

**T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
SOSYAL BİLİMLER ENSTİTÜSÜ**

**EVİRİMSEL HESAPLAMA TEKNİKLERİYLE
HEDEF HAREKET ANALİZİ PROBLEMİ
ÇÖZÜMÜ**

**Levent İNCE
YÜKSEK LİSANS TEZİ
İŞLETME ANABİLİM DALI**

**GEBZE
2007**

**T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
SOSYAL BİLİMLER ENSTİTÜSÜ**

**EVİRİMSEL HESAPLAMA TEKNİKLERİYLE
HEDEF HAREKET ANALİZİ PROBLEMİ
ÇÖZÜMÜ**

**Levent İNCE
YÜKSEK LİSANS TEZİ
İŞLETME ANABİLİM DALI**

**TEZ DANIŞMANI
Doç. Dr. Bülent SEZEN**

**GEBZE
2007**

BU SAYFA BOŞ BIRAKILMIŞTIR

ÖZET

TEZ BAŞLIĞI: Evrimsel Hesaplama Teknikleriyle Hedef Hareket Analizi

Problemi Çözümü

YAZAR ADI: Levent İNCE

Ters problemler, bilim ve matematiğin bir çok alanlarında kullanılan, belirli bir sisteme ait model parametrelerinin gözlemlenen verilerden elde edildiği problemlerdir. Daha sade bir tanımla, cevabın bilindiği ancak sorunun ne olduğunun arandığı problemlerdir. Öte yandan *Hedef Hareket Analizi (HHA)*, özellikle sualtı ortamında hareket eden ses kaynaklarının, sadece ölçülen ses kaynağı yönü bilgisi (kerteriz) kullanılarak kinematik bilgilerinin elde edilmesi işlemidir.

Bu çalışmada, Hedef Hareket Analizi problemi, bir ters problem örneği olarak ele alınmış ve bu problemin çözümü maksadıyla evrimsel hesaplama tekniklerinden *Genetik Algoritmaları* ve bunun yanında *Monte Carlo* denemelerini kullanan yeni bir *Uydurulmuş Alan Sinyal İşleme* yaklaşımı anlatılmıştır.

Bu yaklaşımdaki ana fikir şu şekilde açıklanabilir; Önce sualtı ortamında bir miktar kerteriz ölçümü alınarak kaydedilir. Daha sonra sualtı ortamı taktik resminin rastgele bir şekilde simülasyonunu yapılır. Taktik resim simülasyonu kendi parametrelerini genetik algoritma kontrolünde değiştirerek her seferinde sahte kerteriz bilgileri üretir. Genetik algoritma sayesinde sahte kerteriz bilgileri zaman geçtikçe gerçek kerteriz ölçümlerine yaklaşır ve netice elde edilen sualtı taktik resmi simülasyonu gerçek durumu temsil etmiş olur. Dolayısıyla bu simülasyonu oluşturan parametreler de, gerçek hedef kinematik parametreleridir.

Algoritmayı hayata geçirmek ve yaklaşımın başarısını ve doğruluğunu test etmek amacıyla *TMAGA* adlı bir uygulama yazılımı geliştirilmiştir. *TMAGA* ile gerçekleştirilen deneme sonuçları Monte Carlo simülasyonlarıyla sergilenmiştir.

SUMMARY

TITLE OF THE THESIS: Handling Target Motion Analysis Problem with Evolutionary Computing Techniques

AUTHOR: Levent İNCE

Inverse problems are the kind of problems where the parameters of a model is obtained from the observed data. They have been used in many fields of Mathematics and Science. They are the problems where the results are known instead of questions. On the other hand Target Motion Analysis (TMA) is the process where the kinematics parameters of a moving sound source is determined by using only bearings.

In this work, TMA problem is considered as an inverse problem and new Matched Field Signal processing approach which uses Genetic Algorithm (GA) and Monte Carlo simulation is proposed to establish tracks from bearing-only contacts. The basic idea is to take a number of measurements and run a simulation of underwater tactical situation, then to let the simulation change its parameters until the output matches with the measurement data. When the output of the simulation (i.e. the replica data) matches the real measurement within a predefined degree, it is expected that the simulation resembles the real situation. In this sense, the TMA problem is considered as an optimization issue on a large parameter space and Genetic Algorithm was used to solve it. We developed an application called TMAGA in order to demonstrate the correctness of the algorithm. Monte Carlo simulations demonstrate the results of the experiments conducted with TMAGA.

TEŞEKKÜR

Bana her türlü desteğini esirgemeyen ve bu çalışmanın tamamlanması boyunca benimle birlikte bir takım zorluklara katlanan çok sevgili hayat arkadaşım Devrim'e, sıkıntılı anlarımda verdiği keman konserleriyle moralimi düzelden bir tanecik kızım Ceren'e ve bu tezi yazarken kucağımdan kalkmayarak ve bana bol bol Baby Einstein seyrettirerek çok destek olan biricik bebeğim Alperen'e çok teşekkür ediyorum.

Bu çalışmanın tamamlanmasında her türlü desteğini, hoşgörüsünü benden esirgemeyen, bana her zaman yol gösterici olan ve bu zamana kadar aldığım mühendislik eğitimlerime rağmen bana İşletme bilimi sevgisini ve ilgisini aşılıyarak bu çalışmanın tamamlanmasını sağlayacak motivasyonu oluşturan çok değerli tez danışmanım Yrd.Dç.Dr. Bülent SEZEN'e çok teşekkür ederim.

Gebze Yüksek Teknoloji Enstitüsünde aldığım her derste bana birçok yeni bilgileri öğreten ve bu zamana kadar aldığım mühendislik eğitiminden sonra verdikleri sosyal bilimler eğitimi ile hayata bakış açımı genişleten bütün değerli öğretim üyelerine de ayrıca çok teşekkür ederim.

Yüksek Lisans eğitimimi tamamlamamda bana çok yardımcı olan saygıdeğer mesai arkadaşlarıma gösterdikleri sonsuz hoşgörüden dolayı çok teşekkür ederim.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
İÇİNDEKİLER DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ	x
1. GİRİŞ	1
2. HEDEF HAREKET ANALİZİ (HHA) VE EVRİMSEL HESAPLAMA TEKNİKLERİ	6
2.1. Hedef Hareket Analizi	6
2.2. Evrimsel Hesaplama (Evolutionary Computing), Genetik Algoritmalar	16
2.3. Tmaga (Target Motion Analysis With Genetic Algorithms) Uygulaması	21
2.3.1. Hedef Hareketinin Temsil Edilmesi – Kromozom Yapısı	22
2.3.2. Başlangıç Populasyonunun (Başlangıç Çözüm Kümesi) Yaratılması	25
2.3.3. Temas Bilgilerinin Toplanması	25
2.3.4. TMAGA'nın Kerteriz Bilgilerinden Trak Oluşturması	28
2.3.4.1. Kromozomları Değerlendirmek	32
2.3.4.2. Seçme (Selection)	35
2.3.4.3. Çaprazlama (Crossover)	37
2.3.4.4. Mutasyon (Mutation)	40
2.4. Denemeler	46
2.4.1. TMAGA Geçerlilik Testleri	47
2.4.1.1. Deneme 1 - 3	47
2.4.1.2. Deneme 4 - 6:	51

2.4.2. Ölçüm Verisi Kalitesinin Etkileri	54
2.4.2.1. Deneme 7 - 9:	54
2.4.2.2. Deneme 10 - 12	57
2.4.3. Çaprazlama Olasılığının (Crossover Probability) Algoritma Üzerindeki Etkisi	60
2.4.3.1. Deneme 13 - 15:	61
2.4.4. Mutasyon Olasılığının (Mutation Probability) Algoritma Üzerindeki Etkisi	64
2.4.4.1. Deneme 16 - 18:	65
3. ARAŞTIRMANIN SONUÇLARI VE GELECEK ÇALIŞMALAR İÇİN ÖNERİLER	70
KAYNAKLAR	72
ÖZGEÇMİŞ	75

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Ters Problemler Model Gösterimi	2
2.1. Basit HHA Diyagramı	8
2.2. Farklı Gözlemlenebilirlik Şartları	12
2.3. Klasik Genetik Algoritma Akış Diyagramı	20
2.4. Hedef Hareketini Modelleyen Kromozom Yapısı	22
2.5. Mutlak Doğru Kerterizleri ve Ölçülen Kerterizler	27
2.6. Parametre Uzayı Daraltma Algoritması Kodu	30
2.7. Toplam Sapma Değerinin Hesaplanması	34
2.8. Seçme Prosedürü	37
2.9. Çaprazlama Prosedürü	39
2.10. Mutasyon Prosedürü	41
2.11. TMAGA Akış Diyagramı	45
2.12. Deneme 1 - 3 Taktik Resim	50
2.13. Deneme 4 - 6 Taktik Resim	53
2.14. Deneme 7 - 9 Taktik Resim	56
2.15. Deneme 10 - 12 Taktik Resim	59
2.16. Deneme 13 - 15 Taktik Resim	63
2.17. Deneme 16 - 18 Taktik Resim	67

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Rota karar deęişkenine uygulanan parametre uzayı daraltma algoritmasının gösterimi	31
2.2 Deneme 1 - 3 arası kendi gemi kinematik parametreleri	48
2.3 Deneme 1 - 3 Test Sonuçları	49
2.4: Deneme 4 - 6 arası kendi gemi kinematik parametreleri	51
2.5 Deneme 4 -6 test sonuçları	52
2.6 Deneme 7 -12 arası kendi gemi kinematik parametreleri	54
2.7 Deneme 7 - 9 test sonuçları	55
2.8 Deneme 10 -12 Test Sonuçları	58
2.9 Deneme 13 -15 Test Sonuçları	62
2.10 Deneme 16 -18 Test Sonuçları	66
2.11 Denemelerin genel özeti	69

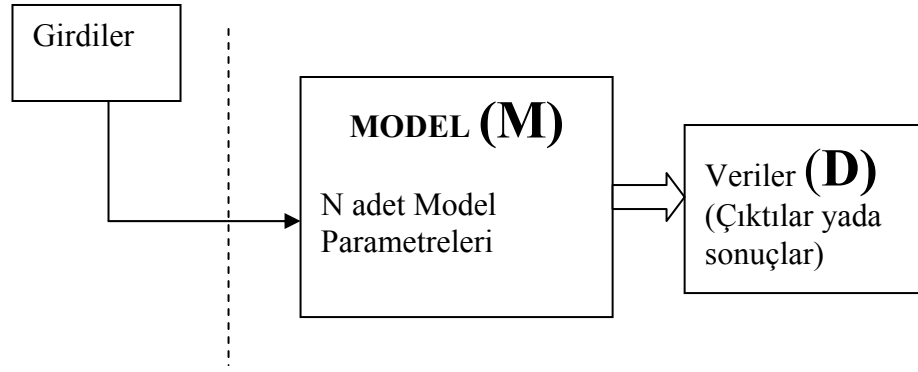
1. GİRİŞ

Ters problemler (Inverse Problems yada Inversion Problem) bilim ve matematiğin bir çok alanlarında kullanılan, bazı model parametrelerinin gözlemlenen verilerden elde edildiği problemlerdir (Tarantola, 2005). Daha sade bir tanımla, cevabın bilindiği ancak sorunun ne olduğunun arandığı problemlerdir. Yada sonuçların bilindiği, sadece bu sonuçlara götüren sebeplerin arandığı problemlerdir. Örneğin bir işletmede 3 ayrı hattan çıkan, miktarları belli sıvılar bir havuzda toplanarak birbirleriyle etkileşime girerek bir bileşim meydana getirmektedirler. Bu esnada herhangi bir kütle kaybı olamamakla beraber, bu sıvılar etkileşim miktarlarına göre kendi özelliklerini kaybetmekte, yeni bir sıvı haline gelmektedirler. Havuzda toplanan bileşimin miktarını bulmak kolay, klasik bir problemdir (üç sıvının kütlelerini toplamak yeterlidir). Ancak bundan daha zor bir problem ise, toplam bileşim miktarının bilindiği ancak hatların herbirinden gelen sıvıların miktarının bilinmediği durumdur. Her hattan ne kadar sıvının geldiğini bulmak bir ters problem konusudur.

Başka bir örnek olarak da şu örneği verebiliriz;

Belirli bir miktardaki paramızı değerlendirmek üzere, döviz, banka faizi, borsa vb. den oluşan bir yatırım sepetine koyuyoruz. Bir ay sonra eğer yatırım sepetinin içeriğini ve her yatırımın ne kadar gelir getirdiğini biliyorsak paramızın ne kadar artacağını bulmak kolay, düz bir problemdir. Öte yandan, bir ay sonra sadece paramızın ne kadar arttığını biliyorsak, yatırım sepetinin içeriğini ve her yatırımın getiri miktarını bulmak bir ters problemdir.

Ters problem kavramını aşağıdaki şekil ile de gösterebiliriz.



Şekil 1.1. Ters Problemler Model Gösterimi

Elimizde bir takım parametrelerden oluşan bir M model mevcuttur. Modelde parametrelerin isimleri bilinmekte ancak değerleri bilinmemektedir. Bu modelden herhangi bir girdi sonucu yada direkt kendi içerisinde bir takım çıktılar (D) alınmaktadır. Bu çıktılardan ne olduğu ortadadır, bizim aradığımız model parametrelerinin değerleridir. Yani, model nasıl bir parametrelerden oluşmaktadır ki biz sonuçta gözlemlenen verileri elde etmekteyiz. Ters problemler bu soruyla ilgilenir.

Ters problemler kendi içinde Lineer ve Non-lineer olmak üzere iki gruba ayrılırlar (Tarantola, 2005). Lineer ters problemlerde, model parametreleri ile veriler arasında lineer bir ilişki mevcuttur. Bu durumda problemi çözmek kolaydır. Öte yandan non-lineer ters problemlerde, veriler ile model parametreleri arasında non-lineer bir ilişki vardır ve bu tür problemleri çözmek gayet zordur (Tarantola, 2005).

Ters problemlerde model parametrelerinin değerleri bilinmemektedir ancak alabileceği değerler genelde sınırlıdır. Her parametrenin alabileceği değerlerin kümesine parametre uzayı denir. Non-lineer ters problemlerde, N adet parametreden oluşan bir modelin parametrelerini bulmak N adet değişik parametre uzayı içerisindeki değerlerin birbirleriyle olan kombinasyonunu bulmak olarak da düşünülebilir. Örneğin 4 parametreden oluşan bir modelde, her parametreye ait uzay 10^6 elemandan oluşuyorsa $(10^6)^4$ değişik parametre kombinasyonu bu modeli oluşturabilmektedir. Bu durumda bizim aradığımız cevap 10^{24} seçenektir. Görüldüğü gibi bu hiçte kolay olmayan bir görevdir. 10^{24} seçenektir herbirini sırayla denemek (exhaustive search) en gelişmiş bilgisayarlarla bile yıllar alabilecek bir işlemdir.

Bu durumda, global bir optimizasyon methodu olarak genetik algoritmaları kullanmak çok akılcı bir davranış olacaktır. Bizim ters problem çözümlerinde genetik algoritmaları kullanma mantığımız şu şekilde doğmuştur.

Eğer çok büyük bir parametre uzayına havi bir modelin çıktısını biliyorsak, genetik algoritmamız, bu parametre uzayı içerisinde parametre kombinasyonları arayarak değişik modeller oluşturur ve oluşturduğu bu modellerden girdileri kulalnarak veriler elde eder. Bu verileri mevcut gerçek verilerle kıyaslayarak aralarındaki farkı minimize etmeye çalışır. Bu yolla, netice de bulacağı parametre kombinasyonu gerçek modeli oluşturacaktır.

Bir çok bilimsel ters problem çözümlerinde genetik algoritmaların sıklıkla kullanıldığı bilinmektedir (Boschetti et al, 1996 ; Lunacek et al, 2004 ; Olariu, 2006). Genetik algoritmaların detayı ilgili bölümde anlatılacaktır.

Bu çalışmada, son bir kaç on yıldır akademik ve askeri çevrelerde önemli seviyede araştırma konusu olmuş, detayları aşağıda anlatılan “Hedef Hareket Analizi (HHA) “ (Target Motion Analysis – TMA) problemi bir ters problem olarak ele alınmış ve genetik algoritma ile çözülmüştür. Aşağıda HHA problemi ve problemin anlaşılması için gerekli ön bilgiler kısaca anlatılmıştır.

Sualtında hareket eden, dolayısıyla radar gibi suüstü hedeflerini hassas bir şekilde takip etmekte kullanılan cihazların takip edemediği hedefler, *sonar* denilen, en genel çerçevede sualtı ses dalgalarını bir seri hidrofon denilen alıcılarla hissedilen cihazlar vasıtasıyla tespit ve analiz edilmeye çalışılırlar. Sonar cihazları, temelde iki modda çalışır. Bunlar aktif ve pasif modlardır. Aktif modda çalışan sonara, *Aktif Sonar*, pasif modda çalışana ise *Pasif Sonar* denir. Aktif sonar, sualtına ses dalgaları gönderir ve gönderdiği dalgaların sualtında bir takım engellere çarpıp geri gelmesini bekler. Bu işleme ping atma denir. Aktif sonarın gönderdiği ses dalgasının gidiş-geliş zamanı arasındaki farktan, sesin o anda sualtında ilerlediği hız da hesaba katılarak (ses sualtında havaya göre 4-5 kat daha hızlı hareket eder), engelin mesafesi bulunur. Bunun yanında sesin geldiği yön de işleme katılarak engelin tam mevkii hesaplanır. Sesin geldiği yöne, askeri terminolojide *kertemiz* (bearing) denir. Engel eğer hareket ediyorsa – ki bu muhtemelen bir

denizaltı yada suüstü gemisidir – bir kaç ping atmayla hedefin anlık mevkiileri bulunur ve bu mevkiiler kullanılarak hedefin hareket parametreleri çıkartılır. Hedefin hareket parametrelerinden kasıt rota ve sürat bilgileridir.

Pasif sonar ise, herhangi bir ses intişarı yapmadan sadece etrafta başka cisimlerden yayımlanan sesleri dinler. Bu esnada tek elde edebileceği bilgi sesin yönü, yani kerterizidir. Herhangi bir mesafe bilgisi çıkarmaya tek başına muktedir değildir.

Sualtı ölçüm cihazlarını ve çalışma modlarını kısaca tanıttıktan sonra, Hedef Hareket Analizinin ne olduğu şimdicedaha kolay anlaşılacaktır.

Hedef Hareket Analizi (HHA), İngilizce karşılığı ile Target Motion Analysis (TMA), sualtında sadece pasif sensörlerden alınan ses kerterizi ve bazen de ses frekansı bilgilerinden, hedefe ait kinematikleri, yani mevki, rota ve sürat bilgilerini çıkarma işlemidir. Hiçte kolay olmayan bu işlemin zorluğunu somut bir benzetmeyle gösterecek olursak, HHA, gözleri bağlı bir insanın kendisinden epey uzakta geçmekte olan bir arabanın, tam yerini, gittiği yönü ve süratini, herhangi bir radar gibi cihaz kullanmadan, sadece sesi dinleyerek bulması işlemine benzetilebilir.

Sadece kerterizlerin kullanılmasıyla yapılan hedef izleme, yani HHA, bir kaç on yıldır aktif bir akademik çalışma alanı olmuştur. Bu çalışmada, HHA problemi çözümü maksadıyla tamamen farklı bir yaklaşım tanıtılmıştır. Sadece kerteriz bilgileri kullanarak hedef kinematik parametrelerini bulmak amacıyla, Kalman Filtresi (Peters, 2001), Extended Kalman Filtreleri (Balckman, 1999) ve varyasyonları (Grossman, 1991 ; Kronhamm, 1998; Moon and Stevens, 1996 ; Peach, 1995) gibi klasik yöntemler yerine, HHA bir ters problem olarak ele alınmış ve problemin çözümünde bir Evrimsel Hesaplama (Evolutionary Calculation) tekniği olan Genetik Algoritmalar (GA) kullanılmıştır.

Geliştirdiğimiz algoritmaları test etmek amacıyla TMAGA denilen bir yazılım test ortamı geliştirilmiştir. Yapılan denemelerde kaydedeğer bir başarı elde edilmiştir.

Bölüm 2 'de sırasıyla HHA probleminin detayları, Evrimsel hesaplama tekniklerinden Genetik Algoritmalar, HHA probleminin Genetik Algoritmalar ile çözümü anlatılacak ve son olarakta yaptığımız deney sonuçları gösterilecektir.

2. HEDEF HAREKET ANALİZİ (HHA) VE EVRİMSEL HESAPLAMA TEKNİKLERİ

Bu bölümde Hedef Hareket Analizi problemi detaylı olarak anlatılacak, bilahare bir evrimsel hesaplama yöntemi olan genetik algoritmalar tanıtılacaktır. Daha sonra HHA probleminin genetik algoritmalar ile çözümü konusu tartışılacak ve bu konuda bir test ortamı olarak geliştirdiğimiz TMAGA adlı uygulama yazılımını inceleyeceğiz. Bölüm sonunda ise TMAGA tarafından gerçekleştirilen denemeler ve elde edilen sonuçlar irdelenecektir.

2.1. Hedef Hareket Analizi

Hedef Hareket Analizi, sadece pasif yöntemlerden elde edilen bilgileri kullanarak hareket eden platformları traklama metodudur. Traklama kavramı, denizcilik ve askeri terminolojide, bir sensörün hareket eden bir cismi takip ederek hangi zamanda hangi noktada olacağını bilmesi, kısaca hareket parametrelerini (mekii, hareket istikameti, hareket hızı ve ivmesi) bulması işlemidir. Bundan sonra traklama dendiği zaman bu anlam kastedilecektir.

HHA, hareket eden platformun mevkii ve hızının direkt olarak ölçülemediği yada radar, aktif sonar gibi aktif yolların kullanılmasının operasyonel olarak çok sıkı bir şekilde arzu edilmediği durumlarda kullanılır. Gizli kalmanın hayati önem taşıdığı denizaltı gemileri gibi platformlar için HHA vazgeçilmez bir yöntemdir. Genel bir HHA probleminde, hedefin düz bir hat üzerinde, herhangi bir rota ve sürat değişikliği yapmadan hareket ettiği farz edilerek, yayımladığı gürültülerin yönü (kerterizi) ve frekansları ölçülür. Bu ölçümler kullanılarak hedefin mevkii, sürati, rotası tahmin edilmeye çalışılır. Bu çalışmada hedef gürültüsünün frekansları dikkate alınmamış, sadece gürültünün geldiği yön bilgisi kullanılmıştır. Hedefin ve ölçüm yapan platformun hareketlerinin geometrisi ve matematiksel modeli aşağıda açıklanmıştır.

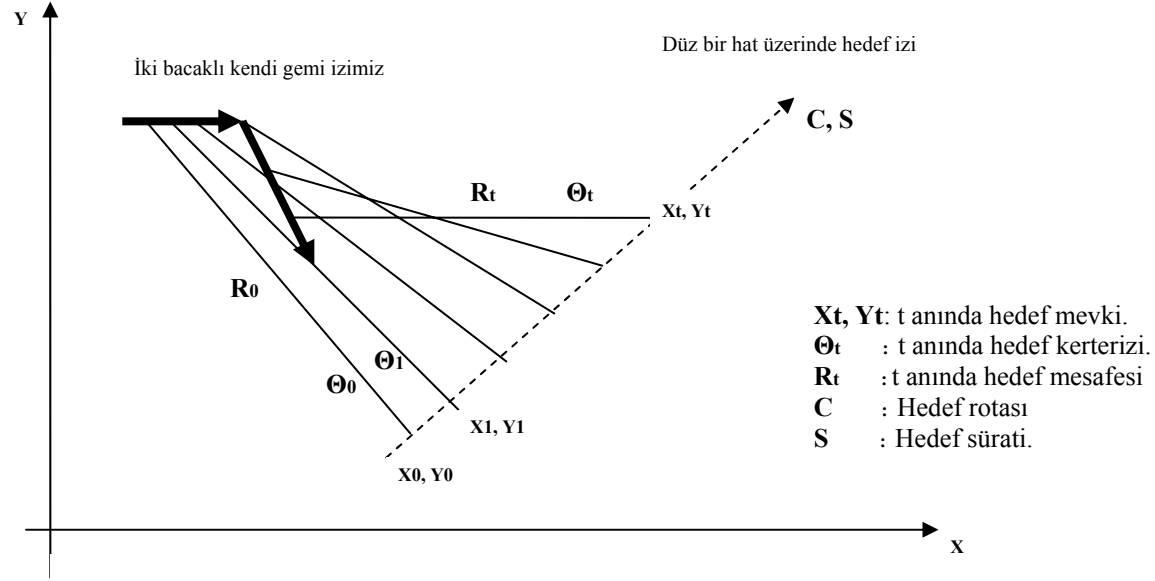
Hedef, sabit bir rota ve srat ile X-Y dzlemi olarak adlandırılabilcek dz bir dzlem zerinde hareket etmektedir (Bkz. Őekil 2.1). Burada ters problem kavramında aıkladığımız model, hedefin hareketidir. Bu modeli oluŐturan parametreler ise bundan sonra bizim zerinde odaklaŐacađımız hedef hareket (kinematik) parametreleridir. Bunlar dz izgi zerinde hareket eden hedefin t_0 anındaki kartezyen koordinat sistemindeki mevkii olan X_0 , Y_0 noktası ve hedefin rota ve sratinden treyen hız bileŐenleri V_x , V_y dir. Her hangi bir t anında hedefin X_t, Y_t pozisyonu bu parametrelerden aŐađıdaki verilen Őekilde ıkartılabilir.

$$V_x = S.Cos(C) \quad (2.1);$$

$$V_y = S.Sin(C) \quad (2.2);$$

$$X_t = X_0 + V_x.t \quad (2.3);$$

$$Y_t = Y_0 + V_y.t \quad (2.4);$$



Şekil 2.1. Basit HHA Diyagramı.

Burada S , hedef süratini, C ise $0 - 360$ derece arası bir değer olabilen hedef rotasını göstermektedir.

Ölçüm yapan platforma genelde literatürde “ownship” denilir. Biz de bundan sonra ölçüm yapan platform için “ownship”in Türkçe karşılığı olan “kendi gemimiz” tabirini kullanacağız.

Hedefle aynı X-Y koordinat sisteminde hareket eden kendi gemimizde konuşlu bir pasif sonar $X_o(t)$, $Y_o(t)$ noktalarında T saniye aralıklarla ölçüm yaparak n adet kerteriz bilgisi üretir. Hedef merkezinden, kendi gemimiz merkezine çizilen bir doğru hattının pozitif Y ekseninden, (coğrafi 360 dereceye tekabül eder) saat yönünde olan açısız mesafesi θ hedefin kerterizidir. Herhangi bir t anındaki bu açı;

$$\theta(t) = \arctan \left(\frac{Y(t) - Y_o(t)}{X(t) - X_o(t)} \right) \longrightarrow$$

$$\theta(t) = \arctan \left(\frac{Y_0 + V_y \cdot t - (Y_{o0} + V_{oy} \cdot t)}{X_0 + V_x \cdot t - (X_{o0} + V_{ox} \cdot t)} \right) \quad (2.5);$$

ifadesiyle tanımlanır.

Yukarıdaki hedef hareket parametrelerinin, ancak ve ancak kendi gemi hareketimizin hedef hareketinin daha yüksek derecede bir türevi olması halinde gözlemlenebileceği çok iyi bilinen bir gerçektir (Ancker, 1958 ; Porat and Friedlander, 1970). Bu şu demektir; Eğer hedef sabit bir süratle düz bir çizgi üzerinde hareket ediyorsa (yani açısız veya doğrusal bir ivme yapmıyorsa), hedef parametrelerinin gözlemlenebilir olması için, kendi gemimizin ivmeli bir hareket yapması gerekiyor demektir. Bu ivmeyi sağlamanın en kolay ve pratik yolu HHA problemi başladıktan sonra herhangi bir an kendi gemimizin bir dönüş icra etmesidir. Bu yüzden, bizim

varsayımlarımız altında, yani hedefin sabit süratle sabit rotada seyretmesi durumunda, HHA probleminin çözümü için kendi gemimiz bir dönüş yapacaktır. Bu olgu, algoritmamız içerisindeki temel varsayımlardan birisidir. HHA probleminde “gözlemlenebilirlik” kavramını bundan sonraki paragrafta açıkladıktan sonra, bir daha bu konu üzerinde durmayacak ve daima hedefin sabit rota- sabit sürat hareket tarzını izlediğini, kendi gemimizin ise sabit sürat – iki bacaklı rota şeklinde hareket edeceğini farz edeceğiz.

HHA probleminin daha iyi bir şekilde anlanabilmesi için, hedef traktama konsepti içerisindeki *gözlemlenebilirlik* kavramının iyi anlaşılması gerekir. HHA problemindeki gözlemlenebilirlik kavramı, kısaca, n adet hedef kerterizi kullanarak sadece ve sadece tek bir hedef parametre seti (yani tek bir çözüm) bulabilme yeteneği olarak tanımlanabilir. Eğer birden fazla sayıda bir çözüm, yani hedef parametre seti mevcutsa yada hedef parametrelerinden birisi sıfıra yada sonsuza yakınsıyorsa, bu parametre seti *gözlemlenemez* denir. Yukarıda da denildiği gibi, hedef parametreleri, ancak ve ancak kendi gemimiz hareketi hedefin hareketinin daha yüksek derecede bir türevi ise, gözlemlenebilir. Örneğin kendi gemimiz ve hedefin her ikisinin de sabit hızla sabit rotada seyrettiği durumu ele alalım. Eşitlik (2.5) 'deki aşağıdaki oran ifadesinin

$$\left(\frac{Y_0 + V_y \cdot t - (Y_{o0} + V_{oy} \cdot t)}{X_0 + V_x \cdot t - (X_{o0} + V_{ox} \cdot t)} \right)$$

hem payını, hem de paydasını aynı sabit bir sayı ile çarparsak, orandaki

$$X_0, V_x, X_o, Y_0, V_y, Y_o, Y_{o0}, V_{oy}, X_{o0}, V_{ox}$$

parametre değerleri

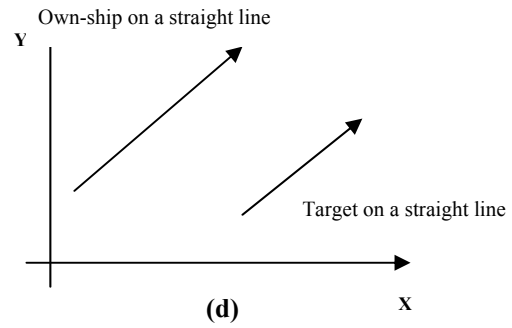
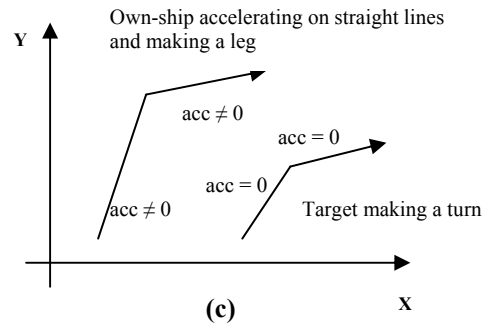
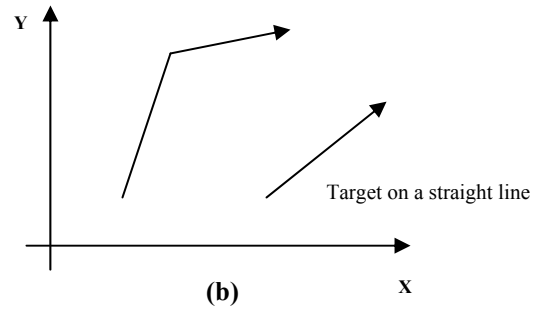
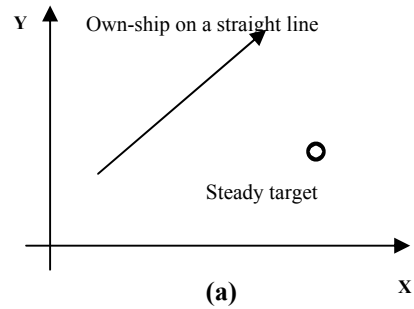
$$cX_0, cV_x, cX_o, cY_0, cV_y, cY_o, cY_{o0}, cV_{oy}, cX_{o0}, cV_{ox}$$

değerleri haline dönüşmesine rağmen oran değişmez ve tamamen farklı iki farklı parametre seti için aynı kerterizleri elde ederiz. Sabit c'nin her farklı değeri için farklı parametre setleri olmasına rağmen aynı kerterizler edilir. Dolayısıyla, elde edilen kerterizlerden sonsuz sayıda çözüm elde

edilebileceđi ortaya ıkar. Problemi daha grsel olarak ifade etmek maksadıyla, Őu durumları inceleyelim. Őekil 2.2 (a), (b) ve (c) de hedef gzlemlenebilir, unk kendi gemimiz hareketi hedef hareketinin daha yksek derecede bir trevidir. Fakat Őekil 2.2 (d) 'de durum byle deđidir, hedef ve kendi gemimizi aynı derecede bir harekete sahiptir.

Őekil 2.2 (a), durađan bir hedefin pozisyonunun tahmini ile ilgili bir durumu sergilemektedir. Bu durumda gemimiz en azından sabit bir hat zerinde sabit bir srat ile hedefi gzlemleyebilecek bir sre kadar ilerlemelidir. Őekil 2.2 (b) ise bizim incelediđimiz durumdur, hedef sabit bir rota zerinde sabit bir srat ile ilerlemektedir. Burada kendi gemimizin bir dnŐ icra etmesi (aısal ivme sađlaması) yada dz bir hatta seyredecekse hızlanma yada yavaŐlama trnde dođrusal bir ivme yapması gerekir. Son olarak, manevra yapan bir hedefin durumu Őekil 2.2 (c) de gsterilmiŐtir. Bu durumda da kendi gemimiz hareketinin hedef hareketinden bir derece da yksek derecede olması gerekecektir. Bu ancak, gemimizin hareket edeceđi iki bacak zerinde sabit rotalarda ilerlerken ivme yapmasıyla mmkn olacaktır.

En yaygın gzlemlenememe (menfi) durumunun rneđi Őekil 2.2 (d) de sergilenmiŐtir. Burada kendi gemimiz ve hedef sabit rota- sabit srat seyri yapmaktadırlar.



Şekil 2.2. Farklı Gözlemlenebilirlik Şartları.

Aynı zamanda gözlemlenebilirliğin derecesinden de burada bahsetmekte fayda vardır. Eğer ölçümler esnasında, iki ardarda gelen kerteriz arasındaki açısal fark yani kerteriz değişimi (bearing rate) yüksekse içinde bulunulan durum yüksek derecede gözlemlenebilir olarak nitelendirilir. Bu durum, hedefin yakın olduğu, süratinin yüksek olduğu yada hedef hız vektörünün görüş hattına (line of sight) dik olduğu durumlarda sağlanır. Kısaca kerteriz değişimini arttıran her durum gözlemlenebilirliği arttıracaktır. Diğer yandan, kerteriz değişiminin az olduğu durumlarda düşük gözlemlenebilirlik meydana gelir. Kerteriz değişimini etkileyen faktörler arasında, mesafa ilk sırada yer alır. Genel bir kabul olarak, uzaktaki hedeflerin HHA'sını çözmek her zaman daha zordur. Denemelerde göstereceğimiz gibi, algoritmamız düşük gözlemlenebilirlik durumlarında da başarıyla HHA problemini çözmüştür.

HHA çözümünde, pasif yollarla elde edilen kerteriz ölçümleri, iki farklı problem çözme yönteminde kullanılır. Bu yöntemlerden ilki, tekrarlamalı (recursive) yöntemdir. Burada, yeni bir kerteriz alınır alınmaz problem yeniden çözülür. Çözüm için kerteriz bilgileri bir yerde kaydedilmez. İkinci yöntem ise toplu işleme (batch processing) yöntemidir. Bu yöntemde, gelen kerterizler geldikçe, zaman bilgileriyle birlikte bir yerde kaydedilir, daha sonra kerteriz toplama işlemi sona erdiğinde bu kerterizlerin hepsi kullanılarak problem bir anda çözülür. Bu yöntem genelde işlemsel olarak pahalıdır çünkü çok miktarda verinin bir anda işlenmesi gerekecektir. Ayrıca gelen kaydetmek için hafızaya ihtiyaç duyulacaktır. Tekrarlamalı yöntemlerde, en yaygın olarak "Kalman Filtreleri" kullanılır ve yukarıda değinildiği gibi geçmiş ölçümleri kaydetmeye gerek yoktur. Yöntem işlemsel olarak da ucuzdur. Ancak bu yöntemin aşağıda açıklanacağı gibi bir çok sakıncası vardır. Bu yöntemleri kullanan algoritmalar çokca literatürde yer almışlardır (Aidala and Hammel, 1983 ; Fagerlund, 1980 ; Gong and Lindgren, 1981 ; Gong and Speyer, 1985 ; Nardone and Lindgren, 1984 ; Wilhoit., 1983).

Toplu işleme tekniklerinde genelde, tahmin edilen durum değişkeni (estimated state variable) ve gerçek ölçümlerin kullanılmasıyla yapılan

hesaplamalarla belirlenen, ölçülen kerterizlerle gelen sinyal frekansları arasındaki farklar ölçüm hatasının standart sapmasıyla normalize edilirler ve bu değerlerinin karelerinin toplamı bulunur. Bu toplam değer, değerlendirme değeri (evaluation value) veya diğer bir tabirle maliyet fonksiyonu (cost function) olarak kullanılır. En az kareler (least square) metodu kullanılarak değerlendirme değeri (evaluation value) minimize edilerek tahmin yapılmaya çalışılır (Fujimoto et al, 1997). Tekrarlamalı teknikler (Genelde Kalman Filtresi kullanan yöntemler), ölçümler ile durum değişkeni arasındaki ilişkilerin lineer bir fonksiyonla ifade edilmesi şeklinde formülize edilirler ve bu durumda toplu işlem tekniklerine denk bir şekilde çözüm elde edilir. Uygun gürültü, geometrik ve çevresel koşullarda, ve tabii ki yüksek gözlemlenebilirlik durumlarında, hedef durumunun sağlıklı bir şekilde tahmini, tekrarlamalı yöntemlerle yapılabilir. Lakin, pratikteki bir çok durum bu koşullar ile uyuşmaz. Özellikle su altı ortamları yüksek seviyede gürültülerin mevcut olduğu ortamlardır. Aynı zamanda her zaman yüksek gözlemlenebilir hedeflerle uğraşmak da mümkün değildir. Bunların yanında, uygun matematiksel modellerin seçilmesindeki belirsizlik, tahmin işlemi esnasında kararsızlığa ve değişkenliğe sebep olur. Üstüne üstlük, ölçülen kerterizlerle (θ) tahmin edilecek olan hedef parametreleri (x , y , V_x , V_y) arasındaki ilişkiyi lineer bir fonksiyonla göstermek mümkün değildir. Bu sebeple, problemin tekrarlamalı bir yöntemle çözümü için gerekli olan ölçüm ve tahmin edilecek parametreler arasındaki ilişkinin lineer fonksiyon olarak gösterilebilmesi şartını sağlamak amacıyla yapılacak herhangi bir lineerleştirme işlemi, ortaya yeni hatalar sürecektir. Modern tahmin tekniklerinin geliştirildiği dönemlerden önce, HHA probleminin çözümü için 1936 ve 1959 metodları denilen bir kaç rastgele metod teklif edilmiştir (Jazwinski, 1973). Modern tahmin metodlarının geliştirilmesiyle, HHA problemi çözümlerinde, Genişletilmiş Kalman Filtreleri (Extended Kalman Filters- EKF) denilen metodlar kullanılmaya başlanılmıştır. EKT yöntemlerinde, Kuzey yönü tabanlı, kartezyen koordinat sistemleri kullanılır ve sabit rotada seyreden hedefin hareketi bir durum vektörü vasıtasıyla modellenir. EKT, klasik Kalman Filtresinin genişletilmiş bir halidir. EKT'de ölçüm ve hedef

parametreleri arasındaki non-lineer ilişki önce lineerleştirilir ve daha sonra bu lineer fonksiyon klasik Kalman Filtresi'ne uygulanır.

Çok gelişmiş bir method olmasına rağmen EKT bir kaç konuda yetersizlik sergiler (Aidala, 1986 ; Johnson and Cohen,1983) ; Weiss and Moore, 1980). Özellikle traktamanın erken safhalarında, EKT ciddi problemler yaşar. Hatta en baştan dengesiz bir duruma düşerek bir daha toparlanması çok uzun bir süre alabilir (Moon and Stevens, 1996). Hatta, mesafe bilgisi daimi olarak sıfır çıkabilir (Moon and Stevens, 1996). EKT'nin bu başlama problemine karşın bir çok öneri getirilmiştir, ancak bu seferde ortaya aşırı non-lineer denklem kümeleri ve aşırı karmaşık metodlar ortaya çıkmıştır.

Hatta başlama anındaki bu anormal davranışları gidermenin yolları olarak, koordinat sistemlerinin etkileri üzerine bir çok araştırma yapılmıştır. (Aidala and Hammel, 1983 ; Gong and Speyer, 1985 ; Springarn, 1987).

Kalman Filtreleri tabanlı HHA çözümlerine ciddi bir alternatif olarak, son yıllarda *Uydurulmuş İşleme Algoritmaları (Matched Processing Algorithms)* ortaya çıkmaya başlamıştır. (Searle, 2005). HHA probleminde kullanılan *Uydurulmuş İşleme Metodu* kısaca şu şekilde tanımlanabilir;

Verilen bir parametre seti için bir taklit sinyali simüle edilir, yani sanki hedef o parametreler ile hareket ediyormuş gibi sahte bir sinyal üretilir. Daha sonra bu sahte sinyal ölçülen sinyal ile kıyaslanır ve aradaki farka bakılarak bir uygunluk skoru belirlenir. HHA çözümünde, tüm parametre uzayı aranarak gerçek ölçülen sinyalle en iyi uyuşan sahte sinyal aranır. Genelde bu arama, parametre uzayı üzerinde eksiksiz arama (exhaustive search) yapılarak icra edilir. İşlemsel olarak, bu işlem aşırı derecede pahalı bir işlemdir. Eksiksiz arama o kadar pahalıdır ki, pratikte ancak çok küçük parametre uzayları için icra edilebilir ve hatta bazı parametreler biliniyor kabul edilerek arama uzayı kabul edilebilir bir boyuta düşürülmeye çalışılır (Fialkowki et al, 2001). Örneğin kullanıcılara sunulan bir çok HHA

uygulamasında, kullanıcılardan hız ve mesafe parametreleri için birer alt ve üst limit belirlemeleri istenir. Hatta bazı durumlarda, hız parametresi tamamen kullanıcıdan beklenir (sonar operatörlerinin hedef hızını, pervane dönüşlerini sayarak bulabilecekleri varsayılır). Parametre uzayını azaltmak için getirilen yaklaşımlardan birisi de , verileri (ölçümleri) en yakın-komşular (nearest neighbors) algoritmaları ile ön işleme tabi tutarak, ölçümlere en çok benzeyen taklit sinyallerini ön plana çıkarmaya çalışmaktır (Ozard et al, 1993).

Genetik Algoritma tabanlı HHA yaklaşımımız aslında bu manada bir Uydurlulmuş İşleme Algoritması'dır. Genetik Algoritmaların, parametre uzayı içerisinde arama yapma hususunda, kullanıcılardan her hangi bir parametre kısıtlaması talep etmeden, çok daha iyi ve güçlü bir performans sergilediğini iddia ediyoruz. Ayrıca Genetik Algoritmanın basitliği ve kolay adapte edilebiliyor olması sayesinde, çok karmaşık bir algoritmanın sunacağı bir çok dezavantajın yaşanmayacaktır.

2.2. Evrimsel Hesaplama (Evolutionary Computing), Genetik Algoritmalar

Bir problemi çözmek, muhtemel çözümlerin bulunduğu bir küme içerisinde uygun çözümü arayarak bulmak olarak da algılanabilir. Genelde her zaman "en iyi" çözümü kovaladığımız için, problem çözme işlemini bir optimizasyon işlemi olarak ele alabiliriz. Küçük arama uzayları için, klasik eksiksiz arama metodu işe yarar, lakin arama uzayı büyüdükçe, diğer daha gelişmiş, yapay zeka tabanlı tekniklere ihtiyaç duyulur. Evrimsel Hesaplama teknikleri bu teknikler arasındadır. Bunlar stokastik algoritmalarıdır, arama methodları doğal bir fenomen olarak kabul edilen Genetik Kalıtım ve Doğal Seleksiyon'dan esinlenerek geliştirilmiştir.

Evrimsel hesaplama algoritmalarının bir kaç değişik versiyonu vardır ve bu farklı versiyonlara ait paradigmanın değişik özelliklerinin bir arada

kullanılmasıyla oluşturan bir çok melez sistem mevcuttur. Fakat yinede tüm evrimsel algoritmaların çatıları aşağı yukarı aynıdır.

Evrimsel hesaplama algoritmaları, belirli bir sayıdaki nesil (iterasyon) boyunca bireylerden oluşan bir populasyon (population) idame ederler. O nesilin populasyonundaki her birey, eldeki problem için potansiyel bir çözümü temsil eder. Her bireyin değerlendirilmeye tabi tutulduğu bir uygunluk değeri (fitness value) vardır. Gelecek nesilde, yeni bir populasyon oluşturulurken, yüksek uygunluk değerine sahip bireyler, uygunluk dereceleri oranında yeni populasyonda yer bulma şansına sahip olurlar (Seçme (Selection) Adımı). Yeni oluşturulan populasyonun bazı bireyleri, genetik operatörler denilen bir takım işlemlerle yeni transformasyonlara uğrayarak yeni çözümleri oluştururlar. Bu transformasyonlardan *mutasyon (mutation)* bazı bireylerde sadece çok ufak bir değişiklik meydana getirerek yeni bireyler oluşturur. Daha karmaşık bir transformasyon olan *çaprazlama (crossover)*, mevcut bireyleri parçalara bölüp, farklı bireylere ait parçaları birleştirmek suretiyle yeni bireyler oluşturur. Belirli sayıda bir nesil geçtikten sonra, algoritma optimum'a yakın, makul bir çözümü bulmuş olur. Evrimsel algoritmaların farklı versiyonları kendi içlerinde, bir bireyi temsil etmekte kullandıkları veri yapısı ve kullandıkları genetik operatörler ve operatörlerin sırası açısından ayrılırlar.

Genetik Algoritmalar (GA) belkide en yaygın şekilde kullanılan evrimsel hesaplama yöntemidir. Bir çok farklı disiplinde çok sayıdaki mühendislik optimizasyon probleminin çözümünde kullanılmışlardır. Örneğin inşaat mühendisliği alanında, bir nehir/su rezervuarı sistemi, ekonomik bir şekilde su döngüsünü sağlamanın optimize edilmesi amacıyla genetik algoritmalar ile dizayn edilmiştir (Fahmy et al 1998). Bilgisayar bilimleri alanında etkin veritabanı hazırlama yöntemi olarak (Cedeno and Vemuri, 1998), elektrik mühendisliğinde termal güç üretiminin optimum programlanması (Dasgupta, 1998) amacıyla kullanılmışlardır.

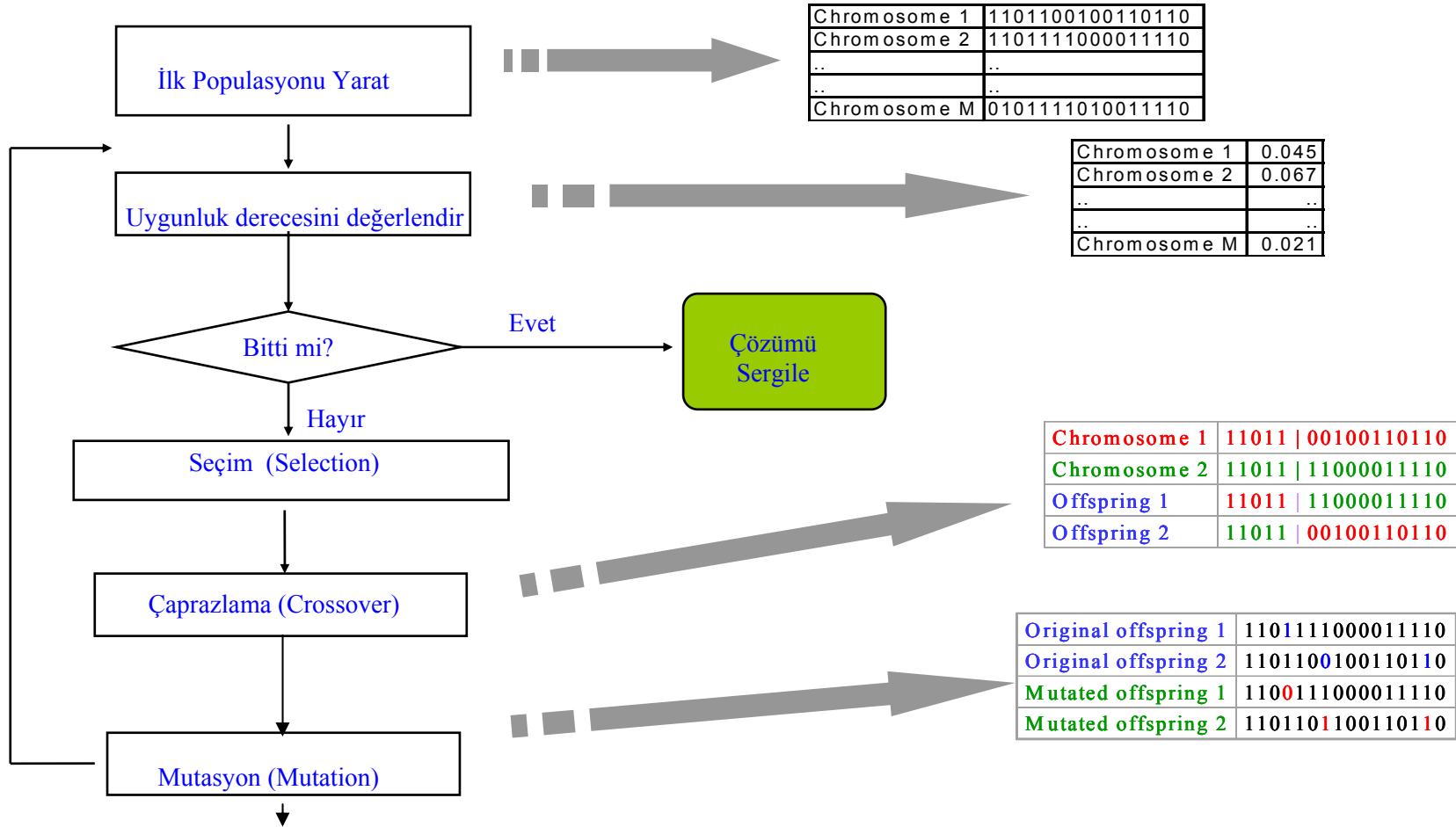
Genetik algoritmalar en genel kabul ile, bir populasyon içerisindeki bireylerin evrim geçirerek daha gelişmiş, daha iyi (muhtemelen de en iyi) formlara dönmesi esasına dayanan bir global optimizasyon tekniği olarak değerlendirilirler. Pozitif bir evrim sağlamayı garanti etmek amacıyla (bireylerin daha iyi cinslere dönmesi) bazı teknikler kullanılır. Biz burada, bir çok ders kitabında çok iyi şekilde dokümente edildiği için bu tekniklerin detaylarından bahsetmeyeceğiz (Eiben and Smith, 2003), (Mitchell,1998) ve (Tomasz, 2006).

Bir problemi genetik algoritma ile çözebilmek için, problem öyle bir şekilde ifade edilmelidir ki, potansiyel çözümler bir alfabenin elemanlarından oluşan bir karakter dizini şeklinde kodlanabilsin. Bazen bu hiçte kolay olmayan bir iştir ve bazı ilave horistiklere ihtiyaç duyulabilir. Bu çalışmada kullanılan alfabe 1 ve 0 lardan oluşan ikili sayı sistemi rakamlarıdır. Çözümleri temsil eden bu karakter dizinine *birey*, *genom* yada daha yaygın kullanımıyla *kromozom* denir.

Bir kromozomun (bireyin) yapısının nasıl olacağına karar verildikten sonra, rastgele kromozomlardan (potansiyel çözümlerden) oluşan bir populasyon meydana getirilir. Daha sonra her kromozomun kodu açılır (kromozomun temsil ettiği potansiyel çözüm ortaya çıkartılır), bazı ölçülere göre potansiyel çözümler test edilir ve skorlanarak bu çözümlerin uygunluk değerleri hesaplanır. Daha yüksek uygunluk derecesine sahip kromozomlar, uygunluk derecelerinin büyüklüğüne nispeten daha fazla yaşama şansı bulacaklar (bir sonraki neslin populasyonunda kendilerine yer bulacaklar) ve aynı zamanda yeni yavrular üretebilmek amacıyla aday ebeveyn olabileceklerdir. Bir sonraki neslin populasyonu, bu kromozomlardan, bunların çifleşmesi sonucu oluşan yeni yavrulardan ve yine bunların içinden mutasyona uğramış olan bireylerden meydana gelecektir. Yeni populasyon üyeleride benzer işlemlere tabi olacak ve bu şekilde yeni yeni nesiller meydana getirilecektir. Bu şekilde işlem belirli bir nesil geçene kadar yada belirli bir sonlandırma koşulu sağlanana dek devam edecektir. Algoritma içinde kaç nesil bu işlemlerin yapılacağına *nesil sayısı (number of generation)*

denir. Bu parametre algoritmanın hızını ve etkinliğini doğrudan etkileyen bir parametredir.

Potansiyel çözümlerin kromozomlar ile gösterilmesi ve genetik algoritma operatörleri ömümüzdeki alt bölümlerde detaylarıyla açıklanmıştır. Aşağıdaki şekil (Şekil 2.3) bir genetik algoritma uygulamasının genel akışını göstermektedir.



Şekil 2.3. Klasik Genetik Algoritma Akış Diyagramı.

Yüksek miktarda hatanın mevcut olduğu durumlarda HHA gerçekten ilgi çekici bir problemdir. Alınan akustik temaslarındaki bu hatanın en büyük sebebi yukarılarda da değinildiği gibi sualtı şartları ve ortamıdır. Bunlara ilaveten sığ sularda yapılan operasyonlarda çevresel, atmosferik ve coğrafi koşullar da hatanın artmasına sebep verebilirler. Kalman filtreleri gibi klasik çözümlerin bu durumlarda fazla işe yaramadığını, kolayca sapabileceklerini yukarıda söylemiştik. Böyle koşullarda evrimsel hesaplama gibi stokastik yaklaşımların, problemin bu karakterine karşı en uygun (hatta tek) yaklaşımlar olduğunu düşünüyoruz. Bu çalışmanın amacı, çok gürültülü ortamlarda ve düşük gözlemlenebilirlik şartlarında HHA probleminin çözümüne yönelik bir metod ve uygulama sunmaktır. Bundan sonraki bölümde çalışmamızın detayları anlatılmıştır.

2.3. TMAGA (Target Motion Analysis with Genetic Algorithms) Uygulaması

Genetik Algoritma ile oluşturduğumuz HHA çözüm algoritmasını denemek için bir test ortamına ihtiyaç vardı. Bu amaçla geliştirdiğimiz algoritmaları denemek için TMAGA adında bir uygulama programı geliştirildi. Bu bölümde bu program kısaca tanıtılacak, bu tanıtma esnasında, algoritmamızın detayları da anlatılacaktır.

TMAGA Visual Basic 6.0 ile geliştirilmiş, bir takım fonksiyon ve veri yapılarından oluşmuş, modüler bir yazılımdır. TMAGA ile, seri halinde, tutarlı bir kerteriz seti beslediği müddetçe, sadece bu kerteriz bilgileri ile herhangi bir hedef traklanabilir. Uygulama yazılımındaki iki önemli değişken, *nesil sayısı (number of generation)* ve *Monte Carlo denemeleri sayısıdır*. Bu değişkenler birbirlerine paralel olarak yada birbirlerinden bağımsız olarak değiştirilerek uygulamanın performansını (hız olarak) yada sonuçların doğruluğunu etkilerler. Bu değişkenler ne kadar yüksek olursa uygulama programının sonuçlanması o kadar çok zaman alır, ancak sonuçlar daha kati olur. Diğer taraftan bu değişkenlerin düşük tutulması programın sonlanmasını hızlandırırken, sonuçların sıhhat derecesini de bir nebze düşürebilir. Yaptığımız denemelerde, programın iki dakika civarında

çalışması sonucu elde edilen sonuçların son derece sağlıklı olduğunu gördük ki iki dakika HHA çözümü için oldukça iyi bir zamandır.

Bundan sonraki alt bölümlerde, öncelikle bir kromozomun (yani potansiyel çözümün) yapısını göstereceğiz. Bu yapı, bizim çözüm kümesi içerisinde hedefin hareket parametrelerini, yani diğer deyişle ters problem kavramındaki aranan modeli nasıl temsil ettiğimizin göstergesidir. Daha sonra uygulama programının kromozomları nasıl manipüle ederek, onları gerçek çözüme (yada gerçek çözüme çok yakın bir çözüme) doğru evrimselleştirdiğini açıklayacağız.

2.3.1. Hedef Hareketinin Temsil Edilmesi – Kromozom Yapısı

Bundan önceki bölümlerde de belirtildiği gibi, bir hedefin belirli bir periyod boyunca, daima düz bir hat üzerinde, herhangi bir dönüş icra etmeden sabit süratle seyredeceği farzedilmektedir. Aynı zamanda, pasif sensörlerden elde edilen veri sadece kerteriz bilgisidir. Bu varsayımlar altında, hedef hareketini temsil eden bir kromozom dört parçadan oluşur. Bunlar, HHA problemi başladığı anda hedefin bulunduğu ilk pozisyonun yatay ve dikey bileşenleri (X_0 , Y_0), hedefin sabit kalacağı farsedilen rotası (Course (C)) ve hızı (Speed (S)) dir. Bu değişkenlere *karar değişkenleri* adı verilir. (X_0 , Y_0), hedefin t_0 anındaki ilk mevkisidir ve daha sonraki t_1 anından t_{n-1} anına kadar olan (X,Y) mevkileri hedefin rota (C) ve sürati (S) kullanılarak hesaplanır. Şekil 2.4 bir kromozomun, yani diğer deyişle bir bireyin genel yapısını göstermektedir.

(___ X_0 ___ | ___ Y_0 ___ | ___ C ___ | ___ S ___)

Fi Şekil 2.4. Hedef Hareketini Modelleyen Kromozom Yapısı

Bir kromozom, hedef hareketini sadece belirli bir süre müddetince modeller. Bu süre HHA problemini çözmek için gerekli olacak minimum kerteriz bilgisinin toplanabilmesi için yeteri kadar uzun olmalıdır. t_i , sistemin pasif sonardan herhangi bir temas (yani kerteriz) aldığı zamanı gösterir. Yani n adet kerteriz alma işlemi t_0 'dan t_{n-1} 'e kadar devam eder. Sistem, bu en son alınan n adet kerteriz bilgisini, genetik algoritmanın değerlendirme safhasında kullanmak üzere kaydeder. t_{n-1} zamanına ulaşıldığında, veri toplama safhası sona erer ve çözüm safhası başlar.

Klasik genetik algoritmalarda, bir kromozom genellikle 1 ve 0 lardan oluşan bir dizidir. Önceden de dediğimiz gibi biz de bu geleneği sürdürdük. Bu sebeple, bu diziyi temsil etmek için, her karakterin bir dizi hücrelerini doldurduğu bir dizi veri yapısı (array) kullandık.

Kromozom içerisindeki her karar değişkeni (X,Y, C ve S), belirli sayıda bitlerle temsil edilir. Bir karar değişkenini temsil edecek bir bit dizgesinin uzunluğu, karar değişkeninin alabileceği değerlerin kümesi olan parametre uzayının büyüklüğü ile belirlenir.

Parametre uzayının büyüklüğü karar değişkeninin alabileceği maksimum ve minimum değerler ile, bu değerlerin hassasiyet miktarına bağlıdır. Burada kastedilen hassasiyet, bir reel sayının ondalık kısmının uzunluğudur. Yani bir sayının ondalık kısmı ne kadar uzunsa, o sayı o kadar hassastır denilir. Bir karar değişkeninin temsil edileceği bit sayısı aşağıdaki formül ile hesaplanır.

$$2^{m-1} < (\text{Max} - \text{Min}) \times 10^p < 2^m - 1 \quad (2.6);$$

Burada :

Max: Karar değişkeninin alabileceği maksimum değer.

Min: Karar değişkeninin alabileceği minimum değer.

p : Hassasiyet derecesi (ondalık kısmın uzunluğu).

m : Karar değişkenini temsil edecek bit sayısı.

Yukarıdaki formülden de görüleceği gibi, bit sayısı dolayısıyla bir kromozomun uzunluğu, karar değişkeninin alabileceği maksimum ve minimum değerlere sıkı sıkıya bağlıdır. Daha sonraki paragraflarda da göreceğimiz gibi, genetik algoritmanın performansı, kromozom uzunluğu tarafından çok güçlü bir şekilde etkilenir. Bu sebeple etkili bir genetik algoritma uygulaması yapabilmek için, kromozom uzunluğu mümkün mertebe küçük tutulmalıdır. Bir deniz sahasında özellikle bir mevki belirtmek için kullanılacak (X,Y) noktasındaki X ve Y değişkenleri çok büyük değerler almaya meyillidirler (0 dan 100.000 metreye kadar) ve böyle büyük değişkenleri temsil etmek için çok uzun kromozomlara ihtiyaç duyarız. Sonuç olarak böyle uzun kromozomlar genetik algoritmanın performansı düşürür. Bu sebeple başka metodlar uygulanmalı, bu metodlarla bir karar değişkeninin maksimum ve minimum değerlerini birbirlerine nisbi olarak yaklaştırmak suretiyle kromozom büyüklüğü mümkün olduğu kadar azaltılmalıdır. Bu yolla, parametre uzayı daraltılmış olur (fakat hala eksiksiz arama yapmak için çok büyüktür). Bizim geliştirdiğimiz daraltma metodu ve bunun karar değişkenlerine nasıl uygulandığı sonraki paragraflarda anlatılmıştır.

Burada, karar değişkeninin hassasiyetinin etkisinden de söz etmekte fayda vardır. Formül (2.6) 'dan anlaşılacağı üzere, hassasiyet derecesi ne kadar büyük olursa, kromozom da o kadar uzun olur. Pratikte, küçük hassasiyet dereceli karar değişkenlerinin, genetik algoritmanın performansı açısından tercih edilir olduğu düşünülebilir. Hassasiyet derecesini küçük tutmakla, sadece kaybedilen şeyin çözüm sonucu bulunan parametrelerin hassasiyetleri olduğu sanılabilir, hatta onbinlerce metre olabilen bir X değişkeni içerisinde santimere bazında hassasiyete ne gerek var denilebilir. Ancak, asıl kaybedilen şey farklıdır. Deneylerimiz sonucunda gördükki, yüksek hassasiyet dereceli karar değişkenleri, genetik algoritmanın global optimizasyon karakterine, bir de gradyen tabanlı optimizasyon karakteri eklemektedir. Global optimizasyon, parametre uzayı içerisinde herhangi lokal bir minimum noktasına takılmadan farklı alanları ziyaret etmemizi sağlarken, gradyen arama lokal minimum noktaları civarında daha hassas arama yapmamızı sağlar.

2.3.2. Başlangıç Populasyonunun (Başlangıç Çözüm Kümesi) Yaratılması

TMAGA çalışmaya ilk başladığı zaman, başlangıç populasyonunu oluşturmak için bir fonksiyon çağrılır. Bu başlangıç populasyonundaki kromozom sayısı (populasyon büyüklüğü) ve kromozom uzunlukları öngörölmüş sistem parametreleri olarak çoktan tanımlanmıştır. Bu başlangıç kromozomlarının uzunluğu doğal olarak çok büyüktür çünkü parametre uzayı çok geniştir ve henüz bir daraltma işlemi yapılmamıştır. Bu kromozom uzunluğu, başlangıç değerleri olarak tanımlanan her parametreye (X, Y, C, S) ait minimum, maksimum ve hassasiyet derecelerinden hesaplanan bit uzunluklarının toplamına eşittir. Bu uzunluk hesaplandıktan sonra, rastgele 1 ve 0 değerleri kromozomların bitlerine yerleştirilir. Her kromozom bitlerden oluşan bir dizi (array) olduğundan, bir populasyon da dizileri tutan bir dizi (array of array) veri yapısı olarak tanımlanır.

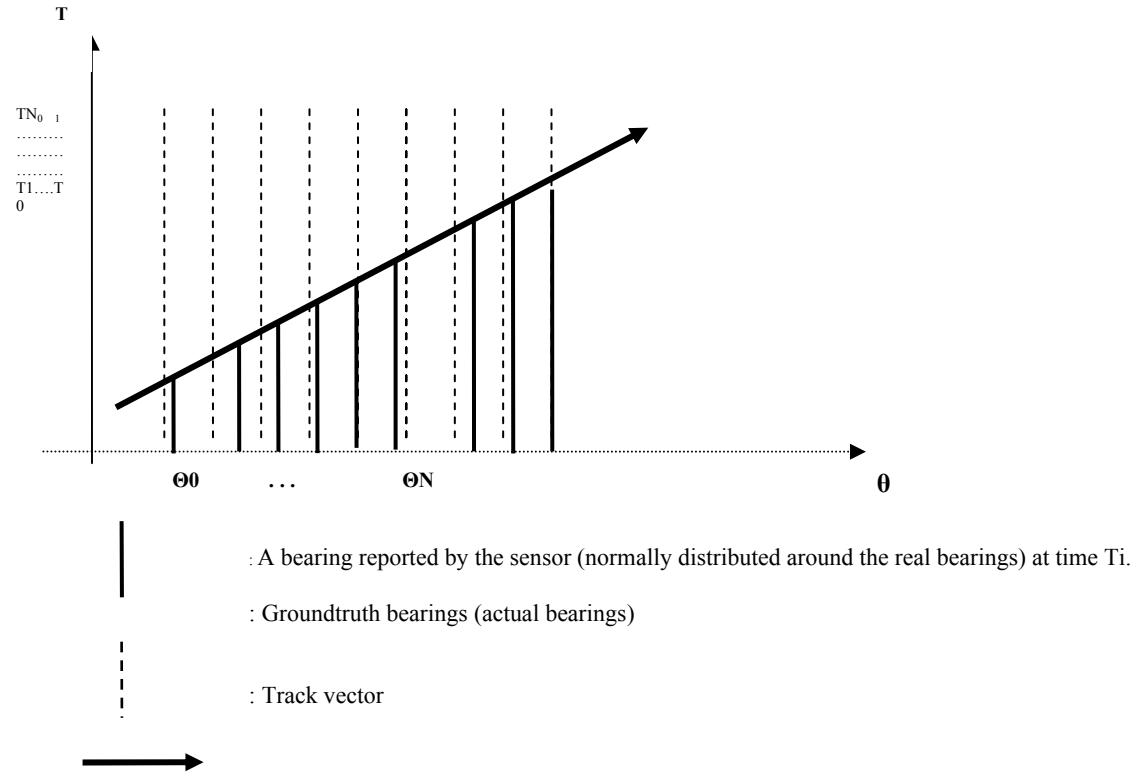
2.3.3. Temas Bilgilerinin Toplanması

TMAGA'nın bir test ortamı olarak dizayn edilmesi ve gerçek dünyaya ait verileri elde etmenin hem pratik olmaması ve hem de çok masraflı olması, gerçek hedef hareketlerini, deniz ve meteorolojik şartları ve sensör sistemlerini taklit edebilecek bir araca ihtiyaç doğurdu. Bu sebeple hedef teması üretmesi amacıyla, yüksek güvenilirlikli bir simülatör olan METEOR seçilmiştir.

METEOR, Deniz Kuvvetleri Komutanlığı Araştırma Merkezi Komutanlığı (Pendik/İstanbul) tarafından geliştirilmiş, Komuta Kontrol Sistemleri çalışmalarında kullanılan bir simülatördür. TMAGA'ya beslenen hedefler METEOR tarafından yaratılarak idame edilirler. Hedefler ya rastgele yada önceden tasarlanmış karar değişkenlerinin (X, Y, C, S) atanmasıyla yaratılırlar. Ya da önceden tanımlı senaryolar hayata geçirilir. Her halikarda, bir hedef yaratılıp aktive edildiği andan itibaren kendine atanmış rota ve sürat

ile (X_0, Y_0) noktasından başlayarak ilerlemeye başlar. Belirli zaman adımlarında, TMAGA METEOR'dan sanki gerçek bir pasif sonardan kerteriz alıyomuş gibi temas bilgisi çekmeye başlar. Bu temasları, kendileriyle ilişkili zaman adımlarıyla birlikte bir veri yapısında saklar. METEOR'dan yayımlanan kerteriz bilgileri aynı zamanda Gauss dağılımına uygun şekilde hata içerir, böylece hatalı kerteriz gönderen bir sensör de simüle edilmiş olunur. Hataların standart sapmaları, METEOR'a önceden tanımlanmış değişken olarak verilir, METEOR bu standart sapma ile normal dağılıma uygun hatalı kerteriz bilgilerini üretir.

Şekil 2.5, METEOR tarafından üretilmiş bir hedefin mutlak doğrusunu (ground truth) ve TMAGA tarafından alınmış, Gauss dağılımlı hatalı kerterizlerini göstermektedir. "Mutlak doğru" ifadesinden kasıt, hedefin gerçek hayatta yaptığı harekettir.



Şekil 2.5. Mutlak Doğru Kerterizleri ve Ölçülen Kerterizler

t_0 anından t_{N-1} anına kadar rapor edilen kerteriz değerleri ve rapor edilme zamanları dahili bir veri tabanında tutulur. Yukarıda da değinildiği gibi N sayısı bize ne HHA çözümü için ne kadar kerteriz bilgisi kullanacağımızı gösterir. Denemelerimizde, N genellikle 20 - 30 arası bir sayı olarak alınmıştır. Gerekli sayıdaki kerteriz bilgisi alınır alınmaz, TMAGA HHA problemini çözmeye başlar.

Gerçek hayatta N değerini çok önemli olduğu aşıkardır. Çünkü ilk olarak çok sayıda kerteriz toplamak çok vakit alır ve bir sualtı platformu için bu kadar süre beklemek her zaman mümkün olmayabilir. Öte yandan, kerteriz sayısı düşük tutulursa, kerterizlerin normal dağılımlı hata içermesi nedeniyle çözümüm hassasiyeti düşer. Aynı zamanda iki arda arda gelen kerterizler arasındaki fark (kerteriz değişim miktarı)'ın da önemli olduğunu gözlemlenebilirlik konusunda önceden belirtmiştik. Kerteriz değişimi miktarı ne kadar büyük olursa (hedef parametrelerinin etkisi yanında, kerteriz alma periyodu da direkt olarak kerteriz değişimini etkiler), hedefin gözlemlenebilirliği o oranda artar.

2.3.4. TMAGA'nın Kerteriz Bilgilerinden Trak Oluşturması

TMAGA esas olarak kerteriz temaslarından trak oluşturmak amacıyla kullanılır. Bu görevi, METEOR'dan aldığı kerteriz bilgilerini zaman adımlarıyla birlikte kaydedip, daha sonra toplu işleme (batch processing) yaparak ifa eder. TMAGA'da Genetik Algoritma Motoru (GAM) denilen yazılım modülü genetik algoritma görevini yerine getirir ve bu modül tüm TMAGA yazılımının çekirdeği kabul edilir. Genetik Algoritma Motoru'nun temel görevi ölçülen kerterizlerle en iyi şekilde uyuşan trak vektörünü üretmektir. Bu görevin detayları aşağıdaki paragraflarda anlatılmıştır.

İlk adım, parametre uzayını mümkün olduğu derecede küçülterek Genetik Algoritma Motoru'nun hızını ve etkinliğini artırmaktır. Genetik Algoritma Motoru öyle şekilde tasarlanmıştır ki, her çalışmasında parametre uzayını belli bir yüzdede daraltır. Başlangıçta, Genetik Algoritma Motoru'nu

düşük “nesil sayısı” değişkeni eksiksiz parametre uzayı üzerinde ard arda birkaç kez çalıştırmak, parametre uzayı üzerinde kaydedeğer bir daralma sağlar. Kısaca bu işlemde amaç, kısa kısa hızlı çalıştırmalar yaparak parametre uzayını nihai çalışma için hazırlamak üzere model parametrelerini dar parantezler içine sokmaktır.

Uygulama yazılımında, normal değeri en az 1000 olması gereken “nesil sayısı” değişkeni 100’e düşürülerek, Genetik Algoritma Motoru 20 defa ardarda çalıştırılmaktadır. Bu yaklaşımın nasıl yazılım olarak gerçekleştirildiği, “rota” karar değişkenine uygulanarak aşağıda bir örnekle açıklanmıştır.

İlk olarak, bir karar değişkeninin maksimum ve minimum değerleri sistem parametresi olarak tanımlanır. Genetik Algoritma Motoru’nun ilk çalışmasından sonra, bu karar değişkenine ait tahmini değer genetik algoritma tarafından bulunur. Bu değer çok hassas olmasına gerek yoktur ki genetik algoritma çok küçük bir “nesil sayısı” değeri ile böyle bir değer bulabilir. Bulunan bu değerle, karar değişkeninin alabileceği maksimum ve minimum değerler arasındaki fark hesaplanır ve %20 kadar maksimum değer aşağı, minimum değer yukarı, bulunan bu tahmini değere doğru çekilir (aradaki farkların %20’si kadar). Bundan sonra bu karar değişkeni için yeni maksimum ve minimum değerler, bu kaydırma işlemi sonucunda elde edilen maksimum ve minimum değerler olur. Bu işlem, her karar değişkeni için 20 kere tekrar eder ve nihayetinde parametre uzayı hatırı sayılır derecede daraltılmış olur. Tabii ki algoritma bu kadar basit değildir. Örneğin, genetik algoritma tarafından bulunan değer maksimum yada minimum değere çok yakınsa, bu sefer maksimum ve minimum değerlerin ikis birlikte bulunan uç noktaya doğru bir miktar çekilir. Benzer şekilde bazı diğer önlemler de alınarak, gerçek değer her zaman maksimum ve minimum değerler arasında kalması sağlanır. Bu algoritmanın yalancı kodu (pseudo code) Şekil 2.6’da verilmiştir.


```
while maximum – minimum > bw // bw : maximum ve minimum değerler arası aru edilen genişlik
{
  if maximum – GA_result > bw /2 Then // GA_result: Genetik Algoritmanın bulduğu değer
    maximum = maximum - (maximum - GA_result ) / 5 // maksimum değeri %20 aşağı çek
  else
    maximum = GA_result + (GA_result - minimum) / 5 // maksimum değeri artır
  end If
  if GA_result - minimum > bw /2 then
    minimum = minimum + ( GA_result - minimum ) / 5 // minimum değeri %20 yukarı çek
  else
    minimum = GA_result - (maximum - GA_result ) / 5
  end If
}
```

Şekil 2.6. Parametre Uzayı Daraltma Algoritması Kodu

Çizelge 2.1'de , gerçek değeri 85 derece olan rota karar değişkeni için, 0-360 derece arası olan parametre uzayının genetik algoritma tarafından nasıl daraltıldığı gösterilmektedir. Çizelgedeki değerler TMAGA'ın çalışması esnasında kaydedilmiştir. Görüldüğü üzere, parametre uzayı 360 derecelik bir genişlikten nihayetinde 21 dereceye daraltılmıştır. Sonuçta 9 bit olması gereken rota karar değişkenine ait bit dizisi uzunluğu 5 bit'e düşürülmüş ve genetik algoritmanın nihai sonucu bulunurken sadece bu bölgeye yoğunlaşması sağlanmıştır.

#	Maks.	GA		Par. Uzayı
		Sonucu	Min.	
1	360	81	0	360
2	304	163	16	288
3	276	76	45	231
4	236	85	51	185
5	206	206	58	148
6	236	120	88	148
7	213	190	94	119
8	208	124	113	95
9	191	118	115	76
10	176	116	103	73
11	164	106	106	58
12	152	94	94	58
13	140	89	82	58
14	130	79	79	51
15	120	81	69	51
16	112	80	71	41
17	106	92	74	32
18	103	78	78	25
19	98	77	73	25
20	94	86	73	21

Çizelge 2.1. Rota karar değişkenine uygulanan parametre uzayı daraltma algoritmasının gösterimi. Bu özel durumda gerçek rota değeri 85 derece idi.

Bu 20 koşmayla, tüm karar değişkenlerine ait parametre uzayları daraltılır, ancak henüz gerçek traklanma yapılmamıştır. Bu yüzden bu koşmalara sahte (dummy) koşma adını veriyoruz.

Daraltma operasyonu hatırı sayılır derecede performans iyileştirmesi sağlasa da, çok büyük bir sakınca doğurabilir diye düşünülebilir. Parametre uzayı güvenilir olmayan (infeasible) bir bölgeye daraltılmış olabilir. Fakat bu çok küçük bir ihtimal dahilinde gerçekleşebilecek bir durumdur, tüm 20 koşmada da genetik algoritmanın çok kötü sonuçlar bulması gerekir. Çok yoğun şekilde icra ettiğimiz denemelerin hiç birinde böyle bir durumla karşılaşmadığımızı burada belirtmemiz gerekir.

Sahte koşmalardan sonra, Genetik Algoritma Motoru için daha yüksek bir nesil sayısı ile gerçek traklama işlemini yapma sırası gelmiştir. Aşağıdaki alt bölümlerde, Genetik Algoritma Motoru'nun icra basamakları açıklanmıştır. Aşağıda açıklanan prosedürlerin, nesil sayısı kadar tekrar edildiğini hatırlatmakta yarar vardır. Bu sayede populasyon içerisinde bireyler git gide iyileşerek gerçek çözüme yaklaşacaklardır.

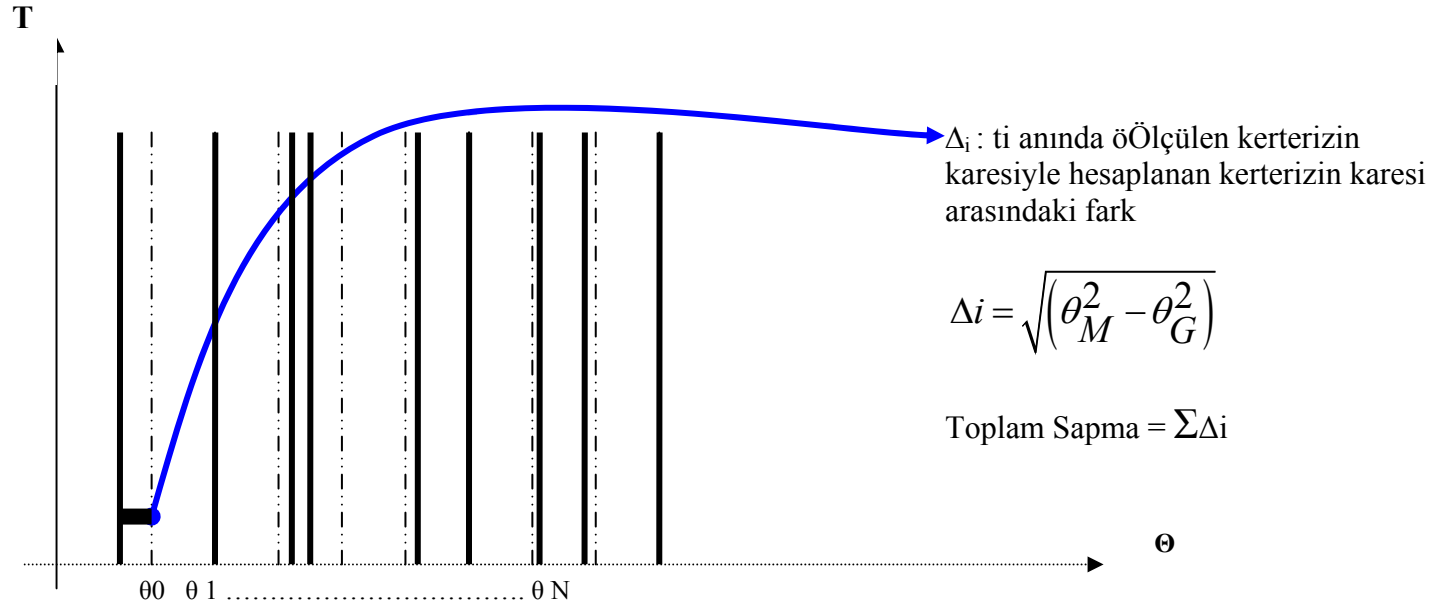
2.3.4.1. Kromozomları Değerlendirmek




Değerlendirme safhasında, t_0 dan t_{n-1} e kadar ölçülen kerterizler, kromozomu oluşturan (X_0, Y_0, C, S) ailesinden farazi olarak üretilen, herbirinin birer zaman etiketi olan kerterizlerle kıyaslanırlar. Her t_i anı için, (X_i, Y_i) noktası son bulunan pozisyondan (X_{i-1}, Y_{i-1}) C rotası boyunca S sürati ile ilerlenerek bulunur. Bu farazi noktaya tekabül eden kerteriz değeri ise aşağıdaki formül vasıtasıyla bulunur.

$$\theta(t) = \arctan \left(\frac{Y(t) - Y_0(t)}{X(t) - X_0(t)} \right) \quad (2.7);$$

Daha sonra bu kerterizlerin kareleriyle ölçülen kerterizlerin karelerinin arasındaki farklar alınır. Herhangi bir andaki iki kerteriz arasındaki farka

sapma denir. Aynı zaman etiketli tüm kerteriz çiftleri arasındaki sapmaların toplamına ise *toplam sapma (total deviation)* ($\sum \Delta_i$) denir. Şekil 2.7, *toplam sapmanın* nasıl hesaplandığını göstermektedir.



-  : Sensor tarafından rapor edilen kerteriz (Trak vektörü civarında normal dağılmış)
-  : Genetik Algoritma tarafından teklif edilen kerteriz
-  : Ölçülen ve hesaplanan kerterizler arasında ki açısal fark.

Şekil 2.7. Toplam Sapma Değerinin Hesaplanması

Daha az toplam sapma daha az hata demektir ve sonuçtaki TMA çözümü daha iyidir. Bu durumda genetik algoritmanın hedef fonksiyonu bu toplam sapmayı minimize etmek olacaktır. Genetik algoritmalar genelde maksimize işlemi için dizayn edildiklerinden, Genetik Algoritma Motorumuz toplam sapmanın tersini yani, $(\sum \Delta i)^{-1}$. değerini maksimize etmeye çalışmaktadır. Genetik algoritma terimleriyle $(\sum \Delta i)^{-1}$. ifadesine “*uygunluk değeri*” (fitness value) denir.

Bu yolla, populasyon içerisindeki her bir kromozom için bir uygunluk değeri bulunur. Bu noktada, nisbi olarak daha yüksek uygunluk değerlerine sahip kromozomlar, ideal sonuca daha yakındır diyebiliriz.

Aşağıdaki alt bölümlerde, klasik genetik algoritma prosedürlerinin nasıl gerçekleştirildikleri örneklerle izah edilecektir. Gerçek bir kromozomun uzunluğu hatırı sayılır derecede uzun olmasına rağmen, verilen örneklerde daha anlaşılır olması adına kısa tutulmuştur.

2.3.4.2. Seçme (Selection)

Bu prosedürde, gelecek nesilde yaşayacak olan kromozomlar seçilir. Her kromozoma, uygunluk değerleri esas alınarak bir “hayatta kalma” olasılık değeri atanır. Bir kromozomun uygunluk derecesi ne kadar yüksekse, hayatta kalma olasılığı da o oranda artar..

Her kromozom için hayatta kalma olasılıkları bulunduktan sonra, gelecek nesil (yeni nesil) , seçme işlemin geçen kromozolar tarafından oluşturulur. Seçme işleminde kromozomlar hayatta kalma olasılıklarına göre seçilirler. Daha yüksek hayatta kalma olasılığına sahip kromozomların seçilme şansları daha yüksektir (fakat yine de yüzde yüz garanti değildir). En düşük hayatta kalma olasılığına sahip bir kromozomun bile seçilme şansı vardır, ancak yeni nesil çok büyük ihtimalle yüksek hayatta kalma olasılığına sahip kromozomların oluşturduğu bir nesil olacaktır.

Bu yolla ideal çözüme daha yakın kromozomlar hayatta kalırken, çözümden uzak, çürük diyebileceğimiz kromozomlar ortadan kaldırılır. Seçme işlemi bittikten sonra, eski populasyon ortadan kaldırılır, yeni populasyon onun yerini alır.

Aşağıdaki örnekte, mevcut populasyondaki yüksek uygunluk derecesine sahip kromozomlar gelecek nesilde hayatta kalmak için daha fazla şansa sahip olacaklardır. Örnekten de görüleceği üzere, mevcut populasyondaki kaliteli bilgi (çözüme yakın parametreler), herhangi bir bozulmaya uğramadan, kromozom öbekleri halinde gelecek nesle aktarılmaktadır.

Mevcut Populasyon

Kromozom #	İçerik	Uyg.Değeri
1	1100101001011011	0.68
2	0101001110011010	0.11
3	0010111101001110	0.01
.
.
.



Gelecek Nesil

Kromozom #	İçerik	Uyg.Değeri
1	1100101001011011	0.68
2	1100101001011011	0.68
3	1100011010010010	0.23
.
.
.

Şekil 2.8. Seçme Prosedürü. Yüksek uygunluk derecesine sahip kromozomlar gelecek nesilde birden fazla miktarda yer alacaktır.

2.3.4.3. Çaprazlama (Crossover)

Giriş bölümünde de belirtildiği gibi genetik algoritmanın klasik prosedürlerinden birisi de *Çaprazlama*'dır. Bu prosedürde, tüm popülasyon taranır ve önceden belirlenmiş bir olasılık değerine göre bazı kromozomlar yeni yavru türetmek amacıyla ebeveyn olarak seçilirler. Herhangi bir kromozomun ebeveyn olarak seçilme olasılığına *çaprazlama olasılığı* denir ve bu değer TMAGA'da bir sistem parametresidir. TMAGA'da ki öngörülen çaprazlama olasılığı 0.2 olarak atanmıştır ki bu ortalama her beş kromozomdan birisi başka bir kromozomla çaprazlanarak yeni yavru üretecek demektir.

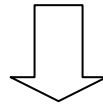
Çaprazlama işleminde, ebeveyn olarak seçilmiş kromozom çifti, rastgele bir kesim noktasından (cut point) iki parçaya bölünürler. Yavrular (iki tane), ikiye bölünmüş ebeveynlerin parçalarının şu şekilde birleştirilmesiyle oluşurlar; birince yavru birinci ebeveynin ilk parçasını alır ve bu parçaya ikinci ebeveynin ikinci parçası eklenir. İkinci yavru için ise bu sefer ikinci ebeveynin ilk parçasına birinci ebeveynin ikinci parçası eklenir. Bu yolla ebeveynlerle aynı uzunluğa sahip, ancak içerik olarak tamamen farklı iki yavru ebeveynlerin yerlerini alır.

İki iyi ebeveyn her zaman kendilerinden daha iyi iki yavru meydana getirir” diyemesekte, çaprazlama işlemi pozitif evrim yaratarak ideal çözüm setine doğru bir iyileştirme sağlar. Çaprazlama operasyonu olmasaydı, ilk nesilde rastgele atanmış bir popülasyonun yüksek uygunluk derecesine sahip üyeleri, genetik algorimanın çalışması boyunca hiç bir iyileşmeye maruz kalmadan son nesile intikal edeceklerdi ve böylece yeni, daha kaliteli yavruların oluşması engellenecekti.

Aşağıdaki şekilde (Şekil 2.9), çaprazlama operasyonu bir örnekle açıklanmıştır. Popülasyon boyunca, ebeveyn olarak rastgele kromozom çiftleri seçilmiş bu kromozomlar yine rastgele seçilmiş kesim noktalarından çaprazlanarak yeni yavru oluşturmuşlardır. Örneğimizde, 1 ve K numaralı kromozomlar ebeveyn olarak seçilmiş, sekizinci bittten sonra çaprazlanarak 1 ve K numaralı iki yeni yavru üretmişlerdir.

Çaprazlamadan Önce

Kromozom #	
1	1001001011100
2	...
..	...
k	0001101100001
..	...
M	...



Çaprazlamadan Sonra

Kromozom #	
1	1001001000001
2	...
..	...
k	0001101111100
..	...
M	...

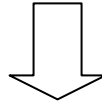
Şekil 2.9. Çaprazlama Prosedürü.

2.3.4.4. Mutasyon (Mutation)

Mutasyon, genetik algoritmanın bir diğerklasik operasyonudur. Genetik algoritmayı global optimizasyon metodu yapan mutasyon operasyonudur. Eđer mutasyon olmasaydı arama, parametre uzayı ierisinde lokal bir minimum noktasına takılır, buradan kurtulması mmkn olmazdı. Parametre uzayı ierisinde lokal minimum noktalarından kurtulmanın tek yolu, bir özm temsil eden kromozomların bitlerinin rastgele olarak ok dřk ihtimallerle seilerek deęerlerinin deęiřtirilmesini saęlayan mutasyon operasyonudur. Bu operasyonda, populasyon taranarak rastgele kromozomlar seilir. Bu seilen kromozomlardan da rastgele bitler seilir ve bu bitlerin “NOT” mantık iřlemiyle deęerleri ters evrilir (1’ler 0, 0’lar 1 olur). Ařaęıdaki rnekte (řekil 2.10), glgelenmiř bitin mutasyona sonucu deęeri ters evrilmiřtir. rneęin mutasyona uęrayan bit kromozomun o andaki deęeri 116 derece (= 111011) olan rota karar deęiřkenine tekabl eden kısmında yer alsın. Eđer bu bařtan ikici bitse, rota deęeri birden 116 dan 84’e (101011) deęiřir.

Mutasyondan Önce

Chromosome #	
1	11001 <u>1</u> 1011001101111
2	...
..	...
k	0011010110011101001
..	...
M	...



Mutasyondan Sonra

Chromosome #	
1	11001 <u>0</u> 1011001101111
2	...
..	...
k	0011010110011101001
..	...
M	...

Şekil 2.10. Mutasyon Prosedürü

Yukarıda behsi geçen tüm prosedürleri “nesil sayısı” kadar tekrar ettikten sonra, Genetik Algoritma Motoru durur. Tüm bu işlemler boyunca, gelmiş geçmiş tüm nesillerin en iyi kromozomu seçilerek, en iyi kromozom olarak kaydedilir. Bu işlem şu şekilde olur;

Başlangıçta, ilk popülasyonun ilk kromozomu en iyi kromozom (en iyi çözüm) olarak nitelendirilir. Daha sonra bu kromozomun uygunluk derecesi popülasyondaki diğer kromozomların uygunluk dereceleriyle karşılaştırılmaya başlanır. Ne zaman daha iyi bir kromozom bulunsa, bu en iyi kromozom olur ve kendisinden sonra gelecek en iyi kromozom bulunana kadar diğer kromozomlarla karşılaştırılmaya devam edilir. Popülasyonun en iyi kromozomu, gelecek nesile en iyi kromozom olarak başlar ve benzer şekilde karşılaştırma işlemi burada da devam ederek, tüm popülasyonlar dolaşarak sona kadar gelinir. Böylece işlemin başından itibaren en iyi çözüm takip edilmiş olur.

En iyi çözümün X_0 , Y_0 , C ve S değerleri gerçek hedef parametreleri olarak nitelendirilir ve bu değerler HHA'nın çözümü olmuş olur. Burada, bu çözümün de bir toplam sapma değerinin (ölçülen kerterizlerle, çözümden türetilen kerterizler arasındaki toplam açısal fark) olduğunu hatırlatmakta fayda vardır ve bu değer gelmiş geçmiş tüm kromozomların toplam sapmasından daha küçüktür. Bu değer, “en iyi çözümün toplam sapması” olarak, daha sonra kullanılmak üzere kaydedilir.

Genetik algoritma, problem çözmenin deterministik bir yaklaşımı olmaktan ziyade daha çok stokastik bir yaklaşımdır. Stokastik yaklaşımların bir takım sakıncalarını bertaraf etmek için diğer başka metodlara ihtiyaç duyulmaktadır. Monte Carlo denemeleri, birden fazla stokastik çözüm yaparak daha sağlıklı bir çözüme ulaşmamızı sağlayan güçlü bir methodtur. Bu sebeple, Genetik Algoritma Motorunu yukarıda bahsettiğimiz şekilde aynı kerteriz verilerini kullanarak, birden fazla kez, birbirinden bağımsız şekilde çalıştırarak bir çeşit Monte Carlo simülasyonu gerçekleştirmeye çalıştık. Monte Carlo denemelerinin sayısını 20 ile 50 arasında bir değer olarak

atadık. Her denemenin diğer denemelerden tamamen bağımsız olan çözümü, toplam sapma değeri ile birlikte kaydedildi. Daha sonra bulunan bu bağımsız çözümlere, toplam sapma değerleri nispetinde birer ağırlık faktörü verildi. HHA parametrelerinin (X_0 , Y_0 , C , S) herbirinin ağırlıklı ortalaması, Monte Carlo denemelerindeki bağımsız çözümlerden elde edilen değerler kullanılarak bulundu. Her Monte Carlo denemesinin ağırlığının nasıl hesaplandığı ve sonuçta HHA parametrelerinin ağırlıklı ortalamalarının ne şekilde bulunduğu aşağıdaki eşitliklerle açıklanmıştır.

$$\lambda_i = \frac{\sum_{j=1}^M \Delta_j}{\Delta_i} \quad (2.8);$$

$$\mu_{X_0} = \frac{\sum_{i=1}^M (X_{0(i)} \times \lambda_{(i)})}{\sum_{j=1}^M \lambda_{(j)}} \quad (2.9);$$

$$\mu_{Y_0} = \frac{\sum_{i=1}^M (Y_{0(i)} \times \lambda_{(i)})}{\sum_{j=1}^M \lambda_{(j)}} \quad (2.10);$$

$$\mu_C = \frac{\sum_{i=1}^M (C_{(i)} \times \lambda_{(i)})}{\sum_{j=1}^M \lambda_{(j)}} \quad (2.11);$$

$$\mu_S = \frac{\sum_{i=1}^M (S_{(i)} \times \lambda_{(i)})}{\sum_{j=1}^M \lambda_{(j)}} \quad (2.12);$$

M : Monte Carlo denemelerinin sayısı

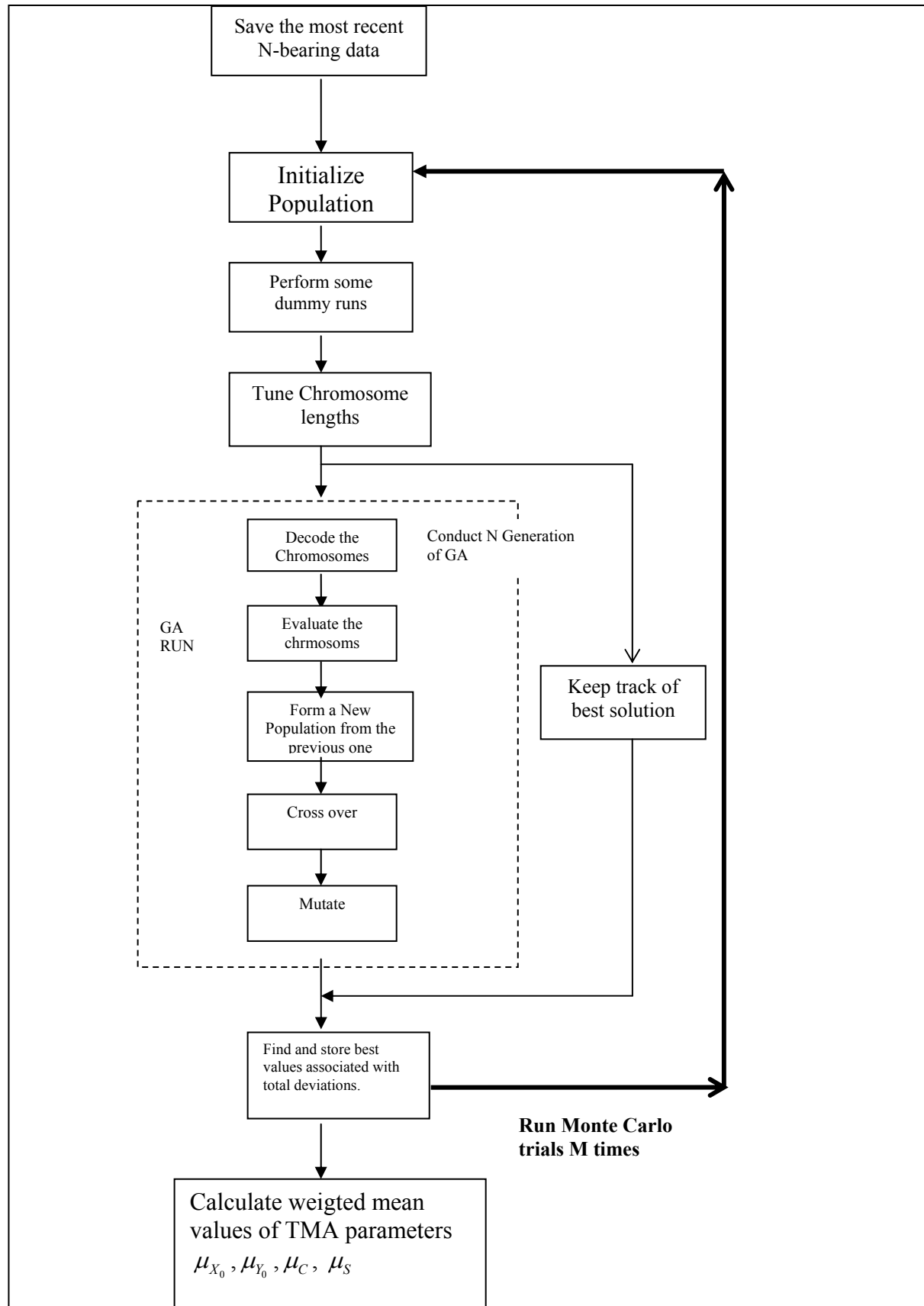
Δ_i : Her denemeye ait en iyi çözümün toplam sapması

λ_i : Her çözüme ait ağırlık faktörü

$\mu_{X_0}, \mu_{Y_0}, \mu_C, \mu_S$: HHA parametrelerinin ağırlıklı ortalamaları.

$\mu_{X_0}, \mu_{Y_0}, \mu_C, \mu_S$ değerleri TMAGA'nın meyveleri, nihayetinde bulmak istediğimiz değerlerdir. Bu değerler aranan hedef parametreleridir.

TMAGA'nın işyeşinin en genel hatlarıyla Şekil 2.11 'de gösterilmiştir.



Şekil 2.11. TMAGA Akış Diyagramı

2.4. Denemeler

Bu bölümde, algoritmamızın geçerliliğini ve başarısını göstermek amacıyla gerçekleştirdiğimiz bir takım denemeleri sonuçları gösterilecek ve açıklanacaktır. Bunun yanında ölçüm kalitesinin yani ölçümler sonucu elde edilen kerterizlerdeki hata miktarının ve genetic algoritmanın olasılık elementlerinin, yani çaprazlama ve mutasyon olasılıklarının çözümler üzerindeki etkisi de bir kaç deneme ile gösterilecektir.

Yaptığımız tüm denemelerde kendi gemi süratimizi sabit tuttuk. Dalmış durumda hareket icra eden bir dizel tahrikli denizaltı gemisinin operasyonel süratine çok yakın bir değer olan 2 m/s sürati, kendi gemi süratimiz olarak tatbik ettik. Bunun yanında, hedefleri sabit sürat ile sabit rota da hareket ediyor farz ettik ve gözlemlenebilirlik durumunu sağlamak amacıyla da kendi gemimize manevra yaptırдық. Manevra yapan kendi gemimiz, her bacakta 12 şer adet olmak üzere toplam 24 adet ölçümü METEOR'dan 15'şer saniye farkla aldı. Böylece TMAGA'nın veri toplama süresini 6 dakika olarak kabul ettik.

Denemeler Y eksenini ters çevrilmiş (pozitif taraf yerine negatif taraf yukarı doğru) bir X-Y koordinat sisteminde gerçekleştirildi. Kendi gemi ve hedef mevkiileri bu koordinat sistemine göre tanımlandı. Mesafe ve sürat ölçümlerinde metrik sistem kullanıldı ve mesafeler metre, süratler metre/saniye olarak ölçüldü. *Populasyon boyutu, nesil sayısı ve Monte Carlo denemeleri sayısı* sırasıyla 50, 500 ve 20 kabul edildi. Diğer taraftan normal denemeler boyunca *çaprazlama olasılığı* 0.2, *mutasyon olasılığı* ise 0.1 olarak atandı.

Her denemede tek bir sonuç yeterli olmasına rağmen çözümler arasında bir tutarlılık olduğunu sergilemek maksadıyla, problem birbirinden bağımsız şekilde 20 defa tekrar çözüldü. Her bağımsız çözüm, denemeleri gerçekleştirdiğimiz Pentium 4, Windows XP işletim sistemli kişisel bilgisayarda 1 ila 3 dakika arası bir zaman aldı.

Tüm bağımsız çözümler ilgili çizelgelerde listelenmiştir. Ayrıca bu çözümlerin ortalaması, bu ortalamanın mutlak doğrudan olan mutlak değer sapması ve yüzdesel sapması da bulunarak çizelgelerin sonuna iliştilmiştir. Bir karar değişkeninin mutlak doğrudan yüzdesel sapmasını bulmak için, bu değişkenin mutlak değer sapması bu değişkene ait maksimum ve minimum değerler arasındaki farka bölünmüştür. Bu sebeple X0, Y0, Rota and Sürat karar değişkenlerinin her birinin yüzdesel sapmalarını bulmak için ilgili mutlak değer sapması karar değişkenine göre 20000m, 20000m, 360 derece veya 25 m/s değerine bölünmüştür.

Ayrıca ortalama çözüm bir adet MS Excel grafiğine konularak mutlak doğru ile genetik algoritma çözümü arasındaki farkın görsel karşılaştırılabilmesine imkan sağlanmıştır. Kendi gemi hareketimizi de bu grafiğe koyduğumuzdan, bu grafiklerden problemin çözüldüğü andaki taktik resmi görmek te mümkün olmaktadır.

Aynı kategorideki denemelere ait çizelgeler ve MS Excel grafikleri birleştirilerek yerden tasarruf edilmiş ve okuyucuya, değişik deneme sonuçlarını görsel olarak aynı anda karşılaştırabilme imkanı sağlanmıştır..

2.4.1. TMAGA Geçerlilik Testleri

Geçerlilik testlerinde değişik kinematik parametrelere sahip hedeflere ait HHA problemlerini çözdük. Hedef kinematik parametreleri, gerçek hayatta en yaygın HHA vakalarını temsil edecek şekilde seçilmiştir (Deneme 1-3). Ayrıca farklı gözlemlenebilirlik durumlarındaki TMAGA başarısı hedef rota parametresiyle oynanarak Deneme 4-6 da test edilmiştir.

2.4.1.1. Deneme 1 - 3

Deneme 1 den Deneme 3'e kadar olan testlerde, değişik kerteriz değişimleri yaratan hedeflere ait HHA problemleri çözülmüştür. Tüm denemler boyunca kendi gemi hareketlerimiz değişmemiştir. Kendi gemimize ait bu hareket parametreleri Çizelge 2.2'de verilmiştir. Deneme 1'de kendi gemimizden 8000 m mesafeye yerleştirilmiş bir hedef doğu yönüne (090⁰)

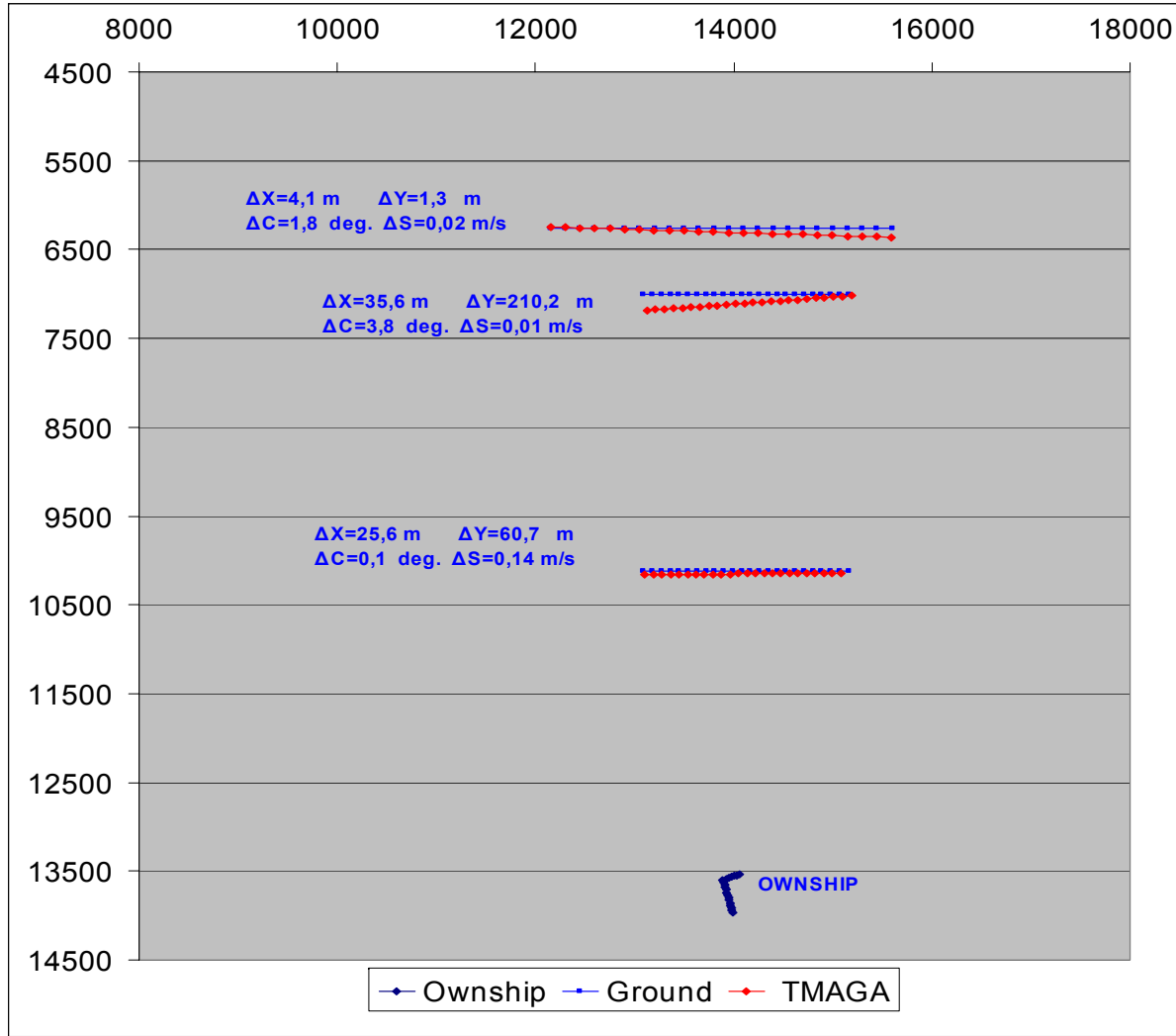
dođru 10 m/s (18 kts) hızla ilerlemektedir. Bu durumdaki kerteriz deđişim miktarı normal ölçülerdedir (ne çok düşük ne de çok yüksek). Deneme 2 'de hedef 7200 m. mesafeye çekilerek hızı 10 m/s 'den 6 m/s 'ye düşürülmüş ve böylece hedefin kerteriz deđişim miktarı azaltılmıştır. Deneme 3 'te ise rota ve sürat Deneme 2'deki deđerlerle aynı kalmış, ancak mesafe 7200 m.'den 4000 m.' ye çekilerek kerteriz deđişim miktarı artırılmak istenmiştir. Çizelge 2.3 'de her denemede icra edilen 20 bađımsız koşmanın sonuçları, bu sonuçların ortalaması ve mutlak doğrudan olan sapma deđerleri (mutal deđer sapması ve yüzdesel sapma) sergilenmiştir.

Kendi Gemimiz Kinematik Deđerleri	STD = 0.3
Sürat = 2 m/s	
$X_0 = 14000, Y_0 = 14000$	
BACAĞ 1	BACAĞ 2
Rota = 345	Rota = 70
Her bacadta 12 kerteriz toplanmıştır,	
15 saniyede bir ölçüm	

Çizelge 2.2: Deneme 1 – 3 arası kendi gemi kinematik parametreleri

#	DEN 1				DEN 2				DEN 3			
	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat
1	11940,6	6011,1	91,2	10,31	12994,7	6942,2	81,4	6,30	12990,2	10050,4	89,6	6,07
2	11950,1	6053,0	91,2	10,26	13003,2	6981,4	85,0	6,18	13006,5	10076,7	91,8	5,97
3	11958,5	6089,7	91,4	10,21	13020,3	7014,9	93,0	5,99	13002,9	10095,6	89,0	6,01
4	11969,6	6121,5	92,0	10,15	13028,4	7110,3	90,3	6,12	13003,9	10115,0	87,5	6,03
5	11978,2	6162,7	91,5	10,11	13019,7	7127,7	83,8	6,16	13018,9	10125,2	91,1	5,90
6	11986,1	6196,6	91,3	10,07	13020,3	7130,4	83,9	6,15	13016,9	10137,5	90,3	5,91
7	11985,8	6197,5	91,3	10,08	13034,6	7147,7	88,5	6,04	13017,0	10139,5	90,1	5,91
8	11993,5	6202,9	92,2	10,03	13025,8	7180,4	82,9	6,05	13012,6	10144,8	88,6	5,95
9	11988,8	6204,0	91,6	10,06	13029,8	7187,4	87,5	6,00	13009,4	10164,3	86,7	5,99
10	12003,4	6254,8	91,9	9,98	13036,9	7208,2	86,9	5,96	13023,2	10177,8	89,4	5,87
11	12010,5	6281,1	92,3	9,94	13036,2	7231,3	82,4	6,03	13033,2	10196,2	90,8	5,79
12	12008,6	6288,8	91,6	9,96	13038,9	7234,8	86,7	5,92	13029,1	10198,2	90,7	5,83
13	12019,3	6313,6	91,9	9,90	13043,4	7242,1	87,3	5,95	13023,5	10200,8	88,1	5,88
14	12019,0	6320,4	91,8	9,91	13048,9	7262,0	90,5	5,90	13037,5	10222,2	91,0	5,76
15	12024,8	6331,6	92,3	9,87	13042,4	7282,5	82,2	6,03	13042,8	10235,1	90,9	5,74
16	12036,8	6381,8	92,4	9,81	13045,8	7301,5	85,3	5,96	13042,0	10244,9	90,0	5,76
17	12043,5	6405,2	92,0	9,79	13055,0	7344,3	86,6	5,87	13039,9	10245,0	88,9	5,78
18	12045,6	6421,3	91,9	9,77	13055,3	7378,8	86,5	5,92	13045,4	10259,1	90,4	5,73
19	12059,9	6468,6	92,3	9,70	13063,8	7448,4	85,5	5,82	13048,2	10259,6	90,3	5,71
20	12063,3	6480,8	92,3	9,68	13069,7	7470,8	87,4	5,83	13069,8	10324,4	92,4	5,55
Mutlak Doğru Ort.	12000	6260	90	10	13000	7000	90	6	13000	10120	90	6
Çözüm Mut.	12004,1	6258,7	91,8	9,98	13035,6	7210,2	86,2	6,01	13025,6	10180,7	89,9	5,86
Sapma	4,1	1,3	1,8	0,02	35,6	210,2	3,8	0,0	25,6	60,7	0,1	0,1
% Sap.	0,02%	0,01%	0,51%	0,08%	0,14%	0,84%	1,05%	0,04%	0,10%	0,24%	0,04%	0,57%

Çizelge 2.3. Deneme1-3 Test Sonuçları



Şekil 2.12. Deneme 1 - 3 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir.

Grafikten de görüleceği üzere, TMAGA kerteriz değişimin nisbi olarak en yüksek olduğu vakada (en yakın hedef) daha başarılıdır. TMAGA'nın en az başarılı olduğu hedef ise ortadaki hedeftir (kerteriz değişimi en düşük olan hedef bu hedeftir). Ancak yine de buradaki tüm testlerde TMAGA'nın oldukça başarılı olduğu söylenebilir.

2.4.1.2. Deneme 4 - 6:

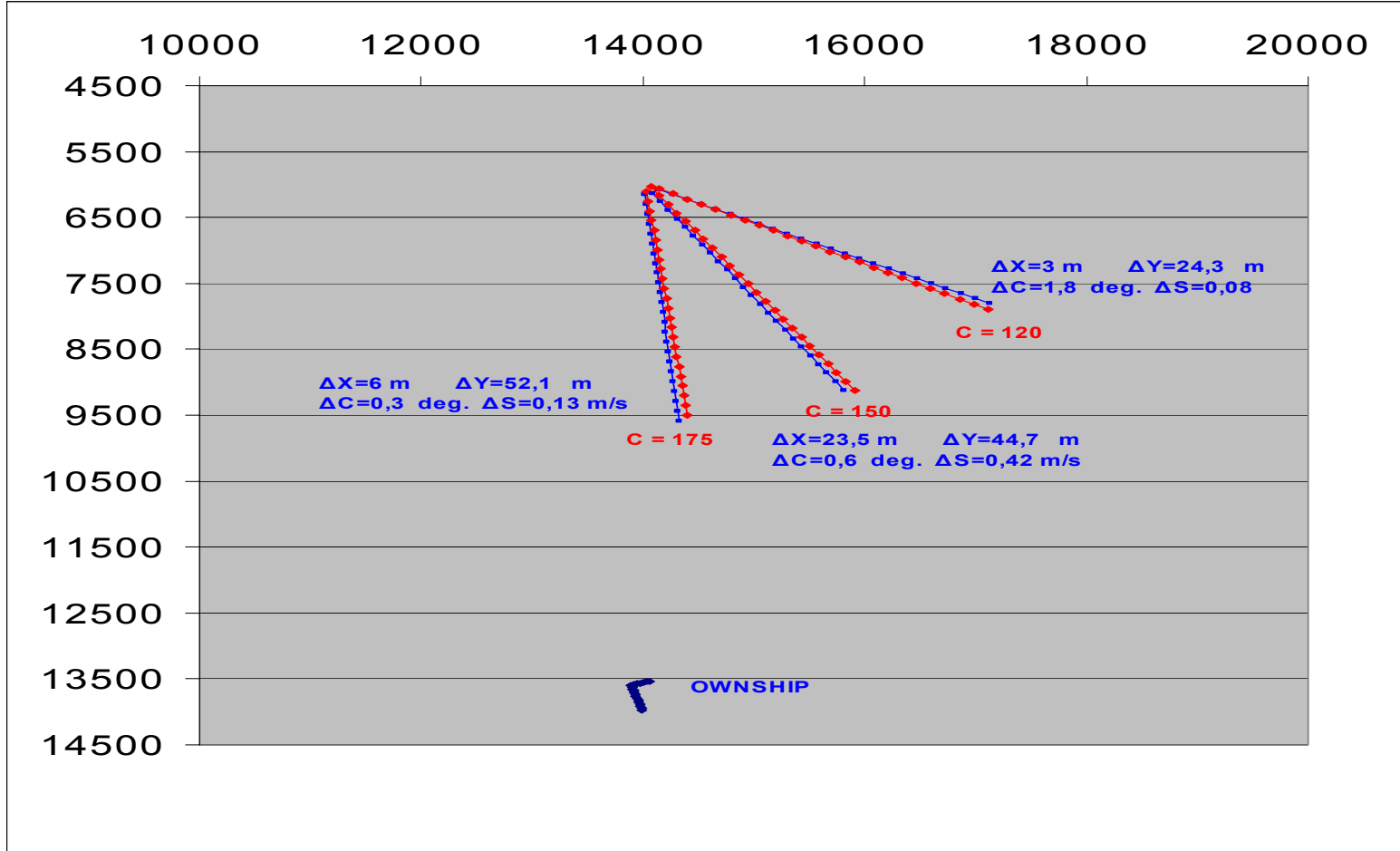
Bu denemelerde, değişik vaziyet zaviyelerine sahip hedeflere ait HHA problemleri çözülmüştür. Bu 3 testde, hedefler aynı noktadan harekete başlarlar ve kendi gemimize doğru sırasıyla 5° ($C= 175^{\circ}$), 30° ($C= 155^{\circ}$) ve 60° ($C= 120^{\circ}$) vaziyet zaviyeleriyle ilerlerler. Hedef süratleri sabittir ve değerleri 10 m/s 'dir (18 kts).

Kendi Gemimiz Kinematik Değerleri	STD = 0.3
Sürat = 2 m/s	
$X_0 = 14000$, $Y_0 = 14000$	
BACAK 1	BACAK 2
Rota = 345	Rota = 70
Her bacakta 12 kerteriz toplanmıştır, 15 saniyede bir ölçüm	

Çizelge 2.4. Deneme 4 – 6 arası kendi gemi kinematik parametreleri

#	DEN				DEN				DEN			
	4				5				6			
	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat
1	14012,5	5848,2	174,3	9,26	13981,5	5777,8	148,7	10,55	14013,2	5849,4	121,5	10,24
2	14017,4	5874,5	175,1	10,04	13982,7	5821,9	148,8	10,51	14011,5	5849,8	121,4	10,25
3	14019,2	5875,4	175,4	10,40	13983,9	5849,8	148,9	10,47	14011,8	5879,6	121,4	10,20
4	14016,9	5880,2	174,7	9,59	13985,7	5865,1	149,3	10,51	14014,2	5916,7	121,9	10,16
5	14016,9	5919,4	175,0	9,84	13984,8	5895,4	149,1	10,46	14013,0	5937,6	121,8	10,13
6	14016,3	5930,3	174,9	9,75	13984,1	5899,4	149,0	10,44	14012,3	5949,7	121,8	10,12
7	14016,1	5931,4	174,8	9,61	13985,4	5916,0	149,3	10,46	14014,2	5958,4	122,0	10,11
8	14018,2	5933,2	175,3	10,23	13990,0	5927,3	149,6	10,47	14013,4	5961,5	121,8	10,10
9	14013,3	5945,3	174,5	9,36	13988,3	5927,8	149,5	10,46	14013,6	5972,1	122,0	10,09
10	14017,1	5945,6	174,9	9,67	13985,3	5938,5	149,2	10,41	14013,0	5979,4	121,9	10,08
11	14015,8	5956,3	175,1	10,05	13983,2	5945,0	149,0	10,38	14012,3	5996,3	122,0	10,07
12	14016,1	5957,7	175,1	10,02	13985,6	5954,4	149,3	10,40	14012,0	5998,5	121,6	10,05
13	14016,6	5965,5	175,1	9,96	13987,5	5978,9	149,5	10,40	14012,0	5998,6	121,7	10,05
14	14015,4	5971,4	174,7	9,41	13990,5	5984,7	150,0	10,47	14012,1	6001,6	121,6	10,04
15	14016,8	5974,9	175,1	9,86	13990,3	5986,3	149,9	10,44	14013,0	6017,4	121,9	10,03
16	14016,6	5977,7	175,0	9,83	13988,6	6046,1	149,8	10,37	14013,0	6021,1	122,1	10,03
17	14016,7	5978,1	175,2	10,09	13989,3	6068,0	149,9	10,34	14013,2	6023,5	121,9	10,02
18	14016,4	6008,6	175,5	10,45	13987,4	6071,2	149,6	10,30	14014,3	6056,1	122,1	9,98
19	14019,1	6009,9	175,7	10,65	13987,2	6082,3	150,0	10,36	14012,2	6066,0	121,9	9,97
20	14007,9	6055,4	177,9	9,47	13988,6	6175,4	150,2	10,25	14015,2	6082,1	122,3	9,95
Mutlak Doğru Ort.	14010	6000	175	10	14010	6000	150	10	14010	6000	120	10
Çözüm Mut.	14016,0	5947,9	174,7	9,87	13986,5	5955,6	149,4	10,42	14013,0	5975,7	121,8	10,08
Sapma	6,0	52,1	0,3	0,13	23,5	44,4	0,6	0,42	3,0	24,3	1,8	0,08
% Sap.	0,02%	0,21%	0,09%	0,51%	0,09%	0,18%	0,15%	1,69%	0,01%	0,10%	0,51%	0,33%

Çizelge 2.5. Deneme 4-6 test sonuçları



Şekil 2.13. Deneme 4 - 6 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir. .

Vaziyet zaviyesinin 5^0 derece olduğu vakanın gözlemlenebilirlik derecesi çok düşük olmasına rağmen, TMAGA bu vakada bile gayet başarılıdır. Sonuç olarak değişik vaziyet zaviyeleri tatbik edilen hedeflerin HHA çözümlerinde TMAGA yüksek oranda başarı göstermiştir.

2.4.2. Ölçüm Verisi Kalitesinin Etkileri

Bu kısımda HHA hesaplamasında kullanılan ölçümlere ait hata miktarlarının algoritma üzerindeki etkileri gösterilecektir. Beklendiği gibi ölçüm hataları arttıkça, HHA çözümü mutlak doğrudan daha fazla miktarda sapacaktır. Aşağıdaki denemelerde, farklı derecelerdeki hata ile kirlenmiş ölçümler ile yapılan HHA çözümleri gösterilecektir. Deneme 7 'den Deneme 12 'y kadar hedef ve kendi gemi parametrelerimiz değişmeyecek sabit kalacaktır (Bkz. Çizelge 2.6). Değişen tek şey, çözümlerde kullanılan ölçümlerin içerdiği hataların standart sapmalarıdır. İlk denemede sıfır hatalı ($\sigma = 0$) ölçümler ile yapılan çözüm test edilecektir. Daha sonraki testlerde hata standart sapmaları sırasıyla 0.5, 1, 2, 5 ve 30 derece olan ölçümler kullanılmaktadır.

Kendi Gemimiz Kinematik Değerleri	STD = 0, 0.5, 1, 2, 5, 30
Sürat = 2 m/s	
$X_0 = 14000, Y_0 = 14000$	
BACAĞ 1	BACAĞ 2
Rota = 345	Rota = 70
Her bacadaki 12 kerteriz toplanmıştır, 15 saniyede bir ölçüm	

Çizelge 2.6. Deneme 7 – 12 arası kendi gemi kinematik parametreleri

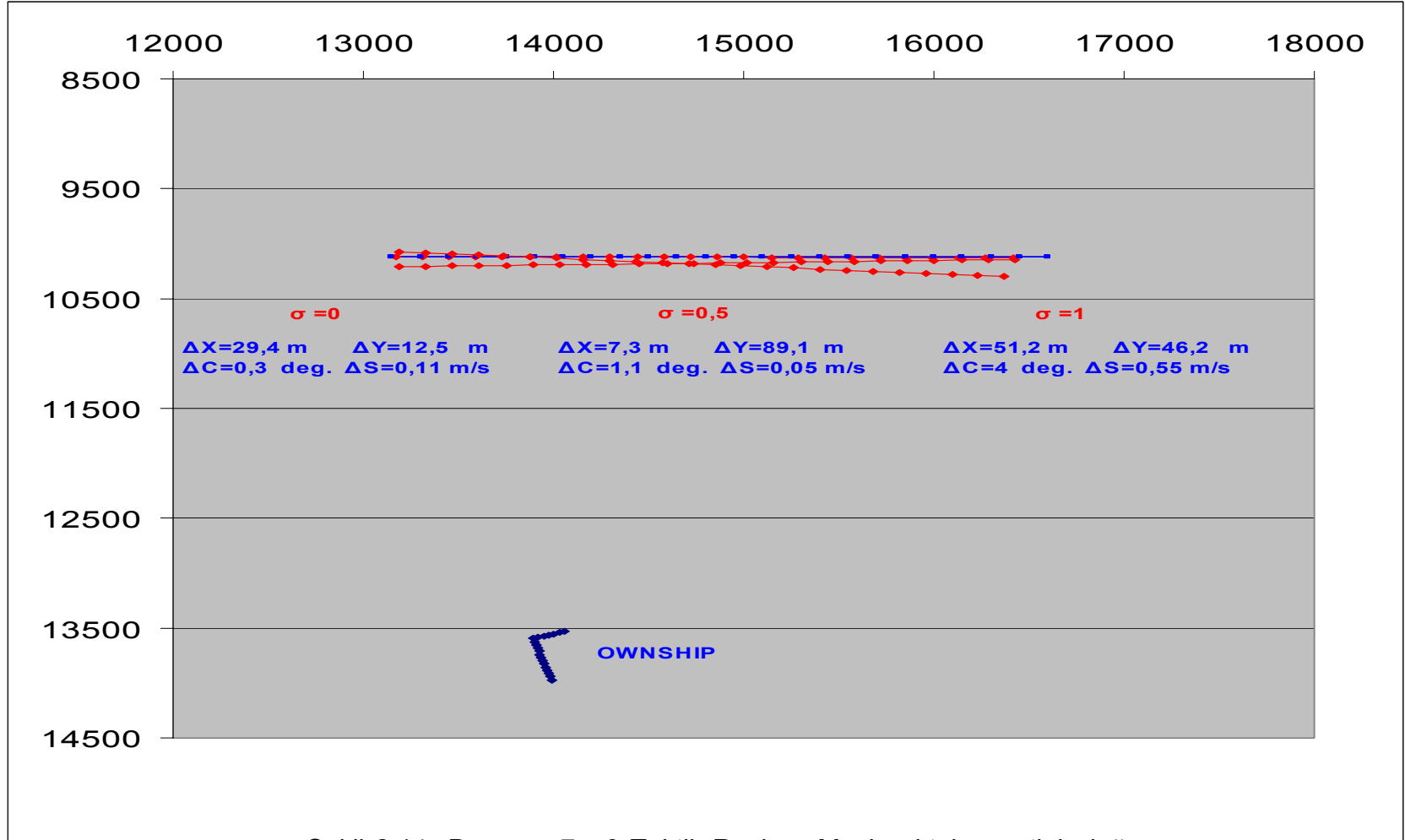
2.4.2.1. Deneme 7 - 9:

Bu kısımdaki denemelerde ölçüm hataları nispeten düşüktür ve su altı ölçümleri için kabul edilebilir ölçülerdedir.

Sıfır hatalı ölçüm vakasını Deneme 7 'de test edeceğiz. Bu testte kullanılan kerterziler, kendi gemi merkezimizden gerçek hedef merkezine çizilen doğru parçalarıdır. Deneme 8 ve Deneme 9 'da ise hata standart sapmaları sırasıyla 0,5 ve 1 derecedir.

#	DEN				DEN				DEN			
	7		$\sigma=0$		8		$\sigma=0,5$		9		$\sigma=1$	
	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat
1	13000,5	10021,1	90,1	9,99	12962,2	10044,9	88,6	10,44	13036,5	10056,5	94,7	9,64
2	12999,8	10127,7	89,7	10,01	12977,5	10103,5	88,7	10,26	13043,9	10126,3	94,3	9,51
3	13005,2	10150,9	89,6	9,95	12992,8	10146,6	88,8	10,13	13050,1	10129,0	94,6	9,48
4	13012,8	10167,1	89,9	9,97	12996,5	10156,7	89,0	10,07	13043,7	10135,8	94,3	9,50
5	13012,9	10171,6	89,9	9,96	12989,9	10158,6	88,5	10,14	13041,8	10138,0	93,6	9,57
6	13023,3	10095,9	90,1	9,86	12993,9	10162,7	88,8	10,08	13049,6	10140,2	94,9	9,45
7	13022,5	10105,0	89,9	9,96	13002,1	10189,8	88,8	10,01	13049,2	10142,7	94,5	9,47
8	13025,7	10117,7	89,9	9,82	13005,1	10192,3	89,1	9,97	13043,8	10152,1	93,6	9,54
9	13025,9	10118,4	89,7	9,93	13008,9	10216,3	89,1	9,91	13042,5	10155,0	93,8	9,52
10	13029,4	10232,5	89,7	9,89	13014,3	10227,6	89,0	9,88	13053,8	10162,5	93,8	9,46
11	13034,5	10248,2	89,8	9,93	13011,1	10228,3	88,7	9,91	13048,5	10162,9	93,8	9,48
12	13034,4	10149,9	89,8	9,84	13015,8	10237,1	89,2	9,84	13054,3	10172,1	94,2	9,41
13	13041,4	10176,6	89,7	9,86	13014,6	10238,4	88,8	9,88	13052,0	10187,1	94,0	9,42
14	13040,2	10178,1	89,6	9,87	13014,8	10248,1	88,8	9,85	13058,3	10187,8	94,0	9,39
15	13041,5	10283,4	89,6	9,86	13019,2	10249,9	88,8	9,83	13048,6	10187,9	94,0	9,43
16	13043,6	10183,6	89,5	9,85	13018,4	10254,3	88,9	9,82	13061,5	10192,7	93,6	9,40
17	13044,1	10191,2	89,5	9,84	13019,2	10255,2	88,8	9,83	13060,1	10199,0	94,1	9,36
18	13047,5	10202,1	89,5	9,80	13026,5	10266,9	89,2	9,74	13061,4	10217,6	94,0	9,33
19	13049,4	10209,7	89,6	9,78	13027,5	10296,2	88,8	9,72	13062,6	10234,1	93,1	9,34
20	13049,8	10209,9	89,5	9,78	13035,8	10309,2	89,0	9,65	13061,4	10243,8	93,7	9,30
Mutlak Doğru Ort.	13000	10120	90	10	13000	10120	90	10	13000	10120	90	10
Çözüm Mut.	13029,4	10132,5	89,7	9,89	13007,3	10209,1	88,9	9,95	13051,2	10166,2	94,0	9,45
Sapma	29,4	12,5	0,3	0,11	7,3	89,1	1,1	0,05	51,2	46,2	4,0	0,55
% Sap.	0,12%	0,05%	0,07%	0,03%	0,03%	0,36%	0,32%	0,21%	0,20%	0,18%	1,12%	2,20%

Çizelge 2.7. Deneme 7-9 test sonuçları



Şekil 2.14 Deneme 7 - 9 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir.

Şekil 2.14 ve Çizelge 2.7 'den de görüleceği üzere bu denemelerin başarıları arasında önemli bir fark bulunmamaktadır. Deneme 7'nin sonucunun mutlak doğru ile bire bir aynı olmasının beklenmesine rağmen, durum öyle değildir. Bu deneme, genetik algoritmanın bir stokastik problem çözme metodu olduğunun güzel bir göstergesidir. Genetik algoritma deterministik olmadığı için her zaman en iyi çözümü garanti etmez.

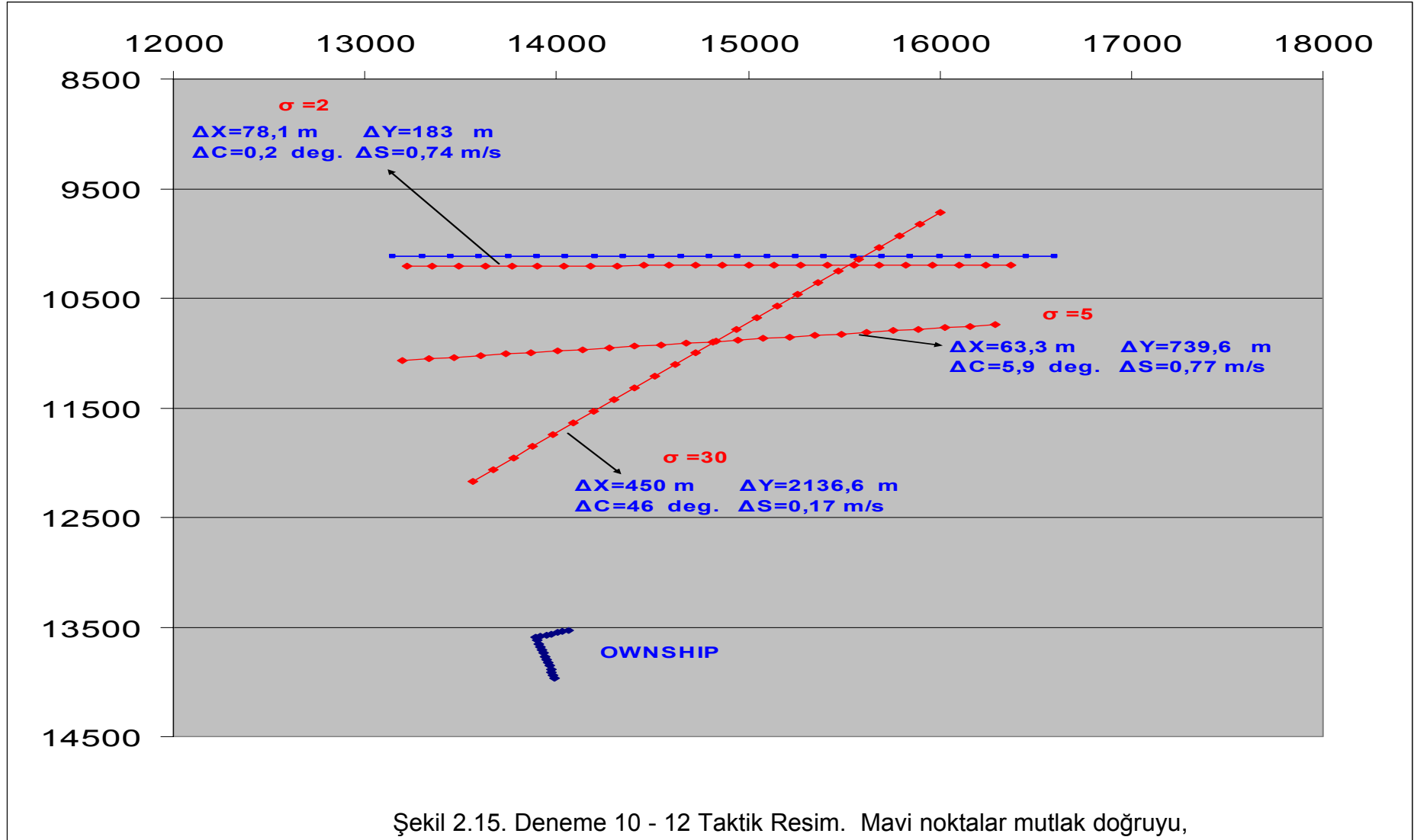
Bu kısımda yaptığımız testlerde, hata standart sapmasının 1 derece (± 3 derece) olduğu vakada bile TMAGA başarılı bir çözüm gerçekleştirmiştir.

2.4.2.2. Deneme 10 - 12

Bu bölümdeki denemelerde ölçüm hataları kayda değer derecede yüksektir. Ölçüm hatalarının standart sapmaları sırasıyla 2, 5 ve 30 (± 90 derece!) dur. Gerçek hayatta böylesine yüksek bir hata miktarına sahip ölçümlerin kabul edilmesi zordur.

#	DEN 10				DEN 11				DEN 12			
	X ₀	Y ₀	Rota	Sürat	X ₀	Y ₀	Rota	Sürat	X ₀	Y ₀	Rota	Sürat
1	13065,2	10178,5	91,6	9,40	12990,5	10685,6	83,9	9,91	13374,2	11987,4	45,2	10,69
2	13046,6	10207,3	89,2	9,59	13022,7	10714,7	84,9	9,63	13423,8	12090,1	46,5	9,83
3	13075,7	10224,3	90,9	9,33	13047,9	10742,7	84,5	9,45	13425,4	12090,7	45,3	10,44
4	13059,5	10228,6	90,2	9,46	13050,1	10764,5	84,9	9,33	13446,8	12101,3	46,2	9,63
5	13070,5	10265,5	90,0	9,34	13044,6	10777,0	87,0	9,14	13406,0	12110,0	44,9	10,41
6	13079,6	10268,6	90,5	9,27	13050,2	10796,8	84,8	9,30	13464,8	12133,7	46,1	9,52
7	13083,7	10279,3	90,4	9,24	13053,2	10797,3	85,0	9,36	13392,1	12135,1	44,7	10,67
8	13084,9	10294,7	90,0	9,21	13058,4	10805,8	85,1	9,27	13428,9	12175,2	47,2	9,44
9	13081,6	10295,3	89,5	9,29	13031,8	10807,6	82,7	9,63	13473,4	12194,6	43,7	10,11
10	13077,0	10321,2	88,5	9,34	13049,3	10814,4	83,2	9,46	13425,9	12208,8	44,5	9,63
11	13072,1	10324,3	89,2	9,29	13075,3	10848,8	85,7	9,07	13441,2	12239,1	44,0	9,82
12	13088,7	10329,9	90,5	9,10	13058,9	10865,8	82,1	9,40	13426,6	12256,6	42,0	10,75
13	13082,6	10331,4	89,3	9,24	13092,2	10917,4	84,1	8,93	13481,3	12269,6	42,4	9,71
14	13075,9	10334,9	88,7	9,32	13075,4	10918,6	83,3	9,12	13534,9	12328,4	44,8	8,20
15	13084,4	10336,7	89,4	9,21	13086,9	10949,4	83,0	9,11	13452,5	12371,2	44,0	9,98
16	13074,5	10347,0	89,7	9,20	13100,7	10952,3	84,5	8,90	13481,1	12380,8	42,4	9,77
17	13086,2	10359,6	89,9	9,14	13066,1	10953,8	84,7	8,98	13460,5	12380,9	41,8	10,01
18	13094,0	10373,5	89,2	9,11	13078,1	10981,6	84,5	8,94	13464,8	12408,1	43,2	9,26
19	13090,0	10377,9	89,8	9,07	13113,5	11043,2	83,0	8,75	13460,7	12498,4	41,9	9,82
20	13089,2	10380,8	89,4	9,12	13100,0	11054,1	81,8	8,96	13550,7	12772,2	40,0	8,95
Mutlak Doğru Ort.	13000	10120	90	10	13000	10120	90	10	13000	10120	90	10
Çözüm Mut.	13078,1	10303,0	89,8	9,26	13062,3	10859,6	84,1	9,23	13450,8	12256,6	44,0	9,83
Sapma	78,1	183,0	0,2	0,74	62,3	739,6	5,9	0,77	450,8	2136,6	46,0	0,17
% Sap.	0,31%	0,73%	0,05%	2,95%	0,25%	2,96%	1,63%	3,07%	1,80%	8,55%	12,77%	0,67%

Çizelge 2.8. Deneme 10-12 Test Sonuçları



Şekil 2.15. Deneme 10 - 12 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir.

TMAGA, standart sapmanın 2 derece (± 6 derece) olduđu vakada bile hala başarı ile çözüm sağlamaktadır. Çözümlerin sadece 24 adet ölçüm ile yapıldığı unutulmamalıdır. Böyle yüksek bir hata miktarı ve az sayıda ölçümle, Kalman Filtresi tabanlı bir metotta filtrenin başlatılması bir mümkün olamayacaktır.

TMAGA standart sapmanın 5 derece olduđu deneme 11 de bile başarılı farz edilebilir. Hedef mutak doğrunun yakın çevresi içerisinde tahmin edilmiştir. Bu çözüm aktif torpido fırlatmaya hazırlanan bir denizaltı için yeterli bir çözümdür.

Deneme 12 de ise hata miktarı aşırı derecede yüksek (± 90 derece) bir ölçüm seti kullanılmıştır. Bu deneme sadece TMAGA'nın aşırı gürültülü ortamlardaki davranışını gözlemleyebilmek maksadıyla yapılmıştır. Test sonucunda, pozisyon tahmini oldukça zayıf kalmasına rağmen, rota ve sürat tahminleri nispeten daha iyidir. Tahmin edilen rota, gerçek rota ile aynı çeyrek daire içindedir ve sürat neredeyse gerçek süratin aynısıdır.

2.4.3. Çaprazlama Olasılığının (Crossover Probability) Algoritma Üzerindeki Etkisi

Çaprazlama olasılığının detayları önceki bölümlerde açıklanmıştı. Şu ana kadar olan denemelerde algoritmaya uyguladığımız çaprazlama olasılığı değeri, HHA problemi için deneme yanılma metoduyla keşfettiğimiz ve bize göre optimum değer olan 0.2 değeri idi. Bu kısımdaki denemelerde çaprazlama olasılığının genetik algoritmanın performansını nasıl etklediğini göreceğiz.

Yaptığımız denemelerde, çaprazlama olasılığı artmasına paralel olarak, seçme (selection) işleminin etkisinin azaldığını farkettilik. Çaprazlama olasılığının artmasıyla birlikte çaprazlamaya giren ebeveynlerin ve sonuçta onların yerini alan yavruların sayıları artmakta, bu sebeple arada yüksek uygunluk derecelerine sahip ebeveynlerde ziyan olabilmektedir. Diğer

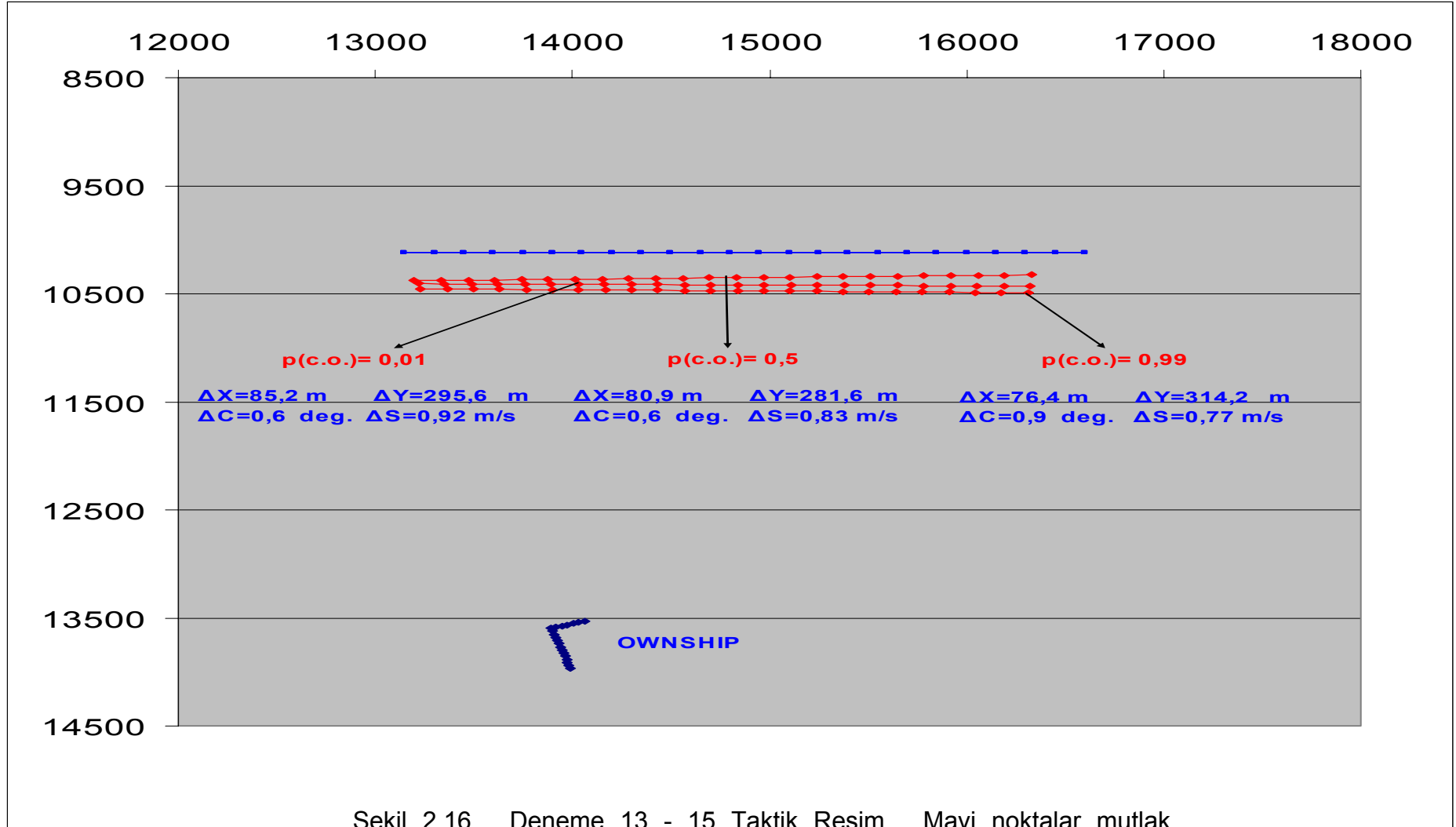
tarafdan aprazlama olasılıđı ok dşk tutulursa, yeni yavrular treyememekte, nesiller arası geiřte kayda deđer deđiřiklikler sađlanamamaktadır. TMAGA'yı tasarlarken yaptığımız testlerde, en uygun aprazlama olasılıđı deđerinin 0.2 olduđunu grdk ve normal testler boyunca bu deđere sadık kaldık.

2.4.3.1. Deneme 13 - 15:

Bu kısımdaki denemelerde nceki blmlerde HHA'sını zdđmz hedefleri kullanacađız. Sırasıyla denemelerimizde genetic algoritmaya 0.01 (ok dřk), 0.5 (nispeten yksek) ve 0.99 (ok yksek) deđerlerinde aprazlama olasılık deđerleri tatbik edeceđiz.

#	DEN 13				DEN 14				DEN 15			
	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat	X₀	Y₀	Rota	Sürat
1	13045,6	10282,3	90,0	9,50	13051,5	10303,8	90,2	9,48	13038,8	10296,6	89,0	9,63
2	13053,0	10309,1	90,2	9,42	13056,6	10317,3	90,3	9,42	13053,9	10370,5	88,6	9,47
3	13070,5	10356,0	90,5	9,26	13063,5	10327,3	90,4	9,38	13058,8	10374,0	88,7	9,44
4	13075,7	10361,5	91,0	9,18	13065,9	10352,2	90,5	9,31	13070,3	10397,3	89,3	9,30
5	13071,5	10373,1	90,5	9,21	13071,3	10362,8	90,7	9,25	13072,3	10408,0	89,3	9,28
6	13074,6	10379,6	90,4	9,19	13076,8	10378,5	90,4	9,23	13070,2	10411,1	89,0	9,30
7	13079,4	10395,9	90,6	9,13	13075,8	10383,9	90,1	9,24	13073,5	10412,4	89,3	9,26
8	13085,9	10401,1	91,1	9,07	13082,5	10399,8	90,6	9,14	13071,2	10414,7	89,1	9,27
9	13086,1	10401,6	91,0	9,07	13077,2	10401,7	89,9	9,23	13073,2	10419,8	89,1	9,27
10	13087,8	10414,4	91,0	9,04	13081,4	10405,1	90,1	9,17	13072,8	10433,7	88,9	9,26
11	13089,2	10418,2	90,8	9,03	13084,5	10411,3	90,5	9,13	13084,3	10438,0	89,8	9,14
12	13085,7	10420,1	90,5	9,07	13084,5	10418,3	90,3	9,13	13078,0	10440,1	89,0	9,22
13	13084,0	10431,9	90,0	9,10	13090,7	10425,2	90,8	9,06	13081,7	10457,9	89,2	9,16
14	13090,6	10441,4	90,2	9,04	13087,1	10425,8	90,2	9,11	13080,6	10470,9	88,8	9,16
15	13090,8	10457,5	90,2	9,02	13093,1	10433,0	90,7	9,04	13086,1	10472,1	89,1	9,12
16	13103,4	10476,2	90,8	8,89	13092,5	10433,8	90,7	9,04	13087,2	10475,4	89,1	9,11
17	13103,4	10478,8	90,8	8,87	13088,9	10439,4	89,9	9,11	13091,7	10485,5	89,2	9,07
18	13108,2	10496,8	90,8	8,83	13095,8	10454,6	90,6	9,00	13090,5	10489,0	89,3	9,07
19	13109,2	10499,4	90,8	8,82	13096,1	10467,1	90,6	8,98	13091,2	10489,1	89,1	9,08
20	13109,7	10514,3	90,4	8,82	13102,6	10490,2	90,6	8,90	13102,9	10531,6	89,3	8,93
Mutlak Doğru Ort.	13000	10120	90	10	13000	10120	90	10	13000	10120	90	10
Çözüm Mut.	13085,2	10415,6	90,6	9,08	13080,9	10401,6	90,4	9,17	13076,4	10434,2	89,1	9,23
Sapma % Sap.	85,2 0,34%	295,6 1,18%	0,6 0,16%	0,92 3,69%	80,9 0,32%	281,6 1,13%	0,4 0,11%	0,83 3,33%	76,4 0,31%	314,2 1,26%	0,9 0,25%	0,77 3,09%

Çizelge 2.9. Deneme 13-15 Test Sonuçları



Şekil 2.16. Deneme 13 - 15 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir.

Tüm bu denemelerde çözümlerin birbirlerine çok yakın oldukları görülmektedir. HHA çözümü için en optimum çaprazlama olasılığı değerini 0.2 olarak bulduğumuzu önceden belirtmiştik. Bu optimum değerden yukarı yada aşağı doğru uzaklaşıldığında, algoritmanın hassasiyetinin düşmeye başladığı müşahade edilmiştir.

Çaprazlama olasılığı değerinin algortima üzerinde beklendiğinin aksine aşırı bir etkisinin olmadığını da görmek ilgi çekicidir. Zira yüksek ve alçak çaprazlama olasılık değerleriyle yapılan testlerde elde edilen sonuçlar gayet başarılıdır ve kabul edilebilirlik sınırları içerisinde.

Özetle bu denemelerden çıkardığımız sonuç, 1-6 arası denemelerde gördüğümüz gibi çaprazlama olasılığı değeri 0.2 olan optimum değere eşitse, çözüm mutlak doğruya çok yakın olacak şekilde çıkmaktadır. Diğer yandan bu değerden uzaklaştıkça, algoritmanın hassasiyeti düşmekte, mutlak doğruya yaklaşılmakta güçlük çekilmektedir.

2.4.4. Mutasyon Olasılığının (Mutation Probability) Algoritma Üzerindeki Etkisi

Mutasyon olasılığının detayları önceki bölümlerde açıklanmıştı. Şu ana kadar olan denemelerde algoritmaya uyguladığımız mutasyon olasılığı değeri, HHA problemi için deneme yanılma metoduyla keşfettiğimiz ve bize göre optimum değer olan 0.1 değeri idi. Bu kısımdaki denemelerde mutasyon olasılığının genetik algoritmanın performansını ne şekilde etkilediğini göreceğiz.

Yaptığımız denemelerde, mutasyon olasılığının artmasıyla birlikte, arama esnasında parametre uzayı içerisinde bülgeden bölgeye atlamaların arttığı teyid edilmiştir. Bu etkinin tercih edilir olmasının beklenmesine rağmen (local minimum noktalarına takılma tehlikesi ortadan kalkmaktadır), bu tür aramalarda aslında parameter uzayının çok yüzeysel tarandığı tespit edilmiştir. Önceden de belirtildiği gibi HHA çözümlerinde, parameter uzayı

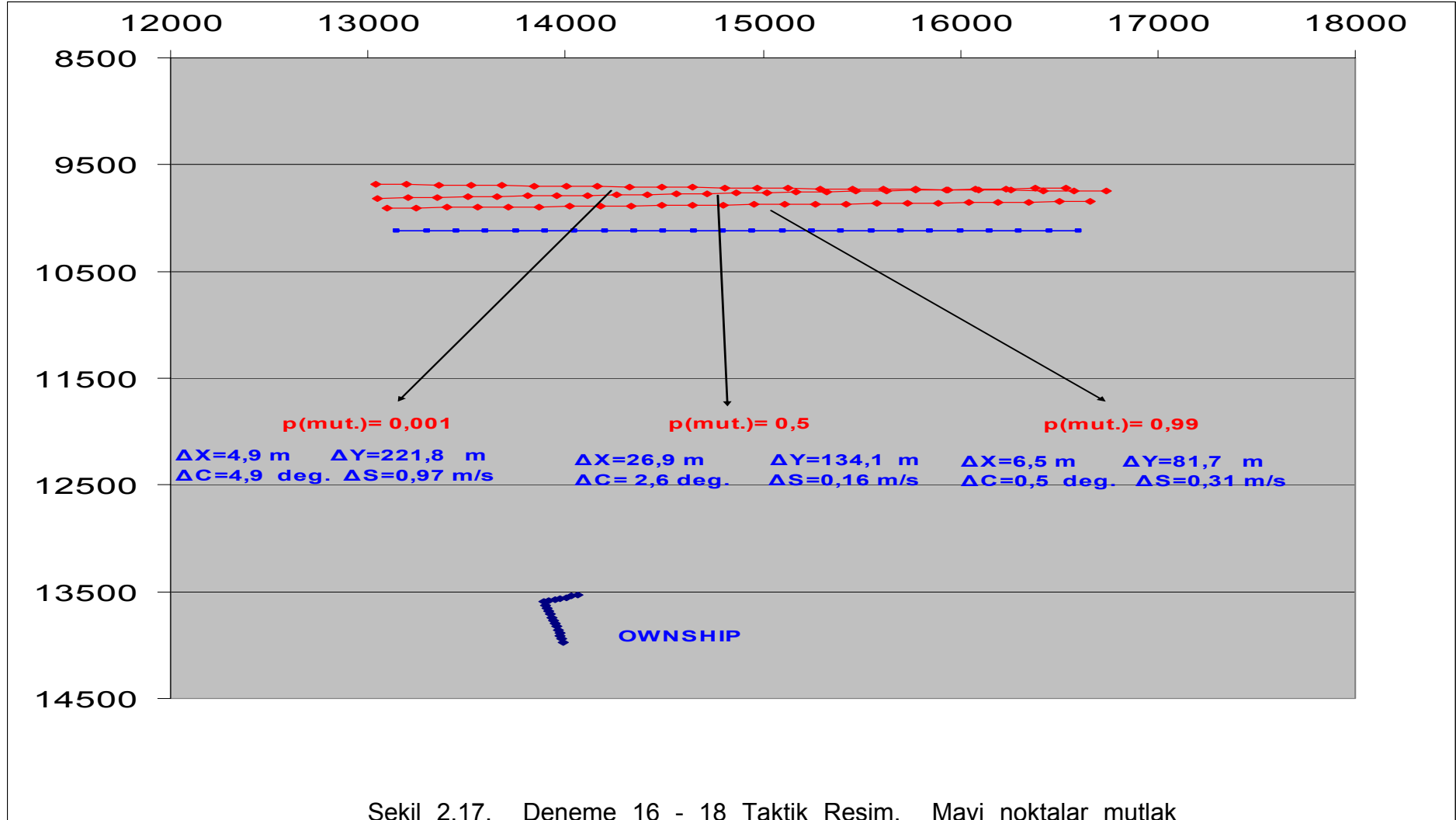
içerisinde bazı bölgelerin bir gradyan arama metodunda olduğu gibi çok detaylı bir şekilde taranması gerekmektedir. Uygulanan yüksek mutasyon olasılığı değeri bazı bölgelerin detaylı taranmasına imkan vermemektedir. Diğer taraftan düşük mutasyon olasılığı, local minimum noktalarına takılma tehlikesini artırmaktadır.

2.4.4.1. Deneme 16 - 18:

Bu kısımdaki denemelerde önceki bölümlerde HHA'sını çözdüğümüz hedefleri kullanacağız. Sırasıyla denemelerimizde genetic algoritmaya 0.001 (çok düşük), 0.5 (nispeten yüksek) ve 0.99 (çok yüksek) değerlerinde mutasyon olasılık değerleri tatbik edeceğiz.

#	DEN 16				DEN 17				DEN 18			
	X_0	Y_0	Rota	Sürat	X_0	Y_0	Rota	Sürat	X_0	Y_0	Rota	Sürat
1	13237,3	8162,7	103,3	11,72	12981,9	9591,2	97,9	10,64	12896,5	9633,5	90,0	11,38
2	13264,8	8582,1	103,4	10,99	12952,3	9764,2	93,2	10,56	12941,3	9705,3	90,3	11,16
3	13175,0	9057,5	109,6	10,70	12923,6	9764,9	90,8	11,07	12933,4	9812,2	90,9	10,75
4	12998,9	9101,6	98,7	11,50	13021,9	9793,8	97,4	10,13	12943,3	9834,5	88,1	11,03
5	12980,5	9184,3	105,0	11,09	13024,3	9830,8	95,4	10,06	12944,3	9885,8	89,8	10,73
6	13042,1	9294,5	103,0	10,90	13049,1	9860,7	94,8	10,28	12932,6	9966,0	87,5	10,91
7	12961,7	9371,0	96,7	11,25	13083,1	9901,2	91,6	10,33	13004,8	9968,0	91,4	10,17
8	13052,7	9377,0	101,5	11,05	13055,6	9903,0	97,2	10,09	12967,6	9978,1	90,0	10,47
9	13154,0	9379,5	92,4	11,11	13000,2	9935,8	93,4	10,26	13085,5	10055,5	94,8	9,61
10	12936,5	9386,8	97,0	11,65	13048,0	9945,2	95,7	9,90	13008,5	10097,4	88,5	10,15
11	13136,8	9420,9	96,2	11,00	12981,7	10034,8	91,4	10,44	13029,0	10128,3	89,6	9,97
12	12919,2	9517,3	90,3	11,05	13004,3	10057,1	90,7	10,21	13003,0	10134,2	88,4	10,15
13	12928,3	9532,3	93,4	11,36	13105,6	10100,2	88,5	10,21	13050,8	10134,8	89,7	10,01
14	12930,0	9634,4	94,4	11,05	13036,2	10121,1	91,2	10,20	13004,2	10142,8	89,2	10,09
15	13029,0	9684,5	93,3	10,82	13062,3	10134,5	91,3	9,90	13045,1	10185,3	89,8	9,76
16	13019,7	9701,6	96,6	10,45	13061,9	10152,4	90,6	9,81	13039,7	10186,6	91,7	9,68
17	13044,5	9711,4	97,4	10,19	13087,1	10189,9	89,2	9,77	13037,5	10260,5	87,3	10,02
18	13097,2	9717,3	92,8	10,66	13025,4	10227,1	90,6	9,77	13013,3	10268,5	85,3	10,20
19	12933,9	9848,9	90,9	10,90	13098,1	10353,2	92,8	9,23	13066,2	10271,2	89,6	9,60
20	13095,2	9891,0	94,2	10,04	13102,7	10472,8	87,1	9,51	13058,4	10337,3	88,0	9,83
Mutlak Doğru Ort. Çözüm Mut. Sapma % Sap.	13000	10120	90	10	13000	10120	90	10	13000	10120	90	10
	13004,9	9898,2	94,9	10,97	13026,9	9985,9	92,6	10,16	12993,5	10038,3	89,5	10,31
	4,9	221,8	4,9	0,97	26,9	134,1	2,6	0,16	6,5	81,7	0,5	0,31
	0,02%	0,97%	1,20%	3,86%	0,11%	0,54%	0,73%	0,66%	0,03%	0,33%	0,14%	1,24%

Çizelge 2.10. Deneme 16-18 Test Sonuçları



Şekil 2.17. Deneme 16 - 18 Taktik Resim. Mavi noktalar mutlak doğruyu, kırmızılar ise TMAGA çözümünü gösterir. .

Denemeler 13-15 'de elde ettiğimiz sonuçların aynısı bu denemelerde de elde ettiğimizi söyleyebiliriz. Mutasyon olasılığı değerinin algoritma üzerindeki etkisi çaprazlama olasılığı değerinin etkisine çok benzemektedir. Optimum mutasyon olasılığı değerinden (0.1) uzaklaştıkça, algoritmanın hassasiyeti benzer şekilde düşmektedir. Düşük ve yüksek mutasyon olasılığı değerleriyle yapılan testlerde elde edilen neticelerin hala başarılı olması, mutasyon olasılığı değerinin de algoritmanın başarısı üzerinde fazla etkisi olmadığını göstergesidir.

Benzer şekilde çözüm ancak optimum mutasyon olasılığı değerinde mutlak doğruya aşırı şekilde yaklaşmakta, diğer alçak ve yüksek değerlerde mutlak doğruya belirli bir mesafede kalmaktadır.

Yapılan denemelerin genel özeti Çizelge 2.11 'de verilmiştir.

#	TEST GİRDİLERİ						SONUÇLAR				
	Mesafe (m)	Vaz. Zav.	Sürat (m/s)	σ	P(çap)	P(mut)	ΔX_0 (m)	ΔY_0 (m)	Δ mesafe (m)	Δ rota (derece)	Δ sürat (m/s)
1	8000	$\approx 80^0$	10	0.3 ⁰	0.2	0.1	4,1	1,3	4	1,8	0,02
2	7200	$\approx 80^0$	6	0.3 ⁰	0.2	0.1	35,6	210,2	213	3,8	0
3	4000	$\approx 80^0$	6	0.3 ⁰	0.2	0.1	25,6	60,7	66	0,1	0,1
4	8000	5 ⁰	10	0.3 ⁰	0.2	0.1	6	52,1	52	0,3	0,13
5	8000	30 ⁰	10	0.3 ⁰	0.2	0.1	23,5	44,4	50	0,6	0,42
6	8000	60 ⁰	10	0.3 ⁰	0.2	0.1	3	24,3	24	1,8	0,08
7	8000	$\approx 80^0$	10	0	0.2	0.1	29,4	12,5	32	0,3	0,11
8	8000	$\approx 80^0$	10	0.5 ⁰	0.2	0.1	7,3	89,1	89	1,1	0,05
9	8000	$\approx 80^0$	10	1.0 ⁰	0.2	0.1	51,2	46,2	69	40	0,55
10	8000	$\approx 80^0$	10	2.0 ⁰	0.2	0.1	78,1	183	199	0,2	0,74
11	8000	$\approx 80^0$	10	5.0 ⁰	0.2	0.1	62,3	739,6	742	5,9	0,77
12	8000	$\approx 80^0$	10	30.0 ⁰	0.2	0.1	450,8	2136,6	2184	460	0,17
13	8000	$\approx 80^0$	10	0.3 ⁰	0.01	0.1	85,2	295,6	308	0,6	0,92
14	8000	$\approx 80^0$	10	0.3 ⁰	0.5	0.1	80,9	281,6	293	0,4	0,83
15	8000	$\approx 80^0$	10	0.3 ⁰	0.99	0.1	76,4	314,2	323	0,9	0,77
16	8000	$\approx 80^0$	10	0.3 ⁰	0.2	0.001	4,9	221,8	222	4,9	0,97
17	8000	≈ 80	10	0.3	0.2	0.5	26,9	134,1	137	2,6	0,16
18	8000	≈ 80	10	0.3	0.2	0.99	6,5	81,7	82	0,5	0,31

Çizelge 2.11 Denemelerin genel özeti

3. ARAŞTIRMANIN SONUÇLARI VE GELECEK ÇALIŞMALAR İÇİN ÖNERİLER

Bu çalışmada, sualtı hedeflerine ait, sadece kerteriz ölçümleri kullanılarak yapılan, Hedef Hareket Analizi problemi bir ters problem örneği olarak ele alınmıştır. Bu, problemin çözümü maksadıyla bir evrimsel hesaplama yöntemi yaklaşımı geliştirilmiş ve bununla birlikte bu yaklaşımın gerçekleştirildiği TMAGA adlı uygulama yazılımı tanıtılmıştır.

Ölçümlerle elde edilen temas bilgilerindeki hata miktarının çok yüksek olduğu ve hedeflerin gözlemlenebilirlik durumlarının düşük olduğu vakalarda, problemin kötü karakteristiklerinin üzerinden gelemek maksadıyla stokastik metodlara ihtiyaç duyulmaktadır. Özellikle sualtı koşulları, ortam gürültüsü miktarını artırmakta, bu koşullarda ölçülen temas bilgileri daha yüksek miktarda hata içermektedir. Kalman Filtresi tabanlı algoritmalarda mevcut olan problemler, özellikle filtrenin başlayabilmesi problemi evrimsel hesaplama yaklaşımında mevcut değildir. Bunun yanında evrimsel hesaplama yönetiminde fonksiyon lineerleştirme ihtiyacı olmamakta, bu sebeple Kalman Filtresi tabanlı yaklaşımlarda karşılaşılan lineerleştirme sonucu ortaya çıkan ilave hatalar görülmemektedir.

Hedef Hareket Analizi yapan TMAGA yazılımımız, HHA problemini, üretmiş olduğu taklit sinyallerle, ölçmüş olduğu gerçek kerterizler arasındaki farkı minimize etmeye çalışarak çözer. Taklit sinyaller TMAGA yazılımının çekirdeği olan Genetik Algoritma Motoru vasıtasıyla işlem boyunca evrime uğratılırlar ve sonuçta gerçek ölçülen sinyallere yaklaştırılırlar.

Algoritmamızın ve TMAGA yazılımının performansını incelemek amacıyla geniş çapta Monte Carlo denemeleri icra edilmiştir. Denemeler beklenenin üzerinde başarılı geçmiş, değişik derecelerde Gauss dağılımlı hata miktarlarıyla kirletilmiş temas bilgilerinden, düşük gözlemlenebilirlik koşullarında dahi hedef hareket modeli parametreleri tahmin edilebilmiştir.

Ayrıca, beklenenin aksine, genetik algoritma ile gerçekleştirilen HHA çözümü için gerekli zaman çok yüksek çıkmamış, normal gözlemlenebilirlik şartlarında 1-3 dakika arası bir sürede TMAGA'nın problemi çözdüğü müşahade edilmiştir.

Bundan sonra bu çalışmanın akabinde yapılabilecek çalışmalar ise şöyle özetlenebilir; Manevra yapan bir hedefa ait HHA problemi çözülebilir. Koromozomlarda kodlanan hedef hareket modeli parametrelerine standart (X_0, Y_0, C, S) parametreleri yanında (X_t, Y_t) dönüş noktası ve (C_n) yeni rota parametreleri de eklenebilir. Aynı zamanda ölçülen sinyal frekansları da çözüme destek sağlamak amacıyla kullanılabilir.

HHA çözümlerinde, birden fazla farklı coğrafi mevkiye konuşlandırılmış ayrıık pasif ve ve/veya aktif sensörlerden alınan ölçümlerin TMAGA'da kullanılmasıyla Çoklu Sensör Veri Füzyonu (Multi Sensör Data Fusion – MSDF) gerçekleştirilebilir. Bu yaklaşımda TMAGA'nın ürettiği taklit sinyallerin değerlendirilmesinde bu farklı sensörlere ait ölçümler kullanılabilir. Yani bu durumda minimize edilecek değer, taklit sinyali ile birden fazla sensöre ait ölçümlerin arasındaki topyekün fark olacaktır.

Ters probemler bir çok mühendislik alanında kullanılmasına rağmen İşletme Biliminde kendisine henüz fazla bir yer bulabilmiş değildir. Ancak ters problemlerin, aslında birer tahmin problemi olduğunu düşünürsek, şüphesiz İşletme Biliminde de bir çok yeni uygulama alanı bulunacaktır. Özellikle rakabet analizi alanında, işletmelerin, rakiplerinin gelecekteki davranışlarını yada mevcut faaliyetlerini şu andaki davranışlarını analiz ederek tahmin edebilmeleri problemi bir ters problem olarak algılanabilir.

Umarız bu çalışma, İşletme biliminde ters problemler konusunda yapılabilecek yeni çalışmalara rehber olacaktır.

KAYNAKLAR

Aidala V.J., Hammel S.E., 1983, Utilization of Modified Polar Coordinates for Bearing Only Tracking, IEEE Trans. 1983, AC-28

Aidala V.J., Kalman Filter Behaviour in Bearings-Only Tracking Applications, IEEE Trans. AES-15

Ancker C.J., 1958, Airbone Direction Finding – The theory of Navigation Errors, 1958.

Balckman S.S., Popoli R., 1999, Design and Analysis of Modern Tracking Systems, Artech House Inc.

Berman Z., 1997, A Reliable Maximum Likelihood Algorithm for Bearing-Only Target Motion Analysis, Proceedings of the 36th Conference on Decision & Control, San Diego, California USA

Cedeno W., Vemuri V.R., Database Design with Genetic Algorithms

Cunningham A., Thomas B., 2005, Target Motion Analysis Visualisation.

Dasgupta D., Optimal Scheduling of Thermal Power Generation Using Evolutionary Algorithms

Eiben A.E., Smith J.E., 2003, Introduction to Evolutionary Computing (Natural Computing Series)

Fagerlund S., 1980, Target Tracking Based on Bearing Only Measurement., LIDS-R-1003, MIT, MA

Fialkowki L, et al., 2001, Matched Field Source Tracking by Ambiguity Surface Averaging, J.Acoust.Soc.Am.110

Fujimoto O., Okita Y., Ozaki S., 1997, Nonlinearity-Compensation Extended Kalman Filter and Its Application to Target Motion Analysis, OKI Tech Rev. No. 159 Vol.63

Gong K.F., Lindgren A.G., 1981, Passive Localization and Motion Analysis with a State Parameter Constraint, Proc. Of 15th Asilomar conf. on Circuits, System and Computers

Gong K.F., Speyer J.L., 1985, A Stochastic Analysis of a Modified Gain Extended Kalman Filter with Applications to Estimation with Bearings Only Measurements, IEEE Trans. AC-30

Grossman W., 1991, Bearings-Only Tracking: A Hybrid Coordinate System Approach, Proceedings of the 30th Conference on Decision and Control, Brighton, England

Jazwinski A.H., 1973, Stochastic Processes and Filtering Theory, Academic Press.

Johnson G.W. Cohen A.O., 1983, Modified Polar Coordinates for Ranging from Doppler and Bearing Measurements, IEEE Proc. 907-910

King J.P, Fahmy H.S., Wentzell M.W., A Genetic Algorithm Approach for River Management

Kronhamn T.R., 1998, Bearings-Only Target Motion Analysis Based on a Multihypothesis Kalman Filter and Adaptive Ownship Motion Control, IEE Proc. – Radar, Sonar Navig., Vol 145, No. 4

Mitchell M., 1998, An Introduction to Genetic Algorithms (Complex Adaptive Systems)

Moon J.R., Stevens C.F., 1996, An Approximate Linearization Approach to Bearings-Only Tracking, The Institution of Electrical Engineers.

Nardone S.C., Lindgren A.G., 1984, Fundamental Properties and Performance of Conventional Bearings Only Target Motion Analysis, IEEE Trans. AC-29

Ozard J. et al., 1993, Speed-up and Detection Performance of a Nearest-Neighbors Algorithm for Matched Field Processing with a Vertical Line Array, IEEE Oceans 86-90

Peach N., 1995, Bearings-Only Tracking Using a Set of Range-Parameterized Extended Kalman Filters, IEE Proc. – Control Theory, Vol 142, No. 1

Peters D.J., 2001, A Practical Guide to Level Once Data Fusion Algorithms, DREA TM 2001-201

Porat P., Friedlander B., 1970, Analysis of Asymptotic Relative Efficiency of MUSIC Algorithm

Searle S.J., 2005, Efficient Matched Processing for Localization of a Moving Acoustic Source, *Signal Processing* 85, 1787-1804

Springarn K., 1987, Passive Position Localization Estimation Using the Extended Kalman Filter System, *IEEE Trans.*, AES-23

Streit, R. L., Walsh M.J., 2002, Bearings-Only Target Motion Analysis With Acoustic Propagation Models of Uncertain Fidelity. *IEEE Trans. on Aerospace and Electronic Systems* 1222-37

Tomasz, Gwiazda D., 2006, Genetic Algorithms Reference

Weiss H., Moore J.B., 1980, Improved Extended Kalman Filter Design for Passive Tracking, *IEEE Trans. AC-25*

Wilhoit G.Z., 1983, Stochastic Control of a Passive Search for Ocean Vehicles, *Ocean Engineering Thesis, MIT, MA*

ÖZGEÇMİŞ

Doğum Yeri ve Tarihi: Nevşehir-1972

Eğitim Bilgileri:

İlkokul: Polatlı, Atatürk İlkokulu 1978-1981

Ankara, Kemal Yurtbilir İlkokulu 1982-1983

Orta Okul : Ankara, Atıfbey Ortaokulu 1984-1986

Lise: İstanbul, Deniz Lisesi 1986- 1990

Lisans: İstanbul, Deniz Harp Okulu 1990-1994

Yüksek Lisans : Naval Postgraduate School Monterey California USA 1998-2000 (Bilgisayar Mühendisliği)

Araştırma Görevi : Halifax, N.S. – Kanada, Defence Research & Development Canada NATO Fellowship Visiting Scientist (2005-2005)

Halen Deniz Kuvvetleri Komutanlığı Araştırma Merkezi Komutanlığında yazılım mühendisi olarak çalışmaktadır.