

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YÖRÜNGEYE DAYALI SLAM PROBLEMİNDE
TOPOLOJİK HARİTALAMA YÖNTEMİ**

**YÜKSEK LİSANS TEZİ
Kontrol. Müh. Müjde BAKIR**

**Anabilim Dalı: Elektrik Mühendisliği
Programı: Kontrol ve Otomasyon Mühendisliği**

Tez Danışmanı: Prof.Dr. Hakan TEMELTAŞ

EYLÜL 2008

**YÖRÜNGEYE DAYALI SLAM PROBLEMİNDE
TOPOLOJİK HARİTALAMA YÖNTEMİ**

**YÜKSEK LİSANS TEZİ
Kontrol Mühendisi Müjde BAKIR
(504051112)**

**Tezin Enstitüye Verildiği Tarih : 15 Eylül 2008
Tezin Savunulduğu Tarih : 25 Eylül 2008**

**Tez Danışmanı : Prof.Dr. Hakan TEMELTAŞ
Diğer Jüri Üyeleri Prof.Dr. Metin GÖKAŞAN
Yrd.Doç.Dr. Osman Kaan EROL**

EYLÜL 2008

ÖNSÖZ

Tez danışmanım Prof. Dr. Hakan TEMELTAŞ'a bu çalışma boyunca yapmış olduğu rehberlik ve her zaman sağlamış olduğu teşvik için teşekkür ederim.

Çalışma aşamalarında bana her türlü desteği sağlayan anne ve babama fedakarlıkları için teşekkür ederim.

Bu çalışmayı iş hayatı ile beraber sürdürmeme olanak veren ve beni hep destekleyen ABB Proses Otomasyonu bölümündeki müdürüm ve diğer proje mühendisi arkadaşlarıma da teşekkür ederim.

Eylül, 2008

Müjde BAKIR

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ	1
1.1. SLAM Tanımı	1
1.2. SLAM Tarihçesi	1
1.3. Tezin Organizasyonu	2
2. SLAM METOTLARI	3
2.1. SLAM Probleminin Formülasyonu ve Yapısı	3
2.1.1. Olasılık Teorisi ve SLAM	5
2.1.1.1. Gözlem ve Hareket Modelleri	5
2.2. SLAM Probleminin Çözüm Yöntemleri ve EKF	7
2.2.1. Genişletilmiş Kalman Filtresi	7
2.3. SLAM Metotlarında Karşılaşılan Sorunlar	9
2.3.1. Hesaplama Zorluğu	10
2.3.2. Veri İlişkilendirme	10
2.3.2.1 Toplu Doğrulama	10
2.3.2.2 Görüntü İşaretleri	11
2.3.3. Çevre Gösterimleri	11
2.3.3.1 Parçalı Gözlenebilirlik ve Gecikmeli Haritalama	11
2.3.3.2 Yörüngeye Dayalı SLAM	11
3. YÖRÜNGEYE DAYALI SLAM	13
3.1. Yörüngeye Dayalı SLAM Tanımı	13
3.2. Yörüngeye Dayalı SLAM Tarihçesi	15
3.3. Yörüngeye Dayalı SLAM Metotları	16
3.3.1. FastSLAM Yöntemi	16
3.3.2. Topolojik Haritalama Yöntemi	17
3.3.3. Seyrek-İnformasyon Yöntemi	17
4. GTTH YÖNTEMİ	19
4.1. Problem Tanımı	19
4.2. Tutarlı Veri Kaydetme Problemi	19

4.3. Poz İlişkileri	21
4.3.1. Poz İlişkilerinin Elde Edilmesi	21
4.3.2. Ağ Yapısının Oluşturulması	22
4.4. Optimal Kestirim Yönetmi	23
4.5. Poz İlişkilerinin Elde Edilmesi	25
4.5.1. Poz Birleştirme İşlemi	25
4.5.2. Tarama Eşleştirme İşleminde Elde Edilen Poz İlişkileri	25
4.5.3. Odometri Ölçümlerinden Elde Edilen Poz İlişkileri	29
4.6. Optimal Poz Kestirimi	32
4.7. Rekürsif Kestirimi	33
4.8. Algoritmanın Uygulanması	35
5. SİMÜLASYON ÇALIŞMALARI	36
5.1. Araç Yörüngesi ve Ağın Oluşturulması	36
5.2. Başlangıç Pozları ve İlk İlişkilerin Hesaplanması	38
5.2.1. Zayıf Bağlantıların Hesaplanması	40
5.2.2. Güçlü Bağlantıların Hesaplanması	42
5.3. G Matrisi ve B vektörünün Hesaplanması	45
5.4. Zayıf Bağlantılardan Oluşan Sistem İçin Sonuçlar	48
5.5. Zayıf ve Güçlü Bağlantılardan Oluşan Sistem İçin Sonuçlar	51
6. SONUÇLAR VE TARTIŞMA	55
KAYNAKLAR	56
EK A MATLAB KODLARI – ZAYIF BAĞLAR İÇİN G MATRİSİ VE B VEKTÖRÜ	58
EK B MATLAB KODLARI – GÜÇLÜ VE ZAYIF BAĞLAR İÇİN G MATRİSİ VE B VEKTÖRÜ	61
EK C MATLAB KODLARI –ZAYIF BAĞLAR İÇİN İTERASYON İŞLEMLERİ	64
EK D MATLAB KODLARI –ZAYIF BAĞLAR VE GÜÇLÜ BAĞLAR İÇİN İTERASYON İŞLEMLERİ	68
ÖZGEÇMİŞ	73

KISALTMALAR

EZLH (SLAM)	: Eş Zamanlı Lokalizasyon ve Haritalama
GKF (EKF)	: Genişletilmiş Kalman Filtresi
GTTH (GCRS)	: Global Tutarlı Alan Taraması Hizalama
LRGC	: Local Registration and Global Correlation

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1 Algoritmanın vektör tanımları.....	3
Tablo 2.2 Vektörlerin tanımladığı kümeler.....	3
Tablo 5.1 Yörünge üzerinde ölçüm yapılan pozisyonlar.....	37
Tablo 5.2 Pozisyonlar için odometri ile elde edildiği varsayılan başlangıç değerleri	39
Tablo 5.3 Güçlü bağlantılar ve ilgili karşılıklı noktalar	42
Tablo C.1 Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları	66
Tablo D.1 Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları	71

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : Robot Hareketi ve Engeller.....	4
Şekil 3.1 : Durumu ve Engel Bilgilerinin İlişkilendirilmesi.....	14
Şekil 3.2 : Engellere Dayalı Olmayan SLAM.....	15
Şekil 3.3 : Gözlemlerin Bağımsızlaştırılması.....	16
Şekil 4.1 : Ölçüm İşlemi	21
Şekil 4.2 : Ağ yapısı	22
Şekil 4.3 : Karşılıklı Noktalar	26
Şekil 4.4 : Kovaryans Kestirimi.....	30
Şekil 5.1 : Ölçüm Yapılan Pozisyonlarla Yörünge Gösterilimi	38
Şekil 5.2 : Zayıf ve Güçlü Bağlar ile Pozisyonların Gösterilimi	40
Şekil 5.3 : Bir Güçlü Bağlantıya Ait Karşılıklı Nokta Çiftinin Lokal Eksenlerdeki Yerleri.....	43
Şekil 5.4 : Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları (Pozisyonlar).....	48
Şekil 5.5 : Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları (Oryantasyonlar).....	49
Şekil 5.6 : Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması.....	50
Şekil 5.7 : Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İkinci İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması.....	50
Şekil 5.8 : Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Pozisyonlar).....	51
Şekil 5.9 : Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Oryantasyonlar).....	51
Şekil 5.10 : Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması.....	52
Şekil 5.11 : Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan İkinci İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması.....	53
Şekil 5.12 : Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan Üçüncü İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması.....	54

SEMBOL LİSTESİ

V_i, V_j	: Robot pozları
G	: Sistem matrisi
B	: Sistem vektörü
D_{ij}	: Poz ilişkisi
C_{ij}	: Poz ilişkisi hata kovaryansı
W_{ij}	: Poz ilişkisi için enerji değeri
J	: Zayıf ilişkiler için hesaplana Jakobiye

YÖRÜNGEYE DAYALI SLAM PROBLEMİNDE TOPOLOJİK HARİTALAMA YÖNTEMİ

ÖZET

Eş zamanlı haritalama ve lokalizasyon problemi, 90'lı yıllar içerisinde otonom araçlarla ilgilenen birçoğk araştırmacının odaklandığı bir konu olmuştur. Lokalizasyon ve haritalama probleminin birbiri içinde tanımlanması ve başlangıç noktasında harita ve veya robot konumları ile ilgili hiçbir bilginin bulunmaması çözümün başa çıkması gereken belirsizlikleri doğurmuştur. Bu belirsizlikler araştırmaların olasılık teorisiyle iç içe geçmesini sağlamıştır. Olasılık teorisi optimal bir kestirimi önerdiği zaman da Kalman filtresi çözümü öne çıkmıştır. Bu filtrenin lineer olmayan sistemlere uyarlanması ile Genişletilmiş Kalman Filtresi (GKF) yöntemi doğmuştur. Bu yöntemin uygulamaya geçirilmesi ile ortaya çıkan bazı problemler araştırmacıları bu yöntem üzerinde daha çok çalışmaya veya alternatif çözümler üretmeye yöneltmiştir. FastSLAM olarak bilinen parçacık filtresi çözümü SLAM problemi için opsiyon bir çözüm olarak ortaya çıkmıştır FastSLAM uygulamalarda algortimaya hız katmıştır ancak karşılaşılan bazı problemler giderilememiştir.

Mobil robotların ortamda engelleri gözleyerek ve bu gözlemlere dayanarak konumlarını belirlediği yöntemlerden sonra bazı araştırmacılar haritanın bir engeller haritası olarak tanımlanmasının zorunlu olup olmadığı sorularını sormuşlardır. Engel tanımlamanın zor olduğu bir çevrede haritanın nasıl çıkarılacağı konusu değişik bir fikrin ortaya çıkmasını sağlamıştır. Bu fikir, temel anlamda robotun yörüngesinin yani izlemiş olduğu yolun da aslında bir harita olduğu düşüncesini savunmaktadır.

Yörüngeye dayalı SLAM metotları genel olarak üç başlıkta toplanmıştır. Birincisi FastSLAM olarak düşünülmüş ama bu yöntemde bütünsel bir yörüngeden bahsedilememektedir. İkinci olarak topolojik yöntemleri temel alan metotlar yer almaktadır. Bu tezin çıkış noktası olan GTTH metodu da (global alan taramalarının tutarlı birleştirilmesi) bu metotlar içinde sayılmaktadır. Bu başlık altında sayılabilecek üçüncü metot algoritmada kovaryanslarla çalışmak yerine informasyon formu ile çalışmayı tercih eden sparse-information formu metotlardır.

Bu tezde yapılan ana çalışma GTTH metodunun anlaşılması, bu metoda göre bir ağ yapısının oluşturulması ve bir mobil robotun çizdiği bir yörüngede başa dönerek tamamladığı tur boyunca tanımladığı pozları doğru olarak kestirip kestiremediğinin simüle edilmesidir. Bu amaçla GTTH metodunun performansı iki farklı simülasyon yapılarak test edilmiştir. Yakınsama sonuçları simülasyon bölümünde ortaya konmuştur.

TOPOLOGICAL METHOD IN TRAJECTORY-ORIENTED SLAM

SUMMARY

The Simultaneous Localization and Mapping Problem has been an issue since the 90's for all the researchers who focus on the autonomous vehicles. The coupling of the two problems namely the localization and the mapping problems forced the start point of the problem with no a priori information at all. Hence, for a convergent solution too many uncertainties to be handled have arised. These uncertainties caused the research to approach the problem in a probabilistic way. The probabilistic theory suggested the optimal estimation and as a result the Kalman Filter was the first address for the solution. Advancing the KF to the nonlinear systems resulted in the EKF (Extended Kalman Filter). Problems came along while implementing the EKF method in the real time applications. The continued work on this area had beneficiant results, one of them was the FastSLAM algorithm. It has accelerated the real time implementation but some issues still remained unsolved.

After the SLAM methods which depend on the features in the environment to build the map and locate the robot according to the these features, some researchers wondered about the must of creating a map of features. In an environment which consists of landmarks which can not be distinguished from each other easily resulted in the idea which defined the trajectory of the robot as the map. This brought a new aspect to the SLAM solutions .

Generally, the trajectory-oriented SLAM methods can be gathered under three main areas. The first can be considered as the FastSLAM but it should not be forgotten that this method can not offer a whole trajectory. Secondly, methods that are based on the topological mapping belong under that header. The method which is handled in this master thesis ,namely the GCSA, method also is also one of these solutions. The third idea depends on an algorithm which prefers the information form rather than the covarinace form, and is the sparse-information form.

The goal of this thesis is to analyze the GCSA method, to build a network according to this method and solve the estimations of the robot poses for a robot completing its trajectory at the point where it has started it. To reveal the performance of this method two diffrent simulation scenarios were undertaken. At the end of this thesis the results of the simulations are demonstrated.

1. GİRİŞ

1.1. SLAM Tanımı

SLAM, eşzamanlı yer tayini ve haritalama problemidir. SLAM, bir mobil robotun bilinmeyen bir çevrede bu çevrenin bilinmeyen bir lokasyonunda iken harita çıkarması ve aynı zamanda bu haritada yerini belirleme problemine çözüm arar^[1].

SLAM problemine önerilecek bir çözümün, robotun harita bilgisine daha önceden sahip olmadığı bir çevrede bulunduğunu ve dışarıdan pozisyon bilgisi alamadığını göz önünde bulundurması gerekir. Böyle bir çözüm sadece robotlar için değil otonom bütün araçlarda uygulama alanı bulacaktır. Ayrıca, bu çözümün bulunması mobil bir robotun tam anlamıyla otonom bir araç halini almasını sağlar.

1.2. SLAM Tarihçesi

SLAM, teorik anlamda farklı matematiksel eşitliklerle tanımlanmış ve çözülmüş bir problem olarak kabul edilebilir. Uygulama alanları olarak kapalı ve açık mekanlar ile su altı, yer altı, canlıların girmesi tehlikeli olabilecek mekanlar sayılabilir. Teorik ve kavramsal anlamda SLAM'ın çözümü bulunmuş bir problem olduğu kabul edilmiştir. Ancak pratik uygulamalara geçilmesiyle başa çıkılması gereken bazı problemler olduğu anlaşılmıştır.

Olasılık yöntemli SLAM probleminin doğuşu 1986 yılında San Fransisco, California'da toplanan IEEE Robotics and Automation Conference'nda olmuştur^[1]. Bu konferans olasılık yöntemlerinin robotik ve yapay zeka konularında kullanılmaya başlanmasının ortaya çıktığı dönemlere denk gelir. Birçok araştırmacı kestirim metotlarını haritalama ve lokalizasyon yöntemlerine uygulamaya başlamıştır. Bu araştırmacıların başlıcaları, Peter Cheeseman, Jim Crowley, Hugh Durrant-White, Raja Chatila, Oliver Faugeras, ve Randal Smith olarak sayılmaktadır.

Haritalama ve lokalizasyon problemlerinin tek bir kestirim problemi olarak tanımlandığı zaman çözümün yakınsadığının farkına varılması SLAM için kavramsal anlamda bir çığır açmıştır. Aynı zamanda, engeller arası korelasyonların artmasının

özme olumlu katkıda bulunmasının ortaya ıkması da büyük bir adım olmuştur. SLAM probleminin yapısı, yakınsama sonucu ve 'SLAM' kısaltmasının ortaya konması 1995 yılında robotik araştırmaları üzerine olan uluslararası bir sempozyumda sunulan bir araştırma makalesinde ilk olarak yapılmıştır.

1.3. Tezin Organizasyonu

Bu tezin odaklanmış olduđu ana konu GTTH (Global Tutarlı Alan Taraması Hizalama) metodudur. Bu metodun anlaşılması öncesinde ilk olarak SLAM problemi genel bir çerçevede anlatılmalıdır. SLAM problemi için önerilmiş standart çözümler ve bu çözümlerle beraber gelen problemler gözden geçirilmelidir. Bunun arkasından yörüngeye dayalı SLAM tanımı yapıp başlıca metotları adlandırılmalıdır. Son olarak yörüngeye dayalı SLAM meetotlarından biri olan GTTH algoritması anlaşılmalı ve farklı durumlar içinde test edilip kendi içinde kıyaslanmalıdır.

2. SLAM METOTLARI

2.1. SLAM Probleminin Formülasyonu ve Yapısı:

SLAM, mobil bir robotun eş zamanlı olarak çevresinin haritasını çıkartıp bu haritayı konumunu tespit etmede kullanması sürecidir. SLAM algoritmalarında robot platformunun yörüngesi ve engellerin konumlarının yer bilgilerine önceden sahip olunmadan robot hareketi esnasında kestirilmesi amaçlanır.

Bilinmeyen bir çevrede, üstünde sensor bulunan ve bilinmeyen engelleri gözlemleyen mobil bir robot düşünelim. Bir k anı için aşağıdaki bilgiler Tablo 2.1'deki vektörler tanımlanabilir ^[1].

Tablo 2.1: Algoritmanın Vektör Tanımları

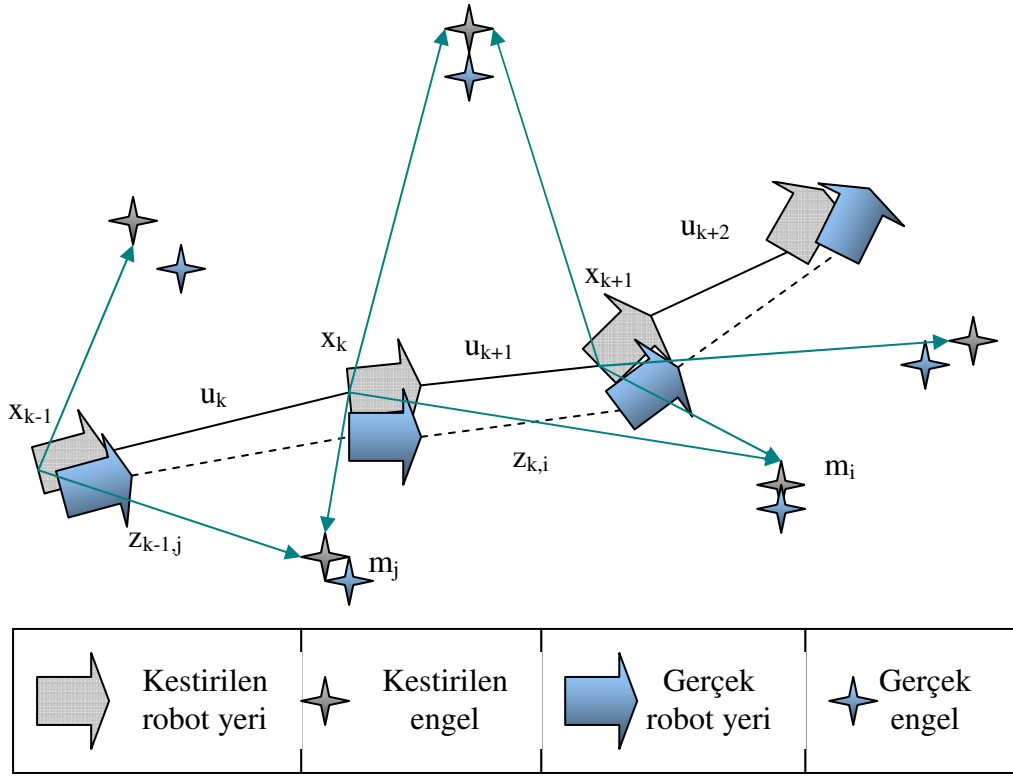
Vektörler	Açıklaması
$\mathbf{x}_k$	Aracın yerini ve oryantasyonun gösteren durum vektörü
$\mathbf{u}_k$	aracı $\mathbf{x}_k$ durumuna getirmek için $k-1$ anında uygulanmış kontrol vektörü
$\mathbf{m}_i$	k anındaki i . engeli tanımlayan vektör
$\mathbf{z}_{ik}$	i . engeli için aracın k anında yaptığı gözlem

Ayrıca bu vektörlerin oluşturmuş olduğu ve algoritmanın bütününde anlamlı olan Tablo 2.2'deki bilgi kümeleri tanımlanır.

Tablo 2.2: Vektörlerin Tanımladığı Kümeler

Gösterilim	İçerdiği Bilgiler	Açıklama
$\mathbf{X}_{0:k}$	$\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$	Geçmiş araç konumları
$\mathbf{U}_{0:k}$	$\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$	Uygulanmış olan kontrol girişleri
$\mathbf{M}$	$\{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$	Engeller kümesi
$\mathbf{Z}_{0:k}$	$\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$	Geçmiş engel gözlemleri

Tanımlı bu vektörlerin anlaşılması için Şekil 2.1'i incelemek yeterlidir. Konumlar, uygulanan kontrol girişleri, gözlemler ve engeller hem gerçek olarak hem de robotun algılamış olduğu biçimde , yani belli bir algılama hatasıyla, gösterilmiştir.



Şekil 2.1: Robot Hareketi ve Engeller

SLAM problemi ikiye ayrılarak daha iyi anlaşılabilir. İlk olarak lokalizasyon probleminin tanımı yapılmalıdır. Bu problemin tanımında m haritasının önceden bilindiği varsayılır. Uygulanan U^k kontrol işaretleri girişleri ile aracın bilinmeyen X^k pozisyonlarına gitmesi sağlanır. Amaç bu X^k pozisyonlarının kestirimlerini doğru bir şekilde yapabilmektir.

İkinci problem olan haritalamaya ise X^k araç konumları önceden bilinmektedir. Haritayı oluşturmak için ilgili kestirimler yapılır.

Eşzamanlı lokalizasyon ve haritalama probleminin tanımı ise bu problemlere göre daha kapsamlıdır. Bu problemin ilk anda elinde bulunan verileri arasında harita hakkında hiçbir bilgi yoktur. Başlangıç konumu x_0 'ın bilindiği varsayılır ve bu konum orijin olarak kabul edilir. Uygulanan U_k kontrol girişi işaretleri bilinmektedir. Harita oluşturulur ve aynı zamanda robotun X_k konumları kestirilmeye çalışılır. SLAM, lokalizasyon ve haritalama problemlerinin bir araya getirilmesi ile oluşturulur.

2.1.1 Olasılık Teorisi ve SLAM

Belirsizlik her kestirim probleminin doğal bir parçasıdır. Belirsizliği matematiksel olarak ifade etmek için olasılıksal eşitlikler kullanılır. Sensörler, yani ölçüm, $P(z_k / x_k, m)$ olasılığı ile ifade edilir. Araç hareketi de, $P(x_k / x_{k-1}, u_k)$ koşullu olasılığı ile tanımlanabilir. Bu iki ifadenin birleştirilmesi ile bileşik posterior $P(x_k, m / Z^k, U^k, x_0)$ ortaya çıkar ve bu olasılık ardışıl adımlarda kestirilmeye çalışılır.

2.1.1.1 Gözlem ve Hareket Modelleri:

Gözlem modeli, konum ve harita bilgisi ışığında $\{x_k, m\}$, z_k gözlemini yapma olasılığını hesaplamada kullanılır. Aşağıdaki gibi tanımlanır:

$$P(z_k / x_k, m) \quad (2.1)$$

Gözlemlerin birbirinden bağımsız olduğu varsayımı ile aşağıdaki denklem yazılabilir:

$$P(Z^k / X^k, m) = \prod P(z_i / x_i, m) \quad (2.2)$$

Yeni gözlemler yapıldıkça elde edilen bilgilerin güncellenmesi gerekir. Bu sebeple Bayes teoreminden yola çıkılarak Bayes adımı oluşturulur. Bu adımda birleşik dağılım fonksiyonu durum terimleri için genişletilir:

$$P(x_k, m, z_k / Z^{k-1}, U^k, x_0) = P(x_k, m / z_k, Z^{k-1}, U^k, x_0) P(z_k / Z^{k-1}, U^k, x_0) \quad (2.3.a)$$

$$= P(x_k, m / Z^k, U^k, x_0) P(z_k / Z^{k-1}, U^k) \quad (2.3.b)$$

Aynı şekilde birleşik dağılım fonksiyonunu gözlem terimleri için de genişletilir:

$$P(x_k, m, z_k / Z^{k-1}, U^k, x_0) = P(z_k / x_k, m, Z^{k-1}, U^k, x_0) P(x_k, m / Z^{k-1}, U^k, x_0) \quad (2.4.a)$$

$$= P(z_k / x_k, m) P(x_k / Z^{k-1}, U^k, x_0) \quad (2.4.b)$$

Bu denklemler yeniden ayarlandığında elde edilen eşitlik:

$$P(x_k, m / Z^k, U^k, x_0) = P(z_k / x_k, m) * P(x_k, m / Z^{k-1}, U^k, x_0) / P(z_k / Z^{k-1}, U^k) \quad (2.5)$$

Robotun hareket halinde olduğu düşünülürse zamana göre robotun durumunun da değiştiği ve dolayısıyla güncellenmesi gerektiği ortaya çıkar. Araç modeli Markov olarak kabul edilirse:

$$P(x_k / x_{k-1}, u_k) = P(x_k / x_{k-1}, u_k, X^{k-2}, U^{k-1}, m) \quad (2.6)$$

Toplam Olasılık Teoreminden:

$$P(x_k, m / Z^{k-1}, U^k, x_0) = \int P(x_k, x_{k-1}, m / Z^{k-1}, U^k, x_0) dx_{k-1} \quad (2.7.a)$$

$$= \int P(x_k / x_{k-1}, m, Z^{k-1}, U^k, x_0) P(x_{k-1}, m / Z^{k-1}, U^k, x_0) dx_{k-1} \quad (2.7.b)$$

$$= \int P(x_k / x_{k-1}, u_k) P(x_{k-1}, m / Z^{k-1}, U^{k-1}, x_0) dx_{k-1} \quad (2.7.c)$$

Toplam Rekürsif Hesaplama:

$$P(x_k, m / Z^k, U^k, x_0) = K.P(z_k / x_k, m) \int P(x_k / x_{k-1}, u_k) P(x_{k-1}, m / Z^{k-1}, U^{k-1}, x_0) dx_{k-1} \quad (2.8)$$

Olasılık dağılımları ile tanımlanan SLAM probleminin aşağıdaki dağılımının her k anı için hesaplanması gerekir ^[1]:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (2.9)$$

Bu olasılık dağılımı; engel konumları ve araç durumu ile kaydedilmiş gözlemler, k zamanına kadar uygulanmış kontrol girişleri ve aracın ilk pozisyon bilgisi verildiğindeki bileşik posterior yoğunluk fonksiyonu olarak açıklanabilir. SLAM problemi için ardışıl ve rekürsif bir çözüm aranır. $P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1})$ dağılımı için $k-1$ anında bir kestirim yapılır, u_k kontrol girişi uygulandıktan sonra Bayes teoremine göre z_k gözlemi hesaplanır. Bu hesaplama için durum geçiş modeli ve gözlem modeli tanımlanmalıdır. Durum geçiş modeli kontrol girişine, gözlem modeli de gözleme göre tanımlanmalıdır.

Gözlem modeli araç konumu ve engellerin konumu bilindiği zaman z_k gözlemini yapmanın olasılığını verir ve aşağıdaki gibi formüle edilebilir:

$$P(z_k | x_k, m) \quad (2.10)$$

Araç için hareket modeli durum geçişleri arasındaki olasılık fonksiyonu ile tanımlanabilir:

$$P(x_k | x_{k-1}, u_k) \quad (2.11)$$

Zaman güncellemesi aşağıdaki gibi yapılır:

$$P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) \times P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1} \quad (2.12)$$

Ölçüm güncellemesi de aşağıdaki eşitliğe bağlıdır:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (2.13)$$

Yukarıdaki denklemler $P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$ bileşik posterior'u, robotun durumu x_k ve harita m 'yi k zamanında tüm gözlemler $Z_{0:k}$ ve tüm kontrol girişleri $U_{0:k}$ bilgisi ile hesaplamaya yarar. Rekürsif işlemler araç modeli $P(x_k | x_{k-1}, u_k)$ ve gözlem modelinin $P(z_k | x_k, m)$ bir fonksiyonudur.

2.2. SLAM Probleminin Çözüm Yöntemleri ve GKF

Olasılık yöntemli SLAM çözümleri gözlem ve hareket modeli için uygun bir ifade ortaya koymayı amaçlar. Bu modeller zaman ve ölçüm güncellemelerini hesaplamak için etkili ve uygun olmalıdır. En genel ifadelerden biri durum uzayı modeline Gaussian gürültüsü eklemek ve genişletilmiş Kalman filtresini (GKF) kullanmaktır. Alternatif bir yöntem de araç hareket modelini daha genel, Gaussian olmayan bir olasılık dağılımına sahip örnek kümesi ile ifade etmektir. Bu işlem Rao-Blackwellized parçacık filtresi veya FastSLAM algoritmasına denk düşer.

2.2.1. Genişletilmiş Kalman Filtresi

Kalman filtresi, SLAM uygulamalarında kullanılan ilk yöntemdir. Lineer Kalman filtresi Bayes kuralını kullanır ^[9]. GKF lineer olmayan modellerde kullanılır ve en iyi kestirim etrafında lineerleştirme yapılır. Bu işlemde Jakobiyen matrisi kullanılır ve problem tekrar lineer bir duruma getirilir.

Genişletilmiş Kalman filtresi, lineer olmayan proses ve gözlem modelleri ile betimlenmiş sistemler için kullanılan lineer rekürsif bir kestirim yöntemidir.

GKF, yakın zamana kadar lokalizasyon, haritalama ve navigasyon problemlerinde en çok kullanılan algoritmalarından biri olmuştur. Bugünkü birçok SLAM algoritmasının temelini oluşturmuştur.

GKF, araç hareket ve gözlem modelleri için analitik modeller kullanır. Hareket ve gözlem hatalarının dağılımlarının unimodal ve sıfır ortalamalı olduğunu varsayar.

GKF yönteminde kullanılan modelleri daha yakından inceleyelim. Araç hareketini betimleyen denklem ^[1]:

$$P(x_k | x_{k-1}, u_k) \Rightarrow x_k = f(x_{k-1}, u_k) + w_k \quad (2.14)$$

Bu denklemde geçen f fonksiyonu aracın kinematik modeli, w_k ise sıfır ortalamalı, Gaussian özelliğine ve Q_k kovaryansına sahip bir bozucu olarak düşünülür.

EKF yönetiminde kullanılan gözlem modeli de ^[1]

$$P(z_k | x_k, m) \Rightarrow z_k = h(x_k, m) + v_k \quad (2.15)$$

olarak tanımlanır. Bu eşitlikte yer alan h fonksiyonu gözlem yapmanın geometrisini ifade eder. v_k ise bir önceki denklemde yer alan w_k gibi sıfır ortalamalı, Gaussian özelliğine ve R_k kovaryansına sahip bir bozucu olarak düşünülür.

Standart GKF metodunda ortalama değer ve kovaryans hesaplandıktan sonra bileşik posterior dağılımı $P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$ yazılacak olursa kullanılacak zaman güncellemesi ve gözlem güncellemesi aşağıdaki eşliklerle ifade edilebilir:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \quad (2.16)$$

$$P_{xx,k|k-1} = \nabla f * P_{xx,k-1|k-1} * \nabla f^T + Q_k \quad (2.17)$$

Burada ∇f en son kestirim için hesaplanmış f fonksiyonunun jakobiyenidir.

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{m}_{k-1} \end{bmatrix} + W_k [z_k - h(\hat{x}_{k|k-1}, \hat{m}_{k-1})] \quad (2.18)$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (2.19)$$

Burada

$$S_k = \nabla h * P_{k|k-1} * \nabla h^T + R_k \quad (2.20)$$

$$W_k = P_{k|k-1} * \nabla h^T + S_k^{-1} \quad (2.21)$$

olarak tanımlanır. Bir önceki denklemdeki düşünce ile benzer bir şekilde ∇h , h fonksiyonunun $\hat{x}_{k|k-1}$ ve $\hat{m}_{k-1}$ değerlerinde hesaplanmış jakobiyenidir. Bu çözüm standart bir GKF çözümü olup bir çok SLAM problemine uygulanmıştır.

2.3. SLAM Çözüm Metotlarında Karşılaşılan Sorunlar

SLAM problemine teorik anlamda çözümler getirildikten sonra uygulamaya geçilmiştir. Gerçek zamanlı uygulamalarda iki büyük problemle karşı karşıya kalınmıştır ^[2]. Bunlar hesaplamadaki zorluk ve veri ilişkilendirme problemleridir. Ayrıca robotun dolaştığı çevreden aldığı bilgileri sadece basit engel tanımlamaları oluşturmada kullanmasından daha fazlasını öneren çevre gösterimi yöntemleri ^[2] de uygulamalarda çeşitli kolaylıklar sağlaması açısından önemlidir.

SLAM probleminin çapı, haritasında bulunan engel sayısının karesiyle orantılı olarak artar. Gerçek zamanlı uygulamalarda bu hızlı büyüme SLAM metotlarının çevrimiçi kullanımını büyük ölçüde kısıtlar. Bu zorluğu azaltmak için üzerinde halen çalışılmakta olan metotlardan söz edilebilir. Bu metotların başlıcaları ^[2] lineer zamanlı durum düzenlemesi, informasyon formunun kullanılması, güncelleme işlemlerinin parçalara bölünmesi ve alt haritalama metotlarıdır. SLAM uygulamalarında karşılaşılan ikinci büyük problem, yeni yapılan gözlemlerin, haritada tutulan engeller ile doğru ilişkilendirilmesi problemidir. Veri ilişkilendirmede yapılan hatalar SLAM algoritmasının toplamda ıraksamasına ve gerçek harita ve konumlar için çok yanlış kestirimler yapılmasına bile sebebiyet verebilir.

Veri ilişkilendirmenin önemi özellikle araç uzun bir turdan sonra daha önce haritaladığı bir alana döndüğü vakit ,yani bir tam turunu tamamladığı zaman, daha önce görmüş olduğu engellerin haritasında var olup olmadığı kararını doğru verebilmesi açısından önemlidir. Veri ilişkilendirme probleminin çözümüne yönelik geliştirilmiş yöntemlerin başlıcaları toplu doğrulama ve görüntü verilerini kullanma olarak sayılabilir. SLAM için yapılan araştırmaların içinde üçüncü bir başlık altında toplanabilecek gelişmeler de engellerde ve haritalarda daha zengin görsel algılamının kullanmasına yönelik olan yöntemlerdir. Bu yöntemler veri ilişkilendirme ve tur-tamamlama probleminden doğmuş olup standart SLAM tanımını değiştirip engel kestirimi yerine yörünge kestirimini önermektedir.

Bu metotlar içinde geciktirilmiş haritalama ve yörünge kestirimine dayalı metotlar sayılabilir.

2.3.1 Hesaplama Zorluğu

Duruma dayalı SLAM formülasyonunda, robot pozlarının ve gözlenen engellerin konumlarının oluşturduğu bileşik bir durum tanımlıdır. Bu yapının önemli bir sonucu proses modelinin sadece araç poz durumlarını ve gözlem modelinin sadece tek bir araç-engel çiftini etkilemesidir. Hesap zorluğunu azaltmayı amaç edinen çözümlerde, SLAM probleminin iki denklemi (zaman ve gözlem güncellenme denklemleri) işlem yükünü azaltacak şekilde yeniden tanımlanır. Zaman güncelleme hesaplamasını kolaylaştırmak için geliştirilmiş metotlar durum arttırması metotlarıdır. Gözlem güncelleme hesaplamasının limitlenmesi için de güncelleme denklemlerinin parçalı formu kullanılır. SLAM optimal kestirimi için farklı formülasyonlar sunan iki yöntemde de işlem sayısını azaltmayı başarır. Standart SLAM formüllerini bilgi formuna getirmek ve bilgi matrisini elde etmek de işlem yükünü azaltır. Alt haritalama yöntemlerinde global harita lokal koordinat sistemleri içeren lokal alanlara bölünüp ve bu koordinat sistemleri içeren alanlar global harita içinde hiyerarşik bir şekilde düzenlenir. Güncellemeler lokal eksenlerde yapıp global haritaya eklenir.

2.3.2 Veri İlişkilendirme:

Veri ilişkilendirme SLAM uygulamalarında her zaman kritik bir konu olmuştur.

Haritaya yeni bilgi eklemekten önce yapılan yeni ölçüm sonuçları, haritada var olan engellerle ilişkilendirilir. Yeni ölçüm sonuçları haritaya eklendikten sonra bu ilişkilendirme bir daha yeniden gözden geçirilemez.

Buradaki problem, tek bir yanlış ilişkilendirmenin bile harita kestiriminde ıraksamaya sebep olabilmesidir.

2.3.2.1 Toplu Doğrulama:

İlk SLAM uygulamalarında her engel ölçümünün ilişkilendirilmesi, gözlenen engelin kestirilen konuma yakınlığının test edilmesi ile yapılıyordu. Eğer araç pozunu kesinlikten çok uzaksa bireysel ilişkilendirme güvenilir bir metot olmaktan çıkar. Bu

işlem toplu, yapıldığında birden fazla ilişkilendirme aynı anda yapılmış olur. Böylece engeller arası geometrik ilişkiler ortak ilişkilendirmede kullanılmış olur.

2.3.2.2 Görüntü İşaretleri:

Farklı algılama yöntemleri; şekil, renk ve desen ile ilgili yapılan ölçümleri sonuçları yararlı bilgiler sağlayabilir. Bu bilgiler iki bilgi kümesini örneğin geometrik yapısı birbiri ile neredeyse aynı olan bir engeli renk ölçümü yaparak birbirinden ayırmada kullanılabilir. SLAM için bu yöntem özellikle tur tamamlama probleminin çözülmesinde faydalı olur.

2.3.3 Çevre Gösterimleri:

SLAM üzerine yapılan ilk çalışmalarda robotun dolaştığı çevrenin basit engeller ile modellendiğini görülür. Bu engeller; noktalar, çizgiler ve yuvarlaklarla ile oluşturulmuş basit geometrik şekillerdir. Daha karmaşık ve düzgün bir yapısalığa sahip olmayan ortamlarda bu kabul geçerli olamayabilir.

2.3.3.1 Parçalı Gözlenebilirlik ve Gecikmeli Haritalama

Bir engel yeri için yapılan tek bir ölçüm gaussian dağılım elde etmeyi sağlamaz. Bir nokta etrafında gaussian dağılımdan söz edebilmek için çoklu ölçüm yapmak şarttır. Bu sebeple ölçüm bilgisini arttırmak daha sağlıklı bir kestirim sağlar ve ölçüm süresinin uzaması bir gecikmeye yol açar. Haritadaki engel gecikmeli olarak tanımlanır. Zaten genel anlamda bir yapının robustluğunu sağlamak için mümkün olduğunca bilgiyi kümelemek ve karar verme sürecini bekletmek her zaman çözüm olarak düşünülebilir.

2.3.3.2 Yörüngeye Dayalı SLAM

Standart SLAM formülasyonunda durum vektörü kestirilen son araç pozu ve gözlenen engeller ile biraraya getirilmesi ile oluşturulur:

$$x_k = [x_{vk}^T, m^T]^T \quad (2.22)$$

SLAM problemi için geliştirilmiş alternatif bir yöntemde engeller bir kenara bırakılarak araç yörüngesi kestirimi yapılır ve durum vektörünü geçmiş ve şimdiki araç pozları oluşturur:

$$\mathbf{x}_k = [\mathbf{x}_{vk}^T, \mathbf{x}_{vk-1}^T, \dots, \mathbf{x}_1^T]^T \quad (2.23)$$

Bu düşünce özellikle birbirinden kolayca ayırt edilemeyen, ayrık engellerin tanımlanamadığı ortamlarda daha güvenilir sonuçlar verir. Harita artık kestirimin bir parçası değildir. Hafızada tutulan keyfi bir bilgi kümesidir. SLAM'ın bu formulasyonunda tanımlı ayrı bir harita bulunmaz. Her poz kestirimi bir tarama bilgisine sahiptir ve bu bilgiler global bir haritaya hizalanarak birleştirilir.

3. YÖRÜNGEYE DAYALI SLAM

3.1. Yörüngeye Dayalı SLAM Tanımı:

Standart SLAM algoritması kestirimi yapılan durum vektörünü, araç pozu ve gözlenen engeller ile oluşturur.

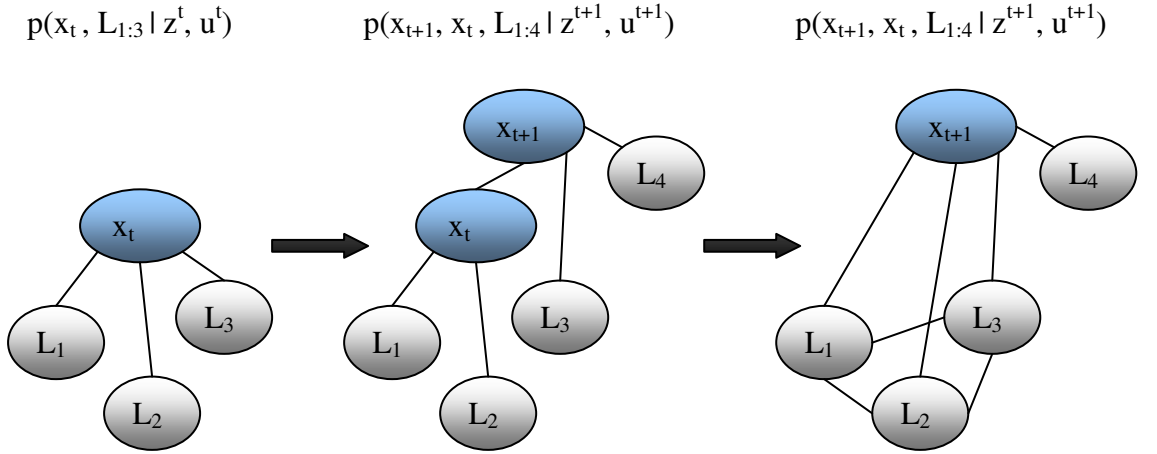
$$x_k = [x_{v_k}^T, m^T]^T \quad (3.1)$$

SLAM problemi için üretilmiş alternatif bir çözüm ise engel bilgilerini durum vektörüne eklememek bunun yerine geçmiş poz bilgilerini kullanmaktır. Bu durumda elde edilen durum vektörü geçmiş pozlardan oluşmaktadır:

$$x_k = [x_{v_k}^T, x_{v_{k-1}}^T, \dots, x_{v_1}^T]^T \quad (3.2)$$

SLAM yaklaşımlarının büyük bir çoğunluğu engellere dayalıdır. Bu yaklaşımlarda robotun çevresinde gördüğü engelleri sensör verisi yardımı ile tanımlaması ve aynı engellerin tekrar gözlenmesi halinde konumunu saptamada kullanması yöntemi kullanılır. Buradan yola çıkılarak engellerin oluşturduğu bir haritanın elde edilmesi beklenir. Lokalizasyon ve engellerden harita çıkarma işlemi iç içe geçmiştir ve bu sebeple robotun engeller ve son pozu için bileşik bir dağılım tanımlaması gerekir. İç içe geçmiş bu hataları GKF kullanarak oluşturmak için kovaryans matrisinin çok fazla çapraz ilişkiyi içermesi gerekmektedir. Harita ve robot için bileşik ilişkileri güncellemek her adımda $O(n^2)$ karmaşıklığına sebep olur. Buradaki n sayısı haritanın içerdiği engel sayısıdır.

Engellere dayalı SLAM'da oluşan karmaşıklığı anlamak için Şekil 3.1'i inceleyelim [3].



Şekil 3.1: Durum ve Engel Bilgilerinin İlişkilendirilmesi

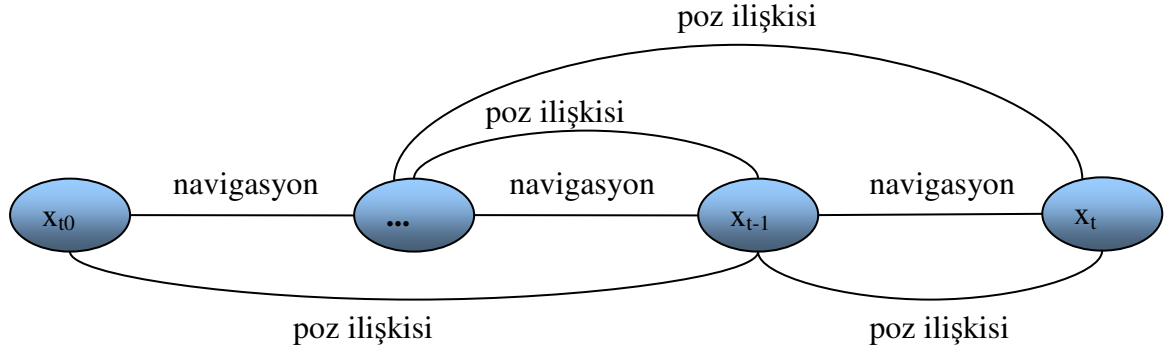
En solda yer alan şekil robotun t anında x_t pozisyonunda bulunduğu ve üç engeli gözlemlediği durumu gösterir. Bu durum için yazılacak dağılım $p(x_t, L_{1:3} | z^t, u^t)$ olarak tanımlanır ve sadece robot pozu ile engeller arası ilişkileri içerip engeller arası ilişkiler henüz tanımlanmamıştır.

Ortada yer alan durum için tanımlanan dağılım $p(x_{t+1}, x_t, L_{1:4} | z^{t+1}, u^{t+1})$ şeklindedir. Bu dağılım durum vektörüne $t+1$ anındaki x_{t+1} pozunun eklendiği; L_3 engelini tekrar gözlemlendiği ve yeni bir engel olan L_4 'ün gözlemlendiği anı tasvir etmektedir. Bu andaki ilişkileri inceleyecek olursak x_{t+1} sadece x_t pozuna bağlantılı olup L_3 ve L_4 'ün de bu poz için birer kısıt tanımladığını görürüz.

Engellere dayalı tipik bir SLAM yaklaşımında sadece son robot pozu için kestirim yapıp daha önceki pozlar durum vektöründen elimine edilir. Bu sebeple x_t robot pozunun durum vektöründen çıkarılması gerekir. Sonuç en sağdaki şekilde gösterilmiştir. Daha önceden x_t ile ilişkileri doğrudan tanımlı olan L_1, L_2, L_3 engelleri artık kendi aralarında bağlantı tanımlamak zorundadır. Bu yüzden engellere dayalı bir SLAM algoritmasının sadece son robot pozunu durum vektörü içinde bulundurmakla ödeyeceği bedel bağlantı karmaşıklığını her adımda arttırmaktır.

Engellere dayalı olmayan SLAM algoritmalarında poz ilişkileri kısıt olarak kullanılıp global pozlar için kestirim yapılır. Bu global pozlar robotun hareket yörüngesindeki ölçüm veya örnekleme yapılan noktalardır. Bu yöntemi anlamak için Şekil 3.2 incelenebilir. Bu şekilde tüm robot pozları ve aralarında tanımlanacak poz ilişkileri gösterilmiştir. Engellerden bahsedilmemektedir. Engellere dayalı SLAM

algoritmalarında olduğu gibi son robot pozu ve hem bu pozla ile hem de kendi aralarında ilişkilendirilen engeller takip edilmeyip sadece robotun ölçüm yapmış olduğu tüm geçmiş robot pozları durum vektöründe yer alır.



Şekil 3.2: Engeller Dayalı Olmayan SLAM

Bu yöntemin uygulanacağı başlıca durum, gözlenecek engellerin kolay olarak birbirinden ayırt edilemediği bir çevrede navigasyon sağlamaktır. Engeller ile ilgili elde edilen veriler durumun bir parçası olmayıp yardımcı bir bilgi kümesini oluşturur. Bu yöntemde harita sadece robotun geçtiği noktalardaki pozlarından oluşur, engeller bir harita tanımlamaz. Her robot pozunda veri yapılan bir alan taraması veya kamera ölçümleri mevcuttur.

3.2. Yörüngeye Dayalı SLAM Tarihçesi:

Yörüngeye dayalı SLAM metotları sadece en son robot pozunu değil bütün hareket yörüngesi boyunca geçilen robot pozlarını içerir. SLAM yöntemleri ile uğraşan birçok insan robot yörüngesi ile çalışmıştır ^[4] (Chatila ve Laumond 1985; Lu ve Milios 1997; Gutmann ve Konolige 2000; Konolige 2004; Eustice et al.2005) ^[4]. Bu başlık altında yapılan çalışmalarda elde edilen yöntemlerde birbirine yakın robot pozları için karşılıklı kısıt tanımlayabilecek alan taraması yapan lazer sensörleri kullanılmıştır. Genel olarak bu tip bir SLAM problemi tüm sensör pozlarını çevreden elde edilen bilgiler yardımı ile optimal kestirmek üzerine kuruludur. Bu problemin uzun bir araştırma geçmişi vardır ^[4]: (Golub ve Plemmons 1980), fotogrametri (photogrammetry), (Brown 1976; Granshaw 1980; Slama 1980; Cooper ve Robson 1996), buket ayarlaması ‘bundle adjustmet’ ve bilgisayar vizyonu, (Faugeras 1993; Szeliski ve Kang 1993, 1994; Triggs et al.2000; Hartley ve Zisserman 2000)

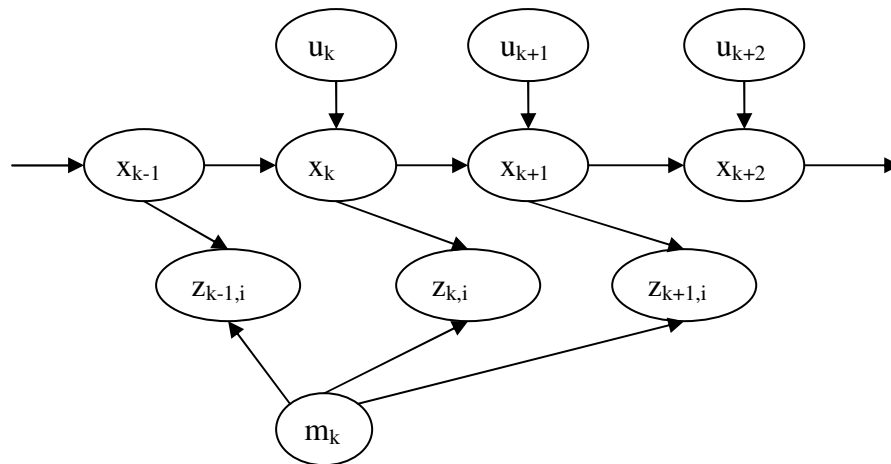
hareketin yapısı ‘structure of motion’, bu arařtırmaların bařlıcalarıdır. Son yıllar ierisinde de SLAM anlamında yeni düşünceler ortaya koyan alıřmalar yapılmıřtır (Deans ve Hebert 2000; Duckett et al.2000, 2002; Howard et al. 2001; Frese ve Duckett 2003; Freese et al. 2005; Folkesson ve Christensen 2004; Folkesson et al. 2005; Freese 2004, 2005; Thrun et al 2005).

3.3. Yörüngeye Dayalı SLAM Metotları

Yörüngeye dayalı SLAM metotları genel anlamda üç bařlıkta toplanabilir^[2]. İlk olarak FastSLAM algoritması yörünge kestirimi için bir örnek olarak düşünülebilir. Bunun sebebi de her paracığın bir yörünge hipotezi tanımlamasıdır. İkinci olarak bu bařlıkta altında sayılabilecek bir metot topojik haritalama yönteminden doğmuřtur. Bu yöntemde bileřik bir durum vektörü yerine robot pozları grafiksel bir ağı yapısında bağılantılandırılarak tanımlanır. Yörüngeye dayalı SLAM metotlarından üçüncüsü seyrek-informasyon-formu ile tanımlı SLAM’ın geliştirilmesi ile ortaya çıkmıřtır.

3.3.1 FastSLAM:

FastSLAM algoritmasında olasılık dağılımı tek bir x_k pozu üzerinde değıl tüm yörüngeyi tanımlayan $X_{0:k}$ vektörü üzerindedir. Yörünge üzerine kořullama yapıldığı an haritada tanımlı engeller birbirinden bağımsız hale gelir, bu durum řekil 3.3’te gösterilmiřtir. FastSLAM için belirleyici özellik haritanın birbirinden bağımsız gaussian dağılımlarla ifade edilip karesel karıřıklığa sahip bileřik harita kovaryansının önlenmesidir^[2].



řekil 3.3: Gözlemlerin Bağımsızlařtırılması

Geçmiş poz durumları kesin olarak bilindiği zaman, gözlemler birbirinden bağımsız olduğu için harita durumları da birbirinden bağımsızdır. FastSLAM yönteminde her parçacık farklı bir yörünge hipotezini tanımlar.

FastSLAM algoritmasında Rao-Blackwellized durumu tanımlıdır. Bu durumda yörünge belli bir ağırlığa sahip örneklerle tanımlanır ve harita analitik olarak hesaplanır. Farklı yörünge örnekleri parçacık filtresinin birer parçaları olarak tanımlanır ve her yörünge engel konumları ile ilgili kestirimlere sahiptir.

Bir k zamanındaki koşullu dağılım $\{w_k^{(i)}, X_{0:k}^{(i)}, P(m | X_{0:k}^{(i)}, Z_{0:k})\}_i^N$ seti ile tanımlıdır. Haritaya eklenen her parçacık birbirinden bağımsız Gaussian dağılıma sahiptir.

$$P(m | X_{0:k}^{(i)}, Z_{0:k}) = \prod_j^M P(m_j | X_{0:k}^{(i)}, Z_{0:k}) \quad (3.3)$$

Rekürsif kestirim yapılırken poz durumları parçacık filtresi ile, harita durumları GKF ile yapılır.

3.3.2 Topolojik Haritalama:

Topolojik haritalama yönteminde robot pozları ve aralarında tanımlanan yer ilişkileri bir ağ yapısı oluşturur.

Bu metotların sunduğu çözümlerin amacı haritaya yeni bilgiler eklenirken bu yeni bilgiler ışığında oluşturulan son pozun geçmiş pozlarla topolojik olarak doğru ilişkilendirilmesidir. Bu işlemin uzun çevrimlerden sonra bile yapılabilmesi de yöntemin tutarlılığını kanıtlamış olur.

Bu tezde de incelenen metot GCRS (globally consistent range alignment for environment mapping) ve bu metotta olduğu öne sürülen bazı eksiklerin düzeltilmesi amacıyla geliştirilmiş LRGC (Local Registration and Global Corelation) metotları bu başlık altında yer alır.

3.3.3 Seyrek-İnformasyon Formu:

Birçok durumda yörünge bilgisini elde tutmak performansa zarardan çok yarar sağlar. SLAM içinde yer alan optimizasyon problemi ‘sparse’ lineer cebir kuralları ile ifade

edilebilir. Lineer cebirin en küçük kareler yöntemine getirdiği çözümler SLAM için anlamlı olabilir.

Informasyon formu sık olarak Gaussian dağılımının kanonik veya doğal olarak ifadesi olarak adlandırılır. Bu ifade biçimini elde etmek için aşağıdaki terimler tanımlanır^[3]:

$$p(\xi_t) = N(\xi_t; \mu_t, \Sigma_t) = N^{-1}(\xi_t; \eta_t, \Lambda_t) \quad (3.4)$$

burada $\Lambda_t = \Sigma_t^{-1}$ ve $\eta_t = \Lambda_t \mu_t$

Sonuç, normal dağılımı ortama değer ve kovaryans ile parametrize etmek $N(\xi_t; \mu_t, \Sigma_t)$ yerine, informasyon vektörü ve informasyon matrisi ile parametrize etmektir $N^{-1}(\xi_t; \eta_t, \Lambda_t)$.

Bu parametrizasyon yapıldıktan sonra oluşturulan informasyon matrisi eğer engellere-dayalı bir SLAM algoritması kullanılacaksa geçmiş robot pozları durum vektöründen elimine edilirken oluşturulan engeller arası bağlantılar sebebiyle gittikçe daha yoğun bir hale gelir. Bu yüzden eski robot pozlarını kullanarak informasyon matrisi oluşturma yoluna gidilir ve bu matrisin seyrek olma ve böyle kalma özelliği korunmaya çalışılır. Bunun için geliştirilmiş çeşitli yöntemler bulunmaktadır. Bazı örnekler, eşzamanlı SAM (kare kökü) yöntemi veya ESDF(exactly sparse delayed-state filters) yöntemi olarak sayılabilir.

4. GTTH (GLOBAL TUTARLI ALAN TARAMASI HİZALAMA) YÖNTEMİ - GCRS (GLOBAL CONSISTENT RANGE SCAN ALIGNMENT)

4.1. Problem Tanımı

GTTH algoritmasının çözmeyi amaçladığı problem, ölçümlerden, burada söz konusu olan alan taramalarından, elde edilen eksen takımlarını, aralarında yer ilişkileri tanımlayarak tutarlı bir şekilde bir araya getirme problemidir ^[6].

Bu yaklaşımda bütün lokal eksen takımları ve aralarındaki yer ilişkileri algoritma için birer veri olarak değerlendirilir. Yer ilişkileri rastlantı değişkeni olarak düşünülüp alan tarama çiftlerinin ilişkilendirilmesi ve odometri ölçümleri yardımı ile oluşturulur.

Bütün yer ilişkileri elde edildikten sonra optimal olarak, maksimum olabilirlik yöntemi kriterine göre birleştirilir. Bu işlemin tutarlı bir şekilde yapılmasını, yer ilişkilerinden tanımlanan kısıtlar ve işlemin tüm eksen takımları için aynı anda uygulanıyor olması sağlar.

4.2. Tutarlı Veri Kaydetme Problemi

Bir yatay tarama tek bir robot pozundan alınan alan ölçümlerinin toplamıdır. Bir alan taraması ise çevrenin kısmi bir görüntüsüdür. Farklı yerlerde yapılan alan taramalarının birleştirilmesi çevre ile ilgili bütünsel bir tanımlama sağlar.

Çevre modeli oluşturulmasında genellikle uygulanan yöntem adım adım yeni verilerin modele eklenmesidir. Sensör verisinden yeni bir eksen takımı elde edildiğinde bir önceki eksen takımına veya toplam global modele ekleme yapılır. Yeni verilerin eksen takımının global modele eklenmesi Kalman filtresi kullanılması ile gerçekleştirilebilir. Ancak bu işlemde her zaman tutarlılık sağlanamaz çünkü veriler birbirinden bağımsız olarak güncellenir. Yeni veri entegre edildikten sonra sistemin tutarlılığı artık kontrol edilemez.

İleri safhalarda çıkabilecek tutarsızlık problemini çözebilmek için lokal eksen takımları bilgisi ile beraber kestirilen poz bilgileri de saklanmalıdır. Ayrıca, poz

düzeltilmelerini ilgili eksen takımlarına yayabilmek için sistematik bir yöntem gereklidir.

Bu yaklaşımın arkasındaki fikir, tüm eksen takımları ve eksen takımları arasındaki yer ilişkilerinin oluşturduğu ağ yapısının temel veri olarak tutulmasıdır ^[6]. Her lokal eksen takımı, tek bir robot pozundan ölçülen sensör verisinin toplamı olarak tanımlanabilir. Robot pozu aynı zamanda lokal koordinat sistemlerini tanımlamada kullanılır. Lokal eksen takımları arasındaki yer ilişkileri, tarama sonuçlarının çift çift eşleştirilmesinden veya doğrudan odometri sonuçlarından elde edilir. Robot pozlarının geçmişi, tüm lokal eksen takımı pozisyonlarını tanımlayan global bir koordinat sisteminde değişkenler olarak tanımlanır. Amaç tüm poz değişkenlerini, oluşturulan ağdaki kısıtları kullanarak kestirmek ve tarama sonuçlarını kestirimi yapılmış poz bilgilerine kaydetmektir. Lokal eksen takımları arasındaki tutarlılık, tüm yer ilişkilerinin aynı anda hesaba katılması ile sağlanır.

Önerilen bu yeni yaklaşımın eski metotlardan farklı olan özellikleri aşağıdaki gibi sıralanabilir:

İlk olarak, nesne eksen takımlarını tanımlamak için tek bir robot pozisyonundan elde edilen sensör ölçümleri kullanılır. Böylece çevrede bulunan nesneleri segmentleme ve tanıma işlemi yapılmasına gerek kalmaz. Tek bir robot pozu bir nesne eksen takımına referans oluşturur. Böylece, geçmiş robot pozlarının saklanması ile nesne eksen takımları arasındaki yer ilişkileri de saklanmış olur.

İkinci olarak, yer ilişkileri kısıtlar, konumlar ise bağımsız değişkenler olarak kullanılır. Ağ yapısında var olan ilişkiler, yeni gözlemler yapıldığında doğrudan güncellenmez. Elde edilen yeni ilişkiler ağa eklenir. Tanımlanmış tüm yer ilişkiler değişkenleri çözmek için kısıt olarak kullanılır, bunun karşılığında güncellenmiş ve tutarlı ilişki setleri oluşturulur. Nesne eksen takımları arasındaki kovaryans matrislerinin bilindiği varsayılmaz. Bu yöntemin durum bilgisi, ilişkilerin oluşturduğu veri kümesidir. Kovaryans matrisleri pozisyon değişkenleri çözülürken bulunur.

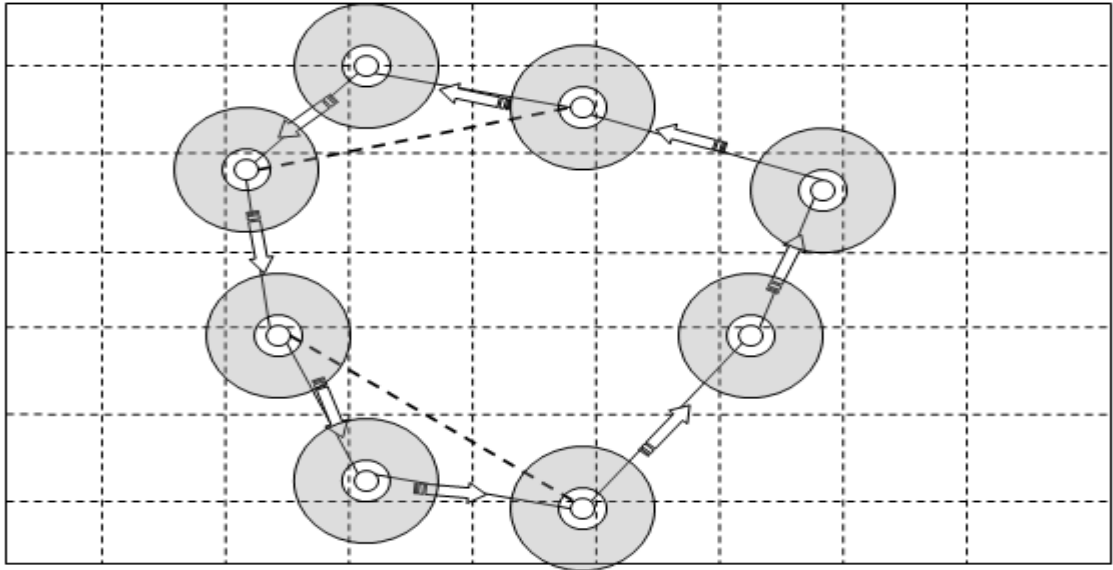
Son olarak, nesne eksen takımları arasındaki yer ilişkilerini elde etmek kolaydır. Bunun sebebi de nesne eksen takımlarının robot pozları üzerinde tanımlı olmasıdır. Odometri ölçümleri yer ilişkilerini doğrudan oluşturur. Ayrıca alan tarama sonuçları eşleştirilmesi ile de ilişki tanımları oluşturulur.

4.3. Poz ilişkileri

Çoklu tarama kaydetme yönteminde, bütün poz ilişkileri kısıt olarak kullanılarak tarama sonuçları global pozlara dönüştürülür. Taramalardan elde edilen pozlar değişken olarak düşünülür. Bir poz ilişkisi, iki poz arasındaki kestirilen yer ilişkisi olarak tanımlanır. Son olarak, bütün pozlar için bir optimizasyon problemi formüle edilerek kestirim yapılır.

4.3.1. Poz ilişkilerinin elde edilmesi

Poz ilişkileri, iki alan taramasının eşleştirilmesinden ve odometri ölçümlerinden elde edilir ^[6]. Şekil 4.1’de yapılan iki türlü ölçümü gösterilmektedir. Robot pozları etrafında çizilmiş olan diskler alan taramasını sembolize etmektedir. Bu alan taramasının açısı kullanılan sensörün tipi ile ilgilidir ve şekilde 360° olarak gösterilmiştir. Pozlar arası oklar ise robotun iki poz arasında katettiği yola bağlı ölçüm yapan odometri sensörünü temsil etmektedir.



Şekil 4.1: Ölçüm İşlemi

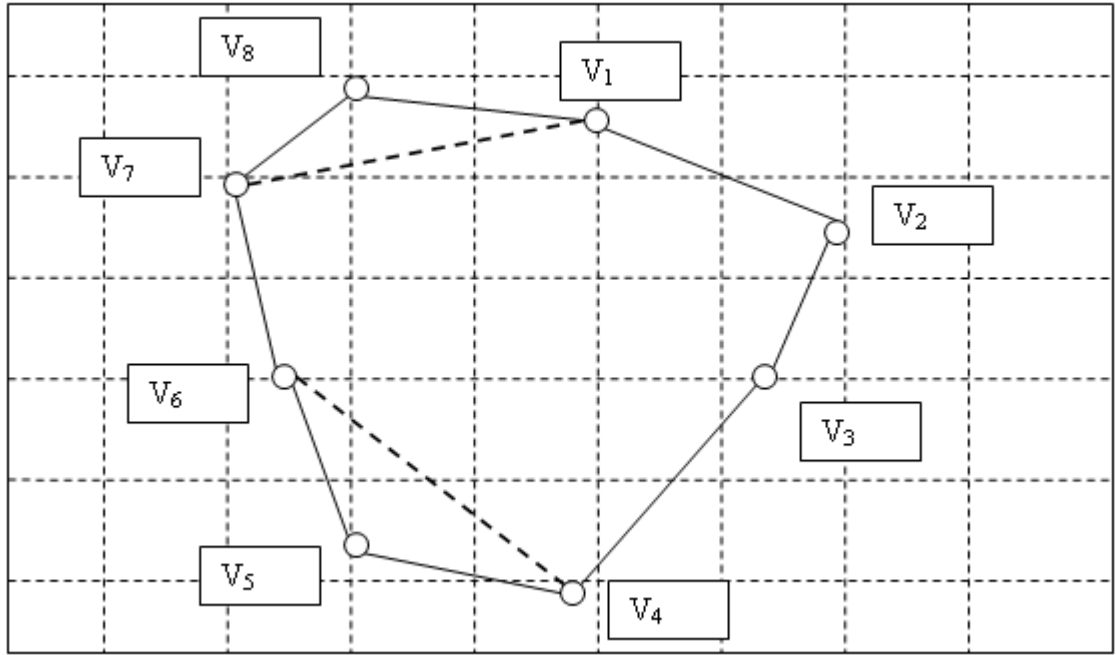
Bir tarama çiftinin eşleştirilmesi ile de poz ilişkisi tanımlanabilir ve bu ilişki odometriye göre daha kesin olarak kabul edilir.

Bu yöntemde önerilen tarama eşleştirme algoritması doğrudan noktadan noktaya eşleştirmeye dayanır ^[8]. Her tarama eşleştirme algoritması için geçerli olan bir

tanımlama yapılmak istenirse bu tip yöntemler iki tarama sonucunu ve birbirlerine göre pozlarının bir başlangıç değerini giriş olarak alıp çıkış olarak bu relatif pozun iyileştirilmiş kestirimini vermeyi amaçlar.

İki taramanın hizalanması sonucunda, bu iki tarama üzerinde karşılıklı nokta çiftleri kaydedilir. Bu nokta çiftleri daha sonra iki poz arasındaki kısıtı oluşturmada yardımcı olur.

4.3.2. Ağ yapısının oluşturulması



Şekil 4.2: Ağ Yapısı

Ağ yapısı, düğümlerden ve düğüm çiftlerinin arasındaki bağlantılardan oluşur. Şekil 4.2’de bir ağ yapısı örneği görülmektedir. Ağdaki bir düğüm, robotun yolu üzerinde durup alan taraması yapmış olduğu bir pozu temsil eder. Şekilde bu noktalar V_i notasyonu ile gösterilmiştir.

Bu algorithmada iki türlü poz ilişkisi tanımlanır. İlk olarak, iki robot pozu yörüngede birbirine komşu ise bu iki düğüm arasında ‘zayıf’ bir bağ tanımlanır ^[6]. İkinci olarak, herhangi iki pozda yapılan alan taramalarında yeterli büyüklükte bir örtüşme varsa, bu iki düğüm arasında ‘güçlü’ bir bağ tanımlanır ^[6]. İki tarama arasında yeterli bir örtüşme olup olmadığına karar vermek için bir yüzde belirlenir, deneysel ölçüm bu yüzdeden fazla çıkmalıdır.

Bu yöntemin amacı, bütün poz ilişkilerini tutarlı bir şekilde bir araya getirmektir. Ağ yapısında tanımlı global pozları temsil eden düğümleri, optimal olarak kestirmek hedeflenir. Bu optimal kestirim için bir objektif fonksiyonu tanımlanır. Bu fonksiyon oluşturulurken tüm poz koordinatları değişken olarak tanımlanır. Ağdaki her bağlantı objektif fonksiyona bir terim olarak yansıtılır ve aynı zamanda bu bağlantı iki düğümü birleştiren bir yay olarak düşünülebilir. Bu yayın minimum enerjiye ulaşması için iki düğüm arasındaki poz ilişkisinin ölçülen değerine eşit olması gerekir. Yay tanımlamasından yola çıkarsak objektif fonksiyon aynı zamanda ağdaki tüm enerjiyi temsil eder. Bu enerji minimize edilerek poz değişkenleri bulunur.

4.4 Optimal Kestirim Yöntemi:

$X_0, X_1, \dots, X_n$ gibi $n+1$ düğümden oluşan bir ağ yapısına sahip olduğumuzu düşünelim. Her X_i düğümü d -boyutlu bir pozisyon vektörünü temsil eder. İki X_i ve X_j düğümü arasında ölçülebilir fark D_{ij} bağlantısı olarak tanımlanır. Bu tanımlamalardan sonra D_{ij} 'nin, X_i ve X_j 'nin ,büyük olasılıkla lineer olmayan, bir fonksiyonu olduğu anlaşılabilir. Bu fonksiyona ölçüm denklemi denir.

D_{ij} 'nin gözlemi aşağıdaki gibi modellenir [6].

$$\bar{D}_{ij} = D_{ij} + \Delta D_{ij} \quad (4.1)$$

Burada ΔD_{ij} ortalama değeri sıfır olan ve bilinen C_{ij} kovaryans matrisine sahip rastlantısal gaussian bir hatadır. $\bar{D}_{ij}$ ölçümleri ve C_{ij} kovaryansı verildiğinde X_i pozisyonlarının optimal kestirimleri bulunur. Ayrıca ölçüm kovaryans matrislerinden yola çıkılarak kestirilen X_i 'lerin kovaryans matrisleri hesaplanır.

Algoritmada kullanılan optimal kestirim kriteri en büyük olabilirlik veya minimum varyans olarak bilinen ilkeye dayanır. Düğüm pozisyonu olan X_i 'ler D_{ij} 'lerin koşullu bileşik olasılıkları maksimize edilerek bulunur. Eğer bütün gözlem hatalarının gaussian ve birbirinden bağımsız olduğu varsayılırsa bu kriter aşağıda tanımlı Mahalanobis mesafesini minimize etmeye eşdeğerdir:

$$W = \sum_{0 \leq i \leq j \leq n} (X_i - X_j - \bar{D}_{ij})^t C_{ij}^{-1} (X_i - X_j - \bar{D}_{ij}) \quad (4.2)$$

Yukarıdaki denklemi kullanmak için her iki düğüm arasında bir bağlantı olduğu varsayılır ve her D_{ij} bağlantısı için gaussian dağılımına ve bilinen bir C_{ij} kovaryansa sahip bir $\bar{D}_{ij}$ gözlemi olduğunu kabul edilir.

W bütün X_i pozisyonlarının bir fonksiyonudur. Çözüm pozisyon farklarından yola çıkılarak yapıldığı için bir X_0 'ı sabit olarak seçilmelidir. $X_1, X_2, \dots, X_n$ düğümleri X_0 'a göre konumları ile tanımlanır.

Bu denklemdeki $\mathbf{X}$, n -boyutlu bir vektördür ve $X_1, X_2, \dots, X_n$ pozisyonlarının biraraya getirilmesi ile oluşturulur. $\mathbf{D}$ matrisi ise, $D_{ij} = X_i - X_j$ olarak tanımlı tüm pozisyon farklarını içerir. $\mathbf{H}$; 1, -1, veya 0'lardan oluşan bir matristir. Sonuç olarak W fonksiyonu matris formunda yazılacak olursa [6]:

$$W = (\bar{D} - HX)' C^{-1} (\bar{D} - HX) \quad (4.3)$$

Burada $\bar{D}$, D_{ij} 'ler için oluşturulmuş, $\bar{D}_{ij}$ kestirimlerinin arka arkaya sıralanmış halidir. $\mathbf{C}, \mathbf{D}'$ nin kovaryansıdır ve C_{ij} alt matrislerinden oluşur.

W 'yi minimize eden $\mathbf{X}$ çözümü aşağıdaki gibi verilir:

$$X = (H' C^{-1} H)^{-1} H' C^{-1} \bar{D} \quad (4.4)$$

$\mathbf{X}'$ in kovaryansı aşağıdaki gibi verilir:

$$C_X = (H' C^{-1} H)^{-1} \quad (4.5)$$

Elde edilen bu denklemlerden sonra bazı tanımlamalar yapılabilir:

İlk olarak aşağıdaki gibi bir G matrisi tanımlanır:

$$G = H' C^{-1} H \quad (4.6)$$

G matrisi $n \times n$ boyutundadır. $d \times d$ boyutundaki alt matrisleri:

$$G_{ii} = \sum_{j=0}^n C_{ij}^{-1} \quad (4.7.a)$$

$$G_{ij} = -C_{ij}^{-1} \quad \text{ve } i \neq j \quad (4.7.b)$$

İkinci bir tanım da B vektörü içindir:

$$B = H^t C^{-1} \overline{D} \quad (4.8.a)$$

$$B = \sum_{j=0; j \neq i}^n C_{ij}^{-1} \overline{D}_{ij} \quad (4.8.b)$$

Bu tanımlamalardan sonra pozisyon kestirimleri ve kovaryans G ve B ile tekrar tanımlanırsa:

$$X = G^{-1} B \quad (4.9)$$

$$C_X = G^{-1} \quad (4.10)$$

4.5. POZ İLİŞKİLERİNİN ELDE EDİLMESİ

4.5.1 Poz Birleştirme İşlemi:

Robotun başlangıç pozunu $V_b = (x_b, y_b, \theta_b)^t$ ile tanımlanan bir vektör olarak kabul edilir. Robot bu pozunu $D = (x, y, \theta)^t$ kadar değiştirip yeni pozuna, $V_a = (x_a, y_a, \theta_a)^t$ ya ulaştığı varsayılırsa, elde edilmiş olan V_a pozunu V_b ve D 'nin bileşiği olarak tanımlanır. V_a , aşağıdaki gibi ifade edilir:

$$V_a = V_b \oplus D \quad (4.11)$$

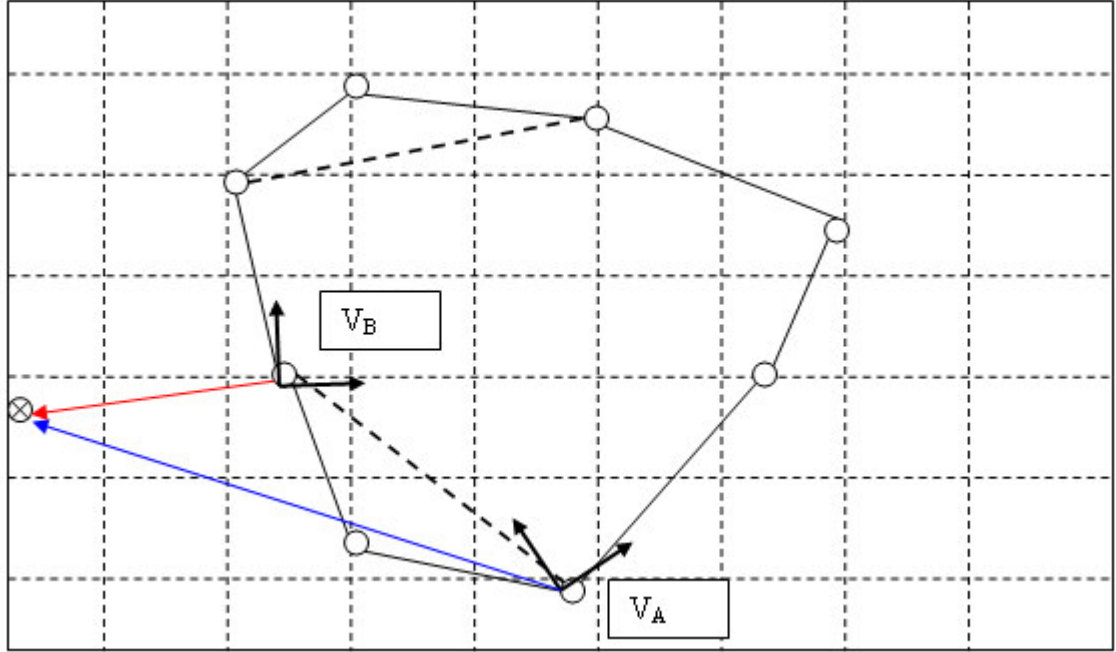
Bu işlemin tersi yani, başlangıç ve varış pozundan, poz değişimini elde etme işlemi de tanımlanabilir:

$$D = V_a \otimes V_b \quad (4.12)$$

4.5.2 Tarama Eşleştirme İşleminde Elde Edilen Poz İlişkileri

V_a ve V_b , ağ yapısında aralarında güçlü bir bağ tanımlı iki düğüm olduğunu varsayalım. Robotun, bu iki düğümde yaptığı iki tarama sırasıyla S_a ve S_b olarak tanımlansın. S_a ve S_b taramalarına eşleştirme algoritması uygulanınca u_{ka}, u_{kb} , $k = 1, \dots, m$ karşılıklı noktaları elde edilir. Bu noktalar iki boyutlu yani oryantasyonu tanımlı olmayan noktalardır. Her nokta çifti u_{ka}, u_{kb} robot çevresinde aynı noktayı göstermekle beraber farklı lokal koordinat sistemlerinde tanımlanmıştır. Şekil 4.3'te

karşılıklı bir nokta çifti gösterilmiştir. Kırmızı ve mavi ile gösterilen iki vektör global anlamda bakıldığında aynı noktaya farklı lokal eksen takımlarından bağlanmıştır. Aslında bu nokta aynı nokta olmasına rağmen iki pozda farklı koordinatlarla tanımlanmıştır.



Şekil 4.3: Karşılıklı Noktalar

Her türlü ölçme ve eşleştirme hatasının göz ardı edersek iki karşılıklı nokta arasındaki ilişki:

$$\Delta Z_k = V_a \oplus u_k^a - V_b \oplus u_k^b = 0 \quad (4.13)$$

Eğer rastlantısal ölçüm hataları dikkate alınırsa, ΔZ_k sıfır ortalamalı ve bilinmeyen bir C_k^Z kovaryansına sahip bir raslantı değişkeni olarak tanımlanabilir. Karşılıklı nokta çiftlerinden poz ilişkisi bir kısıt tanımlamak için aşağıdaki mesafe fonksiyonunu minimize edilir:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^m \left\| (V_a \oplus u_k^a) - (V_b \oplus u_k^b) \right\|^2 \quad (4.14)$$

Bu fonksiyon yeniden düzenlenirse:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^m \left\| (V_a \otimes V_b) \oplus u_k^a - u_k^b \right\|^2 \quad (4.15)$$

Bu denklemden varılabilecek sonuç F_{ab} 'nin D 'nin bir fonksiyonu olduğudur.

$$D = V_a \otimes V_b \quad (4.16)$$

D , ölçüm denklemi olarak kabul edilir.

F_{ab} denklemini daha önce tanımlanmış olan objektif fonksiyonuna yani Mahalanobis mesafe formuna getirmek için ΔZ_k 'nin her terimi lineerleştirilmelidir. Bunu yapmak için V_a ve V_b için yakın kestirimler tanımlanır:

$$\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{\theta}_a)^t \quad (4.17.a)$$

$$\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{\theta}_b)^t \quad (4.17.b)$$

Bu kestirimler tanımlandıktan sonra her iki düğüm için de hata terimleri tanımlanır:

$$\Delta V_a = \bar{V}_a - V_a \quad (4.18.a)$$

$$\Delta V_b = \bar{V}_b - V_b \quad (4.18.b)$$

Karşılıklı noktaların global bir koordinat sistemine uyarlandığı düşünülürse:

$$u_k = (x_k, y_k)^t = V_a \oplus u_k^a \approx V_b \oplus u_k^b \quad (4.19)$$

ΔV_a ve ΔV_b 'nin küçük kabulü ile ΔZ_k 'nin Taylor açılımı:

$$\Delta Z_k = V_a \oplus u_k^a - V_b \oplus u_k^b = (\bar{V}_a - \Delta V_a) \oplus u_k^a - (\bar{V}_b - \Delta V_b) \oplus u_k^b \quad (4.20)$$

$$\Delta Z_k \approx (\bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b) - \begin{pmatrix} 1 & 0 & -y_k \\ 0 & 1 & x_k \end{pmatrix} (\bar{H}_a \Delta V_a - \bar{H}_b \Delta V_b) \quad (4.21)$$

Bu denklemde geçen $\bar{H}_a$ ve $\bar{H}_b$:

$$\bar{H}_a = \begin{pmatrix} 1 & 0 & \bar{y}_a \\ 0 & 1 & -\bar{x}_a \\ 0 & 0 & 1 \end{pmatrix} \quad (4.22.a)$$

$$\bar{H}_b = \begin{pmatrix} 1 & 0 & \bar{y}_b \\ 0 & 1 & -\bar{x}_b \\ 0 & 0 & 1 \end{pmatrix} \quad (4.22.b)$$

Hata denklemi tekrar yazılacak olursa:

$$\Delta Z_k \approx \bar{Z}_k - M_k D \quad (4.23)$$

Burada terimler açıkça yazılacak olursa:

$$\bar{Z}_k = \bar{V}_a \oplus u_k^a - \bar{V}_b \oplus u_k^b \quad (4.24)$$

$$M_k = \begin{pmatrix} 1 & 0 & -y_k \\ 0 & 1 & x_k \end{pmatrix} \quad (4.25)$$

$$D = (\bar{H}_a \Delta V_a - \bar{H}_b \Delta V_b) \quad (4.26)$$

Yukarıda tanımlı D artık ölçüm denklemi olarak kabul edilip $F_{ab}(D)$ bu denkleme göre oluşturulur. $F_{ab}(D)$ eşitliğinin matris formunu elde etmek için $\bar{Z}_k$ değerleri $2m \times 1$ boyutunda Z vektöründe, M_k 'lar $2m \times 3$ boyutunda M matrisinde yazılır ve elde edilen formülasyon:

$$F_{ab}(D) \approx (Z - MD)^t (Z - MD) \quad (4.27)$$

F_{ab} 'yi minimize eden D bulunup $\bar{D}$ 'ye eşitlenirse:

$$\bar{D} = (M^t M)^{-1} M^t Z \quad (4.28)$$

Hata kovaryans matrislerinin sahip olduğu düşünüldüğü form:

$$C_k^Z = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix} \quad (4.29)$$

$\bar{D}$ için kestirilen kovaryans C_D aşağıdaki şekilde tanımlanır:

$$C_D = s^2 (M^T M)^{-1} \quad (4.30)$$

Yukarıda geçen s^2 , σ^2 nin bir kestirimi olup F_{ab} ile hesaplanır:

$$s^2 = (Z - M \bar{D})^T (Z - M \bar{D}) / (2m - 3) = F_{ab}(\bar{D}) / (2m - 3) \quad (4.31)$$

Artık lineerize edilmiş denklemler kullanılarak bu poz ilişkisi için enerji denklemi oluşturulur:

$$W_{ab} \approx (\bar{D} - D)^T C_D^{-1} (\bar{D} - D) \quad (4.32)$$

4.5.3 Odometri Ölçümlerinden Elde Edilen Poz İlişkileri

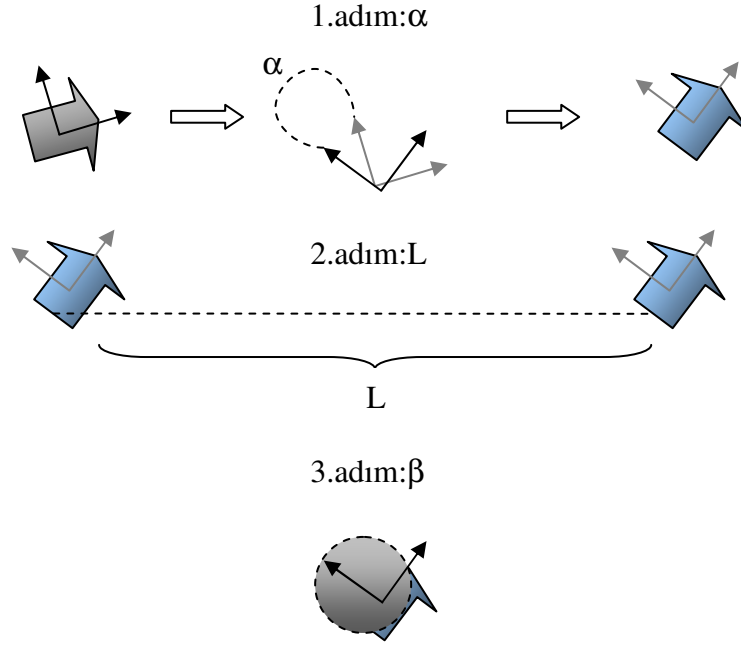
Her güçlü bağlantı için bir enerji terimi tanımlandığı gibi zayıf bağlantılar için de aynı işlem yapılır. Odometri ölçümleri bize robotun V_b pozundan V_a pozuna gelmesi için yaptığı D' poz değişikliği bilgisini verir. Bu yüzden ölçüm denklemi aşağıdaki gibi tanımlanır:

$$D' = V_a \otimes V_b \quad (4.33)$$

Bu ölçüm denklemden yola çıkılarak objektif fonksiyonundaki enerji terimi şu şekilde tanımlanır:

$$W_{ab} = (\bar{D}' - D')^T (C')^{-1} (\bar{D}' - D') \quad (4.34)$$

Bu denklemde $\bar{D}'$, D' ölçümünün gözlenen değeridir. C' ise bu ölçümde yapılan hatanın kovaryansıdır. Ölçüm hatasının kovaryansının kestirilmesi için önce robotun hareket sıralaması göz önüne alınmalıdır. Robot ilk olarak yeni hedef pozisyonuna bakmak için platformunu α değerinde bir açı ile döndürür. İkinci olarak yeni pozisyonuna erişmek için L mesafesi boyunca yer değiştirir. Son olarak elde ettiği bu son pozisyonunda platformu sabit iken robot üzerinde bulunan sensör β gibi bir açı değerinde tarama yapar^[6]. Bu işlem sırası Şekil 4.4'te gösterilmiştir.



Şekil 4.4: Kovaryans Kestirimi

α , L ve β değerlerindeki sapmalar σ_α , σ_L ve σ_β olarak tanımlanır ve değerleri ile doğru orantılı olduğu varsayılır. Bu sıralı poz değişikliği üç boyutlu bir D' vektörü ile tanımlanırsa:

$$D' = (x, y, \theta)^t \quad (4.35)$$

$$x = L \cdot \cos \alpha ; \quad y = L \cdot \sin \alpha ; \quad \theta = \alpha + \beta \quad (4.36)$$

D' poz değişikliği için C' kovaryansını tanımlarsak:

$$C' = J \begin{pmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_L^2 & 0 \\ 0 & 0 & \sigma_\beta^2 \end{pmatrix} J^t \quad (4.37)$$

Burada J , Jakobiyen matrisi olup $(x, y, \theta)^t$ değerlerinin $(\alpha, L, \beta)^t$ değerlerine göre kısmi türevlerinden oluşur:

$$J = \begin{pmatrix} -L \sin \alpha & \cos \alpha & 0 \\ L \cos \alpha & \sin \alpha & 0 \\ 1 & 0 & 1 \end{pmatrix} \quad (4.38)$$

D' poz değişimini enerji terimi oluşturmada kullanabilmek için lineerleştirmek gereklidir. D' ölçümü için hata tanımlanacak olursa:

$$\Delta D' = \bar{D}' - D' \quad (4.39)$$

Herhangi iki poz için yakın kestirmelerinin ifadesi aşağıdaki şekildedir:

$$\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{\theta}_a)^t \quad (4.40.a)$$

$$\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{\theta}_b)^t \quad (4.40.b)$$

Bu kestirimler tanımlandıktan sonra her iki düğüm için hata terimleri tanımlanır:

$$\Delta V_a = \bar{V}_a - V_a \quad (4.41.a)$$

$$\Delta V_b = \bar{V}_b - V_b \quad (4.41.b)$$

$\Delta D'$ için Taylor açılımı yapılırsa:

$$\Delta D' = \bar{D}' - D' = \bar{D}' - ((\bar{V}_a - \Delta V_a) \otimes (\bar{V}_b - \Delta V_b)) \quad (4.42)$$

$$\Delta D' \approx \bar{D}' - (\bar{V}_a \otimes \bar{V}_b) + \bar{K}_b^{-1} (\Delta V_a - \bar{H}_{ab} \Delta V_b) \quad (4.43)$$

Burada tanımlı $\bar{K}_b^{-1}$ ve $\bar{H}_{ab}$

$$\bar{K}_b^{-1} = \begin{pmatrix} \cos \bar{\theta}_b & \sin \bar{\theta}_b & 0 \\ -\sin \bar{\theta}_b & \cos \bar{\theta}_b & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.44)$$

$$\bar{H}_{ab} = \begin{pmatrix} 1 & 0 & -\bar{y}_a + \bar{y}_b \\ 0 & 1 & \bar{x}_a - \bar{x}_b \\ 0 & 0 & 1 \end{pmatrix} \quad (4.45)$$

Bu adımda yeni bir gözlem hatası tanımlanır:

$$\Delta D = -\bar{H}_a \bar{K}_b \Delta D' \quad (4.46)$$

$$\Delta D = \bar{D} - (\bar{H}_a \Delta V_a - \bar{H}_b \Delta V_b) = \bar{D} - D \quad (4.47)$$

Burada tanımlı $\bar{D}$ ve D için denklemler aşağıdaki gibi yazılabilir:

$$\bar{D} = \bar{H}_a \bar{K}_b ((\bar{V}_a \otimes \bar{V}_b) - \bar{D}') \quad (4.48)$$

$$D = \bar{H}_a \Delta \bar{V}_a - \bar{H}_b \Delta \bar{V}_b \quad (4.49)$$

Yeni üretilmiş olan D ölçüm denklemi tarama eşleştirmesinden elde edilen ölçüm denklemi ile örtüşmektedir. $\bar{D}$, D'nin gözlemi olarak düşünülebilir. $\bar{D}$ için kovaryans C, $\bar{D}'$ 'nin kovaryansı C' ile hesaplanır:

$$C = \bar{H}_a \bar{K}_b C' \bar{K}_b' \bar{H}_a' \quad (4.50)$$

Yeni enerji terimi artık aşağıdaki şekli alır:

$$W_{ab} \approx (\bar{D} - D)' C^{-1} (\bar{D} - D) \quad (4.51)$$

4.6. Optimal Poz Kestirimi:

İki tip ölçüm için de enerji terimleri oluşturulduktan sonra tüm ağ için poz kestirimlerinin yapılması gerekir. Ağ yapısında V_i , $i = 0, 1, \dots, n$ gibi robot pozlarından oluştuğu düşünülürse toplam enerji fonksiyonu aşağıdaki gibi tanımlanabilir:

$$W = \sum_{(i,j)} (\bar{D}_{ij} - D_{ij})' C_{ij}^{-1} (\bar{D}_{ij} - D_{ij}) \quad (4.52)$$

Bu denklemde yer alan D_{ij} , V_j ve V_i arasında tanımlı, lineerleştirilmiş poz farkıdır.

İki tip bağlantıdan da elde edilen ölçüm denklemleri:

$$D_{ij} = \bar{H}_i \Delta V_i - \bar{H}_j \Delta V_j \quad (4.53)$$

D_{ij} değerleri için odometri ve alan taraması ölçümlerinden $\bar{D}_{ij}$ gözlemleri doğrudan elde edilir. C_{ij} kovaryansı da bilinmektedir.

Ağ yapısındaki bir düğümü ifade eden durum vektörü:

$$X_i = \bar{H}_i \Delta V_i \quad (4.54)$$

olarak tanımlanır. Bu durum vektörlerinin ve kovaryanslarının C_i^X 'lerin hesaplanması daha önceki bölümde tanımlanmış olan G matris ve B vektörünün oluşturulması ile olur.

V_i ve C_i için hesaplanan durum vektörü ile güncelleme yapılırsa

$$V_i = \bar{V}_i - \bar{H}_i^{-1} X_i \quad (4.55)$$

$$C_i = (\bar{H}_i^{-1}) C_i^X (\bar{H}_i^{-1})^t \quad (4.56)$$

Bu poz kestirimi ve kovaryans hesabı başlangıç pozunun orijinde olduğu varsayımı ile yapılmıştır. Ancak durum böyle değilse çözüm için başlangıç pozuna göre bileşke işlemi yapılmalıdır:

$$V_i' = V_0 \oplus V_i \quad (4.57)$$

$$C_i' = K_0 C_i K_0^t \quad (4.58)$$

$$K_0 = \begin{pmatrix} \cos \theta_0 & -\sin \theta_0 & 0 \\ \sin \theta_0 & \cos \theta_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.59)$$

4.7. Rekürsif Kestirimi:

Ardışıl algoritma daha önceden ziyaret edilmiş noktalarda elde edilmiş en iyi kestirimleri saklar. Her yeni pozda, hem varılmış olan bu pozla ilgili hem de geçmiş pozlarla ilgili ölçümler yapılır. Yapılan bu yeni ölçümler sayesinde yeni pozun kestirimi yapılır ve eski pozlar güncellenir.

$X_1, \dots, X_{n-1}$ vektörleri daha önce kestirim yapılmış olan poz vektörleri olsun. X_n ise ölçüm yapmak üzere olduğumuz pozı gösteren durum vektörü olsun. X_1 'den başlayıp son poz durumu X_n ile biten bir $\mathbf{X}$ vektörü oluşturulur. Bu $\mathbf{X}$ vektörü için yapılan ilk kestirim $\mathbf{X}_0$ ve bu kestirimin kovaryansı $\mathbf{C}_{X_0}^{-1}$ olur. X_n ile ilgili henüz bir tanımlama yapılmadığı için $\mathbf{X}_0$ içinde X_n bileşeninin bulunması gereken yerde keyfi bir değer vardır. Aynı şekilde $\mathbf{C}_{X_0}^{-1}$ kovaryans matrisinde son üç sıra ve satırda sıfırlar yer alır. X_n durumunun tanımladığı pozda yapılmış olan ve aynı zamanda eski pozları da

ilgilendiren yeni ölçümler sisteme eklenmelidir. Ölçüm denkleminin matris formu hatırlanırsa:

$$D = HX \quad (4.60)$$

D değerlerinin ölçümleri $\bar{D}$ vektörü olup kovaryansı da C_D matrisidir. Yeni ölçümler eklendikten sonra X vektörünün güncellenmiş kestirimi en büyük olabilirlik ilkesine göre aşağıdaki fonksiyonu minimize eden değerdir:

$$W = (X - X_0)^t C_{X_0}^{-1} (X - X_0) + (\bar{D} - HX)^t C_D^{-1} (\bar{D} - HX) \quad (4.61)$$

X vektörü için çözüm aşağıdaki gibi verilebilir:

$$X = (C_{X_0}^{-1} + H^t C_D^{-1} H)^{-1} (C_{X_0}^{-1} X_0 + H^t C_D^{-1} \bar{D}) \quad (4.62)$$

X 'in yeni kovaryansı:

$$C_X = (C_{X_0}^{-1} + H^t C_D^{-1} H)^{-1} \quad (4.63)$$

X ve C_X güncellenirken G ve B değerleri her ölçümden sonra düzenli olarak hesaplanır.

$$G = \sum H^t C_D^{-1} H \quad (4.64)$$

$$B = \sum H^t C_D^{-1} \bar{D} \quad (4.65)$$

Her adımda yapılacak işlem öncelikle G matrisinin ve B vektörünün boyutunu X_n ekleneceği düşünülerek arttırmaktır. Daha sonra G ve B güncellenir:

$$G \leftarrow G + H^t C_D^{-1} H \quad (4.66)$$

$$B \leftarrow B + H^t C_D^{-1} \bar{D} \quad (4.67)$$

X ve C_X bulunan bu değerlerden hesaplanılacak olursa

$$X = G^{-1} B \quad (4.68)$$

$$C_x = G^{-1} \quad (4.69)$$

bulunur.

Bu algoritma ile ilgili karşılaşılabilecek problemlerden biri $\mathbf{X}$ vektörünün giderek büyümesidir. Robot yolu uzadıkça algoritma bellek ve buna bağlı performans problemleri yaratır. Bunun için önerilebilecek bir çözüm yeni pozlar eklenirken bazı eski pozların silinmesidir. Alternatif bir yöntemde büyük bir ağ yapısı olduğu zaman bu yapıyı daha küçük parçalara bölmektir.

4.8. Algoritmanın Uygulanması

Algoritmanın uygulanmasında öncelikle ağ yapısının oluşturulması gerekir. Ağ yapısı ortaya çıktıktan sonra ilk olarak odometri ölçümleri değerlendirilir. Ölçüm sonuçlarından başlangıç poz kestirimleri $\bar{V}_1, \dots, \bar{V}_n$ bulunur. Bu başlangıç pozları arasında bağlantılar oluşturulur. Bağlantılardan yola çıkılarak ölçüm vektörleri $\bar{D}_{ij}$ ve kovaryans matrisleri C_{ij} hesaplanır. Bu vektörlerin ve kovaryansların hesaplanması bağlantının zayıf veya güçlü olmasına göre farklı şekillerde hesaplanır. Son olarak $\mathbf{GX} = \mathbf{B}$ sistemi oluşturulur ve $\mathbf{X}$ değerleri bulunur.

$\mathbf{G}$ matrisini ve $\mathbf{B}$ vektörünü oluşturan parçalar C_{ij}^{-1} ve $C_{ij}^{-1}\bar{D}_{ij}$ değerleridir. Güçlü bağlantı durumunda bu değerler $C_{ij}^{-1} = (M^t M) / s^2$ ve $C_{ij}^{-1}\bar{D}_{ij} = (M^t Z) / s^2$. Zayıf bağlantı durumunda ise bu değerler 3x3 boyutlu matrislerin çarpımı ile bulunur. Hesaplama adımlarında en zor işlem $\mathbf{G}$ matrisinin tersini alıp $\mathbf{X}$ durum vektörünün kovaryansını hesaplamaktır.

Algoritmanın kolay anlaşılması açısından ağ yapısının düzgün oluşturulması gerekir. Ağ yapısı düğüm ve bağlantı listelerinden oluşur. Her bağlantı; bağlantı tipi, arasında yer aldığı iki düğümün etiketleri, hesaplanmış ölçüm değeri ve ölçümün kovaryans matrisi bilgilerini içerir. Her düğüm ise bir alan taraması bilgisini içerir.

5. SİMÜLASYON ÇALIŞMALARI

5.1. Araç Yörüngesi ve Ağın Oluşturulması

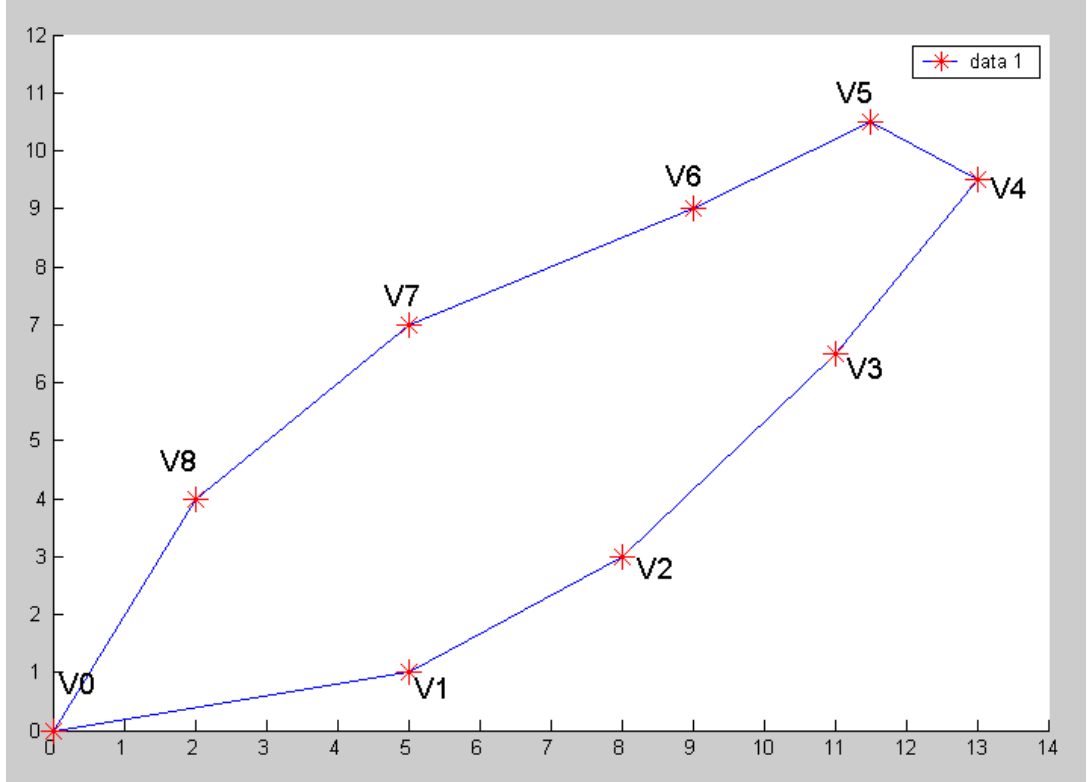
Simülasyon çalışmalarındaki amaç GTTH yönteminin performansını test etmek ve sadece zayıf bağlantılardan oluşan bir sistem yapısını zayıf ve güçlü bağlantılardan oluşan sistem yapısı ile kıyaslamaktır. Simülasyonun ilk aşaması aracın takip edeceği yolun yani yörüngesinin kararlaştırılmasıdır. Bu yöntemde araç, yörüngesi boyunca hareketini sürdürürken belli aralıklarla hareketini durdurup üzerine monte edilmiş olan lazer sensörü ile alan taraması yapmaktadır. Tarama yapmak için hareketine ara verdiği pozisyonlar, çizilen yolun yapı taşları olup ağın oluşturulmasında kullanılır. Robot yörüngesini tamamladıktan sonra başlangıç noktasına geri dönmektedir.

Bu simülasyon için başlangıç pozisyonu dahil dokuz pozisyonlu bir yörünge tasarlanmıştır. Bu pozisyonlar Tablo 5.1 'de yer almaktadır.

Tablo 5.1: Yörünge Üzerinde Ölçüm Yapılan Pozisyonlar

	Global X Eksenindeki Değer	Global Y Eksenindeki Değer	Lokal Eksen Takımının Global Eksen Takımına Göre Oryantasyonu
Pozisyon İsimleri			
V_0	0	0	0
V_1	5.0000	1.0000	0.1974
V_2	8.0000	3.0000	0.5880
V_3	11.0000	6.5000	0.8622
V_4	13.0000	9.5000	0.9828
V_5	11.5000	10.5000	2.5536
V_6	9.0000	9.0000	3.6820
V_7	5.0000	7.0000	3.6052
V_8	2.0000	4.0000	3.9270

Bu pozisyonlardan ışığında yörünge görünümü Şekil 5.1'deki gösterildiği gibi olacaktır.



Şekil 5.1: Ölçüm Yapılan Pozisyonlarla Yörünge Gösterilimi

Simülasyonun çıkış noktası olan yörünge ve pozisyon bilgileri yukarıda belirtilmiştir. Bundan sonraki adım bu pozisyonlara hata içeren başlangıç değerleri atamaktır. Aynı zamanda bu pozisyonlar arası ilişkilerin tanımlanması ve hatalı başlangıç değerleri ile ilk hesapların yapılması gerekmektedir.

5.2. Başlangıç Pozları ve İlk İlişkilerin Hesaplanması

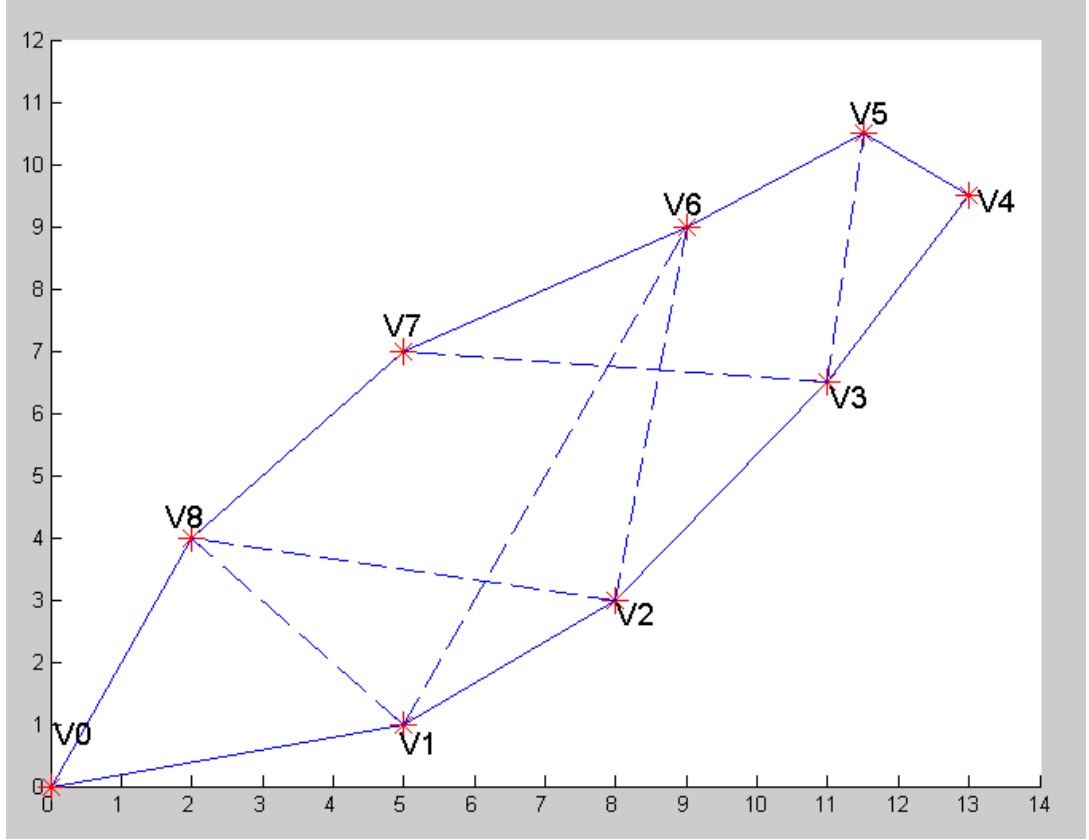
Robot pozları için doğru bir kestirim yapılmadan önce robotun bulunduğu ortamda amaçsız bir yörünge turu yaptığı söylenebilir. Bu turu yaparken robot ölçüm yaptığı pozisyonların konumları ile ilgili odometri sensörü yardımı ile hatalı başlangıç değerleri elde eder. Bu değerler kestirim algoritmasının ilk değerleridir. Simülasyonda kullanılan başlangıç değerleri Tablo 5.2’de listelenmiştir.

Tablo 5.2: Pozisyonlar için Odometri ile Elde Edildiği Varsayılan Başlangıç Değerleri

	Global X Eksenindeki Değer	Global Y Eksenindeki Değer	Lokal Eksen Takımının Global Eksen Takımına Göre Oryantasyonu
Pozisyon İsimleri			
V_0	0	0	0
V_1	5.2000	1.2000	0.2268
V_2	7.8000	2.9000	0.628
V_3	10.7000	6.5000	0.8922
V_4	12.8000	9.6000	1.0228
V_5	11.6000	10.4000	2.5836
V_6	8.8000	9.2000	3.722
V_7	4.8000	7.1000	3.610
V_8	2.1000	4.2000	3.957

Başlangıç değerleri oluşturulurken pozisyonel hatalarda 0.3 civarında ve oryantasyonel hatalarda 0.05 civarında bir değer kullanılmıştır.

Bu ilk değerlerden yola çıkılarak ilk poz ilişkileri hesaplanmalıdır. Bunun için öncelikle zayıf ve güçlü bağlar belirlenmelidir. Zayıf bağlar zaten birbirine komşu olan iki pozisyon arasındaki bağlardır. Güçlü bağlar ise iki komşu olmayan pozisyon arasında tanımlanmalıdır. Simülasyonda kullanılacak bağlantılar Şekil 5.2’de gösterilmiştir.



Şekil 5.2: Zayıf ve Güçlü Bağlar ile Pozisyonların Gösterilimi

Şekilde normal çizgi ile gösterilen bağlar zayıf bağlantılar, kesikli çizgi ile gösterilen bağlar ise güçlü bağlantılardır. İki pozisyon arasında güçlü bağlantı tanımlayabilmek için bu pozisyonlarda yapılan alan taramaları incelenmelidir. Eğer kıyaslanan iki alan taraması arasında yeteri kadar büyüklükte bir örtüşme varsa bu iki pozisyon arasında alan taramaları üstünde karşılıklı noktalar tanımlanır. Bu karşılıklı noktalar global eksen takımında aynı noktayı göstermekle beraber lokal eksen takımlarında farklı vektörlere denk düşerler. Simülasyonda alan taraması simüle edilemeyeceğinden güçlü bağlantılar ve karşılıklı noktalar keyfi seçilmiştir.

5.2.1 Zayıf Bağlantıların Hesaplanması

Zayıf bağlantıların hesaplanmasında kullanılan denklemler 4.48 ve 4.50’de verilmiştir. İlk hesaplama örnek olarak yapılacak olursa:

$$\bar{V}_1 = (5.2 ; 1.2 ; 0.2268)^t \quad (5.1)$$

Bu pozdan yola çıkılarak hesaplanan matrisler:

$$\overline{H}_1 = \begin{bmatrix} 1 & 0 & 1.2 \\ 0 & 1 & -5.2 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$$\overline{K}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

İlk zayıf bağlantı hesaplanacak olursa,

$$\overline{D}_{01} = \overline{H}_1 * \overline{K}_0 * \begin{bmatrix} 0.2 \\ 0.2 \\ 0.04 \end{bmatrix} = \begin{bmatrix} 0.2480 \\ -0.0080 \\ 0.0400 \end{bmatrix} \quad (5.4)$$

Bu zayıf ilişki için aynı zamanda kovaryans hesaplanması gerekmektedir. Bu matrisin hesaplanmasında ölçüm hatalarından yola çıkılır. Ölçüm hatası hesaplanacak üç büyüklük: iki pozisyon arasındaki hareket mesafesi, bir pozisyondan diğerine hareket etmeden önce oryantasyonda yapılan değişiklik ve alan taraması yapılan açı. Bu üç büyüklük C_{01dev} matrisini oluşturmamızda kullanılır.

$$C_{01dev} = \begin{bmatrix} 0.0566 & 0.0068 & -0.0009 \\ 0.0068 & 0.0238 & 0.0043 \\ -0.0009 & 0.0043 & 0.0009 \end{bmatrix} \quad (5.5)$$

Bu matris elde edildikten sonra bu ilişki için kovaryans matrisi aşağıdaki formüle göre hesaplanır:

$$C_{01} = \overline{H}_1 * \overline{K}_0 * C_{01dev} * \overline{K}_0^t * \overline{H}_1^t \quad (5.6)$$

$$C_{01} = \begin{bmatrix} 0.0558 & 0.0111 & 0.0002 \\ 0.0111 & 0.0023 & -0.0002 \\ 0.0002 & -0.0002 & 0.0009 \end{bmatrix} \quad (5.7)$$

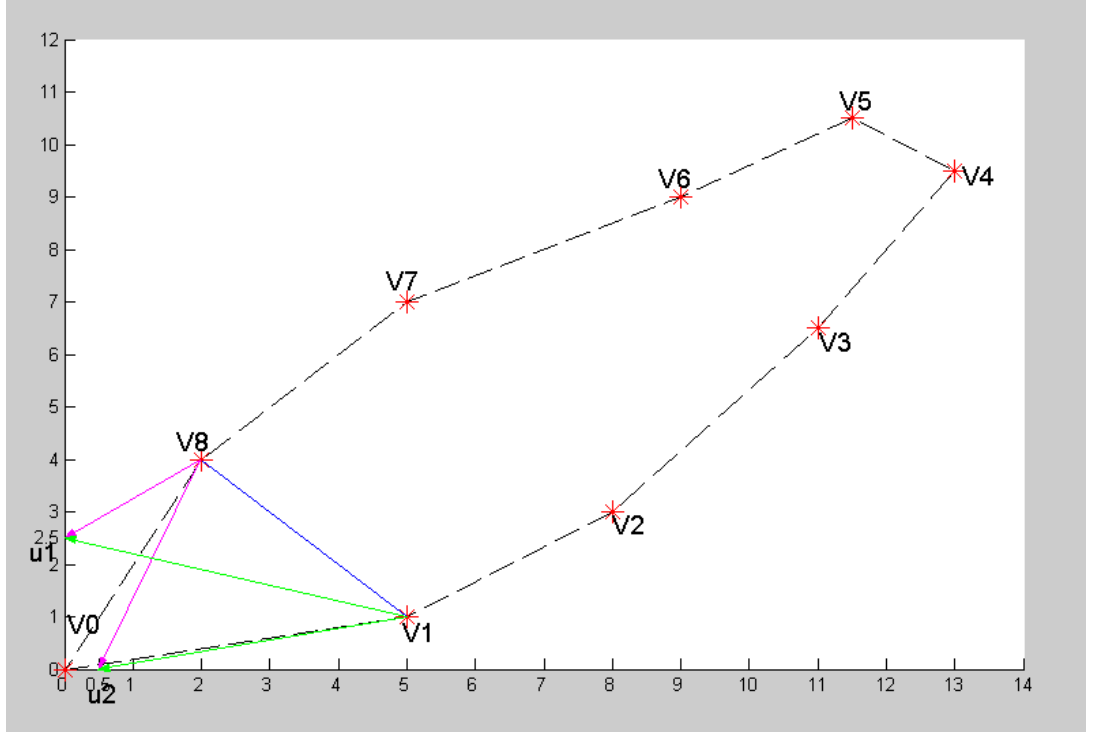
5.2.2 Güçlü Bağlantıların Hesaplanması

Simülasyonun gerçekleştirilmesi için altı adet güçlü bağ seçilmiştir. Her güçlü bağ için iki adet karşılıklı nokta belirlenmiştir. Güçlü bağlar ve seçilen karşılıklı noktalar Tablo 5.3.'te gösterilmiştir.

Tablo 5.3: Güçlü Bağlantılar ve İlgili Karşılıklı Noktalar

Güçlü Bağlantı	Karşılıklı Nokta-1	Karşılıklı Nokta-2
$\overline{D}_{18}$	[0 ; 2.5]	[0.5 ; 0]
$\overline{D}_{16}$	[0 ; 6.5]	[10 ; 0]
$\overline{D}_{26}$	[0 ; 4.5]	[12 ; 0]
$\overline{D}_{37}$	[0 ; 2.5]	[7 ; 0]
$\overline{D}_{28}$	[0 ; 2]	[6 ; 0]
$\overline{D}_{35}$	[0 ; 8]	[13 ; 0]

Güçlü bağlantıların hesaplanması için yapılması gereken ilk işlem karşılıklı noktaların her iki pozisyonunun lokal ekseninde de koordinatlarının belirlenmesidir. Bu işlem Şekil 5.3'te gösterilmiştir.



Şekil 5.3: Bir Güçlü Bağlantıya Ait Karşılıklı Nokta Çiftinin Lokal Eksenlerdeki Yeri

Şekilde ilk bağlantı için belirlenmiş iki karşılıklı nokta $u_1 = [0 ; 2.5]$ ve $u_2 = [0.5 ; 0]$ noktaları gösterilmiştir. Bu koordinatlar global eksen takımı için geçerlidir. V_1 ve V_8 pozisyonlarındaki lokal eksenlere göre bu karşılıklı noktaların tanımlanması gerekir, yani V_8 lokal ekseninde mor renkte gösterilmiş vektörler, V_1 lokal ekseninde ise yeşil ile gösterilmiş vektörler tanımlanmalıdır. Sonuç olarak:

u_1 noktasının V_8 pozisyonunun lokal eksenindeki gösterilimi,

$$u_1^8 = \begin{bmatrix} 2.4749 \\ -0.3536 \end{bmatrix} \quad (5.8)$$

u_1 noktasının V_1 pozisyonunun lokal eksenindeki gösterilimi,

$$u_1^1 = \begin{bmatrix} -4.61 \\ 2.45 \end{bmatrix} \quad (5.9)$$

u_2 noktasının V_8 pozisyonunun lokal eksenindeki gösterilimi,

$$u_2^8 = \begin{bmatrix} 3.8891 \\ 1.7678 \end{bmatrix} \quad (5.10)$$

u_2 noktasının V_1 pozisyonunun lokal eksenindeki gösterilimi,

$$u_2^1 = \begin{bmatrix} -4.6088 \\ -0.0981 \end{bmatrix} \quad (5.11)$$

Bu hesaplamadan sonra (4.24), (4.25) ve (4.26) denklemlerinden yola çıkılarak:

$$\bar{Z}_1 = \begin{bmatrix} -0.0526 \\ 0.1433 \end{bmatrix} \quad (5.12)$$

$$\bar{Z}_2 = \begin{bmatrix} -0.1192 \\ 0.1274 \end{bmatrix} \quad (5.13)$$

$$\bar{M}_1 = \begin{bmatrix} 1 & 0 & -2.5 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.14)$$

$$\bar{M}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \end{bmatrix} \quad (5.15)$$

Bu vektörler bir araya getirildiğinde,

$$M = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (5.16)$$

$$Z = \begin{bmatrix} \bar{Z}_1 \\ \bar{Z}_2 \end{bmatrix} \quad (5.17)$$

Sonuç olarak $\bar{D}_{18}$ bağlantısı hesaplanacak olursa,

$$\bar{D}_{18} = (M^t * M)^{-1} M^t Z \quad (5.18)$$

$$\bar{D}_{18} = \begin{bmatrix} -0.1194 \\ 0.1421 \\ -0.0268 \end{bmatrix} \quad (5.19)$$

Kovaryans ise (4.30) ve (4.31) denklemleri ile bulunur,

$$C_{D16} = \begin{bmatrix} 0.3139 & -0.0308 & 0.1231 \\ -0.0308 & 0.1662 & -0.0246 \\ 0.1231 & -0.0246 & 0.0985 \end{bmatrix} * 10^{-5} \quad (5.20)$$

5.3. G Matrisi ve B Vektörünün Hesaplanması ve Sistemin Oluşturulması

Simülasyon için sıradaki adım hesaplanan zayıf ve güçlü bağlar ve bunların kovaryansları ile ağ yapısını doğru bir şekilde ifade edebilmektir. Bunun için ağ sistemini G sistem matrisi ve B sistem vektörü ile ifade etmek gerekir. G ve B (4.7.a), (4.7.b), (4.8.a) ve (4.8.b) eşitlikleri ile hesaplanır. Bu örnekte elde edilen G matrisinin elemanları:

$$G_{11} = C_{01}^{-1} + C_{12}^{-1} + C_{16}^{-1} + C_{18}^{-1} \quad (5.21)$$

$$G_{22} = C_{12}^{-1} + C_{23}^{-1} + C_{26}^{-1} + C_{28}^{-1} \quad (5.22)$$

$$G_{33} = C_{23}^{-1} + C_{34}^{-1} + C_{35}^{-1} + C_{37}^{-1} \quad (5.23)$$

$$G_{44} = C_{34}^{-1} + C_{45}^{-1} \quad (5.24)$$

$$G_{55} = C_{35}^{-1} + C_{45}^{-1} + C_{56}^{-1} \quad (5.25)$$

$$G_{66} = C_{16}^{-1} + C_{26}^{-1} + C_{56}^{-1} + C_{67}^{-1} \quad (5.26)$$

$$G_{77} = C_{37}^{-1} + C_{67}^{-1} + C_{78}^{-1} \quad (5.27)$$

$$G_{88} = C_{18}^{-1} + C_{28}^{-1} + C_{78}^{-1} + C_{80}^{-1} \quad (5.28)$$

$$G_{12} = G_{21} = - C_{12}^{-1} \quad (5.29)$$

$$G_{13} = G_{31} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.30)$$

$$G_{14} = G_{41} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.31)$$

$$G_{15} = G_{51} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.32)$$

$$G_{16} = G_{61} = - C_{16}^{-1} \quad (5.33)$$

$$G_{17} = G_{71} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.34)$$

$$G_{18} = G_{81} = - C_{18}^{-1} \quad (5.35)$$

$$G_{23} = G_{32} = - C_{23}^{-1} \quad (5.36)$$

$$G_{24} = G_{42} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.37)$$

$$G_{25} = G_{52} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.38)$$

$$G_{26} = G_{62} = - C_{26}^{-1} \quad (5.39)$$

$$G_{27} = G_{72} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.40)$$

$$G_{28} = G_{82} = - C_{28}^{-1} \quad (5.41)$$

$$G_{34} = G_{43} = - C_{34}^{-1} \quad (5.42)$$

$$G_{35} = G_{53} = - C_{35}^{-1} \quad (5.43)$$

$$G_{36} = G_{63} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.44)$$

$$G_{37} = G_{73} = - C_{37}^{-1} \quad (5.45)$$

$$G_{38} = G_{83} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.46)$$

$$G_{45} = G_{54} = - C_{45}^{-1} \quad (5.47)$$

$$G_{46} = G_{64} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.48)$$

$$G_{47} = G_{74} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.49)$$

$$G_{48} = G_{84} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.50)$$

$$G_{56} = G_{65} = - C_{56}^{-1} \quad (5.51)$$

$$G_{57} = G_{75} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.52)$$

$$G_{58} = G_{85} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.53)$$

$$G_{67} = G_{76} = - C_{67}^{-1} \quad (5.54)$$

$$G_{68} = G_{86} = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0] \quad (5.55)$$

$$G_{78} = G_{87} = -C_{78}^{-1} \quad (5.56)$$

G matrisi bu işlemlerin sonunda 24 x 24 boyutunda bir matris olur. B vektörü ise aşağıdaki şekilde hesaplanır.

$$B_1 = C_{01}^{-1} * \bar{D}_{01} + C_{12}^{-1} * \bar{D}_{12} + C_{16}^{-1} * \bar{D}_{16} + C_{18}^{-1} * \bar{D}_{18} \quad (5.57)$$

$$B_2 = -C_{12}^{-1} * \bar{D}_{12} + C_{23}^{-1} * \bar{D}_{23} + C_{26}^{-1} * \bar{D}_{26} + C_{28}^{-1} * \bar{D}_{28} \quad (5.58)$$

$$B_3 = -C_{23}^{-1} * \bar{D}_{23} + C_{34}^{-1} * \bar{D}_{34} + C_{35}^{-1} * \bar{D}_{35} + C_{37}^{-1} * \bar{D}_{37} \quad (5.59)$$

$$B_4 = -C_{34}^{-1} * \bar{D}_{34} + C_{45}^{-1} * \bar{D}_{45} \quad (5.60)$$

$$B_5 = -C_{35}^{-1} * \bar{D}_{35} - C_{45}^{-1} * \bar{D}_{45} + C_{56}^{-1} * \bar{D}_{56} \quad (5.61)$$

$$B_6 = -C_{16}^{-1} * \bar{D}_{16} - C_{26}^{-1} * \bar{D}_{26} - C_{56}^{-1} * \bar{D}_{56} + C_{67}^{-1} * \bar{D}_{67} \quad (5.62)$$

$$B_7 = -C_{37}^{-1} * \bar{D}_{37} - C_{67}^{-1} * \bar{D}_{67} + C_{78}^{-1} * \bar{D}_{78} \quad (5.63)$$

$$B_8 = C_{08}^{-1} * \bar{D}_{08} - C_{18}^{-1} * \bar{D}_{18} - C_{28}^{-1} * \bar{D}_{28} - C_{78}^{-1} * \bar{D}_{78} \quad (5.64)$$

Elde edilen G ve B kullanılarak durum vektörü X ve kovaryans C_X 4.9 ve 4.10 denklemlerine göre hesaplanır. X vektörü elde edildikten sonra pozisyon vektörlerinde düzeltme işlemi yapılır. Bunun için elde edilen X vektörünün ilgili elemanları -1, 1 veya 0 değerlerinden oluşan bir katsayı matrisi ile çarpılır ve sonuç eski pozisyon vektöründen çıkarılır. Aşağıda ilk poz için yapılan ilk hesaplama örneği vardır .

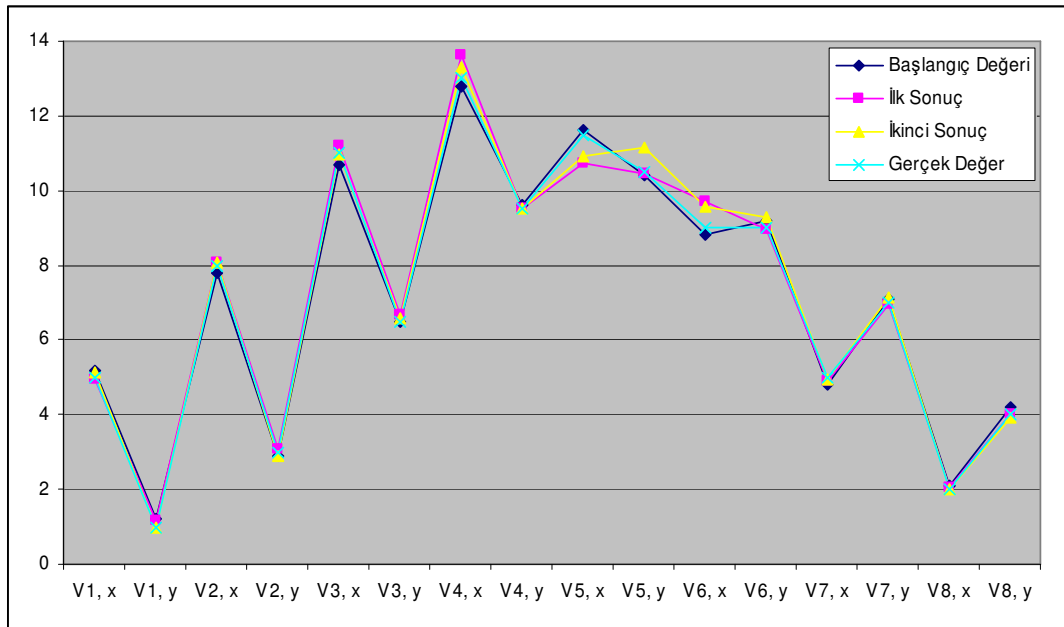
$$X_1 = \begin{bmatrix} X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} -0.0866 \\ -0.0821 \\ 0.0674 \end{bmatrix} \quad (5.64)$$

$$H_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.65)$$

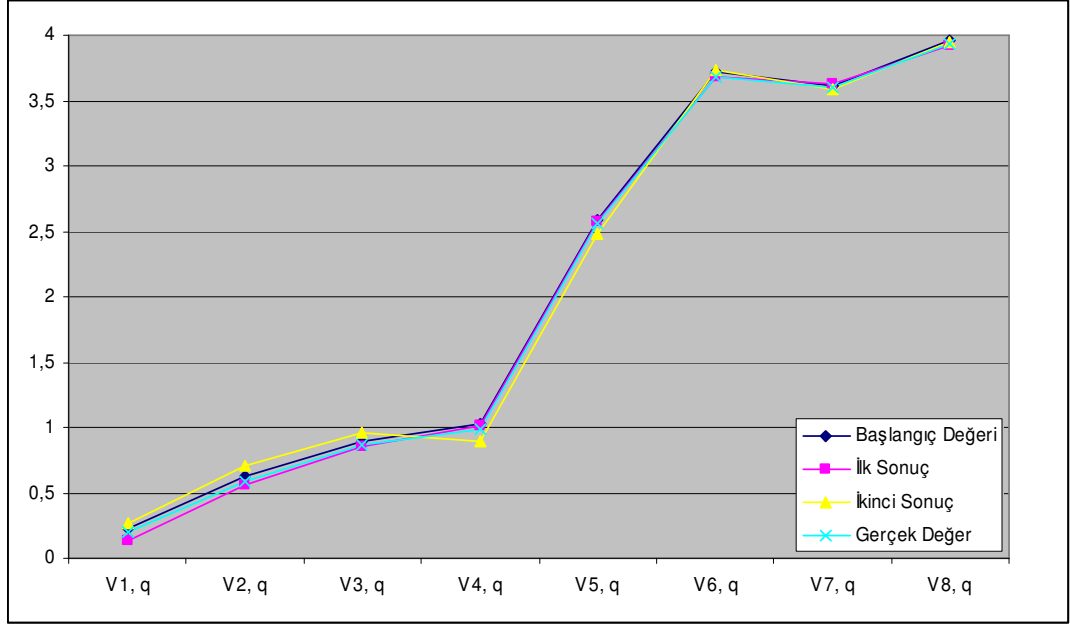
$$V_1 = \bar{V}_1 - H_1 * X_1 = \begin{bmatrix} 5.1134 \\ 1.1179 \\ 0.1594 \end{bmatrix} \quad (5.66)$$

5.4. Zayıf Bağlantılardan Oluşan Sistem için Sonuçlar

G ve B ile oluşturulan sistemde güçlü bağlantılar olmadığı varsayıp bu terimler hesaba katılmazsa algoritma sadece zayıf bağlantılardan yola çıkarak işlem yapmış olur. Bu şekilde elde edilmiş pozisyonlar ve kaçınıcı iterasyon sonunda ortaya çıktığı Şekil 5.4'te ve Şekil 5.5'te yer almaktadır.

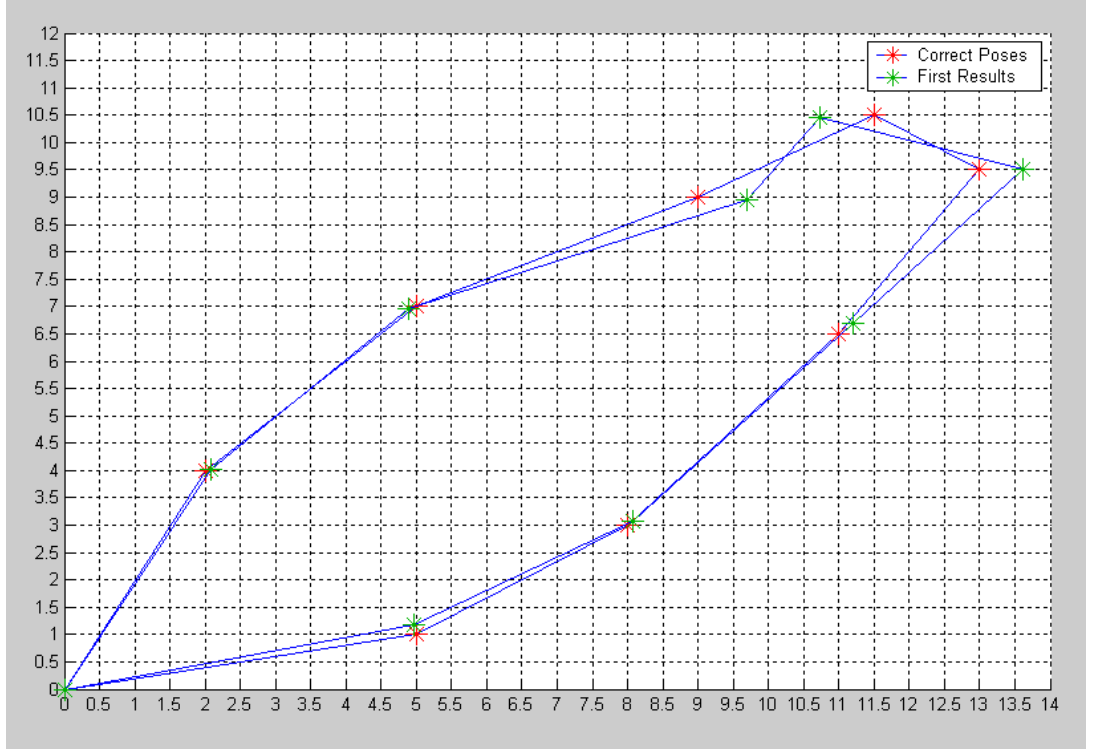


Şekil 5.4: Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Pozisyonlar)

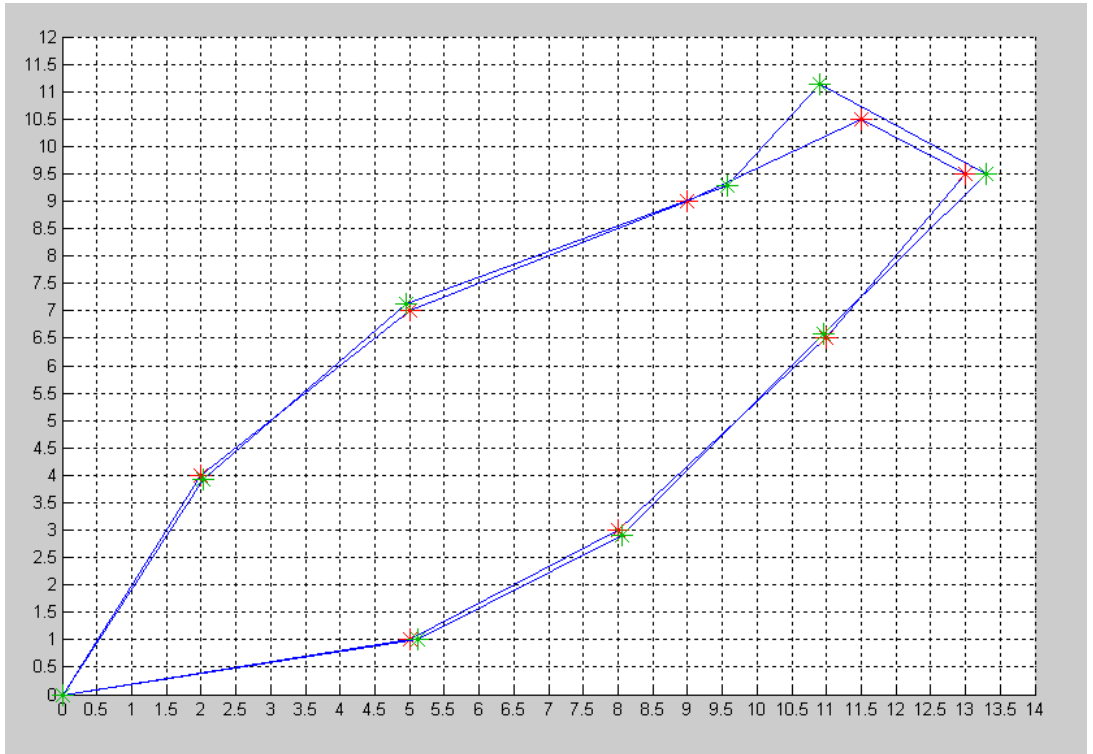


Şekil 5.5: Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Oryantasyonlar)

Bu şekilden de görüldüğü gibi başlangıç pozisyonlarından gerçek pozisyonlara doğru bir yakınsama vardır. Bu noktalar arasında farklılık göstermektedir. İlk iterasyon sonuçları Şekil 5.6'dan , ikinci iterasyon sonuçları de Şekil 5.7'den görsel olarak takip edilebilir. Şekil 5.6 incelendiğinde V_1 , V_2 , V_3 , V_7 ve V_8 pozisyonlarındaki yakınsamanın çok yüksek olduğu gözlenir. Ancak V_4 , V_5 ve V_6 pozları arasındaki geçişlerde araç yörüngesinde bozulma bulunmaktadır. Şekil 5.7 incelendiğinde ise bir önceki iterasyona göre V_4 pozisyonunda da yakınsama olduğunu ve V_1 , V_2 , V_3 , V_7 ve V_8 pozisyonlarındaki iyi sonuçların devam ettiğini görüyoruz. Ayrıca V_4 , V_5 ve V_6 pozlarını içeren yörüngesinde parçasında da düzelme bulunmaktadır.



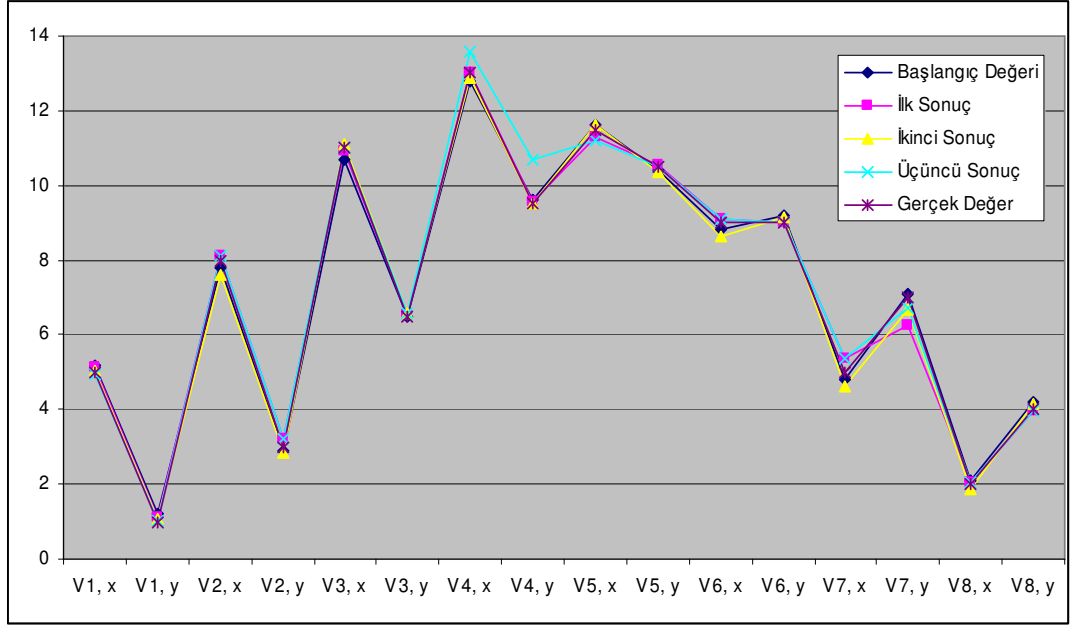
Şekil 5.6: Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması



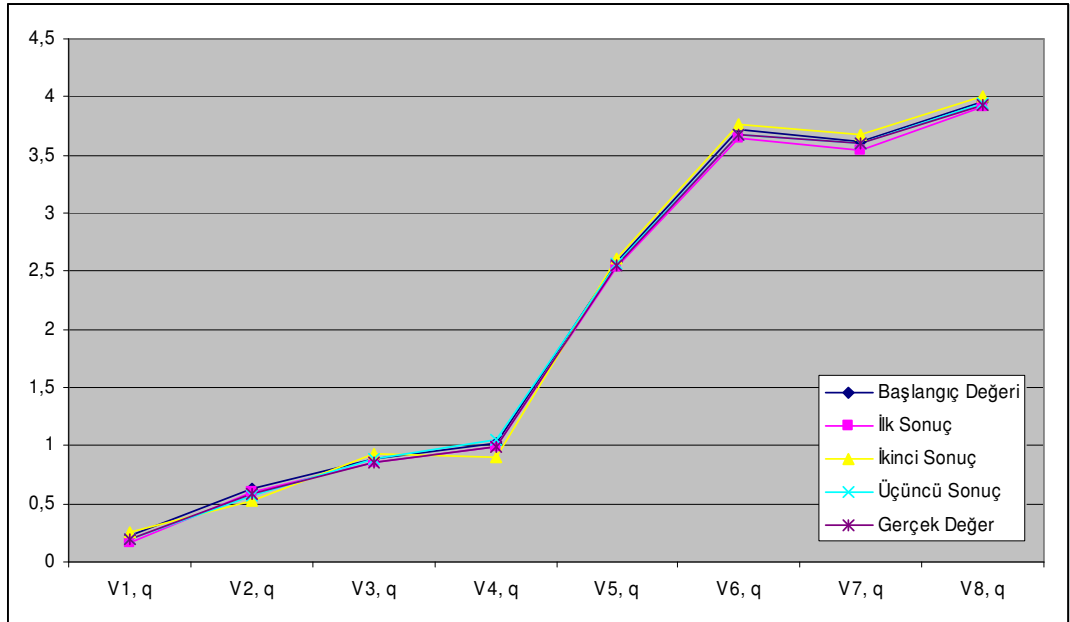
Şekil 5.7: Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İkinci İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması

5.5. Zayıf ve Güçlü Bağlantılardan Oluşan Sistem için Sonuçlar

G ve B ile oluşturulan sistemde güçlü bağlantılar da hesaba katılıp işlem yapıldığında elde edilen iterasyon sonuçları Şekil 5.8 ve 5.9'da yer almaktadır.

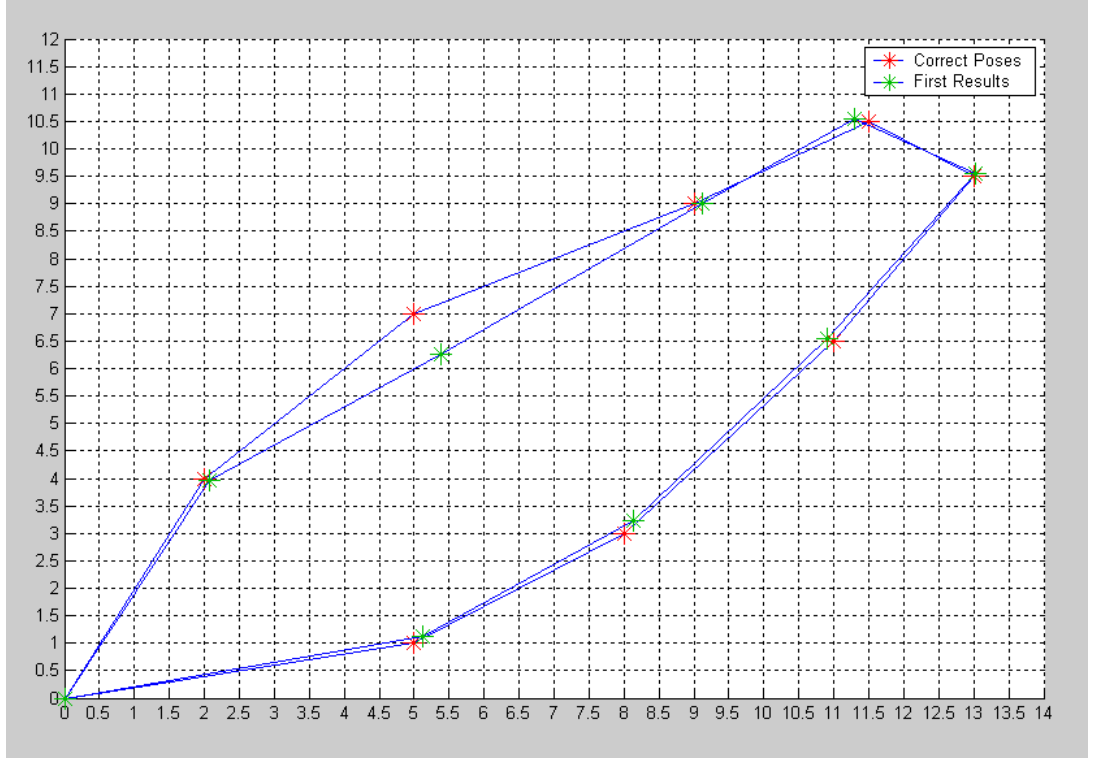


Şekil 5.8: Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Pozisyonlar)

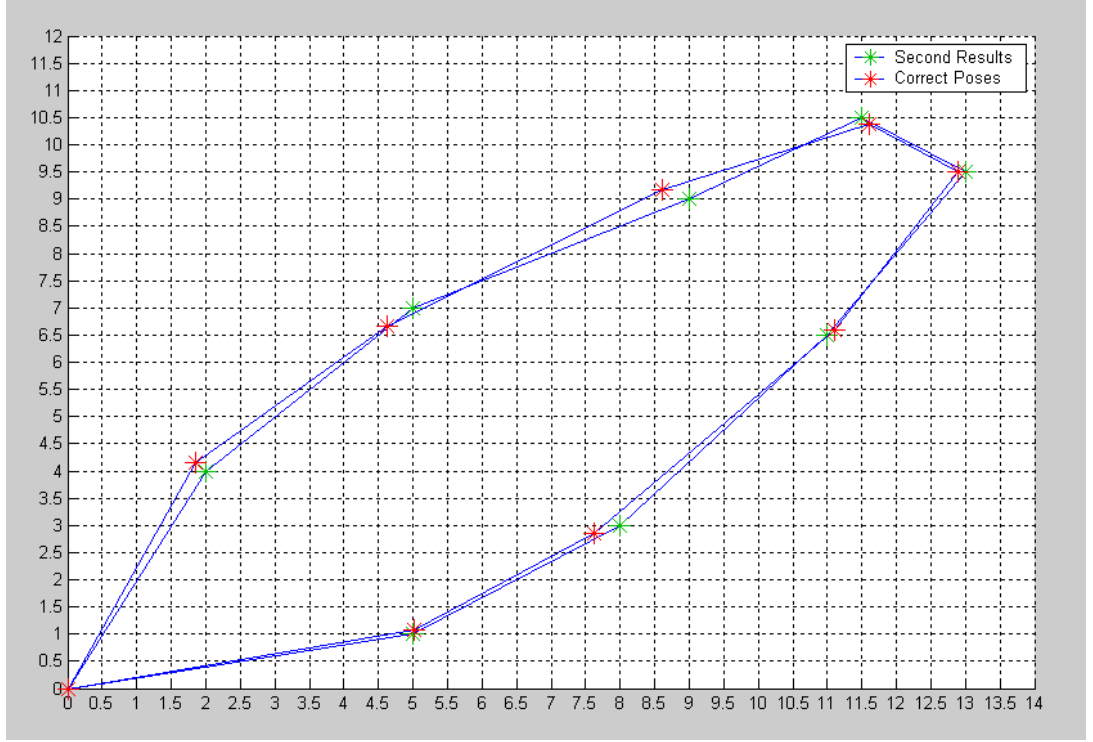


Şekil 5.9: Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları (Oryantasyonlar)

Şekillerden tüm pozisyon bilgileri için bir yakınsama olduğu anlaşılabılır. Bu yakınsamayı daha rahat anlamak için her iterasyon sonucunu gerçek pozisyon değerleri ile kıyaslayan Şekil 5.10, Şekil 5.11 ve Şekil 5.12 incelenmelidir. Şekil 5.10’da ilk iterasyon sonuçları ile gerçek pozisyon değerleri karşılaştırılmaktadır. Bu şekilden V_7 pozisyonu dışındaki bütün pozisyonlarda genel bir yakınsama olduğu anlaşılmaktadır.

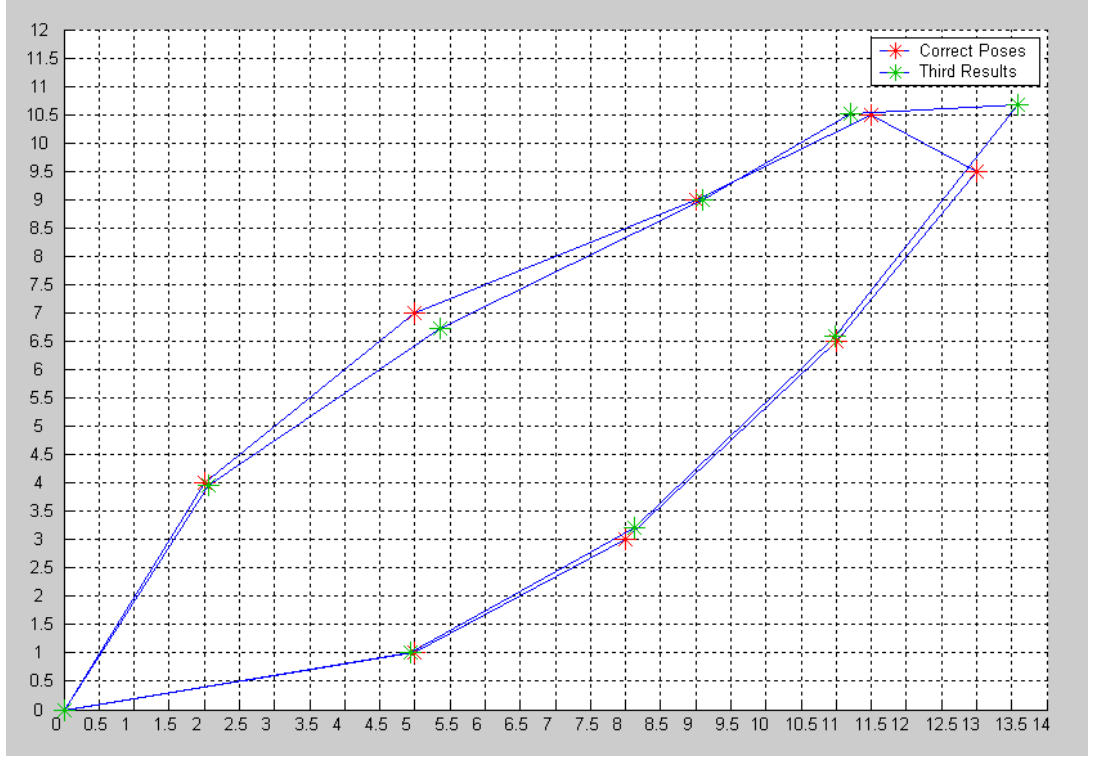


Şekil 5.10: Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan İlk İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması



Şekil 5.11: Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan İkinci İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması

Şekil 5.11 incelendiğinde görülen, V_7 pozisyonu da dahil olmak üzere tüm pozisyonların gerçek pozisyon değerlerine yaklaşmasıdır. Özellikle, V_1 , V_3 , V_4 ve V_5 gerçek pozisyonları ile neredeyse örtüşmektedir. Şekil 5.12'den üçüncü iterasyon sonuçlarına bakıldığında V_1 , V_3 , V_6 ve V_8 pozisyonlarında gerçek değerlere maksimum yakınsama görülmektedir. Bu iterasyonun tek kötü sonucu V_4 pozisyonu ile ilgili olup o da sonraki iterasyonlarda gerçek değerine yaklaşacaktır.



Şekil 5.12: Zayıf Bağlantılardan ve Güçlü Bağlantılardan Oluşan Sistem için Yapılan Üçüncü İterasyon Sonuçları ile Gerçek Pozisyonların Kıyaslanması

6. SONUÇLAR VE TARTIŞMA

Bu tezde amaçlanan yörüngeye dayalı SLAM çözümleri içinde yer alan, topolojik haritama yöntemlerinden GTTH algoritmasını anlamak ve simüle etmektir.

Algoritmanın temelinde ağ yapısının iyi oluşturulması yatmaktadır. Bunun sağlanması için ilk olarak poz ilişkilerinin iyi hesaplanması gerekir. İlk poz ilişkisi odometri ölçümlerinden doğrudan tanımlanabilmektedir. Ancak ölçüm sırasında yapılabilen hatalar göz ardı edilmemelidir. İkinci poz ilişkisi alan tarama çiftlerinin örtüştürülmesi sonucu oluşur. Bir ağ yapısında bu tip poz ilişkilerinin mümkün olduğunca çok tanımlanması odometri hatalarının etkisini azaltmaktadır. Poz ilişkileri tanımlandıktan sonra graf teoriden yola çıkılarak kurulan sistem ardışıl adımlarla kestirilir. Kestirim aşamasında ilişkilerden yola çıkılan sistem matrisi ve sistem vektörü hesaplanmaktadır. Bu hesaplamalardan elde edilen yeni poz kestirimlerinin gerçek pozlara yakınsadığı gözlenmelidir.

Simülasyon senaryosunda iki çeşit örnek vardır. İlk örnekte sadece zayıf ilişkilerden oluşan bir sistem, ikinci örnekte ise hem zayıf hem de güçlü ilişkilerden oluşan bir sistem ele alınmıştır. Elde edilen sonuçlar hem liste olarak hem de görsel olarak sunulmuştur. İki simülasyondan da elde edilen sonuçlar pozisyonların gerçek değerlerine yakınsadığını göstermektedir. Ancak güçlü bağların sisteme eklendiği simülasyonda sadece zayıf bağlardan oluşan simülasyona göre yörünge şeklinin korunmasının daha başarılı olduğu gözlenmiştir.

Tezin simülasyon çalışmalarından yola çıkılarak GTTH yöntemini gerçek zamanlı bir uygulamada test etmek bu yöndeki çalışmaların bir sonraki adımı olmalıdır. Gerçek bir odometri sensörü ile çalışmak odometri verilerinin hatalarının sonuçlarının daha doğru gözlenmesi açısından faydalı olacaktır. Ayrıca lazer sensörü ile yapılan alan tarama sonuçlarının örtüştürülmesi ve karşılıklı noktaların bu örtüşmeler sonucu hesaplanması işlemi gerçek dünyada çalışıldığı zaman daha anlamlı hale gelecektir. Simülasyon sonuçlarından daha hızlı bir yakınsama süreci elde edilebilir.

KAYNAKLAR

- [1] **Bailey, T. ve Durrant-Whyte, H.F.**, 2006. Simultaneous Localization and Mapping (SLAM): Part I, *IEEE Robotics and Automation Magazine*, **June 2006**, 99-108.
- [2] **Bailey, T. ve Durrant-Whyte, H.F.**, 2006. Simultaneous Localization and Mapping (SLAM): Part II, *IEEE Robotics and Automation Magazine*, **September 2006**, 108-117.
- [3] **Eustice, R.M., Singh, H. ve Loenard, J.J.**, 2006. Exactly Sparse Delayed-State Filters for View-Based SLAM, *IEEE Transactions On Robotics*, *December 2006*, **22, 6**, 1100-1112.
- [4] **Dellaert, F. ve Kaess, M.**, 2006. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing, *The International Journal of Robotics Research*, *December 2006*, **25, 12**, 1181-1203.
- [5] **Newman, P., Cole D. ve Ho, K.**, 2006. Outdoor SLAM Using Visual Appearance and Laser Ranging, *Proceedings of International Conference Robotics Automation International Journal of Robotics Research*, 2006.
- [6] **Lu, M., ve Milios E.**, 1997. Globally Consistent Range Scan Alignment for Environment Mapping, *Autonomous Robots*, *April 1997*, **4, 5**, 333-349.
- [7] **Gutmann, J. S., ve Konolige K.**, 1999. Incremental Mapping of Large Cyclic Environments, *Proceedings of IEEE International Symposium Computational Intelligence Robotics Automation*, 1999, 318-325.
- [8] **Lu, M., ve Milios E.**, 1997. Robot Pose Estimation in Unknown Environments By Matching 2D Range Scans, *Journal of Intelligent and Robotic Systems*, *1997, 4* , 249-275.
- [9] **Newman, P.M.**, 2006. EKF Based Navigation and SLAM, *SLAM Summer School 2006 Background Materials and Notes* , Oxford University, Oxford.
- [10] **SLAM Summer School**, 2002. Lecture Notes, Presentation, Practical Exercises and Codes

- [11] **SLAM Summer School**, 2004. Lecture Notes, Presentation, Practical Exercises and Codes
- [12] **SLAM Summer School**, 2006. Lecture Notes, Presentation, Practical Exercises and Codes

EK A MATLAB KODLARI – ZAYIF BAĞLAR İÇİN G MATRİSİ VE B VEKTÖRÜ

G11 = inv(C01) + inv(C12);
G22 = inv(C12) + inv(C23);
G33 = inv(C23) + inv(C34);
G44 = inv(C34) + inv(C45);
G55 = inv(C45) + inv(C56);
G66 = inv(C56) + inv(C67);
G77 = inv(C67) + inv(C78);
G88 = inv(C78) + inv(C80);

G12 = -inv(C12);
G13 = [0 0 0 ; 0 0 0 ; 0 0 0];
G14 = [0 0 0 ; 0 0 0 ; 0 0 0];
G15 = [0 0 0 ; 0 0 0 ; 0 0 0];
G16 = [0 0 0 ; 0 0 0 ; 0 0 0];
G17 = [0 0 0 ; 0 0 0 ; 0 0 0];
G18 = [0 0 0 ; 0 0 0 ; 0 0 0];

G21 = -inv(C12);
G23 = -inv(C23);
G24 = [0 0 0 ; 0 0 0 ; 0 0 0];
G25 = [0 0 0 ; 0 0 0 ; 0 0 0];
G26 = [0 0 0 ; 0 0 0 ; 0 0 0];
G27 = [0 0 0 ; 0 0 0 ; 0 0 0];
G28 = [0 0 0 ; 0 0 0 ; 0 0 0];

G31 = [0 0 0 ; 0 0 0 ; 0 0 0];
G32 = -inv(C23);
G34 = -inv(C34);
G35 = [0 0 0 ; 0 0 0 ; 0 0 0];
G36 = [0 0 0 ; 0 0 0 ; 0 0 0];
G37 = [0 0 0 ; 0 0 0 ; 0 0 0];
G38 = [0 0 0 ; 0 0 0 ; 0 0 0];

$G41 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G42 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G43 = -\text{inv}(C34);$
 $G45 = -\text{inv}(C45);$
 $G46 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G47 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G48 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G51 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G52 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G53 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G54 = -\text{inv}(C45);$
 $G56 = -\text{inv}(C56);$
 $G57 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G58 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G61 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G62 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G63 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G64 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G65 = -\text{inv}(C56);$
 $G67 = -\text{inv}(C67);$
 $G68 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G71 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G72 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G73 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G74 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G75 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G76 = -\text{inv}(C67);$
 $G78 = -\text{inv}(C78);$

$G81 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G82 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$$G83 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G84 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G85 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G86 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G87 = -\text{inv}(C78);$$

$$G = [G11 \ G12 \ G13 \ G14 \ G15 \ G16 \ G17 \ G18 ; G21 \ G22 \ G23 \ G24 \ G25 \ G26 \ G27 \ G28 ; \\ G31 \ G32 \ G33 \ G34 \ G35 \ G36 \ G37 \ G38 ; G41 \ G42 \ G43 \ G44 \ G45 \ G46 \ G47 \ G48 ; G51 \\ G52 \ G53 \ G54 \ G55 \ G56 \ G57 \ G58 ; G61 \ G62 \ G63 \ G64 \ G65 \ G66 \ G67 \ G68 ; G71 \ G72 \\ G73 \ G74 \ G75 \ G76 \ G77 \ G78 ; G81 \ G82 \ G83 \ G84 \ G85 \ G86 \ G87 \ G88];$$

$$B1 = \text{inv}(C01)*D01 + \text{inv}(C12)*D12;$$

$$B2 = -\text{inv}(C12)*D12 + \text{inv}(C23)*D23;$$

$$B3 = -\text{inv}(C23)*D23 + \text{inv}(C34)*D34;$$

$$B4 = -\text{inv}(C34)*D34 + \text{inv}(C45)*D45;$$

$$B5 = -\text{inv}(C45)*D45 + \text{inv}(C56)*D56;$$

$$B6 = -\text{inv}(C56)*D56 + \text{inv}(C67)*D67;$$

$$B7 = -\text{inv}(C67)*D67 + \text{inv}(C78)*D78;$$

$$B8 = \text{inv}(C08)*D08 - \text{inv}(C78)*D78;$$

$$B = [B1 ; B2 ; B3 ; B4 ; B5 ; B6 ; B7 ; B8];$$

EK B MATLAB KODLARI – GÜÇLÜ VE ZAYIF BAĞLAR İÇİN G MATRİSİ VE B VEKTÖRÜ

$$G11 = \text{inv}(C01) + \text{inv}(C12) + \text{inv}(C16) + \text{inv}(C18);$$

$$G22 = \text{inv}(C12) + \text{inv}(C23) + \text{inv}(C26) + \text{inv}(C28);$$

$$G33 = \text{inv}(C23) + \text{inv}(C34) + \text{inv}(C35) + \text{inv}(C37);$$

$$G44 = \text{inv}(C34) + \text{inv}(C45);$$

$$G55 = \text{inv}(C35) + \text{inv}(C45) + \text{inv}(C56);$$

$$G66 = \text{inv}(C16) + \text{inv}(C26) + \text{inv}(C56) + \text{inv}(C67);$$

$$G77 = \text{inv}(C37) + \text{inv}(C67) + \text{inv}(C78);$$

$$G88 = \text{inv}(C18) + \text{inv}(C28) + \text{inv}(C78) + \text{inv}(C80);$$

$$G12 = -\text{inv}(C12);$$

$$G13 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G14 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G15 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G16 = -\text{inv}(C16);$$

$$G17 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G18 = -\text{inv}(C18);$$

$$G21 = -\text{inv}(C12);$$

$$G23 = -\text{inv}(C23);$$

$$G24 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G25 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G26 = -\text{inv}(C26);$$

$$G27 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G28 = -\text{inv}(C28);$$

$$G31 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G32 = -\text{inv}(C23);$$

$$G34 = -\text{inv}(C34);$$

$$G35 = -\text{inv}(C35);$$

$$G36 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G37 = -\text{inv}(C37);$$

$$G38 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$G41 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G42 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G43 = -\text{inv}(C34);$
 $G45 = -\text{inv}(C45);$
 $G46 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G47 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G48 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G51 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G52 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G53 = -\text{inv}(C35);$
 $G54 = -\text{inv}(C45);$
 $G56 = -\text{inv}(C56);$
 $G57 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G58 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G61 = -\text{inv}(C16);$
 $G62 = -\text{inv}(C26);$
 $G63 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G64 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G65 = -\text{inv}(C56);$
 $G67 = -\text{inv}(C67);$
 $G68 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$

$G71 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G72 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G73 = -\text{inv}(C37);$
 $G74 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G75 = [0\ 0\ 0; 0\ 0\ 0; 0\ 0\ 0];$
 $G76 = -\text{inv}(C67);$
 $G78 = -\text{inv}(C78);$

$G81 = -\text{inv}(C18);$
 $G82 = -\text{inv}(C28);$

$$G83 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G84 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G85 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G86 = [0 \ 0 \ 0 ; 0 \ 0 \ 0 ; 0 \ 0 \ 0];$$

$$G87 = -\text{inv}(C78);$$

$$G = [G11 \ G12 \ G13 \ G14 \ G15 \ G16 \ G17 \ G18 ; G21 \ G22 \ G23 \ G24 \ G25 \ G26 \ G27 \ G28 ; \\ G31 \ G32 \ G33 \ G34 \ G35 \ G36 \ G37 \ G38 ; G41 \ G42 \ G43 \ G44 \ G45 \ G46 \ G47 \ G48 ; G51 \\ G52 \ G53 \ G54 \ G55 \ G56 \ G57 \ G58 ; G61 \ G62 \ G63 \ G64 \ G65 \ G66 \ G67 \ G68 ; G71 \ G72 \\ G73 \ G74 \ G75 \ G76 \ G77 \ G78 ; G81 \ G82 \ G83 \ G84 \ G85 \ G86 \ G87 \ G88];$$

$$B1 = \text{inv}(C01)*D01 + \text{inv}(C12)*D12 + \text{inv}(C16)*D16 + \text{inv}(C18)*D18;$$

$$B2 = -\text{inv}(C12)*D12 + \text{inv}(C23)*D23 + \text{inv}(C26)*D26 + \text{inv}(C28)*D28;$$

$$B3 = -\text{inv}(C23)*D23 + \text{inv}(C34)*D34 + \text{inv}(C35)*D35 + \text{inv}(C37)*D37;$$

$$B4 = -\text{inv}(C34)*D34 + \text{inv}(C45)*D45;$$

$$B5 = -\text{inv}(C35)*D35 - \text{inv}(C45)*D45 + \text{inv}(C56)*D56;$$

$$B6 = -\text{inv}(C16)*D16 - \text{inv}(C26)*D26 - \text{inv}(C56)*D56 + \text{inv}(C67)*D67;$$

$$B7 = -\text{inv}(C37)*D37 - \text{inv}(C67)*D67 + \text{inv}(C78)*D78;$$

$$B8 = \text{inv}(C08)*D08 - \text{inv}(C18)*D18 - \text{inv}(C28)*D28 - \text{inv}(C78)*D78;$$

$$B = [B1 ; B2 ; B3 ; B4 ; B5 ; B6 ; B7 ; B8]$$

EK C MATLAB KODLARI –ZAYIF BAĞLAR İÇİN İTERASYON İŞLEMLERİ VE SONUÇLAR

% First Results:

X = [0.2365 ; -0.0249 ; 0.0944 ; -0.2811 ; -0.1722 ; 0.0616 ; -0.4975 ; -0.1924 ;
0.0393 ; -0.8102 ; -0.0811 ; 0.0078 ; -0.8766 ; -0.0668 ; -0.0133 ; -0.8927 ; 0.2435 ; -
0.0448 ; -0.0899 ; -0.1491 ; 0.0203 ; 0.0249 ; -0.1789 ; 0.0335];

X1 = [X(1) ; X(2) ; X(3)];

X2 = [X(4) ; X(5) ; X(6)];

X3 = [X(7) ; X(8) ; X(9)];

X4 = [X(10) ; X(11) ; X(12)];

X5 = [X(13) ; X(14) ; X(15)];

X6 = [X(16) ; X(17) ; X(18)];

X7 = [X(19) ; X(20) ; X(21)];

X8 = [X(22) ; X(23) ; X(24)];

H1 = [1 0 0 ; 0 -1 0 ; 0 0 1];

H2 = [1 0 0 ; 0 1 0 ; 0 0 1];

H3 = [1 0 0 ; 0 1 0 ; 0 0 1];

H4 = [1 0 0 ; 0 -1 0 ; 0 0 1];

H5 = [-1 0 0 ; 0 1 0 ; 0 0 -1];

H6 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H7 = [1 0 0 ; 0 -1 0 ; 0 0 -1];

H8 = [1 0 0 ; 0 -1 0 ; 0 0 1];

%Poses after First Iteration

V1 = [4.9635 ; 1.1751 ; 0.1324];

V2 = [8.0811 ; 3.0722 ; 0.5664];

V3 = [11.1975 ; 6.6924 ; 0.8529];

V4 = [13.6102 ; 9.5189 ; 1.0150];

V5 = [10.7234 ; 10.4668 ; 2.5703];

V6 = [9.6927 ; 8.9565 ; 3.6772];

V7 = [4.8899 ; 6.9509 ; 3.6303];

V8 = [2.0751 ; 4.0211 ; 3.9235];

% Second Results

$X = [0.1579 ; -0.1751 ; 0.1407 ; 0.0268 ; -0.1664 ; 0.1451 ; -0.2352 ; 0.1174 ; 0.1121 ; -0.3047 ; 0.0198 ; 0.1169 ; -0.1805 ; -0.6704 ; 0.0848 ; -0.1153 ; -0.3269 ; 0.0602 ; -0.0694 ; -0.1792 ; 0.0452 ; -0.0477 ; -0.0884 ; 0.0266]$;

$X1 = [X(1) ; X(2) ; X(3)]$;

$X2 = [X(4) ; X(5) ; X(6)]$;

$X3 = [X(7) ; X(8) ; X(9)]$;

$X4 = [X(10) ; X(11) ; X(12)]$;

$X5 = [X(13) ; X(14) ; X(15)]$;

$X6 = [X(16) ; X(17) ; X(18)]$;

$X7 = [X(19) ; X(20) ; X(21)]$;

$X8 = [X(22) ; X(23) ; X(24)]$;

$H1 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1]$;

$H2 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1]$;

$H3 = [-1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1]$;

$H4 = [-1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 1]$;

$H5 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 1]$;

$H6 = [-1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1]$;

$H7 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 1]$;

$H8 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1]$;

$V1 = [5.1214 ; 1.0000 ; 0.2731]$;

$V2 = [8.0543 ; 2.9058 ; 0.7115]$;

$V3 = [10.9623 ; 6.5750 ; 0.9650]$;

$V4 = [13.3055 ; 9.4991 ; 0.8981]$;

$V5 = [10.9039 ; 11.1372 ; 2.4855]$;

$V6 = [9.5774 ; 9.2834 ; 3.7374]$;

$V7 = [4.9593 ; 7.1301 ; 3.5851]$;

$V8 = [2.0274 ; 3.9327 ; 3.9501]$;

Tablo C.1: Sadece Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları

Pozisyon Adı	Başlangıç	İlk Sonuç	İkinci Sonuç	Gerçek Değer
--------------	-----------	-----------	--------------	--------------

	Değeri			
V_1, x	5.2000	4.9635	5.1214	5.0000
V_1, y	1.2000	1.1751	1.0000	1.0000
V_1, θ	0.2268	0.1324	0.2731	0.1974
V_2, x	7.8000	8.0811	8.0543	8.0000
V_2, y	2.9000	3.0722	2.9058	3.0000
V_2, θ	0.6280	0.5664	0.7115	0.5880
V_3, x	10.7	11.1975	10.9623	11.0000
V_3, y	6.5000	6.6924	6.5750	6.5000
V_3, θ	0.8922	0.8529	0.9650	0.8622
V_4, x	12.8000	13.6102	13.3055	13.0000
V_4, y	9.6000	9.5189	9.4991	9.5000
V_4, θ	1.0228	1.0150	0.8981	0.9828
V_5, x	11.6000	10.7234	10.9039	11.5000
V_5, y	10.4000	10.4668	11.1372	10.5000
V_5, θ	2.5836	2.5703	2.4855	2.5536
V_6, x	8.8000	9.6927	9.5774	9.0000
V_6, y	9.2000	8.9565	9.2834	9.0000
V_6, θ	3.7220	3.6772	3.7374	3.6820
V_7, x	4.8000	4.8899	4.9593	5.0000
V_7, y	7.1000	6.9509	7.1301	7.0000
V_7, θ	3.6100	3.6303	3.5851	3.6052
V_8, x	2.1000	2.0751	2.0274	2.0000

V_8, y	4.2000	4.0211	3.9327	4.0000
V_8, θ	3.9570	3.9235	3.9501	3.9270

EK D MATLAB KODLARI –ZAYIF BAĞLAR VE GÜÇLÜ BAĞLAR İÇİN İTERASYON İŞLEMLERİ VE SONUÇLAR

% The First Results

X1 = [-0.0866 ; -0.0821 ; 0.0674 ; 0.3292 ; 0.3221 ; 0.0309 ; 0.2099 ; 0.0356 ; 0.0434 ; 0.2215 ; 0.0401 ; 0.0394 ; -0.3109 ; 0.1495 ; 0.0433 ; 0.3118 ; -0.2016 ; 0.0702 ; -0.5756 ; 0.8405 ; -0.0687 ; 0.0320 ; -0.2293 ; 0.0397];

X1 = [X(1) ; X(2) ; X(3)];

X2 = [X(4) ; X(5) ; X(6)];

X3 = [X(7) ; X(8) ; X(9)];

X4 = [X(10) ; X(11) ; X(12)];

X5 = [X(13) ; X(14) ; X(15)];

X6 = [X(16) ; X(17) ; X(18)];

X7 = [X(19) ; X(20) ; X(21)];

X8 = [X(22) ; X(23) ; X(24)];

H1 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H2 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H3 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H4 = [-1 0 0 ; 0 1 0 ; 0 0 1];

H5 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H6 = [-1 0 0 ; 0 -1 0 ; 0 0 1];

H7 = [1 0 0 ; 0 1 0 ; 0 0 -1];

H8 = [1 0 0 ; 0 -1 0 ; 0 0 1];

% The Poses after the First Correction:

V1 = V1e - H1 * X1;

V2 = V2e - H2 * X2;

V3 = V3e - H3 * X3;

V4 = V4e - H4 * X4;

V5 = V5e - H5 * X5;

V6 = V6e - H6 * X6;

V7 = V7e - H7 * X7;

V8 = V8e - H8 * X8;

$V1 = [5.1134 ; 1.1179 ; 0.1594];$
 $V2 = [8.1292 ; 3.2221 ; 0.5971];$
 $V3 = [10.9099 ; 6.5356 ; 0.8488];$
 $V4 = [13.0215 ; 9.5599 ; 0.9834];$
 $V5 = [11.2891 ; 10.5495 ; 2.5403];$
 $V6 = [9.1118 ; 8.9984 ; 3.6518];$
 $V7 = [5.3756 ; 6.2595 ; 3.5413];$
 $V8 = [2.0680 ; 3.9707 ; 3.9173];$

% The second Calculation:

$X = [0.0935 ; -0.0478 ; 0.0979 ; 0.5146 ; 0.3671 ; 0.0752 ; 0.1931 ; 0.0586 ; 0.0747 ;$
 $0.1241 ; -0.0617 ; 0.0783 ; -0.3277 ; 0.1725 ; 0.0746 ; 0.4933 ; -0.1667 ; 0.1106 ;$
 $0.7445 ; -0.3925 ; 0.1405 ; 0.2234 ; -0.1905 ; 0.0838];$

$X1 = [X(1) ; X(2) ; X(3)];$
 $X2 = [X(4) ; X(5) ; X(6)];$
 $X3 = [X(7) ; X(8) ; X(9)];$
 $X4 = [X(10) ; X(11) ; X(12)];$
 $X5 = [X(13) ; X(14) ; X(15)];$
 $X6 = [X(16) ; X(17) ; X(18)];$
 $X7 = [X(19) ; X(20) ; X(21)];$
 $X8 = [X(22) ; X(23) ; X(24)];$
 $H1 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1];$
 $H2 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 1];$
 $H3 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1];$
 $H4 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$
 $H5 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1];$
 $H6 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1];$
 $H7 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1];$
 $H8 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1];$

% The Poses after the Second Correction:

$V1 = V1e - H1 * X1;$
 $V2 = V2e - H2 * X2;$
 $V3 = V3e - H3 * X3;$

$$V4 = V4e - H4 * X4;$$

$$V5 = V5e - H5 * X5;$$

$$V6 = V6e - H6 * X6;$$

$$V7 = V7e - H7 * X7;$$

$$V8 = V8e - H8 * X8;$$

$$V1 = [5.0199 ; 1.0701 ; 0.2573];$$

$$V2 = [7.6146 ; 2.8550 ; 0.5219];$$

$$V3 = [11.1030 ; 6.5942 ; 0.9235];$$

$$V4 = [12.8974 ; 9.4982 ; 0.9051];$$

$$V5 = [11.6168 ; 10.3770 ; 2.6149];$$

$$V6 = [8.6185 ; 9.1651 ; 3.7624];$$

$$V7 = [4.6311 ; 6.6520 ; 3.6818];$$

$$V8 = [1.8446 ; 4.1612 ; 4.0011];$$

% The third Calculation

$$X = [0.0823 ; -0.0527 ; 0.0682 ; 0.5039 ; 0.3624 ; 0.0457 ; 0.1123 ; 0.0136 ; 0.0440 ; 0.6973 ; -1.1837 ; 0.1406 ; -0.4085 ; 0.1275 ; 0.0439 ; 0.4822 ; -0.1716 ; 0.0810 ; 0.7207 ; 0.0762 ; 0.0816 ; 0.2126 ; -0.1954 ; 0.0543];$$

$$X1 = [X(1) ; X(2) ; X(3)];$$

$$X2 = [X(4) ; X(5) ; X(6)];$$

$$X3 = [X(7) ; X(8) ; X(9)];$$

$$X4 = [X(10) ; X(11) ; X(12)];$$

$$X5 = [X(13) ; X(14) ; X(15)];$$

$$X6 = [X(16) ; X(17) ; X(18)];$$

$$X7 = [X(19) ; X(20) ; X(21)];$$

$$X8 = [X(22) ; X(23) ; X(24)];$$

$$H1 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$$

$$H2 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ -1];$$

$$H3 = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 1];$$

$$H4 = [-1 \ 0 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ -1];$$

$$H5 = [1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$$

$$H6 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$$

$$H7 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$$

$$H8 = [-1 \ 0 \ 0 ; 0 \ -1 \ 0 ; 0 \ 0 \ 1];$$

% The Poses after the Third Correction:

$$V1 = V1e - H1 * X1;$$

$$V2 = V2e - H2 * X2;$$

$$V3 = V3e - H3 * X3;$$

$$V4 = V4e - H4 * X4;$$

$$V5 = V5e - H5 * X5;$$

$$V6 = V6e - H6 * X6;$$

$$V7 = V7e - H7 * X7;$$

$$V8 = V8e - H8 * X8;$$

$$V1 = [4.9376 ; 1.0174 ; 0.1891];$$

$$V2 = [8.1185 ; 3.2174 ; 0.5676];$$

$$V3 = [10.9907 ; 6.5806 ; 0.8795];$$

$$V4 = [13.5947 ; 10.6819 ; 1.0457];$$

$$V5 = [11.2083 ; 10.5045 ; 2.5710];$$

$$V6 = [9.1007 ; 8.9935 ; 3.6814];$$

$$V7 = [5.3518 ; 6.7282 ; 3.6002];$$

$$V8 = [2.0572 ; 3.9658 ; 3.9468];$$

Tablo D.1: Güçlü ve Zayıf Bağlantılardan Oluşan Sistem için Yapılan İterasyon Sonuçları

Pozisyon Adı	Başlangıç Değeri	İlk Sonuç	İkinci Sonuç	Üçüncü Sonuç	Gerçek Değer
V_{1, x}	5.2000	5.1134	5.0199	4.9376	5.0000
V_{1, y}	1.2000	1.1179	1.0701	1.0174	1.0000
V_{1, θ}	0.2268	0.1594	0.2573	0.1891	0.1974
V_{2, x}	7.8000	8.1292	7.6146	8.1185	8.0000
V_{2, y}	2.9000	3.2221	2.8550	3.2174	3.0000
V_{2, θ}	0.6280	0.5971	0.5219	0.5676	0.5880

V₃, x	10.7	10.9099	11.1030	10.9907	11.0000
V₃, y	6.5000	6.5356	6.5942	6.5806	6.5000
V₃, θ	0.8922	0.8488	0.9235	0.8795	0.8622
V₄, x	12.8000	13.0215	12.8974	13.5947	13.0000
V₄, y	9.6000	9.5599	9.4982	10.6819	9.5000
V₄, θ	1.0228	0.9834	0.9051	1.0457	0.9828
V₅, x	11.6000	11.2891	11.6168	11.2083	11.5000
V₅, y	10.4000	10.5495	10.3770	10.5045	10.5000
V₅, θ	2.5836	2.5403	2.6149	2.5710	2.5536
V₆, x	8.8000	9.1118	8.6185	9.1007	9.0000
V₆, y	9.2000	8.9984	9.1651	8.9935	9.0000
V₆, θ	3.7220	3.6518	3.7624	3.6814	3.6820
V₇, x	4.8000	5.3756	4.6311	5.3518	5.0000
V₇, y	7.1000	6.2595	6.6520	6.7282	7.0000
V₇, θ	3.6100	3.5413	3.6818	3.6002	3.6052
V₈, x	2.1000	2.0680	1.8446	2.0572	2.0000
V₈, y	4.2000	3.9707	4.1612	3.9658	4.0000
V₈, θ	3.9570	3.9173	4.0011	3.9468	3.9270

ÖZGEÇMİŞ

Müjde BAKIR, 1982 yılında İstanbul’da doğdu. İlkokulu bitirdikten sonra ortaokul ve liseyi İstanbul Erkek Lisesi’nde okudu. 2001 yılında İstanbul Teknik Üniversitesi Kontrol Mühendisliği programına başladı ve 2005 yılında bu bölümden mezun oldu. Aynı sene , İstanbul Teknik Üniversitesi Kontrol ve Otomasyon Mühendisliği yüksek lisans bölümünde eğitime devam etti. 2006 yılında ABB Sanayi A.Ş.’de Proses Otomasyonu Bölümü’nde Proje Mühendisi olarak başladığı işte halen çalışmaktadır.