	0.400	0.369	0.441
	0.100	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	9.6xE7	6.4xE7
		0.100	0.100	0.100	0.100	0.100	0.106	0.098
		0.125	0.130	0.135	0.140	0.145	0.129	0.125
		0.300	0.325	0.350	0.375	0.400	0.351	0.309



Şekil 4.5 Üç grup değişkenin optimizasyonunda kullanılan algoritmanın adımları

Üç grup kinematik parametrenin optimizasyonu için, alt maliyet fonksiyonu $f_3(x)$ ve Şekil 4.5'teki algoritma yardımıyla Çizelge 4.5 ve Çizelge 4.7'deki sonuçlar elde edilmiştir.

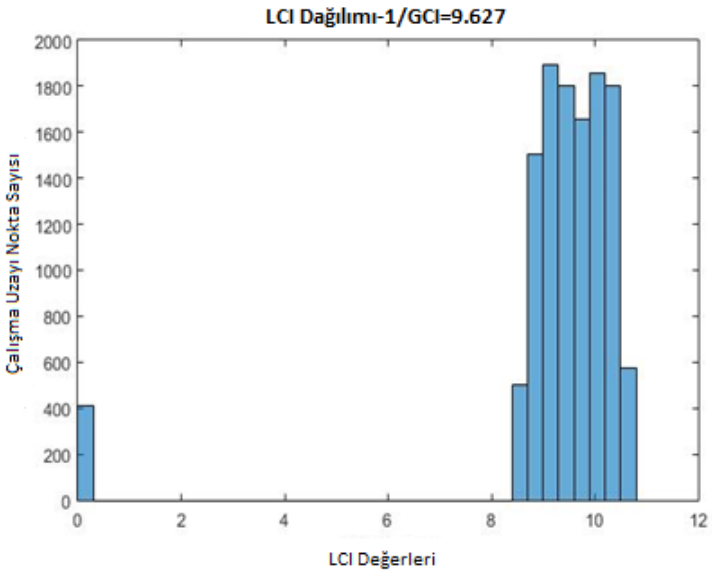
Çizelge 4.3'teki alt optimal değerler kullanılarak ters kinematik denklemler yardımıyla aşağıdaki Çizelge 4.9 oluşturulmuştur.

Çizelge 4.9 PRS'nin alt optimal parametre değerlerine göre bacak uzunlukları

$r_{b,opt}$ / $r_{p,opt}$	$L_{maks\ [m]}$	$L_{min\ [m]}$	$L_{uzunluk\ [m]}$	U	*Erişilebilen Nokta Sayısı*
0.127/0.071	0.4550	0.2928	0.3739	1.2505	11592
0.127/0.072	0.4552	0.2926	0.3739	1.2503	11588
0.128/0.071	0.4554	0.2929	0.3742	1.2486	11588

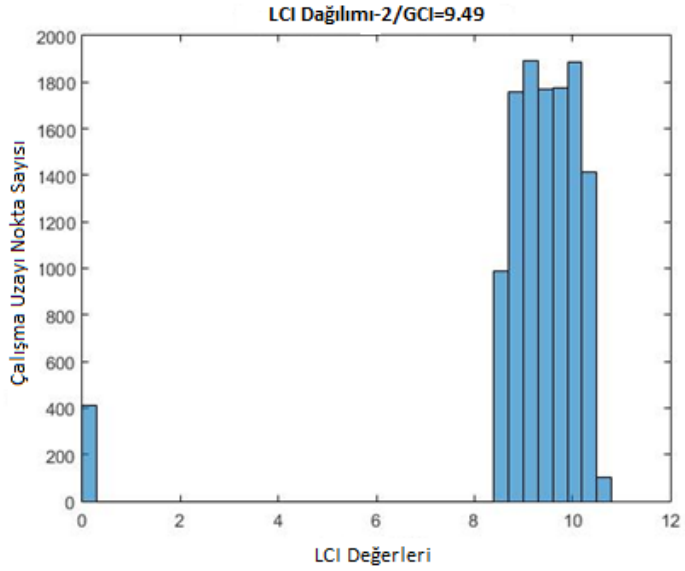
Çizelge 4.9'da hareketli ve sabit platformun alt optimal yarıçap değerlerine göre elde edilen *L_{maks}*, *L_{min}*, *u (homojenlik indisi)* ve *erişilebilen nokta sayısı* değerleri gösterilmiştir. Hesaplanmış olan bu değerlere göre Çizelge 4.7'deki optimal parametre değerleri kullanılarak her bir çözüm için, GCI ve LCI değerleri hesaplanmıştır. Şekil 4.6'da ilk alt optimal sonuçlar ile elde edilmiş LCI (yerel koşul indisi) değerleri gösterilmektedir. Şekil 4.6'da gösterildiği üzere yaklaşık olarak çalışma uzayında erişilemeyen 400 nokta bulunmuştur. Erişilebilen noktaların LCI değerleri 8 ile 11

arasında değişmektedir. GCI, (Global koşul indisi) LCI değerlerinin aritmetik ortalaması olarak ifade edilmektedir.



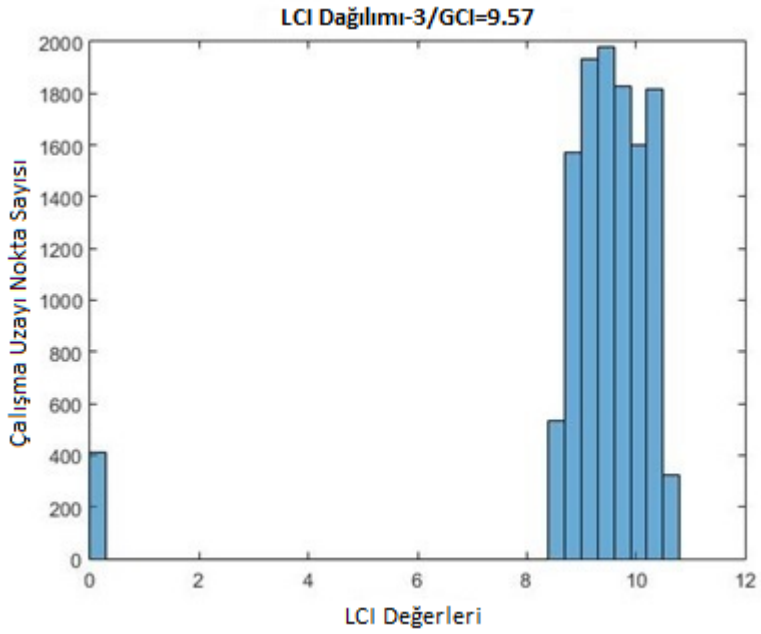
Şekil 4.6 Erişilebilir/erişilemez çalışma uzayındaki noktaların LCI değerleri, alt optimal değerlere göre, $r_{b,opt}$ = 0.127 [m], $r_{p,opt}$ = 0.071 [m] ve l_i = 0.300-0.450 [m]

Şekil 4.6'da $r_{b,opt}$ 0.127 [m] değerine, $r_{p,opt}$ 0.071 [m] değerine eşit ve l_i 0.300-0.450 [m] değer aralığında olduğu zaman, elde edilmiş yerel koşul sayılarının (LCI) çalışma uzayındaki nokta sayısına göre dağılımını ve global koşul sayısını (GCI) göstermektedir. Grafiktende anlaşılacağı üzere GCI değeri 9.62 ye eşittir.



Şekil 4.7 Erişilebilir/erişilemez çalışma uzayındaki noktaların LCI değerleri, alt optimal değerlere göre, $r_{b,opt}$ = 0.127 [m], $r_{p,opt}$ = 0.072 [m] ve l_i = 0.300-0.450 [m]

Şekil 4.7'de $r_{b,opt}$ 0.127 [m] değerine, $r_{p,opt}$ 0.072 [m] değerine eşit ve l_i 0.300-0.450 [m] değer aralığında olduğu zaman, elde edilmiş yerel koşul sayılarınının (LCI) çalışma uzayındaki nokta sayısına göre dağılımını ve global koşul sayısını (GCI) göstermektedir. Grafikten'de anlaşılacağı üzere GCI değeri 9.49 a eşittir.



Şekil 4.8 Erişilebilir/erişilemez çalışma uzayındaki noktaların LCI değerleri, alt optimal değerlere göre, $r_{b,opt}$ = 0.128 [m], $r_{p,opt}$ = 0.071 [m] ve l_i = 0.300-0.450 [m]

Şekil 4.8'de $r_{b,opt}$ 0.128 [m] değerine, $r_{p,opt}$ 0.071 [m] değerine eşit ve l_i 0.300-0.450 [m] değer aralığında olduğu zaman, elde edilmiş yerel koşul sayılarınının (LCI) çalışma uzayındaki nokta sayısına göre dağılımını ve global koşul sayısını (GCI) göstermektedir. Grafiktende anlaşılacağı üzere GCI değeri 9.57 ye eşittir.

Şekil 4.6-7-8'deki bütün yerel koşul sayıları homojen jakobyen matrisinden elde edilmiş değerlerdir. Jakobyen matrisini homojenleştirmek için matrisin bütün elemanları matrisin derecesine bölünmüştür. Şekil 4.6-7-8'de gösterildiği üzere, yerel koşul sayısı LCI ve global koşul sayısı GCI değerleri çok yakındır; bunun nedeni iki parametrenin, $r_{b,opt}$ ve $r_{p,opt}$,sadece 0.001[m] kadar birbirlerinden farklı olmasıdır. Grafiklerde LCI değerlerinin dağılımının farklı olduğu görülmektedir; ancak bu GCI değerlerini kayda değer şekilde etkilememektedir. Çizelge 4.9'da gösterildiği üzere, bacak uzunluğunun l_i kısıtlarını değiştirerek bütün çalışma uzayına erişebilme imkanımız vardır. Bu çalışmayla önceden tasarım aşamasında PRS'nin bacak uzunluklarını yeniden belirleyerek bunun

gerçekleştirilebileceği görülmüştür. Eğer çizelge 4.7 ile 4.9'u karşılaştıracak olursak hesaplanan sonuçlar birbirinden farklıdır; fakat birbirine çok yakındır. Bunun sebebi başlangıç değerlerinin optimizasyonun yakınsama toleransını ve optimizasyon prosedürünü etkilemesidir. Çizelge 4.7 ile Çizelge 4.8'i incelediğimizde ardışık kuadratik programlama yöntemiyle ve iç noktalar yöntemiyle elde ettiğimiz optimal ve alt optimal sonuçların eşit ve birbirine yakın olması bulunan sonucun doğruluğunu göstermektedir.

4.2 Sonuç

PRS tasarlarken geometrik parametreleri belirlemek verimlilik açısından çok önemlidir. Onların kompleks yapıları gereği tasarımcılar tarafından optimize edilmeye ihtiyaç duyarlar. Literatürde PRS'in kinematiğinin optimizasyonu ile ilgili birçok çalışma bulunmaktadır. Bu araştırmalarda, çeşitli optimizasyon yöntemleri kullanılmıştır. Bu tezde, 3x3 SP mekanizmasını ardışık kuadratik programlama optimizasyon yöntemi kullanılarak önceden tanımlanmış çalışma uzayı ve üç grup parametre için (hareketli platformun çapı, sabit platformun çapı, bacak uzunlukları) optimizasyonunun verimliliği kontrol edilmiştir. Ardışık kuadratik programlama tabanlı optimizasyondan elde edilen optimal ve alt optimal değerler iç noktalar yöntemiyle elde edilen sonuçlar ile karşılaştırılarak optimizasyon doğrulanmıştır. Sonuç olarak, çok parametrenin aynı anda optimizasyonu için ilgili maliyet fonksiyonunu formüle etmenin kolay olmadığı ve sonuçların kontrol edilmesi gerektiği tespit edilmiştir. Bu tezde ilk olarak, hareketli platform yarıçapını r_p sonrasında, hareketli platform r_p ve sabit platform yarıçapını r_b, eş zamanlı olarak; son olarakta, r_p, r_b ve bacak uzunlukları l_i aynı anda optimize edilmiştir. Optimizasyon esnasında, toleranslar, algoritma adımları ve uygulanan/tanımlanmış maliyet fonksiyonlarının yakınsama verimliliğini etkilediği gözlemlenmiştir. Optimizasyonun başında her parametre için seçilen başlangıç değerlerinin optimizasyonun yakınsama toleransını etkilediği görülmüştür. İleri ki çalışmalar için, optimize edilecek kinematik parametre sayısı arttırılarak farklı maliyet fonksiyonları ile farklı optimizasyon yöntemleri araştırılıp uygulanabilir.

KAYNAKLAR

[1] Marlet, J.P., (2006). Parallel Robots, Second Edition, Springer, INRIA

[2] Cappel, K., (1967). Motion Simulator, patent no: 3.295.224, United State Patent Office.

[3] Pollard, Jr. W.L.G., (1940). Spray Painting Machine, patent no: 2.213.108, United State Patent Office.

[4] Minsky, M., (1972). Manipulator design vignettes, Research Report 267, MIT AI Lab.

[5] Hunt, K.H., (1978). Kinematic geometry of mechanisms, Clarendon Press, Oxford.

[6] Yıldız, İ., (2012). Uzaysal hareket eden taşıtların Stewart platform mekanizması ile tek noktadan kuvvet geri beslemeli kontrolü, Doktora tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 295837.

[7] Özkol, İ., Yurt, S.N., Alp, H., Anlı, E. (2005). ”Paralel Mekanizmaların Kinematiği, Dinamiği ve Çalışma Uzayı”,Havacılık ve Uzay Teknolojileri Dergisi, 2-1: 19-36.

[8] Saputra, V.B., Ong, S.K. ve Nee, A.Y.C. (2015). ”A swarm optimization approach for solving workspace determination of parallel manipulators”, Robotica, 33: 649–668.

[9] Gokul, N., Shrivatsan, R., Venkatasubramanian, K. ve Dash, A.K., (2015). “Determination Of Constant Orientation Workspace Of A Stewart Platform By Geometrical Method”, Applied Mechanics and Materials, 813-814: 997-1001.

[10] Zeng, R., Dai,S., Xie,W. ve Bhat, R., (2015). ”Constraint Conditions Determination for Singularity-free Workspace of Central Symmetric Parallel Robots“, Elsevier, 48-3: 1930–1935.

[11] Hosseini, M.A., Daniali, H.M., (2011), “Dexterous Workspace Shape and Size Optimization of Tricept Parallel Manipulator”, International Journal of Robotics, 2: 18-26

[12] Hosseini, M.A., Daniali, H.M., (2015). “Cartesian workspace optimization of Tricept parallel manipulator with machining application”, Robotica, 33: 1948–1957.

[13] Pond, G., Carretero, J.A., (2006). “Formulating Jacobian matrices for the dexterity analysis of parallel manipulators”, Mechanism and Machine Theory, 41: 1505–1519.

[14] Abbasnejad, G., Daniali, H.M. ve Kazemi, S.M., (2011). “A new approach to determine the maximal singularity-free zone of 3-RPR planar parallel manipulator”, Robotica, 30: 1005–1012.

[15] Marlet, J.P., (1996). ”Workspace-Oriented Methodology for Designing a Paralel Manipulator”, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, April 1996, Minneapolis, 4: 3726-3731.

[16] Ganesh, M., Bihari, B., Rathore, V.S. and Kumar, D.,(2016). ”Determination of the closed-form workspace area expression and dimensional optimization of planar parallel manipulators”, Robotica, 35: 2056-2075

[17] Gan, D., Dias, J. ve Seneviratne, L., (2016). ”Unified kinematics and optimal design of a 3rRPS metamorphic parallel mechanism with a reconfigurable revolute joint”, Mechanism and Machine Theory, 96: 239–254.

[18] Jianxun, F., Feng, G., (2016). “Optimal Design of a 3-Leg 6-DOF Parallel Manipulator for a Specific Workspace”, Chinese Journal of Mechanical Engineering, 29: 659-668

[19] Jiang, Q., Gosselin, C.M., (2009). “Geometric Optimization of the MSSM Gough–Stewart Platform”, Journal of Mechanisms and Robotics, ASME, 1: 1-8

[20] Lou, Y., Liu, G., Chen, N. ve Li, Z., (2005), “Optimal Design of Parallel Manipulators for Maximum Effective Regular Workspace”, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2-6 Aug 2005, Edmonton.

[21] Zhang, D., Wei, B., (2015). "Global Stiffness and Well-Conditioned Optimization Analysis of 3UPU-UPU Robot Based on Pareto Front Theory”, Springer, 9320: 124-133.

[22] Lara-Molina, F.A., Rosairo, J.M., ve Dumur, D., (2010). “Multi-Objective Design of a Parallel Manipulator Using Global Indices”, The Open Mechanical Engineering Journal, 4: 37-47.

[23] Kelaiaia, R., Zaatri, A., ve Company, O., (2012). “Multiobjective Optimization of 6-dof UPS Parallel Manipulators”, Advanced Robotics, 26: 1885-1913.

[24] Shirazi, A.R., Fakhrabadi, M.M.S. ve Ghanbari, A., (2012). “Optimal Design of a 6 DOF Parallel Manipulator Using Particle Swarm Optimization”, 26: 1419-1441.

[25] Kurtz, R., Hayward, V., (1992). “Multiple-Goal Kinematic Optimization of a Parallel Spherical Mechanism with Actuator Redundancy”, IEEE Transactions on Robotics, 8: 644-651.

[26] Cirillo, A., Cirillo, P., De Maria, G., Marino, A. ve Natale, C., (2017). “Optimal custom design of both symmetric and unsymmetrical hexapod robots for aeronautics applications”, Robotics and Computer-Integrated Manufacturing, 44: 1-16.

[27] Zhufeng, S., Xioaqiang, T., Liping, W. ve Dengfeng, S., (2015). ”Atlas Based Kinematic Optimum Design of the Stewart Parallel Manipulator”, Chınese Journal Of Mechanıcal Engıneerıng, 28: 20-28 .

[28] Babu, S.R., Raju, V.R., Ramji, K., (2013). “Design For Optimal Performance Of 3-RPS Parallel Manipulator Using Evolutionary Algorithms”, Transactions of the Canadian Society for Mechanical Engineering, 37: 135-160.

[29] Kelaiaia, R., Zaatri, A., Company, O., Chikh, L., (2016). “Some investigations into the optimal dimensional synthesis of parallel robots”, IJIET, Springer, 83: 1525-1538

[30] Fattah, A., Jazi, S.H., (2001). “Optimal Design of Parallel Manipulators”, ICAR 2001, August 22-25 2001, Isfahan.

[31] Lopes, A.M., Pires, E.J.S., ve Barbosa, M.R.,(2012). “Design of a Parallel Robotic Manipulator using Evolutionary Computing”, International Journal of Advanced Robotic Systems, SAGE, 9: 1-13.

[32] Lou, Y., Zhang, Y., Huang, R., Chen, X. ve Li, Z., (2014) “Optimization Algorithms for Kinematically Optimal Design of Parallel Manipulators”, IEEE Transaction on Automation Science and Engineering, 11: 574-584

[33] Baig, R.U., Pugazhenthi, S., (2015). “Design Optimization of Stewart Platform Configuration for Active Vibration Isolation”, Indian Journal of Science and Technology, 8: 1-8.

[34] Liu, Z.,Tang, X., Shaoa, Z. ve Wang,L., (2016). ”Dimensional Optimization of The Stewart Platform Based on Inertia Decoupling Characteristic”, Robotica, 34: 1151–1167.

[35] Huang, T., Whitehouse, D.J. ve Wang, J., (1998). “The Local Dexterity, Optimal Architecture and Design Criteria of Parallel Machine Tools”, Elsevier, 47: 346-350.

[36] Gao, Z., Zhang, D. ve Ge, Y., (2010) “Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches”, Robotics and Computer-Integrated Manufacturing, Elsevier, 26: 180–189.

[37] Cardou, P., Bouchard, S., ve Gosselin, C., (2010). “Kinematic-Sensitivity Indices for Dimensionally Nonhomogeneous Jacobian Matrices”, IEEE Trans. On Robotics, Vol. 26: 166-173.

[38] Zhang, Y., Yao, Y., (2006). “Kinematic Optimal Design of 6-UPS Parallel Manipulator”, Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, June 25 - 28, 2006, Luoyang, China.

[39] Abedinnasab, M.H., Yoon, Y.J., Zohoor, H., (2012). “Exploiting Higher Kinematic Performance Using a 4-Legged Redundant PM Rather than Gough-Stewart Platforms”, Serial and Parallel Robot Manipulators - Kinematics, Dynamics, Control and Optimization, InTech, ISBN:978-953-51-0437-7, 43-65.

[40] Luces, M., Mills, J.K., Benhabib, B., (2016). “A Review of Redundant Parallel Kinematic Mechanisms,” Journal of Intelligent Robotic System, Springer, 86: 175-198

[41] Srinivas, P.C., Kumar, Y.D., Rao, P.D., Sreenivasulu, V., (2013). “Kinematic Analysis and Optimum Design of 8-8 Redundant Spatial In-Parallel Maniputator”, IJIET, 2: 452-460

[42] Yıldız, İ., Ömürlü, V.E., Ekicioğlu, Z., Güney, A., (2010). “6 Serbestlik Dereceli Parallel Mekanizmadaki İleri Kinematik Analiz Yöntemleri”, TOK2010, Yıldız Teknik Üniversitesi, İstanbul.

[43] Baker, K., March 29, 2005. (Revised January 14, 2013). “Singular Value Decomposition Tutorial”.

[44] Azimli, A., (2011), "Matematiksel Optimizasyon", Papatya Yayıncılık, Istanbul.

[45] Rao, S.S., (2009). "Engineering Optimization", Fourth Edition, John Wiley, New Jersey.

[46] Modungwa, D., Tlale, N.S. ve Twala, B., (2012). “Optimization Approaches Applied in Dimensional Synthesis of Parallel Mechanisms,” Transaction on Control and Mechanical Systems, 1: 57-64.

[47] Bazaraa, S., M., Sherali, D.,H., Shetty, M., C., (2006), "Nonlinear Programming Theory and Algorithms, Third Edition, John Wiley, New Jersey

EK-A

MATLAB KODLARI

Tek değişkenli optimizasyonda koşul sayılarının hesaplanması

```
function[Z]=condition_number_optz(radius_t,px,py,pz,x_ang,y
_ang,z_ang)

deg2rad = pi/180; % dereceden radyana dönüşüm
% Üst ve alt plakada bağlantı noktalarının tanımlanması
alpha_b =0; % alt plakada 60 derecelik ofset açısının
oluşturulması için bağlantı açısı
alpha_t =2*(pi/3); % üst plakada 60 derecelik ofset
açısının oluşturulması için bağlantı açısı
radius_b = 0.175; % alt plaka yarıçapı
% radius_t = 0.150; % üst plaka yarıçapı
lamda_1=(pi/3)-(alpha_t/2);
lamda_3=(3*pi/3)-(alpha_t/2);
lamda_5=(5*pi/3)-(alpha_t/2);
lamda_2=(lamda_1+alpha_t);
lamda_4=(lamda_3+alpha_t);
lamda_6=(lamda_5+alpha_t);
Blamda_1=(pi/3)-(alpha_b/2);
Blamda_3=(3*pi/3)-(alpha_b/2);
Blamda_5=(5*pi/3)-(alpha_b/2);
Blamda_2=(Blamda_1+alpha_b);
Blamda_4=(Blamda_3+alpha_b);
Blamda_6=(Blamda_5+alpha_b);
q1=([radius_t*cos(lamda_1) radius_t*sin(lamda_1) 0]);
q3=([radius_t*cos(lamda_3) radius_t*sin(lamda_3) 0]);
q5=([radius_t*cos(lamda_5) radius_t*sin(lamda_5) 0]);
b2=[radius_b*cos(Blamda_2) radius_b*sin(Blamda_2) 0];
b4=[radius_b*cos(Blamda_4) radius_b*sin(Blamda_4) 0];
b6=[radius_b*cos(Blamda_6) radius_b*sin(Blamda_6) 0];
%Bağlantı noktalarının tanımlanması
%Alt bağlantı noktaları
XB1=b2(1,1); YB1=b2(1,2); ZB1=b2(1,3);
XB2=b2(1,1); YB2=b2(1,2); ZB2=b2(1,3);
```

```
XB3=b4(1,1); YB3=b4(1,2); ZB3=b4(1,3);
XB4=b4(1,1); YB4=b4(1,2); ZB4=b4(1,3);
XB5=b6(1,1); YB5=b6(1,2); ZB5=b6(1,3);
XB6=b6(1,1); YB6=b6(1,2); ZB6=b6(1,3);
%Üst bağlantı noktaları
xT1=q1(1,1); yT1=q1(1,2); zT1=q1(1,3);
xT2=q3(1,1); yT2=q3(1,2); zT2=q3(1,3);
xT3=q3(1,1); yT3=q3(1,2); zT3=q3(1,3);
xT4=q5(1,1); yT4=q5(1,2); zT4=q5(1,3);
xT5=q5(1,1); yT5=q5(1,2); zT5=q5(1,3);
xT6=q1(1,1); yT6=q1(1,2); zT6=q1(1,3);

%Üst bağlantı noktalarının alt orjine göre koordinatlarının
belirlenmesi
XT1=(cos(z_ang)*cos(y_ang))*xT1+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT1+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT1+px;
XT2=(cos(z_ang)*cos(y_ang))*xT2+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT2+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT2+px;
XT3=(cos(z_ang)*cos(y_ang))*xT3+(cos(z_ang)*sin(y_ang)*sin
( x_ang)-sin(z_ang)*cos(x_ang))*yT3+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT3+px;
XT4=(cos(z_ang)*cos(y_ang))*xT4+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT4+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT4+px;
XT5=(cos(z_ang)*cos(y_ang))*xT5+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT5+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT5+px;
XT6=(cos(z_ang)*cos(y_ang))*xT6+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT6+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT6+px;
YT1=(sin(z_ang)*cos(y_ang))*xT1+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT1+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT1+py;
YT2=(sin(z_ang)*cos(y_ang))*xT2+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT2+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT2+py;
YT3=(sin(z_ang)*cos(y_ang))*xT3+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT3+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT3+py;
YT4=(sin(z_ang)*cos(y_ang))*xT4+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT4+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT4+py;
YT5=(sin(z_ang)*cos(y_ang))*xT5+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT5+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT5+py;
YT6=(sin(z_ang)*cos(y_ang))*xT6+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT6+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT6+py;
```

```
ZT1=(-sin(y_ang))*xT1+(cos(y_ang)*sin(x_ang))*yT1+
(cos(y_ang)*cos(x_ang))*zT1+pz;
ZT2=(-sin(y_ang))*xT2+(cos(y_ang)*sin(x_ang))*yT2+
(cos(y_ang)*cos(x_ang))*zT2+pz;
ZT3=(-sin(y_ang))*xT3+(cos(y_ang)*sin(x_ang))*yT3+
(cos(y_ang)*cos(x_ang))*zT3+pz;
ZT4=(-sin(y_ang))*xT4+(cos(y_ang)*sin(x_ang))*yT4+
(cos(y_ang)*cos(x_ang))*zT4+pz;
ZT5=(-sin(y_ang))*xT5+(cos(y_ang)*sin(x_ang))*yT5+
(cos(y_ang)*cos(x_ang))*zT5+pz;
ZT6=(-sin(y_ang))*xT6+(cos(y_ang)*sin(x_ang))*yT6+
(cos(y_ang)*cos(x_ang))*zT6+pz;
%Bacak boyları
L1=((XT1-XB1)^2+(YT1-YB1)^2+(ZT1-ZB1)^2)^(1/2);
L2=((XT2-XB2)^2+(YT2-YB2)^2+(ZT2-ZB2)^2)^(1/2);
L3=((XT3-XB3)^2+(YT3-YB3)^2+(ZT3-ZB3)^2)^(1/2);
L4=((XT4-XB4)^2+(YT4-YB4)^2+(ZT4-ZB4)^2)^(1/2);
L5=((XT5-XB5)^2+(YT5-YB5)^2+(ZT5-ZB5)^2)^(1/2);
L6=((XT6-XB6)^2+(YT6-YB6)^2+(ZT6-ZB6)^2)^(1/2);

if
((L1<0.3)||(L1>0.45)||L2<0.3)||(L2>0.45||L3<0.3)||(L3>0.45|
|L4<0.3)||(L4>0.45||L5<0.3)||(L5>0.45||L6<0.3)||(L6>0.45)
%    fprintf('\nL1 sinir disi deger aldi : %f\n',L1);
%end
%if ((L2<0.3)||(L2>0.45))
%    fprintf('\nL2 sinir disi deger aldi : %f\n',L1);
%end
%if ((L3<0.3)||(L3>0.45))
%    fprintf('\nL3 sinir disi deger aldi : %f\n',L1);
%end
%if ((L4<0.3)||(L4>0.45))
%    fprintf('\nL4 sinir disi deger aldi : %f\n',L1);
%end
%if ((L5<0.3)||(L5>0.45))
%    fprintf('\nL5 sinir disi deger aldi : %f\n',L1);
%end
%if ((L6<0.3)||(L6>0.45))
%    fprintf('\nL6 sinir disi deger aldi : %f\n',L1);
%end
fprintf('\nL sinir disi deger aldi : \n')
Z=0;
else
%pause;

%Jakobien Matrisinin bulunması
LX1=XT1-XB1;    LX2=XT2-XB2;    LX3=XT3-XB3;    LX4=XT4-XB4;
LX5=XT5-XB5; LX6=XT6-XB6;
LY1=YT1-YB1;    LY2=YT2-YB2;    LY3=YT3-YB3;    LY4=YT4-YB4;
LY5=YT5-YB5; LY6=YT6-YB6;
```

```
LZ1=ZT1-ZB1;    LZ2=ZT2-ZB2;    LZ3=ZT3-ZB3;    LZ4=ZT4-ZB4;
LZ5=ZT5-ZB5; LZ6=ZT6-ZB6;

J11=LX1/L1; J21=LX2/L2; J31=LX3/L3; J41=LX4/L4; J51=LX5/L5;
J61=LX6/L6;
J12=LY1/L1; J22=LY2/L2; J32=LY3/L3; J42=LY4/L4; J52=LY5/L5;
J62=LY6/L6;
J13=LZ1/L1; J23=LZ2/L2; J33=LZ3/L3; J43=LZ4/L4; J53=LZ5/L5;
J63=LZ6/L6;
J14=(LX1*((-sin(z_ang)*cos(y_ang))*xT1+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT1+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT1
)+LY1*((cos(z_ang)*cos(y_ang))*xT1+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT1+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT1))/L1;
J24=(LX2*((-sin(z_ang)*cos(y_ang))*xT2+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT2+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT2
)+LY2*((cos(z_ang)*cos(y_ang))*xT2+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT2+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT2))/L2;
J34=(LX3*((-sin(z_ang)*cos(y_ang))*xT3+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT3+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT3
)+LY3*((cos(z_ang)*cos(y_ang))*xT3+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT3+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT3))/L3;
J44=(LX4*((-sin(z_ang)*cos(y_ang))*xT4+(-
sin(z_ang)*sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*
yT4+(-sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*
sin(x_ang))*zT4)+LY4*((cos(z_ang)*cos(y_ang))*xT4+
(cos(z_ang)*sin(y_ang)*sin(x_ang)-sin(z_ang)*cos(x_ang)*
yT4+(cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*
sin(x_ang))*zT4))/L4;
J54=(LX5*((-sin(z_ang)*cos(y_ang))*xT5+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT5+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT5
)+LY5*((cos(z_ang)*cos(y_ang))*xT5+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT5+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT5))/L5;
J64=(LX6*((-sin(z_ang)*cos(y_ang))*xT6+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT6+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT6
)+LY6*((cos(z_ang)*cos(y_ang))*xT6+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT6+(cos(z_ang)*
sin(y_ang)* cos(x_ang)+sin(z_ang)*sin(x_ang))*zT6))/L6;
J15=(LX1*((-
cos(z_ang)*sin(y_ang))*xT1+(cos(z_ang)*cos(y_ang)*
sin(x_ang))*yT1+(cos(z_ang)*cos(y_ang)*cos(x_ang))*zT1)
+LY1*((-sin(z_ang)*sin(y_ang))*xT1+(sin(z_ang)*cos(y_ang)
```

```
*sin(x_ang))*yT1+(sin(z_ang)*cos(y_ang)*cos(x_ang))*zT1)+LZ
1*((-cos(y_ang))*xT1+(-sin(y_ang)*sin(x_ang))*
yT1+(-sin(y_ang)*cos(x_ang))*zT1))/L1;
J25=(LX2*((-cos(z_ang)*sin(y_ang))*xT2+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT2+(cos(z_ang)*cos(y_ang)*
cos(x_ang))*zT2)+LY2*((-sin(z_ang)*sin(y_ang))*xT2+
(sin(z_ang)*cos(y_ang)*sin(x_ang))*yT2+(sin(z_ang)*
cos(y_ang)*cos(x_ang))*zT2)+LZ2*((-cos(y_ang))*
xT2+(-sin(y_ang)*sin(x_ang))*yT2+(-sin(y_ang)*cos(x_ang))
*zT2))/L2;
J35=(LX3*((-cos(z_ang)*sin(y_ang))*xT3+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT3+(cos(z_ang)*cos(y_ang)*
cos(x_ang))*zT3)+LY3*((-sin(z_ang)*sin(y_ang))*
xT3+(sin(z_ang)*cos(y_ang)*sin(x_ang))*yT3+(sin(z_ang)*
cos(y_ang)*cos(x_ang))*zT3)+LZ3*((-cos(y_ang))*xT3+
(-sin(y_ang)*sin(x_ang))*yT3+(-sin(y_ang)*cos(x_ang))*
zT3))/L3;
J45=(LX4*((-cos(z_ang)*sin(y_ang))*xT4+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT4+(cos(z_ang)*cos(y_ang)*
cos(x_ang))*zT4)+LY4*((-sin(z_ang)*sin(y_ang))*xT4+
(sin(z_ang)*cos(y_ang)*sin(x_ang))*yT4+(sin(z_ang)*
cos(y_ang)*cos(x_ang))*zT4)+LZ4*((-cos(y_ang))*xT4+
(-sin(y_ang)*sin(x_ang))*yT4+(-sin(y_ang)*cos(x_ang))*
zT4))/L4;
J55=(LX5*((-cos(z_ang)*sin(y_ang))*xT5+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT5+(cos(z_ang)*cos(y_ang)*
cos(x_ang))*zT5)+LY5*((-sin(z_ang)*sin(y_ang))*xT5+
(sin(z_ang)*cos(y_ang)*sin(x_ang))*yT5+(sin(z_ang)*
cos(y_ang)*cos(x_ang))*zT5)+LZ5*((-cos(y_ang))*xT5
+(-sin(y_ang)*sin(x_ang))*yT5+(-sin(y_ang)*cos(x_ang))*
zT5))/L5;
J65=(LX6*((-cos(z_ang)*sin(y_ang))*xT6+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT6+(cos(z_ang)*cos(y_ang)*
cos(x_ang))*zT6)+LY6*((-sin(z_ang)*sin(y_ang))*xT6
+(sin(z_ang)*cos(y_ang)*sin(x_ang))*yT6+(sin(z_ang)*
cos(y_ang)*cos(x_ang))*zT6)+LZ6*((-cos(y_ang))*xT6+
(-sin(y_ang)*sin(x_ang))*yT6+(-sin(y_ang)*cos(x_ang))
*zT6))/L6;
J16=(LX1*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT1+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT1)+LY1*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT1+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT1)+LZ1*
((cos(y_ang)*cos(x_ang))*yT1-(cos(y_ang)*sin(x_ang))*
zT1))/L1;
J26=(LX2*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT2+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT2)+LY2*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT2+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*zT2)
```

```
+LZ2*((cos(y_ang)*cos(x_ang))*yT2-(cos(y_ang)*sin(x_ang))*
zT2))/L2;
J36=(LX3*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT3+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT3)+LY3*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT3+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT3)+LZ3*
((cos(y_ang)*cos(x_ang))*yT3-(cos(y_ang)*sin(x_ang))*
zT3))/L3;
J46=(LX4*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*
sin(x_ang))*yT4+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+
sin(z_ang)*cos(x_ang))*zT4)+LY4*((sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*yT4+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*zT4)+LZ4*
((cos(y_ang)*cos(x_ang))*yT4-(cos(y_ang)*sin(x_ang))*
zT4))/L4;
J56=(LX5*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT5+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT5)+LY5*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT5+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT5)+LZ5*
((cos(y_ang)*cos(x_ang))*yT5-(cos(y_ang)*sin(x_ang))*
zT5))/L5;
J66=(LX6*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT6+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT6)+LY6*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT6+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT6)+LZ6*
((cos(y_ang)*cos(x_ang))*yT6-(cos(y_ang)*sin(x_ang))*
zT6))/L6;

J=[J11 J12 J13 J14 J15 J16;J21 J22 J23 J24 J25 J26;J31 J32
J33 J34 J35 J36;J41 J42 J43 J44 J45 J46;J51 J52 J53 J54 J55
J56;J61 J62 J63 J64 J65 J66];

Z= cond(J,'fro');

end
```

İki değişekenli optimizasyonda koşul sayılarının hesaplanması

```
function[Z]=
condition_number_optz(x,px,py,pz,x_ang,y_ang,z_ang)

deg2rad = pi/180; % dereceden radyana dönüşüm
% Üst ve alt plakada bağlantı noktalarının tanımlanması
alpha_b =0; % alt plakada 60 derecelik ofset açısının
oluşturulması için bağlantı açısı
alpha_t =2*(pi/3); % üst plakada 60 derecelik ofset
açısının oluşturulması için bağlantı açısı
```

```
%xin2 radius_b = 0.175; % alt plaka yarıçapı
%xin1 radius_t = 0.150; % üst plaka yarıçapı
lamda_1=(pi/3)-(alpha_t/2);
lamda_3=(3*pi/3)-(alpha_t/2);
lamda_5=(5*pi/3)-(alpha_t/2);
lamda_2=(lamda_1+alpha_t);
lamda_4=(lamda_3+alpha_t);
lamda_6=(lamda_5+alpha_t);
Blamda_1=(pi/3)-(alpha_b/2);
Blamda_3=(3*pi/3)-(alpha_b/2);
Blamda_5=(5*pi/3)-(alpha_b/2);
Blamda_2=(Blamda_1+alpha_b);
Blamda_4=(Blamda_3+alpha_b);
Blamda_6=(Blamda_5+alpha_b);
q1=([x(1)*cos(lamda_1) x(1)*sin(lamda_1) 0]);
q3=([x(1)*cos(lamda_3) x(1)*sin(lamda_3) 0]);
q5=([x(1)*cos(lamda_5) x(1)*sin(lamda_5) 0]);
b2=[x(2)*cos(Blamda_2) x(2)*sin(Blamda_2) 0];
b4=[x(2)*cos(Blamda_4) x(2)*sin(Blamda_4) 0];
b6=[x(2)*cos(Blamda_6) x(2)*sin(Blamda_6) 0];

%Bağlantı noktalarının tanımlanması
%Alt bağlantı noktaları
XB1=b2(1,1); YB1=b2(1,2); ZB1=b2(1,3);
XB2=b2(1,1); YB2=b2(1,2); ZB2=b2(1,3);
XB3=b4(1,1); YB3=b4(1,2); ZB3=b4(1,3);
XB4=b4(1,1); YB4=b4(1,2); ZB4=b4(1,3);
XB5=b6(1,1); YB5=b6(1,2); ZB5=b6(1,3);
XB6=b6(1,1); YB6=b6(1,2); ZB6=b6(1,3);
%Üst bağlantı noktaları
xT1=q1(1,1); yT1=q1(1,2); zT1=q1(1,3);
xT2=q3(1,1); yT2=q3(1,2); zT2=q3(1,3);
xT3=q3(1,1); yT3=q3(1,2); zT3=q3(1,3);
xT4=q5(1,1); yT4=q5(1,2); zT4=q5(1,3);
xT5=q5(1,1); yT5=q5(1,2); zT5=q5(1,3);
xT6=q1(1,1); yT6=q1(1,2); zT6=q1(1,3);

%Üst bağlantı noktalarının alt orjine göre koordinatlarının
belirlenmesi
XT1=(cos(z_ang)*cos(y_ang))*xT1+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT1+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT1+px;
XT2=(cos(z_ang)*cos(y_ang))*xT2+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT2+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT2+px;
XT3=(cos(z_ang)*cos(y_ang))*xT3+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT3+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT3+px;
```

```
XT4=(cos(z_ang)*cos(y_ang))*xT4+(cos(z_ang)*sin(y_ang)*sin
(x_ang)-sin(z_ang)*cos(x_ang))*yT4+(cos(z_ang)*sin(y_ang)*
cos(x_ang)+sin(z_ang)*sin(x_ang))*zT4+px;
XT5=(cos(z_ang)*cos(y_ang))*xT5+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT5+(cos(z_ang)*
sin(y_ang)* cos(x_ang)+sin(z_ang)*sin(x_ang))*zT5+px;
XT6=(cos(z_ang)*cos(y_ang))*xT6+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT6+(cos(z_ang)*
sin(y_ang)* cos(x_ang)+sin(z_ang)*sin(x_ang))*zT6+px;
YT1=(sin(z_ang)*cos(y_ang))*xT1+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT1+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT1+py;
YT2=(sin(z_ang)*cos(y_ang))*xT2+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT2+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT2+py;
YT3=(sin(z_ang)*cos(y_ang))*xT3+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT3+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT3+py;
YT4=(sin(z_ang)*cos(y_ang))*xT4+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT4+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT4+py;
YT5=(sin(z_ang)*cos(y_ang))*xT5+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT5+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT5+py;
YT6=(sin(z_ang)*cos(y_ang))*xT6+(sin(z_ang)*sin(y_ang)*sin
(x_ang)+cos(z_ang)*cos(x_ang))*yT6+(sin(z_ang)*sin(y_ang)*
cos(x_ang)-cos(z_ang)*sin(x_ang))*zT6+py;
ZT1=(-
sin(y_ang))*xT1+(cos(y_ang)*sin(x_ang))*yT1+(cos(y_ang)*cos
(x_ang))*zT1+pz;
ZT2=(-sin(y_ang))*xT2+(cos(y_ang)*sin(x_ang))*yT2+
(cos(y_ang)*cos(x_ang))*zT2+pz;
ZT3=(-sin(y_ang))*xT3+(cos(y_ang)*sin(x_ang))*yT3+
(cos(y_ang)*cos(x_ang))*zT3+pz;
ZT4=(-sin(y_ang))*xT4+(cos(y_ang)*sin(x_ang))*yT4+
(cos(y_ang)*cos(x_ang))*zT4+pz;
ZT5=(-sin(y_ang))*xT5+(cos(y_ang)*sin(x_ang))*yT5+
(cos(y_ang)*cos(x_ang))*zT5+pz;
ZT6=(-sin(y_ang))*xT6+(cos(y_ang)*sin(x_ang))*yT6+
(cos(y_ang)*cos(x_ang))*zT6+pz;

%Bacak boyları
L1=((XT1-XB1)^2+(YT1-YB1)^2+(ZT1-ZB1)^2)^(1/2);
L2=((XT2-XB2)^2+(YT2-YB2)^2+(ZT2-ZB2)^2)^(1/2);
L3=((XT3-XB3)^2+(YT3-YB3)^2+(ZT3-ZB3)^2)^(1/2);
L4=((XT4-XB4)^2+(YT4-YB4)^2+(ZT4-ZB4)^2)^(1/2);
L5=((XT5-XB5)^2+(YT5-YB5)^2+(ZT5-ZB5)^2)^(1/2);
L6=((XT6-XB6)^2+(YT6-YB6)^2+(ZT6-ZB6)^2)^(1/2);
```

```
if
((L1<0.3)||(L1>0.45)||L2<0.3)||(L2>0.45||L3<0.3)||(L3>0.45|
|L4<0.3)||(L4>0.45||L5<0.3)||(L5>0.45||L6<0.3)||(L6>0.45)
%    fprintf('\nL1 sinir disi deger aldi : %f\n',L1);
%end
%if ((L2<0.3)||(L2>0.45))
%    fprintf('\nL2 sinir disi deger aldi : %f\n',L1);
%end
%if ((L3<0.3)||(L3>0.45))
%    fprintf('\nL3 sinir disi deger aldi : %f\n',L1);
%end
%if ((L4<0.3)||(L4>0.45))
%    fprintf('\nL4 sinir disi deger aldi : %f\n',L1);
%end
%if ((L5<0.3)||(L5>0.45))
%    fprintf('\nL5 sinir disi deger aldi : %f\n',L1);
%end
%if ((L6<0.3)||(L6>0.45))
%    fprintf('\nL6 sinir disi deger aldi : %f\n',L1);
%end
fprintf('\nL sinir disi deger aldi : \n')
Z=0;
else
%pause;
%Jakobien Matrisinin bulunması
LX1=XT1-XB1;    LX2=XT2-XB2;    LX3=XT3-XB3;    LX4=XT4-XB4;
LX5=XT5-XB5; LX6=XT6-XB6;
LY1=YT1-YB1;    LY2=YT2-YB2;    LY3=YT3-YB3;    LY4=YT4-YB4;
LY5=YT5-YB5; LY6=YT6-YB6;
LZ1=ZT1-ZB1;    LZ2=ZT2-ZB2;    LZ3=ZT3-ZB3;    LZ4=ZT4-ZB4;
LZ5=ZT5-ZB5; LZ6=ZT6-ZB6;
J11=LX1/L1; J21=LX2/L2; J31=LX3/L3; J41=LX4/L4; J51=LX5/L5;
J61=LX6/L6;
J12=LY1/L1; J22=LY2/L2; J32=LY3/L3; J42=LY4/L4; J52=LY5/L5;
J62=LY6/L6;
J13=LZ1/L1; J23=LZ2/L2; J33=LZ3/L3; J43=LZ4/L4; J53=LZ5/L5;
J63=LZ6/L6;
J14=(LX1*((-sin(z_ang)*cos(y_ang))*xT1+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT1+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))
*zT1)+LY1*((cos(z_ang)*cos(y_ang))*xT1+(cos(z_ang)*
sin(y_ang)*sin(x_ang)-sin(z_ang)*cos(x_ang))*yT1+
(cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))
*zT1))/L1;
J24=(LX2*((-sin(z_ang)*cos(y_ang))*xT2+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*
yT2+(-sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*
sin(x_ang))*zT2)+LY2*((cos(z_ang)*cos(y_ang))*xT2+
```

```
(cos(z_ang)*sin(y_ang)*sin(x_ang)-sin(z_ang)*cos(x_ang))*
yT2+(cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*
sin(x_ang))*zT2))/L2;
J34=(LX3*((-sin(z_ang)*cos(y_ang))*xT3+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT3+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT3
)+LY3*((cos(z_ang)*cos(y_ang))*xT3+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT3+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT3))/L3;
J44=(LX4*((-sin(z_ang)*cos(y_ang))*xT4+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT4+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT4
)+LY4*((cos(z_ang)*cos(y_ang))*xT4+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT4+(cos(z_ang)
*sin(y_ang)* cos(x_ang)+sin(z_ang)*sin(x_ang))*zT4))/L4;
J54=(LX5*((-sin(z_ang)*cos(y_ang))*xT5+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT5+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT5
)+LY5*((cos(z_ang)*cos(y_ang))*xT5+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT5+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT5))/L5;
J64=(LX6*((-sin(z_ang)*cos(y_ang))*xT6+(-sin(z_ang)*
sin(y_ang)*sin(x_ang)-cos(z_ang)*cos(x_ang))*yT6+(-
sin(z_ang)*sin(y_ang)*cos(x_ang)+cos(z_ang)*sin(x_ang))*zT6
)+LY6*((cos(z_ang)*cos(y_ang))*xT6+(cos(z_ang)*sin(y_ang)*
sin(x_ang)-sin(z_ang)*cos(x_ang))*yT6+(cos(z_ang)*
sin(y_ang)*cos(x_ang)+sin(z_ang)*sin(x_ang))*zT6))/L6;
J15=(LX1*((-cos(z_ang)*sin(y_ang))*xT1+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT1+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT1)+LY1*((-sin(z_ang)*sin(y_ang))*xT1+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT1+(sin(z_ang)*cos(y_ang)*cos(x_ang
))*zT1)+LZ1*((-cos(y_ang))*xT1+(-sin(y_ang)*
sin(x_ang))*yT1+(-sin(y_ang)*cos(x_ang))*zT1))/L1;
J25=(LX2*((-cos(z_ang)*sin(y_ang))*xT2+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT2+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT2)+LY2*((-sin(z_ang)*sin(y_ang))*xT2+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT2+(sin(z_ang)*cos(y_ang)*cos(x_ang
))*zT2)+LZ2*((-cos(y_ang))*xT2+(-sin(y_ang)*sin(x_ang))*
yT2+(-sin(y_ang)*cos(x_ang))*zT2))/L2;
J35=(LX3*((-cos(z_ang)*sin(y_ang))*xT3+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT3+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT3)+LY3*((-sin(z_ang)*sin(y_ang))*xT3+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT3+(sin(z_ang)*cos(y_ang)*cos(x_ang
))*zT3)+LZ3*((-cos(y_ang))*xT3+(-sin(y_ang)*sin(x_ang))*
yT3+(-sin(y_ang)*cos(x_ang))*zT3))/L3;
J45=(LX4*((-cos(z_ang)*sin(y_ang))*xT4+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT4+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT4)+LY4*((-sin(z_ang)*sin(y_ang))*xT4+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT4+(sin(z_ang)*cos(y_ang)*cos(x_ang
```

```
))*zT4)+LZ4*((-cos(y_ang))*xT4+(-sin(y_ang)*sin(x_ang))*
yT4+(-sin(y_ang)*cos(x_ang))*zT4))/L4;
J55=(LX5*((-cos(z_ang)*sin(y_ang))*xT5+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT5+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT5)+LY5*((-sin(z_ang)*sin(y_ang))*xT5+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT5+(sin(z_ang)*cos(y_ang)*cos(x_ang
))*zT5)+LZ5*((-cos(y_ang))*xT5+(-sin(y_ang)*
sin(x_ang))*yT5+(-sin(y_ang)*cos(x_ang))*zT5))/L5;
J65=(LX6*((-cos(z_ang)*sin(y_ang))*xT6+(cos(z_ang)*
cos(y_ang)*sin(x_ang))*yT6+(cos(z_ang)*cos(y_ang)*cos(x_ang
))*zT6)+LY6*((-sin(z_ang)*sin(y_ang))*xT6+(sin(z_ang)*
cos(y_ang)*sin(x_ang))*yT6+(sin(z_ang)*cos(y_ang)*cos(x_ang
))*zT6)+LZ6*((-cos(y_ang))*xT6+(-sin(y_ang)*sin(x_ang))*
yT6+(-sin(y_ang)*cos(x_ang))*zT6))/L6;
J16=(LX1*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT1+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT1)+LY1*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT1+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT1)+LZ1*((cos(y_ang)*
cos(x_ang))*yT1-(cos(y_ang)*sin(x_ang))*zT1))/L1;
J26=(LX2*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT2+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT2)+LY2*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT2+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT2)+LZ2*
((cos(y_ang)*cos(x_ang))*yT2-(cos(y_ang)*sin(x_ang))*
zT2))/L2;
J36=(LX3*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT3+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT3)+LY3*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT3+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT3)+LZ3*((cos(y_ang)*
cos(x_ang))*yT3-(cos(y_ang)*sin(x_ang))*zT3))/L3;
J46=(LX4*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT4+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)
*cos(x_ang))*zT4)+LY4*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT4+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT4)+LZ4*
((cos(y_ang)*cos(x_ang))*yT4-(cos(y_ang)*sin(x_ang))*
zT4))/L4;
J56=(LX5*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT5+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT5)+LY5*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT5+(-sin(z_ang)*sin(y_ang)*
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT5)+LZ5*((cos(y_ang)*
cos(x_ang))*yT5-(cos(y_ang)*sin(x_ang))*zT5))/L5;
J66=(LX6*((cos(z_ang)*sin(y_ang)*cos(x_ang)+sin(z_ang)*sin
(x_ang))*yT6+(-cos(z_ang)*sin(y_ang)*sin(x_ang)+sin(z_ang)*
cos(x_ang))*zT6)+LY6*((sin(z_ang)*sin(y_ang)*cos(x_ang)-
cos(z_ang)*sin(x_ang))*yT6+(-sin(z_ang)*sin(y_ang)*
```

```
sin(x_ang)-cos(z_ang)*cos(x_ang))*zT6)+LZ6*
((cos(y_ang)*cos(x_ang))*yT6-(cos(y_ang)*sin(x_ang))*
zT6))/L6;

J=[J11 J12 J13 J14 J15 J16;J21 J22 J23 J24 J25 J26;J31 J32
J33 J34 J35 J36;J41 J42 J43 J44 J45 J46;J51 J52 J53 J54 J55
J56;J61 J62 J63 J64 J65 J66];

Z= cond(J,'fro');


End
```

Üç değişkenli optimizasyonda koşul sayılarının hesaplanması için kullanılan fonksiyon [Z], iki değişkenli için kullanılan fonksiyon [Z] ile aynıdır.

Tek değişkenli optimizasyonda maliyet fonksiyonunun hesaplanması

```
function Cost = optimizationfunctionz(radius_t)

P = [ ...
      0.06 0.06 0.4
     -0.06 0.06 0.4
     -0.06 -0.06 0.4
     0.06 -0.06 0.4
     0.06 0.06 0.3
     -0.06 0.06 0.3
     -0.06 -0.06 0.3
     0.06 -0.06 0.3
     0.02 0.02 0.38
     -0.02 0.02 0.38
     -0.02 -0.02 0.38
     0.02 -0.02 0.38
     0.02 0.02 0.32
     -0.02 0.02 0.32
     -0.02 -0.02 0.32
     0.02 -0.02 0.32

     ];
A = [ ...
     5 5 5
     -5 5 5
     5 -5 5
     5 5 -5
     -5 -5 5
     5 -5 -5
     -5 5 -5
     -5 -5 -5
     2.5 2.5 2.5
     -2.5 2.5 2.5
     2.5 -2.5 2.5
     2.5 2.5 -2.5
     -2.5 -2.5 2.5
     2.5 -2.5 -2.5
```

```
    -2.5 2.5 -2.5
    -2.5 -2.5 -2.5
    ];
subCost = zeros(256,1);
for i=1:size(P,1)
    for j=1:size(A,1)
        px = P(i,1);
        py = P(i,2);
        pz = P(i,3);
        x_ang = A(i,1)*pi/180;
        y_ang = A(i,2)*pi/180;
        z_ang = A(i,3)*pi/180;
        [Z] = condition_number_optz(...
            radius_t, ...
            px,...
            py,...
            pz,...
            x_ang,...
            y_ang,...
            z_ang);

        if (Z==0)
            subCost((i-1)*16+j) = 1e6;
        else
            subCost((i-1)*16+j) = radius_t * (Z-1);
        end
    end
end

Cost = sum(subCost);
end
```

İki değişkenli optimizasyonda maliyet fonksiyonunun hesaplanması

```
function Cost = optimizationfunctionz(x)

P = [ ...
    0.06 0.06 0.4
    -0.06 0.06 0.4
    -0.06 -0.06 0.4
    0.06 -0.06 0.4
    0.06 0.06 0.3
    -0.06 0.06 0.3
    -0.06 -0.06 0.3
    0.06 -0.06 0.3
     0.02 0.02 0.38
    -0.02 0.02 0.38
    -0.02 -0.02 0.38
    0.02 -0.02 0.38
    0.02 0.02 0.32
```

```
    -0.02 0.02 0.32
    -0.02 -0.02 0.32
    0.02 -0.02 0.32
    ];
A = [ ...
    5 5 5
    -5 5 5
    5 -5 5
    5 5 -5
    -5 -5 5
    5 -5 -5
    -5 5 -5
    -5 -5 -5
    2.5 2.5 2.5
    -2.5 2.5 2.5
    2.5 -2.5 2.5
    2.5 2.5 -2.5
    -2.5 -2.5 2.5
    2.5 -2.5 -2.5
    -2.5 2.5 -2.5
    -2.5 -2.5 -2.5
    ];
subCost = zeros(256,1);
for i=1:size(P,1)
    for j=1:size(A,1)
        px = P(i,1);
        py = P(i,2);
        pz = P(i,3);
        x_ang = A(i,1)*pi/180;
        y_ang = A(i,2)*pi/180;
        z_ang = A(i,3)*pi/180;
        [Z] = condition_number_optz(...
            x,...
            px,...
            py,...
            pz,...
            x_ang,...
            y_ang,...
            z_ang);

        if (Z==0)
            subCost((i-1)*16+j) = 1e6;
        else
            subCost((i-1)*16+j) = (x(1)* x(2)) * (Z-1);
        end
    end
end

Cost = sum(subCost);
End
```

Üç değişkenli optimizasyonda maliyet fonksiyonunun hesaplanması

```
function Cost = optimizationfunctionz(x)


P = [ ...
    0.06 0.06 0.4
    -0.06 0.06 0.4
    -0.06 -0.06 0.4
    0.06 -0.06 0.4
    0.06 0.06 0.3
    -0.06 0.06 0.3
    -0.06 -0.06 0.3
    0.06 -0.06 0.3
     0.02 0.02 0.38
    -0.02 0.02 0.38
    -0.02 -0.02 0.38
    0.02 -0.02 0.38
    0.02 0.02 0.32
    -0.02 0.02 0.32
    -0.02 -0.02 0.32
    0.02 -0.02 0.32
    ];
A = [ ...
    5 5 5
    -5 5 5
    5 -5 5
    5 5 -5
    -5 -5 5
    5 -5 -5
    -5 5 -5
    -5 -5 -5
    2.5 2.5 2.5
    -2.5 2.5 2.5
    2.5 -2.5 2.5
    2.5 2.5 -2.5
    -2.5 -2.5 2.5
    2.5 -2.5 -2.5
    -2.5 2.5 -2.5
    -2.5 -2.5 -2.5
    ];
subCost = zeros(256,1);
for i=1:size(P,1)
    for j=1:size(A,1)
        px = P(i,1);
        py = P(i,2);
        pz = P(i,3);
        x_ang = A(i,1)*pi/180;
        y_ang = A(i,2)*pi/180;
```

```
        z_ang = A(i,3)*pi/180;
        [Z] = condition_number_optz(...
            x,...
            px,...
            py,...
            pz,...
            x_ang,...
            y_ang,...
            z_ang);

        if (Z==0)
            subCost((i-1)*16+j) = 1e6;
        else
            subCost((i-1)*16+j)    =    ((x(1)*    x(2))    *
((x(3)+x(4)+x(5)+x(6)+x(7)+x(8))/6)*(Z-1));
        end
    end
end

Cost = sum(subCost);
end
```

EK-C

Tek değişkenli optimizasyonda kısıtların yazılması

```
function [c,ceq]=constrains(radius_t)
c=[radius_t-0.175;
-radius_t+0.070];
 ceq=[];
```

İki değişkenli optimizasyonda kısıtların yazılması

```
function [c,ceq]=constrains(x)
c=[x(1)-0.125;
-x(1)+0.070;
x(2)-0.175;
-x(2)+0.125;]; ceq=[];
```

Üç değişkenli optimizasyonda kısıtların yazılması

```
function [c,ceq]=constrains(x)
c=[x(1)-0.125;
-x(1)+0.070;
x(2)-0.175;
-x(2)+0.125;
-x(3)+0.300;
x(3)-0.450;
-x(4)+0.300;
x(4)-0.450;
-x(5)+0.300;
x(5)-0.450;
-x(6)+0.300;
x(6)-0.450;
-x(7)+0.300;
x(7)-0.450;
-x(8)+0.300;
x(8)-0.450;]; ceq=[];
```

Maksimum koşul sayısının bulunması için matlab kodları

```
%Maksimum Z'in tanımlanması ve indexlenmesi arzu edilen
indisin grafinin çizilmesi

figure(13)

Threshold = 1000;
index = find(Z<Threshold);
[K1 K2 K3 K4 K5 K6] = ind2sub(size(Z),index);
K = [K1 K2 K3 K4 K5 K6];

fprintf('Total Condition Number Calculated : %d\n',
numel(Z));
fprintf('Total Number of Condition Number Smaller then %.2f
is %d', Threshold, size(K,1));

kprime =4400;
k1 = K(kprime,1);
k2 = K(kprime,2);
k3 = K(kprime,3);
k4 = K(kprime,4);
k5 = K(kprime,5);
k6 = K(kprime,6);

fprintf('\npx = %f\n',px(k1));
fprintf('py = %f\n',py(k2));
fprintf('pz = %f\n',pz(k3));
fprintf('x_ang = %f\n',x_ang(k4));
fprintf('y_ang = %f\n',y_ang(k5));
fprintf('z_ang= %f\n',z_ang(k6));

fprintf('Condition Number : %f\n', Z(k1,k2,k3,k4,k5,k6))
fprintf('Maximum Condition Number : %g',
max(max(max(max(max(max(Z)))))));
px_ = px(k1);
py_ = py(k2);
pz_ = pz(k3);
```

```
x_ang_ = x_ang(k4);
y_ang_ = y_ang(k5);
z_ang_ = z_ang(k6);

tolerance = 1e-3;

k1 = find(abs(px-px_)<tolerance);
k2 = find(abs(py-py_)<tolerance);
k3 = find(abs(pz-pz_)<tolerance);
k4 = find(abs(x_ang-x_ang_)<tolerance);
k5 = find(abs(y_ang-y_ang_)<tolerance);
k6 = find(abs(z_ang-z_ang_)<tolerance);

V = Z(k1,k2,k3,:,:,:);
V = reshape(V,length(x_ang),length(y_ang),length(z_ang));
p1 = x_ang_;
p2 = y_ang_;
p3 = z_ang_;

slice(V,p1,p2,p3);
colorbar

xlabel(['x_ang (' num2str(p1) ')']);
ylabel(['y_ang (' num2str(p2) ')']);
zlabel(['z_ang (' num2str(p3) ')']);

%Çalışma uzayının tanımlanması ve ilgili koordinatlar icin
kosul sayısı alt programın çağrılması,koşul sayılarının
büyükten küçüğe sıralanması

px = (-0.06:0.04:0.06)';
py = (-0.06:0.04:0.06)';
pz = (0.3:0.02:0.4)';
x_ang = (-5*pi/180:2.5*pi/180:5*pi/180)';
y_ang = (-5*pi/180:2.5*pi/180:5*pi/180)';
z_ang = (-5*pi/180:2.5*pi/180:5*pi/180)';

Z                                                                =
zeros(length(px),length(py),length(pz),length(x_ang),length
(y_ang),length(z_ang));
TotalCalculations = numel(Z);
figure;hold;
t=0;
k = 0;
for k1=1:length(px)
    for k2=1:length(py)
        for k3=1:length(pz)
            for k4=1:length(x_ang)
                for k5=1:length(y_ang)
                    for k6=1:length(z_ang),
```

```
                        t=t+1;
                        Z(k1,k2,k3,k4,k5,k6)
= condition_number(...
                            px(k1),...
                            py(k2),...
                            pz(k3),...
                            x_ang(k4),...
                            y_ang(k5),...
                            z_ang(k6));

plot(t,Z(k1,k2,k3,k4,k5,k6),'.');max Z
                        k = k+1;

fprintf('\n%f',k/TotalCalculations*100)
                    end
                end
            end
        end
    end
end

disp(Z);
M = reshape(Z,k,1);
disp(M);
Y=sort(M,'descend');
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Cenk ERYILMAZ

Doğum Tarihi ve Yeri : 17.12.1980-Üsküdar

Yabancı Dili : İngilizce

E-posta : cenk.eryilmaz@tubitak.gov.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Makine Müh./İmalat	Yıldız Teknik Üniversitesi	2004
Lisans	Makine Müh.	Yıldız Teknik Üniversitesi	2002
Lise	Elektronik	Tuzla Anadolu Teknik Lisesi	1998

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2008	Tübitak	Mekanik Tasarım Uzmanı
2005	Festo	Proje Mühendisi
2002	Pemaks	Ar-Ge Mühendisi

YAYINLARI

Makale

1. Eryilmaz, C., Omurlu, V.E., (2019). “SQP Optimization of 6dof 3x3 UPU Parallel Robotic System for Singularity Free and Maximized Reachable Workspace”, Journal of Robotics, volume 2019, Article ID 3928705, 14 pages.

Bildiri

1. Eryilmaz, C., Omurlu, V.E., (2016). “Multi-Objective SQP Optimization of a Parallel Robotic System for Singular Free and Maximized Reachable Workspace”, International Conference on Advances in Materials and Processing Technologies (AMPT 2016), paper no: 072.