



T.C.

ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇARPIŞMASIZ VE KİLİTLENMESİZ ÇOK-ROBOTLU MONTAJ HATTI
DENGELEME**

İsmail Enes PARLAK

Prof. Dr. Erdal Emel

(Danışman)

YÜKSEK LİSANS TEZİ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA, 2017

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

İsmail Enes PARLAK

TEZ ONAYI

İsmail Enes PARLAK tarafından hazırlanan “Çarpışmasız ve Kilitlenmesiz Çok-robotlu Montaj Hattı Dengeleme” adlı tez çalışması aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Prof. Dr. Erdal EMEL

Başkan : Prof. Dr. Erdal EMEL

Uludağ Ü. Mühendislik Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Ali Yurdun ORBAK

Uludağ Ü. Mühendislik Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

Üye : Dr. Öğr. Üyesi Aytaç YILDIZ

Bursa Teknik Ü. Mühendislik ve Doğa Bilimleri Fakültesi,
Endüstri Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Ali BAYRAM

Enstitü Müdürü

6 / 7 / 2018

ÖZET

Yüksek Lisans Tezi

ÇARPIŞMASIZ VE KİLİTLENMESİZ ÇOK-ROBOTLU MONTAJ HATTI DENGELEME

İsmail Enes PARLAK

Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erdal EMEL

Endüstriyel otomasyonun en çarpıcı uygulamalarından biri olan, robotik montaj hatları günümüzde giderek artan oranlarda manuel sistemlere tercih edilmektedir. Üstelik, ürün çıktısını artırmak için montaj istasyonlarında birlikte ve koordineli çalışan robotlar, akıllı sistemlere benzetilmeye çalışılmaktadır. Bir iş istasyonunda birden fazla robotun aynı iş üzerinde çalışması, doğal olarak çarpışma tehlikesi sorunu ortaya çıkarmaktadır. Bu tez çalışmasında, çarpışmasız ve kilitlenmesiz çok robotlu montaj hattı dengeleme problemi için iki aşamalı bir çözüm yöntemi önerilmiştir. Montaj hattında yer alan istasyonlarda, karşılıklı ikiye adet robotik manipülatör çalıştığı öngörülmüştür. Problemin en önemli kısıtlarından biri olan çarpışma engelleme durumu için literatürde yer alan kullanışlı bir yöntem benimsenmiştir. Çarpışma durumu ortadan kalktığında, montaj hattında yer alan tek tip ürün için öncül-ardıl ilişkilerine göre görevler robotik manipülatörlere, rastgele dizi sıralama yöntemiyle atanmıştır. Bunun yanı sıra farklı çevrim süreleri için dengeleme problemi çözülmüş ve açılan istasyon sayılarındaki değişim ile beraber istasyonlarda oluşan aylak zamanlar hesaplanmıştır. Elde edilen sonuçlar yöntemin, çarpışma kısıtı olan çok robotlu montaj hattı dengeleme problemi için oldukça kullanışlı olduğunu göstermektedir.

Anahtar Kelimeler: Çarpışmasız yörünge planı, robotik manipülatörler, montaj hattı dengeleme problemi, çok-işçili montaj hatları

ABSTRACT

MSc Thesis

COLLISION-FREE AND DEADLOCK-FREE MULTI-ROBOT ASSEMBLY LINE BALANCING

İsmail Enes PARLAK

Uludag University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineering

Supervisor: Prof. Dr. Erdal EMEL

Robotic assembly lines, one of the most striking applications of industrial automation, are now increasingly preferred to manual systems. Moreover, joint and coordinated robots in assembly workstations to increase product output are being tried to simulate intelligent systems. When multiple robots operate on the same job in a workstation, the problem of collision risk naturally arises. In this thesis study, a two-step method is proposed for a collision-free and deadlock-free multi-robotic assembly line balancing problem. Two robotic manipulators are assumed to operate in each of the stations on the assembly line. To operate under collision-free conditions, which is one of the most important constraints of the problem, a useful method has been adapted from the literature. The problem of simultaneous search for a collision-free task assignment while keeping a balanced workload in a single model assembly line under precedence constraints has been solved using a randomly permuting arrays method. The problem is solved for different cycle times to calculate idle times in the stations and number of workstations. The results indicate that the method is highly useful for multi-robot assembly line balancing problem under collision-free constraints.

Keywords: Collision-free trajectory planning, robotic manipulators, assembly line balancing problem, multi-manned assembly lines

TEŐEKKÜR

Yüksek Lisans eğitiminin boyunca ve bu tez çalışmanın gerçekleştirilmesinde; bilgi ve tecrübeleriyle beni yönlendiren, sabır ve hoşgörüyüyle çalışmamı bilimsel temeller ışığında yürütmemi sağlayan ve desteğini benden esirgemeyen değerli ve saygı değer hocam Prof. Dr. Erdal EMEL'e, yüksek lisans eğitiminin boyunca sağladıkları katkılarında dolayı bölümdeki tüm hocalarıma, Bursa Teknik Üniversitesi'nde beraber çalıştığım ve bana her zaman destek olan saygı değer hocam Dr. Öğr. Üyesi Aytaç YILDIZ'a, bu tez çalışması boyunca beni destekleyen aileme ve arkadaşım Murat'a sonsuz şükran ve teşekkürlerimi sunarım.

İsmail Enes PARLAK

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
KISALTMALAR.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI.....	3
2.1. Robotik Montaj Hatları Dengeleme Problemi.....	3
2.2. Çok-adamlı İş İstasyonları.....	6
2.3. Robotik Manipülatörlerin Yörünge Planı ve Çarpışma-Kilitlenme Engelleme.....	9
3. MATERYAL VE YÖNTEM.....	19
3.1. Materyal.....	19
3.1.1. Robotik manipülatörlerin bağlantı elemanları.....	19
3.1.2. Robotik manipülatörlerin eklemleri.....	21
3.1.3. Genel koordinat sistemi kinematik modelleri.....	25
3.1.4. Denavit-Hartenberg modeli.....	27
3.1.5. İleri ve ters kinematik.....	29
3.1.6. Puma 560 robotik manipülatörünün kinematik modeli.....	30
3.1.7. Çarpışmasız yörünge planlama.....	33
3.1.8. Çok-işçili montaj hattı dengeleme probleminin matematiksel modeli.....	38
3.2. Yöntem.....	45
3.2.1. Dizi elemanlarını rastgele sıralama yöntemi.....	45
3.2.2. Görev atama ve uygunluk fonksiyonu hesabı.....	46
4. BULGULAR.....	55
4.1. Uygulama.....	55
4.1.1. Uygulama örneği.....	55
4.1.2. Sonuçlar.....	59
4.2. Bulgular.....	67
5. TARTIŞMA VE SONUÇ.....	68
KAYNAKLAR.....	69

EKLER.....	74
EK 1 Örnek Problem için Görev Koordinatları	75
EK 1 (devamı) Örnek Problem için Görev Koordinatları.....	76
EK 1 (devamı) Örnek Problem için Görev Koordinatları.....	77
EK 2 MATLAB Kodu.....	78
EK 3 İstasyon Süreleri (Çevrim Süresi 70 sn)	89
EK 4 Robotik Manipulatörlerin Çalışma Süreleri (70 sn)	90
EK 5 İstasyon Süreleri (Çevrim Süresi 50 sn)	92
EK 6 İstasyon Süreleri (Çevrim Süresi 60 sn)	93
EK 7 İstasyon Süreleri (Çevrim Süresi 80 sn)	94
EK 8 İstasyon Süreleri (Çevrim Süresi 90 sn)	95
ÖZGEÇMİŞ	96

KISALTMALAR

Kisaltmalar	Açıklama
RMHDP	Robotik Montaj Hattı Dengeleme Problemi
MHD	Montaj Hattı Dengeleme
BMHDP	Basit Montaj Hattı Dengeleme Problemi
BMHDP-I	Tip 1 Basit Montaj Hattı Dengeleme Problemi
BMHDP-II	Tip 2 Basit Montaj Hattı Dengeleme Problemi
GA	Genetik Algoritma
PSO	Parçacık Sürü Optimizasyonu
EA	Evrimsel Algoritma
MIP	Mixed Integer Programming (Karışık Tamsayılı Programlama)
BBA	Branch-and-bound Algorithm (Dal-sınır Algoritması)
ÇMHDP	Çok-İşçili Montaj Hattı Dengeleme Problemi
NP	Non-deterministic Polynomial-time (Deterministik Olmayan Polinom Zaman)
ACO	Ant Colony Optimization (Karıncı Koloni Optimizasyonu)
CPU	Central Processing Unit
SA	Simulated Annealing (Tavlama Benzetimi)
PTP	Point-to-point (Noktadan Noktaya)
DH	Denavit-Hartenberg
RPA	Randomly Permuting Array (Rastgele Dizi Sıralama)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Montaj hattı dengeleme problemlerinin sınıflandırılması	4
Şekil 2.2. Robotik montaj hattı	5
Şekil 2.3. SCARA robotu.....	11
Şekil 2.4. Robot koordinasyon yöntemlerinin sınıflandırılması	14
Şekil 3.1. Bir uzamsal bağlantının iki farklı görünümü	20
Şekil 3.2. Kinematik bağlantı.....	20
Şekil 3.3. Küresel bağlantı	21
Şekil 3.4. Düzlemsel bağlantı	21
Şekil 3.5. Dönel eklem.....	22
Şekil 3.6. Prizmatik eklem	23
Şekil 3.7. Silindirik eklem.....	23
Şekil 3.8. Küresel eklem	24
Şekil 3.9. İki keyfi kartezyen koordinat sistemi.....	26
Şekil 3.10. Eklemler için Denavit-Hartenberg parametreleri.....	29
Şekil 3.11. PUMA 560 robotik manipülatörü	31
Şekil 3.12. İki bağlantılı iki düzlemsel robot için çalışma uzayı	35
Şekil 3.13. Bir robotik manipülatöre ait tutucu uç elemanın çalışma uzayında izlediği yörüngenin noktalar ile gösterimi	35
Şekil 3.14. Robotik manipülatörlerin birbirine değen daireler ile gösterimi	36
Şekil 3.15. Robot kollarına ait yörüngeler	36
Şekil 3.16. Robot kollarına ait tutucu uç elemanların yörüngeleri	37
Şekil 3.17. PUMA-benzeri robotik manipülatör	38
Şekil 3.18. Robotik manipülatörlere ait bağlantı elemanları.....	38
Şekil 3.19. Çok-işçili montaj hattı yerleşimi.....	39
Şekil 3.20. Çok-işçili bir montaj hattı örneğine ait öncüllük ilişkileri.....	40
Şekil 3.21. Basit montaj hattı için görev atamaları	40
Şekil 3.22. Çok-işçili montaj hattı için görev atamaları	40
Şekil 3.23. Rastgele sıralama yöntemine ait yalancı kod.....	46
Şekil 3.24. Bir istasyonda atanan görevler ve aylak süreleri gösteren çizelge	50
Şekil 3.25. Çakışma kontrolüne ait sözde kod	51
Şekil 3.26. Çakışmasız montaj hattı dengeleme yöntemine ait sözde kod.....	53
Şekil 3.27. Çakışmasız montaj hattı dengeleme yönteminin akış diyagramı.....	54
Şekil 4.1. İki robotik manipülatörün yer aldığı benzetimler	56
Şekil 4.2. Tutucu uç elemanın bir noktaya farklı şekillerde ulaşma durumu.....	57
Şekil 4.3. Öncül-ardıl ilişkileri diyagramı	58
Şekil 4.4. İstasyon 1 için çizelge (çevrim süresi 70 sn)	60
Şekil 4.5. İstasyon 2 için çizelge (çevrim süresi 70 sn)	60
Şekil 4.6. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 70 sn)	60
Şekil 4.7. İstasyon 1 için çizelge (çevrim süresi 50 sn)	61
Şekil 4.8. İstasyon 2 için çizelge (çevrim süresi 50 sn)	61
Şekil 4.9. İstasyon 3 için çizelge (çevrim süresi 50 sn)	62

Şekil 4.10. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 50 sn)	62
Şekil 4.11. İstasyon 1 için çizelge (çevrim süresi 60 sn)	63
Şekil 4.12. İstasyon 2 için çizelge (çevrim süresi 60 sn)	63
Şekil 4.13. İstasyon 3 için çizelge (çevrim süresi 60 sn)	63
Şekil 4.14. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 60 sn)	64
Şekil 4.15. İstasyon 1 için çizelge (çevrim süresi 80 sn)	65
Şekil 4.16. İstasyon 2 için çizelge (çevrim süresi 80 sn)	65
Şekil 4.17. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 80 sn)	65
Şekil 4.18. İstasyon 1 için çizelge (çevrim süresi 90 sn)	66
Şekil 4.19. İstasyon 2 için çizelge (çevrim süresi 90 sn)	66
Şekil 4.20. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 90 sn)	66

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 4.1. Robotik manipülatöre ait eklemlerin açısız erişimi	56
Çizelge 4.2. Robotik manipülatörün DH parametreleri (Jha 2015)	57



1. GİRİŞ

Dördüncü Endüstri Devrimi olarak da tanımlanan Endüstri 4.0 ile birlikte robotlar üretimde önemli bir rol üstlenmektedir. Otomatik iş yapan makineler olarak robotlar, mevcut durumda sanayi üretiminde önemli bir yere sahiptir. Bunun yanı sıra üretimde verimliliği ve kaliteyi arttırmak amacıyla üretim sitemlerinde robot kullanımını hızla yaygınlaştırmaktadır. Robotların en çok kullanıldığı alanlardan biri montaj hatlarıdır. Montaj hatlarında, giderek artan oranlarda robotlar manuel işgücünün yerini almaktadır (Gao ve ark. 2009). Ayrıca talep çeşitliliğinin artmasından dolayı, daha esnek ve verimli sistemlere gereksinim duyulmaktadır. Bu bağlamda, sistemin esnekliğini ve verimliliğini önemli ölçüde artırabilen robotların kullanım avantajları şu şekilde özetlenebilir:

- Robotlar insanlara göre daha kaliteli ve kusursuz işlem yapabilir.
- Daha kısa sürede daha fazla miktarda üretim yapabilir.
- Ara vermeden 7/24 sabit bir hızda çalışabilir.
- Tehlikeli görevleri insanların yerine yaparak iş kazalarının azalmasını sağlar.
- İnsan için riskli olan çalışma ortamlarında kolaylıkla çalışabilir.
- Ağır yükleri kolaylıkla ve güvenli bir şekilde taşıyabilir.
- Daha geniş ürün yelpazesinde üretim yapabilmelerinden dolayı zaman kazandırır.
- Iskartaya çıkan malzeme sayısını daha aza indirgeyebilir.
- İşçilik maliyetlerinin azalmasını sağlar.
- Daha düşük üretim maliyeti sağlar.
- Herhangi bir iş için özel amaçlı olarak tasarlanabilir ve bu yüzden çok farklı alanlarda kolaylıkla kullanılabilir.

Bazı montaj hatlarında daha hızlı ve çok sayıda üretim yapabilmek için bir istasyonda birden fazla işçi çalışabilmektedir. Robotik hatlarda da benzer şekilde birden fazla robot beraber çalışabilmektedir. Ancak robotların aynı ortamda birlikte çalışması bazı sorunlar ortaya çıkarmaktadır. Bu sorunlardan en önemlisi, robotların çalışma uzayında doğal olarak çarpışma ve kilitlenme potansiyellerinin olmasıdır. Çoklu robot montaj

hattında, sürekliliği sağlamak için çarpışma ve kilitlenmelerin giderilmesi oldukça kritik bir öneme sahiptir.

Bu çalışmada, punta kaynağı yapılan çok robotlu iş istasyonlarında çarpışmaları ve kilitlenmeleri önleyecek şekilde robotik manipülatörlere dengeli bir şekilde görev atamayı sağlayan bir yöntem önerilmektedir. Literatürde, robotların çarpışması ile ilgili birçok çalışma mevcuttur. Ancak bu problem, görev atamalarının yapıldığı bir montaj hattında daha önce dikkate alınmamıştır. Çarpışmaların önlenmesinin amaçlandığı çalışmalarda robotlar için genelde yörünge planlaması yapılmakla birlikte yörünge planlamasının göz ardı edildiği ve yalnızca çakışmaları engellemeye yönelik çalışmalar da mevcuttur. Bu tez çalışmasını literatürde yer alan diğer çalışmalardan ayıran en önemli özellik, montaj hattındaki iş istasyonlarında birlikte çalışması gereken robotik manipülatörlere çarpışma olmadan dengeli iş atamasının yapılmasıdır.

2. KAYNAK ARAŞTIRMASI

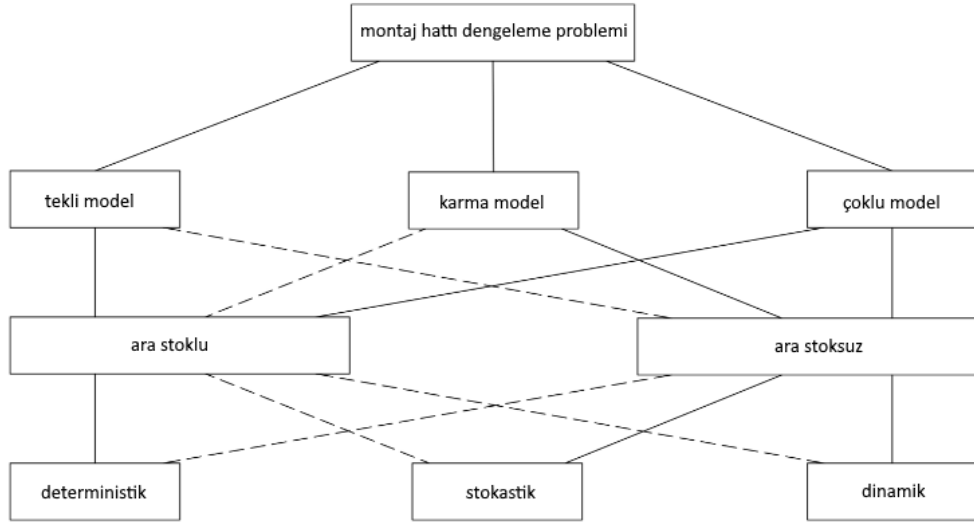
Bu bölüm sırasıyla üç alt başlık altında incelenmiştir: Robotik montaj hattı dengeleme problemi, çok-adamlı iş istasyonları ve robotik manipülatörlerin yörünge planı ve çarpışmalarının-kilitlenmelerinin engellenmesi.

2.1. Robotik Montaj Hatları Dengeleme Problemi

Montaj hatları yüksek miktarlarda standartlaşmış ürünler üretmek için seri üretim sistemlerinde yaygın olarak kullanılmaktadır. Montaj hatları bir dizi seri bağlı iş istasyonundan ve iş istasyonlarını bağlayan bir malzeme taşıma sisteminden oluşurlar. Tekrarlayan montaj işlemleri (görevler), ürün hat boyunca akarken iş istasyonlarında gerçekleştirilir. Becker ve Scholl (2006) montaj hatlarını; tek modelli, çok modelli ve karma modelli olmak üzere üç tip olarak sınıflandırmaktadır.

Birçok montaj hattı tipi var olmakla birlikte, klasik, otomatik ve yalın üretim modelleri içeren çeşitleri bulunmaktadır. Bu montaj hattı sistemleri genellikle farklı tipte ürünleri işlemek için kullanılır. Şekil 2.1'de, montaj hattı dengeleme problemlerinin bir sınıflandırması verilmektedir.

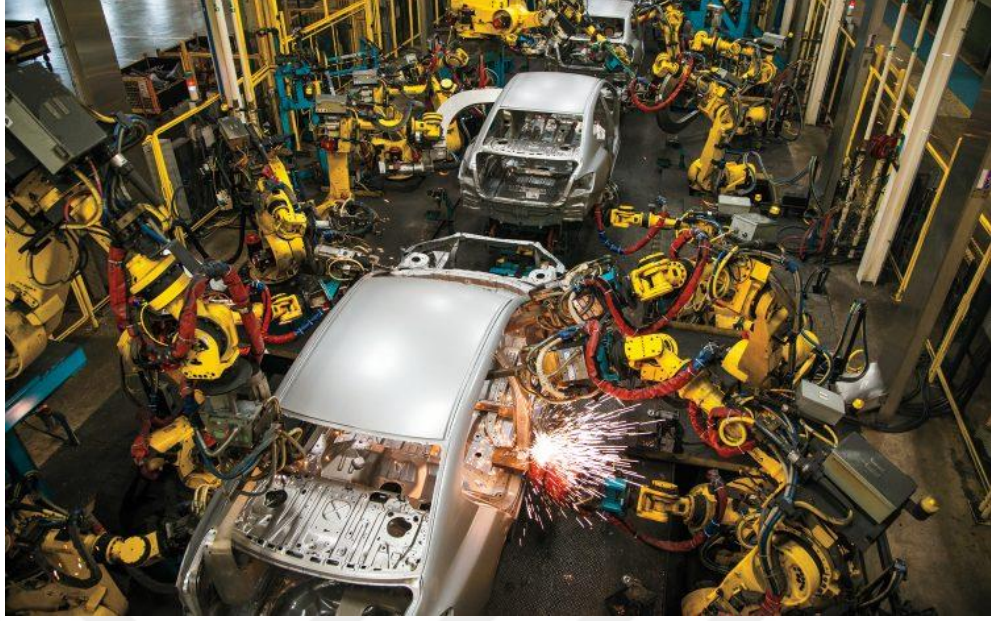
Montaj hattı dengeleme problemlerinde (MHDP) tek modelli ürünün üretimi söz konusuysa bunlara basit montaj hattı dengeleme problemleri (BMHD) denir. Literatürde BMHDP, tasarım ve operasyonel düzeylere ilişkin bazı kararlar vermek amacıyla, sırasıyla BMHDP-I (tip 1) ve BMHDP-II (tip 2) olarak ele alınmıştır. BMHDP-I, önceden belirlenmiş bir çevrim süresi kullanarak hattaki iş istasyonlarının sayısının en aza indirilmesi ile ilgili iken, BMHDP-II ise, önceden belirlenmiş iş istasyonu sayısı ile çevrim süresini en küçüklemek ile ilgilidir.



Şekil 2.1. Montaj hattı dengeleme problemlerinin sınıflandırılması (Scholl 1999)

Modern sanayide, insan emeğinin insan kaynaklarındaki artan maliyetlerinden dolayı robotlar ile değiştirilme eğilimi vardır. Maliyet azaltma avantajlarından ayrı olarak, robotlar daha esnek olabilirler ve farklı görevleri yerine getirmek için programlanabilirler. Robotların kullanıldığı montaj hatlarına robotik montaj hatları (RMH) denilmektedir. Şekil 2.2’de bir robotik montaj hattı gösterilmiştir. Robotik montaj hattı dengeleme (RMHD) problemi, hattın verimliliğini artırmak için istasyonlara ve her istasyon için mevcut robota görevler vermektir (Gao ve ark. 2009).

Montaj hattı dengeleme probleminin matematiksel formülü ilk olarak Salveson (1955) tarafından ortaya atılmıştır. Daha sonrasında ise problemi çözmek için çeşitli optimizasyon yöntemleri ve metasezgiseller geliştirilmiştir. RMHD probleminin ilk olarak ele alınışı ise 90’lı yılların başlarına dayanmaktadır. RMHD problemi ilk defa Rubinovitz ve ark. (1993) tarafından tanımlanmıştır. Rubinovitz ve arkadaşları bu çalışmada problemi ayrıntılı olarak ele alıp, üretim ortamlarında robotların avantajlarını incelemişlerdir. Problemin çözümü için dal sınır algoritmasına dayanan sınır arama yöntemi kullanmışlardır.



Şekil 2.2. Robotik montaj hattı (Berman 2016)

Kim ve Park (1995), parçaların ve araçların tahsisini düşünerek aynı problem için kesme düzlemi algoritmasını önermişlerdir. Genetik algoritma (GA), hibrit GA ve parçacık sürü optimizasyonu (PSO), düz hatlarda RMHD-II problemini çözecek şekilde sırasıyla Levitin ve ark. (2006), Gao ve ark. (2009) ve Nilakantan ve Ponnambalam (2012) tarafından geliştirilmiştir. Yoosefelahi ve ark. (2012) düz RMHD problemini üç amaç için incelemişlerdir: çevrim süresinin, robot kurulum maliyetinin ve toplam robot maliyetinin en aza indirilmesi. Yazarlar problemi çözmek için yeni bir tam sayılı doğrusal programlama modeli ve evrimsel algoritma (EA) sunmuşlardır. Daoud ve ark. (2014), düz RMHD probleminin endüstriyel bir durumunu ele almışlardır. Hat, belirli toplama ve yerleştirme robotlarını içermektedir. Sistemin hat verimliliğini en üst düzeye çıkarmak ve her bir robot için bileşenlerin ve konum noktalarının en iyi ve dengeli dağılımını bulmak için birçok meta-sezgisel algoritma sunulmuştur. Nilakantan ve Ponnambalam (2016), literatürde ilk kez çevrim süresini en aza indirmek için U-şekilli RMHD problemini ele almışlardır. Aynı problem için, Nilakantan ve ark. (2016_2) çevrim süresini ve toplam enerji tüketimini en aza indirilmesini bir amaç fonksiyonu içinde bir araya getiren bir EA sunmuştur. Çil ve ark. (2017), karma model tipi için RMHD problemini incelemişlerdir. İki matematiksel model sunup, bunları çözüm verimliliği açısından birbirleriyle karşılaştırmışlardır. Ek olarak, problemi çözmek için ışın arama (IA) yaklaşımının çeşitli uyarlamaları geliştirilmiştir. Ayrıca Çil ve ark.

(2017_2), döngü süresini en aza indirmek için bir ardışık ışın arama algoritması geliştirilmişlerdir. Algoritmanın performansı, literatürdeki diğer algoritmalar ile karşılaştırılmıştır. Li ve ark. (2017) iki taraflı robotik montaj hattı dengeleme problemi için ayrık guguk kuşu arama algoritmaları sunmuşlardır.

Son zamanlarda yapılan çalışmalardan biri de Müller ve ark.'nın (2017) hattın üretim oranını maksimuma çıkarmak için stokastik hataların meydana geldiği robotik montaj hatlarında GA sundukları çalışmadır. Zixiang ve ark. (2017) robotik karma model montaj hattının eş zamanlı dengelenmesi ve sıralanması için matematiksel model ve metasezgiseller önermişlerdir.

Literatürdeki tüm bu çalışmalar RMHD probleminin birçok sorununa değinmesine rağmen, bilindiği kadarıyla aynı hatta birden fazla robotun çarpışma olmaksızın senkronize çalışması üzerine herhangi bir çalışma bulunmamaktadır.

2.2. Çok-adamlı İş İstasyonları

Üretimde artan talepler nedeniyle daha kısa sürede ya da daha az istasyon ile daha fazla çıktı alınması amaçlanmaktadır. Bunu sağlamanın yollarından biri de bir iş istasyonunda birden fazla eş zamanlı çalışanın bulunmasıdır.

Montaj hatlarıyla ilgili literatürde çok sayıda çalışma olmasına rağmen çok-adamlı montaj hatları dengeleme (ÇMHD) konusunda yapılan çalışmalar sınırlıdır. Bilindiği kadarıyla çok-adamlı montaj hattı dengeleme probleminin çözümü ile ilgili ilk yöntem Dimitriadis (2006) tarafından önerilmiştir. ÇMHD'yi toplam istasyon sayısını en aza indirmek amacıyla, Hoffmann'ın prosedürünün modifiye edilmesine dayanan bir sezgisel yöntem geliştirmiştir (Hoffmann, 1963). Bu çalışmanın ardından Becker ve Scholl (2009), ÇMHD'yi hattaki toplam işçi sayısını minimuma indirerek iyileştirmişlerdir. Problem, görevleri temsil montaj noktalarını içerir. Bazı görev zamanlarının çevrim süresini aşmasına izin verilir. Problem tanımlandıktan sonra, karma bir tam sayı programlama (MIP) modeli oluşturulmuş ve problem için bazı önemli alt sınırlama denklemleri önerilmiştir.

Scholl ve Klein (1997), basit montaj hattı dengeleme problemlerini çözmek için iyi bilinen SALOME yöntemine dayalı bir dal ve sınır algoritması (BBA) geliştirmişlerdir. Yöntem, istasyonlar üzerine bir dallanma stratejisi kullanır. Her istasyon içinde optimum görev atamasını belirlemek için ise ayrı bir BBA uygulanır. Önerilen BBA'nın, bazı değişikliklerle büyük boyutlu örnekleri çözmek için bir sezgisel olarak kullanılabilceği belirtilmiştir.

Çevikcan ve ark. (2009), yer kısıtlarına sahip bir karma model ÇMHDP göz önünde bulundurmuş ve bunu çözmek için beş fazlı bir sezgisel önermişlerdir. Yatay dengeleme adı verilen ilk aşama, görevleri çevrim zamanlarını aşmadan ve öncelik ilişkilerini ihlal etmeden gruplara ayırır. Dikey dengeleme adı verilen ikinci aşamada, birinci aşama ile aynı kısıtları göz önüne alarak, ortalamadan daha fazla iş yüküne sahip olan grupların bazı görevleri, daha az iş yüküne sahip diğer gruplara aktarılmaktadır. Fiziksel istasyonların oluşturulduğu üçüncü aşama, bazı görev gruplarını birleştirerek her istasyon için işçi sayısını ve atanan görevleri belirler. Günlük talep durumlarına bağlı olarak, model sıralama ve işçi transfer sistemi adı verilen son iki aşama, günlük model sıralama ve günlük çalışan transfer operasyonlarına ayrılmıştır. Chang ve Chang (2010) toplam istasyon sayısının en aza indirildiği bir karma model ÇMHDP'yi göz önünde bulundurmuş ve bu problem için bir matematiksel model önermişlerdir. LINGO yazılımını kullanarak problemi çözmüş ve modelin karmaşıklık derecesini analiz etmişlerdir.

Fattahi ve ark. (2011), ÇMHDP için birincil amaç olarak toplam işçi sayısını ve ikincil amaç olarak çok-adamlı istasyon sayısını en aza indirecek matematiksel bir model geliştirmişlerdir. Daha sonra, orta ve büyük boyutlu örnekleri çözmek için bir karınca koloni optimizasyonu (ACO) önermişlerdir. Önerilen algoritmada, her bir karınca, kısıtlar göz önünde bulundurularak geçerli görevlerden birini geçerli istasyona atamaya çalışır. Yeni istasyonlar ilk olarak azami sayıda işçi ile açılmakta ve diğer işçi sayıları ise rastgele ve Dimitriadis'in yaklaşımının bir karışımı olan yöntemle göre belirlenmektedir (Dimitriadis 2006). Uygun görevler kümesinden bir görev seçmek için karıncaların feromon miktarlarını kullanan karınca kolonisi sistem durumu geçiş kuralı

ve Helgeson ve Brine (1961) tarafından sunulan Sıralanmış Pozisyonel Ağırlıklar uygulanmıştır. Seçilen görev, göreve daha önce başlayabilecek işçiye atanır. Hesaplamalı deneyler yoluyla, önerilen ACO'nun performansı, çeşitli problem örnekleri üzerinde, Dimitriadis (2006) sezgiseli ile karşılaştırılmış ve sezgiselin yüksek performansa sahip olduğu belirtilmiştir. Yazdanparast ve Hajihosseini (2011), görev sürelerinin istasyondaki işçi sayısına bağlı olduğunu varsaymak suretiyle ÇMHDP'yi değerlendirmişlerdir. Fattahi ve arkadaşları (2011) tarafından problemin çözümü için önerilen modele dayanan bir karışık tam sayılı programlama (MIP) formülasyonu geliştirmişlerdir.

Kellegöz ve Toklu (2012) tarafından yapılan çalışmada toplam işçi sayısını en aza indirmek için Atlayıcı (Jumper) adında yeni bir BBA tanıtılmıştır. Becker ve Scholl (2009) tarafından önerilen BBA değiştirilip, Atlayıcı ile karşılaştırılmış ve sonuçlar, Atlayıcı'nın hem CPU süreleri hem de bulunan uygun çözümlerin kalitesi açısından daha iyi bir performans sergilediğini göstermiştir. Kazemi ve Sedighi (2013) tarafından hazırlanan makale, maliyet odaklı bir karma model olan ÇMHDP'yi ele almış ve problemin çözümü için matematiksel bir formülasyon sunulmuştur. Aynı çalışmada, sezgisel olarak problemi çözmek için bir GA ve bir PSO da önerilmiştir. Roshani ve ark. (2013) ÇMHDP'nin çözümü için tavlama benzetimi (SA) sezgiseli önermişlerdir. SA yaklaşımlarında, çözümlerin öncelikli bir görev listesi olarak sunulması ve iki taraflı montaj hattı dengeleme problemleri için (Özcan ve Toklu, 2009) tarafından önerilen yöntem kullanılarak komşu çözümler üretilmektedir.

Sepahi ve Naini (2014), ÇMHDP üzerine çalışıp, modellerindeki iş istasyonlarının sayısını en aza indirmek için, öncelik kurallarına dayalı bazı basit sezgisel yaklaşımlar benimsemişlerdir. Kellegöz ve Toklu (2015), hattaki toplam işçi sayısının minimize edildiği ÇMHDP'yi değerlendirmiş, problemin bir MIP formülasyonunu geliştirmiş ve öncelik kurallarına dayalı bir sezgisel önermişlerdir. GA tabanlı bir geliştirme prosedürü de ayrıca sunulmuştur. Kıyaslama örneklerini dikkate alarak kullanılan hesaplamalı deneyler, sezgisel yöntemlerin verimliliğini kanıtlamıştır.

Yalnızca belirli bir çevrim süresi için işçi sayısını ve iş istasyonlarını en aza indirmekle kalmayıp aynı zamanda çok-adamlı iş istasyonlarındaki iş yükü arasındaki farkı da

azaltan ÇMHDP için bir matematiksel formülasyon Yılmaz ve Yılmaz (2015) tarafından tartışılmıştır. Ayrıca Yılmaz ve Yılmaz (2016), işçilerin hat üzerinde uzmanlaşmış beceri veya ekipmana ihtiyaç duymaları nedeniyle önceden takımlara atandıkları ÇMHDP'yi incelemişlerdir. Problemi çözmek için bir matematiksel formülasyon ve sezgisel çözüm yaklaşımı önermişlerdir.

Kellegöz (2017), birincil hedef olarak işçi sayısını ve ikincil hedef olarak iş istasyonlarının sayısını en aza indirmeyi hedefleyen ÇMHDP için tavlama benzetimine dayalı (Gantt şeması temelli) yeni bir matematiksel formül önermiştir. Roshani ve Giglio (2017) belirli bir sayıda iş istasyonu için çevrim süresini en aza indirmek amacıyla bir ÇMHDP üzerinde çalışmışlardır. Problemi çözmek için yeni bir matematiksel formülasyon ve iki tavlama benzetimi yaklaşımı geliştirmişlerdir.

Tüm bu çalışmalar artan talebi karşılamak amacıyla istasyonlardaki işçi sayısını artırmaya yönelik olup, literatürde çok robotlu montaj hattı dengeleme problemi üzerine bilindiği kadarıyla benzer bir çalışma mevcut değildir. Bu tez çalışmasında ise istasyonlarda insanların yerini robotik manipülatörler almıştır. Burada temel sorun manipülatörlerin çarpışması veya birbirlerini kilitlemeleridir. Bir sonraki alt bölümde robotik manipülatörlerin çarpışmalarının engellenmesi ve kilitletmesinin önlenmesi ile ilgili çalışmalara yer verilmiştir.

2.3. Robotik Manipülatörlerin Yörünge Planı ve Çarpışma-Kilitlenme Engelleme

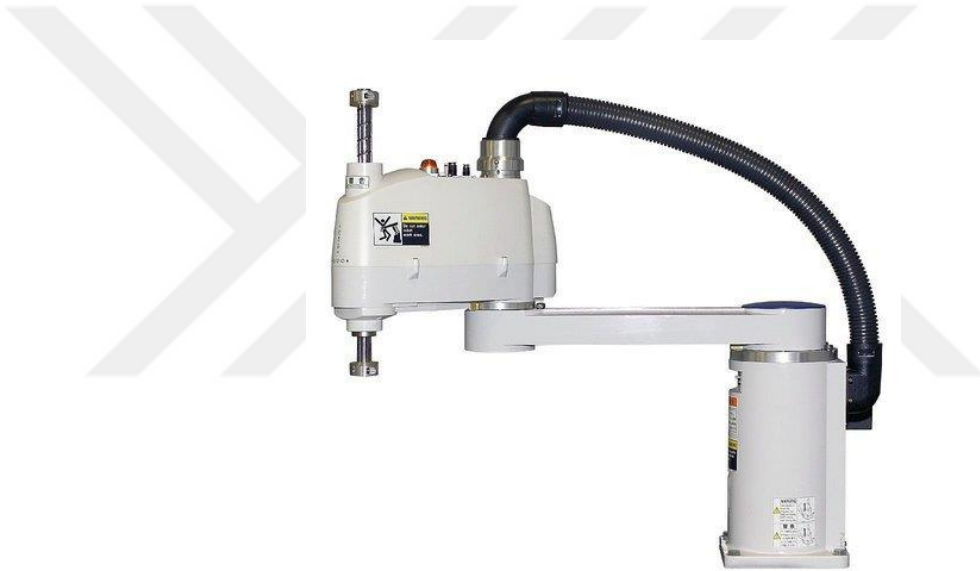
Üretim hatlarında verimliliği artırmak, kalitesizlik maliyetlerini azaltmak ve çıktı miktarını artırmak amacıyla robotlar, giderek daha fazla oranda insanların yerini almaktadır. Robotlar bir üretim alanında taşıma, tutma, kaldırma, paketleme, boyama ve kaynak yapma gibi birçok işlevi yerine getirebilmektedirler. Bu tez çalışmasında, istasyonlarında punta kaynağı yapabilen robotik manipülatörlerin kullanıldığı bir montaj hattı dengeleme problemi ele alınmıştır. Uygulamada istasyonlarda birden fazla robotik manipülatör kullanılabilir. Böylelikle, çevrim zamanı kısaltılıp, hattın kapasitesi artırılabilir. Aynı ortamda birden fazla robot çalıştığı zaman, çarpışmasız hareketin sağlanması ise oldukça kritik bir konudur.

O'Donnell ve Lozano-Perez (1989), birlikte çalışan iki robotik manipulatörün çarpışma ve kilitlenme olmadan yörünge planlamasını yapmak için bir yöntem önermişlerdir. Planlama, yörüngeler üzerinde tanımlanan belirli yönlendirme komutları verilerek yapılmıştır. Rotaların koordinasyonundaki temel problem, iki robot arasındaki çarpışmayı ve kilitlenmeyi önlemek, yani her bir manipulatörün diğerinin devam etmesini beklediği durumları ortadan kaldırmaktır. Her bir robotik manipulatörün yörüngesinin çevrimdışı planlandığı ve bir dizi yol segmentinden oluştuğu varsayılmıştır. Bu varsayımlar altında, yörünge koordinasyon probleminin, iş atölyesi çizelgeleme problemine benzer olduğu belirtilmiştir.

Shih ve ark. (1991), ortak bir çalışma alanına sahip iki SCARA robotun çarpışmasız yörünge uzaylarını planlamak için bir yöntem önermişlerdir. Şekil 2.3'te SCARA tipi bir robot gösterilmiştir. Çalışmada robotik manipulatörler tek bir sistem olarak ele alınmıştır. Ayrıca robotların çakışmasını temsil eden bir çarpışma haritası oluşturulmuştur. Çalışma uzayında çakışmasız bir yörüngenin birincil kısıt olarak bulunması ve ortak uzaydaki amaç fonksiyonunun en küçüklenmesi ile yörünge planı oluşturulmaktadır. Bir eğim projeksiyon yöntemi, mevcut ve istenen eklem açıları arasındaki hatayı en aza indirir ve diğer kriterleri optimize eder. Önerilen formülasyon, çoklu SCARA robotları kullanan esnek montaj hücrelerinde uygulanmıştır.

Zurawski ve Phang (1992), ortak bir çalışma alanında çalışan iki robotik manipulatör için çarpışmasız yörünge planlaması yaklaşımı sunmuşlardır. Yaklaşım, robotik manipulatörlerin, önceden bilinmeyen engeller (örneğin, montaj hücresi) olmaksızın bilinen ve yapılandırılmış bir ortamda çalıştıklarını ve iyi tanımlanmış tekrarlanan görevleri bağımsız olarak gerçekleştirdiğini varsayar. Ayrıca, her robotik manipulatörün ortak çalışma alanında iki iyi tanımlanmış konumda görev yapabildiği varsayılmaktadır. Robotik manipulatörler ana / yardımcı statüye atanır. Ana robotik manipulatör görevlerini tamamlamak için önceliğe sahiptir. Bu yöntemin performansı, ortak çalışma alanında yalnızca bir robotik manipulatörün bulunmasına izin veren yaklaşım ile karşılaştırılmıştır. Önerilen yöntemin öncelikli olarak çevrim dışı yol planlaması için geliştirilmiş olmasına rağmen, ek varsayımlarla, çevrimiçi yol tespiti için de kullanılabileceği belirtilmiştir.

Koga ve Latombe (1994), engeller arasında yer alan iki hareket uzayı arasında hareketli bir nesneyi manipüle etmek amacıyla işbirliği içinde bulunan birkaç robotik manipülatör için otomatik hareket yörüngelerinin ele alındığı bir çalışma sunmuşlardır. Çarpışmaları önlemek için robotlar bir koldan diğerine geçerek nesnenin kavramasını değiştirebilirler. Hareketli nesnenin aynı anda hareket eden iki kolla hareket ettirilebileceği bir durum da göz önünde bulundurulmuştur. Bu planlama probleminin çözümü için bir yaklaşım, üç boyutlu ortamda hareket eden üç koldan oluşan bir robot sistemi ile tarif edilmiş ve gösterilmiştir. Bu yaklaşımı uygulayan bir planlayıcı ile yapılan deneyler, sadece hızlı değil, aynı zamanda çarpışmasız yollar bulmakta da güvenilir olduğunu göstermektedir.



Şekil 2.3. SCARA robotu

Rana ve Zalzal (1995), çoklu robotik manipülatörlerin çarpışmasız yörünge planlaması için evrimsel bir algoritma geliştirmişlerdir. İki robot manipülatörünün yörüngelerini planlamak için küresel bir yörünge planlama tekniği kullanılır. Yörüngeler, robotların geçmesi gereken bir geçiş noktaları dizisiyle temsil edilir. Yörünge planlama algoritması, yörünge uzunluklarını en aza indirmek için evrimsel teknikleri kullanırken, hızdaki değişimleri ve robotik manipülatörler arasındaki çarpışmayı en aza indirmek için yörüngeler üzerinde yer alan hareket noktalarının dağılımını kullanır. Bu algoritma, daha önce sunulan GA tabanlı yol planlayıcılarından farklıdır. Çünkü geçiş noktaları, robotun çalışma alanı üzerinde sabit bir ızgara (grid) üzerinde yer alan noktalarla sınırlı

değildir. Geliştirilen algoritma ile elde edilen sonuçlar, iki robotik manipülatör için planlanan çarpışmasız yörüngeler ve daha sonra üç boyutlu operasyonel alanda hareket eden iki adet 3 serbestlik dereceli robotik manipülatörü için sunulmuştur. Çarpışmaları kontrol etmek için kullanılan yöntem çevrimdışı olarak kullanılmakta olup, hem oldukça basittir hem de esnek ve verimli bir kullanım sunmaktadır.

Sutdhiraksa ve Zurawski (1996), ortak bir çalışma alanında eş zamanlı olarak montaj işlemini gerçekleştiren iki robot kolu içeren montaj görev çizelgelemesine dair bir yaklaşım sunmuşlardır. Önerilen yaklaşım, arama yönteminin ve sezgisel fonksiyonların kombinasyonuna dayalı olarak, en uygun aday çözümlerle sonuçlanan bir yaklaşım üzerinedir.

Ridao ve ark. (1998), iki robotik manipülatörün koordineli hareket planlarını elde etmek için bir EA ve yerel arama algoritması birleşimine dayalı bir yöntem sunmuşlardır. Çalışmada Ayrık Planlama (Decoupled Planning) yaklaşımı kullanılmıştır. Bu amaçla, problem iki alt probleme ayrıştırılmıştır: Her bir robot manipülatörü için birbirinden bağımsız olarak çarpışma olmayan bir yörüngenin bulunduğu yörünge planlaması, sadece sabit engellerin göz önünde bulundurulduğu ve diğer robotik manipülatörlerle çarpışmayı önlemek için yörüngelerin zamanlandığı ve senkronize edildiği yörünge planlaması.

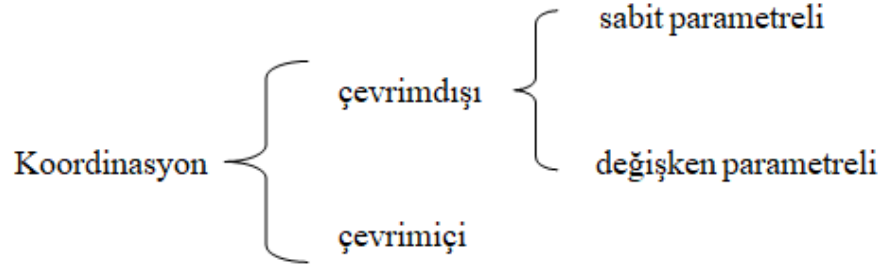
Svestka ve Overmars (1998), çoklu robotlar için yörünge planlama problemine pozisyonlarını aynı statik ortamdaki uygun hareketlerle değiştirecek bir dizi robotun yer aldığı farklı bir yaklaşım sunmuşlardır. Daha önceki çalışmalarda kullanılan ayrık planlama yaklaşımından ziyade, koordineli bir yaklaşım kullanılmıştır. Sonuç olarak herhangi bir çözülebilir problemin sınırlı bir süre içinde çözüleceğini göstermişlerdir. Robotların hareketlerini depolayan bir veri yapısı iki adımda oluşturulmuştur. İlk olarak, sadece bir robot için bir yol haritası oluşturulmuştur. Bunun için yaklaşımın farklı robot türlerine kolayca uygulanabileceğini garantileyen olasılıksal yörünge planlayıcısını kullanmışlardır. İkinci adımda, bu basit yol haritalarının bir kısmı karma robot için bir yol haritasına eklenmiştir. Yöntemi araba benzeri robotlara uygulamışlar ve karmaşık ortamlarda beş adete kadar araba benzeri robot içeren problemlerin, bir ön

işlem adımından sonra, hesaplama sürelerinde, saniyeler içinde başarıyla çözüldüğünü gösteren simülasyon sonuçlarını sunmuşlardır. Böyle bir ön işlem adımının, belirli bir statik ortam için sadece bir kez gerçekleştirilmesi gerektiği belirtilmiştir.

Aronov ve ark. (1999), belirli sayıda engel içeren bir ortak çalışma alanında çalışan robotik manipülatör çiftleri ve üçlüleri için hareket planlama problemini incelemişlerdir. Bu tür problemleri çözümenin standart bir yolunun, robotların serbestlik dereceleri bireysel robotların serbestlik derecelerinin toplamı olan tek bir robot olarak görmek olduğunu söylemişlerdir.

Todt ve ark.'na (2000) göre, iki ya da daha fazla robotun koordinasyonu, robotların hareketlerini kendi aralarında uyumlu hale getirmekten ibarettir. Böylece robotlar, görevlerini kendi aralarında çarpışma olmadan yürütürler. Bu durum, hızlarını sabitleyerek, robotların aynı zamanda aynı yeri geçmeyecek şekilde ayarlanmasıyla elde edilir. Çalışmada, daha önce yayınlanmış eserlerde kullanılan temel kavramlar analiz edilmiş ve bir sınıflandırma yapılmıştır. Her bir kategori altında örnek çalışmalar incelenmiş ve koordinasyon yöntemlerinin sınıflandırması yapılmıştır (Şekil 2.4).

Bennewitz ve ark. (2001), ayırık planlama tekniklerinin popüler ve verimli olmasına rağmen çoğu zaman optimal sonucu elde edemedikleri için dezavantajlı olduklarını söylemişlerdir. Çalışmalarında, öncelikli ve ayrıştırılmış planlama teknikleri için öncelik şemalarını optimize etmek adına bir yöntem sunmuşlardır. Yaklaşım, çözüm bulmak için yamaç tırmanışı ile rastgele bir arama yapar. Genel yol uzunluklarını en aza indirir. Yöntem, gerçek robotlarda ve kapsamlı simülasyon çalışmalarında uygulanmış ve test edilmiştir. Deneysel sonuçlar, çalışmada önerilen yöntemin başarısızlık sayısını ciddi ölçüde azaltabildiğini ve farklı öncelikli veya ayrıştırılmış yol planlama teknikleri ve hatta büyük robot grupları için genel yol uzunluğunu önemli ölçüde azaltabildiğini göstermektedir.



Şekil 2.4. Robot koordinasyon yöntemlerinin sınıflandırılması (Todt ve ark. 2000)

Jager ve Nebel (2001), aralarında çarpışma ve kilitlenme olmaması için birden fazla mobil robotun bağımsız olarak planlanan yörüngelerini koordine etmek için bir yöntem sunmuşlardır. İki robot arasındaki mesafe belirli bir değerin altına düştüğünde, planlanan yörüngeleri hakkında bilgi alışverişinde bulunurlar ve bir çarpışma tehlikesi olup olmadığını belirlerler. Muhtemel bir çarpışma tespit edilirse, çarpışmaları önlemek için hareketlerini izlerler ve gerekirse, yörüngelerinin belirli bölümleri arasında boşa kalma süreleri eklerler. Görevi güvenilir bir şekilde gerçekleştirmek için üç farklı dağıtım algoritmasının bir kombinasyonunu kullanmışlardır. Genel bir senkronizasyon kullanmayan bu algoritmalar aynı zamanda birbirlerini de etkilemezler.

Akella ve Hutchinson (2002), ortak bir çalışma alanında çalışan birden fazla robotik manipülatörün çarpışma olmadan hareketlerini koordine etmenin önemli bir yetenek olduğunu belirtmişler ve yörüngeleri (yol ve hız tanımlanmış) belirlendiğinde, birden fazla robotik manipülatörün hareketlerini koordine etme görevini ele almışlardır. Bu çarpışmasız bağımsız yörünge koordinasyonu probleminin özellikle, otomotiv endüstrisindeki kaynak ve boyama istasyonlarında ortaya çıktığını söylemişlerdir. Minimum zaman çözümlerini otomatik olarak oluşturmak için bu problemlerin karışık tam sayılı programlama modelini geliştirmişlerdir. Bu yöntem, hem mobil robotlara hem de eklemlili kollara sahip robotlara uygulanabilir olup robotların serbestlik derecelerinin sayısı üzerinde herhangi bir kısıtlama koymamaktadır. Bu yöntemin başlıca avantajı, birden fazla robotun hareketlerini koordine etme yeteneğidir. Robotların yörüngeleri belirlendiğinde bile, çoklu robotların koordinasyon süresini en küçükleme probleminin NP-zor olduğunu belirtmişlerdir. Akella ve Peng (2004), robot dinamiklerini dikkate alan önceki yaklaşımlar için, birden fazla robotik manipülatörün

hareketini koordine etme görevini ele almışlardır. Robotik manipülatörler için temel bir zaman ölçekleme kuralı kullanılmıştır.

Kobetski ve ark. (2006) literatürde, endüstriyel sistemlerin kontrolü ve genel optimizasyon algoritmalarının geliştirilmesi gibi araştırma alanlarına ayrı ayrı önem verilmiş olmasına rağmen, bu iki araştırma alanını bir şekilde birleştirmeye ihtiyaç olduğunu öne sürmüşlerdir. Çalışmada, endüstriyel robot hücreleri için tasarlanmış iki çizelgeleme algoritması performans açısından ele alınmıştır. Ayrık olay sistemlerinde çalışan arama yöntemlerinden biri olan ve bir düğümden hedef bir düğüme en kısa yolu bulmayı sağlayan A* tabanlı bir arama algoritmasının yeni bir gösterimi, iyi bilinen karışık tam sayılı programlama algoritmasına karşı önerilmiş ve kıyaslanmıştır.

Saha ve Isto (2006), temel ayrık planlama yaklaşımı ile yeni bir çoklu robot planlama stratejisi geliştirmişlerdir. Deneysel sonuçlar, bu yeni stratejinin, ayrık planlama yaklaşımının güvenilirliğine önemli ölçüde katkıda bulunduğunu göstermektedir. Testlerde, yeni strateji, hem merkezi hem de ayrık planlama yaklaşımını geride bırakarak, çoklu robot planlama problemlerini çözmek için, hem merkezi hem de ayrık planlama yaklaşımından daha iyi bir seçenek olabileceğini göstermektedir.

Zhang ve ark. (2009), 40 serbestlik derecesine sahip insansı robotlarda tüm vücut için yörünge planlama algoritması sunmuşlardır. Hareket planlama problemi, düşük boyutlu alt problemler dizisine ayrıştırılmıştır. Formülasyon, insansı bir modelin sıkı sıkıya bağlı alt modellerden oluşan bir sistem olması gerçeğine dayanmaktadır ve alt problemleri aşamalı olarak çözmek için sınırlandırılmış bir koordinasyon şeması kullanmaktadır. Ayrıca dar alanlarda çakışmasız yörüngeleri hesaplamak için yerel bir yol iyileştirme algoritması önermişlerdir.

Spensieri ve ark. (2013), çarpışmaları önleyerek, çevrim süresini en aza indirmek için ortak bir ortamı paylaşan robotik manipülatörlerin yörüngelerini planlama problemini incelemişlerdir. Bu problem, birçok robotik manipülatörün bir araba gövdesini birleştirmek için kaynak işlemleri yaptığı otomotiv endüstrisinde özellikle önemlidir. Çalışmanın başlıca katkıları şunlardır:

- Yörünge planlama problemini, atölye çizelgeleme problemine benzer bir şekilde modellemek
- Bir dal ve sınır optimizasyon algoritması ile yörünge planlama problemini çözmek

Algoritmanın doğruluğunun ve performansının, genel amaçlı bir paket programda çözülen karışık tam sayılı doğrusal programlama formülasyonu ile karşılaştırılarak değerlendirildiği bir hesaplama çalışması sunulmuştur. Bu çerçevede otomotiv endüstrisinden bir endüstriyel test örneği çözülmüştür.

Afaghani ve Aiyama (2014), noktadan noktaya hareket (PTP) komutları kullanılarak kontrol edilen iki robotik manipülatör arasındaki çarpışmaların ve kilitlenmelerin önlenmesi için çevrimiçi bir sistem kurmayı amaçlamışlardır. Her iki robot da aynı çalışma alanını paylaşmakta ve sistemi başlattıktan sonra gönderilecek komutları önceden bilmemektedirler. Robotların tutucu uç elemanlarının (end-effector) arasındaki çarpışmaları tespit etmek için bir çarpışma haritası tasarımı önermişlerdir. Çalışmada, robotik manipülatörlerin tüm gövdeleri arasındaki çarpışmaları tespit etmek ve onları haritada çarpışma alanları olarak göstermek için gelişmiş bir çarpışma haritası yöntemi oluşturulmuştur. Ayrıca harita, herhangi bir robotik manipülatörün diğerinin önünde bir engel haline gelmesi durumunda meydana gelebilecek kilitlenmelerin önüne geçmek için kullanılmıştır.

Lee ve ark. (2014), montaj önceliği kısıtlamaları olduğunda iki kollu, iki robotun koordinasyonunu robot için verilen her yörüngeyi dikkate alan ayrık planlama yöntemiyle açıklamaya çalışmışlardır. Yöntem, robotların serbestlik derecesi yüksek olduğunda etkilidir.

Lim ve Han (2014), çalışma alanında hareket ettirilen çift kollu robotun karmaşık parçaları içeren montaj görevinin, özellikle robot hareketi, montaj görevi sırasında herhangi bir çarpışmayı önlemek için kısıtlandığında, ağır hesaplama yükü getirdiğini belirtmişlerdir. Bu nedenle, çift kollu robotun operasyonel alandaki karmaşık parçaları içeren montaj görevi için bir yörünge planlama algoritması önermişler ve önerilen yöntemin etkinliğini doğrulamak için simülasyon uygulaması gerçekleştirmişlerdir. Sonuçlar, önerilen yörünge planlama yönteminin, özellikle hesaplama zamanı

açısından, yapılandırma alanındaki geleneksel yörünge planlama yöntemine kıyasla daha etkili olduğunu göstermektedir.

Wu ve ark. (2016), paylaşılan bir çalışma alanında birden çok robotun hareket planı problemini incelemişlerdir. Amaç, robotların yörüngelerini koordine etmek için bir sıralama düzeni bulmak, böylece aralarındaki çarpışmalardan ve kilitlenmelerden (her bir robotun diğerinin devam etmesini beklediği durum) kaçınmaktır. İş istasyonunun bilindiğini, robotların yörüngelerinin önceden planlanabileceğini ve tüm robotların eşzamanlı olarak hareket ettiğini varsayımlardır. Koordinasyonun, en kısa yolu bulmayı sağlayan A* sezgisel algoritması ile sağlanabileceğini savunmuşlardır. Önerilen yöntem iki adımdan oluşur:

- Robotlar ve çevre arasında çarpışmadan kaçınmayı dikkate alan yörünge planlaması
- Tüm olası çarpışma durumlarını hesaplamak ve koordinat belirleme problemini yol segmentlerini çizelgeleme problemine çevirmek

Önerilen algoritmanın deneysel koşullarda uygulaması yapılmış olup, önerilen yaklaşımın endüstriyel koşullarda da uygulanabilir olduğu belirtilmiştir.

Son olarak Yoon ve ark. (2017), gölge uzay modelleme kavramını tanıtarak, çarpışmasız bir görev planlama yöntemi sunmuşlardır. Gölge uzayı, robotun hareketi tarafından işgal edilen bir çalışma alanı alt kümesi olarak tanımlanır. Çarpışma, iki manipülatörün gölge uzaylarının kesişiminin hesaplanmasıyla tespit edilebilir ve böylece önlenebilir. Yazarlar ikili robotik manipülatörler için çarpışmasız optimal görev dizilerini oluşturmak amacıyla ikili kromozomlara sahip bir GA önermiştir.

Bu tez çalışmasında diğer çalışmalardan farklı olarak, çarpışmasız ve kilitlenmesiz görev olan robotik manipülatörlerin yer aldığı bir montaj hattında, robot ve istasyonların iş yüklerini dengeleme amacıyla iki aşamalı bir rassal sıralama algoritması önerilmektedir. Çarpışma kontrolü için Rana ve Zalzala'nın (1995) çalışmasından esinlenilmiştir.



3. MATERYAL VE YÖNTEM

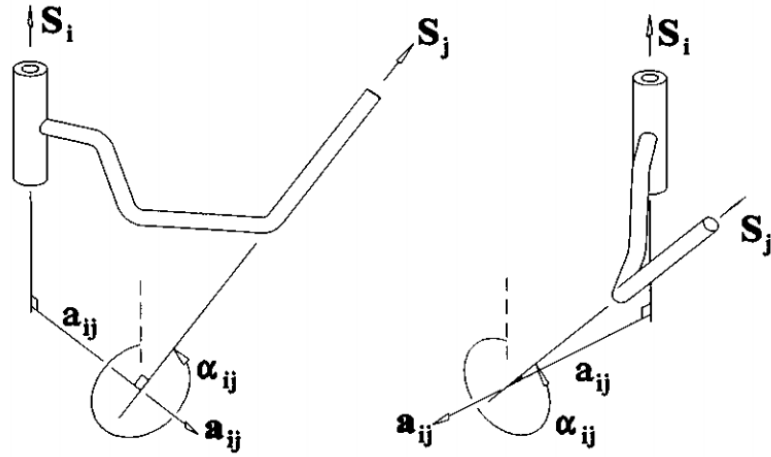
Robotik montaj hatlarında insanların yer aldığı montaj hatlarına göre bazı farklılıklar vardır. Bunlardan en önemlisi otonom olarak hareket eden robotların çarpışma ve kilitlenme durumudur. Bu bölümde sırasıyla, robotik manipülatörlerden oluşan iş istasyonlarının yer aldığı robotik montaj hattı dengeleme problemine ait materyale, bu problemin çözümü için önerilen yönteme ve bu yöntemleri oluşturan prosedürlere yer verilmiştir.

3.1. Materyal

Bu bölümde robotik manipülatörlere ait teknik özelliklerle birlikte kinematik bilgilerine, çok-işçili montaj hattı dengelemenin matematiksel modeline, çarpışma engelleme yöntemine ve görev atamalarında kullanılan rastgele dizi sıralama algoritmasına yer verilmiştir.

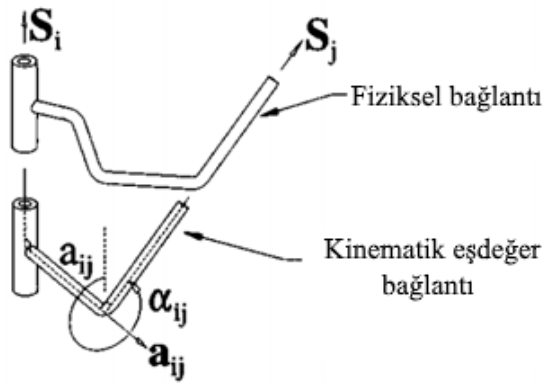
3.1.1. Robotik manipülatörlerin bağlantı elemanları

"Robotik manipülatör" terimi, bağlantı elemanları (links) ve eklemlerden (joints) oluşan seri bağlı bir zinciri tanımlar. Robotik manipülatör bir ucundan zemine sabitlenmiş olup, en uç noktasına ise bir tutucu uç eleman veya kavrama elemanı eklenmiştir. Bağlantıların katı birer cisim olduğu varsayılmıştır. Şekil 3.1, genel olarak, birim yön vektörleri \mathbf{S}_i ve \mathbf{S}_j ile etiketlenmiş bir çift ardışık eklem eksenini birleştiren bir bağlantı elemanını göstermektedir. İki skaler parametre, bağlantı uzunluğu a_{ij} ve büküm açısı α_{ij} , bu çift eğrilmiş eksenlerin göreceli konumunu tanımlar. Bağlantı uzunluğu, eksenler arasındaki karşılıklı dikey mesafedir ve büküm açısı, \mathbf{S}_i ve \mathbf{S}_j vektörleri arasındaki açıdır. Birim vektör \mathbf{a}_{ij} , Şekil 3.1'de gösterildiği gibi $\mathbf{S}_i \times \mathbf{S}_j = \mathbf{a}_{ij} \sin \alpha_{ij}$ ile tanımlanır. Açıkçası, \mathbf{S}_i ve \mathbf{S}_j birim vektörlerinin doğrultusu serbestçe belirlenir. Ayrıca $\mathbf{S}_i \times \mathbf{S}_j$ (ya da $\mathbf{S}_j \times \mathbf{S}_i$) çarpımının sonucu ise her zaman \mathbf{a}_{ij} 'nin doğrultusunu belirler. Bu doğrultu, α_{ij} açısına saatin ters yönünde sağ el kuralı uygulanarak bulunur.



Şekil 3.1. Bir uzamsal bağlantının iki farklı görünümü (Crane III ve Duffy 2008)

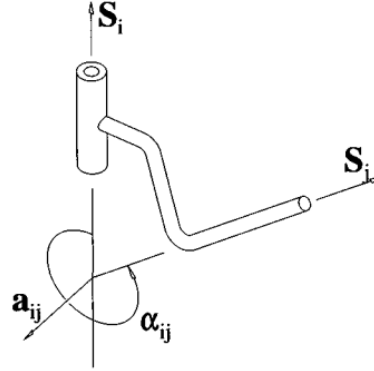
Bir seri bağlı manipülatörün kinematik modeli, robotun her fiziksel bağlantısının, \mathbf{a}_{ij} vektörü boyunca çizilmiş bir bağlantı ile yer değiştirilmesiyle yapılır. Bunun nedeni, asıl bağlantının fiziksel şeklinin geometrik bir öneminin olmamasıdır. Daha doğrusu, bir bağlantının geometrisi, bağlantı elemanı uzunluğu a_{ij} ve büküm açısı α_{ij} ile birlikte \mathbf{a}_{ij} , \mathbf{S}_i ve \mathbf{S}_j vektörlerinin yönleri ile tanımlanır. Şekil 3.2, Şekil 3.1'de gösterilen fiziksel bağlantının kinematik olarak eşdeğerini göstermektedir.



Şekil 3.2. Kinematik bağlantı (Crane III ve Duffy 2008)

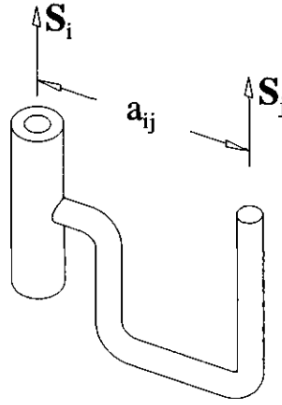
Şekil 3.3 ve 3.4 iki özel durumu göstermektedir. Birincisi \mathbf{S}_i ve \mathbf{S}_j vektörleri arasındaki dikey mesafe sıfır olduğunda, yani \mathbf{S}_i ve \mathbf{S}_j sonlu bir noktada kesiştiği zaman meydana gelir. Buna küresel bağlantı denir. İkincisi, büküm açısı α_{ij} sıfır veya π olduğunda

oluşur. Bu durumda, bağlantı düzlemseldir (planar) ve \mathbf{S}_i ve \mathbf{S}_j vektörleri paralel veya anti-paraleldir, yani \mathbf{S}_i ve \mathbf{S}_j sonsuzda bir noktada kesişir.



Şekil 3.3. Küresel bağlantı (Crane III ve Duffy 2008)

Özet olarak, bir bağlantı, bağlantı elemanı uzunluğu a_{ij} ve büküm açısı α_{ij} gibi iki skaler parametre ile tanımlanır. Birim vektör \mathbf{a}_{ij} 'nin genliği ise $\|\mathbf{S}_i \times \mathbf{S}_j\| = a_{ij} \sin \alpha_{ij}$ ile bulunur.

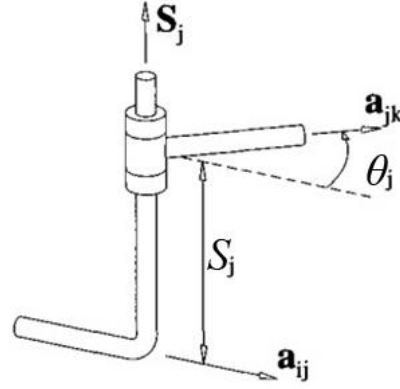


Şekil 3.4. Düzlemsel bağlantı (Crane III ve Duffy 2008)

3.1.2. Robotik manipulatörlerin eklemleri

Bir çift ardışık bağlantı arasındaki göreceli hareketin doğası, bağlantı eklemi türüne göre belirlenir. En basit ve en yaygın eklemlerden biri, R harfi ile gösterilen dönel (rotational) eklemdir. Bu eklem, Şekil 3.5'te gösterildiği gibi iki bağlantı elemanını

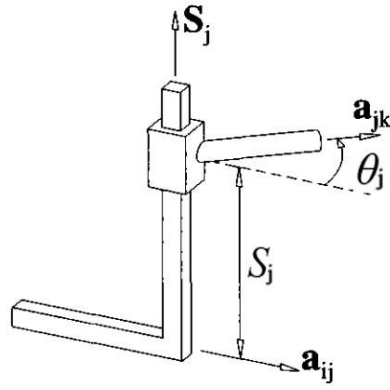
birbirine bağlar. jk bağlantı elemanı, vektör S_j etrafında ij bağlantı elemanına göre dönebilir.



Şekil 3.5. Dönel eklem (Crane III ve Duffy 2008)

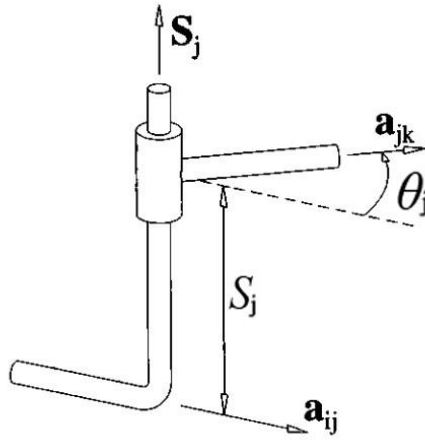
Böylece jk bağlantı elemanı, ij bağlantı elemanı ile ilgili olarak bir serbestlik derecesine sahip olur. Eklem açısı θ_j iki bağlantı elemanının dönüşünü rotasyonunu ölçer ve birim vektörü S_j etrafında sağ el kuralına göre a_{ij} ve a_{jk} arasındaki açı olarak tanımlanır. jk bağlantısı, sadece ij bağlantısına göre göreceli olarak dönebildiğinden, S_j mesafesi bir sabittir. Bu parametre eklem öteleme (offset) mesafesi olarak adlandırılır (a_{ij} ve a_{jk} vektörlerine normal doğrultudaki uzunluktur). Özetle, bir döner eklem, değişken eklem açısı θ_j ve sabit öteleme değeri S_j ile tam olarak açıklanabilir.

P harfi ile gösterilen prizmatik eklem, jk bağlantısının, vektör S_j ile paralel olarak, ij bağlantısına göre bir serbestlik derecesine sahip olmasını sağlar (bakınız Şekil 3.6). θ_j açısı bir sabittir ve dönel eklem ile aynı şekilde ölçülür, yani vektör S_j etrafında sağ el kuralı ile ölçülen a_{ij} ve a_{jk} vektörleri arasındaki açıdır. Öteleme mesafesi S_j , prizmatik eklem için bir değişkendir.



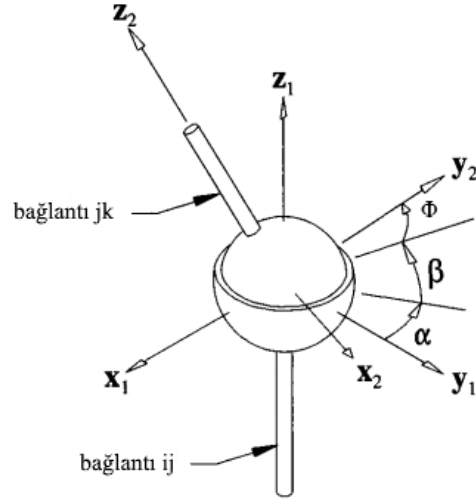
Şekil 3.6. Prizmatik eklem (Crane III ve Duffy 2008)

C harfi ile temsil edilen silindirik eklem, jk bağlantı elemanının Şekil 3.7'de gösterildiği gibi ij bağlantı elemanına göre vektör S_j etrafında dönmesine ve a_{ij} doğrultusunda ötelenmesine izin verir. Böylece jk bağlantı elemanı, ij 'ye göre iki bağımsız serbestlik derecesine sahiptir. Eklem açısı θ_j ve denge mesafesi S_j 'nin her ikisi de değişkendir.



Şekil 3.7. Silindirik eklem (Crane III ve Duffy 2008)

Küresel eklem veya soket eklemi, Şekil 3.8'de gösterilmiştir. jk bağlantısı, ij bağlantı elemanına göre üç serbestlik derecesine sahiptir. Bu üç serbestlik derecesi, ij bağlantı elemanına bağlı $(x_1y_1z_1)$ koordinat sistemini, jk bağlantı elemanına bağlı $(x_2y_2z_2)$ koordinat sistemi ile hizalayan, α , β , Φ ile tanımlanan üç dönme açısı olarak düşünülebilir. Şekil 3.8'de, her bir bağlantı elemanına bağlı olan koordinat sistemlerinin orijinleri, küresel eklemde yer almaktadır.



Şekil 3.8. Küresel eklem (Crane III ve Duffy 2008)

jk bağlantısına bağlanan koordinat sistemi 2, aşağıdaki üç rotasyonu gerçekleştirerek, ij 'ye bağlanan koordinat sistemi 1'den elde edilebilir:

- Z eksenine ile ilgili bir α rotasyonu
- değiştirilmiş X eksenine ile ilgili β rotasyonu
- değiştirilmiş Z eksenine ile ilgili Φ rotasyonu

İkinci koordinat sisteminde bilinen bir noktayı ilk koordinat sistemine dönüştüren dönüşüm basitçe

$${}^1_2\mathbf{R} = \mathbf{R}_{z\alpha}\mathbf{R}_{x\beta}\mathbf{R}_{z\Phi} \quad (3.1)$$

olarak verilmiştir. Burada

$$\mathbf{R}_{z\alpha} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

$$\mathbf{R}_{x\beta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix} \quad (3.3)$$

$$\mathbf{R}_{z\Phi} = \begin{bmatrix} \cos\Phi & -\sin\Phi & 0 \\ \sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

olarak verilmiştir.

Birinci ve ikinci koordinat sistemi arasındaki ilişki birçok farklı şekilde tanımlanabilir, yani, sırası ve karşılık gelen dönme açıları değiştirilebilir. Küresel bir eklemin tasarımı ve uygulanması karmaşık bir süreç olabilir (Crane III ve Duffy 2008).

3.1.3. Genel koordinat sistemi kinematik modelleri

Genel koordinat sistemi kinematik modelleri, kavramsal olarak, her bir bağlantıya göre sabit olan kartezyen koordinat sistemlerinin birleştirilmesine dayanmaktadır. İki ardışık bağlantının koordinat sistemi arasındaki uzamsal dönüşüm, iki bağlantıyı birbirine bağlayan eklemin pozisyonunun bir fonksiyonudur. Robotikte Denavit ve Hartenberg (1955) tarafından tanıtılan ve daha sonra Pieper (1968) ve Paul (1981) tarafından benimsenen (4x4) homojen dönüşüm matrisi, bu uzamsal dönüşümleri tanımlamak için en yaygın yaklaşım haline gelmiştir. Genel homojen dönüşüm matrisi **C**, aşağıdaki gibi ifade edilir:

$$\mathbf{C} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

ya da vektör bileşenleri açısından,

$$\begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Burada,

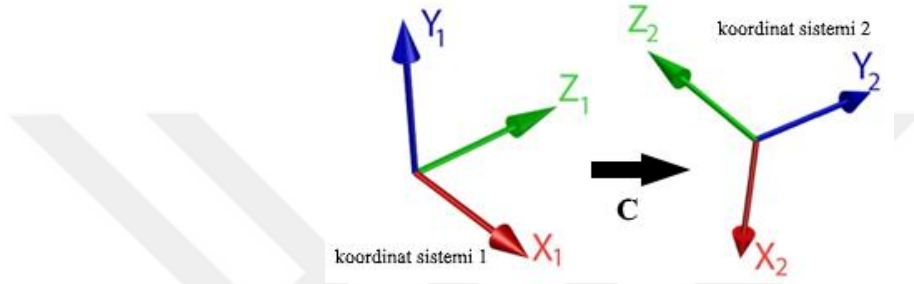
$$\mathbf{n} = [n_x n_y n_z]^T, \quad (3.7)$$

$$\mathbf{o} = [o_x o_y o_z]^T, \quad (3.8)$$

$$\mathbf{a} = [a_x a_y a_z]^T, \quad (3.9)$$

$$\mathbf{p} = [p_x p_y p_z]^T. \quad (3.10)$$

Şekil 3.9'da iki rastgele kartezyen koordinat sistemi gösterilmektedir. Eğer C matrisi, koordinat sistemi 1'den koordinat sistemi 2'ye uzamsal dönüşümü tanımlarsa, \mathbf{n} , \mathbf{o} ve \mathbf{a} birim yön vektörleri koordinat sistemi 2'nin X, Y ve Z eksenlerinin sırasıyla koordinat sistemi 1'e göre yönünü belirtir. \mathbf{p} vektörü, koordinat sistemi 2'nin koordinat sistemi 1'e göre konumunu belirler. \mathbf{n} , \mathbf{o} ve \mathbf{a} tarafından oluşturulan (3×3) 'lük ortogonal dönüşüm matrisi $\mathbf{R}_{(3 \times 3)}$, (3.5) matrisinin yönelim (orientational) bileşeni ve \mathbf{p} ise (3.5)'in öteleme (translational) bileşenidir (Stone 1987).



Şekil 3.9. İki keyfi kartezyen koordinat sistemi (Stone 1987)

Koordinat sistemi 1'deki bir P noktasının koordinatları, $\mathbf{x}_1 = [x_1 \ y_1 \ z_1 \ 1]^T$, koordinat sistemi 2'ye göre, $\mathbf{x}_2 = [x_2 \ y_2 \ z_2 \ 1]^T$ koordinatları açısından aşağıdaki gibi ifade edilir:

$$\mathbf{x}_1 = \mathbf{C}\mathbf{x}_2 \quad (3.11)$$

Homojen dönüşüm matrisi (3.5), robotikte yaygın olarak kullanılan ve Paul (1981) tarafından sunulan çeşitli matematiksel özelliklere sahiptir.

İki kartezyen koordinat sistemi arasındaki dönüşüm, her zaman temel dönüşümlerin bir bileşimi olarak ayrıştırılabilir. Paul'un notasyonunu kullanarak, altı temel dönüşüm aşağıda verilmiştir.

- Trans $(x,0,0)$ -- X eksenini boyunca x birim ötele
- Trans $(0,y,0)$ -- Y eksenini boyunca y birim ötele
- Trans $(0,0,z)$ -- Z eksenini boyunca z birim ötele

- $\text{Rot}(x,\theta)$ -- X eksenini etrafında θ açısı kadar döndür
- $\text{Rot}(y,\theta)$ -- Y eksenini etrafında θ açısı kadar döndür
- $\text{Rot}(z,\theta)$ -- Z eksenini etrafında θ açısı kadar döndür

3.1.4. Denavit-Hartenberg modeli

Denavit-Hartenberg modeli, fiziksel yorumu, katı tanımı ve çarpımsal yapısı nedeniyle standart robot kinematik modeli haline gelmiştir.

$$\mathbf{T}_n = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \dots \cdot \mathbf{A}_n \quad (3.12)$$

Denklem (3.12) 'de, (4x4) homojen dönüşüm matrisi \mathbf{T} , manipülâtörün tabanına göre sabitlenmiş genel bir koordinat sistemine göre bir manipülâtörün tutucu uç elemanına göre sabitlenmiş bir koordinat sisteminin konumunu ve yönünü tanımlar. Kavramsal olarak, (3.12)'nin sağ tarafı, manipülâtör bağlantılarının her birine göre sabitlenmiş koordinat sistemleri arasındaki uzamsal ilişkileri tarif eder. Denavit-Hartenberg bağlantı koordinat sistemleri ($i = 1, \dots, n$ için τ_i^2) ileri dönüşüm matrisleri \mathbf{A}_i 'nin aşağıdaki formül tarafından hesaplanacak şekilde belirtilmiştir.

$$\mathbf{A}_i = \mathbf{A}_i(q_i) \equiv \text{Rot}(z, \theta_i) \text{Trans}(0,0, d_i) \text{Trans}(a_i, 0,0) \text{Rot}(x, \alpha_i) \quad (3.13)$$

Denklem (3.12) için girdi parametreleri, n adet genelleştirilmiş q_i ortak koordinatlardır. Genelleştirilmiş koordinatlar, eklem fiziksel doğasını açıkça belirtmeden (yani, ister dönel ister prizmatik olsun) eklem pozisyonlarını temsil etmek için kullanılır. (3.13)'ün genişletilmiş hali

$$\mathbf{A}_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & -\sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

olarak elde edilir.

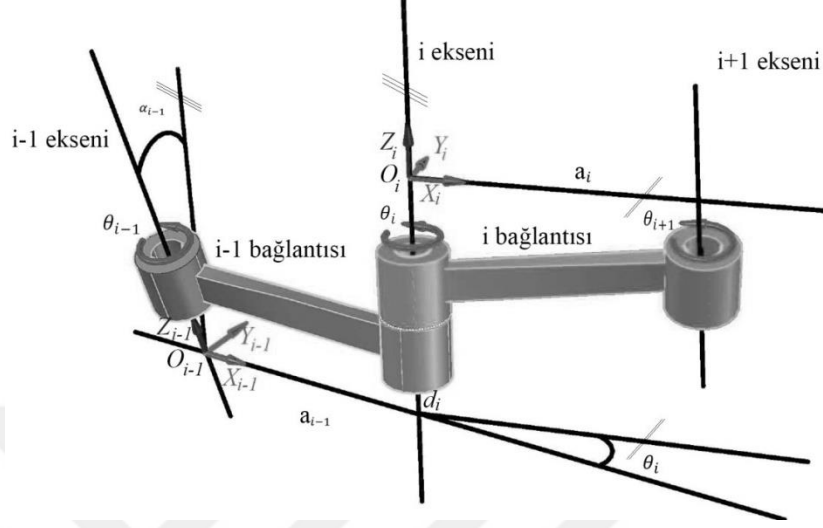
Denklem (3.13)'te koordinat sistemi τ_{i-1} 'den koordinat sistemi τ_i 'ye olan A_i dönüşümü, dört adet Denavit-Hartenberg parametresinin (θ_i , d_i , a_i ve α_i) bir fonksiyonudur. Bir n serbestlik derecesine sahip manipülatör, n kontrol edilebilir eklem pozisyonu olan $4n$ adet parametre belirlenmesini gerektirir. Şekil 3.10, bu parametrelerin sırasıyla, eklemler için fiziksel tanımını göstermektedir.

Denavit-Hartenberg modelinin özellikleri, bağlantı koordinat sistemlerini belirtmek için uygulanan Denavit-Hartenberg kurallarından gelmektedir. Denavit-Hartenberg kuralları, ardışık eklem eksenleri arasındaki uzamsal ilişkilerin geometrik bir analizini takip eder. Denavit-Hartenberg kuralları, koordinat sistemleri için aşağıdaki tanımları gerektirir (Stone 1987) :

- Koordinat sistemi τ_{i-1} 'in Z eksenini, i eklem eksenine paralel olmalıdır.
- Koordinat sistemi τ_{i-1} 'in orijini, eklem $i-1$ ve eklem i eksenini ile eklem i eksenleri arasındaki ortak normalin kesişme noktasında eklem i ekseninde bulunmalıdır.
- Koordinat sistemi τ_{i-1} 'in X eksenini, eklem $i-1$ ve eklem i eksenleri arasındaki ortak normale paralel olmalıdır. X ekseninin pozitif yönü, i eklem eksenine doğru işaret eder.
- Koordinat sistemi τ_{i-1} 'in Y eksenini, X eksenini birim yön vektörü ile Z eksenini birim yön vektörünün vektörel kesişim sonucu olarak tanımlanır.
- Eklem i ve eklem $i+1$ eksenleri kesişiyorsa, kesişme noktası τ_{i-1} koordinat sisteminin orijini'dir.
- Eğer i ve $i+1$ eklemlerinin eksenleri koordinat sistemi τ_{i-1} 'in orijinine paralel olarak seçilirse, bir sonraki bağlantı için d_{i+1} eklem mesafesi sıfıra eşit olur.
- Temel bağlantı τ_0 koordinat çerçevesinin orijini, bağlantı 1 koordinat çerçevesi τ_1 'in orijini ile çakışır.
- Son koordinat sistemi τ_n 'in orijini, bir sonraki son koordinat sistemi τ_{n-1} 'in orijini ile çakışır.

Bu atamalar, (3.12)'deki Denavit-Hartenberg modelinin fonksiyonel formunu garanti eder. Bağlantı koordinat çerçevelerini belirtmek için farklı kurallar kullanarak, (3.12) ile aynı çarpımsal yapıya sahip alternatif kinematik modeller formüle edilebilir. Katı

bağlantılar ve tek serbestlik dereceli eklemlere sahip robotlar için (3.12) modeli kesin sonuç verir (Stone 1987).



Şekil 3.10. Eklemler için Denavit-Hartenberg parametreleri (Craig 1989)

3.1.5. İleri ve ters kinematik

Kinematik için geleneksel çözüm yaklaşımı, son zamanlarda özellikle bilgisayar grafikleri (animasyon film karakterleri vb.), endüstriyel ortam ve modern teknolojinin çeşitli alanlarında oldukça önem kazanmıştır. Tüm bu uygulama alanlarında tutucu uç elemanının ve robotik manipülatörün ortak değişkenlerinin kartezyen koordinatlarının hem yönelimini hem de konumunu değerlendirmek gereklidir. Tutucu uç elemanının pozisyon ve yönelimini ve eklem değişkenlerini değerlendirmek için denklem (3.13) ifadesi ile verilen homojen dönüşüm matris yöntemi benimsenebilir. Bu yöntem eklem ve bağlantıların kinematik ilişkisini tanımlamak için geleneksel bir araçtır. Dahası, bu gösterim yöntemi, robotik manipülatörün tutucu uç elemanına ait pozisyonun takip edilebilmesi için uzun yıllardan beri kullanılmaktadır. Öte yandan bu yöntem, bir sistemin 6 serbestlik dereceli gösterimi için aşırı derecede işlem yükü getirir. Fazladan hesaplama yapılacak olması genellikle daha fazla hesaplama maliyeti ve daha fazla depolama alanı tüketir. Bu aynı zamanda daha fazla karmaşıklık yaratan matematiksel işlemler sorunu ile de ilgilidir. Bu nedenle, koordinatların temsil edilmesi için birçok

alternatif yöntem tanıtılmıştır. Önerilen yöntem her zaman, mekanizmanın temsili ve sistemin dönüşümü için daha az karmaşık ve sayısal anlamda verimli olmalıdır.

Basit bir şekilde ileri kinematik, bir robotik manipülatöre ait tüm eklem açıları bilinirken, tutucu uç elemanın konumunu bulmak olup, ters kinematik ise bir robotik manipülatörün yalnızca tutucu uç elemanının konumu bilinirken eklem açılarının bulunması olarak tanımlanabilir.

Kinematik, moment/kuvvetleri veya hareket kaynaklarını analiz etmeden göreceli hareket oluşturmak için eklemlerle bağlı bağlantı veya zincir sistemi ile anlaşılabilir. Robot bağlantısının zamanın fonksiyonu ile bir sabit koordinat veya temel koordinat sistemine göre hareketinin analitik olarak incelenmesi, robot kinematiği olarak tanımlanabilir. Robot bağlantısının kinematiği, hız, sarsıntı, ivme vb. gibi yüksek türevlerinin çalışmasını da sağlar.

Kinematik bir zincir, bağlantıların kinematik çiftlerinden meydana gelir. Literatürde yer alan yöntemlerin başlıca farklılıkları, koordinat sistemlerinin birbirine eklenme tarzından kaynaklanmaktadır. Bu nedenle Denavit-Hartenberg parametreleri yaygın olarak kullanılmaktadır. Homojen dönüşüm matrisi tabanlı yöntemler, yerel koordinat sistemlerinin bağlantılara ve eklem değişkenlerine yerleştirilmesi için daha iyidir (Jha 2015).

3.1.6. Puma 560 robotik manipülatörünün kinematik modeli

İleri kinematik

Endüstriyel uygulamalar için oldukça uygun teknolojik özelliklere sahip olan ve 6 adet dönel ekleme sahip PUMA 560 robotunun (Şekil 3.11) ileri kinematiği gibi bir dönüşüm matrisinden (Denklem 3.5) elde edilebilir. Burada X , Y ve Z robotik manipülatörün uç noktasının koordinatları, d ve a Denavit-Hartenberg parametreleri, $c_i = \cos\theta_i$, $s_i = \sin\theta_i$, $c_{ij} = \cos(\theta_i + \theta_j)$ ve $s_{ij} = \sin(\theta_i + \theta_j)$ olmak üzere,

$$X = c_1(d_2s_{23} + a_3c_{23} + a_2c_2) - s_1d_3 \quad (3.15)$$

$$Y = s_1(d_2s_{23} + a_3c_{23} + a_2c_2) + c_1d_3 \quad (3.16)$$

$$Z = -(-d_4c_{23} + a_3s_{23} + a_2s_2) \quad (3.17)$$

$$a_x = c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \quad (3.18)$$

$$a_y = s_1(c_{23}c_4s_5 + s_{23}c_5) - c_1c_4c_5 \quad (3.19)$$

$$a_z = -s_{23}c_4s_5 + c_{23}c_5 \quad (3.20)$$

$$o_x = c_1[-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6] + s_1[-s_4c_5s_6 + c_4c_6] \quad (3.21)$$

$$o_y = s_1[-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6] + c_1[-s_4c_5s_6 + c_4c_6] \quad (3.22)$$

$$o_z = s_{23}(c_4c_5s_6 + s_4c_6) + c_{23}s_5s_6 \quad (3.23)$$

$$n_x = c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - s_1[s_4c_5s_6 + c_4s_6] \quad (3.24)$$

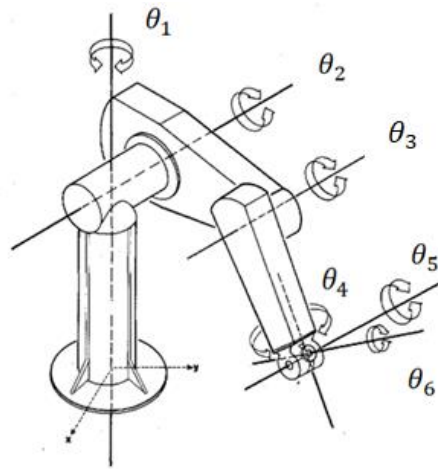
$$n_y = s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1[s_4c_5s_6 + c_4s_6] \quad (3.25)$$

$$n_z = s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \quad (3.26)$$

olarak elde edilir (Jha 2015).

Ters kinematik

İleri kinematik denklemleri, denklem (3.15) - (3.26) kullanarak, PUMA 560 robotik manipülâtörünün ters kinematiki aşağıdaki gibi türetilerek, θ_1 , θ_2 , θ_3 , θ_4 , θ_5 ve θ_6 eklem açıları elde edilir.



Şekil 3.11. PUMA 560 robotik manipülâtörü

$$\theta_1 = \text{atan2} \left(\pm \sqrt{(X^2 + Y^2 - d_3^2)}, d_3 \right) - \text{atan2}(X, Y) \quad (3.27)$$

$$\theta_2 = \text{atan2} \left(-Z, \pm \sqrt{X^2 + Y^2 - d_3^2} \right) - \text{atan2} \left(\pm \sqrt{a_3^2 + d_4^2 - b_2^2}, b_2 + a_2 \right) \quad (3.28)$$

Yukarıdaki 3.28 denkleminde yer alan b_2 ifadesi aşağıdaki şekilde yazılabilir.

$$b_2 = \frac{p^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2} \quad (3.29)$$

Burada $p = \sqrt{P_x^2 + P_y^2 + P_z^2}$ olmak üzere, θ_2 parametresi ayrıca aşağıdaki gibi yazılabilir:

$$\theta_2 = \text{atan2} \left(\pm \sqrt{X^2 - Y^2 - d_3^2}, Z \right) - \text{atan2} \left(\pm \sqrt{a_3^2 + d_4^2 - b_2^2}, b_2 + a_2 \right) - \frac{\pi}{2} \quad (3.30)$$

$$\theta_3 = \text{atan2} \left(\pm \sqrt{a_3^2 + d_4^2 - b_2^2}, b_2 \right) - \text{atan2}(d_4, a_3) \quad (3.31)$$

Robotik manipülâtörün küresel bileği olduğu durumda, kol ve bilek ayrılabilir. Kol için dönüşüm matrisi şu şekilde verilebilir:

$$\mathbf{R}_A = \begin{bmatrix} c_1 c_{23} & s_1 & -c_1 s_{23} \\ s_1 c_{23} & -c_1 & -s_1 s_{23} \\ -s_{23} & 0 & -c_{23} \end{bmatrix} \quad (3.32)$$

Kol için konum matrisi şu şekilde verilebilir:

$$\mathbf{P}_A = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} c_1(c_{23}a_3 - s_{23}d_4 + c_2a_2) - s_1d_3 \\ s_1(c_{23}a_3 - s_{23}d_4 + c_2a_2) + c_1d_3 \\ -s_{23}a_3 - c_{23}d_4 - s_2a_2 \end{bmatrix} \quad (3.33)$$

Küresel bilek için genel denklem, daha önceden verilen rotasyon matrisine, Z-Y-Z Euler açısının eklenmesiyle bulunabilir. Z-Y-Z Euler açısının genel gösterimi,

$$Z_1 Y_2 Z_3 = \begin{bmatrix} c_1 c_2 c_3 - s_1 s_3 & -c_3 s_1 - c_1 c_2 s_3 & c_1 s_2 \\ c_1 s_3 + c_2 c_3 s_1 & c_1 c_3 - c_2 s_1 s_3 & s_1 s_2 \\ -c_3 s_2 & s_2 s_3 & c_2 \end{bmatrix} \quad (3.34)$$

olmak üzere, küresel bilek için genel denklem,

$$Z_{\theta_4} Y_{-\theta_5} Z_{\theta_6} = \mathbf{G} \quad (3.35)$$

olarak elde edilir.

Bu sayede \mathbf{G} matrisinin elemanlarını aşağıdaki gibi elde edilir ($i=1, 2, 3$):

$$g_{1i} = (c_1 c_{23})r_{1i} + (s_1 c_{23})r_{2i} - s_{23}r_{3i} \quad (3.36)$$

$$g_{2i} = s_1 r_{1i} - c_1 r_{2i} \quad (3.37)$$

$$g_{3i} = -(c_1 s_{23})r_{1i} - (s_1 s_{23})r_{2i} - c_{23}r_{3i} \quad (3.38)$$

Buradan,

$$\theta_4 = \begin{cases} -\pi - \text{atan2}(g_{23}, g_{13}), & \text{eğer } \sigma_3 < 0 \text{ ise} \\ \text{atan2}(g_{23}, g_{13}), & \text{diğer durumlarda} \end{cases} \quad (3.39)$$

burada, $\sigma_3 = \pm \sqrt{g_{31}^2 + g_{32}^2}$ olarak verilmiştir.

$$\theta_5 = -\text{atan2}(\sigma_3, g_{33}) \quad (3.40)$$

$$\theta_6 = \begin{cases} -\text{atan2}(g_{32}, g_{31}), & \text{eğer } \sigma_3 < 0 \text{ ise} \\ \pi - \text{atan2}(g_{32}, g_{31}), & \text{diğer durumlarda} \end{cases} \quad (3.41)$$

Böylece ters kinematik ile eklemlere ait tüm açılar elde edilmiş olur.

3.1.7. Çarpışmasız yörünge planlama

Aynı istasyonda birden fazla robotik manipülatörün aynı iş üzerinde çalışması, çarpışma ve kilitleme gibi bazı teknik problemler ortaya çıkarmaktadır. Aynı çalışma uzayında

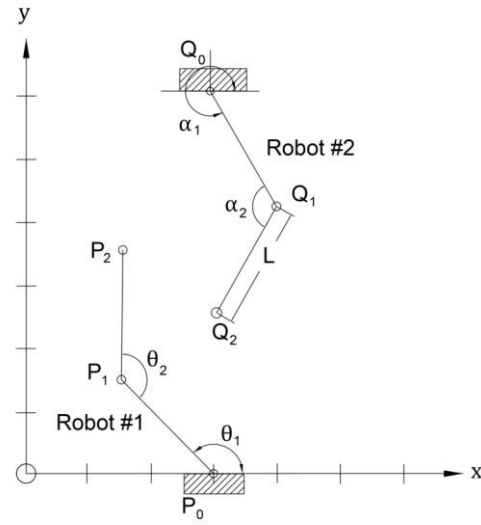
yer alan robotik manipulatörler yörünge hareketleri sırasında, hem birbirleriyle hem de çalışma uzayında yer alan engeller ile çarpışma tehlikesiyle karşı karşıya kalırlar. Bu durum çarpışma problemini ortaya çıkarmaktadır. Bunun yanı sıra robotik manipulatörlerden birinin, yörünge hareketi sırasında diğer bir robotik manipulatörün yörüngesine girecek olduğunda çarpışmayı engellemek amacıyla durması gerekebilir. Bu durum da kilitlenme problemini ortaya çıkarmaktadır. Kilitlenme problemi robotik manipulatörlerin aylak zamanının artmasına neden olmaktadır.

Literatürde robotların (robotik manipulatörler ve mobil robotlar) çarpışma problemi genel olarak iki şekilde çözülmektedir: Çevrimiçi ve çevrimdışı çizelgeleme. Çevrimiçi çizelgelemede robotların anlık olarak iletişimleri söz konusu olurken, çevrimdışı çizelgelemede robotlar arasında anlık bağlantı ve iletişim yoktur. Bu tez çalışmasındaki çarpışma kontrolü çevrimdışı olarak uygulanmıştır.

Rana ve Zalzala (1995), çevrimdışı çarpışmasız yörünge planlama için bir yöntem sunmuşlardır. Yöntemin gösterimi için ilk olarak iki adet 2 serbestlik derecesine sahip düzlemsel robot ele alınmıştır. Şekil 3.12’de robotik manipulatörler için x-y operasyonel uzayı gösterilmektedir. Burada ξ_1 (robot #1) için çalışma uzayı $C_1 = \theta_1 \times \theta_2 \in \mathbb{R}^2$ olup, ξ_2 (robot #2) için çalışma uzayı $C_2 = \alpha_1 \times \alpha_2 \in \mathbb{R}^2$ olarak gösterilmiştir. ξ_2 , ξ_1 için engel olarak görünürken, aynı şekilde ξ_1 de ξ_2 için engel olarak görülmektedir.

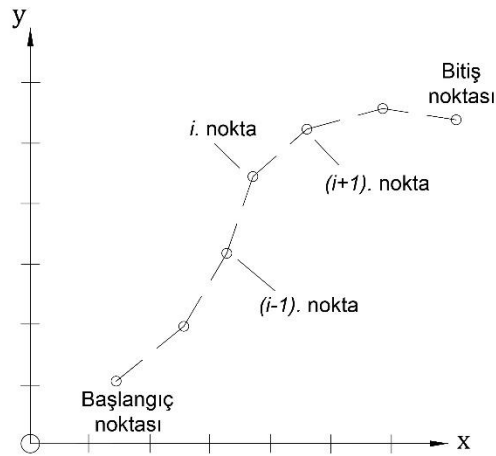
ξ_1 ve ξ_2 için yörüngeler sırasıyla C_1 ve C_2 ‘deki noktalar dizisi olarak dikkate alınmıştır. Robotun hedef konuma ulaşması için bu noktalardan geçmesi gerekmektedir. Şekil 3.13’te ardışık noktalar ile verilen çalışma uzayındaki bir tutucu uç yörüngesi gösterilmektedir.

Yörünge #1 ve yörünge #2’deki noktalar, ξ_1 ve ξ_2 ’in aynı zaman biriminde buldukları pozisyonlara karşılık gelmektedir. İki robot kolu arasındaki çarpışmayı belirlemek için, her iki yörüngeye karşılık gelen noktalar arasındaki çarpışma belirlenmelidir.

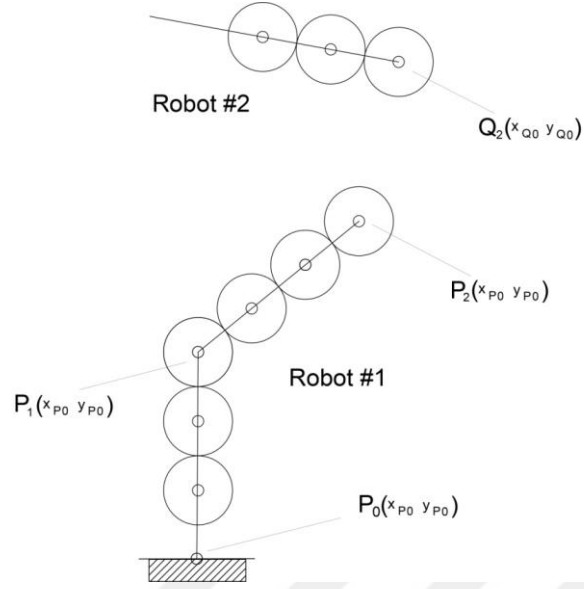


Şekil 3.12. İki bağlantılı iki düzlemsel robot için çalışma uzayı (Rana ve Zalzala 1995)

Robot kollarının birbiriyle çarpışıp çarpışmadığını anlamak için, bağlantılar üzerinde bulunduğu varsayılan dairelerin birbirine dokunup dokunmadığına ya da üst üste binip binmediğine bakılması yeterlidir. Belirli bir büyüklükteki bağlantı için bu dairelerin boyutu, adedi ve konumu ile birlikte, dairelerin birbirlerine olan yakınlığı ile ilgili gereken doğruluk derecesi, Beaumont ve Crowder (1989) tarafından ele alınmıştır. Her iki koldaki tüm dairelerin merkezleri arasındaki mesafeler önceden bilinmektedir. Belirtilen mesafeler, dairelerin yarıçapının iki katından daha azsa, çarpışma meydana gelir. Şekil 3.14, birbirlerine dokunan daireler ile temsil edilen robot bağlantılarını göstermektedir.

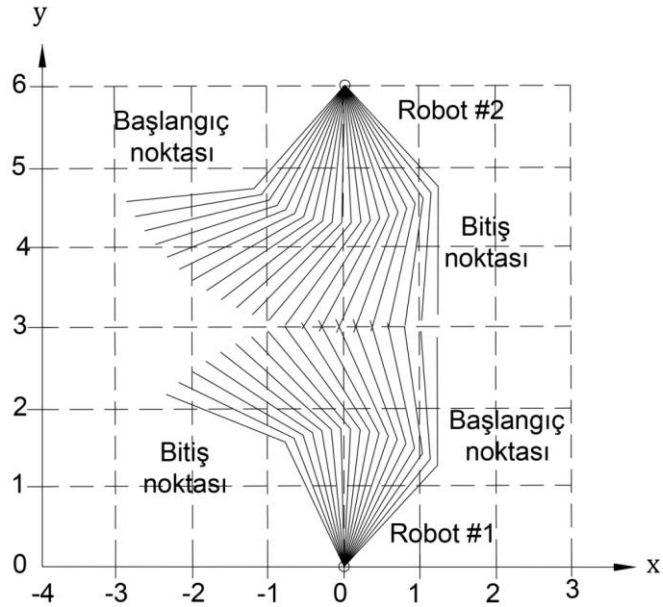


Şekil 3.13. Bir robotik manipülatöre ait tutucu uç elemanın çalışma uzayında izlediği yörüngenin noktalar ile gösterimi (Rana ve Zalzala 1995)

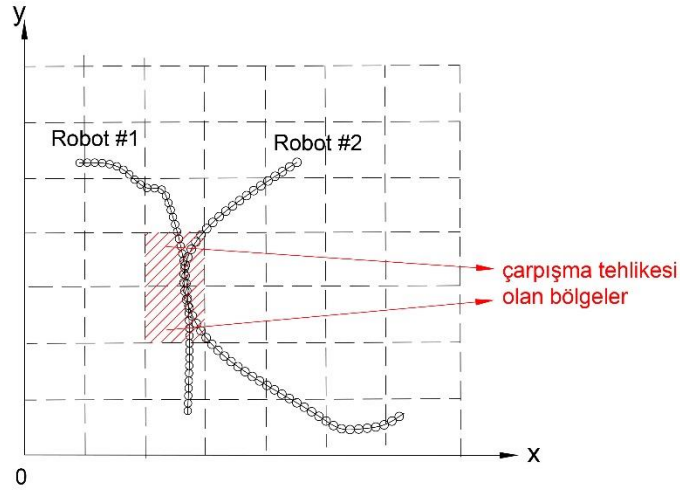


Şekil 3.14. Robotik manipülatörlerin birbirine değen daireler ile gösterimi (Rana ve Zalzala 1995)

Rana ve Zalzala (1995), çarpışmasız yörünge planlama probleminin çözümü için GA-tabanlı bir arama tekniğini kullanılmışlardır. Şekil 3.15 ve 3.16 'de sırasıyla robot kollarının yörüngeleri ve tutucu uç elemanın yörüngesi gösterilmiştir.



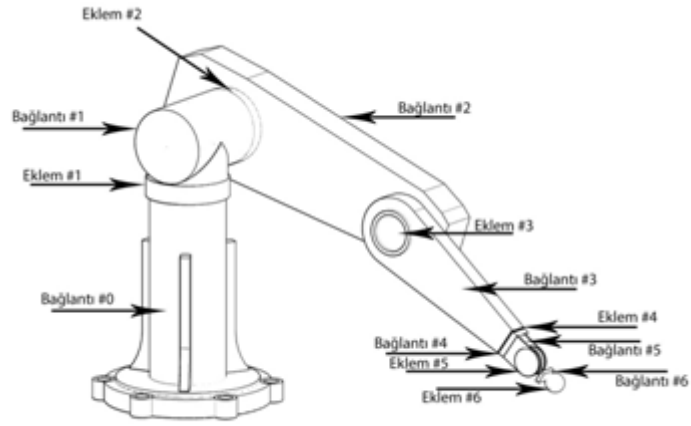
Şekil 3.15. Robot kollarına ait yörüngeler (Rana ve Zalzala 1995)



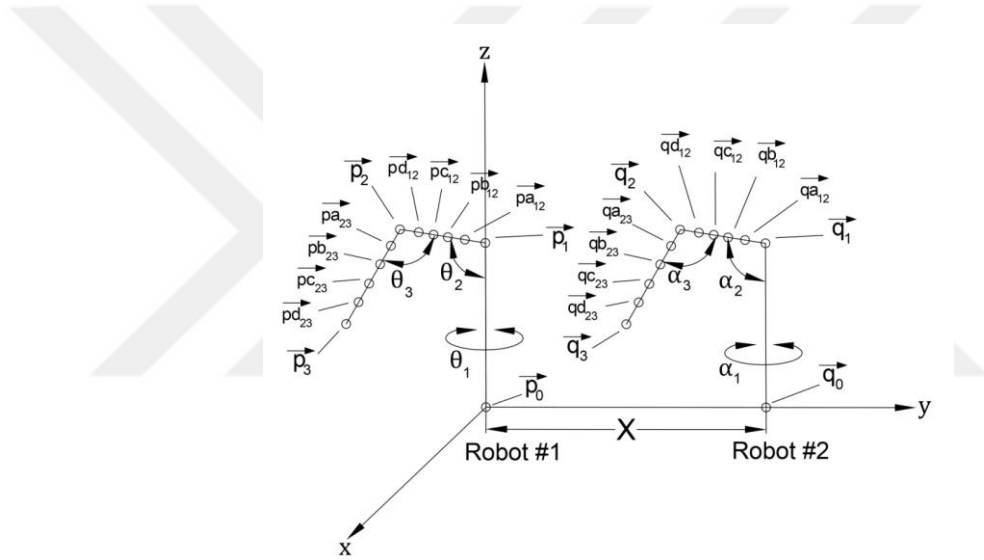
Şekil 3.16. Robot kollarına ait tutucu uç elemanların yörüngeleri (Rana ve Zalzala 1995)

Yazarlar düzlemsel robotların çarpışmasız yörünge planlaması için sundukları bu yöntemi üç boyutlu uzayda yer alan iki adet PUMA®-benzeri robotik manipülatörler için genişletmişlerdir. PUMA® serisi robotik manipülatörler döner eklemlili (6R) 6 serbestlik derecesine sahip robotik manipülatörlerdir. İlk üç derece, bileği belirli bir konuma almak için kullanılır. Son üç derece ise bileğin istenilen konumda yönelimi için kullanılır. Bu nedenle esasen ilk üç serbestlik derecesi çarpışmaya neden olmaktadır. Çarpışma kontrolü için yazarların makalesinde sadece ilk üç bağlantı değerlendirilmiş olup bu tez çalışmasında da robotik manipülatörlerin ilk üç bağlantısı çarpışma kontrolü için göz önünde bulundurulmuştur. Şekil 3.17'de PUMA®-560 serisi bir robotik manipülatör gösterilmekte olup, Şekil 3.18'de ise bu robotik manipülatörün bağlantı elemanları gösterilmiştir.

Bu çalışmada robotik manipülatörlerin çarpışmasını engellemek amacıyla, Rana ve Zalzala'nın (1995) çok-kollu robotlar için çarpışmasız yörünge planlaması adına sundukları evrimsel algoritmadan yararlanılmıştır. Çalışmada kullanılan robotik manipülatörler de PUMA-benzeri robotik manipülatörlerdir. Çarpışma kontrolü için kullanılacak iki robotik manipülatör, birbirinin özdeşi olarak kabul edilmiştir.



Şekil 3.17. PUMA-benzeri robotik manipülör



Şekil 3.18. Robotik manipülörlere ait bağlantı elemanları (Rana ve Zalzala 1995)

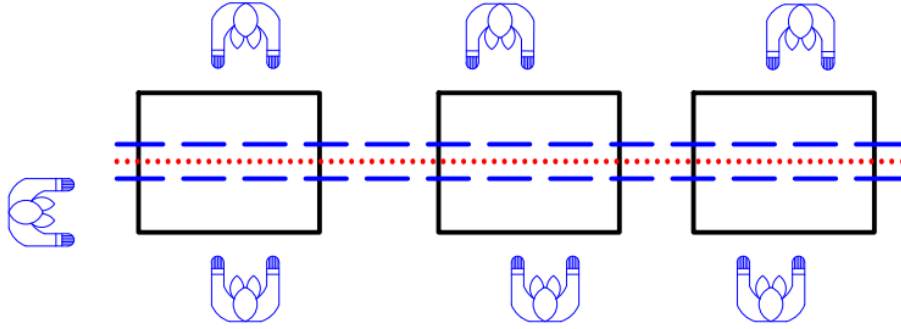
3.1.8. Çok-işçili montaj hattı dengeleme probleminin matematiksel modeli

Bu tez çalışmasındaki problem, çarpışmasız ve kilitlenmesiz robotik montaj hattı dengeleme problemidir. Problem, bazı farklılıklar olmasına rağmen yapı olarak çok-işçili montaj hattı dengeleme problemine benzemektedir.

Çok-işçili montaj hattı dengeleme problemleri, ürün özelliklerine göre her bir iş istasyonuna birden fazla çalışanın atanabileceği yeni bir geliştirilmiş montaj hattı dengeleme problemidir. Bu tür dengeleme problemleri tipik olarak, otomotiv endüstrisi

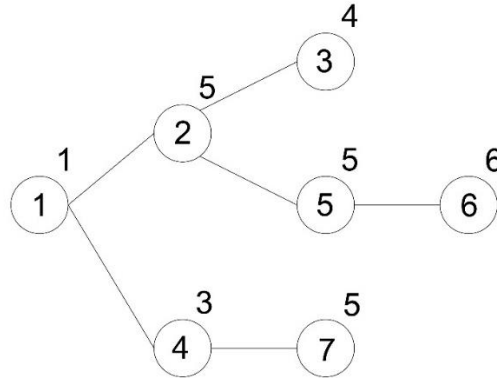
gibi büyük boyutlu ve yüksek hacimli ürünlere sahip sanayilerde ortaya çıkar (Dimitriadis 2006).

Çok-işçili bir montaj hattının (ÇMH), basit bir montaj hattına göre, montaj hattının uzunluğunun, çıktı süresinin, alet ve demirbaşların maliyetinin, malzeme taşıma, işçi hareketi ve kurulum süresinin azaltılması gibi önemli avantajları vardır. Bu avantajlar, ÇMH'nin büyük boyutlu ürünlerin montajında kullanılması için yeterlidir (Fattahi ve ark. 2011). Çok-işçili montaj hattı yerleşiminin tipik bir örneği Şekil 3.19'da gösterilmiştir.



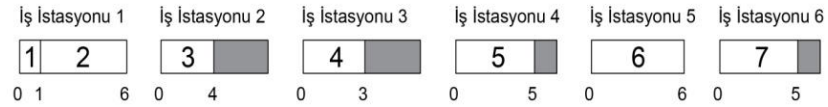
Şekil 3.19. Çok-işçili montaj hattı yerleşimi

Çok-işçili bir montaj hattı kullanımındaki ana hedef, hattın toplam etkinliğini optimize ederken, basit montaj hattının hat uzunluğunu azaltmaktır (Dimitriadis 2006). Şekil 3.20, her bir iş istasyonunda çevrim süresi 6 olan ve en fazla üç işçinin kabul edildiği montaj hattı dengeleme probleminin bir örneğini göstermektedir. Şekilde gösterildiği gibi, montaj hattında yedi adet montaj işlemi yürütülmektedir. Her işlem bir daire ile gösterilmiştir. İşlem numaraları dairelere dahil edilir ve işlem süreleri dairelerin üstünde gösterilir.

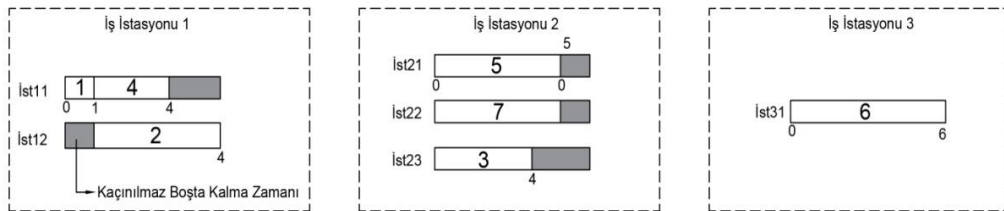


Şekil 3.20. Çok-işçili bir montaj hattı örneğine ait öncüllük ilişkileri

Bu problem, montaj hattı dengelemeye ait iki yaklaşımla çözülür. Şekil 3.21 ve 3.22, bu iki yaklaşımla elde edilen en uygun çözümü göstermektedir. Şekil 3.21, basit montaj hattı dengeleme yaklaşımı ile elde edilen en uygun çözümü gösterirken, Şekil 3.22, çok-işçili montaj hattı yaklaşımı ile elde edilen en uygun çözümü göstermektedir. Her işlem için başlangıç zamanı ve bitiş zamanı, işleme ait çubuğun yanında gösterilir. Gölge dikdörtgenler iki ardışık işlem arasındaki kaçınılmaz gecikmeyi veya çevrim süresinin sonunda boşa kalma süresini belirtir (Fattahi ve ark. 2011).



Şekil 3.21. Basit montaj hattı için görev atamaları



Şekil 3.22. Çok-işçili montaj hattı için görev atamaları

Kim ve ark. (2009) tarafından önerilen iki taraflı montaj hattı dengeleme probleminin matematiksel modeli Fattahi ve ark. (2011) tarafından ÇMHDP'yi formüle etmek için geliştirilmiştir.

Çok-işçili bir montaj hattında, önceden belirlenmiş bir çevrim süresi içinde, bir dizi belirlenmiş görev (montaj operasyonları) çok-işçili iş istasyonlarına atanır. Ürün özelliklerine göre her bir iş istasyonuna birden fazla işçinin atanma olasılığı vardır. Her iş istasyonuna maksimum sayıda işçi atanabilir, ancak bu durum önceden tanımlanmış bir M_{max} sayısı (her bir iş istasyonuna atanabilecek maksimum işçi sayısı) ile sınırlandırılır. Önerilen model optimal iş istasyonu sayısını belirlerken, montaj hattının toplam etkinliği, toplam işçi sayısı ve toplam boşta kalma süresi, basit montaj hattının optimal çözümü ile aynı kalmaktadır. Önerilen modelde, aşağıdaki varsayımlar dikkate alınmıştır.

Seri hat düzeni üzerinde tek tip bir ürünün seri üretimi göz önünde bulundurulmakta ve çevrim süresi montaj hattı boyunca sabit kalmaktadır. Her i görevi, atandığı istasyonda deterministik ve bağımsız bir t_i işlem süresine sahiptir. Her görevin öncülleri önceden bilinmektedir. İş istasyonları arasındaki taşıma süresi göz ardı edilir. Paralel görevler ve paralel istasyonlara izin verilmez.

Matematiksel model aşağıdaki gibi tanımlanmıştır. İndislerde, i ve h görevleri, j iş istasyonlarını, k ve l işçileri belirtmektedir. Parametreler ve karar değişkenleri aşağıdaki gibidir.

Parametreler

J	İş istasyonlarının kümesi, $J = \{1, 2, \dots, j, \dots, n\}$
K	Her iş istasyonundaki işçilerin kümesi, $K = \{1, \dots, k, \dots, M_{max}\}$
$P(i)$	i görevinin hemen öncesindeki öncüllerinin kümesi
$P_a(i)$	i görevinin tüm öncüllerinin kümesi
$S(i)$	i görevinin hemen sonrasındaki ardıllarının kümesi
$S_a(i)$	i görevinin tüm ardıllarının kümesi

Ψ	Pozitif olan büyük bir sayı
$THL = \left\lceil \frac{\sum_{i \in I} t_i}{Ct} \right\rceil$	Montaj hattındaki teorik en az işçi sayısı

Karar Değişkenleri

x_{ijk}	i görevi j istasyonundaki k işçisine atanırsa 1, aksi halde 0 değerini alır.
st_i	i görevinin başlangıç zamanı
w_{jk}	k işçisi j istasyonunda kullanılırsa 1, aksi halde 0 değerini alır.
ws_{jk}	k adet işçi j istasyonunda kullanılırsa 1, aksi halde 0
y_{ih}	eğer i görevi, aynı çalışanın görev ataması sırasındaki görev h 'den daha erken yürütülürse 1, aksi halde 0 değerini alır.

Kolaylık sağlamak için, m görevinin öncüllük grafiğinin tek ve son bir düğümü olduğu varsayılır. Eğer birden fazla son düğüm varsa, görev süresi 0 olan sanal bir düğüm, tüm son düğümlerin bir ardılı olarak tanımlanır ve m etiketi artırılır. Önerilen matematiksel formülasyon aşağıdaki gibidir:

Amaç Fonksiyonu:

$$\text{Min} \sum_{j \in J} \sum_{k \in K} w_{jk} + \frac{1}{\delta} \sum_{j \in J} \sum_{k \in K} j \cdot x_{mjk}, \quad \delta = n \cdot M_{max} + 1 \quad (3.42)$$

Kısıtlar:

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \quad \forall i \in I \quad (3.43)$$

$$\sum_{j \in J} \sum_{k \in K} j \cdot x_{hjk} \leq \sum_{j \in J} \sum_{k \in K} j \cdot x_{ijk}, \quad \forall i \in I, h \in P(i) \quad (3.44)$$

$$st_i + t_i \leq Ct, \quad \forall i \in I \quad (3.45)$$

$$st_i - st_h + \Psi. \left(1 - \sum_{k \in K} x_{hjk} \right) + \Psi. \left(1 - \sum_{k \in K} x_{ijk} \right) \geq t_h, \quad \forall i \in I, h \in P(i), j \in J \quad (3.46)$$

$$st_h - st_i + \Psi. (1 - x_{hjk}) + \Psi. (1 - x_{ijk}) - \Psi. (1 - y_{ih}) \geq t_i, \quad \forall i \in I, h \in \{r | r \in I - (P_a(i) \cup S_a(i)) \text{ ve } i < r\}, j \in J, k \in K \quad (3.47)$$

$$st_i - st_h + \Psi. (1 - x_{hjk}) + \Psi. (1 - x_{ijk}) - \Psi. (y_{ih}) \geq t_i, \quad \forall i \in I, h \in \{r | r \in I - (P_a(i) \cup S_a(i)) \text{ ve } i < r\}, j \in J, k \in K \quad (3.48)$$

$$\sum_{i \in I} x_{ijk} - m. w_{jk} \leq 0, \quad \forall j \in J, k \in K \quad (3.49)$$

$$\sum_{k \in K} k. w_{sjk} - \sum_{k \in K} w_{jk} = 0, \quad \forall j \in J \quad (3.50)$$

$$w_{j(k+1)} \leq w_{jk}, \quad \forall j \in J, k = 1, 2, \dots, M_{max} - 1 \quad (3.51)$$

$$st_i \geq 0, \quad \forall i \in I \quad (3.52)$$

$$x_{ijk} \in \{0,1\} \quad \forall i \in I, j \in J, k \in K \quad (3.53)$$

$$y_{ih} \in \{0,1\}, \quad \forall i \in I, h \in \{r | r \in I - (P_a(i) \cup S_a(i)) \text{ ve } i < r\} \quad (3.54)$$

$$w_{jk} \in \{0,1\}, \quad \forall j \in J, k \in K \quad (3.55)$$

Basit montaj hattında olduğu gibi çok-işçili montaj hattında, basit montaj hattının toplam etkinliğine ulaşmak için, işçi sayısı birincil hedef olarak optimize edilmeli ve sonrasında çok-işçili iş istasyonlarının sayısı en aza indirilmeye çalışılmalıdır. Amaç fonksiyonu (3.42)'deki birinci terim, hattaki toplam işçi sayısına karşılık gelir. İkinci terim, açılan çok-işçili iş istasyonlarının toplam sayısını temsil eder. Ağırlıklandırma faktörü $\frac{1}{\delta} = \frac{1}{nM_{max}+1}$, ikinci terimin daima 1 'den daha küçük olması nedeniyle, ikinci terimin ikincil hedef olmasını sağlar. Denklem (3.43), atama kısıtlamasıdır ve her görevin bir iş istasyonunda tam olarak bir işçiye atanmasını sağlar. Denklem (3.44), öncelik kısıtlamasıdır ve görevler arasındaki tüm öncelik ilişkilerinin yerine getirilmesini sağlar. Denklem (3.45), çevrim süresi kısıtlamasıdır ve her görevin döngü süresinin bitiminden önce bitmesini sağlar. Denklem (3.46), (3.47) ve (3.48), sıralama

kısıtlamalarını kontrol eder. Denklem (3.49), her bir i ve h görevi için, eğer h görevi, i görevinin bir öncülü ise, öncelik sırasını doğrulamak için görev i 'nin, h görevi tamamladıktan sonra başlatılmasını sağlar. Eğer i ve h görevleri, aralarında öncelik ilişkisine sahip değilse ve aynı işçiye atanırsa, denklem (3.47) veya (3.48) aktif hale gelir. Eğer i , h 'den önce atanırsa ($y_{ih} = 1$), denklem (3.47), $st_h - st_i \geq t_i$ olur. Diğer durumda h , i 'den önce atanırsa, denklem (3.48), $st_i - st_h \geq t_h$ olur. Buna rağmen, iki görev farklı iş istasyonlarına atanırsa, denklem (3.46), (3.47), (3.48)'e yeteri kadar büyük bir sayı olan Ψ (örnek olarak, $\Psi = C_{time}$) eklenerek bu görevler belirtilen kısıtlarda değerlendirilmez. Denklem (3.49), Deckro (1989) 'nun basit bir montaj hattı dengeleme problemi için geliştirdiği işçi kısıtlamasıdır. Bu kısıt, çok-işçili montaj hattı dengeleme problemleri için uyarlanmıştır. Eğer bir görev j istasyonundaki k işçisine atanırsa, w_{jk} , 1 'e eşit olmalıdır. Denklem (3.50), her bir iş istasyonunda kullanılan işçi sayısını doğrular. Eğer j istasyonunda k adet işçi kullanıldıysa, ws_{jk} , 1 'e eşit olmalıdır. Denklem (3.51), çok-işçili bir iş istasyonunda işçi indeks sırasını gözlemler. Bu kısıt, işçilerin artan bir şekilde yüklenmesini sağlar. Denklem (3.52), tüm görevlerin başlama zamanının 0 'dan büyük veya 0 'a eşit olması gerektiğini belirtir. Denklem (3.53, 3.54, 3.55), tamsayı değişkenleri gösterir.

Bu tez çalışmasında ise montaj hattında insanların yerini robotik manipülatörler almaktadır. Her bir iş istasyonundaki robotik manipülatör sayısı iki adet olarak sabitlenmiştir. Bu durum amaç fonksiyonunun birinci teriminde yer alan hatlardaki toplam işçi sayısını değiştirmektedir. Her iş istasyonunda iki adet robotik manipülatör yer aldığından dolayı hattaki toplam robotik manipülatör sayısı, açılan iş istasyonlarının toplamının iki katı olarak ortaya çıkmaktadır. Denklem (3.43)'te verilen atama kısıdı ise bir görev grubunun yalnızca bir istasyona atanması ve bir görev grubuna ait bir alt görevin yalnızca tek bir robotik manipülatöre atanmasını sağlayacak şekilde özelleştirilmelidir. Diğer tüm kısıtlar tezdeki probleme uygun olmakla birlikte, matematiksel modele çarpışma kontrolü kısıtı eklemek gerekmektedir. Bu kısıt tamamen robotik manipülatörlerin kinematik bilgisi ile ilgili olduğundan, modeli doğrusal olmayan programlama durumuna getirmektedir. Bundan dolayı tezdeki problemin çözümü için doğrusal tamsayılı programlama modeli yerine diziyi rastgele

sıralama (RPA) yöntemi ile Rana ve Zalzala 'nın (1995) çarpışma kontrolü yönteminin birlikte kullanıldığı hibrit bir sezgisel arama yöntemi geliştirilmiştir.

Basit montaj hattı dengeleme problemi NP-zor olarak bilinir (Ege ve ark. 2009, Yeh ve Kao 2009). Dolayısıyla, BMHDP 'nin geliştirilmiş hali olan ÇMHDP de, güçlü bir şekilde NP-zordur.

3.2. Yöntem

3.2.1. Dizi elemanlarını rastgele sıralama yöntemi

Problemdede yer alan görev gruplarının ataması, dizinin elemanlarını rastgele sıralama (RPA) yöntemiyle (Cormen ve ark. 2001) yapılmaktadır. Birçok rassal algoritma, verilen bir diziye ait elemanları rassal olarak sıralayarak yeni bir dizi elde etmeyi sağlar (Rassallığı kullanmanın birçok yolu vardır). Genellik kaybı olmaksızın 1'den n 'ye kadar olan elemanları içeren bir A dizisinin verildiği varsayalım. Burada amaç dizinin rastgele bir sıralamasını üretmektir.

Yaygın bir yöntem, dizinin her bir $A[i]$ elemanını rassal bir öncelik değeri $P[i]$ ile başka bir diziye atamak ve daha sonra bu dizinin elemanlarını önceliklere göre sıralamayı sağlar. Yönteme ait yalancı kod Şekil 3.23'te verilmiştir. Örnek olarak bir $A = (1, 2, 3, 4)$ dizisi ele alalım. Bu dizi için rastgele öncelik değerleri $P = (25, 4, 56, 17)$ üretilmiş olsun. Öncelik değerleri kullanılarak sıralanmış yeni bir $B = (2, 4, 1, 3)$ dizisi elde edilir. Buradaki sıralama öncelik değerlerinin olduğu dizinin en küçük değerli elemanının olduğu sıradaki elemana denk gelen A dizisi elemanı B dizisinde en başa gelir. Prosedür aynı şekilde devam ettirilerek yeni bir sıralanmış dizi elde edilir.

RASTGELE-SIRALA (A)

```
1  $n \leftarrow A$  dizisinin uzunluđu
2 for  $i \leftarrow 1$ 'den  $n$ 'ye kadar
3     do  $P[i] = \text{RANDOM}(1, n^3)$ 
4  $A$ 'yı sırala ( $P$ 'yi sıralama kriteri olarak kullan)
```

Şekil 3.23. Rastgele sıralama yöntemine ait yalancı kod

Şekilde 3.23'te yer alan üçüncü satır, 1 ile n^3 arasında rastgele bir sayı seçer. P 'deki tüm önceliklerin benzersiz olmasını sağlamak için 1 ile n^3 gibi geniş bir aralık kullanılır (Cormen ve ark. 2001).

3.2.2. Görev atama ve uygunluk fonksiyonu hesabı

Robotik montaj hattında yer alan robotik manipülatörler ve iş istasyonları için en önemli unsurlardan biri amaç fonksiyonu ya da uygunluk fonksiyonu elde etmede kullanılan görev atama yöntemidir. Bu tezdeki probleme ait varsayımlar aşağıdaki gibidir:

- Bir önceki alt bölümde verilen matematiksel modelden farklı olarak burada görev yerine görev grupları ve bu görev gruplarının içerdiği alt görevler tanımlanmıştır.
- Alt görevler, robotik manipülatörlerin erişeceği ve işlem yapacağı noktalardır ve bu noktaların koordinatları önceden bilinmektedir.
- Robotik manipülatörler aynı teknolojik özelliklere sahiptir ve özdeş olarak belirlenmiştir ancak sabitlendikleri konumlar farklıdır.
- Görev grupları arasındaki öncüllük ilişkileri bilinmektedir ve değişken değildir.
- Görev gruplarında yer alan alt görevlerin süreleri robotik manipülatörlerin buldukları bir önceki konuma göre değişmekte olup istasyonlardan bağımsızdır. Bir görev grubunda yer alan alt görevin tamamlanma süresi, robotik manipülatörün bir önceki alt görevi tamamlanmasından sonra bir sonraki alt göreve varması arasında geçen süre ve bu süreye dahil olan sabit bir işlem süresi ile belirlenir.

- Her bir robotik manipülatör, tüm görev gruplarında yer alan alt görevleri işleyebilme yeteneğine sahiptir.
- Tüm görev grupları her bir istasyonda işlenebilir.
- Her istasyonda iki adet robotik manipülatör yer almaktadır.
- Malzeme taşıma, yükleme ve boşaltma sürelerinin yanı sıra kurulum ve alet değiştirme süreleri ihmal edilebilir veya etkinlik sürelerine dahil edilir. Böyle bir robotik hat üzerinde aletler genellikle, bir istasyonda alet değişikliklerinin en aza indirileceği şekilde tasarlanmıştır.
- Robotların satın alma maliyeti dikkate alınmaz.
- Montaj hattı, tek tip bir ürün için dengelenmiştir.
- Tüm girdi parametreleri kesin olarak bilinmektedir.
- Bir görev grubu, iki veya daha fazla iş istasyonu arasında bölünemez.
- Tüm görev grupları ve alt görevler işlenmelidir.
- Montaj hattının seri olduğu kabul edilir ve parçanın taşınması anında gerçekleşir.
- Bir görev grubunda yer alan bir alt görev yalnızca tek bir robotik manipülatör tarafından işlenebilir, yani alt görevler robotik manipülatörler arasında bölünemez. Özetle bir alt görev yalnızca bir robotik manipülatöre atanabilir.
- Bir istasyondaki toplam işlem süresi çevrim süresini aşamaz.
- Çevrim süresi sabittir ve önceden bilinmektedir.
- Her bir robotik manipülatörün hareket ve işleme hızı sabittir ve önceden bilinmektedir.
- Robotik manipülatörlerin çalışma uzayları sınırlıdır.
- Robotik manipülatörlere ait ileri ve ters kinematik bilgileri önceden bilinmektedir.

Bu tez çalışmasında robotik manipülatörlere alt görevlerin atanması belirli öncül-ardıl ilişkilerine göre yapılmıştır. Yöntem ilk olarak n adet görev grubu için, öncül-ardıl ilişkilerine göre rassal olarak belirli bir atama sırası belirler. Daha sonra bu atama sırasına göre yine aynı şekilde rassal olarak görev grupları, istasyonlar boyunca sabit olan çevrim süresi aşılmayacak şekilde istasyonlara atanır.

Yöntem bu atama işlemlerini yaparken her görev grubu atamasından sonra çarpışma kontrolü yapar. Çarpışma kontrolü Rana ve Zalzala'nın (1995) çarpışma kontrolü yöntemiyle yapılmaktadır.

Görev gruplarının her biri belirli sayıda alt görevlerden oluşmaktadır. Bu alt görevler istasyonda bulunan robotik manipülatörlere sırasıyla atanır. Bir görev grubuna ait alt görevlerin tamamı atandıktan sonra çarpışma kontrolü gerçekleşir. Eğer mevcut atamada çarpışma meydana geliyorsa görev grubundaki alt görevler rassal olarak robotik manipülatörlere tekrar atanır. Bu durum çarpışma durumu ortadan kalkana kadar tekrarlanır.

Bir görev grubunda çarpışma meydana gelmiyorsa diğer görev grubundaki işlerin robotik manipülatörlere atanması işlemi devam eder. Önerilen yöntem ile çarpışmasız sürekli hareket sağladığından, kilitlenme problemi de doğal olarak çözülmüş olur.

Bu şekilde tüm görev grupları istasyonlara çarpışma durumu olmayacak şekilde atandıktan sonra her bir iterasyonda uygunluk fonksiyonu U , hesaplanır. Her iterasyonda hesaplanan uygunluk fonksiyonu (3.60), iki farklı (3.57) ve (3.58) terim içermektedir.

İlk terim istasyonlardaki robotlara atanan görev grupları arasındaki aylak zamanı içerir. t_{ijk} , $i = \{1, 2, \dots, r\}$ robotunun $j = \{1, 2, \dots, m\}$ istasyonundaki $k = \{1, 2, \dots, n\}$ görevini tamamlama süresi olmak üzere,

$$s_{jk} = |t_{1jk} - t_{2jk}| \quad (3.56)$$

ifadesi bir istasyondaki bir görev grubu bittiğinde oluşan aylak zamanı göstermektedir. Bu ifade, tüm istasyonlar ve istasyonlara atanan görev grupları için aşağıdaki formül (3.57) ile hesaplanır ve uygunluk fonksiyonunun ilk terimini oluşturur.

$$\sum_{k=1}^n \sum_{j=1}^m s_{jk} \quad (3.57)$$

Uygunluk fonksiyonun ikinci terimi ise çevrim süresi ile bir istasyona atanan tüm görev gruplarının o istasyondaki tamamlanma süresi arasındaki farktır. T_j , j istasyonuna atanan görev gruplarını belirten küme olmak üzere, bir istasyon için uygunluk fonksiyonunun ikinci terimi aşağıdaki formül (3.58) ile hesaplanır:

$$(3C_{time} - \sum_{j=1}^m \Delta_j) \quad (3.58)$$

Burada Δ_j terimi,

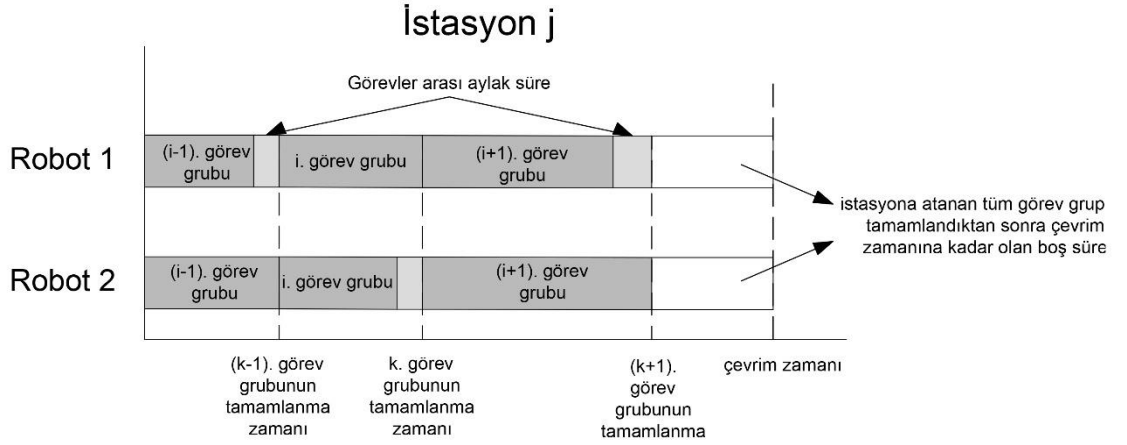
$$\Delta_j = \sum_{k=1}^{|T_j|} \max\{t_{1jk}, t_{2jk}\} \quad (3.59)$$

denklemleri ile bulunur.

Böylece uygunluk fonksiyonunda yer alan iki terim toplamından fonksiyonun değeri hesaplanır. U_{aday} , uygunluk fonksiyonu aday değeri olup, fonksiyonun değeri denklem (3.60) ile elde edilir:

$$U_{aday} = \sum_{k=1}^n \sum_{j=1}^m s_{jk} + \left(3C_{time} - \sum_{j=1}^m \Delta_j \right) \quad (3.60)$$

Uygunluk fonksiyonu için kullanılan yaklaşım, Şekil 3.24'te verilen Gantt diyagramı üzerinde gösterilmiştir. Burada amaç uygunluk fonksiyonu değerini en küçükmektir. Her bir iterasyonda bulunan U_{aday} değeri, iterasyon sonunda o ana kadar bulunan en iyi uygunluk fonksiyonu değeri olan U_{eniye} ile karşılaştırılır. Eğer $U_{aday} \leq U_{eniye}$ ise, mevcut iterasyondaki U_{aday} değeri, U_{eniye} olarak seçilir. Bu işlem, önceden belirlenen iterasyon sayısına ulaşıncaya kadar devam eder. Böylece iş istasyonlarındaki aylak süreleri azaltmaya yönelik bu yöntem ile montaj hattı dengeli bir duruma getirilmeye çalışılır.



Şekil 3.24. Bir istasyonda atanan görevler ve aylak süreleri gösteren çizelge

Materyal bölümünde verilen matematiksel modele ek olarak, bu tezde ele alınan problem için bazı kısıtların eklenmesi kaçınılmazdır. İlk olarak matematiksel modelde yer alan ve bir görevin yalnızca bir iş istasyonundaki işçiye atanması kısıtı aşağıdaki gibi iki kısıta dönüştürülmüştür:

$$\sum_{j=1}^m y_{jk} = 1, \quad \forall k \quad (3.61)$$

$$\sum_{i=1}^r \sum_j^m x_{ijkl} = 1, \quad \forall k, \forall l \quad (3.62)$$

Burada y_{jk} değişkeni, ikili değişken olup, eğer k görev grubu j istasyonuna atanıyorsa 1, aksi halde 0 değerini alır. x_{ijkl} değişkeni de ikili değişken olup, k görevinin l alt görevi j istasyonundaki i robotuna atanıyorsa 1, aksi halde 0 değerini alır. Denklem (3.61) bir k görev grubunun yalnızca tek bir j istasyonuna atanmasını garanti ederken, denklem (3.62), bir k görev grubuna ait l alt görevinin j istasyonundaki yalnızca tek bir robota atanmasını sağlar.

ÇAKIŞMA-KONTROLÜ

```
1 //Girdi: Robotik manipülatörlere ait yörüngeler
2 //Girdi: Bağlantı elemanlarının uzunlukları
3 //Girdi: Kinematik bilgisi
4 //Girdi: Eklem ve bağlantıların  $t$  anındaki konumları
5 Robotik manipülatörlere ait bağlantıları eşit zaman aralıkları ile konumlandır ( $n$  nokta)
6 ( $R1_t^i$  ve  $R2_t^j$  sırasıyla 1. ve 2. robotun  $t$ . zaman birimindeki koordinatları)
7 for  $t = 1$ 'den  $T$ 'ye kadar
8     for  $i = 0$ 'dan  $n$ 'ye kadar
9         for  $j = 0$ 'dan  $n$ 'ye kadar
10            if  $R1_t^i \neq R2_t^j$  ise
11                Çakışma durumu yok
12            elseif
13                Çakışma var
14                break
15            end
16        end
17    end
18 end
```

Şekil 3.25. Çarpışma kontrolüne ait sözde kod

Çarpışma durumunu kontrol eden kısıt ise birim zamanda bir robotik manipülatöre ait önceden belirlenmiş ve robotik manipülatörün eklem ve bağlantılarının üzerinde yer alan daireleri temsil eden koordinatların, tek tek diğer robotik manipülatöre ait tüm dairelerin koordinatlarıyla karşılaştırmasını yapan kısıttır (3.63).

O_i , robotik manipülatörler üzerinde yer alan dairelerin kümesi ve T kümesi bir istasyonda yer alan iki robotun kendilerine atanan görev gruplarını işleme zamanlarının kümesi olmak üzere,

$$|\varphi_{1jof}^{xyz} - \varphi_{2juj}^{xyz}| > 0, \quad \forall j \in m, \forall f \in T, \forall o \in O_1, \forall u \in O_2 \quad (3.63)$$

Burada φ_{ijof}^{xyz} değişkeni, j . istasyondaki i . robotun f . zaman birimindeki o . dairesinin üç boyutlu uzaydaki (x,y,z) koordinatlarını temsil etsin. Bu kısıdı daha açık bir şekilde anlatmak gerekirse, φ_{1jof}^{xyz} (o , robotik manipülatörün son dairesi olsun) değişkeni j istasyonundaki birinci robotunun f . zaman birimindeki uç noktasına ait dairesel gösterimin koordinatlarını içerdiğini varsayalım. Bu koordinat noktası ikinci robotun

eklem ve bağlantıları üzerinde yer alan, aynı zaman birimindeki tüm dairelerinin koordinatlarından farklı olması durumunda çarpışma durumunun mevcut birim zaman için o istasyonda söz konusu olmadığını gösterir. Aksi halde robotik manipülatörlerin çarpışma durumu ortaya çıkmaktadır.

Çarpışma kontrolünün çok fazla işlem yükü getirmesinin yanında, doğrusal olmayan bir yapıya sahip olması çarpışma kontrolü için Rana ve Zalzala (1995)'nin yöntemini daha kullanışlı kılmaktadır. Çarpışma kontrolüne ait sözde kod Şekil 3.25'te yer almaktadır.

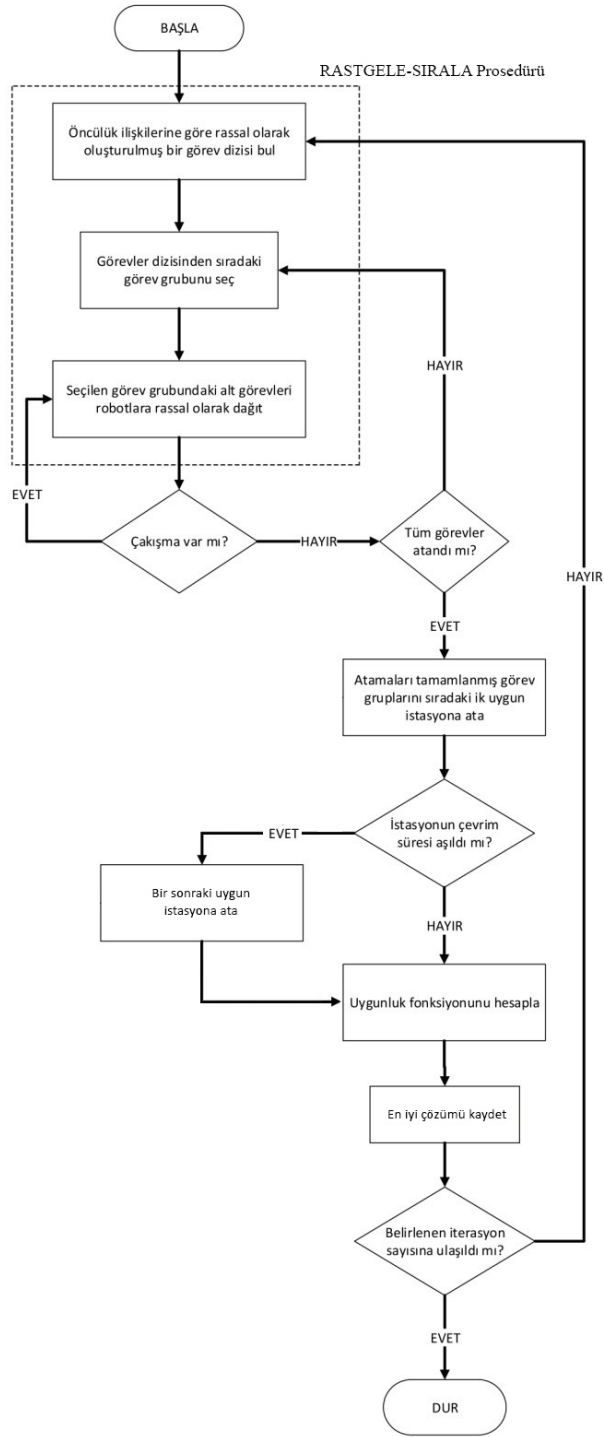
Çarpışmasız ve kilitlenmesiz görev atama ile yapılan bu istasyon dengeleme yöntemine ait yalancı kod Şekil 3.26'da verilmiştir. Bir sonraki bölümde probleme ait veriler tüm detaylarıyla verilmiş olup, bu veriler kullanılarak tez çalışmasında açıklanan yöntem için bir uygulama örneği yer almaktadır.

ÇAKIŞMASIZ_MONTAJ_HATTI_DENGELEME_YÖNTEMİ

```
1 // Girdi: Çevrim süresi ( $C_{time}$ ) ve  $A[i]$  görevler dizisi
2 // Girdi: İstasyonlara atanacak robotik manipülatörlere ait kinematik bilgileri (R1: 1. robotik
3 // manipülatör, R2: ikinci robotik manipülatör
4 // Girdi: Görev gruplarındaki alt görevlere ait koordinatlar ve görev grupları arasındaki öncüllük
5 // ilişkileri
6  $p$  ve  $r$  alt görevler indisi (her bir görev grubunda  $S$  adet)
7 MesafeMatrisi ( $p,r$ ) : Alt görevler arasındaki tüm mesafeleri içeren matris ( $S \times S$ )
8  $I$  : iterasyon sayısı,  $is = 0$ 
9  $M_{max}$ : Yeterince büyük bir sayı
10 Uygunluk fonksiyonu ( $U_{eniye}$ ) =  $M_{max}$ 
11 while  $is \leq I$ 
12     RASTGELE-SIRALA ( $A$ )
13     for  $i = 1$  'den  $n$  'e
14         RASTGELE-SIRALA  $A[i]$ , ( $A[i]$  'ye ait alt görevler için)
15         Alt görevleri sırasıyla alt görev dizisine göre robotik manipülatörlere ata
16         Toplam süreyi hesapla (MesafeMatrisi ( $p,r$ ))
17         Robotlara ait tüm birim zamanlardaki koordinatları hesapla
18         ÇAKIŞMA-KONTROLÜ
19             Çakışma yoksa devam, varsa 11. satıra dön
20     end
21      $\tau_j$  : istasyonlardaki toplam işleme süresi
22     for  $j = 1$  'den  $m$  'ye kadar (istasyonlar)
23         while  $\tau_j \leq C_{time}$ 
24              $A[i]$  görev grubunu  $j$ . istasyona ata
25         end
26     end
27     Uygunluk fonksiyonu ( $U_{aday}$ ) hesapla
28     if  $U_{aday} \leq U_{eniye}$ 
29          $U_{eniye} = U_{aday}$ 
30     end
31      $is = is + 1$ 
32 end
33 return  $U_{eniye}$ 
```

Şekil 3.26. Çarpışmasız montaj hattı dengeleme yöntemine ait sözde kod

Bu tezdeki problemin çözümünde kullanılan yöntemine ait iş akış diyagramı Şekil 3.27'de verilmiştir.



Şekil 3.27. Çarpışmasız montaj hattı dengeleme yönteminin akış diyagramı

4. BULGULAR

4.1. Uygulama

4.1.1. Uygulama örneği

Model için oluşturulan uygulama örneğinde 8 adet görev grubu tanımlanmıştır. Görev gruplarının her biri 6 adet görevden oluşmaktadır. Bu görevler, montaj hattında işlenecek olan tek tip ürün için yapılacak punta kaynağı noktalarının üç boyutlu uzaydaki koordinatlarıyla tanımlanmıştır. Görev grupları ve bu gruplardaki görevlere ait punta noktalarının koordinatları EK 1'de verilmiştir. Modelin çözümünde beş farklı çevrim süresi (C_{time}) (50, 60, 70, 80 ve 90 saniye) kullanılmıştır. Farklı çevrim sürelerinin kullanım amacı, açılan iş istasyonları sayısındaki değişimi gözlemlemektir. Her bir iş istasyonunda iki adet robotik manipülatör yer almaktadır. Bu robotik manipülatörler, PUMA 560 - benzeri (6R) robotik manipülatörlerdir.

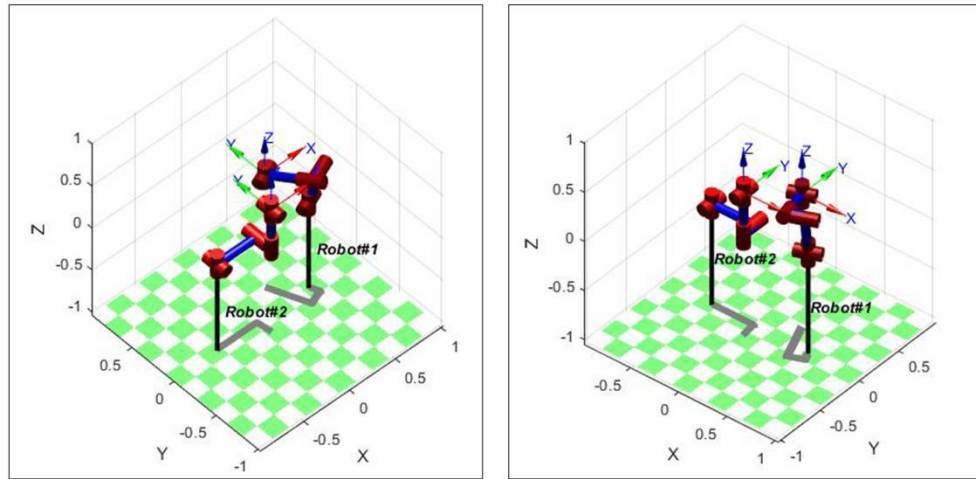
PUMA 560 (Montaj için Programlanabilir Evrensel Makine veya Programlanabilir Evrensel Manipülasyon Kolu) altı eksenli eklemleri olan endüstriyel bir robottur (Şekil 3.17). PUMA 560 robotunun tutucu uç elemanı, 0.1 mm konumsal tekrarlanabilirlik ile 2.5 kg'lık bir nominal yükü çalıştırmak üzere tasarlanmıştır. Bu robotik manipülatörün maksimum tutucu uç elemanı hızı 1m/s'ye ulaşmaktadır. Altı eklemin tamamı DC servo motorlar aracılığıyla harekete geçirilir. Eklem sınırları Çizelge 4.1'de verilmiştir.

PUMA 560 robotik manipülatörleri, kompakt tasarımı, yüksek hız oranı, tekrarlanabilirliği ve esnekliği nedeniyle, küçük nesnelere veya endüstriyel parçaların işlenmesinde en çok kullanılan robotik manipülatörlerden biridir. En karmaşık uygulamalar veya karmaşık parçaların montajı PUMA 560 robotu ile yapılabilir. Örneğin PUMA 560 robotu, otomotiv panelleri, küçük elektrik motorları ve devre kartı baskıları gibi birçok alanda kullanılabilir. Çizelge 4.2'den PUMA 560 robotik manipülatörünün ileri ve ters kinematiklerini hesaplamak için Denavit-Hartenberg değişkenleri ve parametreleri kullanılmıştır.

Çizelge 4.1. Robotik manipülatöre ait eklemlerin açısıl erişimi

Eklemler	Erişim Açısı (Derece)
Bel (waist)	320
Omuz	266
Dirsek	284
Bilek eğimi	200
Bilek yuvarlanması	280
Bilek rotası	532

Montaj hattında yer alan her bir robotik manipülatör, aynı teknolojik ve fiziksel özelliklere sahiptir. Aralarındaki tek fark, temellerinin sabit olarak bulunduğu konumdur. Şekil 4.1'de iki robotik manipülatörün bir arada buldukları benzetim örnekleri verilmiştir.



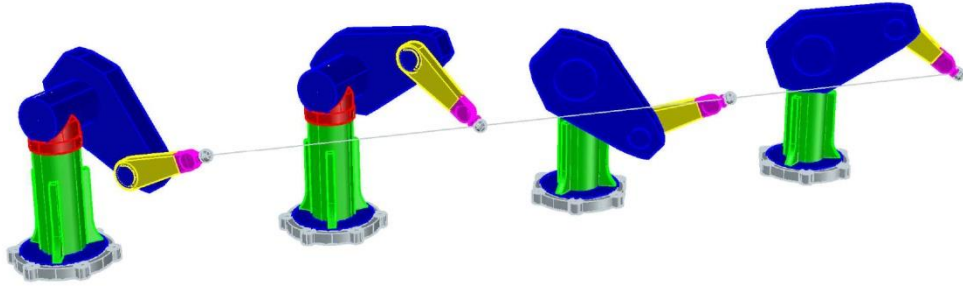
Şekil 4.1. İki robotik manipülatörün yer aldığı benzetimler

Ayrıca, robotik manipülatörlerde bir noktaya robotik manipülatörün tutucu uç elemanının farklı yönelimlerle ulaşabilme sorunu olarak bilinen tekilik (singularity) sorunu, bu tez çalışmasında dikkate alınmamıştır. Şekil 4.2'de PUMA-benzeri robotik manipülatöre ait tutucu uç elemanın bir noktaya birden fazla açısıl yönelim gidebileceğini gösteren bir benzetim yer almaktadır.

İki robotik manipülatörün bulunduğu çalışma uzayının hacmi 1 m^3 olarak alınmıştır. İstasyonlara atanan görevler bu çalışma uzayının dışına çıkmamaktadır.

Çizelge 4.2. Robotik manipulatörün DH parametreleri (Jha 2015)

Eklemler	θ_i (derece)	d_i (metre)	a_i (metre)	α_i (derece)
0	$\theta_1 = \pm 160^\circ$	0	0	90
1	$\theta_2 = -225^\circ$ 'den $+45^\circ$ 'ye	0	0	0
2	$\theta_3 = -45^\circ$ 'den $+225^\circ$ 'ye	$d_3 = 0.1244$	$a_3 = 0.4318$	-90
3	$\theta_4 = \pm 110^\circ$	$d_4 = 0.4318$	$a_4 = 0.0203$	90
4	$\theta_5 = \pm 100^\circ$	0	0	-90
5	$\theta_6 = \pm 266^\circ$	0	0	0



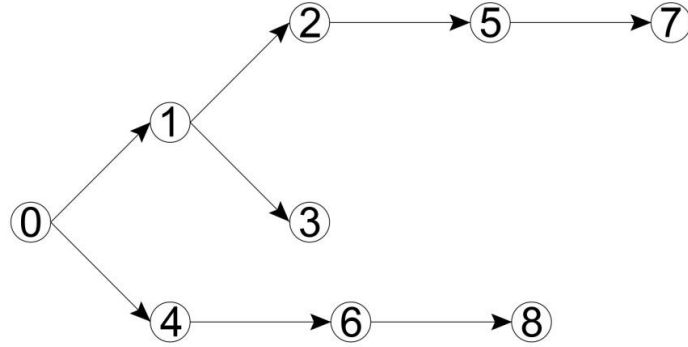
Şekil 4.2. Tutucu uç elemanın bir noktaya farklı şekillerde ulaşma durumu

Robotik manipulatörlere ait çalışma uzayı, robotların uzanabileceği maksimum yörüngelerden daha küçük olmayacak şekilde tasarlanmıştır. Robotik manipulatörlerle ilgili teknik özellikler ve çalışma uzayına ait varsayımlar aşağıdaki verilmiştir:

- Robotik manipulatörlere ait kinematik bilgileri önceden bilinmektedir.
- Çalışma uzayında işlenecek parçaya ait punta noktaları, robotik manipulatörlerin her ikisinin de ulaşabileceği noktalar olarak alınmıştır.
- Montajı yapılacak olan iş parçası ile robotik manipulatörlerin çakışmasını engellemek için çalışma uzayına iş parçasını doğrudan yerleştirmek yerine üç boyutlu uzayda iş parçasını temsil eden rastgele noktalar kullanılmıştır.
- Çalışma uzayında robotik manipulatörlere engel teşkil edecek başka herhangi bir parça bulunmamaktadır.

Robotik manipülatörlere, bir görev grubundaki görevler birer birer atanır. Bu işlem görev gruplarındaki görevlerin tamamı atanana kadar devam eder. Montaj hattında işlenecek parçaya ait öncül-ardıl ilişkileri Şekil 4.3'te verilmiştir. Şekilden de anlaşılacağı gibi istasyonlara toplamda 8 adet görev grubu atanmalıdır. Şekilde yer alan 0 düğümü yalancı düğümdür. Burada dikkat edilmesi gereken konu, görev grupları arasında öncül-ardıl ilişkileri varken, görev gruplarının kendi içlerindeki görevlerde öncül-ardıl ilişkilerinin bulunmamasıdır.

Bir robotik manipülatörün bir işi yapması demek, kendisine atanan bir görev noktasındaki punta kaynağı işlemini bitirdikten sonra bir sonraki görev noktasına giderken harcadığı süre ve ulaştığı görev noktasındaki 2 saniyelik punta kaynağı işlemidir. Robotik manipülatörlerin aldığı yol ile bu yolu kat etme süresi doğru orantılıdır.



Şekil 4.3. Öncül-ardıl ilişkileri diyagramı

Şekil 3.26 ile verilen yalancı kod MATLAB 2013 ortamında kodlanmıştır. MATLAB programında, Corke (1996) tarafından geliştirilen ve robotlarla ilgili çalışmalarda kullanılan, Robotics Toolbox kütüphanesinden, bu tez çalışmasındaki robotik manipülatörlere ait teknik hesaplamalar için yararlanılmıştır (<http://petercorke.com/wordpress/toolboxes/robotics-toolbox>, 2018). Yararlanılan kütüphane aşağıdaki konular için kullanılmıştır:

- Robotik manipülatörlerin tanımlanması
- Robotik manipülatörlere ait çalışma uzaylarının oluşturulması

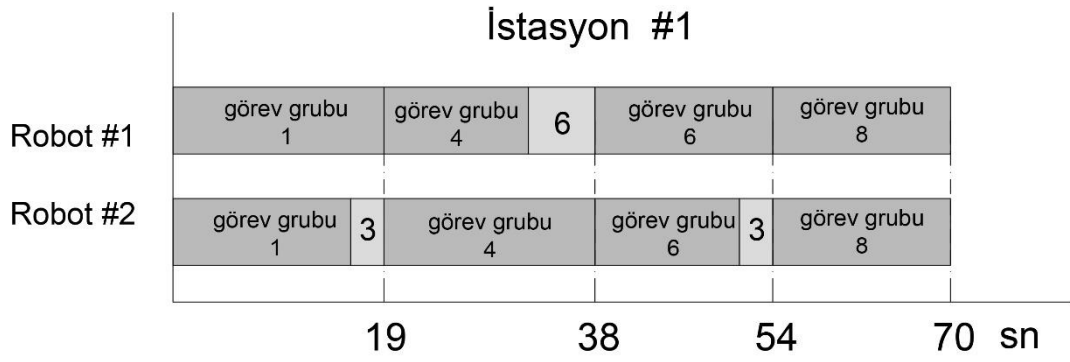
- İleri ve ters kinematik hesabı
- Robotik manipülatörlere ait yörüngelerin hesaplanması

Bölüm 3'te verilen yöntem önceden belirlenmiş bir uygulama örneğinin çözümü için MATLAB programında kodlanmış olup, yazılan kod EK 2'de verilmiştir. Kod, beş farklı montaj çevrim süresi (50, 60, 70, 80 ve 90 saniye) için çalıştırılmış olup, çevrim süresinin 70 saniye olduğu örnek için 500 iterasyon, diğer çevrim süreleri için de 50 iterasyon sonunda durdurulmuştur.

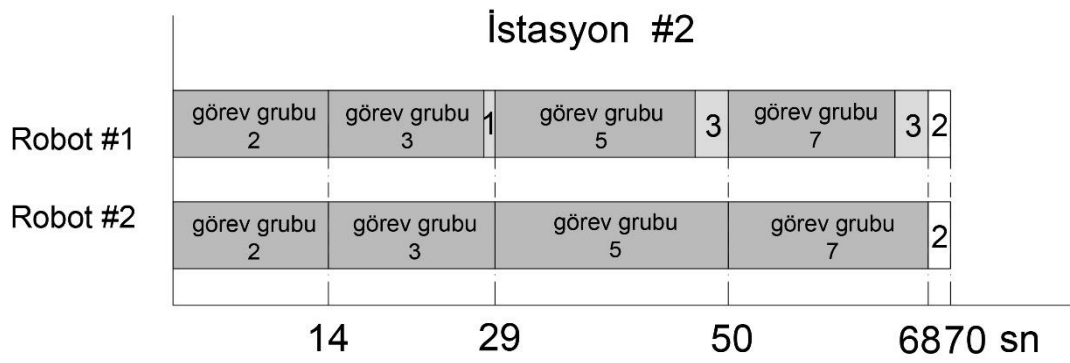
4.1.2. Sonuçlar

İlk olarak Şekil 4.3'te verilen görev gruplarının, öncül-ardıl ilişkilerine göre rassal olarak atanma dizisi belirlenmiştir. Bu dizi ilk olarak 1 numaralı görev grubuyla başlayıp, sonrasında sırasıyla, 4, 6, 8, 2, 3, 5 ve son olarak 7 numaralı görev grubundan oluşmaktadır. Ayrıca görev gruplarının atama sırası için elde edilen bu dizi tüm çevrim süreleri için kullanılmıştır.

Çevrim süresinin 70 saniye olduğu örnek için 50 iterasyon sonundaki en iyi uygunluk fonksiyonu değeri 21 olup toplam iki istasyon açılmıştır ve çözüm süresi 899,7 saniye sürmüştür. İstasyon 1 burada tam olarak dolmakla birlikte, istasyon 1'den uygunluk fonksiyonuna eklenen atıl zamanlar, yalnızca robotların kendi aralarında oluşan ve her bir görev grubuna ait görevlerin bitmesiyle (bir istasyona atanan bir görev grubundaki tüm görevler, iki robot arasında paylaşılır ve bu görevlerin tamamı işlenmeden bir başka görev grubuna ait göreve başlanamaz) ilgili olan atıl zamanlardır (Şekil 4.4). İstasyon 1'e toplam dört adet görev grubu atanmıştır. İkinci istasyonda ise durum ilk istasyondan biraz farklıdır. İstasyon 1'de tüm görev grupları çevrim süresini tam doldururken, istasyon 2'de görev grupları bittiğinde çevrim süresine kadar 2 saniyelik atıl bir zaman kalmaktadır (Şekil 4.5).

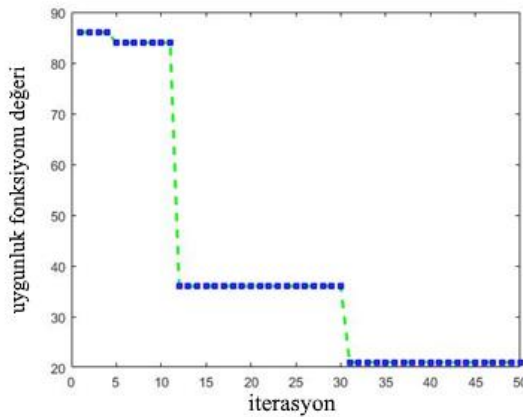


Şekil 4.4. İstasyon 1 için çizelge (çevrim süresi 70 sn)



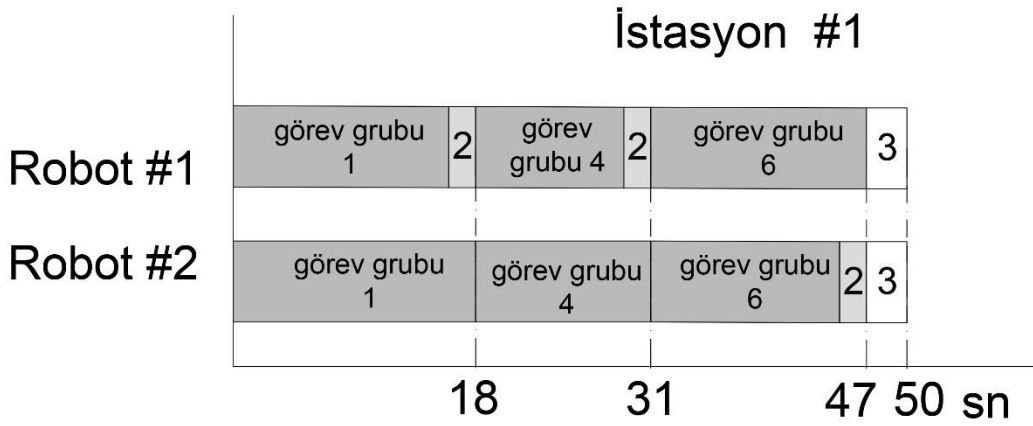
Şekil 4.5. İstasyon 2 için çizelge (çevrim süresi 70 sn)

Çevrim süresinin 70 sn olduğu örnek için tüm iterasyonlara bakılırsa (EK 3) aday çözümlerin yalnızca %4'ünde üçüncü istasyon açılmamıştır. Her bir aday çözümün yer aldığı ve aday çözümlerde, robotik manipülatörlere atanan her bir görev grubunun süresi EK 4'te verilmiştir. Ayrıca 50 iterasyon için en iyi uygunluk fonksiyonu değerinin değişimi Şekil 4.6'da yer almaktadır.

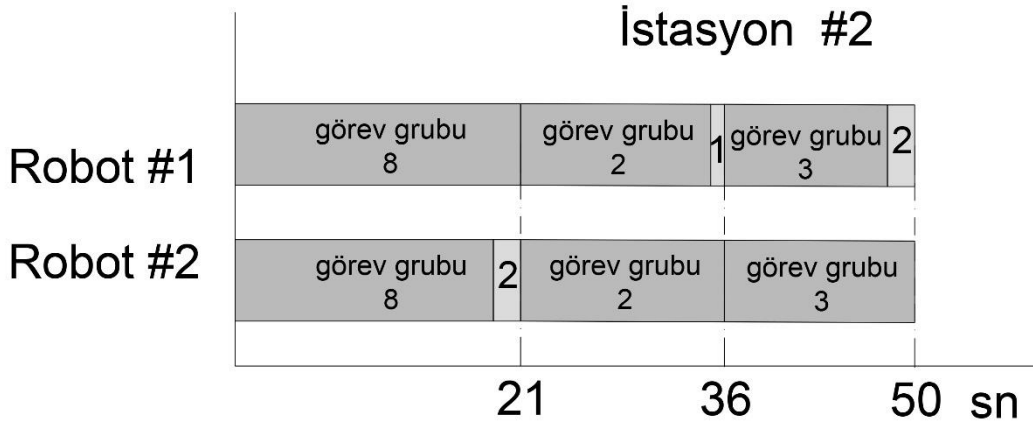


Şekil 4.6. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 70 sn)

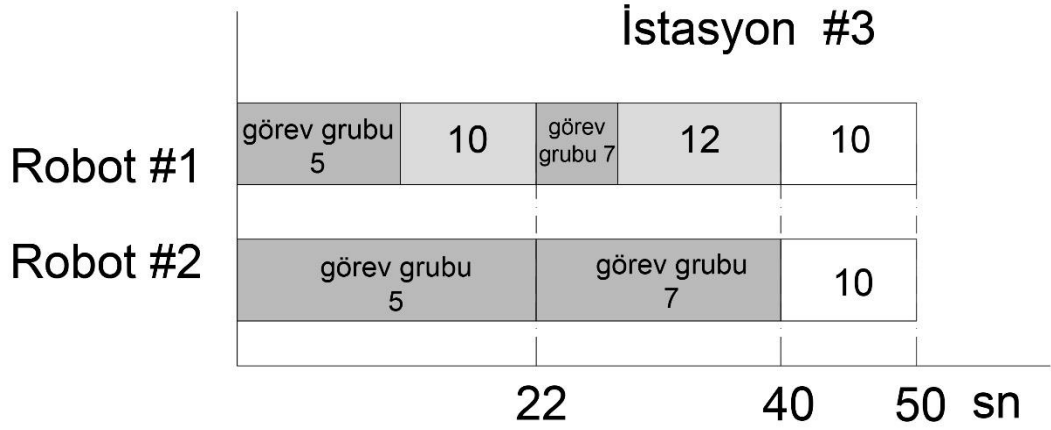
Çevrim süresi 50 sn olan örneğin çözümü, 50 iterasyon sonucunda 972 saniye sürmüş olup en iyi uygunluk fonksiyonu değeri 46 olarak bulunmuştur. Açılan istasyon sayısı en iyi çözümde 3 olmakla birlikte tüm iterasyonlar bazında aday çözümlerin yalnızca birinde dördüncü istasyon açılmamıştır (EK 5) (dördüncü istasyonun açılmadığı iterasyondaki aday çözüm en iyi çözümdür). İlk istasyondan uygunluk fonksiyonuna eklenen atıl zaman değeri 9 (Şekil 4.7), ikinci istasyondan gelen değer 5 (Şekil 4.8) ve üçüncü istasyondan gelen değer 32'dir (Şekil 4.9).



Şekil 4.7. İstasyon 1 için çizelge (çevrim süresi 50 sn)

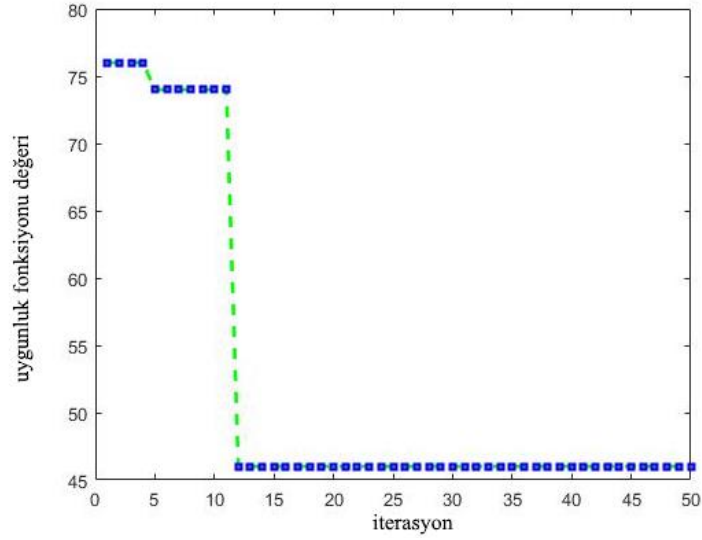


Şekil 4.8. İstasyon 2 için çizelge (çevrim süresi 50 sn)



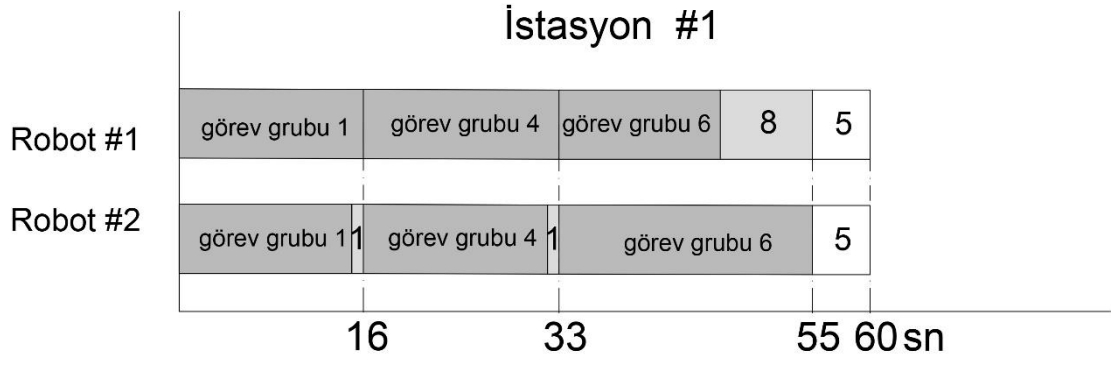
Şekil 4.9. İstasyon 3 için çizelge (çevrim süresi 50 sn)

Uygunluk fonksiyonu için en iyi değer 50 iterasyondaki değişimi Şekil 4.10'da verilmiştir.

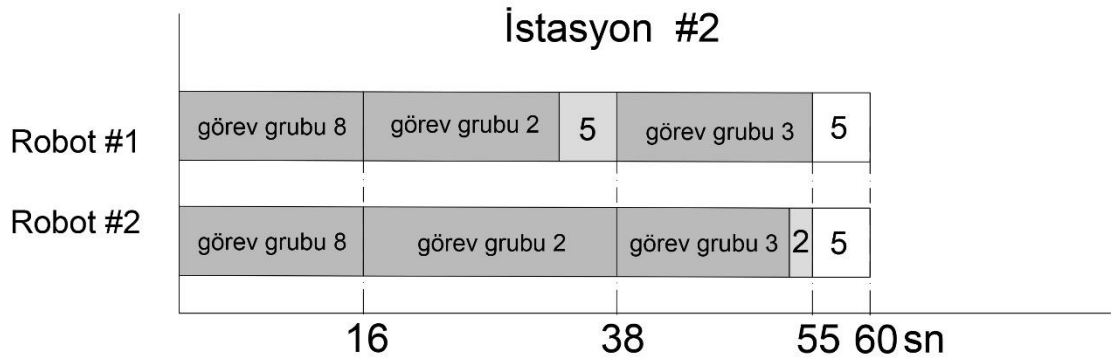


Şekil 4.10. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 50 sn)

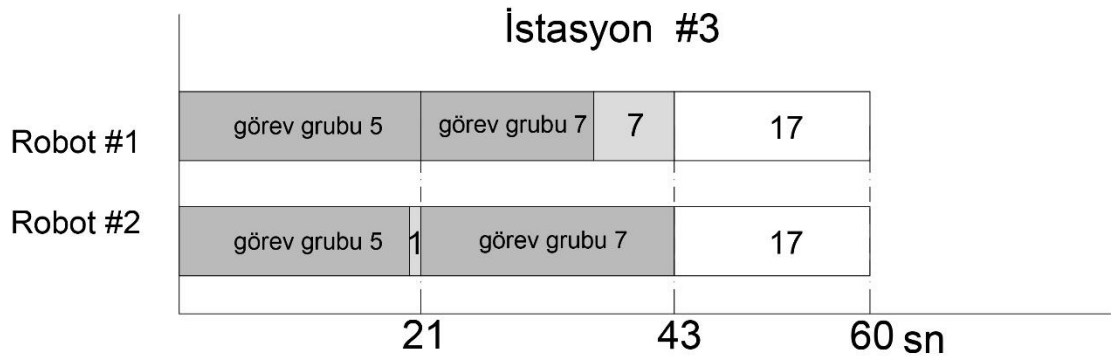
Çevrim süresi 60 sn olarak alındığında, 50 iterasyon sonundaki en iyi uygunluk fonksiyonu değeri 52 olup, çözüm süresi 941,8 saniye sürmüştür. En iyi çözümde açılan istasyon sayısı 3 olmakla birlikte tüm aday çözümlerin yalnızca %14'ünde dördüncü istasyonu açma gereksinimi duyulmuştur (EK 6). İlk istasyondan uygunluk fonksiyonuna eklenen atıl zaman değeri 15 (Şekil 4.11), ikinci istasyondan gelen değer 12 (Şekil 4.12) ve üçüncü istasyondan gelen değer 25'dir (Şekil 4.13). Uygunluk fonksiyonu için en iyi değer 50 iterasyondaki değişimi, çevrim süresinin 60 olarak alındığı örnek için Şekil 4.14'te verilmiştir.



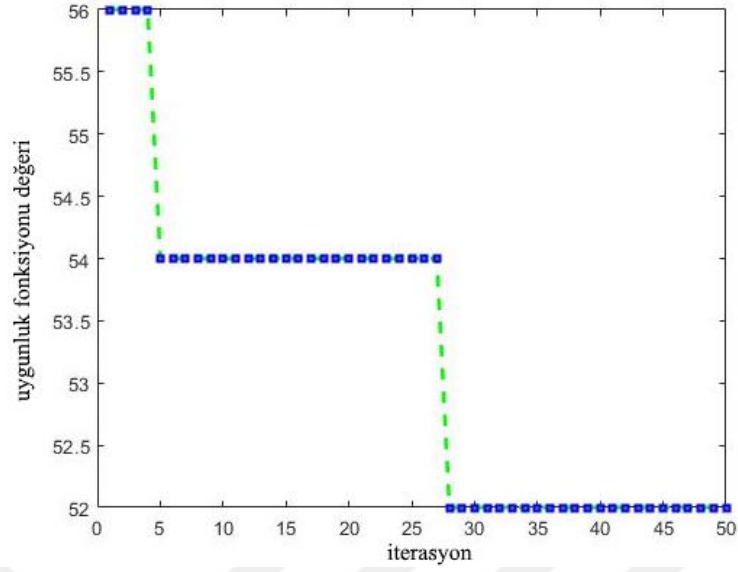
Şekil 4.11. İstasyon 1 için çizelge (çevrim süresi 60 sn)



Şekil 4.12. İstasyon 2 için çizelge (çevrim süresi 60 sn)

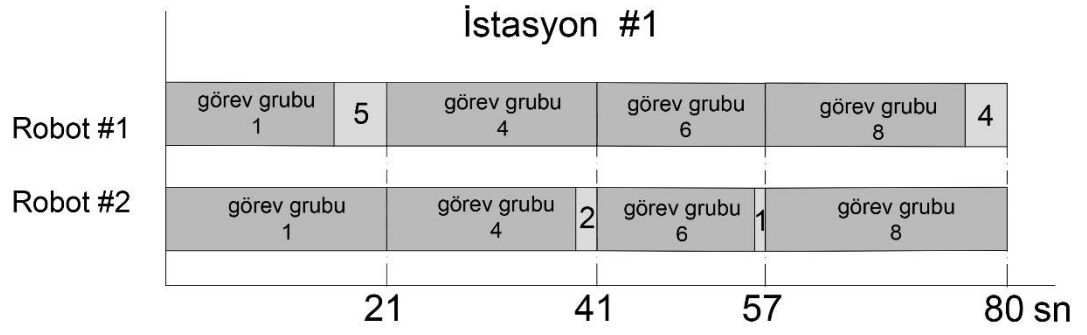


Şekil 4.13. İstasyon 3 için çizelge (çevrim süresi 60 sn)

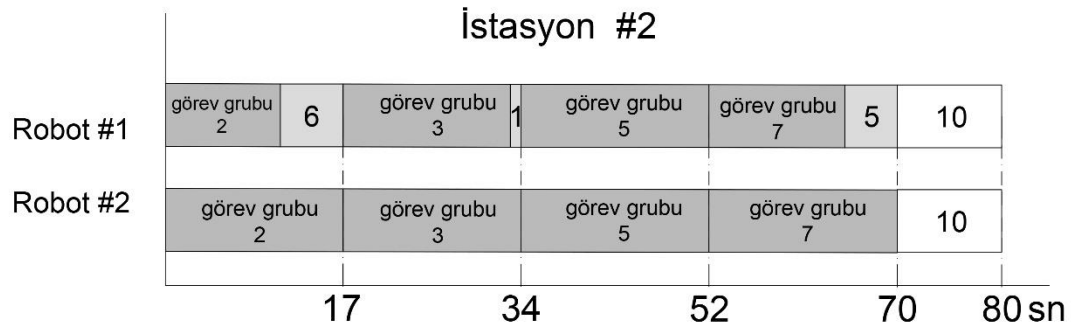


Şekil 4.14. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 60 sn)

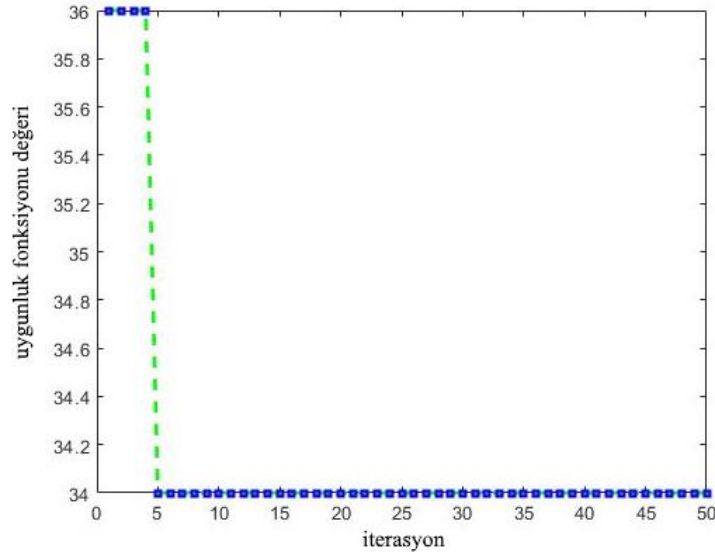
Çevrim süresi 80 sn olarak alındığında ise 50 iterasyon sonucunda en iyi uygunluk fonksiyonu değeri 34 olup, çözüm süresi 1029,1 saniye olarak ölçülmüştür. En iyi çözümde açılan istasyon sayısı 2 olmakla birlikte, aday çözümlerin %36'sında üçüncü istasyonu açma gereksinimi duyulmuştur (EK 7). İlk istasyondan uygunluk fonksiyonuna eklenen atıl zaman değeri 15 (Şekil 4.15) ve ikinci istasyondan gelen değer 12 olarak bulunmuştur (Şekil 4.16). Uygunluk fonksiyonu için en iyi değer 50 iterasyondaki değişimi, çevrim süresinin 80 olarak alındığı örnek için Şekil 4.17'de verilmiştir. Şekil 4.17'den de anlaşılacağı üzere 5. iterasyonda en iyi uygunluk fonksiyonu değeri elde edildikten sonra herhangi bir iyileştirme olmamıştır.



Şekil 4.15. İstasyon 1 için çizelge (çevrim süresi 80 sn)



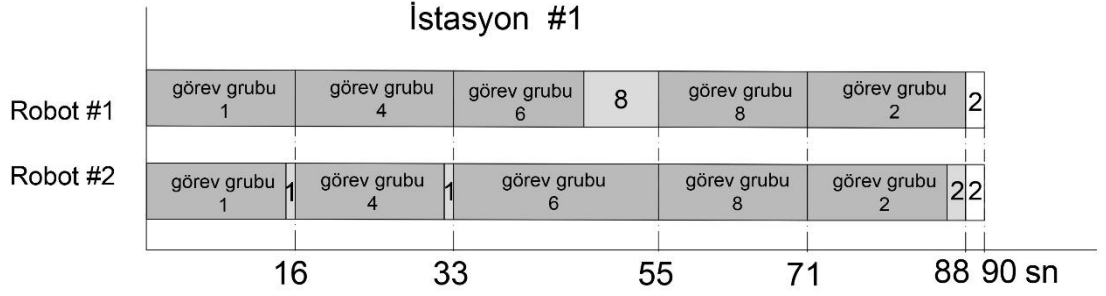
Şekil 4.16. İstasyon 2 için çizelge (çevrim süresi 80 sn)



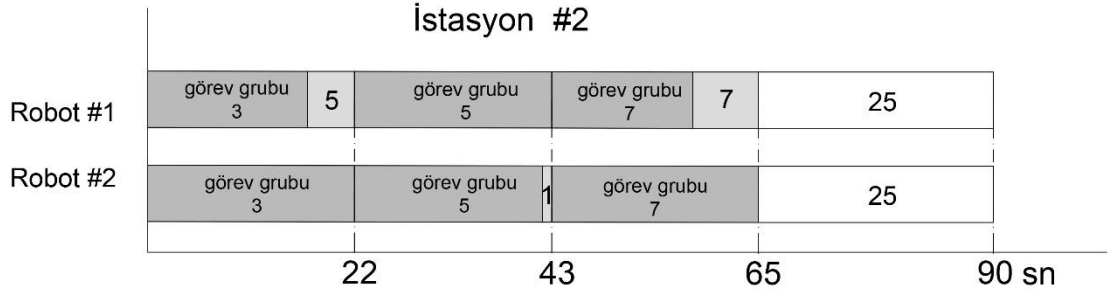
Şekil 4.17. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 80 sn)

Son olarak çevrim süresinin 90 sn olarak alındığı örneğin çözümünden elde edilen en iyi uygunluk fonksiyonu değeri 52 olup, çözüm süresi 938,9 saniye olarak ölçülmüştür. En iyi çözümde açılan istasyon sayısı 2 olmakla birlikte, aday çözümlerin tamamında 2

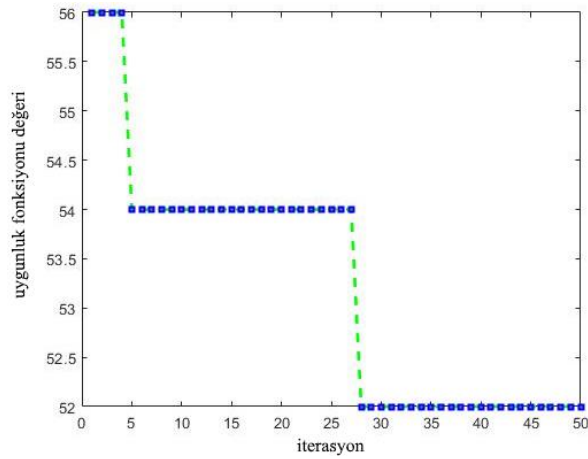
istasyon açıldığı görülmektedir (EK 8). İlk istasyondan uygunluk fonksiyonuna eklenen atıl zaman değeri 14 (Şekil 4.18) ve ikinci istasyondan gelen değer 38 olarak bulunmuştur (Şekil 4.19). Uygunluk fonksiyonu için en iyi değer 50 iterasyondaki değişimi, çevrim süresinin 90 sn olarak alındığı örnek için Şekil 4.20'de verilmiştir.



Şekil 4.18. İstasyon 1 için çizelge (çevrim süresi 90 sn)



Şekil 4.19. İstasyon 2 için çizelge (çevrim süresi 90 sn)



Şekil 4.20. En iyi uygunluk fonksiyonu değerinin 50 iterasyondaki değişimi (çevrim süresi 90 sn)

50 iterasyonda beş farklı çevrim süresi için çözüm yapıldıktan sonra, çevrim süresi 70 sn olan durum için iterasyon sayısı 500'e çıkarılmış, en iyi uygunluk fonksiyonu değeri 20 olarak bulunmuş ve çözüm süresi 9782,6 saniye olarak ölçülmüştür. Çevrim süresinin 70 sn olduğu durum üzerinden 50 iterasyon sonucunda uygunluk fonksiyonunun en iyi değeri 21 bulunurken, 500 iterasyon sonucunda bu değer 20 olarak elde edilmiştir. Bu durum, iterasyon sayısının fazlasıyla artırılmasına rağmen, uygunluk fonksiyonu değerinde kayda değer bir iyileştirme olmadığını göstermektedir.

4.2. Bulgular

Çarpışmasız ve kilitlenmesiz çok robotlu montaj hattı dengeleme probleminin çözülmesine yönelik olarak bu tez çalışmasında önerilen iki aşamalı rastsal sıralama yöntemi, farklı çevrim süreleri için test edilmiştir. Her bir çevrim süresine ait aday uygunluk fonksiyonu değerleri ve açılan istasyonların sayısı ile ilgili bazı ilişkiler söz konusudur. Farklı çevrim süreleriyle elde edilen çözümlere ait en iyi uygunluk fonksiyonu değerleri, montaj hattında en az sayıda açılan istasyonların olduğu iterasyonlardan birinde elde edilmiştir.

Belirlenen çevrim süresi ve çözüm süresi arasında doğrudan bir ilişki olmamasına rağmen verilen görevlerin ve istasyonlarda bulunan robotik manipülatörlerin sayısının artması problemi daha karmaşık hale getireceği açıktır. Çok-işçili montaj hattı dengeleme probleminin NP-zor olduğu bilinmektedir. Dolayısıyla bu probleme işçilerin (bu tez çalışmasında işçilerin yerini robotik manipülatörler almıştır) çarpışmama kısıtı eklendiğinde problemin çözümü oldukça zor bir yapıya bürünmektedir.

Rassal sıralama yönteminin kullanımı sürekli iyileştirmeyi garanti edemediğinden, iterasyon sayısının artırılması da daha iyi bir çözüm bulmayı garanti edemez. Bunun yanı sıra bu yöntem, çarpışma kısıtı söz konusu olduğunda, robotik manipülatörlerin çarpışmayacak şekilde görevleri paylaşması bakımından, farklı permütasyonların denenmesi nedeniyle oldukça kullanışlıdır.

5. TARTIŞMA VE SONUÇ

Bu tez çalışmasında çok robotlu montaj hattı dengeleme problemi, robotik manipülatörlerin çarpışmaması ve birbirlerini kilitlememesi kısıtıyla beraber ele alınmıştır. Bu problemin çözümü için rassal dizi sıralama yöntemi kullanılmıştır. Problem NP-zor yapıya sahiptir.

Çalışmada ilk olarak robotik montaj hatları ve çok-işçili montaj hatları dengeleme problemiyle ilgili literatür ele alınmış ve daha sonra, çarpışma kısıtını sağlayabilmek için robotik manipülatörlerde çarpışmasız yörünge planlama ile ilgili literatür, detaylı bir şekilde incelenmiştir.

Problemin çözümünde MATLAB programından ve MATLAB için geliştirilmiş olan Robotic Toolbox kütüphanesinden yararlanılmıştır. Kullanılan yöntem, öncül-ardıl ilişkilerine sahip görev gruplarının olduğu ve seri bir montaj hattı içeren çarpışmasız ve kilitlenmesiz çok-robotlu montaj hattı dengeleme probleminin çözümü için özellikle çarpışmama kısıtını sağlama açısından kullanışlıdır.

İleriki çalışmalarda istasyonlardaki robotik manipülatörlerin sayısının artırılması, robotik manipülatörlerin bulunduğu istasyonlara işçilerin eklenmesi, robotik manipülatörlere ait dinamik ve kinematik parametreler daha detaylı bir şekilde kullanılarak metasezgisel yöntemlerden yararlanılması, tek modellenmiş montaj hattı yerine çoklu model veya karma model montaj hattının ele alınması ve düz montaj hattı yerine karmaşık bir yapıya sahip olan montaj hattının göz önünde bulundurulması gibi yeni problemler üzerinde çalışılabilir.

KAYNAKLAR

- Afaghani, A.Y., Aiyama, Y. 2014.** Advanced-collision-map-based on-line collision and deadlock avoidance between two robot manipulators with PTP commands. International Conference on Automation Science and Engineering (CASE), 1244-1251, 18-22 Ağustos, 2014, Taipei, Tayvan.
- Akella, S., Hutchinson, S. 2002.** Coordinating the motions of multiple robots with specified trajectories. International Conference on Robotics and Automation, 1:624-631, Mayıs, 2002, Washington, Amerika Birleşik Devletleri.
- Akella, S., Peng, J. 2004.** Time-scaled coordination of multiple manipulators. International Conference on Robotics and Automation, 4: 3337-3344, Nisan, 2004, Los Angeles, Amerika Birleşik Devletleri.
- Aronov, B., de Berg, M., van der Stappen, A.F., Svestka, P., Vleugels, J. 1999.** Motion planning for multiple robots. *Discrete & Computational Geometry*, 22(4): 505-525.
- Beaumont, R.G., Crowder, R.M. 1989.** Two-armed robot systems-a review of current theory and the development of algorithms for real-time collision avoidance. In *Controllers for Robotic Applications-Concepts and Implementations*, IEE Colloquium on (pp. 1-1). IET.
- Becker, C., Scholl, A. 2006.** A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3), 694-715.
- Becker, C., Scholl, A. 2009.** Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. *European Journal of Operational Research*, 199: 359-374.
- Bennewitz, M., Burgard, W., Thrun, S. 2001.** Optimizing schedules for prioritized path planning of multi-robot systems. IEEE International Conference on Robotics and Automation, 1: 271-276, 21-26, Mayıs, 2001, Seul, Kore.
- Berman, N., 2016.** 10 Auto Industry Jobs that Will Die Due to Automation. <http://moneyinc.com/10-auto-industry-jobs-will-die-due-automation>(Erişim tarihi: 2018).
- Chang, H.J., Chang, T.M. 2010.** Simultaneous Perspective-Based Mixed-Model Assembly Line Balancing Problem. *Tamkang Journal of Science and Engineering*, 13(3): 327-336.
- Corke, P.I. 1996.** A robotics toolbox for MATLAB. IEEE Robotics & Automation Magazine. 1(3):24-32.
- Corke, P.I. 1996.** Robotic toolbox for MATLAB. <https://petercorke.com/wordpress/toolboxes/robotics-toolbox>. (Erişim tarihi: 2018).
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. 2001.** Introduction to algorithms second edition. *The MIT Press*, Massachusetts, USA. 92 pp.
- Craig, J.J. 1989.** Introduction to Robotics: mechanics and control second edition. *Addison-Wesley Publishing Company, Inc.* USA. 75 pp.
- Crane III, C. D., Duffy, J. 2008.** Kinematic analysis of robot manipulators. *Cambridge University Press*. Birleşik Krallık, 20-27 pp.
- Çevikcan, E., Durmuşoğlu, M.B., Ünal, M.E. 2009.** A team-oriented design methodology for mixed model assembly systems. *Computers & Industrial Engineering*, 56: 576-599.

- Çil, Z.A., Mete, S., Ağpak, K. 2017.** An Efficient Heuristic Algorithm for Solving Robotic Assembly Line Balancing Problem. 2017 8th International Conference on Information Technology (ICIT), 17-18 Mayıs, 2017, Amman, Jordan.
- Çil, Z.A., Mete, S., Ağpak, K. 2017.** Analysis of the type II robotic mixed-model assembly line balancing problem, *Engineering Optimization*, 49(6): 990-1009.
- Daoud, S., Chehade, H., Yalaoui, F., Amodeo, L. 2014.** Solving a robotic assembly line balancing problem using efficient hybrid methods. *J. Heuristics*, 20: 235-259.
- Deckro, R.F. 1989.** Balancing cycle time and workstations. *IIE Trans* 21:106–11.
- Dimitriadis, S.G. 2006.** Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. *Computers & Operations Research*, 33: 2757-2774.
- Ege, Y., Azizoglu, M., Ozdemirel, N.E. 2009.** Assembly line balancing with station paralleling. *Computers & Industrial Engineering*, 57(4), 1218-1225.
- Fattahi, P., Roshani, A. 2011.** A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem. *The International Journal of Advanced Manufacturing Technology*, 53(1-4): 363-378.
- Gao, J., Sun, L., Wang, L., Gen, M. 2009.** An efficient approach for type II robotic assembly line balancing problems. *Computers & Industrial Engineering*, 56: 1065–1080.
- Hartenberg, R.S., Denavit, J. 1955.** A kinematic notation for lower pair mechanisms based on matrices, *J. Appl. Mech.*, 77(2): 215-221.
- Helgeson, W.B., Birnie, D.P. 1961.** Assembly line balancing using the ranked positional weight technique. *Journal of Industrial Engineering*, 12(6): 394-398.
- Hoffmann, T.R. 1963.** Assembly Line Balancing with a Precedence Matrix. *Management Science*, 9(4): 551-562.
- Jager, M., Nebel, B. 2001.** Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. International Conference on Intelligent Robots and Systems, 3:1213-1219, 29 Ekim – 3 Kasım, 2001, Havai, Amerika Birleşik Devletleri.
- Jha, P. 2015.** Inverse kinematic analysis of robot manipulators. *Ph.D. Thesis*, National Institute of Technology, Rourkela, India.
- Kazemi, A., Sedighi, A. 2013.** A cost-oriented model for multi-manned assembly line balancing problem. *Journal of Optimization in Industrial Engineering*, 6(13): 13-25.
- Kellegöz, T., Toklu, B. 2012.** An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations. *Computers & Operations Research*, 39(12): 3344-3360.
- Kellegöz, T., Toklu, B. 2015.** A priority rule-based constructive heuristic and an improvement method for balancing assembly lines with parallel multi-manned workstations. *International Journal of Production Research*, 53(3): 736-756.
- Kellegöz, T. 2017.** Assembly line balancing problems with multi-manned stations: a new mathematical formulation and Gantt based heuristic method. *Annals of Operations Research*, 253(1): 377-404.
- Kim, H., Park, S. 1995.** A strong cutting plane algorithm for the robotic assembly line balancing problem. *International Journal of Production Research*, 33(8): 2311-2323.
- Kim, Y.K., Song, W.S., Kim, J.H. 2009.** A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Comput. Oper. Res.* 36:853–865.

- Kobetski, A., Sepensieri, D., Fabian, M. 2006.** Scheduling algorithms for optimal robot cell coordination-a comparison. International Conference on Automation Science and Engineering, 381-386, 7-10 Ekim, 2006, Şangay, Çin.
- Koga, Y., Latombe, J.C. 1994.** On multi-arm manipulation planning. In Robotics and Automation. IEEE International Conference, 945-952, Mayıs, 1994, Kaliforniye, Amerika Birleşik Devletleri.
- Lee, D.H., Kim, D.H., Lee, J.Y., Han, C.S. 2014.** Collision-free coordination of two dual-arm robots with assembly precedence constraint. 14th International Conference on Control, Automation and Systems (ICCAS), 515-520, 22-25 Ekim, 2014, Gyeonggi-do, Kore.
- Levitin, G., Rubinovitz, J., Shnits, B. 2006.** A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research*, 168: 811-825.
- Li, Z., Dey, N., Ashour, A.S., Tang, Q. 2017.** Discrete cuckoo search algorithms for two-sided robotic assembly line balancing problem. *Neural Computing & Applications*, 1-12.
- Lim, S.J., Han, C.S. 2014.** Operational space path planning of the dual-arm robot for the assembly task. *International journal of precision engineering and manufacturing*, 15(10): 2071-2076.
- Müller, C., Grunewald, M., Spengler, T.S. 2017.** Redundant configuration of robotic assembly lines with stochastic failures. *International Journal of Production Research*, 56(10): 3662-3682
- Nilakantan, J.M., Ponnambalam, S.G. 2012.** An efficient PSO for type II robotic assembly line balancing problem. 8th IEEE International Conference on Automation Science and Engineering, 20-24 Ağustos, 2012, Seoul, Korea
- Nilakantan, J.M., Ponnambalam, S.G. 2016.** Robotic U-shaped assembly line balancing using particle swarm optimization. *Engineering Optimization*, 48(2): 231-252
- Nilakantan, J.M., Ponnambalam, S.G., Jawahar, N. 2016.** Design of energy efficient RAL system using evolutionary algorithms. *Engineering Computations*, 33(2): 580-602
- O'Donnell, P.A., Lozano-Pérez, T. 1989.** Deadlock-free and collision-free coordination of two robot manipulators. Proc. IEEE Int. Conf. on Robot. and Autom. 89: 484-489.
- Paul, R. P. 1981.** Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators. *The MIT Press*. Cambridge, MA.
- Pieper, D. L. 1968.** The kinematics of manipulators under computer control (No. CS-116). Stanford Univ Ca Dept Of Computer Science. Stanford.
- Rana, A.S., Zalzala, A.M. 1995.** An evolutionary algorithm for collision free motion planning of multi-arm robots. 1st IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications, 123-130, Sheffield, Birleşik Krallık .
- Ridao, M.A., Riquelme, J., Camacho, E.F., Toro, M. 1998.** An evolutionary and local search algorithm for planning two manipulators motion. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 105-114, Haziran, 1998, Berlin, Almanya.
- Roshani, A., Roshani, A., Roshani, A., Salehi, M., Esfandyari, A. 2013.** A simulated annealing algorithm for multi-manned assembly line balancing problem. *Journal of Manufacturing Systems*, 32(1): 238-247.

- Roshani, A., Giglio, D. 2017.** Simulated annealing algorithms for the multi-manned assembly line balancing problem: minimising cycle time. *International Journal of Production Research*, 55(10): 2731-2751.
- Rubinovitz, J., Bukchin, J., Lenz, E. 1993.** RALB – A Heuristic Algorithm for Design and Balancing of Robotic Assembly Lines, *CIRP Annals*, 42(1): 497-500.
- Saha, M., Isto, P. 2006.** Multi-robot motion planning by incremental coordination. International Conference on Intelligent Robots and Systems, 5960-5963, 9-15 Ekim, 2006, Pekin, Çin.
- Salveson, M.E. 1955.** The assembly line balancing problem. *J. Indust. Engrg*, VI: 18–25.
- Scholl, A., Klein, R. 1997.** Salome: A Bidirectional Branch-and-Bound Procedure for Assembly Line Balancing. *INFORMS Journal on Computing*, 9(4): 319-334.
- Scholl, A. (1999).** Balancing and sequencing of assembly lines. (Contributions to Management Science) Physica.
- Scholl, A., Becker, C. 2006.** State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168: 666–693.
- Sepahi, A., Naini, S. J. 2014.** Multi-manned Assembly Line Balancing Problem with Variable Task Times. *European Journal of Academic Essays* 1(4): 68-75.
- Shih, C.L., Sadler, J.P., & Gruver, W.A. 1991.** Collision avoidance for two SCARA robots. International Conference on Robotics and Automation, 674-679, Nisan, 1991, Kaliforniya, Amerika Birleşik Devletleri.
- Spensieri, D., Bohlin, R., Carlson, J.S. 2013.** Coordination of robot paths for cycle time minimization. International Conference on Automation Science and Engineering (CASE), 522-527, Ağustos, 2013.
- Stone, H.W. 1987.** Kinematic modeling, identification and control of robotic manipulators. *Kluwer Academic Publishers*, Carnegie-Mellon University, Norwell, Massachusetts, 7-13 pp.
- Sutdhiraksa, S., Zurawski, R. 1996.** Robotic assembly tasks scheduling using Petri nets and heuristic functions. The IEEE International Conference Industrial Technology, 131-135, 1996.
- Svestka, P., Overmars, M.H. 1998.** Coordinated path planning for multiple robots. *Robotics and autonomous systems*, 23(3): 125-152.
- Todt, E., Rausch, G., Suárez, R. 2000.** Analysis and classification of multiple robot coordination methods. International Conference on Robotics and Automation, 4: 3158-3163, Nisan, 2000, Kaliforniya, Amerika Birleşik Devletleri.
- Wu, H., Deng, H., Chen, L., Guan, Y., Zhang, H. 2016.** Sequence-modification based collision-free motion planning of multiple robots workcell. International Conference on Robotics and Biomimetics (ROBIO), 1135-1140, 3-7 Aralık, 2016 Qindao, Çin.
- Yazdanparast, V., Hajhosseini, H. 2011.** Multi-manned production lines with labor concentration. *Australian Journal of Basic and Applied Sciences*, 5(6): 839-846.
- Yeh, D.H., Kao, H.H. 2009.** A new bidirectional heuristic for the assembly line balancing problem. *Computers & Industrial Engineering*, 57(4), 1155-1160.
- Yılmaz, H., Yılmaz, M. 2015.** Multi-manned assembly line balancing problem with balanced load density. *Assembly Automation*, 35(1): 137-142.
- Yılmaz, H., Yılmaz, M. 2016.** A multi-manned assembly line balancing problem with classified teams: a new approach. *Assembly Automation*, 36(1): 51-59.

- Yoon, H.J., Chung, S.Y., Hwang, M.J. 2017.** Shadow space modeling for task planning of dual manipulators. 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 28 Haziran – 1 Temmuz, 2017, Jeju, Kore.
- Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., Ardakani, H.D. 2012.** Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. *Journal of Manufacturing Systems*, 31: 139-151.
- Zhang, L., Pan, J., Manocha, D. 2009.** Motion planning of human-like robots using constrained coordination. International Conference on Humanoid Robots, 188-195, 7-10 Aralık, Paris, Fransa.
- Zixiang, L., Nilakantan, J.M., Qiuhua, T., Nielsen, P. 2017.** Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line. *Engineering Optimization*, 50(5): 877-893.
- Zurawski, R., Phang, S. 1992.** Path planning for robot arms operating in a common workspace. In Industrial Electronics, Control, Instrumentation, and Automation, Power Electronics and Motion Control., Proceedings of the International Conference, 618-623, Kasım, 1992 Melbourne, Avustralya.

EKLER

- EK 1** Örnek Problem için Görev Koordinatları
- EK 2** MATLAB Kodu
- EK 3** İstasyon Süreleri (Çevrim Süresi 70 sn)
- EK 4** Robotik Manipulatörlerin Çalışma Süreleri (70 sn)
- EK 5** İstasyon Süreleri (Çevrim Süresi 50 sn)
- EK 6** İstasyon Süreleri (Çevrim Süresi 60 sn)
- EK 7** İstasyon Süreleri (Çevrim Süresi 80 sn)
- EK 8** İstasyon Süreleri (Çevrim Süresi 90 sn)

EK 1 Örnek Problem için Görev Koordinatları

Koordinatlar	Görev grubu 1		
	x	y	z
Görev 1	0,6260	0,1209	0,0495
Görev 2	0,1891	0,4904	0,0821
Görev 3	0,0427	0,8530	0,1057
Görev 4	0,2819	0,2703	0,1665
Görev 5	0,5386	0,2085	0,6210
Görev 6	0,7060	0,4387	0,2760

Koordinatlar	Görev grubu 3		
	x	y	z
Görev 1	0,4893	0,1835	0,1948
Görev 2	0,3377	0,3685	0,2259
Görev 3	0,2417	0,4868	0,4302
Görev 4	0,3532	0,3786	0,2622
Görev 5	0,0430	0,3507	0,2217
Görev 6	0,0292	0,7791	0,0714

Koordinatlar	Görev grubu 5		
	x	y	z
Görev 1	0,0318	0,3816	0,6797
Görev 2	0,0462	0,7952	0,1626
Görev 3	0,0971	0,1869	0,1190
Görev 4	0,3171	0,6463	0,3404
Görev 5	0,4588	0,1978	0,1499
Görev 6	0,4401	0,3225	0,2362

EK 1 (devamı) Örnek Problem için Görev Koordinatları

Koordinatlar	Görev grubu 7		
	x	y	z
Görev 1	0,0293	0,4688	0,0611
Görev 2	0,2682	0,6029	0,1101
Görev 3	0,0557	0,3335	0,0341
Görev 4	0,0341	0,6927	0,4366
Görev 5	0,1692	0,6330	0,5004
Görev 6	0,4065	0,6808	0,2465

Koordinatlar	Görev grubu 2		
	x	y	z
Görev 1	0,0344	0,7547	0,2238
Görev 2	0,1386	0,2511	0,7537
Görev 3	0,1493	0,6160	0,3804
Görev 4	0,2575	0,4733	0,5678
Görev 5	0,0497	0,2963	0,3012
Görev 6	0,1537	0,4609	0,2089

Koordinatlar	Görev grubu 4		
	x	y	z
Görev 1	0,0377	0,1829	0,1734
Görev 2	0,0987	0,4899	0,0605
Görev 3	0,2619	0,1679	0,3993
Görev 4	0,1366	0,5005	0,6569
Görev 5	0,4235	0,0012	0,6790
Görev 6	0,0908	0,4624	0,6358

EK 1 (devamı) Örnek Problem için Görev Koordinatları

Koordinatlar	Görev grubu 6		
	x	y	z
Görev 1	0,5181	0,1759	0,4587
Görev 2	0,2407	0,3411	0,6620
Görev 3	0,6951	0,2428	0,2564
Görev 4	0,2548	0,2691	0,5822
Görev 5	0,3445	0,0911	0,3181
Görev 6	0,0067	0,5466	0,6456

Koordinatlar	Görev grubu 8		
	x	y	z
Görev 1	0,3460	0,3125	0,0130
Görev 2	0,2100	0,4907	0,0335
Görev 3	0,5550	0,1042	0,5678
Görev 4	0,2746	0,0710	0,0666
Görev 5	0,2682	0,1296	0,7290
Görev 6	0,7386	0,3138	0,1114

EK 2 MATLAB Kodu

```
startup_rvc
mdl_puma560
%İŞ KÜMESİ 1
Tcoordfix1=[0.6260 , 0.1209 , 0.0495 ; 0.1891 , 0.4904 ,
0.0821; 0.0427 , 0.8530 , 0.1057 ; 0.2819 , 0.2703 , 0.1665;
0.5386 , 0.2085 , 0.6210 ; 0.7060 , 0.4387 , 0.2760];
Tcoord1 = Tcoordfix1;
%İŞ KÜMESİ 2
Tcoordfix2 = [0.0344, 0.7547, 0.2238; 0.1386, 0.2511, 0.7537;
0.1493, 0.6160, 0.3804; 0.2575, 0.4733, 0.5678; 0.0497, 0.2963,
0.3012; 0.1537, 0.4609, 0.2089];
Tcoord2 = Tcoordfix2;
%İŞ KÜMESİ 3
Tcoordfix3 = [0.4893, 0.1835, 0.1948; 0.3377, 0.3685, 0.2259;
0.2417, 0.4868, 0.4302; 0.3532, 0.3786, 0.2622; 0.0430, 0.3507,
0.2217; 0.0292, 0.7791, 0.0714];
Tcoord3 = Tcoordfix3;
%İŞ KÜMESİ 4
Tcoordfix4 = [0.0377, 0.1829, 0.1734; 0.0987, 0.4899, 0.0605;
0.2619, 0.1679, 0.3993; 0.1366, 0.5005, 0.6569; 0.4235, 0.0012,
0.6790; 0.0908, 0.4624, 0.6358];
Tcoord4 = Tcoordfix4;
%İŞ KÜMESİ 5
Tcoordfix5 = [0.0318 , 0.3816 , 0.6797 ;0.0462 , 0.7952 ,
0.1626; 0.0971 , 0.1869 , 0.1190 ; 0.3171 , 0.6463 , 0.3404;
0.4588, 0.1978, 0.1499; 0.4401, 0.3225, 0.2362];
Tcoord5=Tcoordfix5;
%İŞ KÜMESİ 6
Tcoordfix6=[0.5181, 0.1759, 0.4587; 0.2407, 0.3411, 0.6620;
0.6951, 0.2428, 0.2564; 0.2548, 0.2691, 0.5822; 0.3445, 0.0911,
0.3181; 0.0067, 0.5466, 0.6456];
Tcoord6=Tcoordfix6;
%İŞ KÜMESİ 7
Tcoordfix7 = [0.0293, 0.4688, 0.0611; 0.2682, 0.6029, 0.1101;
0.0557, 0.3335, 0.0341; 0.0341, 0.6927, 0.4366; 0.1692, 0.6330,
0.5004; 0.4065, 0.6808, 0.2465];
Tcoord7 = Tcoordfix7;
%İŞ KÜMESİ 8
Tcoordfix8 = [0.3460, 0.3125, 0.0130; 0.2100, 0.4907, 0.0335;
0.5550, 0.1042, 0.5678; 0.2746, 0.0710, 0.0666; 0.2682, 0.1296,
0.7290; 0.7386, 0.3138, 0.1114];
Tcoord8 = Tcoordfix8;
rng('default');
%Rastgele dizi elemanlarını sıralama
Gorevler = randperm(8)
for i=1:8
    switch Gorevler(i)
        case 1
            Oncelik(i)=0;
        case 2
```

EK 2 (devamı) MATLAB Kodu

```
        Oncelik(i)=1;
    case 3
        Oncelik(i)=1;
    case 4
        Oncelik(i)=0;
    case 5
        Oncelik(i)=2;
    case 6
        Oncelik(i)=4;
    case 7
        Oncelik(i)=5;
    case 8
        Oncelik(i)=6;
    end
end
Oncelik
xco =[ ];
yco =[ ];
Atanan=(0);
Oncelik;
aa=1;bb=1;
x=length(Gorevler);
while length(Gorevler)>=1 && aa>=1 && bb>=1
    for aa=1:x
        for bb=1:length(Atanan)
            if Oncelik(aa)==Atanan(bb)
                Atanan=[Atanan Gorevler(aa)];
                Gorevler(aa)=[ ];
                Oncelik(aa)=[ ];
                break
                aa=1;
            end
        end
    end
    aa;
    if aa==length(Gorevler)||aa>length(Gorevler)
        break
    end
end
aa=1;
end
Atanan(1)=[ ]; Atanan
% tüm koordinatların matris hesabı
Acoords = [ ];
for j=1:8
    switch Atanan(j)
        case 1
            Acoords = [Acoords;Tcoordfix1];
        case 2
            Acoords = [Acoords;Tcoordfix2];
        case 3
```

EK 2 (devamı) MATLAB Kodu

```
        Acoords = [Acoords;Tcoordfix3];
    case 4
        Acoords = [Acoords;Tcoordfix4];
    case 5
        Acoords = [Acoords;Tcoordfix5];
    case 6
        Acoords = [Acoords;Tcoordfix6];
    case 7
        Acoords = [Acoords;Tcoordfix7];
    case 8
        Acoords = [Acoords;Tcoordfix8];
    end
end
%Yöntemin başlangıcı
a=1;
uy_fon_eniyi=999;
while a<=İterasyonsayısı
    ist1=0;ist2=0;ist3=0;ist4=0;
    ist1array=[ ];
    ist2array=[ ];
    ist3array=[ ];
    ist4array=[ ];
    UUU=[ ];
    Stimes=0;
    Allcoords=Acoords;
    Ctime =70;
    for tsk=1:8
        format bank
        THome = transl(0.3,0.3,0.4);
        TBhome = transl(0.6,-0.2,0.4);
        Ahome = [0.3,0.3,0.4]; % A robotunun home koordinatları
        Bhome = [0.6,-0.2,0.4]; % B robotunun home koordinatları
        %mesafe matrisi hesabı
        MesafeMatrisi=zeros(6,6);
        for p=1:6
            for r=1:6
                MesafeMatrisi(p,r) = round(sqrt((Allcoords(p,1)-
                Allcoords(r,1))^2+...
                (Allcoords(p,2)-Allcoords(r,2))^2+(Allcoords(p,3)-
                Allcoords(r,3))^2)*10);
            end
        end
    end
    l=1;
    while l<100
        RATdurations = [ ];
        RBTdurations = [ ];
        TasksA = (Ahome);
        TasksB = (Bhome);
        P = randperm(6);
        %Görev gruplarındaki görevlerin robotlara dağıtılması
```

EK 2 (devamı) MATLAB Kodu

```
TasksA(2,:) = Allcoords(P(1),:);
TasksB(2,:) = Allcoords(P(2),:);
Ahome2Tldur = round(sqrt((TasksA(1,1)-TasksA(2,1))^2+...
    (TasksA(1,2)-TasksA(2,2))^2+(TasksA(1,3)-
TasksA(2,3))^2)*10);
    Bhome2Tldur = round(sqrt((TasksB(1,1)-TasksB(2,1))^2+...
(TasksB(1,2)-TasksB(2,2))^2+(TasksB(1,3)-TasksB(2,3))^2)*10);
    RATdurations = [RATdurations Ahome2Tldur];
    RBTdurations = [RBTdurations Ahome2Tldur];
    TasksA(3,:)=TasksA(2,:);
    TotalRATduration = Ahome2Tldur+2;
    RATdurations=[RATdurations 2];
    TasksB(3,:)=TasksB(2,:);
    TotalRBTduration = Bhome2Tldur + 2;
    RBTdurations=[RBTdurations 2];
    if mod(6,2) == 0
        for i=4:2:6
            TasksA(i,:)=Allcoords(P(i-1),:);
            TotalRATduration = TotalRATduration +
MesafeMatrisi(P(i-1),P(i-3));
            RATdurations =[RATdurations MesafeMatrisi(P(i-
1),P(i-3))];
            TasksA(i+1,:) = Allcoords(P(i-1),:);
            TotalRATdurations = TotalRATduration + 2;
            RATdurations = [RATdurations 2];
            TasksB(i,:)=Allcoords(P(i),:);
            TotalRBTduration = TotalRBTduration +
MesafeMatrisi(P(i),P(i-2));
            RBTdurations =[RBTdurations
MesafeMatrisi(P(i),P(i-2))];
            TasksB(i+1,:) = Allcoords(P(i),:);
            TotalRBTdurations = TotalRBTduration + 2;
            RBTdurations = [RBTdurations 2];
        end
    else
        for i=3:2:6
            TasksA(i+1,:)=Allcoords(P(i),:);
            TotalRATduration = TotalRATduration +
MesafeMatrisi(P(i),P(i-2));
            RATdurations =[RATdurations
MesafeMatrisi(P(i),P(i-2))];
            TasksA(i+2,:) = Allcoords(P(i),:);
            TotalRATdurations = TotalRATduration + 2;
            RATdurations = [RATdurations 2];
        end
        for i=4:2:6-1
            TasksB(i,:)=Allcoords(P(i),:);
            TotalRBTduration = TotalRBTduration +
MesafeMatrisi(P(i),P(i-2));
```

EK 2 (devamı) MATLAB Kodu

```
        RBTdurations =[RBTdurations
MesafeMatrisi(P(i),P(i-2))];
        TasksB(i+1,:) = Allcoords(P(i),:);
        TotalRBTdurations = TotalRBTduration + 2;
        RBTdurations = [RBTdurations 2];
    end
end
TasksA;
TasksB;
TotalRATduration;
TotalRBTduration;
RATdurations;
RBTdurations;
Stimes=Stimes+abs(TotalRATduration-TotalRBTduration); %
aylak zaman hesabı
% T ve q'ların belirlenmesi
% Robot A için:
TAAs=[];
TAAe=[];
qAAs=[];
qAAe=[];
for i=1:length(TasksA)-1
    TAs=transl(TasksA(i,:));
    TAe=transl(TasksA(i+1,:));
    TAAs(:, :, i)=TAs;
    TAAe(:, :, i)=TAe;
    qAs=p560.ikine6s(TAs);
    qAAs(i, :)=qAs;
    qAe=p560.ikine6s(TAe);
    qAAe(i, :)=qAe;
end
% Robot B için:
TBBs=[];
TBBe=[];
qBBs=[];
qBBe=[];
for i=1:length(TasksB)-1
    TBs=transl(TasksB(i,:));
    TBe=transl(TasksB(i+1,:));
    TBBs(:, :, i)=TBs;
    TBBe(:, :, i+1)=TBe;
    qBs=p560.ikine6s(TBs);
    qBBs(i, :)=qBs;
    qBe=p560.ikine6s(TBe);
    qBBe(i, :)=qBe;
end
qAA=[];
qBB=[];
tsA=0;
tsB=0;
```

EK 2 (devamı) MATLAB Kodu

```
teA=RATdurations(1);
teB=RBTdurations(1);
tAA=[];
tBB=[];
for z=1:length(RATdurations)
    teA=RATdurations(z)+tsA;
    tA=[tsA:0.05:teA]';
    tsA=teA;
    qA =jtraj(qAAs(z,:),qAAe(z,:),tA);
    qAA = [qAA ;qA];
    tAA=[tAA ;tA];
end
for y=1:length(RBTdurations)
    teB=RBTdurations(y)+tsB;
    tB=[tsB:0.05:teB]';
    tsB=teB ;
    qB =jtraj(qBBs(y,:),qBBe(y,:),tB);
    qBB = [qBB;qB];
    tBB=[tBB; tB]; %boyutu artan matris
end
tBB;
tAA;
%Bağlantı koordinatlarının belirlenmesi
for i=1:min(length(tAA),length(tBB))
    TA1=p560.links(1).A(qAA(i,1));
    TA2=p560.links(2).A(qAA(i,2));
    TA3=p560.links(3).A(qAA(i,3));
    TTA1=double(TA1);
    TTA2=double(TA2);
    TTA3=double(TA3);
    LinkA1coord=TTA1; %Bağlantı 1'in başlangıç noktasının
koordinatları (base konumu)
    LinkA2coord=TTA1*TTA2;
    midpointA1x=((LinkA2coord(1,4)-
LinkA1coord(1,4))/2)+LinkA1coord(1,4); %İlk bağlantının orta
noktasının x değeri
    midpointA1y=((LinkA2coord(2,4)-
LinkA1coord(2,4))/2)+LinkA1coord(2,4); %İlk bağlantının orta
noktasının y değeri
    midpointA1z=((LinkA2coord(3,4)-
LinkA1coord(3,4))/2)+LinkA1coord(3,4); %İlk bağlantının orta
noktasının z değeri
    SplitA1a=[((midpointA1x-
LinkA1coord(1,4))/2)+LinkA1coord(1,4),...
((midpointA1y-
LinkA1coord(2,4))/2)+LinkA1coord(2,4),...
((midpointA1z-
LinkA1coord(3,4))/2)+LinkA1coord(3,4)];
    SplitA1b=[(LinkA2coord(1,4)-
midpointA1x)/2+midpointA1x,...
```

EK 2 (devamı) MATLAB Kodu

```
(LinkA2coord(2,4)-
midpointA1y)/2+midpointA1y,(LinkA2coord(3,4)-
midpointA1z)/2+midpointA1z];
    LinkA3coord=TTA1*TTA2*TTA3;
    midpointA2x=( (LinkA3coord(1,4)-
LinkA2coord(1,4))/2)+LinkA2coord(1,4);
    midpointA2y=( (LinkA3coord(2,4)-
LinkA2coord(2,4))/2)+LinkA2coord(2,4);
    midpointA2z=( (LinkA3coord(3,4)-
LinkA2coord(3,4))/2)+LinkA2coord(3,4);
    SplitA2a=[ (midpointA2x-
LinkA2coord(1,4))/2+LinkA2coord(1,4),...
    (midpointA2y-
LinkA2coord(2,4))/2+LinkA2coord(2,4),...
    (midpointA2z-
LinkA2coord(3,4))/2+LinkA2coord(3,4)];
    SplitA2b=[ (LinkA3coord(1,4)-
midpointA2x)/2+midpointA2x,...
    (LinkA3coord(2,4)-
midpointA2y)/2+midpointA2y,(LinkA3coord(3,4)-
midpointA2z)/2+midpointA2z];
    TAendeffector=p560.fkine(qAA(i,:));
    TTAendeffector=double(TAendeffector);
    midpointA3x=( (TTAendeffector(1,4)-
LinkA3coord(1,4))/2)+LinkA3coord(1,4);
    midpointA3y=( (TTAendeffector(2,4)-
LinkA3coord(2,4))/2)+LinkA3coord(2,4);
    midpointA3z=( (TTAendeffector(3,4)-
LinkA3coord(3,4))/2)+LinkA3coord(3,4);
    SplitA3a=[ (midpointA3x-
LinkA3coord(1,4))/2+LinkA3coord(1,4),...
    (midpointA3y-
LinkA3coord(2,4))/2+LinkA3coord(2,4),...
    (midpointA3z-
LinkA3coord(3,4))/2+LinkA3coord(3,4)];
    SplitA3b=[ (TTAendeffector(1,4)-
midpointA3x)/2+midpointA3x,...
    (TTAendeffector(2,4)-
midpointA3y)/2+midpointA3y,(TTAendeffector(3,4)-
midpointA3z)/2+midpointA3z];
RobotAmatris=[SplitA1a(1,1),SplitA1a(1,2),SplitA1a(1,3);midpoint
A1x,midpointA1y,midpointA1z;SplitA1b(1,1),SplitA1b(1,2),SplitA1b
(1,3);LinkA2coord(1,4),LinkA2coord(2,4),LinkA2coord(3,4);...
SplitA2a(1,1),SplitA2a(1,2),SplitA2a(1,3);midpointA2x,midpointA2
y,midpointA2z;SplitA2b(1,1),SplitA2b(1,2),SplitA2b(1,3);LinkA3co
ord(1,4),LinkA3coord(2,4),LinkA3coord(3,4);...
SplitA3a(1,1),SplitA3a(1,2),SplitA3a(1,3);midpointA3x,midpointA3
y,midpointA3z;SplitA3b(1,1),SplitA3b(1,2),SplitA3b(1,3);TTAendef
fector(1,4),TTAendeffector(2,4),TTAendeffector(3,4)];
    format bank,(RobotAmatris);
```


EK 2 (devamı) MATLAB Kodu

```
R1 = sprintf('%5.2f',RobotAmatris);
R1B = sscanf(R1,'%f');

R1R=[R1B(1,1),R1B(13,1),R1B(25,1);R1B(2,1),R1B(14,1),R1B(26,1);R
1B(3,1),R1B(15,1),R1B(27,1);...

R1B(4,1),R1B(16,1),R1B(28,1);R1B(5,1),R1B(17,1),R1B(29,1);R1B(6,
1),R1B(18,1),R1B(30,1);...

R1B(7,1),R1B(19,1),R1B(31,1);R1B(8,1),R1B(20,1),R1B(32,1);R1B(9,
1),R1B(21,1),R1B(33,1);...

R1B(10,1),R1B(22,1),R1B(34,1);R1B(11,1),R1B(23,1),R1B(35,1);R1B(
12,1),R1B(24,1),R1B(36,1)];

%Robot B
TB1=p560_2.links(1).A(qBB(i,1));
TB2=p560_2.links(2).A(qBB(i,2));
TB3=p560_2.links(3).A(qBB(i,3));
TTB1=double(TB1);
TTB2=double(TB2);
TTB3=double(TB3);
LinkB1coord=TTB1+R2base;
LinkB2coord=(TTB1*TTB2)+R2base;
midpointB1x=((LinkB2coord(1,4)-
LinkB1coord(1,4))/2)+LinkB1coord(1,4);%İlk kolun orta noktasının
x değeri
midpointB1y=((LinkB2coord(2,4)-
LinkB1coord(2,4))/2)+LinkB1coord(2,4);%İlk kolun orta noktasının
y değeri
midpointB1z=((LinkB2coord(3,4)-
LinkB1coord(3,4))/2)+LinkB1coord(3,4);%İlk kolun orta noktasının
z değeri
SplitB1a=[((midpointB1x-
LinkB1coord(1,4))/2)+LinkB1coord(1,4),...
((midpointB1y-
LinkB1coord(2,4))/2)+LinkB1coord(2,4),...
((midpointB1z-
LinkB1coord(3,4))/2)+LinkB1coord(3,4)];
SplitB1b=[(LinkB2coord(1,4)-
midpointB1x)/2+midpointB1x,...
(LinkB2coord(2,4)-
midpointB1y)/2+midpointB1y,(LinkB2coord(3,4)-
midpointB1z)/2+midpointB1z];
LinkB3coord=(TTB1*TTB2*TTB3)+R2base;
midpointB2x=((LinkB3coord(1,4)-
LinkB2coord(1,4))/2)+LinkB2coord(1,4);
midpointB2y=((LinkB3coord(2,4)-
LinkB2coord(2,4))/2)+LinkB2coord(2,4);
```

EK 2 (devamı) MATLAB Kodu

```
midpointB2z=((LinkB3coord(3,4)-
LinkB2coord(3,4))/2)+LinkB2coord(3,4);
SplitB2a=[(midpointB2x-
LinkB2coord(1,4))/2)+LinkB2coord(1,4),...
((midpointB2y-
LinkB2coord(2,4))/2)+LinkB2coord(2,4),...
((midpointB2z-
LinkB2coord(3,4))/2)+LinkB2coord(3,4)];
SplitB2b=[(LinkB3coord(1,4)-
midpointB2x)/2+midpointB2x,...
(LinkB3coord(2,4)-
midpointB2y)/2+midpointB2y,(LinkB3coord(3,4)-
midpointB2z)/2+midpointB2z];
TBendeffector=p560_2.fkine(qBB(i,:));
TTBendeffector=double(TBendeffector);
midpointB3x=((TTBendeffector(1,4)-
LinkB3coord(1,4))/2)+LinkB3coord(1,4);
midpointB3y=((TTBendeffector(2,4)-
LinkB3coord(2,4))/2)+LinkB3coord(2,4);
midpointB3z=((TTBendeffector(3,4)-
LinkB3coord(3,4))/2)+LinkB3coord(3,4);
SplitB3a=[(midpointB3x-
LinkB3coord(1,4))/2)+LinkB3coord(1,4),...
((midpointB3y-LinkB3coord(2,4))/2)+LinkB3coord(2,4),...
((midpointB3z-
LinkB3coord(3,4))/2)+LinkB3coord(3,4)];
SplitB3b=[(TTBendeffector(1,4)-
midpointB3x)/2+midpointB3x,...
(TTBendeffector(2,4)-
midpointB3y)/2+midpointB3y,(TTBendeffector(3,4)-
midpointB3z)/2+midpointB3z];
RobotBmatris=[SplitB1a(1,1),SplitB1a(1,2),SplitB1a(1,3);midpoint
B1x,midpointB1y,midpointB1z;SplitB1b(1,1),SplitB1b(1,2),SplitB1b
(1,3);LinkB2coord(1,4),LinkB2coord(2,4),LinkB2coord(3,4);...
SplitB2a(1,1),SplitB2a(1,2),SplitB2a(1,3);midpointB2x,midpointB2
y,midpointB2z;SplitB2b(1,1),SplitB2b(1,2),SplitB2b(1,3);LinkB3co
ord(1,4),LinkB3coord(2,4),LinkB3coord(3,4);...
SplitB3a(1,1),SplitB3a(1,2),SplitB3a(1,3);midpointB3x,midpointB3
y,midpointB3z;SplitB3b(1,1),SplitB3b(1,2),SplitB3b(1,3);TTBendef
fector(1,4),TTBendeffector(2,4),TTBendeffector(3,4)];
format bank,(RobotBmatris); %virgülden sonra 2 basamak
almak için
R2 = sprintf('%5.2f',RobotBmatris);%sayıları string
yapmak için
R2B = sscanf(R2,'%f'); %sting ifadeleri sayı yapmak için

R2R=[R2B(1,1),R2B(13,1),R2B(25,1);R2B(2,1),R2B(14,1),R2B(26,1);R
2B(3,1),R2B(15,1),R2B(27,1);...
```

EK 2 (devamı) MATLAB Kodu

```
R2B(4,1),R2B(16,1),R2B(28,1);R2B(5,1),R2B(17,1),R2B(29,1);R2B(6,
1),R2B(18,1),R2B(30,1);...

R2B(7,1),R2B(19,1),R2B(31,1);R2B(8,1),R2B(20,1),R2B(32,1);R2B(9,
1),R2B(21,1),R2B(33,1);...

R2B(10,1),R2B(22,1),R2B(34,1);R2B(11,1),R2B(23,1),R2B(35,1);R2B(
12,1),R2B(24,1),R2B(36,1)]; %Nihai koordinat matrisi
    for j=1:12
        for k=1:12
            if
R1R(j,1)==R2R(k,1)&&R1R(j,2)==R2R(k,2)&&R1R(j,3)==R2R(k,3)
                l= l+1;
            else
                l=100;
            end
        end
    end
end
end
end
end
uu = max(TotalRATduration,TotalRBTduration);
Allcoords(1,:)=[];
Allcoords(1,:)=[];
Allcoords(1,:)=[];
Allcoords(1,:)=[];
Allcoords(1,:)=[];
Allcoords(1,:)=[];
UUU = [UUU uu];
end
dr=1;
for dr=1:8
    ist1=ist1+UUU(dr);
    ist1array=[ist1array UUU(dr)];
    if ist1 > Ctime
        ist1=ist1-UUU(dr);
        ist1array(end)=[];
        ist2=ist2+UUU(dr);
        ist2array=[ist2array UUU(dr)];
        if ist2 > Ctime
            ist2=ist2-UUU(dr);
            ist2array(end)=[];
            ist3=ist3+UUU(dr);
            ist3array=[ist3array UUU(dr)];
            if ist3 > Ctime
                ist3=ist3-UUU(dr);
                ist3array(end)=[];
                ist4=ist4+UUU(dr);
                ist4array=[ist4array UUU(dr)];
            end
        end
    end
end
end
```

EK 2 (devamı) MATLAB Kodu

```
        end
    end
    if ist3==0 && ist4==0
        uy_fon_aday=(Ctime-ist1)+(Ctime-ist2)+Stimes
    elseif ist4==0
        uy_fon_aday=(Ctime-ist1)+(Ctime-ist2)+(Ctime-
ist3)+Stimes
    else
        uy_fon_aday=(Ctime-ist1)+(Ctime-ist2)+(Ctime-
ist3)+(Ctime-ist4)+Stimes
    end
    if uy_fon_aday < uy_fon_eniyi
        uy_fon_eniyi = uy_fon_aday;
    end
    xco = [ xco a];
    yco = [ yco uy_fon_eniyi];
    a=a+1;
end
uy_fon_eniyi
plot(xco,yco,'--gs',...
'LineWidth',2,...
'MarkerSize',5,...
'MarkerEdgeColor','b', 'MarkerFaceColor',[0.5,0.5,0.5])
```

EK 3 İstasyon Süreleri (Çevrim Süresi 70 sn)

İstasyon Sayısı	İstasyon 1'in toplam süresi	İstasyon 2'nin toplam süresi	İstasyon 3'ün toplam süresi	İstasyon 4'ün toplam süresi	Uygunluk Fonksiyonu Değeri (çev. sür. 70)
1	64.00	55.00	39.00	0,00	86.00
2	65.00	55.00	36.00	0,00	96.00
3	59.00	60.00	39.00	0,00	98.00
4	56.00	60.00	39.00	0,00	94.00
5	57.00	57.00	36.00	0,00	84.00
6	61.00	56.00	45.00	0,00	96.00
7	56.00	69.00	21.00	0,00	91.00
8	59.00	70.00	19.00	0,00	89.00
9	58.00	62.00	38.00	0,00	90.00
10	62.00	68.00	28.00	0,00	99.00
11	56.00	70.00	18.00	0,00	96.00
12	68.00	69.00	0,00	0,00	36.00
13	58.00	62.00	37.00	0,00	89.00
14	64.00	63.00	38.00	0,00	97.00
15	59.00	58.00	43.00	0,00	87.00
16	62.00	55.00	42.00	0,00	98.00
17	59.00	68.00	19.00	0,00	101.00
18	63.00	69.00	17.00	0,00	97.00
19	69.00	56.00	16.00	0,00	101.00
20	62.00	61.00	45.00	0,00	86.00
21	69.00	63.00	18.00	0,00	100.00
22	59.00	55.00	37.00	0,00	90.00
23	56.00	55.00	47.00	0,00	87.00
24	55.00	57.00	40.00	0,00	98.00
25	62.00	54.00	37.00	0,00	85.00
26	60.00	60.00	38.00	0,00	96.00
27	70.00	62.00	17.00	0,00	91.00
28	55.00	55.00	43.00	0,00	82.00
29	59.00	51.00	40.00	0,00	97.00
30	60.00	69.00	20.00	0,00	89.00
31	70.00	68.00	0,00	0,00	21.00*
32	58.00	53.00	39.00	0,00	94.00
33	69.00	60.00	41.00	0,00	93.00
34	66.00	57.00	38.00	0,00	96.00
35	60.00	64.00	41.00	0,00	97.00
36	58.00	57.00	35.00	0,00	101.00
37	65.00	63.00	45.00	0,00	98.00
38	63.00	56.00	37.00	0,00	94.00
39	68.00	57.00	20.00	0,00	101.00
40	68.00	57.00	20.00	0,00	85.00
41	66.00	55.00	40.00	0,00	90.00
42	59.00	69.00	22.00	0,00	104.00
43	65.00	65.00	44.00	0,00	94.00
44	69.00	70.00	19.00	0,00	89.00
45	60.00	59.00	37.00	0,00	101.00
46	61.00	56.00	39.00	0,00	97.00
47	64.00	55.00	36.00	0,00	88.00
48	70.00	59.00	16.00	0,00	87.00
49	60.00	63.00	45.00	0,00	102.00
50	69.00	63.00	18.00	0,00	98.00

EK 4 Robotik Manipulatörlerin Çalışma Süreleri (70 sn)

İterasyon Sayısı	Robotlar	1. görev grubu	4. görev grubu	6. görev grubu	8. görev grubu	2. görev grubu	3. görev grubu	5. görev grubu	7. görev grubu	Uygunluk Fonksiyonu Değeri
1	Robot 1	26,00	21,00	13,00	21,00	18,00	12,00	20,00	12,00	86,00
	Robot 2	17,00	15,00	17,00	19,00	17,00	16,00	19,00	19,00	
2	Robot 1	14,00	16,00	11,00	21,00	12,00	16,00	18,00	16,00	96,00
	Robot2	26,00	18,00	21,00	16,00	18,00	15,00	14,00	18,00	
3	Robot 1	20,00	15,00	15,00	15,00	13,00	8,00	13,00	16,00	98,00
	Robot 2	25,00	19,00	12,00	20,00	18,00	22,00	20,00	19,00	
4	Robot 1	24,00	14,00	16,00	17,00	12,00	10,00	22,00	15,00	94,00
	Robot2	19,00	16,00	11,00	21,00	18,00	21,00	18,00	17,00	
5	Robot 1	16,00	20,00	16,00	19,00	11,00	16,00	18,00	13,00	84,00
	Robot 2	21,00	18,00	15,00	23,00	17,00	17,00	18,00	18,00	
6	Robot 1	15,00	16,00	15,00	15,00	10,00	17,00	14,00	16,00	96,00
	Robot2	23,00	23,00	14,00	16,00	23,00	14,00	28,00	17,00	
7	Robot 1	19,00	13,00	15,00	15,00	13,00	15,00	16,00	15,00	91,00
	Robot 2	17,00	20,00	17,00	18,00	18,00	15,00	21,00	18,00	
8	Robot 1	16,00	13,00	11,00	16,00	16,00	12,00	20,00	17,00	89,00
	Robot2	25,00	18,00	16,00	17,00	17,00	15,00	21,00	19,00	
9	Robot 1	17,00	18,00	15,00	17,00	12,00	10,00	18,00	15,00	90,00
	Robot 2	24,00	16,00	16,00	24,00	21,00	17,00	21,00	17,00	
10	Robot 1	17,00	13,00	13,00	16,00	13,00	18,00	16,00	12,00	99,00
	Robot2	22,00	20,00	20,00	15,00	20,00	12,00	28,00	14,00	
11	Robot 1	20,00	13,00	16,00	17,00	12,00	10,00	16,00	11,00	96,00
	Robot 2	23,00	17,00	15,00	18,00	19,00	16,00	18,00	17,00	
12	Robot 1	16,00	11,00	16,00	21,00	14,00	6,00	12,00	12,00	36,00
	Robot2	18,00	13,00	14,00	19,00	15,00	18,00	22,00	14,00	
13	Robot 1	20,00	18,00	14,00	15,00	18,00	12,00	16,00	12,00	89,00
	Robot 2	23,00	17,00	17,00	27,00	15,00	17,00	19,00	18,00	
14	Robot 1	21,00	14,00	9,00	19,00	9,00	12,00	16,00	17,00	97,00
	Robot2	17,00	22,00	21,00	23,00	20,00	20,00	20,00	18,00	
15	Robot 1	17,00	15,00	15,00	16,00	20,00	10,00	16,00	16,00	87,00
	Robot 2	26,00	18,00	15,00	17,00	18,00	21,00	20,00	23,00	
16	Robot 1	14,00	17,00	14,00	21,00	12,00	12,00	15,00	12,00	98,00
	Robot2	25,00	21,00	16,00	16,00	19,00	15,00	21,00	21,00	
17	Robot 1	15,00	14,00	12,00	17,00	14,00	16,00	14,00	14,00	101,00
	Robot 2	19,00	21,00	19,00	16,00	13,00	11,00	21,00	19,00	
18	Robot 1	18,00	12,00	11,00	14,00	14,00	15,00	16,00	15,00	97,00
	Robot2	16,00	24,00	21,00	17,00	19,00	15,00	18,00	17,00	
19	Robot 1	13,00	17,00	15,00	17,00	14,00	7,00	16,00	13,00	101,00
	Robot 2	19,00	16,00	13,00	20,00	18,00	16,00	20,00	16,00	
20	Robot 1	22,00	13,00	19,00	18,00	9,00	16,00	16,00	16,00	86,00
	Robot2	24,00	18,00	20,00	24,00	21,00	11,00	26,00	19,00	
21	Robot 1	16,00	17,00	12,00	20,00	12,00	7,00	15,00	16,00	100,00
	Robot 2	17,00	18,00	16,00	15,00	18,00	22,00	21,00	18,00	
22	Robot 1	22,00	15,00	17,00	16,00	14,00	12,00	18,00	14,00	90,00
	Robot2	17,00	20,00	15,00	17,00	16,00	22,00	17,00	19,00	
23	Robot 1	23,00	18,00	14,00	18,00	15,00	12,00	14,00	17,00	87,00
	Robot 2	17,00	16,00	15,00	18,00	19,00	18,00	28,00	19,00	
24	Robot 1	18,00	18,00	14,00	15,00	10,00	10,00	15,00	13,00	98,00
	Robot2	22,00	17,00	15,00	18,00	18,00	21,00	21,00	19,00	
25	Robot 1	19,00	21,00	14,00	11,00	14,00	18,00	18,00	17,00	85,00
	Robot 2	23,00	20,00	18,00	19,00	17,00	12,00	18,00	19,00	
26	Robot 1	20,00	17,00	9,00	15,00	16,00	12,00	18,00	14,00	96,00
	Robot2	16,00	17,00	23,00	25,00	19,00	16,00	15,00	20,00	
27	Robot 1	23,00	13,00	12,00	19,00	15,00	17,00	16,00	14,00	91,00
	Robot 2	16,00	16,00	14,00	20,00	19,00	14,00	23,00	17,00	

EK 4 (devamı) Robotik Manipulatörlerin Çalışma Süreleri (70 sn)

28	Robot 1	16,00	17,00	14,00	16,00	17,00	17,00	21,00	15,00	82,00
	Robot 2	15,00	16,00	22,00	16,00	22,00	15,00	20,00	22,00	
29	Robot 1	20,00	13,00	12,00	15,00	10,00	13,00	19,00	15,00	97,00
	Robot 2	16,00	23,00	16,00	20,00	18,00	13,00	20,00	20,00	
30	Robot 1	17,00	14,00	12,00	17,00	14,00	15,00	20,00	16,00	89,00
	Robot 2	20,00	21,00	19,00	18,00	18,00	11,00	20,00	18,00	
31	Robot 1	19,00	13,00	16,00	16,00	14,00	14,00	18,00	15,00	21,00
	Robot 2	16,00	19,00	13,00	16,00	14,00	15,00	21,00	18,00	
32	Robot 1	13,00	13,00	14,00	17,00	16,00	10,00	18,00	16,00	94,00
	Robot 2	26,00	18,00	13,00	17,00	18,00	18,00	19,00	20,00	
33	Robot 1	16,00	19,00	9,00	16,00	16,00	8,00	18,00	15,00	93,00
	Robot 2	27,00	22,00	20,00	19,00	17,00	24,00	21,00	20,00	
34	Robot 1	17,00	15,00	19,00	15,00	12,00	12,00	17,00	12,00	96,00
	Robot 2	24,00	23,00	14,00	20,00	25,00	12,00	19,00	19,00	
35	Robot 1	23,00	9,00	19,00	17,00	12,00	11,00	19,00	14,00	97,00
	Robot 2	16,00	18,00	15,00	19,00	21,00	24,00	20,00	21,00	
36	Robot 1	15,00	14,00	9,00	16,00	13,00	12,00	16,00	14,00	101,00
	Robot 2	21,00	16,00	21,00	16,00	19,00	22,00	19,00	16,00	
37	Robot 1	14,00	17,00	12,00	15,00	14,00	9,00	16,00	15,00	98,00
	Robot 2	27,00	18,00	20,00	20,00	19,00	24,00	24,00	21,00	
38	Robot 1	14,00	17,00	12,00	17,00	12,00	12,00	19,00	15,00	94,00
	Robot 2	25,00	21,00	17,00	20,00	18,00	18,00	17,00	18,00	
39	Robot 1	19,00	13,00	12,00	18,00	15,00	9,00	21,00	12,00	101,00
	Robot 2	17,00	19,00	12,00	15,00	18,00	18,00	16,00	20,00	
40	Robot 1	18,00	18,00	14,00	16,00	10,00	16,00	20,00	16,00	85,00
	Robot 2	19,00	17,00	14,00	17,00	21,00	16,00	18,00	20,00	
41	Robot 1	24,00	15,00	13,00	15,00	12,00	12,00	16,00	16,00	90,00
	Robot 2	21,00	22,00	20,00	22,00	19,00	14,00	20,00	20,00	
42	Robot 1	16,00	13,00	16,00	16,00	9,00	17,00	13,00	11,00	104,00
	Robot 2	15,00	23,00	20,00	16,00	17,00	13,00	22,00	19,00	
43	Robot 1	24,00	17,00	15,00	15,00	10,00	12,00	16,00	14,00	94,00
	Robot 2	17,00	24,00	17,00	22,00	25,00	18,00	23,00	21,00	
44	Robot 1	24,00	22,00	11,00	14,00	12,00	16,00	15,00	17,00	89,00
	Robot 2	18,00	19,00	23,00	21,00	18,00	15,00	15,00	19,00	
45	Robot 1	13,00	17,00	16,00	16,00	11,00	9,00	16,00	12,00	101,00
	Robot 2	25,00	19,00	15,00	22,00	18,00	19,00	20,00	17,00	
46	Robot 1	18,00	16,00	15,00	17,00	12,00	11,00	10,00	14,00	97,00
	Robot 2	19,00	26,00	16,00	19,00	17,00	20,00	22,00	17,00	
47	Robot 1	22,00	17,00	10,00	21,00	16,00	12,00	18,00	15,00	88,00
	Robot 2	20,00	22,00	20,00	17,00	22,00	10,00	17,00	18,00	
48	Robot 1	20,00	19,00	10,00	23,00	18,00	12,00	18,00	12,00	87,00
	Robot 2	21,00	17,00	16,00	16,00	18,00	14,00	18,00	16,00	
49	Robot 1	18,00	12,00	13,00	19,00	13,00	9,00	14,00	14,00	102,00
	Robot 2	16,00	23,00	19,00	17,00	17,00	27,00	25,00	20,00	
50	Robot 1	20,00	13,00	11,00	15,00	17,00	12,00	14,00	13,00	98,00
	Robot 2	17,00	18,00	16,00	17,00	21,00	15,00	25,00	18,00	

EK 5 İstasyon Süreleri (Çevrim Süresi 50 sn)

İterasyon Sayısı	İstasyon 1'in toplam süresi	İstasyon 2'nin toplam süresi	İstasyon 3'ün toplam süresi	İstasyon 4'ün toplam süresi	Uygunluk Fonksiyonu Değeri
1	47.00	38.00	34.00	39.00	76.00
2	44.00	42.00	34.00	36.00	86.00
3	44.00	35.00	40.00	39.00	88.00
4	40.00	37.00	39.00	39.00	84.00
5	41.00	39.00	34.00	36.00	74.00
6	46.00	48.00	40.00	28.00	86.00
7	39.00	50.00	39.00	18.00	81.00
8	43.00	50.00	36.00	19.00	79.00
9	42.00	40.00	38.00	38.00	80.00
10	42.00	50.00	38.00	28.00	89.00
11	40.00	50.00	37.00	17.00	86.00
12	47.00	50.00	40.00	0,00	46.00
13	41.00	44.00	35.00	37.00	79.00
14	43.00	44.00	40.00	38.00	87.00
15	44.00	32.00	41.00	43.00	77.00
16	46.00	37.00	34.00	42.00	88.00
17	40.00	50.00	37.00	19.00	91.00
18	42.00	38.00	34.00	35.00	87.00
19	36.00	35.00	50.00	20.00	91.00
20	42.00	44.00	37.00	45.00	76.00
21	35.00	36.00	40.00	39.00	90.00
22	42.00	50.00	40.00	19.00	80.00
23	41.00	33.00	37.00	47.00	77.00
24	40.00	33.00	39.00	40.00	88.00
25	44.00	37.00	35.00	37.00	75.00
26	37.00	48.00	35.00	38.00	86.00
27	39.00	34.00	36.00	40.00	81.00
28	49.00	44.00	38.00	22.00	72.00
29	43.00	49.00	38.00	20.00	87.00
30	41.00	37.00	33.00	38.00	79.00
31	38.00	46.00	36.00	18.00	81.00
32	44.00	49.00	37.00	20.00	84.00
33	49.00	39.00	41.00	41.00	83.00
34	47.00	39.00	37.00	38.00	86.00
35	41.00	38.00	45.00	41.00	87.00
36	37.00	37.00	41.00	35.00	91.00
37	45.00	40.00	43.00	45.00	88.00
38	46.00	37.00	36.00	37.00	84.00
39	50.00	36.00	39.00	20.00	91.00
40	41.00	37.00	47.00	41.00	75.00
41	46.00	42.00	33.00	40.00	80.00
42	39.00	36.00	34.00	41.00	94.00
43	48.00	39.00	43.00	44.00	84.00
44	46.00	44.00	49.00	19.00	79.00
45	44.00	38.00	37.00	37.00	91.00
46	45.00	35.00	37.00	39.00	87.00
47	44.00	41.00	34.00	36.00	78.00
48	40.00	39.00	50.00	16.00	77.00
49	41.00	38.00	44.00	45.00	92.00
50	38.00	48.00	46.00	18.00	88.00

EK 6 İstasyon Süreleri (Çevrim Süresi 60 sn)

İterasyon Sayısı	İstasyon 1'in toplam süresi	İstasyon 2'nin toplam süresi	İstasyon 3'ün toplam süresi	İstasyon 4'ün toplam süresi	Uygunluk Fonksiyonu Değeri
1	47.00	56.00	55.00	0,00	56.00
2	60.00	60.00	36.00	0,00	66.00
3	59.00	60.00	39.00	0,00	68.00
4	56.00	60.00	39.00	0,00	64.00
5	57.00	57.00	36.00	0,00	54.00
6	46.00	54.00	45.00	17.00	126.00
7	56.00	51.00	39.00	0,00	61.00
8	59.00	49.00	40.00	0,00	59.00
9	58.00	45.00	55.00	0,00	60.00
10	58.00	58.00	42.00	0,00	69.00
11	56.00	53.00	35.00	0,00	66.00
12	47.00	54.00	36.00	0,00	76.00
13	58.00	45.00	54.00	0,00	59.00
14	43.00	44.00	60.00	18.00	127.00
15	59.00	58.00	43.00	0,00	57.00
16	46.00	56.00	57.00	0,00	68.00
17	59.00	47.00	40.00	0,00	71.00
18	59.00	55.00	35.00	0,00	67.00
19	51.00	54.00	36.00	0,00	71.00
20	58.00	44.00	47.00	19.00	116.00
21	51.00	60.00	39.00	0,00	70.00
22	59.00	55.00	37.00	0,00	60.00
23	56.00	55.00	47.00	0,00	57.00
24	55.00	57.00	40.00	0,00	68.00
25	44.00	54.00	55.00	0,00	55.00
26	60.00	60.00	38.00	0,00	66.00
27	53.00	56.00	40.00	0,00	61.00
28	55.00	55.00	43.00	0,00	52.00
29	59.00	51.00	40.00	0,00	67.00
30	60.00	51.00	38.00	0,00	59.00
31	54.00	45.00	39.00	0,00	61.00
32	58.00	53.00	39.00	0,00	64.00
33	49.00	56.00	45.00	20.00	123.00
34	59.00	58.00	44.00	0,00	66.00
35	60.00	60.00	45.00	0,00	67.00
36	58.00	57.00	35.00	0,00	71.00
37	45.00	59.00	48.00	21.00	128.00
38	46.00	55.00	55.00	0,00	64.00
39	50.00	54.00	41.00	0,00	71.00
40	51.00	54.00	40.00	0,00	55.00
41	60.00	42.00	59.00	0,00	60.00
42	59.00	50.00	41.00	0,00	74.00
43	48.00	57.00	48.00	21.00	124.00
44	46.00	60.00	52.00	0,00	59.00
45	60.00	59.00	37.00	0,00	71.00
46	45.00	52.00	59.00	0,00	67.00
47	56.00	59.00	40.00	0,00	58.00
48	56.00	55.00	34.00	0,00	57.00
49	60.00	56.00	52.00	0,00	72.00
50	54.00	53.00	43.00	0,00	68.00

EK 7 İstasyon Süreleri (Çevrim Süresi 80 sn)

İterasyon Sayısı	İstasyon 1'in toplam süresi	İstasyon 2'nin toplam süresi	İstasyon 3'ün toplam süresi	İstasyon 4'ün toplam süresi	Uygunluk Fonksiyonu Değeri
1	80.00	78.00	0,00	0,00	36.00
2	65.00	73.00	18.00	0,00	126.00
3	79.00	79.00	0,00	0,00	48.00
4	77.00	78.00	0,00	0,00	44.00
5	80.00	70.00	0,00	0,00	34.00*
6	77.00	68.00	17.00	0,00	126.00
7	74.00	72.00	0,00	0,00	41.00
8	76.00	72.00	0,00	0,00	39.00
9	79.00	79.00	0,00	0,00	40.00
10	78.00	80.00	0,00	0,00	49.00
11	74.00	70.00	0,00	0,00	46.00
12	68.00	69.00	0,00	0,00	56.00
13	76.00	63.00	18.00	0,00	119.00
14	64.00	63.00	38.00	0,00	127.00
15	76.00	61.00	23.00	0,00	117.00
16	77.00	61.00	21.00	0,00	128.00
17	76.00	70.00	0,00	0,00	51.00
18	80.00	69.00	0,00	0,00	47.00
19	71.00	70.00	0,00	0,00	51.00
20	78.00	71.00	19.00	0,00	116.00
21	71.00	79.00	0,00	0,00	50.00
22	76.00	75.00	0,00	0,00	40.00
23	74.00	65.00	19.00	0,00	117.00
24	73.00	79.00	0,00	0,00	48.00
25	79.00	74.00	0,00	0,00	35.00
26	79.00	79.00	0,00	0,00	46.00
27	73.00	76.00	0,00	0,00	41.00
28	71.00	60.00	22.00	0,00	112.00
29	79.00	71.00	0,00	0,00	47.00
30	78.00	71.00	0,00	0,00	39.00
31	70.00	68.00	0,00	0,00	41.00
32	75.00	75.00	0,00	0,00	44.00
33	69.00	80.00	21.00	0,00	123.00
34	78.00	64.00	19.00	0,00	126.00
35	79.00	65.00	21.00	0,00	127.00
36	74.00	76.00	0,00	0,00	51.00
37	65.00	63.00	45.00	0,00	128.00
38	63.00	75.00	18.00	0,00	124.00
39	68.00	77.00	0,00	0,00	51.00
40	68.00	77.00	0,00	0,00	35.00
41	80.00	61.00	20.00	0,00	120.00
42	75.00	75.00	0,00	0,00	54.00
43	65.00	65.00	44.00	0,00	124.00
44	69.00	70.00	19.00	0,00	119.00
45	78.00	78.00	0,00	0,00	51.00
46	80.00	76.00	0,00	0,00	47.00
47	76.00	79.00	0,00	0,00	38.00
48	79.00	66.00	0,00	0,00	37.00
49	79.00	69.00	20.00	0,00	132.00
50	71.00	79.00	0,00	0,00	48.00

EK 8 İstasyon Süreleri (Çevrim Süresi 90 sn)

İterasyon Sayısı	İstasyon 1'in toplam süresi	İstasyon 2'nin toplam süresi	İstasyon 3'ün toplam süresi	İstasyon 4'ün toplam süresi	Uygunluk Fonksiyonu Değeri
1	85.00	73.00	0,00	0,00	56.00
2	86.00	70.00	0,00	0,00	66.00
3	79.00	79.00	0,00	0,00	68.00
4	77.00	78.00	0,00	0,00	64.00
5	80.00	70.00	0,00	0,00	54.00
6	77.00	85.00	0,00	0,00	66.00
7	89.00	57.00	0,00	0,00	61.00
8	76.00	72.00	0,00	0,00	59.00
9	82.00	76.00	0,00	0,00	60.00
10	78.00	80.00	0,00	0,00	69.00
11	90.00	54.00	0,00	0,00	66.00
12	83.00	54.00	0,00	0,00	76.00
13	85.00	72.00	0,00	0,00	59.00
14	87.00	78.00	0,00	0,00	67.00
15	76.00	84.00	0,00	0,00	57.00
16	83.00	76.00	0,00	0,00	68.00
17	90.00	56.00	0,00	0,00	71.00
18	80.00	69.00	0,00	0,00	67.00
19	89.00	52.00	0,00	0,00	71.00
20	86.00	82.00	0,00	0,00	56.00
21	89.00	61.00	0,00	0,00	70.00
22	76.00	75.00	0,00	0,00	60.00
23	74.00	84.00	0,00	0,00	57.00
24	73.00	79.00	0,00	0,00	68.00
25	81.00	72.00	0,00	0,00	55.00
26	85.00	73.00	0,00	0,00	66.00
27	90.00	59.00	0,00	0,00	61.00
28	88.00	65.00	0,00	0,00	52.00
29	79.00	71.00	0,00	0,00	67.00
30	78.00	71.00	0,00	0,00	59.00
31	84.00	54.00	0,00	0,00	61.00
32	75.00	75.00	0,00	0,00	64.00
33	88.00	82.00	0,00	0,00	63.00
34	86.00	75.00	0,00	0,00	66.00
35	79.00	86.00	0,00	0,00	67.00
36	90.00	60.00	0,00	0,00	71.00
37	85.00	88.00	0,00	0,00	68.00
38	83.00	73.00	0,00	0,00	64.00
39	86.00	59.00	0,00	0,00	71.00
40	89.00	56.00	0,00	0,00	55.00
41	88.00	73.00	0,00	0,00	60.00
42	75.00	75.00	0,00	0,00	74.00
43	87.00	87.00	0,00	0,00	64.00
44	90.00	68.00	0,00	0,00	59.00
45	82.00	74.00	0,00	0,00	71.00
46	80.00	76.00	0,00	0,00	67.00
47	85.00	70.00	0,00	0,00	58.00
48	79.00	66.00	0,00	0,00	57.00
49	79.00	89.00	0,00	0,00	72.00
50	86.00	64.00	0,00	0,00	68.00

ÖZGEÇMİŞ

Adı Soyadı : İsmail Enes Parlak
Doğum Yeri ve Tarihi : Sakarya / 16.11.1990
Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Şükrü Şankaya Anadolu Lisesi - 2008
Lisans : Uludağ Üniversitesi End. Müh. - 2015
Anadolu Üniversitesi İktisat - 2017
Yüksek Lisans : Uludağ Üniversitesi End. Müh. - 2018

Çalıştığı Kurum/Kurumlar ve Yıl : Bursa Teknik Üniversitesi - 2017

İletişim (e-posta) : ensparlak@gmail.com

Yayınları :

Yıldız, A., Karakoyun, F., Parlak, İ.E. 2018. The most suitable mobile RFID reader selection by using interval type-2 fuzzy topsis method, *Sigma Journal of Engineering and Natural Sciences*. (In press).