İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞI DESTEKLİ KALMAN FİLTRESİ YARDIMIYLA HEDEF İZLEME

Y. LİSANS TEZİ

Tarkan SANCAKDAR

Enstitü No: 511991081

Anabilim Dalı: Uçak Mühendisliği

Programı: Uçak Mühendisliği

Tez Danışmanı: Doç.Dr. Ramazan TAŞALTIN

MAYIS 2002

İÇİNDEKİLER

KISALTMALAR iii
TABLO LİSTESİ iv
ŞEKİL LİSTESİ v
SEMBOL LİSTESİ vi
ÖZET vii
SUMMARY ix

1. GİRİŞ 1

2. YAPAY SİNİR AĞLARI 4

2.1.Giriş 5
2.2. Biyolojik Sinir Ağları 6
2.3. Yapay Sinir Ağının Özellikleri 8
2.4. Sinir Hücresi Yapısı 9
2.5. Aktivasyon Fonksiyonları 12
2.5.1. Threshold Fonksiyonu 12
2.5.2. Parçalı-Lineer Fonksiyon 12
2.5.3. Sigmoid Fonksiyonu 12
2.6. YSA Modelleri 14
2.6.1. İleri Beslemeli Ağlar 15
2.6.1.1. Tek Katlı İleri Beslemeli Ağlar 15
2.6.1.2. Çok Katlı İleri Beslemeli Ağlar 15
2.6.2. Geri Beslemeli Ağlar 16
2.7. Yapay Sinir Ağlarında Öğrenme 17
2.7.1. Hebb Öğrenme Kuralı 19
2.7.2. Perceptron Öğrenme Kuralı 20
2.7.3. Widrow-Hoff Öğrenme Kuralı 21
2.7.4. Delta Öğrenme Kuralı 21
2.8. Geriye Yayılım Algoritması 22

3. KALMAN FİLTRESİ 29

3.1.Giriş 29
3.2. Bir Dinamik Modelin Kestiriminde Optimallik 31
3.3. Kalman Filtresine Giriş 32
3.4. Kalman Filtresi Matematik Modeli 36

4. UYDU NAVİGASYON SİSTEMLERİ 42

4.1.Giriş 42
4.2. Uydu Navigasyon Sistemleri Yapısı 43
4.2.1. Uzay Bölümü 43

4.2.2. Kullanıcı Bölümü 44
4.2.3. Kontrol Bölümü 45
4.3. Bir Cismin Konumunun Tespiti 47
4.4. GPS Ölçümlerini Etkileyen Hatalar 49
5. KALMAN FİLTRESİ İLE GPS'NİN BİRLEŞTİRİLMESİ 51
5.1. Uyduların Hareket Simülasyonu 52
5.2. Uçağın Uçuş Simülasyonu 53
5.3. GPS Ölçümlerinin Kalman Filtresi ile Düzeltilmesi 55
5.4. Simülasyon Sonuçları 58
6. KALMAN FİLTRESİNİN YSA İLE BİRLEŞTİRİLMESİ 60
6.1. YSA Destekli Kalman Filtresi Yapısının Kurulması 60
6.2. Simülasyon Sonuçları 64
7. SONUÇ 68
KAYNAKLAR 70
EKLER 72
ÖZGEÇMİŞ 78

KISALTMALAR

YSA	: Yapay Sinir Ağları
ÇKP	: Çok Katmanlı Perceptron
KF	: Kalman Filtresi
INS	**:** Ataletse Navigasyon Sistemi
GPS	: Global Positioning System
NAVSTAR	: Navigation System with Time and Ranging

TABLO LİSTESİ

		Sayfa No
Tablo 4.1.	GPS Standart hata modeli	50
Tablo 5.1.	Uydunun simülasyonunda kullanılan başlangıç değerleri	53
Tablo 6.1.	YSA'nın Kalman Filtresini düzeltme oranı	66

ŞEKİL LİSTESİ

		Sayfa No
Şekil 2.1	Yapay sinir ağı modeli	11
Şekil 2.2	Treshold fonksiyonu	12
Şekil 2.3	Parçalı-Lineer Fonksiyon	13
Şekil 2.4	Sigmoid fonksiyonu	13
Şekil 2.5	Tek katmanlı ileri beslemeli ağ	15
Şekil 2.6	Çok katmanlı ileri beslemeli ağ	16
Şekil 2.7	Tek katmanlı geri beslemeli yapay sinir ağı	17
Şekil 2.8	Denetimli öğrenme blok diyagram	18
Şekil 2.9	Denetimsiz öğrenme blok diyagramı	18
Şekil 2.10	Hebb öğrenme kuralının şematik hali	20
Şekil 2.11	Perceptron öğrenme kuralının şematik hali	20
Şekil 2.12	Delta öğrenme kuralı şematik gösterimi	22
Şekil 2.13	Çıkış katmanındaki j. nöronu detaylı olarak gösteren sinyal akış grafiği	23
Şekil 2.14	Çıkış nöronuna gizli nöron eklenmiş halde sinyal akış diyagramı	25
Şekil 2.15	Hata sinyalinin geriye yayılımını gösteren sinyal akış diyagramı	28
Şekil 3.1	Kalman Filtresinin Matematiksel Temeli	30
Şekil 3.2	m_1 ölçme şartına bağlı şartlı yoğunluk fonksiyonu	33
Şekil 3.3	m_2 ölçme şartına bağlı şartlı yoğunluk fonksiyonu	34
Şekil 3.4	$m_1$ve m_2 ölçme şartına bağlı şartlı yoğunluk fonksiyonu	35
Şekil 3.5	Temel Kalman Filtresi	36
Şekil 3.6	Ayrık Kalman Filtre çevrimi	39
Şekil 3.7	Kalman Filtresinin çalışmasını gösteren diyagram	40
Şekil 3.8	Kalman filtresinin yapı şeması	40
Şekil 4.1	Uydu Navigasyon Sistemi	43
Şekil 4.2	GPS alıcı bölümü elemanları	45
Şekil 4.3	Kontrol bölümü ve işlemleri	46
Şekil 4.4	Navigasyon uyduları ile cismin konumunun bulunması	48
Şekil 5.1	Uçağın x ekseni yönündeki hareketi	58
Şekil 5.2	Uçağın x yönündeki gerçek, GPS, Kalman Fitresi sonuçları	58
Şekil 5.3	Uçağın sistem bozuntusuna bağlı olarak KF çıkışları	59
Şekil 6.1	Kalman Filtresinin YSA ile birlikte çalışması	61
Şekil 6.2	Kullanılan YSA yapıları.	62
Şekil 6.3	KF ve YSA destekli KF çıkışı	63
Şekil 6.4	Tek gizli kat. YSA'nın Kalman Filtresini düzeltme sonucu	65
Şekil 6.5	İki gizli katmana sahip YSA'nın karesel hata grafiği	67
Şekil 6.6	Tekgizli katmana sahip YSA'nın karesel hata grafiği	67

SEMBOL LİSTESİ

A, B, C, D : Sistem Matrisleri

X_t : t anında sistem durum vektörü

Φ : Kalman Filtresi durum geçiş matrisi

Δt : Örnekleme zamanı

Q : Sistem bozuntusunun korelasyon matrisi

R : Ölçüm bozuntusunun korelasyon matrisi

w_t : Sistem bozuntusu

v_t : Ölçüm bozuntusu

W : Ağırlık matrisi

η : Öğrenme katsayısı

α : Momentum katsayısı

δ_k : Lokal grandyant

Y_D : Giriş uzayı

U_D : Çıkış uzayı

Üniversitesi : **İstanbul Teknik Üniversitesi**

Enstitüsü : **Fen Bilimleri**

Anabilim Dalı : **Uçak Mühendisliği**

Programı : **Uçak Mühendisliği**

Tez Danışmanı : **Doç. Dr. Ramazan TAŞALTIN**

Tez Türü ve Tarihi : **y. Lisans – Mayıs 2002**

ÖZET

YSA DESTEKLİ KALMAN FİLTRESİ İLE HEDEF İZLEME

Tarkan SANCAKDAR

Kalman filtresi, öz yinelemeli ve ardışık veri işleme yeteneği ile, bir hedefin pozisyon, hız ve ivmesini tahmin etmek için yaygın şekilde kullanılır. Ancak Kalman filtresinin başarılı çalışabilmesi için şart olan gauss dağılımlı sistem gürültüsü pratik uygulamalarda her zaman olmayabilir. Bu durum Kalman filtresinin tahmin hatasını arttırır. Böyle bir durumda Kalman filtresinin tahmin hatasını azaltmak için çeşitli yaklaşımlar vardır. Bu yaklaşımlardan birisi birçok Kalman filtresinin paralel olarak işletilmesidir. Bu yaklaşım doğru bir hedef izleme sağlasa da hesaplama süresinin artması ve hesaplamanın daha kompleks hale dönüşmesinden dolayı tercih edilen bir yaklaşım değildir.

Yukarıda anlatılan yaklaşımın getirdiği zorluğu çözmek için yapay sinir ağları ile farklı bir yaklaşım getirilmiştir. Bu yaklaşım, Kalman filtresinin tahmin yapabilme kapasitesini yapay sinir ağlarının öğrenme yeteneğini birleştirmektedir. Böylece hedef izleme doğruluğu ve hızı arttırılmaktadır.

Bu çalışmada, yukarıda anlatılan yapay sinir ağları ile desteklenmiş Kalman filtresinin hedef izleme performansını incelenmiştir ve yapay sinir ağı kullanılmayan Kalman filtresi ile karşılaştırmasını yapılmıştır. Ayrıca farklı YSA yapılarının hata düzeltmedeki performansını inceleyeceğim. Bu tezde üç temel aşama olacaktır. Birinci aşamada ayrık Kalman filtresi ile GPS'ten alınacak bilgilere göre uçağın durum bilgileri tahmin edilecektir. İkinci aşamada Kalman filtresinin hatası hesaplanıp bu hatalarla yapay sinir ağı eğitilecektir. Kalman filtresinin yapacağı hataları önceden öğrenmiş olan yapay sinir ağı Kalman Filtresine eklendiğinde sistem çıkışında Kalman Filtresinin hatalarından arınmış sonuçlar elde edilecektir. Yapay sinir ağının eğitiminde, popüler bir yöntem olan geriye yayılım algoritması kullanılacaktırYapay sinir ağı ile Kalman filtresinin birlikte çalışması şöyle olacaktır. Radardan alınan bilgilere göre Kalman filtresi uçağın durum bilgilerini tahmin edecektir. Kalman filtresinin yapacağı hataları öğrenen yapay sinir ağı bu hataları giderir ve daha doğru bir sonuç elde edilir. Son olarak da bu iki yaklaşımın hedef izlemede performansları karşılaştırılacaktır.

Anahtar Kelimeler: Yapay Sinir Ağları, Kalman Filtresi, GPS

SUMMARY

Kalman filters has wide range of use in target tracking to estimate the velocity and acceleration of a flying object object due to its ability to process the squential data. However Kalman filter asumes that the system noise has Gaussian distribution characteristic, and this assumption is violated in most practical stuations, resulting the increase of Kalman filter estimation error. There are different techniques to reduce this estimation error. One method to reduce the estimation error is to use parallel Kalman filters. This approach however increases the calculation burden ant it is not preferred in most practical stuations.

One approach to solve the above problem is to use neural networks to reduce the Kalman filter estimation error. This method combines the advantages of Kalman filter estimation techniques with those of NN learning capacity. Thus the accuracy and speed of target tracking algorithm is increased.

In this work, Kalman filter based target track tracking algorithm is improved by using neural networks. The performance of Kalman filter algrithm is compared with that of neural network aided Kalman filter algorithm. Different NN structures is examined to obtain their success in suppression of estimation error. This subject is investigated in three parts. In the first part GPS data is processed by Kalman filter and aircraft position is solely estimated by it. In the second part Kalman filter error is used to train NN and aircraft. Once the NN is trained and after that the trained NN is used to get more correct results.

In the last section performance of these two approaches are compared in target tracking applications.

1. GİRİŞ

Çağımızda bilgisayar teknolojisinin ve ölçme aletlerinin hızlı bir şekilde gelişmesi, hemen hemen tüm meslek ve bilim dallarında anında sonuçlar üretebilen sistemlerin kurulmasına olanak sağlamıştır. Otomasyonu sağlayacak ve kullanıcılara direkt sonuç ürünlerini sunabilecek bu tür sistemlerin oluşturulması, günümüzde bilim adamlarının öncelikli konusunu oluşturmaktadır. GPS (Global Positioning System) tekniği ile konumların belirlenmesi, uçak kontrolünde önemli bir gelişme olmuştur. Her türlü hava şartında ve gece-gündüz sürekli kullanılabilmesi diğer navigasyon sistemlerine karşı bir üstünlük oluşturmaktadır. Bu yeteneklerinden dolayı hava, kara ve deniz taşıtlarında konum bulma problemi GPS sistemi yardımı ile çözülmektedir. Bununla beraber optimal kestirim ilkesini koruyan ve bilgisayarlarda daha hızlı çalışarak anında sonuçlar üretebilen algoritmaların denenmesi bilimsel çalışmaların başında gelmektedir. Bu süreç yavaş yavaş kendini tüm dünyada göstermektedir. Bu algoritmaların başında da Kalman Filtresi gelmektedir.

20. yy teknolojisindeki başdöndürücü gelişmeler, son yıllarda bilim adamlarının insan ve insan davranışlarına olan ilgisini körüklemekte, aynı zamanda da düşüncelerini uygulamaya dökme imkanı tanımaktadır. Sinir ağları, insan ve yüksek sınıftan diğer canlılarda bulunan yaşamsal mekanizmanın temelidir. Düşünme gibi sadece insan türünde bulunan yeteneklerin dışında sinir ağları canlının fiziksel özelliklerini, kimyasal dengesini, savunma mekanizmalarını oluşturur. Bilim adamları son 10 yılda basit ve nispeten yavaş hücrelerin bir araya geldiklerinde nasıl oluyor da dünyanın en hızlı bilgisayarından daha hızlı işlem yapabildiklerini araştırmaktadır. Araştırmalar, insanları nöronların basit modellerini yapıp, bu

modeller ile oluşturdukları devrelere insansı bazı işlemleri yaptırmaya götürmüştür. Nöronun modellenmesi, ele alınan örneğe bağlı olarak çok farklı şekillerde olabilir. Öğrenme yeteneği olan bir nöron yanında, öğrenme yeteneği olmayan fakat çok hızlı cevap verebilen bir nöron modeli yapmak da mümkündür.

Bu çalışmada YSA (yapay sinir ağları), GPS ve Kalman Filtresi hakkında bilgi verildikten sonra bu üçünün üçünün birlikte kullanıldığı, bir cismin x,y,z ekseninde pozisyonlarını hesaplayan bir navigasyon problemi örnek olarak verilmiştir.

2. bölümde ilk önce genel olarak sinir ağlarının tarihsel gelişiminden bahsedilmiştir. Biyolojik sinir ağları ile yapay sinir ağları tanıtılıktan sonra basit bir YSA hücre yapısı anlatılmıştır. Daha sonra bu hücre yapısında kullanılan aktivasyon fonksiyonları ve YSA modelleri açıklanmıştır. YSA'da kullanılan öğrenme kuralları anlatıldıktan sonra, son olarak geriye yayılım algoritmasının teorisi ele alınmıştır.

3. bölümde kestirim probleminin Kalman Filtresine gelinceye kadar kısa bir tarihi ele alınmış ve dinamik bir sistem kestiriminde optimallik nasıl olmalı konusu açıklanmıştır. Kalman Fitresinin daha iyi anlaşılması için ilk önce örnek üzerinden bir kestirimin nasıl olduğu anlatılmış ve sonra Kalman Filtresinin matematik modeli verilmiştir.

4. bölümde navigasyon uydu sistemlerinin yapısı açıklanmıştır. Navigasyon uyduları ile bir cismin konumunun nasıl bulunduğu anlatıldıktan sonra GPS ölçümlerini etkileyen faktörler ortaya konulmuştur.

5. bölümde Kalman Filtresi ile GPS ölçümlerinin birleştirilmesi anlatılmıştır. İlk önce 4 GPS uydusunun hareketinin, daha sonra bir uçağın x,y,z yönündeki doğrusal hareketinin simülasyonu yapılıp bu simülasyonlar üzerinden GPS ölçümleri elde edilmiştir. Daha sonra bozuntulu olan bu ölçümler Kalman

Filtresi yardımı ile bozuntulardan arındırılmaya çalışılmıştır. Kalman Filtresi uygulanırken çeşitli sistem bozuntularında performansı izlenmiştir.

6. bölümde Kalman Filtresi sonuçlarının YSA ile nasıl düzeltileceği konusu ele alınmıştır. Farklı YSA yapılarının sonucu nasıl etkilediği gözlenmiştir.

2. YAPAY SİNİR AĞLARI

2.1. GİRİŞ

Bu bölümde yapay sinir ağları ele alınacaktır. İlk önce biyolojik sinir ağları tanıtılacak ardından yapay nöron modeli incelenecektir. Yapay sinir ağları mimarisi incelendikten sonra başlıca öğrenme kuralları, avantajları ve dezavantajları anlatılacaktır.[1] Bu konulara başlamadan önce asgari bir alt yapıyı oluşturmak için yapay sinir ağlarının basit bir tarifi ve tarihçesi verilecektir.

Sayısal bilgisayarlar tamamen kendileri için tanımlanmış komut dizilerini yürütmekte oldukça başarılıdırlar. Öte yandan insan beyni, sayısal bilgisayarlarla yapılması zor olan gürültülü bir ortamda ve kısmen bozulmuş bilgi ile görme, konuşma, bilgi düzeltmesi ve şekil tanıma gibi işlerde başarılı bir davranış gösterir. Beynin işleme elemanları (nöronlar), çağdaş bilgisayarların işleme elemanlarından daha yavaş olmalarına rağmen, beyin bu işi nasıl başarmaktadır? Elektro kimyasal elemanlar olan nöronlar, milisaniye mertebelerinde cevap verebilirken, şu andaki elektronik teknoloji ürünleri nonosaniye mertebesinde durum değiştirebilirler. Teknolojik gelişmeye rağmen söz konusu yeteneklere sayısal bilgisayarlar üzerinde alışılagelmiş yapay zeka yöntemlerini kullanarak doyurucu bir şekilde erişilememesi, birçok bilim adamını böyle yeteneklere sahip makinelerin tasarlanması konusunda yeni yaklaşımlar için insan beynini incelemeye itmiştir.

Yapay sinir ağ modelleri üzerinde, yapay zekanın çeşitli alanlarında yıllarca çalışılmıştır. Bir yapay sinir ağı, insan sinir sisteminin bazı fonksiyonlarını modelleyen ve onun bazı yeteneklerini yakalamak isteyen basit hesaplama birimlerinin (nöronların) yoğun bir paralel dizisidir. Nöral ağların başka bir tanımı şu şekilde verilebilir. Yapay sinir ağları, basit elemanların yoğun bir şekilde paralel olarak bağlanmasıyla oluşan ağlardır ve gerçek dünyadaki cisimlerin, aynen biyolojik sinir sistemlerinin yaptığı gibi ilişkide bulunabilmeleri için, belli bir düzende birbirlerine bağlanmış halidir.[2]

Yapay sinir ağları konusunda literatüre geçmiş ilk büyük çalışma 1943 yılında W. S. McCulloch ve W. A. Pitts tarafından yapılmıştır.[3] Bu tarihte McCulloch ve Pitts ilk hücre modelini geliştirdiler ve birkaç hücrenin arabağlasımını incelediler. 1949 yılında ise bir nöropsikolojist olan Hebb, sinir ağının davranışını ortaya çıkarmış ve hücre bağlarının değerlerini ayarlamak için bir öğrenme kuralı geliştirmiştir. [4]

1958 yılında Rosenblatt, algılayıcı (perceptron) modelini ve öğrenme kuralını geliştirdi. Bugün kullanılan kuralların temellerini koydu.[5] Perceptron, karmaşık adaptif davranışlarda bulunma yeteneği olan bir öğrenme makinasıydı. Perceptronun en büyük dezavantajı lineer ayrılabilen fonksiyonları gerçekleyebilmesiydi. Devrenin öğrenme kapasitesi, giriş bilgileri arasındaki bu sınıflandırmaya sıkı sıkıya bağlıdır. Makalenin en önemli noktası, o zamana kadar yapılan çalışmaların aksine, Rosenblatt'ın, sayısal lojik fonksiyonların beynin modellenmesine uygun olmadığını vurgulamasıdır.

Minsky, 1960'larda yürüttüğü araştırmasının bir sonucu olarak, basit tek katmanlı ağların bazı çok kolay problemleri çözemeyeceklerini keşfetti. O zamanlarda çok katmanlı ağlar çok az anlaşılmıştı ve çok katmanlı eğitmeyi gösteren hiçbir teori yoktu. 1969 yılında Minsky ve Papert, beraber yazdıkları "Perceptron" adlı kitapta, tek katmanlı ağların XOR işlemini gerçekleştirmek gibi basit işlemleri yapamayacağını ispatlamışlardır.[6] Minsky ve Papert şunları yazmışlardır.

" Perceptron birçok kısıtlarına rağmen (ve hatta bunlardan dolayı) incelemeye değer olduğunu göstermiştir. Dikkati çeken bir çok özelliği vardır:

lineerliği, ilgi çeken öğrenme teoremi, -bir çeşit paralel hesaplama olan- kullandığı yöntemin basitliği. Bütün bu özelliklerin çok katmanlı ağlarda da olması için hiçbir neden yoktur. Yine de, katmanları arttırmanın verimsiz olacağı yolundaki içten gelen yargımızı açıklamak (veya reddetmek) için, çok katmanlı versiyonu önemli bir araştırma problemi olarak nitelendiriyoruz. Belki de çok katmanlı makine için bir öğrenme teoreminin bulunamamasının nedeni güçlü bir yakınsaklık teoreminin bulunamayışıdır."

Perceptronun bu başarısızlığından sonra yapay sinir ağları üzerinde yapılan çalışmalar hemen hemen durma noktasına gelmiştir. Çalışmalar sadece birkaç bilim adamının kişisel gayreti ile son derece dağınık ve yavaş olarak 1980'li yıllara kadar ilerledi. 1980'li yıların başında Hopfield adlı bir nörobiyoloğun yayınladığı üç makale [7-9] ile yapay sinir ağları üzerindeki çalışmaları tekrar canlandırdı.

1986'da, Rumelhart, esas olarak 1974 yılında Werbos tarafından bulunan çok katmanlı algılayıcı (perceptron) tipi ağlar için geriye yayılma algoritması denen bir eğitim algoritması geliştirdi.[10] Bu çalışmayla yapay sinir ağları alanında bir çığır açtılar. Bugün en çok kullanılan algoritma da bu algoritmadır.

Geriye yayılım yönteminde öğrenme ve öğrenmenin hızlanabilmesi, eğitim kümesinin ağa sunulması gibi konular değişik çalışmalarda ele alınmıştır.[18-21] M. Riedmiller, çeşitli öğrenme yöntemlerinin karşılaştırılması üzerine bir çalışma yapmıştır.[11]

2.2. Biyolojik Sinir Ağları

Yüzyıllar boyunca insanlığın en büyük amacı işlerini kolaylaştıracak makineler yapmak olmuştur. Bu büyük tutkunun ardındaki yegane sebep insanın hoşlanmadığı işleri başkalarına yaptırıp kendine ve sevdiği işlere daha fazla vakit ayırması olarak ifade edilebilir. İnsanlığın emekleme dönemlerinde kaldıracın, tekerleğin bulunmasından beri bu böyle süre gelmiş ve bilim adamları bugün akıllı

makineler geliştirmekle uğraşmaktadırlar. Yapay sinir sistemleri akıllı makinelere bir örnek teşkil etmekte olup daha kaliteli bir yaşam için büyük vaadiler sunmaktadır.[1]

Bilindiği gibi insanlar ve hayvanlar örneğin bir görüntünün tanımlanmasında en gelişmiş bilgisayarlardan bile daha iyidirler. Bunun başlıca sebeplerinden bir tanesi beyin sahibi olan varlıklarda beynin geleneksel sayısal bilgisayarlardan çok farklı bir şekilde çalışması olarak ifade edilebilir. Bilgisayarlar ardarda bir çok işlemi hatasız ve çok hızlı bir şekilde gerçekleştirebilirler fakat ses tanıma veya görüntü tanıma gibi bazı özelliklerini maddeler halinde yazarsak kıyaslama yapmamız daha kolay olur.

Sayısal bilgisayarlar:

1. Seri çalışır.
2. Hızlı birimlerden oluşur(İşlem hızı saniyede milyon işlem mertebesinde)
3. Tam tanımlı komut dizini (programlı hesaplama) gerektirir.
4. Belli bir adımdaki yanlış hesaplama, sonucu çok değiştirir.
5. Giriş verilerindeki bir hata zamanla yayılır.
6. Alt birimlerdeki bozukluk tün sistemi devre dışı bırakabilir.

Beyin:

1. Paralel çalışır.
2. Yavaş birimlerden oluşur.(işlem hızı saniyede 10 işlem mertebesinde).
3. Bilgi çevresine, adaptif olarak işlemsel yeteneklerini ayarlayarak cevap verir.
4. Alt birimlerin, yapılan ana işe katkısı azdır. Bu nedenle bir adımdaki yanlış hesaplama sonuca fazla etki etmez.
5. Giriş verilerindeki gürültüye duyarsızdır.
6. Sistemin bir kısmının ölmesi performansı düşürür fakat tüm sistemi devre dışı bırakmaz.

Yukarıdaki maddelerden de görüldüğü gibi insan beyni yavaş birimlerden oluşmasına rağmen paralel çalışma ve bilgi çevresine adaptif bir şekilde cevap vermesi ile muazzam bir sistem oluşturmaktadır. Ayrıca insan beyni üzerinde yapılan araştırmalara göre beyinde ortalama yüz milyar hücre bulunduğu ve her hücrenin

ortalama olarak birbirleri ile on bin bağlantı yaptığı düşünülürse sistemin kompleksliği daha da iyi anlaşılır.

2.3. Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağlarının alışılagelmiş bilgi işleme yöntemlerine göre üstünlükleri şu şekilde sıralanabilir.[33,34]

1. **Paralellik:** Alışılagelmiş olduğu gibi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da özellikle hız problemlerine neden olur. Mesela, bilgisayarlar beyne göre çok hızlı çalışmasına rağmen beynin toplam hızı bilgisayarlara göre kıyaslanamayacak kadar yüksektir. Doğaları gereği sinir ağları kendilerini düzenlerler başka bir deyişle her düğüm sistem içinde diğer tüm işlemcilerden bağımsız olarak kendi girişleri üzerinde çalışan bir işlemci olarak görülebilir. Yani, ağ bir çözüme yakınsarken, tüm işlemciler meşguldür.
2. **Gerçekleme Kolaylığı:** Yapay sinir ağları, karışık fonksiyonlar yerine basit işlemleri içerdiği için gerçeklemek kolaydır.
3. **Yerel Bilgi İşleme: Y**apay sinir ağlarında her bir işlem birimi, çözülecek problemin tümü ile ilgilenmek yerine, sadece problemin bir parçası ile ilgilenir. Hücrelerin çok basit işlemler yapmalarına rağmen, sağlanan görev paylaşımı sayesinde çok karmaşık ve zor problemler çözülebilmektedir.
4. **Öğrenebilirlik:** Alışılagelmiş veri işleme yöntemlerinin çoğu programlama ile hesaplamaya dayanır. Bu yöntemler ile tam tanımlı olmayan bir problem çözülemez. Ayrıca, herhangi bir problemin çözümü için probleme yönelik algoritma geliştirmek gerekir. Yapay sinir ağları ise problemleri, verilen örnekler ile çözer. Çözülecek problemler için yapı aynıdır.
5. **Genelleme:** Geleneksel bilgisayarların bir karakteristiği, istenilen çıkışı üretmek için, tam doğru girişlere ihtiyaç göstermeleridir. Öte yandan sinir ağları girişlerine değişimler olsa bile çıkışı üretebilir. Örneğin, basılı harfleri tanımak için eğitilmiş bir sistem, giriş karakterleri %40 oranında bozulsa bile doğru olarak çalışabilir. Yani, sistem daha önceden tam o tipten hiçbir

şey görmemiş olmasına rağmen aynen insanların tamam olmayan veya kısmen hatalı girişlerle yaptığı gibi tanıyabilmektedir. Çalışmalar, insanların çoğunun bir metindeki harflerin yarısından fazlası silinmiş olsa bile o metini okuyabildiklerini göstermiştir. Bilgisayarlar gürültülü girişlere katlanacak şekilde programlanabilirler ancak getirilen ek hesapsal yük çoğu kez bu algoritmaların pratik uygulamalarda kullanılmasını engellemektedir. Sinir ağları bu genellemeyi ayrıntılı programlamadan ziyade, yapılarından dolayı başarırlar.

6. **Çıkarım yapma: Y**apay sinir ağları tam olmayan bir eğitim kümesinden tam doğruyu çıkarabilir. Örneğin; alfabeyi tanımak için eğitilmiş bir yapay sinir ağına farklı biçimde yazılmış harfler gösterildiğinde harfleri tanıyabilmektedir. Yani sistem kendine hiç öğretilmemiş bir şeyleri hatırlamıştır. Diğer bir deyişle sistem eğitim kümesinin özünü çıkarmış ve saklamıştır. Böylece eksik girişlere uygun cevap verebilmektedir.
7. **Hız: Y**apay sinir ağları giriş bilgileri ile istenilen çıkış bilgilerini ilişkilendiren eşlendiren(asosyatif) bellekler olarak görülebilirler. Girişe yeni bilgiler sunulduğunda, ilgili bilgiler çıkışta üretilir. Sinir ağlarında çıkışları üretmek için gerekli zaman, saklanan ilişkilerin sayısından bağımsızdır. Yani, "arama"ya karşı hiçbir şey düşmez. Gerekli zaman ağ kararlılığı ile ilgili olan bir süredir ki bu da birçok mimaride sabittir. On milyon ilişkiyi saklayan bir ağ, on bin ilişkiyi saklayan bir ağ kadar hızlıdır.

2.4. Sinir Hücresi Yapısı

Bugüne kadar verilmiş tüm yapay sinir ağları modellerinin ortak özelliklerini içeren bir tanım aşağıda verilmiştir.[13]:

I. Ağ, II-VIII de verilen hücre denilen yapı taşlarının birbirlerine IX-XII de belirtilen bir geometri ile bağlanmasından oluşan bir sistemdir.

II. Hücreler, genelde çok girişli ve tek çıkışlı, yüksek dereceden lineer olmayan dinamik alt devrelerdir.

III. Hücrelerin yapısı şekil 2.1'de gösterildiği biçimdedir.

IV. Tek olan çıkış işareti, dışarıya çıkış olarak alınabilir veya diğer hücrelere giriş olacak şekilde kullanılabilir.

V. Bir hücre, dış girişleri, komşu hücre çıkışlarını ve bir eşik değerini giriş olarak kabul eder.

VI. Hücre çıkışındaki lineer olmayan cebrik fonksiyon tek giriş-tek çıkışlıdır ve özel olarak lineer olabilen herhangi bir fonksiyondur.

VII. Hücre parametreleri, hücreden hücreye değişebilir.

VIII. Ağ, genelde çok boyutlu katmanların ileri veya geri beslemeli olacak şekilde bağlanmasından oluşur.

IX. Ağdaki her katman, hücrelerin çok boyutlu bir dizidir.

X. Katman içinde, her bir hücre komşularına (belli bir metrik uyarınca) aynı şekilde bağlıdır. Böylece katman içi bağlantı geometrisi düzgündür.

XI. Katmanlar arası bağlantı, bir metrik ile tanımlı düzgün bir geometriye sahiptir.

XII. Ağda yalnızca bağlantı ağırlıkları değişebilir.

XIII. Bağlantı ağırlıkları, ya eğiticili bir öğrenme kuralı ile ya eğiticisiz bir öğrenme kuralı ile değişebilir ya da önceden belirlenen değerlerde sabit tutulur.

XIV. Eğiticili bir öğrenme kuralı (eğer varsa); giriş-çıkış örnek çiftlerinin alındığı, bilinmeyen bir giriş fonksiyonuna bu örnekler yardımı ile yaklaşılmasını sağlayan bir bağlantı ağırlık değişim kuralıdır.

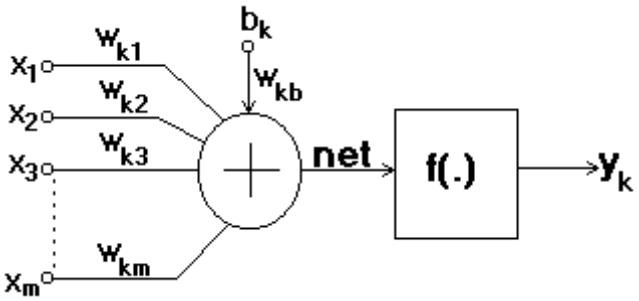
XV. Eğiticisiz öğrenme kuralı (eğer varsa); giriş örneklerinin kümelendirilmesini sağlayan bir bağlantı ağırlık değişim kuralıdır.

Şekil 2.1.'de bir sinir hücresinin modeli görülmektedir. Bu model, yapay sinir ağı dizaynında temel olan bir formdur. Aşağıda bu sinir hücresinin modelinin temel elemanları açıklanmıştır.

1. Snaps yada birleşme ağları: Bunlar ağırlık olarak karakterize edilebilir. Özelde, x_j girişini k. nörona bağlayan ağırlık w_{kj} diye adlandırılır. Ağırlığı tanımlayan ilk alt indis bağlandığı nöronu, ikinci alt indis ise girişi nitelendirmektedir. İnsan beyninden farklı olarak yapay sinirde ağırlığın değeri negatif değer de olabilir.

2. Toplayıcı: Toplayıcı, girişlerle kendilerini nörona bağlayan ağırlıklarla çarpımlarını toplayan bir elemandır.
3. Aktivasyon fonksiyonu: Aktivasyon fonksiyonu sinir hücresinin çıkışını sınırlar. Çıkışın belli genlik değeri içinde tutulmasını sağlar. Giriş sinyalinin sonsuz değerinde bir limit değere ulaşır.

Bunların dışında özel olarak b_k ile gösterilen yanlılık(bias) terimine sahiptir. "Bias", pozitif veya negatif olmasına bağlı olarak, aktivasyon fonksiyonu girişine gelen toplam değeri azaltır veya azaltır.



Şekil 2.1. Yapay sinir ağı modeli

Burada sinir ağına uygulanan giriş kümesi $x_1, x_2, .., x_m, b_k$ şeklinde olup, vektör formunda X olarak ifade edilir. Her bir giriş sinyali kendisi ile ilgili ağırlık $w_{k1}, w_{k2}, ..., w_{km}, w_{kb}$ ile çarpılır. Ağırlıklar da vektörel biçimde W_k olarak gösterilen, bu ikili çarpımların toplamı;

$$net(X, W_k) = W_k^T X \tag{2.1}$$

Burada;

$$W_k = \begin{bmatrix} w_{k1} & w_{k2} & \dots & w_{km} & w_{kb} \end{bmatrix}^T$$

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_m & b_k \end{bmatrix}^T$$

Ağırlıklı toplam, yani $net(X, W_k)$ değerinin hesabından sonra, bunun bir aktivasyon fonksiyonu $f(.)$ üzerinden geçirilmesi ile nöronun çıkışı üretilir.

$y_k = f(W_k^T X)$ yada

$$y_k = f(\sum_{i=1}^{m} w_{kj} x_i) \tag{2.2}$$

2.5. Aktivasyon Fonksiyonları

2.5.1. Treshold Fonksiyonu

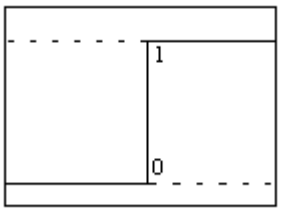
Treshold fonksiyonu şekil 2.2'de görüldüğü gibidir. Bu fonksiyon ;

$$f(net) = \begin{cases} 1, & net \geq 0 \\ 0, & net < 0 \end{cases} \tag{2.3}$$

yada

$$f(net) = sign(net) \tag{2.4}$$

gibi gösterilebilir.



Şekil 2.2. Treshold fonksiyonu

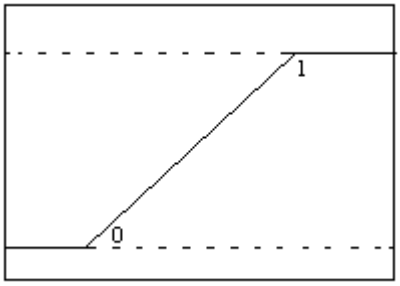
Bu modelde, nöron çıkışı, $net(X, W_k^T)$ değeri pozitif ise 1 aksi halde 0'dır. Bu durum, McCulloch&Pitts modeli özelliklerini gösterir.

2.5.2. Parçalı-Lineer Fonksiyon

Parçalı-lineer fonksiyon aşağıda tanımlandığı gibidir.

$$f(net) = \begin{cases} 1, & net \geq 1 \\ net, & 1 > net > -1 \\ -1 & net \leq -1 \end{cases} \tag{2.5}$$

Şekil 2.3'de bu fonksiyonun grafiği görülmektedir.



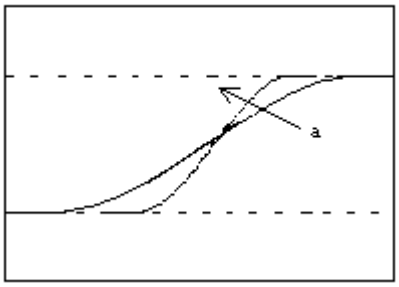
Şekil 2.3. Parçalı-Lineer Fonksiyon

2.5.3. Sigmoid Fonksiyon

Sigmoid fonksiyonu, yapay sinir ağlarında yukarda anlatılan fonksiyonlardan oldukça farklıdır. Bu fonksiyon lineer davranış ile nonlineer davranış arasında çok güzel bir denge sağlar. Bu fonksiyon ile nonlineer sistemlerin davranışı daha iyi karakterize edilebilir. Bu fonksiyon aşağıdaki gibi tanımlanır.

$$f(net) = \frac{2}{1+\exp(-a \cdot net)} - 1 \qquad (2.6)$$

Burada a, sigmoid fonksiyonun eğim parametreleri olarak adlandırılır. Bu parametrenin değişmesiyle şekil 2.4'de görüldüğü gibi farklı eğimlerde sigmoid fonksiyon elde edilir. a parametresinin sonsuza gitmesi ile sigmoid fonksiyonu treshold fonksiyonu haline dönüşür. Bununla beraber sigmoid fonksiyonunda sınır değerler arasında değişim görülürken, treshold fonksiyonunda değişme olmaz.



Şekil 2.4. Sigmoid fonksiyonu

Sigmoid fonksiyonu yerine kullanılan diğer bir fonksiyon ise tanjant hiperbolik fonksiyonudur.

$$f(net) = \tanh(net) \tag{2.7}$$

Yukarıda açıklanan aktivasyon fonksiyonları –1 ile 1 aralığında tanımlıdır. Fakat bazı durumlarda fonksiyonun 1 ile 0 arasında tanımlanması istenebilir. Böyle bir durumda (2.5) 'deki fonksiyonlar aşağıdaki gibi tanımlanır.

$$f(net) = \begin{cases} 1, & net \geq \frac{1}{2} \\ net, & \frac{1}{2} > net > -\frac{1}{2} \\ 0 & net \leq -\frac{1}{2} \end{cases} \tag{2,8}$$

Sigmoid fonksiyonu 1 ile 0 arasında tanımlanmak isteniyorsa

$$f(net) = \frac{1}{1 + \exp(-a \cdot net)} \tag{2.9}$$

şeklinde bir fonksiyon kullanılır.

2.6. Yapay Sinir Ağı Modelleri

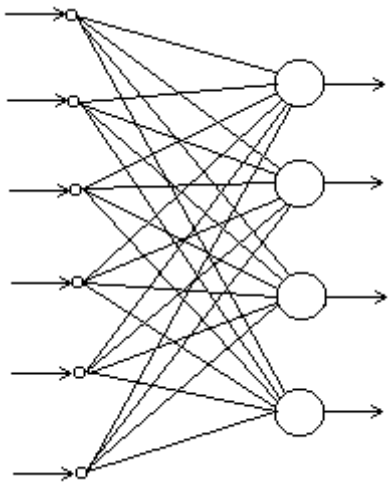
Son 50 yılda çok sayıda ağ modelleri geliştirilmiştir. Bu modellerde biyolojik sinir sisteminin prensiplerinden ve mühendislikte bilinen teorilerden yararlanılmıştır. Genel olarak yapay sinir ağları, iki sınıfa ayrılır: İleri beslemeli ağlar ya da cebrik ağlar ve geri beslemeli ağlar ya da dinamik ağlar. İleri beslemeli ağlar, örüntü tanıma(pattern recognition) problemlerinde, geri beslemeli ağlar ise optimizasyon problemlerinin çözümünde başarılı olarak kullanılabilmektedir.[14]

2.6.1. İleri Beslemeli Ağlar

İleri beslemeli ağlar, işaretin ağ girişinden çıkışına doğru tek yönlü olarak iletildiği ağlardır. Bir katmandaki hücreler, ancak bir önceki katmanın çıkışlarını giriş olarak alabilir. İleri beslemeli ağlar kendi içinde ikiye ayrılır.

2.6.1.1. Tek Katmanlı İleri Beslemeli Ağlar

Genelde tek bir hücre, istenen giriş-çıkış işlevini yerine getiremez. Bunun için bağlantı ağırlıkları dışında aynı özelliklere sahip hücreler bir araya getirilerek katmanlar oluşturulur. En basit şekliyle, katmanlı bir sinir ağı giriş katmanı olarak giriş sinyallerinin ağa uygulandığı katman düşünülebilir. Giriş katmanındaki sinyaller ağırlıklarla çıkış katmanına bağlanmıştır. Böyle bir yapı, tek katmanlı yapay sinir ağı olarak isimlendirilir. Bu yapıda bilgi işlemesi girişten çıkışa doğrudur. Giriş katmanı, hesaplama yapılmadığı için katman olarak sayılmaz. M girişli ve N çıkışlı tek katmanlı ileri beslemeli bir ağ şekil 2.5'de gösterilmiştir.[33]

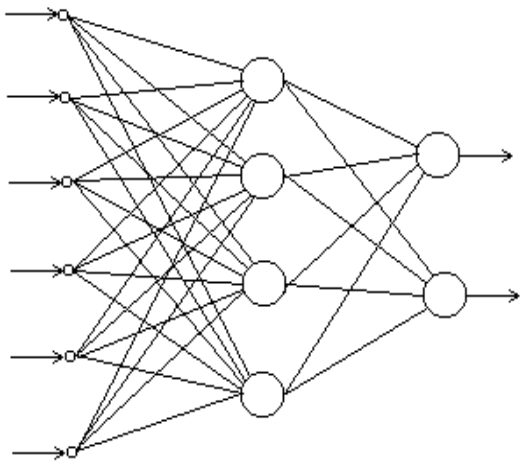


Şekil 2.5. Tek katmanlı ileri beslemeli ağ

2.6.1.2. Çok Katmanlı İleri Beslemeli Ağlar

Bu tip sinir ağları, bir yada daha fazla gizli katman bulundurmaları ile ayırt edilebilirler. Gizli katman, giriş katmanı ile çıkış katmanı arasında ilişkiyi oluşturur. Gizli katmanın eklenmesiyle, yapay sinir ağının yüksek derecede matematik işlemler

Giriş katmanının boyutu arttığına, gizli katmanın yüksek derecede matematik işlem yapma kapasitesi daha bir önem taşımaktadır. Şekil 2.6'de çok katmanlı ileri yol ağlarına bir örnek görülmektedir. Buradaki ağ yapısı, tam bağlı ileri besleme ağ yapısıdır. Bazı networklerde her nöron her nöronla bağlı olmayabilir. Böyle yapılara ise parçalı bağlı ağ yapıları denir.

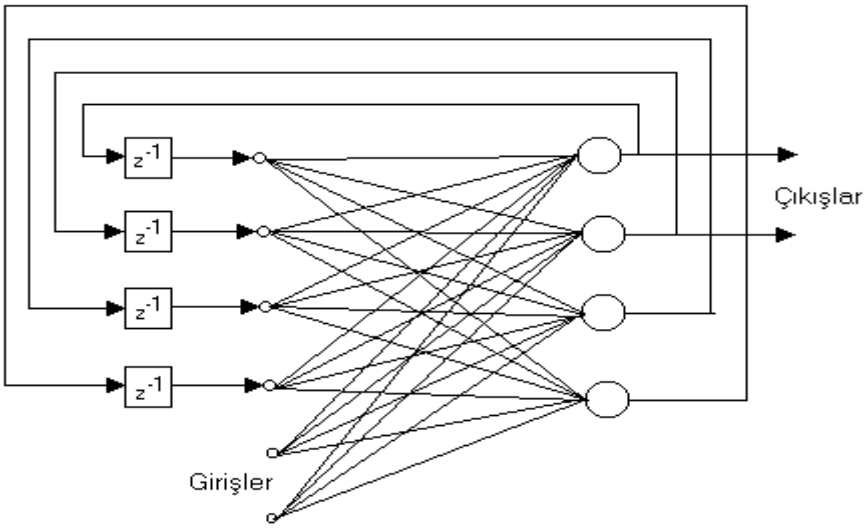


Şekil 2.6. Çok katmanlı ileri beslemeli ağ

2.6.2. Geri Beslemeli Yapay Sinir Ağları

Bu tür ağlarda en belirgin özellik çıkış katmanından elde edilen sinyalin giriş katına uygulanmasıdır. Şekil 2.7.'de bu yapıya bir örnek görülmektedir. Bu şekilde gösterilen yapıda öz geri besleme yoktur. Öz geri besleme, her bir nöronun çıkışının kendi girişlerine tekrar uygulanmasıdır. Ayrıca bu yapı tek katmalı bir sinir ağının geri beslemesidir. Ayrıca çok katmalı ağlarda da geri besleme uygulanabilir.

Geri besleme döngüsü yapay sinir ağının performansında ve öğrenme kabiliyetinde çok büyük etki oluşturur. Geri beslemeli ağlarda birim zaman gecikme elemanı bulunur. Bunların özelliği nonlineer dinamik davranış sergilemeleridir.



Şekil 2.7. Tek katmanlı geri beslemeli yapay sinir ağı

2.7. Yapay Sinir Ağlarında Öğrenme

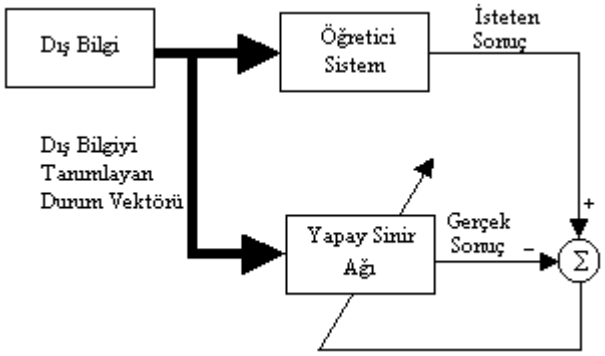
Yapay sinir ağlarında öğrenme işlemi, sinir hücreleri arasındaki bağlantıyı sağlayan bağların ağırlıklarının belli bir kurala göre değiştirilmesi işlemi ile sağlanır. Öğrenme kuralı, giriş işaretlerine ve transfer fonksiyonu tarafından sağlanan değerlere cevap olarak yerel bellekteki ağırlıkların hepsini veya bazılarını değiştiren bir denklem olarak bilinir. Öğrenme kuralı giriş işaretlerinin doğasına bağlı olarak, işleme elemanının zamanla değişmesine imkan sağlar. Bu şekilde ağ kendisini istenen cevaplara uyarlayabilir ve kendi içinde bilgiyi düzenleyebilir, yani öğrenebilir.

Yapay sinir ağları programlama yerine örneklerle eğitilir. Çocukların annelerini tanımak için karşılaştırmalı fizyoloji hakkında hiçbir şey bilmedikleri gibi, programcılar sinir ağına tanıtılacak cisimlerin nicel tanımlarını veya söz konusu cisimleri benzer cisimlerden ayırmak için lojik kriter kümeleri sağlamak zorunda değillerdir. Bunun yerine bir sinir ağına bazen tanımları ile beraber (anne, masa veya dağlar) cisim örnekleri (yüzler, nesneler veya görüntüler) girilir. Yapay sinir ağlarına öğrenme için sunulacak eğitim kümesi, girişlerden ve varsa çıkışlardan oluşmaktadır. Her adımda eğitim kümesinden sırayla veya rastgele örnekler alınmakta ve ağın bu

girişlere ilişkin çıkışı bulunmaktadır. Adım sonunda veya tüm örnekler bir defa ağa uygulandıktan sonra (çevrim sonunda) belirli bir amaç ölçütü uyarınca ağın bağlantı ağırlıkları değiştirilmektedir. Ağırlıkların değiştirilmesiyle öğrenme sağlanmış olur. Öğrenme, adım sonunda değiştiriliyorsa veri uyarlamalı, çevrim sonunda değiştiriliyorsa grup uyarlamalı öğrenme olarak adlandırılır.

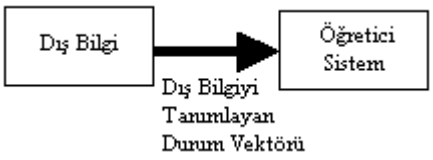
Sinir ağlar denetimli ve denetimsiz olmak üzere iki şekilde eğitilmektedir:

Denetimli öğrenmede, ağ' a hem giriş hem de istenen çıkış bilgisi(hedef vektörü) girilir. Her denemeden sonra ağ kendi çıkışını doğru cevaplar ile karşılaştırır ve çıkış hatası kabul edilebilecek seviyeye kadar ağırlıklarını değiştirerek iterasyon yapar.



Şekil 2.8. Denetimli öğrenme blok diyagramı

Denetimsiz öğrenmede hiçbir hedef vektörü yoktur. Giriş vektörü sisteme uygulanır. Sistem, bu giriş vektörü sisteme uygulandığında uyumlu bir çıkış (muhtemelen eğitimden önce tahmin edilemeyen) üretecek şekilde kendini değiştirebilir. Böyle sistemler sınıflama problemleri için kullanılmışlar ve özellikle Kohonen ile Grossberg tarafından geliştirilmişlerdir.



Şekil 2.9. Denetimsiz öğrenme blok diyagramı

Her iki eğitim türünden sonra da ağ gerçek girişleri işlemek için hazırdır. Bu noktada, yapılacak işin niteliğine bağlı olarak bir operatör ağın kendisini yeni bilgiye uyarlamasını durdurmak ve işlemesini hızlandırmak için öğrenme kuralını devreden çıkartıp ağırlıkları dondurabilir. Örneğin, eğer ağ yazılı bir metni sesli olarak okumak için yeteri kadar eğitilmiş ise gerçek bilgi üzerinde öğrenmesini devam ettirmeye gerek yoktur. Öte yandan örneğin ağ, yakıtı harcadıkça kütlesi azalan, yörüngedeki bir uzay aracının hareketini kontrol etmekte kullanılacaksa, ağ kendisini değişen şartlara uyarlamaya devam ettirmelidir. Böyle sistemlerde, sadece denetimsiz öğrenmenin kullanılması anlamlı olmaktadır. Belli başlı öğrenme kuralları aşağıda anlatılmıştır.

2.7.1. Hebb Öğrenme Kuralı

Hebb öğrenme kuralına göre öğrenme sinyali sinir hücresinin çıkışına eşittir. Buna göre;

$$r = f(w_i^T x) \tag{2.10}$$

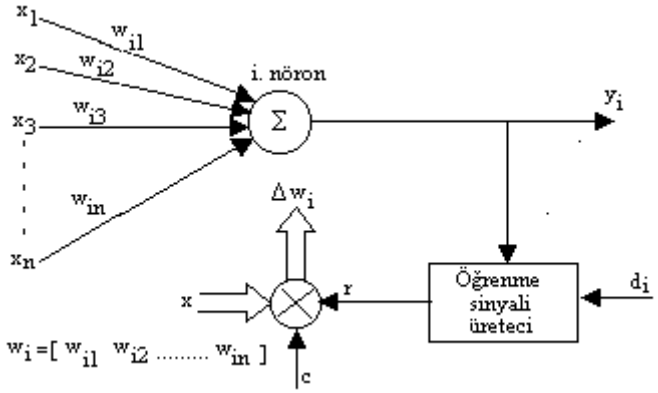
olur. Buna karşılık ağırlık vektöründeki artış aşağıdaki gibi tarif edilir.

$$\Delta w_i = cf(w_i^T x)x \tag{2.11}$$

veya

$$\Delta w_{ij} = cy_i x_j \tag{2.12}$$

Bu öğrenme kuralı başlangıçta sıfıra yakın değerler ile başlamayı gerektirmektedir. Hebb kuralı tam bir ileri beslemeli, eğiticisiz öğrenme kuralıdır. (2.12) nolu ifadeye dikkat edilecek olursa şunu anlayabiliriz. Eğer giriş ve çıkıştan oluşan çarpım pozitif ise o zaman w_{ij} ağırlığında bir artma, negatif ise bir azalma olur.



Şekil 2.10. Hebb öğrenme kuralının şematik hali

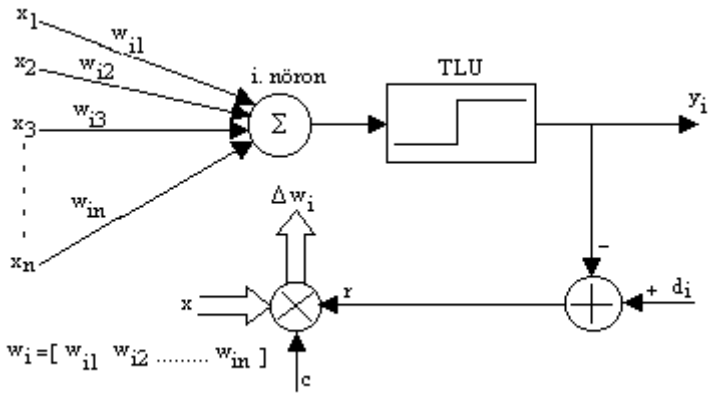
2.7.2. Perceptron Öğrenme Kuralı

Perceptron öğrenme kuralı için öğrenme sinyali nöronun istenen çıkışı ile gerçek çıkışı arasındaki fark olarak alınır. Böylece öğrenme eğiticili öğrenme sınıfına girer. Bu öğrenme sinyalini aşağıdaki gibi ifade edebiliriz.

$$r = d_i - y_i \tag{2.13}$$

Burada $y_i = \text{sgn}(w_i^T x)$, d_i de nörondan istenen çıkıştır. Bu kuralda elde edilen hataya göre ağırlık uyarlamaları da aşağıdaki gibidir.

$$\Delta w_{ij} = c\left[d_i - \text{sgn}(w_i^T x)\right]x_j \tag{2.14}$$



Şekil 2.11. Perceptron öğrenme kuralının şematik hali

(2.14) nolu ifadeden görüldüğü üzere bu kural sadece bipolar ikili çıkış veren nöronlara uygulanabilir. Bu ağırlık uyarlaması sadece y_i yanlış olduğu zaman yapılır. Eğitime rastgele başlangıç şartları ile başlayabiliriz. Perceptron öğrenme kuralı eğiticili öğrenme ile eğitilen yapay sinir ağlarında çok önemli bir yere sahiptir.

2.7.3. Widrow-Hoff Öğrenme Kuralı

Widrow-Hoff öğrenme kuralı da eğiticili öğrenme tabanlı bir kuraldır. Bu kural aktivasvon kuralının doğasından bağımsızdır. Çünkü nöronun istenen çıkışı ile gerçek çıkışı arasındaki hatayı minimize etmeye çalışır. Bu kural için öğrenme sinyali aşağıdaki gibi verilmiştir.

$$r = d_i - w_i^T x \tag{2.15}$$

Buna karşılık ağırlık uyarlaması da aşağıdaki şekle gelir.

$$\Delta w_{ij} = c(d_i - w_i^T x)x_j \tag{2.16}$$

Widrow-Hoff öğrenme kuralı Delta öğrenme kuralının özel bir durumu olarak kabul edilir.

2.7.4. Delta Öğrenme Kuralı

Delta öğrenme kuralı sadece sürekli aktivasyon fonksiyonları ve eğiticili öğrenme durumlarında geçerlidir. Delta diye adlandırılan öğrenme sinyali aşağıdaki gibidir.

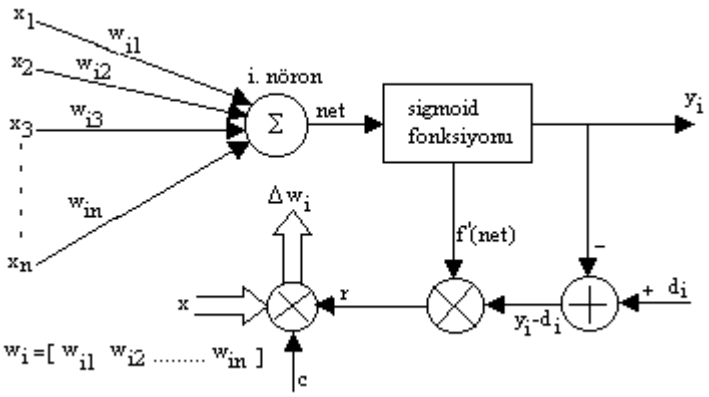
$$r = \left[d_i - f(w_i^T x)\right] f'(w_i^T x) \tag{2.17}$$

Yukarıdaki ifadede $f'(w_i^T x)$, aktivasyon fonksiyonu $f(net)$'in türevidir. Bu kural istenen çıkış ile gerçek çıkış arasındaki hatanın karesinin minimizasyonu

şartından çıkartılabilir. Son olarak Delta öğrenme kuralında ağırlık uyarlamaları aşağıdaki ifadeye göre yapılmaktadır.

$$\Delta w_{ij} = \eta(d_i - y_i) f'(net_i) x_j \tag{2.18}$$

Burada η öğrenme sabiti olup 0ile 1 arasında bir değerde olur.



Şekil 2.12. Delta öğrenme kuralı şematik gösterimi

2.8. Geriye-Yayılım (Back-Propagation) Algoritması

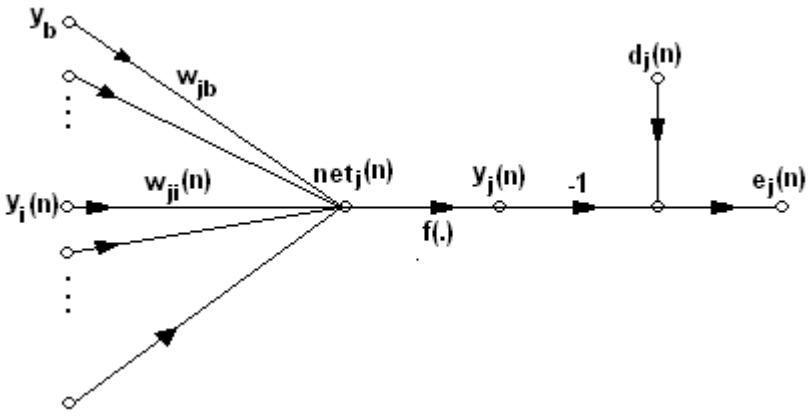
Çıkış katmanındaki j. nöronun hata sinyali (2.19)'daki gibi tanımlanmıştır.

$$e_j(n) = d_j(n) - y_j(n) \tag{2.19}$$

Burada n, iterasyonun eğitim örneğinde olduğunu gösterir. j. nörona ait toplam karesel hata, $\frac{1}{2}e_j^2(n)$ ifadesinin tüm eğitim kümesi için toplanmasıyla elde edilir. Çıkış katmanındaki tüm nöronların toplam karesel hatası ise şöyle gösterilir.

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n) \tag{2.20}$$

Burada C, çıkış katmanındaki nöronların hepsini içermektedir. $E(n)$, toplam karesel hata, sinir ağındaki bütün serbest parametrelerin (ağırlıklar ve bias seviyesi) bir fonksiyonudur. Öğrenme işleminin amacı, bu serbest parametrelerin değerlerini karesel toplam hatanın, $E(n)$, değerini minimum yapacak şekilde değiştirmektir.



Şekil 2.13. Çıkış katmanındaki j. nöronu detaylı olarak gösteren sinyal akış grafiği

Şekil 2.13.'de gösterilen $net_j(n)$ çıkış katmanındaki j. nöronun aktivasyon fonksiyonunun girişidir. Aşağıdaki gibi ifade edilir.

$$net_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \qquad (2.21)$$

Burada, m, j. nöronun toplam giriş sayısını (bias değeri dahil) göstermektedir. w_{jb} ağırlığı (girişi değeri $y_b = +1$) j. nörona uygulanan bias değerine eşittir. Buna göre n. iterasyondaki j. nöronun çıkış değeri aşağıda gösterildiği gibi olur.

$$y_j = f_j(net_j) \qquad (2.22)$$

Ağırlıkların hangi yönde değişeceğine karar verebilmek için $E(n)$'nin $w_{ji}(n)$ ağırlıklarına göre negatif gradyantının $(\nabla E(n))$ bulunması gerekir. Daha

sonra ağırlıklar toplam karesel hatayı azaltacak şekilde ayarlanır. Zincir kuralına göre $(\nabla E(n))$ gradyantının gösterimi aşağıdaki gibidir.

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{ji(n)}} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial net_j(n)} \frac{\partial net_j(n)}{\partial w_{ji}(n)} \tag{2.23}$$

(2.20) nolu eşitliğin her iki tarafının $e_j(n)$ 'ye göre türevini alırsak;

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \tag{2.24}$$

ifadesini elde ederiz. (2.19) nolu eşitliğin her iki tarafının $y_j(n)$ 'ye göre türevini alırsak;

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \tag{2.25}$$

ifadesini elde ederiz. (2.21) nolu eşitliğin her iki tarafını $net_j(n)$ ' ye göre türevini alırsak;

$$\frac{\partial y_j(n)}{\partial net_j(n)} = f_j'(net_j(n)) \tag{2.26}$$

ifadesini elde ederiz. Son olarak, (2.22) nolu eşitliğin her iki tarafını $w_{ji}(n)$ 'ye göre türevini alırsak;

$$\frac{\partial net_j(n)}{\partial w_{ji}(n)} = y_i(n) \tag{2.27}$$

ifadesini elde ederiz. (2.23) nolu eşitliğin içinde (2.24), (2.25), (2.26), (2.27), nolu ifadeleri kullanacak olursak aşağıdaki eşitlik oluşur.

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{ji(n)}} = -e_j(n) f_j'(net_j(n)) y_i(n) \tag{2.28}$$

Delta kuralına göre ağırlık uyarlaması aşağıdaki gibi yapılır.

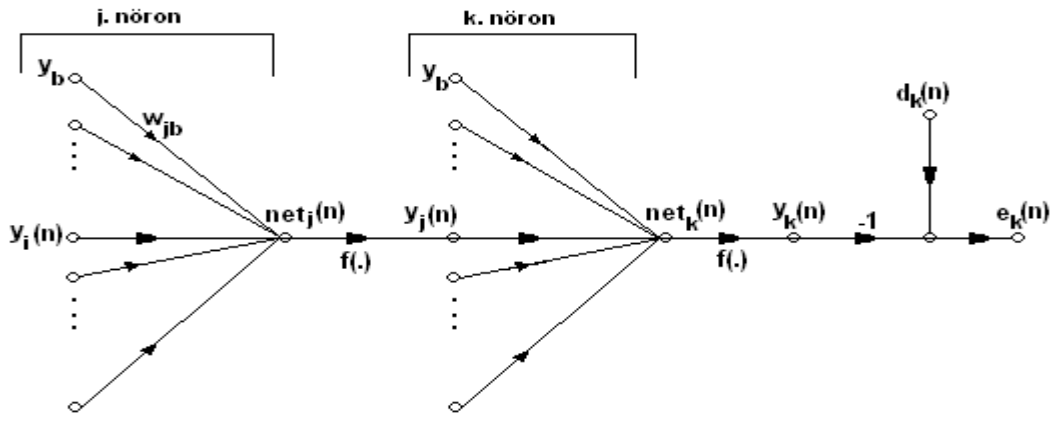
$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \tag{2.29}$$

Burada η, geriye yayılım algoritmasının öğrenme sabitidir. (2.29) nolu eşitlikte kullanılan eksi işaretinin sebebi ağırlık değişiminin $E(n)$' yi azaltacak yönde olması istendiği içindir. (2.28) ve (2.29) nolu ifadeleri kullanarak

$$\Delta w_{ji}(n) = -\eta \delta_j(n) y_i(n) \tag{2.30}$$

eşitliğini elde ederiz. Burada $\delta_j(n)$ lokal gradyant olarak tanımlanır ve aşağıdaki gibi ifade edilir.

$$\begin{aligned} \delta_j(n) &= -\frac{\partial E(n)}{\partial net_j(n)} \\ &= -\frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial net_j(n)} \\ &= e_j(n) f_j'(net_j(n)) \end{aligned} \tag{2.31}$$



Şekil 2.14. Çıkış nöronuna gizli nöron eklenmiş halde sinyal akış diyagramı

Lokal gradyant ağırlıkların değişim miktarını gösterir. (2.31) nolu ifadeye göre j. çıkış nöronunun lokal gradyant, $\delta_j(n)$, bu nöronun hata sinyalinin, $e_j(n)$, yine bu nöronun aktivasyon fonksiyonunun türevi, $f_j'(net_j(n))$, ile çarpılması ile elde edilir.

Eğer gizli katmandaki bir nöronun lokal gradyantını hesaplamak istiyorsak şekil 2.14.'de gösterildiği gibi bir durum ortaya çıkar. Bu durumda nöronun çıkışından elde edilmesi istenen belli bir değer yoktur. Gizli katmadaki bir nöronun hata sinyalini elde edebilmek için bu nöronun çıkışına bağlı tüm nöronların hata sinyallerini hesaplamak gereklidir. Bu durum da geriye yayılım algoritmasını daha karmaşık hale getirmektedir. (2.31) nolu ifadeyi tekrar gizli katmandaki bir nöron için düzenlersek ;

$$\begin{aligned}\delta_j(n) &= -\frac{\partial E(n)}{\partial y_j(n)}\frac{\partial y_j(n)}{\partial net_j(n)} \\ &= -\frac{\partial E(n)}{\partial y_j(n)} f_j'(net_j(n))\end{aligned} \tag{2.32}$$

ifadesini elde ederiz. Burada $\partial E(n)/\partial y_j(n)$ kısmi türevini hesaplamak için aşağıdaki işlemler yapılmalıdır. Şekil 2.14.'den görüldüğü üzere çıkış katmanındaki k. nöronun karesel toplam hatası aşağıdaki gibi hesaplanır.

$$E(n) = \tfrac{1}{2}\sum_{k\in C} e_k^2(n) \tag{2.33}$$

burada kullanılan k indisi (2.20) nolu ifadede kullanılan j indisinin yerine kullanılmıştır. Çıkış katmanını ifade eder. (2.33) nolu ifadenin $y_j(n)$'ye göre türevini alırsak aşağıdaki eşitliği elde ederiz.

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)} \tag{2.34}$$

Sonra $\partial e_k(n)/\partial y_j(n)$ kısmi türevini hesaplamak için (2.34) nolu eşitlikte zincir kuralı uygulanır.

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial net_k(n)} \frac{\partial net_k(n)}{\partial y_j(n)} \tag{2.35}$$

Ayrıca, şekil 2.14'den görüleceği üzere

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - f_k(net_k(n)) \end{aligned} \tag{2.36}$$

Burada k. nöron çıkış katmanına aittir. Böylece aşağıdaki ifade elde edilir.

$$\frac{\partial e_k(n)}{\partial net_k(n)} = -f_k'(net_k(n)) \tag{2.37}$$

Yine şekil 2.14.'den görüleceği üzere;

$$net_k(n) = \sum_{j=0}^{m} w_{kj}(n) y_j(n) \tag{2.38}$$

Burada m, k. nörona uygulanan toplam giriş sayısıdır. (2.38) nolu ifadenin $y_j(n)$ 'ye göre türevini alırsak;

$$\frac{\partial net_k(n)}{\partial y_j(n)} = w_{kj}(n) \tag{2.39}$$

ifadesini elde ederiz. (2.37) ve (2.39) nolu ifadeleri (2.35) nolu eşitliğin içinde kullanarak istenilen kısmi türev elde edilmiş olur.

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_k e_k(n) f_k'(net_k(n)) w_{kj}(n)$$

$$= -\sum \delta_k(n) w_{kj}(n) \tag{2.40}$$

Burada $\delta_k(n)$, (2.31) nolu ifadede tanımlanan lokal gradyanttır. Tek fark j indisi yerine k indisinin kullanılmış olmasıdır.

Son olarak, (2.32) ve (2.40) nolu ifadelerden, gizli katmandaki *j*. nöronun lokal gradyantının geriye yayılım formülü elde edilir.

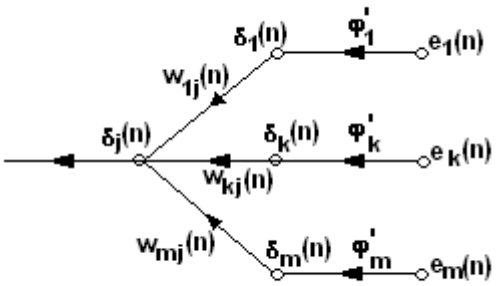
$$\delta_j(n) = f_j'(net_j(n)) \sum \delta_k(n) w_{kj}(n) \tag{2.41}$$

Şekil 2.15., (2.41) ifadesinin sinyal akış diyagramında ifadesini göstermektedir.

Yukarıda anlatılan geriye yayılım algoritmasını toparlamak gerekirse; $\Delta w_{ji}(n)$, *i.* nöronu *j*. nörona bağlayan ağırlığın değişimi olmak üzere delta kuralının tanımına göre

$$\left(\Delta w_{ji}(n)\right) = \left(\eta\right) \cdot \left(\delta_j(n)\right) \cdot \left(y_i(n)\right) \tag{2.42}$$

ifadesini elde ederiz. Burada $y_i(n)$, *j.* nöronun giriş sinyalidir.



Şekil 2.15. Hata sinyalinin geriye yayılımını gösteren sinyal akış diyagramı

3. KALMAN FİLTRESİ

3.1. GİRİŞ

GPS tekniği ile konumların belirlenebilmesi, yeni uygulama alanlarının açılmasını sağladığı gibi, klasik yöntemlerle çözüm bulunabilen bir çok uygulama için de yeni yöntemlerin arayışını hızlandırmıştır. Günümüzde, klasik ölçme ve değerlendirme aşamaları yerini bilgisayar destekli ve anında çözümler üretebilen sistemlere bırakmıştır. GPS tekniği ile birkaç santimetre doğruluğunda anlık konumların belirlenebilmesi, dinamik sistemlerin hızlı bir şekilde izlenebilmesi olanağını da sağlamıştır. Sistem ve kontrol mühendislerinin bu gibi olanakları kullanarak birçok ölçme ve kontrol sistemlerini oluşturmaları, klasik değerlendirme ve kestirim algoritmalarının yerine, optimal kestirim ilkesini koruyan ve bilgisayarlarda daha hızlı çalışarak anında sonuçlar üretebilen algoritmaların arayışına girilmesine neden olmuştur. Bu algoritmaların başında Kalman Filtresi gelir.

Kalman Filtresinden önce, kestirim teorilerinde bazı önemli gelişmeler olmuştur. R. A. Fishher, en büyük olasılıklı kestirim (maximum likelihood estimation) fikrini geliştirdi. Daha sonra 1941'de Kolmogorov, 1942'de Nobert Wiener, birbirlerinden bağımsız olarak bir filtre modeli oluşturdular.[16] Bu filtre literatürde Wiener Filtresi yada Wiener-Kolmogorov Filtresi olarak da anılır. Ancak

Gauss'un yaklaşımı ile Wiener ve Kolmogorv'un yaklaşımı arasında bazı farklılıklar vardır. Bu filtre gerçekte kestirim problemlerini en küçük kareler yöntemine göre çözmemekteydi. Buna rağmen Wiener Filtresi de Kalman Filtresinin matematiksel alt yapısını oluşturan önemli gelişmelerden biridir.

1955'de John Hopkins Üniversitesi'nden J. W. Follin, dinamik sistemler için tekrarlı bir yaklaşım modeli geliştirdi. R. E. Kalman'la birlikte Richard Bucy'nin çalışmalarının gelişmesini sağlamıştır. [16] R. E. Kalman, geliştirdiği yeni bir lineer filtre modelini 1960 yılında yayınlamıştır. [17] Söz konusu bu model literatürde Kalman Filtresi olarak anılır. Şekil 3.1.'de Kalman Fitresinin matematiksel yapı taşları gösterilmektedir.



Şekil 3.1. Kalman Filtresinin Matematiksel Temeli

Kalman Filtresi 1960 yılında ortaya atıldıktan sonra elektrik mühendisleri ve istatikçiler tarafından geliştirilmiş ve mühendislik uygulamalarının bir çoğunda kullanılmaya başlanmıştır. Kalman Filtresi genel olarak bütün dinamik sistemlerin durum değişkenlerinin kestiriminde kullanılabilir. Dinamik sisteme ilişkin parametreler ve birbirleri ile ilişkileri, filtrenin dinamik modelinde doğru bir şekilde tamınlanabildiği ve ölçülerle bağı da uygun bir şekilde kurulabildiği sürece, bütün dinamik sistemlerin durum değişkenlerinin kestirimi Kalman Filtresinin konusu olabilirler.

Yöntemin anlık (real-time) çalışmalarda kullanılabilmesi, diğer bir çok yönteme göre avantaj sağlamış ve uygulama alanlarının da genişlemesine neden olmuştur. GPS tekniği ile anlık konum belirlenebilmesi ve Kalman Filtresinin de ölçümlerdeki bozuntuları izole etmesi bu ikisinin deniz, kara ve hava taşıtlarının anlık olarak izlenmesi uygulamalarında birlikte kullanılmasını sağlanıştır. [18] Kalman Filtresi, kimyadan fiziğe, jeodoziden sistem mühendisliği uygulamalarına kadar yüzlerce uygulamada kullanılmıştır.

Yapay sinir ağlarının gelişmesi ile birlikte Kalman Filtresi bu alanda da uygulama alanı bulmuştur. Bizzat yapay sinir ağının eğitim algoritmasında kullanıldığı gibi[19][20] sistem tanımlamada da[21] kullanılmıştır. Ayrıca kontrol sistemlerinde [22][23], hedef izleme alanında [24] ve daha pek çok alanda beraber uygulanmışlardır.

3.2. Bir Dinamik Model Kestiriminde Optimallik

Kalman Filtresi sistem hatalarının istatiksel özelliliğini, sistem dinamiği ile birleştirir. Sistem durumunun kestirimi için, Kalman Filtresinde sistem dinamiğinin durum uzayı denklemleri kullanılır Durum vektörleri, minimum hata varyansını üretecek optimize edilmiş Kalman kazancı ile hesaplanır. Bu nedenle Kalman Fitresi, optimal filtre diye de adlandırılır. Lineer sistemler için Kalman Filtresi en iyi kestirimi sağlayacak bir yapıya sahiptir.

Bir kestirimin optimal olup olmadığı, kestirim sonuçlarının gerçek kabul edilebilecek değerlere yakınlığına göre değerlendirilir. Fakat pratikte, gerçek değerler bilinmediği için kestirim sonuçlarının gerçek değerlere yakınlığından bahsedilemez. Bu nedenle bir kestirimin optimal olup olmadığı, en küçük kareler yöntemine göre belirlenir. Diğer bir deyişle, kestirim sonucu bulunan düzeltmelerin kareleri toplamının en küçük olması şartı, optimallik kriteri olarak düşünülür.

Bir dinamik model kestiriminde bu kriterin sağlanması için tek boyutlu bir kestirim problemini düşünelim. Başka bir deyişle durum vektörü sadece bir bilinmeyenden oluşsun. Herhangi bir t anında bu bilinmeyen x_t'nin, ölçme modelinden bulunan kestirimi x_t^m ve dinamik modelden bulunan öncül kestirim x_t^- olsun. Buna göre;

$$\left(x_t - x_t^m\right)^2 + \left(x_t - x_t^-\right)^2 \rightarrow \text{ minimum} \qquad (3.1)$$

ifadesi minimum şartı sağlıyorsa; kestirim sonucu bulunan $\hat{x}_t$'ye, x_t'nin en küçük kareler kestirimi yada optimal kestirimi denir. x_t^m ve x_t^-'nin farklı ağırlıklara sahip olması durumunda en küçük kareler koşulu aşağıdaki gibi olur.

$$P_m\left(x_t - x_t^m\right)^2 + P_w\left(x_t - x_t^-\right)^2 \rightarrow \text{minimum} \qquad (3.2)$$

Gerçekte durum vektörleri çok boyutlu olurlar. Bu durumda, ölçmelere gelen düzeltmeler V vektörü ile ve dinamik modelden bulunan öncül kestirimlere gelen düzeltmeler w vektörü ile gösterilirse, optimal kestirim aşağıdaki θ fonksiyonun minimum olmasıyla tanımlanır.

$$\theta = V^T P_m V + w^T P_w w \qquad (3.3)$$

Burada P_m, ölçme modelinin ağırlık matrisi, P_w, dinamik modelin ağırlık matrisidir. Sonuç olarak, en küçük kareler kestirimi, optimal olarak düşünülür ve aşağıdaki özelliklerin oluşmasını sağlar.

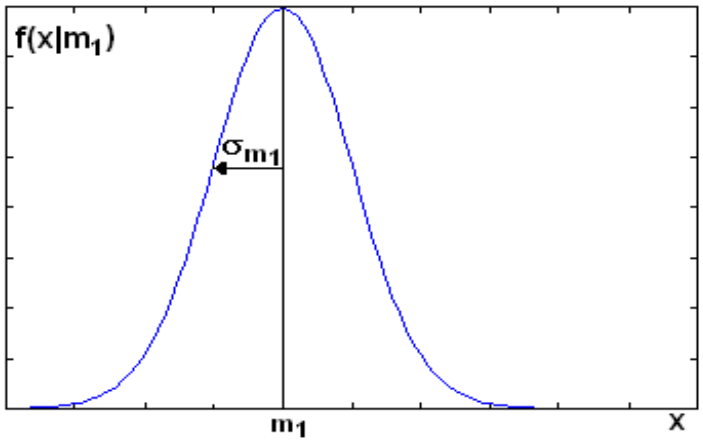
1. Kestirimler matematik beklentiye(expected value) sadıktır.
2. Kestirimler en küçük varyansa sahiptirler.
3. Eğer hatalar normal dağılımda ise kestirimler, kestirimler en olasılıklı değerlerdir. [25]

Kalman Filtresi de bir dinamik sistemin kestirimini optimal kestirim yöntemi ile tamınlar. Gerçekte filtre sonucu bulunan değerler, en küçük kareler kestirimiyle özdeştir.

3.3. Kalman Filtresi'ne Giriş

Kalman Filtresi'ni daha iyi anlayabilmek için şöyle bir örnekle başlayabiliriz. Bizim elimizde bir ölçü aleti olsun. Bu ölçü aleti ile t_1 anından bir ölçme yaptığımızda ölçü aletinden okunan değer m_1 olarak okunsun. Ancak aletsel,

gerek kişisel gerekse dış ortamlardan dolayı kaynaklanan hatalardan dolayı m_1 ölçmesinin σ_{m1} gibidir standart sapmaya sahip olması kaçınılmazdır. Böylece t_1 anında x_1 değerinin m_1 ölçme şartına bağlı olasılığı tahmin edilebilir. x değerinin normal (Gauss) bozuntu dağılımlı şartlı olasılık fonksiyonu şekil 3.2'deki gibidir.



Şekil 3.2. m_1 ölçme şartına bağlı şartlı yoğunluk fonksiyonu

Ölçümün σ_{m1}'in değerinin büyüklüğü, bize bozuntunun bu ölçüm içindeki miktarını gösterir. σ_{m1} değerinin büyük olması olasılıklı değerin x değeri boyunca geniş bir alana yayılmasına neden olacaktır. Başka bir deyişle bu ölçü aletinin, ölçüm kalitesinin iyi olmadığını gösterir. Sonuç olarak t_1 anı için en iyi kestirim;

$$\hat{x}_1 = m_1 \tag{3.4}$$

ve bu kestirimin varyansı;

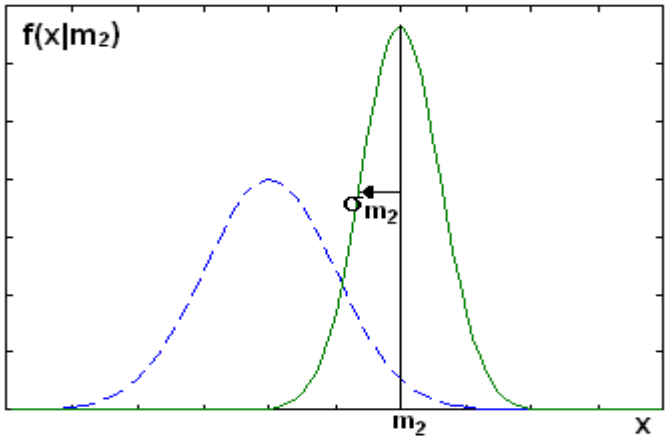
$$\sigma_1^2 = \sigma_{m1}^2 \tag{3.5}$$

şeklinde hesaplanabilir. Bu arada t_2 anında σ_{m2} standart sapmasına sahip bir ölçüm aleti ile m_2 ölçmesi yapılmış olsun. İkinci ölçüm aletinin standart sapması ilkinden küçük olduğu düşünülsün. Bu durumda m_2 şartına bağlı ilkine göre daha dar bir olasılık eğrisi oluşturacaktır. Bu durum şekil 3.3.'de görülmektedir. Yapılan ikinci ölçme birincisinden daha doğru olduğu için sinyalin büyüklüğünün kestirimi olarak m_2 değeri kabul edilebilir. Fakat her iki ölçmeyi de kullanarak m_1 ve m_2 ölçmelerine bağlı olarak yeni bir kestirimde bulunabilir. m_1 ve m_2 ölçmelerine bağlı olarak yeni kestirimin matematik beklentisi

$$E(x) = \mu = \left[\frac{\sigma_{m2}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_1 + \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_2$$

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{m1}^2} + \frac{1}{\sigma_{m2}^2} \tag{3.6}$$

şeklinde hesaplanır.



Şekil 3.3. m_2 ölçme şartına bağlı şartlı yoğunluk fonksiyonu

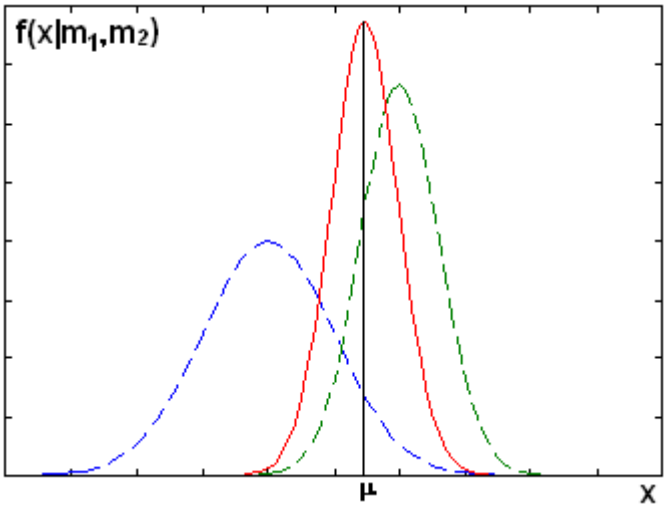
Yeni kestirimin standart sapması σ hem σ_{m1}'den hem de σ_{m2}'den küçük olacaktır ve bu yeni kestirimin her iki ölçmeden de daha doğru olacağı kesindir. Buna göre yeni kestirim;

$$\hat{x}_2 = \mu \tag{3.7}$$

olacaktır ve bu kestirimin varyansı da σ^2'dir. Bu durumu şekil 3.4.'de görebiliriz. (3.6) ve (3.7) ifadelerini kullanarak aşağıdaki ifadeyi yazabiliriz.

$$\hat{x}_2 = \left[\frac{\sigma_{m2}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_1 + \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_2$$

$$= m_1 + \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] (m_2 - m_1) \tag{3.8}$$



Şekil 3.4. m_1 ve m_2 ölçme şartına bağlı şartlı yoğunluk fonksiyonu

(3.6) ve (3.7) ifadelerini kullanarak aşağıdaki ifadeyi yazabiliriz.

$$\hat{x}_2 = \left[\frac{\sigma_{m2}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_1 + \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] m_2$$

$$= m_1 + \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right](m_2 - m_1) \qquad (3.8)$$

$\hat{x}_1 = m_1$ olduğu düşünülürse Kalan Filtresinin temel formu aşağıdaki gibi yazılabilir.

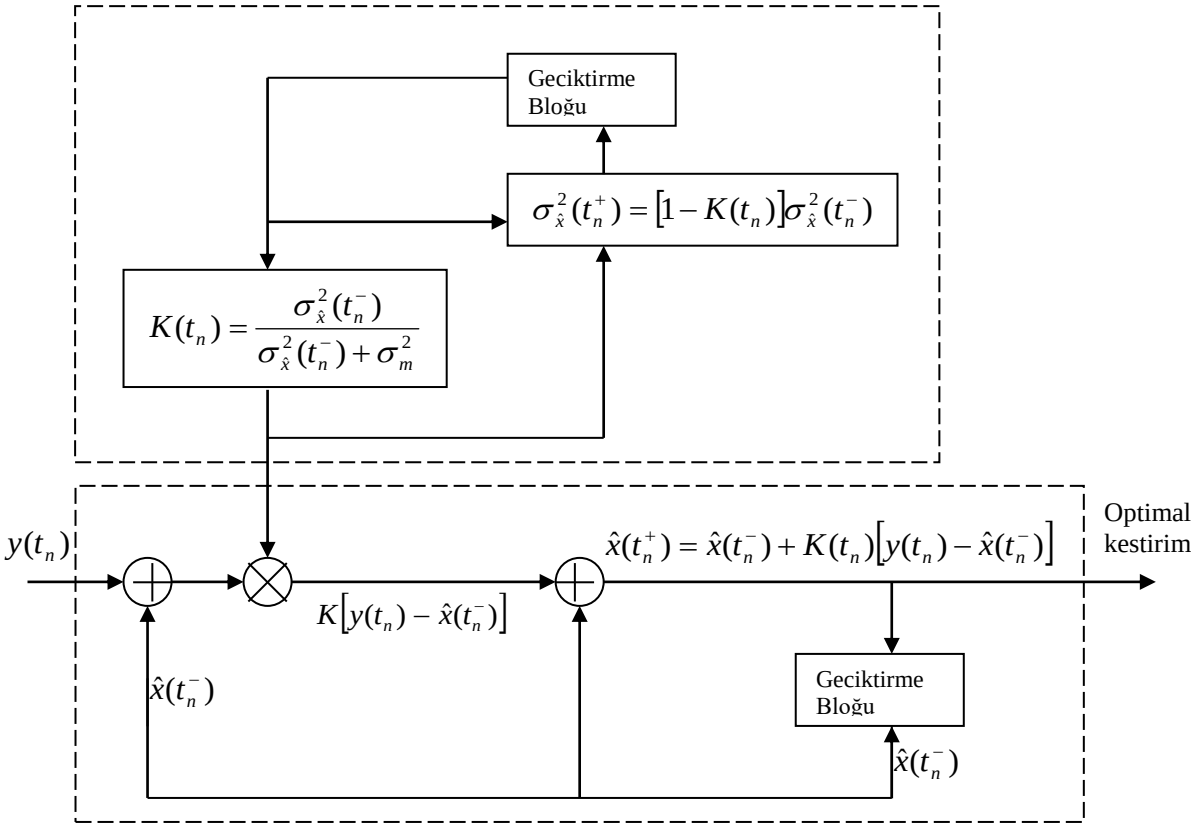
$$\hat{x}_2 = \hat{x}_1 + K_2(m_2 - \hat{x}_1)$$

$$K_2 = \left[\frac{\sigma_{m1}^2}{\sigma_{m1}^2 + \sigma_{m2}^2}\right] \qquad (3.9)$$

Burada K_2 Kalman filtresi kazancıdır. Böylece bir kestirim problemi çözülebilir. Bu işlemleri bir blok diyagram halinde gösterecek olursak şekil 3.5'deki temel filtre yapısını elde ederiz. [25]

Gerçek uygulamalarda durum vektörleri dinamik olarak değişen ve daha fazla değişkenlere sahiptir. Örneğin, bir uçağın durum vektörleri olarak üç yöndeki

hız ve pozisyonları olabilir. Ayrıca, sistem durum modeli gerçek gürültü ve sistem bozuntuları içeren model hatası içermektedir.



Şekil 3.5. Temel Kalman Filtresi

3.4. Kalman Filtresinin Matematik Modeli

Kalman Filtresi üç temel aşamadan oluşur. $t<t_i$ koşuluyla bir dinamik sistemin t_i anındaki durum değişkenlerinin, sistemin önceki özelliklerinden faydalanarak hesaplanmasına öncül kestirim (prediction, extrapolasyon) aşaması denir. Eğer herhangi bir t anında dinamik bir sistemin t_i anındaki durum değişkenleri $t = t_i$ koşulu ile t_i anındaki ölçülerde kullanılarak hesaplanıyorsa, bu aşamaya filtre aşaması (update) denir. Eğer herhangi bir t anında dinamik bir sistemin t_i anındaki durum değişkenleri $t>t_i$ koşuluyla, t_i anına kadar ki ölçülerle birlikte hesaplanıyorsa bu aşamaya da yumuşatma aşaması denir. [17] Anlık uygulamalarda filtre ilk iki aşama ile sınırlı kalmak zorundadır. Çünkü yumuşatma aşaması yalnızca bütün

ölçümler sona erdikten sonra uygulanabilir ve anlık ölçümlerde böyle bir şans yoktur.

Kalman Filtre denklemi iki kısımdan oluşmaktadır. Bunlar sırayla dinamik model ve ölçme modeli ;

$$x_k = \Phi x_{k-1} + Bu_k + w_{k-1} \tag{3.10}$$

$$z_k = Hx_k + v_k \tag{3.11}$$

Burada $x \in \Re^n$ ayrık zamandaki sistemin durum vektörüdür. $z \in \Re^m$, durum vektörünün gözlem matrisidir. w_k ve v_k, sistem ve ölçüm bozuntularını göstermektedir. Ölçüm bozuntuları birbirinden bağımsız ve gauss dağılımına uygun rastgele değişkenlerdir.

$$\begin{aligned} p(w) &\approx N(0, Q) \\ p(v) &\approx n(0, R) \end{aligned} \tag{3.12}$$

Burada, Q, sistem gürültü kovaryans matrisi, R, ölçüm gürültü kovaryans matrisidir. Bu değerler gerçekte her ölçümde değişmektedir. Ancak hesaplarda kolaylık
olması sebebiyle bunları sabit olarak almaktayız.

A matrisi $n \times n$ boyutunda olup k-1 adımdaki durumu k adımına bağlayan geçiş matrisidir. B matrisi $n \times l$ boyutunda kontrol matrisidir. $u \in \Re^l$ kontrol vektörüdür. Kontrol vektörünün her sistemde olması gerekli değildir. Ölçülecek sistemin yapısına göre kullanılıp kullanılmayacağına karar verilir. H, sistemin $m \times n$ boyutlu gözlem matrisidir.

$\hat{x}_k^- \in \Re^n$, K. adımdaki öncül kestirim (ekstrapolasyon) olarak tamınlayalım. $\hat{x}_k \in \Re^n$, k. adımdaki ölçüm değerini, z_k, kullanarak elde edilen filtrelenmiş durum vektörüdür. (3.9) denklemini tekrar yazarsak aşağıdaki denklemi elde ederiz.

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \tag{3.13}$$

$(z_k - H\hat{x}_k^-)$, innonasyon süreç olarak tamınlanır ve $\tilde{z}$ olarak da gösterilir. İnnovasyon süreç tahmin edilen ölçüm, $H\hat{x}_k^-$, ile gerçek ölçüm, z_k, arasındaki farklılığın derecesini gösterir. Eğer innovasyon süreç sıfır ise tahmin edilen değer ile gerçek ölçüm değeri birbirleri ile tamamen aynıdır. $n \times m$ boyutundaki K katsayısı, optimal filtrenin matris kazanç katsayısıdır.

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ &= \frac{P_k^- H^T}{HP_k^- H^T + R} \end{aligned} \tag{3.14}$$

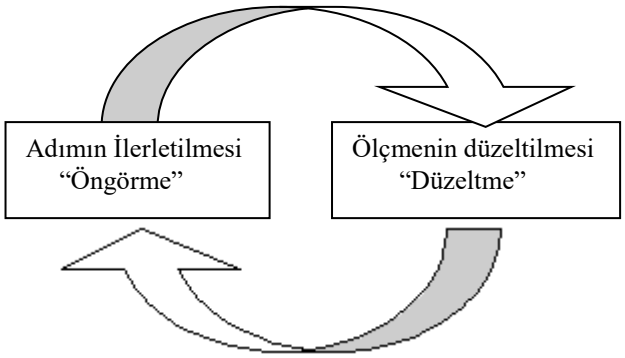
Burada, ölçüm hata kovaryansı, R ,azaldıkça K Kalman kazancı artmaktadır. R, sıfıra yaklaştıkça K değeri de H^{-1}'e yaklaşır. Diğer taraftan öncül kestirim hata kovaryansı, P_k^-, sıfıra yaklaştıkça K kazancıda sıfıra yaklaşacaktır. Başka bir deyişle ölçüm hata kovaryansı, R, sıfıra yaklaştıkça gerçek ölçüm değeri z_k'nında güvenilirliği artar, tahmin edilen ölçüm değerinin güvenilirliği azalır. Aksi durumda öncül kestirim hata kovaryansı, P_k^-, sıfıra yaklaştıkça gerçek ölçüm değeri z_k'nında güvenilirliği azalır, tahmin edilen ölçüm değerinin güvenilirliği artar.

Öncül kestirim hata kovaryansı, P_k^-, aşağıdaki gibi tanımlanır.

$$P_k^- = AP_{k-1}A^T + Q \tag{3.15}$$

Kalman Filtresi yapısını iki gruba ayırabiliriz. Birincisi "adımın ilerletilmesi (time update)", ikincisi "ölçümün düzeltilmesi (measurement update)" olarak isimlendirilebilir. Adımın ilerletilmesinde, bir sonraki adım için öncül kestirimin elde edilmesinde kullanılan durum ve hata kovaryans kestirimlerinin bir adım sonraya taşınması sağlanır. Ölçümün düzeltilmesi ise yeni ölçüm değeri ile öncül kestirim değeri kullanılarak filtrelenmiş durum vektörü elde edilir. Olaya şu açıdan da bakılabiliriz. Adım ilerletme, ölçümle ilgili öngörmede bulunurken, ölçümün

düzeltilmesi, yapılan bu öngörmeyi düzeltmektedir ve gerçek ölçüme yaklaştırmaktadır. O zaman Kalman Filtresini aşağıdaki şekildeki gibi gösterebiliriz.



Şekil 3.6. Ayrık Kalman Filtre çevrimi.

Kalman Filtresinin çalışması şu şekilde olmaktadır. Sistem çalışmaya başlamadan önce ölçüm hata kovaryansı, R, ve sistem hata kovaryansı, Q, belli olmalıdır. Sistem çalışmaya başladığında ilk önce t_0 anı için başlangıç değerleri atanır ya da ilk ölçülerden hesaplanır. Daha sonra bu başlangıç değerlere göre t_1 anı için durum vektörünün ve ilk öncül kestirimi yapılır ve ölçül kestirimin hata kovaryansı hesaplanır. t_1 anı geldiğinde öncül kestirimin hata kovaryansından Kalman kazancı elde edilir. t_1 anındaki ölçme kullanılarak Kalman kazancı ve öncül kestirimle birlikte filtrelenmiş durum vektörü elde edilir. Sistem ardışık olarak ölçmelerin sona erdiği n. adıma kadar bu şekilde çalışmaya devam eder. Şekil 7.'de Kalman Filtresinin çalışmasını gösteren diyagram görülmektedir. [26]

Kalman filtresinin önemli özellikleri aşağıda verilmiştir:

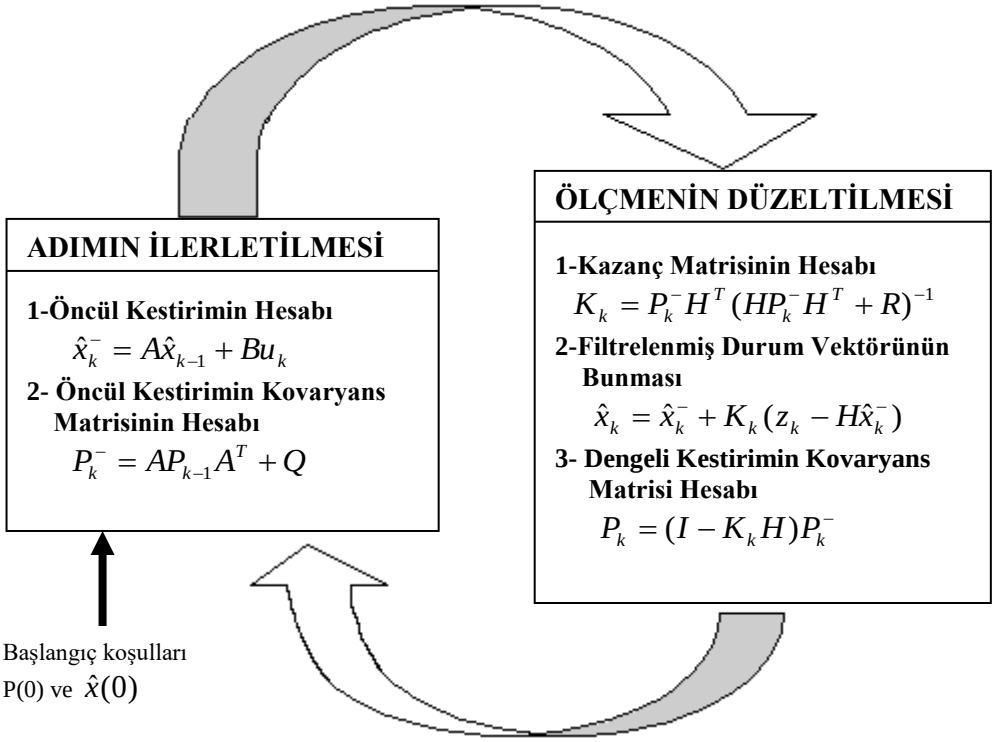
1- Dinamik sistemin matematik modeli tamamen belli olduğunda, filtre bilgisayarda kolay olarak gerçekleştirilir

2- Filtrenin yardımı ile bulunmuş değer ölçüme göre lineerdir

3- Filtreleme hatasının korelasyon matrisi P_k filtreinin lineer olması nedeniyle z_k ölçümüne bağlı değildir ve önceden hesaplanabilir

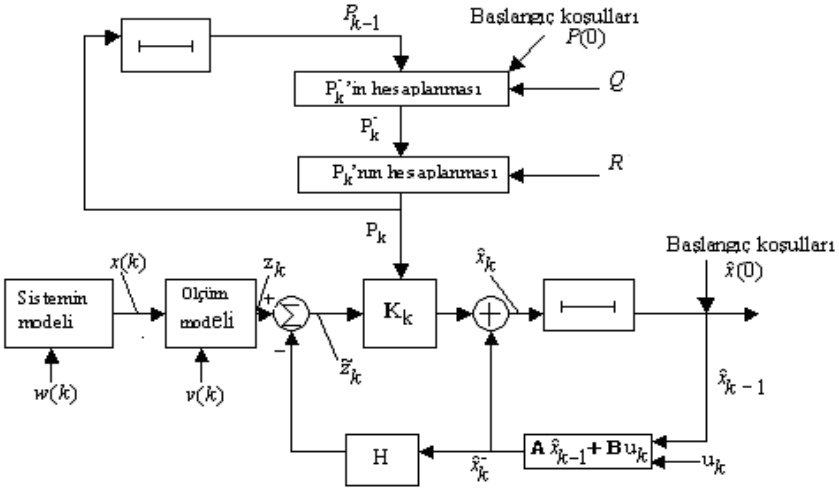
4- Filtreleme algoritmaları çok boyutlu durum için kolay olarak uygulanabilir

5- Durgun dinamik sistemler için kararlı durumda Kalman filtresi Viner filtresi ile denk düşer.[27]



Şekil 3.7. Kalman Filtresinin çalışmasını gösteren diyagram

Kalman Filtresinin yapı şeması ise şekil 3.8.'de gösterilmektedir.



Şekil 3.8. Kalman filtresinin yapı şeması **[**27]

Navigasyon ölçümlerinin işlenmesi probleminde, Kalman filtresi geniş bir şekilde uygulanmaktadır. Bu filtre, aşağıdaki amaçlarla kullanılır [28]:

1) Ölçüm hatalarının küçültülmesi ve ölçülen parametrelerin daha doğru değerlerinin bulunması;

2) Çeşitli bilgi kaynaklarının kompleksleştirilmesi;

3) Uçağın durum vektörünün ölçülemeyen parametrelerinin bulunması;

4) Uçakta meydana gelen bozuntuların teşhisi,

Sistemin matematik modeli önceden bilinmediğinde veya çalışma sürecinde değiştirildiğinde, adaptif filtrelerin kullanılması daha önemlidir. Bu zaman değerlendirme işlemi parametrelerin ve (veya) modelin yapısının identifikasyonu işlemi ile birleştiriliyor.

Denklemler nonlineer olduğunda ise önce onların lineerleştirilmesi gerekir.

4. UYDU RADYO NAVİGASYON SİSTEMLERİ

4.1. Giriş

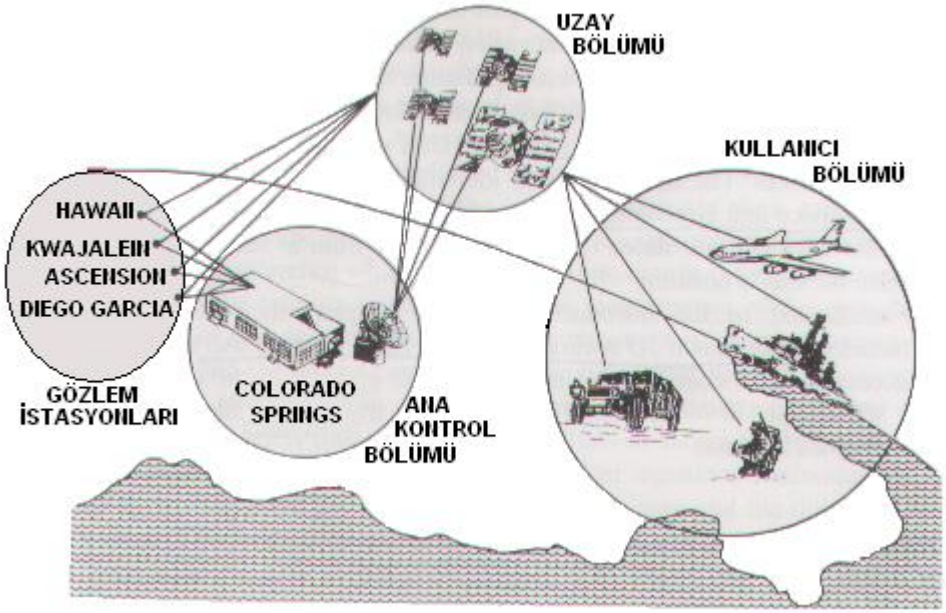
Teknolojik gelişimle birlikte ilk uydunun (SNUPTIK 1) Rusya tarafından 1 Kasım 1957'de uzaya fırlatılmasıyla uzay çalışmaları başlamış ve bu ilk fırlatmadan sonra değişik görevler için değişik uyduların uzaya fırlatılması çalışmalarına hız verilmiştir. Rusya'nın gönderdiği SNUPTIK 1 uydusu ile yapılan gözlemlerin tetiklediği ilk uydu navigasyon sistemi olan TRANSIT 1'i geliştirme çalışmaları 1958'in sonlarında başlamıştır. TRANSIT 1 1964'e kadar süren çalışmalar sonunda tamamlanmıştır. [29]

1973 yılında Pentagon'da toplanan bazı kuvvet komutanlıkları temsilcileri ve sivillerin hazırladığı ve navigasyonda devrim olacak plan ile NAVSTAR'ın temelleri atılmış oldu. NAVSTAR, diğer adı ile GPS; halen günümüzde kullanılan 24 uydudan oluşan uydu navigasyon sistemidir. Bu sistem ile herhangi bir hava koşulunda, dünyanın herhangi bir yerindeki herhangi bir kişinin konumunu tespit edilebilmektedir. 1978 yılında ise bu projenin ürünü olan ilk prototip uydu yörüngeye yerleştirilmiştir. On beş yıllık gir gelişme ve iyileştirmeden sonra GPS sistemi 1993 yılında kuruluş aşamasını tamamlamış ve tam kapasite çalışmaya başlamıştır. [30]

4.2. Uydu Navigasyon Sistemlerinin Yapısı

Uydu navigasyon sistemleri üç temel kısımdan oluşmaktadır.

- Uzay bölümü
- Kullanıcı bölümü
- Kontrol bölümü



Şekil 4.1. Uydu Navigasyon Sistemi

4.2.1. Uzay Bölümü

Uzay bölümü, 6 yörünge ve her bir yörüngede 4 uydu olmak üzere 24 uydudan oluşmaktadır. Yörüngeler 60° aralıkta ve ekvator düzlemi ile 55º'lik açı yapacak şekilde planlanmıştır. Uydular yaklaşık yerden 20000 km yükseklikte ve yörünge zamanları 11 saat 58 dakikadır. Uydular içinde anten, osilatör, sinyal kaydedici, atomik saat, mikroişlemci, sinyal gönderici gibi elektronik ve mekanik yapılardan oluşmuş yaklaşık 845 kg ağırlığındadır. Dünyanın herhangi bir yerinden en az 4 en fazla 10 uydu gözlemek mümkündür. Uydular görevleri için gerekli olan enerjiyi her biri 7.2 m^2 olan iki güneş kollektöründen sağlarlar. Ayrıca uyduda

bulunan 3 tane nikel-kadmiyum pil ise uydunun güneşi görmediği zamanlar uyduya gerekli olan enerjiyi sağlamaktadır.

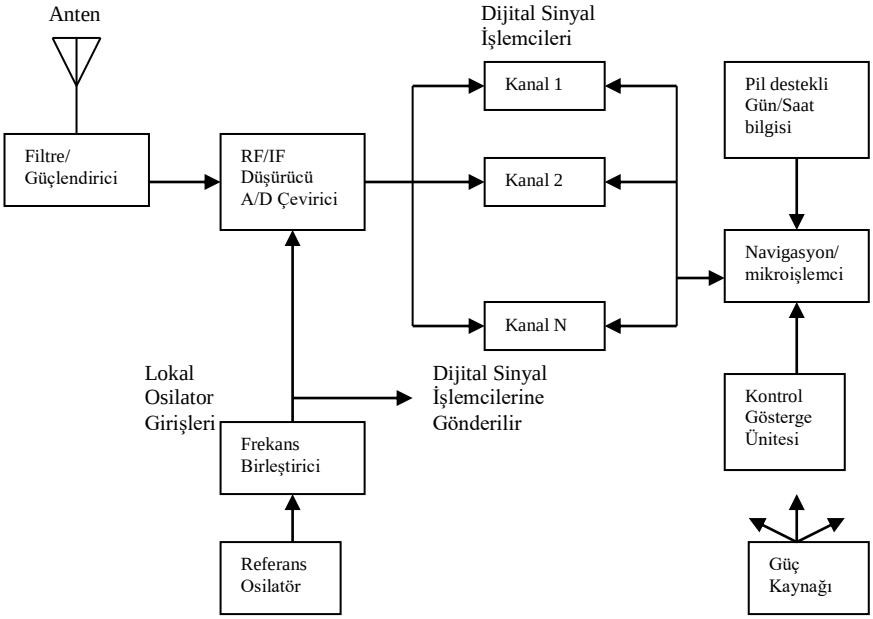
GPS uyduları ana kontrol merkezince hesaplanan ve yeryüzündeki yer antenleri aracılığı ile gönderilen yörünge bilgilerini alır ve düzeltilmiş zaman bilgileri ile birlikte 2 temel frekans üzerinden yayınlar. Uydularda L1 ve L2 taşıyıcı frekansları bulunmaktadır. Bu taşıyıcı frekanslardan L1 ile P ve C/A kodları, L2 taşıyıcı frekansı ile P kodu , sistem mesajları ve navigasyon bilgileri iletilmektedir. Uydu sinyalleri 10.23 Mhz'lik temel frekanstan oluşmaktadır. Uydunun bu temel frekansı 154 ile çarpılırsa L1 taşıyıcı dalga frekansı(1575.42), 120 ile çarpılırsa L2 taşıyıcı frekans(1227.60) elde edilir. P kodu 10.23 Mhz'lik temel frekanstan, C/A kodu ise P kodunun 0.1 ile çarpılmasıyla meydana gelmiştir. İki ayrı frekansta taşıyıcı dalga kullanılmasının amacı iyonosfer tabakasının meydana getirdiği kırılmadan dolayı oluşacak gecikmeyi ortadan kaldırmaktır. [29]

4.2.2. Kullanıcı Bölümü

Kullanıcı bölümü GPS alıcısı diye de adlandırabiliriz. NAVSTAR-GPS kullanıcı bölümü bir anten, bir alıcı, gerekli donanıma sahip data işlemci ve görüntü kontrol ünitesinden oluşur. Teknolojinin gelişmesine paralel olarak kullanıcı bölümünde önemli ölçüde gelişmeler olmuştur. Analog aletlerden dijital aletlere bir dönüşüm yaşanmıştır. 1970'li yıllarda üretilen ve ilk olarak askeri amaçla kullanılan GPS alıcıları çok büyük ve ağır aletlerdi. Bugünün teknolojisi ile bu aletler daha fazla kabiliyete sahip olmakla birlikte bir "kol saati" büyüklüğüne indirilmiştir. GPS alıcısı seçimi kullanılacak uygulamaya göre değişebilir. Şekil 4.2'de GPS kullanıcı bölümünün blok diyagramı görülmektedir.

GPS alıcısının çalışma prensibi şu şekildedir. Anlık konum tespiti için kullanıcı 4 veya daha fazla uydudan gelen sinyalleri ölçerek kendisinin üç boyutlu konumu, zamanı ve eğer hareketli ise kendi hızını hesaplayabilir. Uydudan gelen sinyaller antenler yardımı ile alınır ve güçlendirilerek alıcı kısmına aktarılır. Sinyali değerlendirecek olan sinyal işleyiciye bilginin ulaşması için sinyal daha düşük bir frekansa çevrilir. Bundan sonra ise uydudan gelen sinyallerdeki data bilgileri çözülür ve anlaşılır duruma getirilir. Bu işlemin yapılabilmesi için P ve C/A kodlarının

bilinmesi gerekir. Ayrıca alıcıya gelen sinyallerin hangi uydudan geldiğini tespit etmek için sinyal işleyici her bir uydunun sahip olduğu kendine has kodları çözer. Diğer bir önemli konu ise sinyallerin değerlendirildiği kanal sayısıdır. Kanal sayısının artması ile daha çok sayıda uydudan sinyal işleme olanağı artacaktır. GPS alıcısındaki kanal sayısını arttırmak fiziksel olarak mümkün olduğu gibi uygun bir programla da kanal sayısı arttırılabilir. GPS alıcı aletinin diğer bir önemli ünitesi aletin mikroişlemci ünitesidir. Bu ünite ile tüm sistem yönlendirilir ve navigasyon işlemi yapılır. Enerji kaynağı alet için gerekli olan enerjiyi sağlar. GPS alıcılarının çoğu 12V'luk bir gerilim ile çalışır. GPS aletinin kullanıcı ile iletişimi sağlamak amacı ile display ve klavye kullanılır. İşlenen bilgilerin kaydedilmesi için alıcıya monte edilmiş olan bilgisayarın hafızası kullanılabilir. [30]



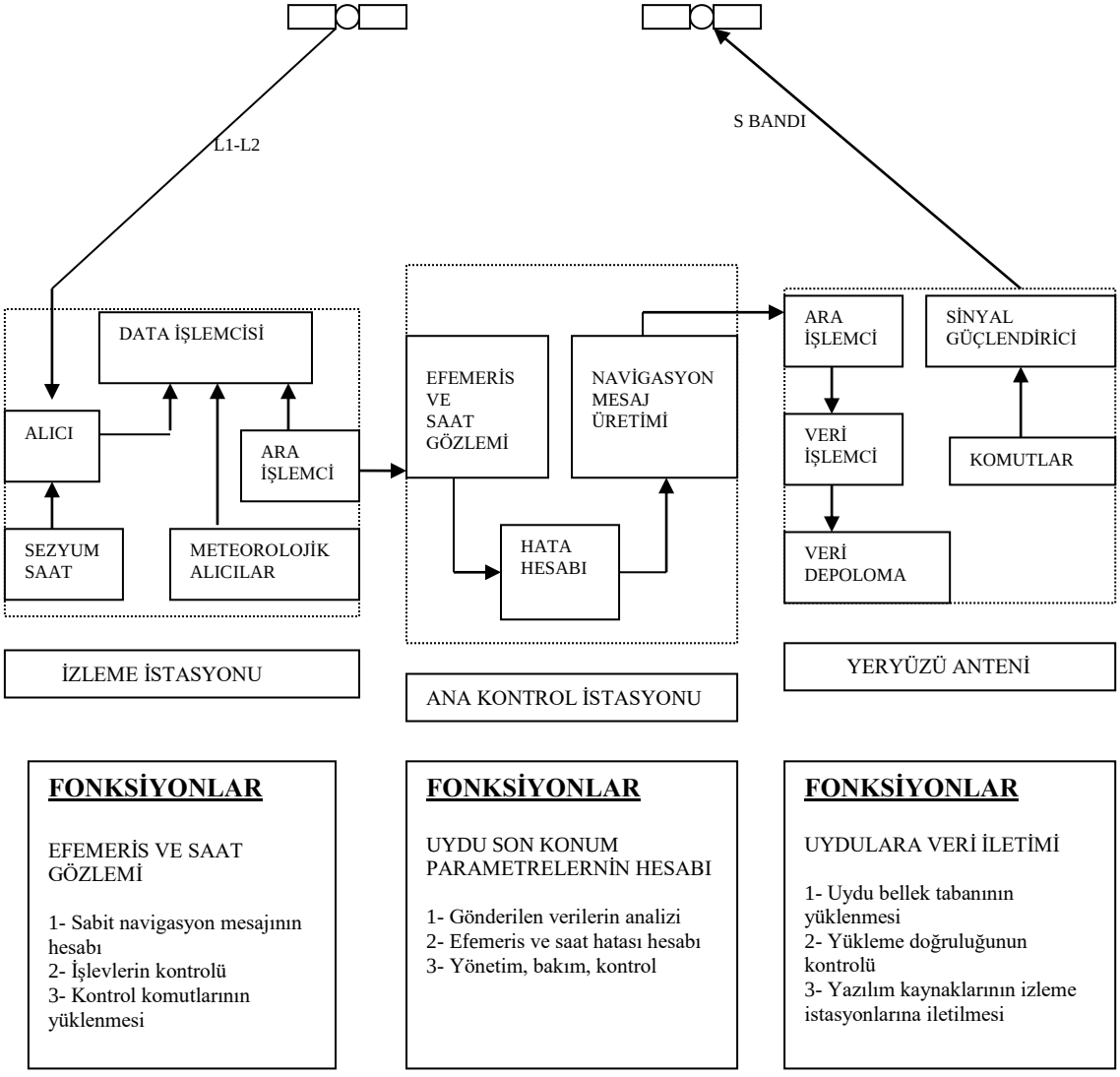
Şekil 4.2. GPS alıcı bölümü elemanları

4.2.3. Kontrol Bölümü

Kontrol bölümünü birbirinden farklı üç ayrı elemandan (istasyondan) oluşur.

- Gözlem istasyonu
- Ana kontrol istasyonu
- Yer anteni

Ana kontrol istasyonu Amerika'da Colorado Springs'te kurulmuştur. Diğer izleyici (gözlem) istasyonlar Hawaii, Kwajalein adası, Ascension adası, Diego Garcia'da bulunmaktadırlar. Kontrol bölümü Amerika Savunma Bakanlığı tarafından işletilir. Gözlem istasyonlarının amacı navigasyon uydularını takip ederek onların hareket parametreleri hakkında bilgileri biriktirmektir. Ana kontrol bölümü gözlem



Şekil 4.3. Kontrol bölümü ve işlemleri

istasyonlarının ölçümlerine dayanarak yörünge parametrelerini ve uyduların sonraki pozisyonlarının bilgilerini önceden belirleyerek gerekli uydu saat düzeltme parametrelerini günceller. GPS sisteminin doğruluğu kontrol bölümünün uyduları günlük olarak yapılan güncellemesine bağlıdır. Ana kontrol bölümünde yer anteni

bulunmamakla birlikte diğer izleyici istasyonlarda yer anteni bulunmaktadır. Yer anteni ana kontrol istasyonunun uydunun yörünge parametreleri hakkında bulduğu sonuçları uyduya gönderir ve uydunun hareketinin genel kontrolünü sağlar. Şekil 4.3'te kontrol bölümü ayrıntılı olarak gösteren blok diyagram görülmektedir. [29] [27][31]

4.3 Bir Cismin Konumunun Navigasyon Uyduları ile Tespit Edilmesi

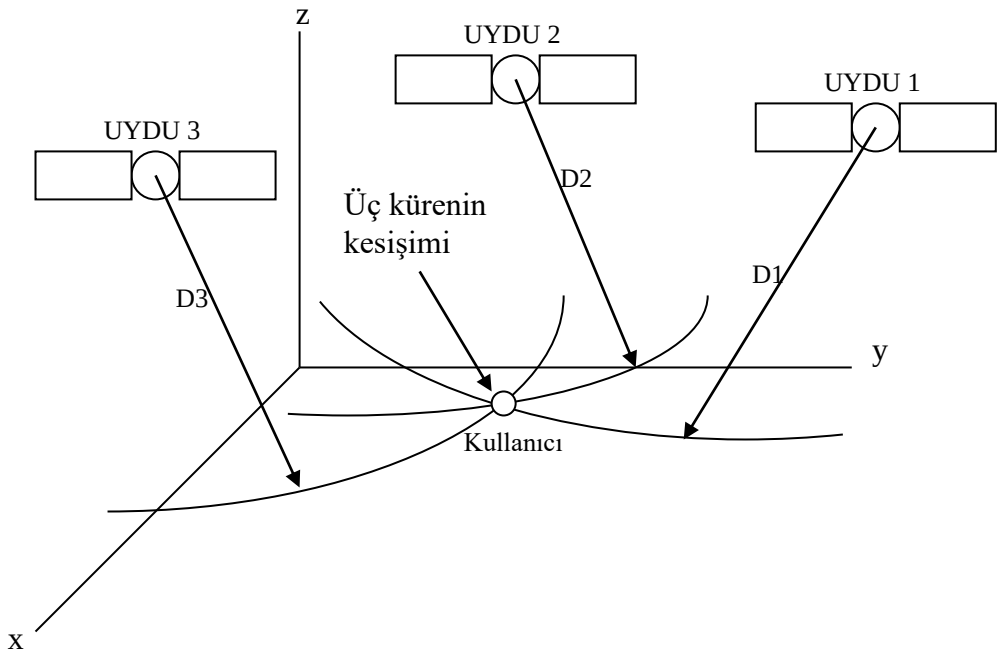
Navigasyon uyduları ile bir cismin konumunu tespit etmek için ilk önce yer istasyonları uydunun yörünge parametrelerini hesaplar sonra bulunan sonuçları uyduya verir ve uyduda bulunan zaman dönüştürücüsünü düzeltir. Uydu belli bir periyotta yörünge ve zaman bilgisi hakkındaki sinyalleri uzaya göndermektedir. Alıcı anteninin yardımı ile uydudan gelen sinyaller alınır. Alınan bu sinyaller güçlendirilir, filtrelendirilir ve kullanılan radyo navigasyon türüne bağlı olarak zaman gecikmesi ve Doppler frekans kayması ölçülür. Tüm bu veriler uçak bilgisayarına verilir. Bilgisayara operatörce kontrol bloğundan Grinviç zamanı, uçağın hızı ve uçağın yönelme açısının değerleri girilir. Söz konusu verilere dayanarak önceden bilgisayara yazılmış program kullanılarak cismin (uçak) koordinatları hesaplanır ve görüntülenir.

Uçağın konumunun tespit olunma doğruluğu aynı zamanda gözlemleri yapılabilinen uyduların miktarına ve onlarla alış-veriş frekansıyla orantılı olarak artar.

Uçağın konumunu mesafe ölçüm yöntemi ile en az üç uydunun yardımı ile tespit etmek mümkündür. Bu durumda navigasyon parametresi olarak uydulardan uçağa kadar olan mesafe ele alınacaktır. Uçağın konumu ise merkezleri uyduların kütle merkezinde yerleşen, yarıçapları ise uçağa kadar olan mesafeye eşit olan üç kürenin kesişme noktası olarak bulunacaktır. Şekil 4.4'de üç uydu ile uçağın koordinatlarının bulunmasını gösteren şema görülmektedir.

Uçağın koordinatları, D_1, D_2, D_3 yarıçaplı üç kürenin (hal yüzeylerinin) kesişme noktası olarak bulunacaktır. Üç kürenin kesişmesinden iki kesişme noktası

bulunduğundan dolayı kullanılan yöntemin sonucu belli değildir. Yöntemin belirli olmasını sağlamak için dördüncü bir navigasyon uydusunun kullanılması veya doğruluğu düşük olan ek bir navigasyon yönteminin uygulanması gerekmektedir.



Şekil 4.4. Navigasyon uyduları ile cismin konumunun bulunması

NAVSTAR GPS uyduları 6 yörüngede dörder uydu yerleştirilmiş olarak toplam 24 adet uydudan oluşmaktadır. Cismin radyo görme alanında aynı zamanda en az 4 uydunun bulunması gerekir. Cismin koordinatları mesafe ölçüm yönteminin yardımı ile bulunur. Konumu karakterize eden Deskardes koordinatlarını bulmak için aşağıdaki biçimde 4 cebrik denklem çözülür.

$$(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 = (D_i - \delta_t)^2 \qquad i = 1,...,4 \qquad (4.1)$$

Burada x_i, y_i, z_i, I sayılı uydunun Deskardes koordinatlarıdır. x_c, y_c, z_c, cismin konumunun Deskardes koordinatlarıdır. D_i I sayılı uydunun cisimden olan mesafesi, δ_t cismin alıcısının zaman skalası ile sistemin ana saati arasındaki farkın oluşturduğu mesafe hatasıdır.

Böylece bulunmuş 4 nonlineer cebrik denklemden oluşan denklemler sistemi iterasyon yöntemlerinin yardımı ile bilgisayarda kolaylıkla çözülebilir. Bu durumda Newton-Rapson yönteminin uygulanması daha etkilidir. Sonuçta cismin konumunun Deskardes koordinatları x_c, y_c, z_c bir anlamlı olarak tespit edilmiş olunur.

4.4.GPS Ölçümlerini Etkileyen Hatalar

GPS ölçümlerini etkileyen hata kaynaklarını 5 sınıfta toplayabiliriz:

1- **Efemeris(önceden hesaplanmış yörünge parametreleri) verileri:** Kontrol istasyonu uydunun posizyonu ile ilgili verileri önceden hesaplar ve bu hesaplara göre uydu parametrelerini günceller. Ancak uydunun yörüngesi çeşitli etkiler sebebi ile önceden hesaplanmış yörüngeden sapar. Bundan dolayı uydu, kendisinin bulunduğu yörünge parametrelerini doğru olarak gönderemez. Efemeris hataları bu sebepten oluşur.

2- **Uydu saati:** GPS ölçümlerini önemli derecede etkileyen diğer bir unsur uydu saatinin güvenilirliğidir. GPS uyduları atom saati (sezyum ve rubidyum) kullanmaktadırlar. Bu saatlerin sapması 24 saatte 10^{-8} sn olduğu tahmin edilmektedir. Bu sapma 3.5 metrelik bir hataya neden olur. 1984 yılında yayımlanan bir raporda uyduların 24 saatlik sapması 4.1 m olduğu yazmaktadır. Buna ek olarak ABD bilerek eklediği saat hatası da vardır. Bilerek eklenen bu saat hatasına "seçici olarak eklenen hata" denir. Ancak günümüzde bu hatayı uygulamamaktadır. Uydu saati hatası C/A ve P kodlarını aynı şekilde etkiler. Bu hata diferansiyel düzeltme teknikleri ile ortadan kaldırılabilir.

3- **Atmosfer:** Atmosfer etkisinin oluşturduğu hatalar kendi içinde ikiye ayrılır.

a) İyonosfer hataları: İyonosfer tabakasındaki serbest elektronlardan dolayı, GPS sinyali bu tabakada ışık hızında hareket edemez. GPS sinyalindeki bu gecikme serbest elektronların sayısı ile doğru orantılı olarak değişir. Bu hata çeşitli tekniklerle düzeltilmektedir.

b) Troposfer hataları: Bu hatalar troposfer tabakasındaki sıcaklık, basınç ve nemden dolayı oluşan hatalardır. GPS sinyalleri bu etkilerden dolayı ışık hızında hareket edemez. Bundan dolayı oluşan gecikme troposfer hatalarını meydana getirir.

4- **Çok ışınlılık(multipath):** Bu hata GPS sinyalinin büyük yapılardan yansıyarak alıcıya gelmesi ile oluşur. Yansıyan bu sinyal, direkt olarak gelen fakat başka nedenlerden dolayı zayıflayan GPS sinyali bastırabilir. Alıcı anteninin uygun bir yere yerleştirilmesi bu ektiyi azaltabilir.

5- **Alıcı:** Alıcının teknolojisi ile orantılı olan bir hatadır. Gelişen teknoloji ile bu hata önemli seviyede düşülmüştür. [30]

Tablo 4.1. GPS Standart hata modeli[32]

Bölüm	**Hata Kaynağı**	**1 σ Hatası (m)**
Uzay Bölümü	Uydu saat hatası	3.0
	Uydu bozuntuları	1.0
	Diğer(Isıl radyasyon vb.)	0.5
Kontrol Bölümü	Efemeris hatası	4.2
	Diğer(İtki performansı vb.)	0.9
Kullanıcı Bölümü	İyonosfer gecikmesi	2.3
	Troposfer gecikmesi	2.0
	Alıcı hatası	1.5
	Çok ışınlılık(multipath)	1.2
	Diğer	0.5
Sistem	Toplam	6.6

5. KALMAN FİLTRESİ İLE GPS SİSTEMİNİN BİRLEŞTİRİLMESİ

Havacılıkta navigasyon ve kontrol problemlerinin çözümü günümüzde çok çeşitli yöntemler kullanılarak gerçekleştirilmektedir. Bu yöntemler arasında farklılıklar bulunmasına rağmen temelde aynı amacı gerçekleştirmek için çalılışmaktadır. Bu amaç doğruluğun maksimum seviyede arttırılması ve güvenilirliğn sağlanmasıdır. Ne var ki navigasyon için kullanılan her yöntem yeterli doğruluğu ve güvevilirliği sağlayamamaktadır. Buradan hareketle talep olunan şartları yerine getirmek için birleştirilmiş yöntemlerin kullanılması gerektiği ortaya çıkmaktadır.

Uçuşun en zor ve en riskli aşaması olan inişte, özelikle yüksekliğin cok doğru bir şekilde bilinmesi gerekmektedir. Aksi taktirde kazayı meydana getiren durumlar ortaya çıkacaktır ki havacılıkta meydana gelen kazaların çoğunluğu da zaten inem rejiminde meydana gelmektedir.

Navigasyon problemlerinde GPS yüksek bir doğruluk sağlamasına rağmen navigasyon sisteminin temel elemenı olan Ataletse Navigasyon Sistemi (INS)'den vazgeçmek mümkün olmayacaktır. Günümüzde yapılan çalışmalarda GPS, diğer navigasyon sistemlerine destekçi olarak diğer navigasyon sistemlerinin doğruluğunu arttırmak için kullanılmaktadır. Böylece navigasyon probleminde yüksek doğruluk sağlanmış olur.

Bu çalışmanın amacı GPS uydularından alınan gürültülü konum ve hız ölçümlerini ilk önce optimall Kalman Filtresi ile birleştirerek bozuntuların belli bir dereceye kadar düşürülmesidir. Daha sonra Kalman Filtresinin hata modelini yapay sinir ağına eğitip Kalman Filtresinin hatasını en düşük seviyeye düşürmektir.

5.1. Uyduların Hareket Simulasyonu

Uydular tarafından bir cismin konumunun tespit edilmesi için ilk önce uyduların konumları bilinmelidir. Bunun için Kepler kanununa göre uyduların hareket denklemleri çıkartılır. M, dünyanın kütlesi, γ, Kepler sabiti olmak üzere uydunun üç boyuttaki hareketini gösteren aşağıdakiş denklemleri yazabiliriz.

$$\begin{aligned} \frac{d^2x}{dt^2} &= \gamma M \frac{x}{r^3} \\ \frac{d^2y}{dt^2} &= \gamma M \frac{y}{r^3} \\ \frac{d^2z}{dt^2} &= \gamma M \frac{z}{r^3} \end{aligned} \qquad (5.1)$$

Bu denklemlerden uydunun x, y, z yönündeki hareketi bulunur. Her bir uydunun koordinat merkezine uzaklığı, r_i, hesaplanır. Bu koordinat sisteminin başlangıcı dünyanın merkezidir.

$$r_i = \sqrt{x_{ui}^2 + y_{ui}^2 + z_{ui}^2} \qquad i = \overline{1,4} \qquad (5.2)$$

Uyduların yörüngede durabilmeleri için hızlarının (5.3) denklemini sağlaması gerekmektedir.

$$V_x^2 + V_y^2 + V_z^2 = 7.9^2\, km/sn \qquad (5.3)$$

Simülasyonda kullanılan ilk değerler aşağıdaki tabloda verilmiştir.

Tablo 5.1. Uydunun simülasyonunda kullanılan başlangıç değerleri

	1. uydu	2. uydu	3. uydu	4. uydu
x	$10\times10^6 m$	$-22.5\times10^6 m$	$-21\times10^6 m$	$15\times10^6 m$
y	$20\times10^6 m$	$22\times10^6 m$	$-18\times10^6 m$	$-10\times10^6 m$
z	$23\times10^6 m$	$15\times10^6 m$	$20\times10^6 m$	$18\times10^6 m$
V_x	$5000 m/sn$	$4000 m/sn$	$4000 m/sn$	$4000 m/sn$
V_y	$5000 m/sn$	$5000 m/sn$	$4500 m/sn$	$5000 m/sn$
V_z	$3522.8 m/sn$	$4627.1 m/sn$	$5114 m/sn$	$4627.1 m/sn$

Dünyanın kütlesi, $M = 5.976\times10^{24} kg$

Kepler sabiti, $\gamma = 6.67\times10^{-11} m^3/kgs^2$

Uyduların konumları bulunduktan sonra (4.1) denkleminin yardımı her bir uydunun uçağa olan uzaklığı ve zaman kayması hesaplanabilir.

5.2. Uçuş Simulasyonu

Uyduların hareketi tanımlandıktan sonra, bu uydular tarafından konumu belirlenecek cismin (uçağın) hareket simülasyonu için dinamik denklem (5.4) denkleminde verilmiştir.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du \qquad (5.4)$$

Burada x durum değişkenlerini içeren durum vektör matrisi, A matrisi durum transfer matrisi, B ise kontrol geçiş matrisi ve C gözlem denklemi transfer matrisi ve D'de gözlem denklemi kontrol geçiş matrisi ve u'da kontrol giriş vektörüdür.dir. Ancak simülasyonda bu denklemleri ayrık formda yazacağız.

$$X_{t+1} = AX_t + Bu_t$$
$$Y_t = CX_t + Du_t \tag{5.5}$$

$$A = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, B = 0, H = 0$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.6}$$

Bu denklemle uçağın x, y, z koordinatlarındaki hızları ve pozisyonları elde edilir. Simülasyonda kullanılan uçağın ilk hızları;

$$V_x = 246\, m/sn$$
$$V_y = 43 m/sn \tag{5.7}$$
$$V_z = 0 m/sn$$

Uçağın deniz seviyesinden 20km yukarıda uçtuğu düşünülmektedir.Uçağın ilk konumları ise;

$$x = 0m$$
$$y = 0m \tag{5.8}$$
$$z = 6386000\ m$$

Bu değerler uçağın gerçekte sahip olduğu değerlerdir. Ancak GPS sistemiyle uçağın koordinatlarını bulurken önceki bölümde ifade edilen sebeplerden dolayı ölçüm değerlerinede sıfır ortalamalı beyaz gauss bozuntuları olmaktadır. Bu ölçüm değerlerindeki bozuntuları ortadan kaldırmanın en optimal yolu ise Kalman Filtresini kullanmaktır.

5.3. GPS ve Kalman Filtresinin Birleştirilmesi

Bir sisteminin ölçüm değerlerini Kalman Fitresi ile birleştirilmesi için ilk şart o sistemin matematik modelini Kalman Filtresine sunmaktır. Sistemin modeli (5.9) ifadesinde gösterildiği gibidir.

Kalman Filtresi ölçümleri filtrelerken sistemin kendi gürültüsünden oldukça etkilenir. Eğer sistem gürültüsü az ise Kalman Filtresi iyi sonuçlar vermektedir. Ancak sistem gürültüsü artarsa Kalman Filtresinin çıkışları istenilen şekilde olmayacaktır. Uçak için bunu düşündüğümüzde eğer durgun bir havada uçuyorsa sistem gürültüsü azdır. Rüzgarlı bir havada uçuş gerçekleştiriliyorsa sistem gürültüsü büyük olacaktır.

$$X_{t+1} = \Phi_t X_t + Gw \tag{5.9}$$

Sistemin ölçme modeli ;

$$Y_t = HX_t + v_t \tag{5.10}$$

Burada Φ matrisi, durum geçiş matrisi olup A matrisi ile aynıdır ve aşağıda gösterildiği gibidir.

$$\Phi_t = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.11}$$

Δt, örnekleme zamanıdır ve ölçülen dataların alınma zamanı ile aynı olduğu kabul edilmektedir. X_t, durum vektörüdür. Uçağın hız ve pozisyon bilgilerini içermektedir.

$$X_t = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix}^T \tag{5.12}$$

G, sistem bozuntusunun geçiş matrisidir. Aşagıda gösterildiği gibidir.

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.13}$$

w, sistemin bozuntu vektörüdür.

$$w = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T \tag{5.14}$$

Sistemin bozuntusunun, sıfır ortalamalı beyaz Gauss gürültüsü olduğu kabul edilmektedir. Sistem bozuntusunun korelasyon matrisi Q'dur. Eğer sistem bozuntusu büyükse Q'da büyüktür. Eğer sistem bozuntusu küçükse Q'da küçüktür. Bu çalışmada Q'nun çeşitli değerleri için simülasyon tekrarlanmıştır. (5.13)'de ifade edildiği gibidir.

$$Q_n = \begin{bmatrix} q_n & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.15}$$

Sistem bozuntusunun korelasyon matrisi olarak, (5.15) ifadesinde q_n sabitinin sırasıyla 0.001, 0.01, 0.1, 1 alınarak Kalman Filtresi'nin çeşitli sonuçları elde edilmiştir.

Y_k, sistemin gözlem matrisidir. Uçağın ölçümleri yapılan durumlarını içermektedir ve (5.16) ifadesinde gösterildiği gibidir.

$$Y_t = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{5.16}$$

v_t, ölçüm bozuntusudur ve sıfır ortalamalı ve beyaz Gauss gürültüsüdür. Ölçüm bozuntusunun korelasyon matrisi R'dir ve aşağıda gösterilmiştir.

$$R = \begin{bmatrix} \sigma_{\Delta x}^2 & 0 & 0 \\ 0 & \sigma_{\Delta y}^2 & 0 \\ 0 & 0 & \sigma_{\Delta z}^2 \end{bmatrix} \tag{5.17}$$

Burada $\sigma_{\Delta x}^2$, $\sigma_{\Delta y}^2$, $\sigma_{\Delta z}^2$ GPS sonuçlarının hata varyanslarıdır.

Ölçüm matrisi , H, gözlem matrisi ile aynı olup (5.18) ifadesinde gösterilmiştir.

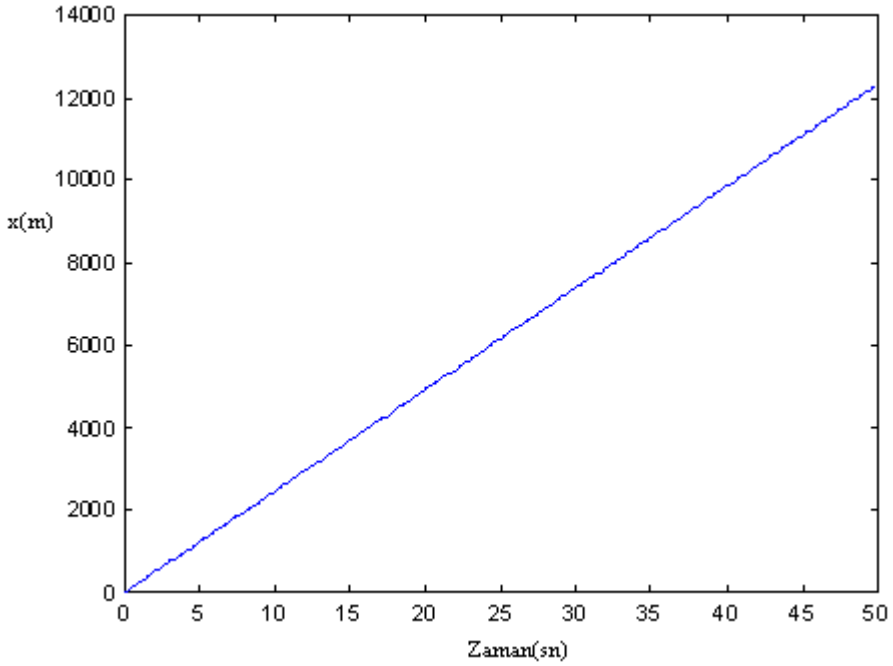
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.18}$$

Başlangıç koşulları olarak P(0) değerinin yeteri kadar büyük seçilmesi gerekir. P değeri büyük seçilmesi durumunda süreç içinde uygun değere doğru bir azalma eğrisi gösterecektir. Ancak P değeri yeteri kadar büyük seçilmezse hata kovaryans matrisi değeri belli bir değere oturmayacak ve Kalman Filtresi hatalı sonuç verecektir.

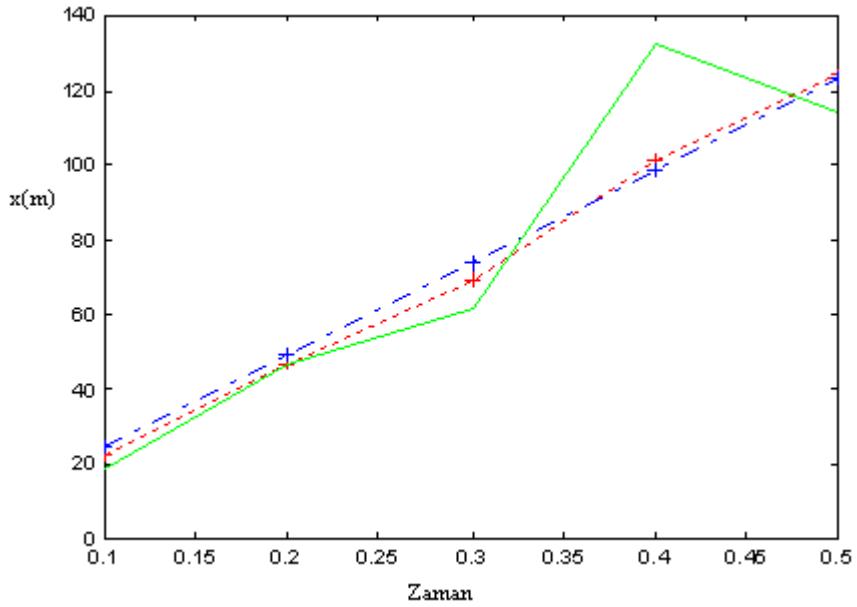
$$P(0) = diag([120 \quad 120 \quad 120])$$
$$\hat{x}(0) = [0;0;6386000\ ]^T \tag{5.19}$$

5.4. Simülasyon Sonuçları

Simülasyon, Matlab ortamında yazılmış programlar yardımı ile oluşturulmuştur. Yapılan bu programlara göre uçağın, uyduların ve Kalman Filtresinin sonuçları aşağıda verilmiştir.



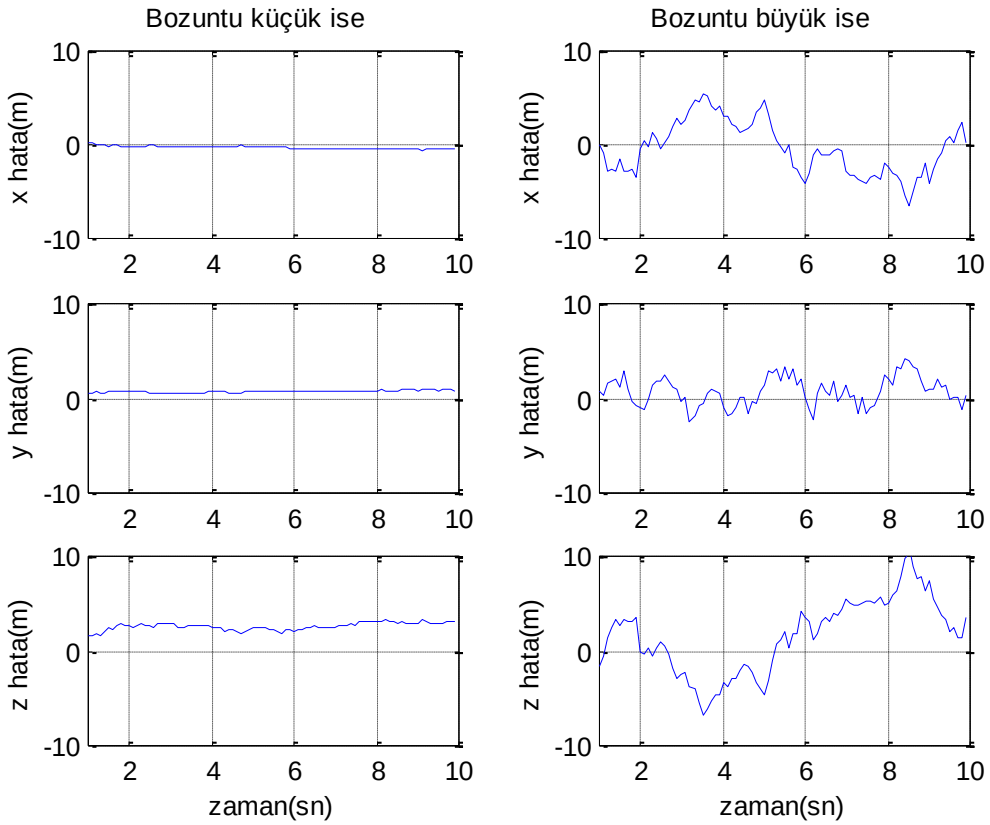
Şekil 5.1. Uçağın x ekseni yönündeki hareketi



Şekil 5.2. Uçağın x yönündeki gerçek, GPS, Kalman Fitresi sonuçları

Şekil 5.2'de uçağın x yönündeki hareketi ve beraberinde GPS ölçüm sonuçları ve Kalman Filtresi sonuçları daha iyi anlaşılması için küçük bir aralıkta görülmektedir. Burada Düz çizgi GPS ölçüm sonuçunu, küçük kesikli çizgi Kalman Filtresi sonucunu, büyük kesikli çizgi ise uçağın gerçek durumunu göstermektedir.

Daha önceden de söylendiği gibi sistem bozuntusu Kalman Filtresi'nin sonuçlarını önemli derecede etkilemektedir. Şekil 5.3'de bu etkinin sonuçları açık bir şekilde görülebilmektedir. Uçak sakin havada uçuyorsa sistem bozuntusu az olacak ve Kalman Filtresi ideal bir çıkış verecek ve gerçek duruma yakın sonuçlar verecektir. Eğer sistem bozuntusu fazla ise bu durumda gerçek durum ile Kalman Filtresi arasındaki fark büyüyecektir.



Şekil 5.3. Uçağın sistem bozuntusuna bağlı olarak KF çıkışları

6. Kalman Filtresinin YSA ile Birleştirilmesi

6.1. YSA Destekli Kalman Filtresi Yapısının Kurulması

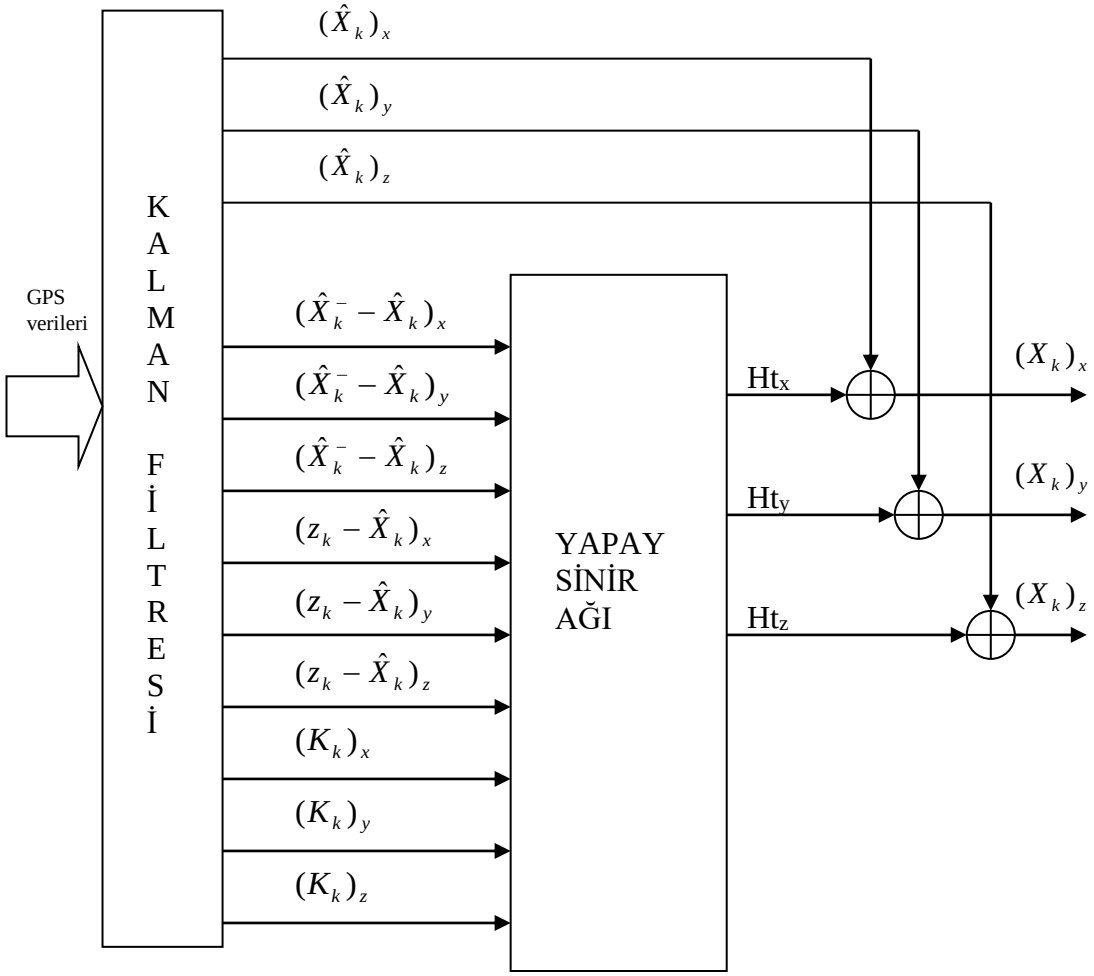
Bu bölümde Kalman Filtresi yapay sinir ağı ile desteklenmesi anlatılacaktır. Kalman hatalarının YSA ile modellenmesi ile daha iyi bir hedef izleme olanağı sağlanmaktadır. Kalman Filtresinin YSA ile birlikte çalışması şekil 1.' de gösterildiği gibidir.

Bu çalışmada YSA yapısı olarak çok katmanlı perceptronlar (ÇKP) kullanılmıştır. ÇKP'lar oldukça basit yapıları sebebiyle yaygın olarak kullanılan yapay sinir ağlarıdır. Eğitim algoritması olarak ise yine popüler bir algoritma olan geriye yayılım algoritması kullanılmıştır

Yapay sinir ağının uygulaması 3 aşamadan oluşmaktadır. 1. aşama, yapının kurulması, 2. aşama, eğitim ve test, 3. aşama geri çağırma aşamasıdır.

YSA'nın giriş ve çıkışı şöyledir: giriş vektörü olarak 9 giriş sinyali, çıkış vektörü olarak 3 çıkış sinyalinden oluşmaktadır. İlk üç giriş x, y, z yönündeki öncül kestirim değerleri ile filtrelenmiş durum vektörü arasındaki fark $(\hat{X}_k^- - \hat{X}_k)_x$, $(\hat{X}_k^- - \hat{X}_k)_y$, $(\hat{X}_k^- - \hat{X}_k)_z$ olarak alınmıştır. İkinci üç giriş, x, y, z yönündeki ölçüm

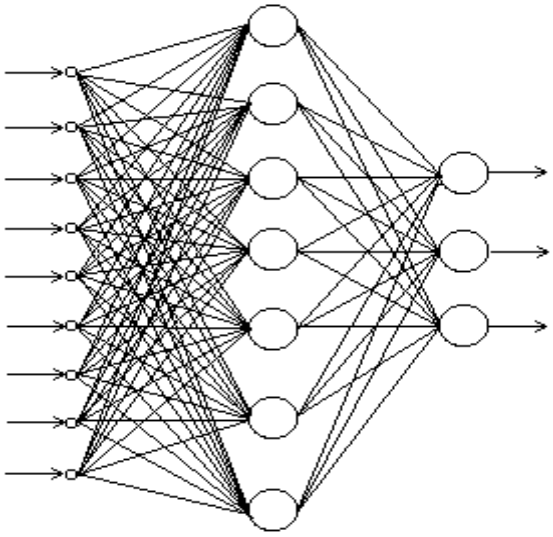
değerleri ile filtrelenmiş durum vektörü arasındaki fark $(z_k - \hat{X}_k)_x$, $(z_k - \hat{X}_k)_y$, $(z_k - \hat{X}_k)_z$ olarak alınmıştır. Son üç giriş ise x, y, z, yönündeki Kalman kazancı $(K_k)_x$, $(K_k)_y$, $(K_k)_z$ olarak alınmıştır. Çıkış sinyali olarak x, y, z yönündeki uçağın simülasyon değerleri ile filtrelenmiş durum vektörü arasındaki fark $Ht_x = (X - \hat{X}_k)_x$, $Ht_y = (X - \hat{X}_k)_y$, $Ht_z = (X - \hat{X}_k)_z$ olarak alınmıştır.



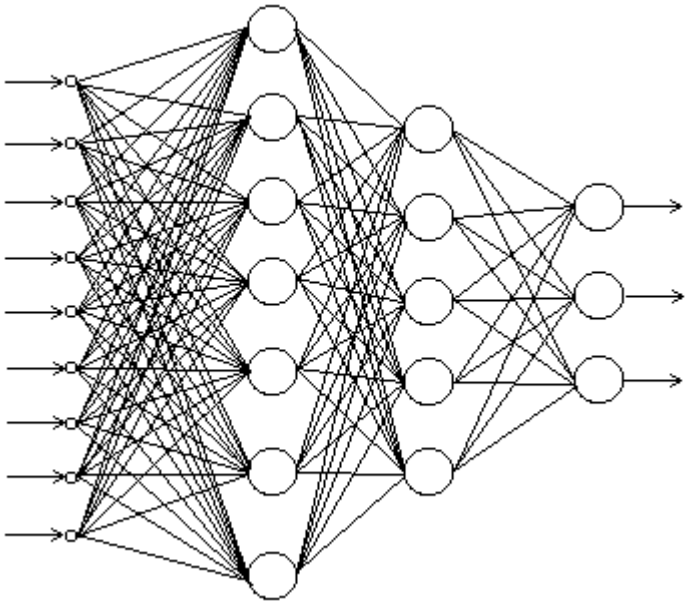
Şekil 6.1. Kalman Filtresinin YSA ile birlikte çalışması

Yapay sinir ağında nonlineerliği sağlayan gizli katmandır. Teoride tek bir gizli katman nonlineerliği saylamaktadır. Ancak gizli katman sayısı yapay sinir ağının performansını önemli derecede etkilemektedir. Gizli katmanla ilgili temel bilgilere göre çok az nöron kullanılması eğitimin zayıf olmasına sebep olabilir. Nöron sayısının çok fazla olması ile çok iyi bir eğitim yanında interpolasyon ve extrapolasyon özelliklerinde negatif etkiye sebep olabilir.

Performans değişimi incelemek için gizli katma sayısı 1, 2, 3, olan ayrı üç yapı kurulmuştur. Tek gizli katmanlı yapıda 7 tane sinir hücresi kullanıldı. İki gizli katmanlı yapıda, ilk gizli katmanda 7 ikincisinde 5 tane sinir hücresi kullanılmıştır. Şekil 6.2 a,b'de bu yapılar gösterilmiştir.



a) Tek gizli katmanlı YSA (7 Nöron)



b) Çift gizli katmanlı YSA (7-5 Nöron)

Şekil 6.2. Kullanılan YSA yapıları.

Aktivasyon fonksiyonu olarak sigmodial fonksiyon kullanılmıştır. YSA'daki nöronların öğrenme katsayısı ve momentun katsayısı iyi bir sonuca ulaştıracak değerlerde seçilmelidir

$$Y_D = \begin{bmatrix} (\hat{X}_k^- - \hat{X}_k)_x \\ (\hat{X}_k^- - \hat{X}_k)_y \\ (\hat{X}_k^- - \hat{X}_k)_z \\ (z_k - \hat{X}_k)_x \\ (z_k - \hat{X}_k)_y \\ (z_k - \hat{X}_k)_z \\ (K_k)_x \\ (K_k)_y \\ (K_k)_z \end{bmatrix} \qquad U_D = \begin{bmatrix} (X - \hat{X}_k)_x \\ (X - \hat{X}_k)_y \\ (X - \hat{X}_k)_z \end{bmatrix} \tag{6.1}$$

Her örnek tek bir giriş sinyali vektörü ve buna karşılık gelen çıkış sinyali vektörü içerir. Bu çalışmada giriş sinyali olarak aşağıda belirtilen Y_D vektörü, çıkış sinyali olarak ise U_D vektörü kullanılmıştır.

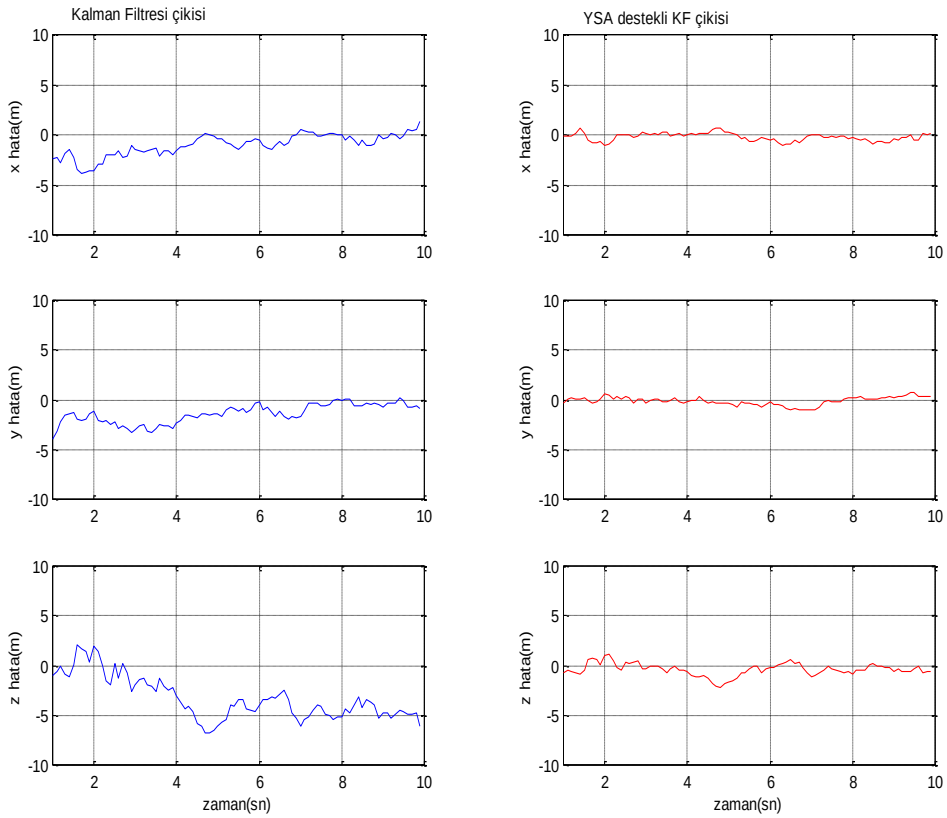
YSA'da ağırlıkların hesaplanması standart delta kuralına göre yapılmıştır. Delta kuralına göre;

$$\Delta W_{ji}^l = \eta \rho_j^l(n) u_i^{l-1}(n) + \alpha \Delta W_{ji}^l(n-1) \tag{6.2}$$

Burada η öğrenme katsayısı, α momentum katsayısı, $\Delta W_{ji}^l(n)$ n anında l. katmandaki i. nöronu $(l-1)$. katmandaki j. nörona bağlayan ağırlığın değişimi, $\rho(n)$ l. katmandaki j. nöronun n anındaki çıkış değeri ile hedef değer arasındaki fark , $u(n)$ n anında $(l-1)$. katmanda i. nöronun çıkış değeridir.

6.2. Simülasyon Sonuçları

Yapay sinir ağının eğitim aşamasında, Kalman Filtresinin 100 iterasyonluk çıkışı, normalizasyon işleminden sonra sinir ağına giriş örneği olarak verilir. Sinir ağının ağırlıkları ilk olarak rastgele olarak seçilir. Uçağın gerçek pozisyonu simülasyondan elde edilmiş olduğundan, geriye yayılım algoritması öğretmen ile öğrenme modeli kullanılmıştır.. Sinir ağının ağırlıkları Delta öğrenme kuralı kullanılarak ayarlanmıştır. Her iterasyon sonunda hataların karesi hesaplanır. Eğer hata limitine ulaşılmadıysa ağırlıklar değiştirilir. Ağırlıkların değiştirilmesi hata limitine ulaşıncaya kadar devam edilir. Ağırlıkların değiştirilmesi işlemi bitince geri çağırma aşaması gelir.

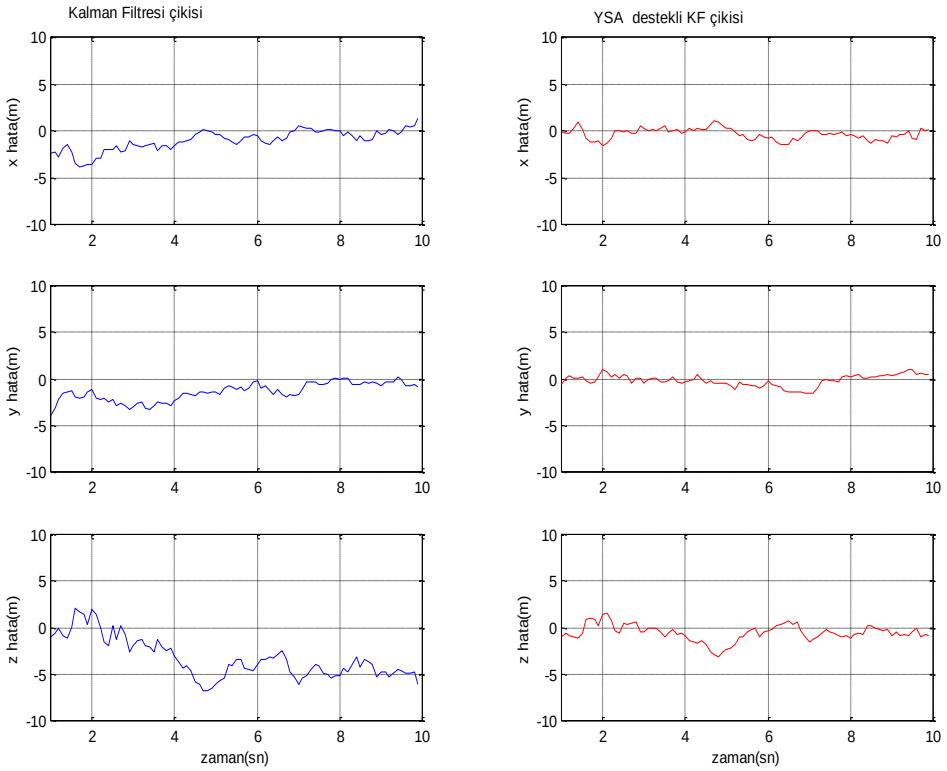


Şekil 6.3. KF ve YSA destekli KF çıkışı

Geri çağırma aşamasında Kalman Filtresi gerçek değerleri yapay sinir ağı girişine sunar ve YSA çıkışı Kalman Filtresinin hata modeline en yakın çıkışı üretir. Yapay sinir ağının bu çıkışları Kalman Filtresinin filtrelenmiş durum vektörü

elemanları ile toplanarak uçağın derçek pozisyonuna en yakın iyileştirilmiş bir değer elde edilir.

Şekil6.3'te Kalman Filtresi ile YSA destekli Kalman Filtresinin çıkışları karşılaştırılmıştır. Sistem bozuntusunun arttığı durumda Kalman Filtresi çıkışının hatası büyümekte şekil 6.3 te görüldüğü gibi 10m'ye varan sapmalar gösterebilmektedir. Sistem bozuntusu arttığında açıktırki bu hatalar dahada büyüyecektir. Ancak Kalman Filtresinin bu hata modelini iyi bir eğitimle YSA'ya öğretebilirsek bu hatanın Kalman Filtresinin çıkışındaki bu hatayı ortadan kaldırabiliriz. Şekil 6.3 teknik olanaklar dahilinde yapılan iyi bir eğitim ile hedef izleme sonuçları geçek değere daha da yaklaştırılmıştır.



Şekil 6.4. Tek gizli kat. YSA'nın Kalman Filtresini düzeltme sonucu

Şekil 6.4'te tek katmanlı YSA kullanılarak yapılan Kalman hatasını düzeltme grafiği görülmektedir.Tek katmanlı YSA'nın öğrenme kabiliyeti çift katmanlıya göre daha azdır. Şekil 6.3'te çift katmanlı bir YSA'nın Kalman Filtresi'nin hatasını düzeltme performansı görülmektedir. Bu iki grafikten

anlaşılmaktadır ki katman sayısı arttıkça YSA'nın öğrenme kapasitesi de artmaktadır.

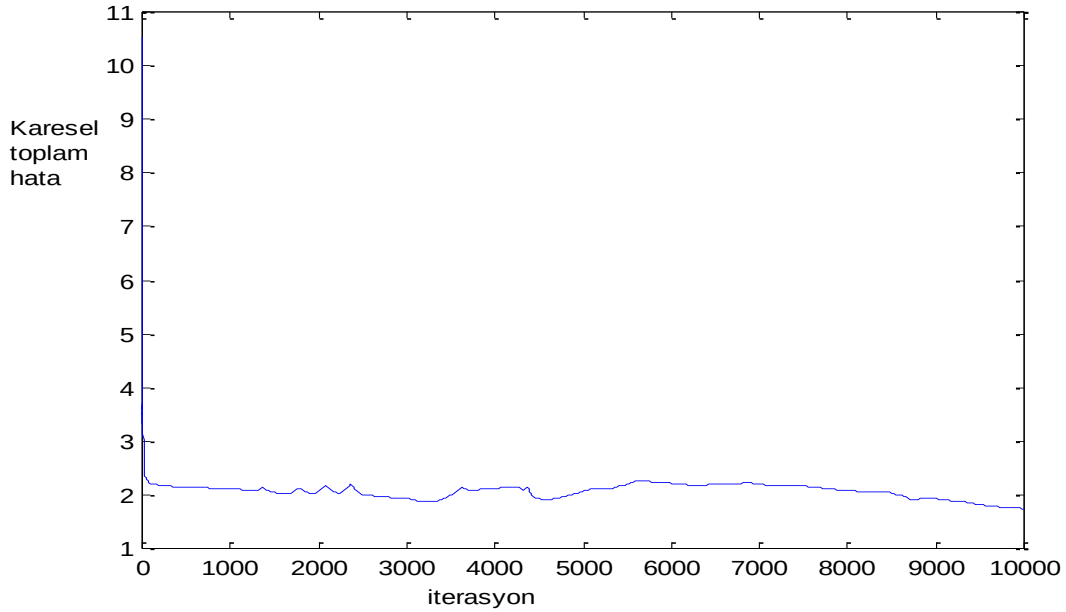
Tablo 6.1'de YSA'nın Kalman Filtresi'nin hatasını ne kadar oranda küçülttüğü görülmektedir. Ayrıca tek gizli katmanlı YSA ile çift gizli katmanlı YSA arasındaki fark da görülmektedir. Açık bir şekilde görülmektedir ki YSA, Kalman Filtresindeki büyük hataları sönümlemektedir. Ancak küçük hatalarda bu sönümleme işini iyi yapamamaktadır. Bunun sebebi teknik olanaksızlıktan dolayı iyi bir eğitim yapılamamamış olmasıdır.

Tablo 6.1. YSA'nın Kalman Filtresini düzeltme oranı

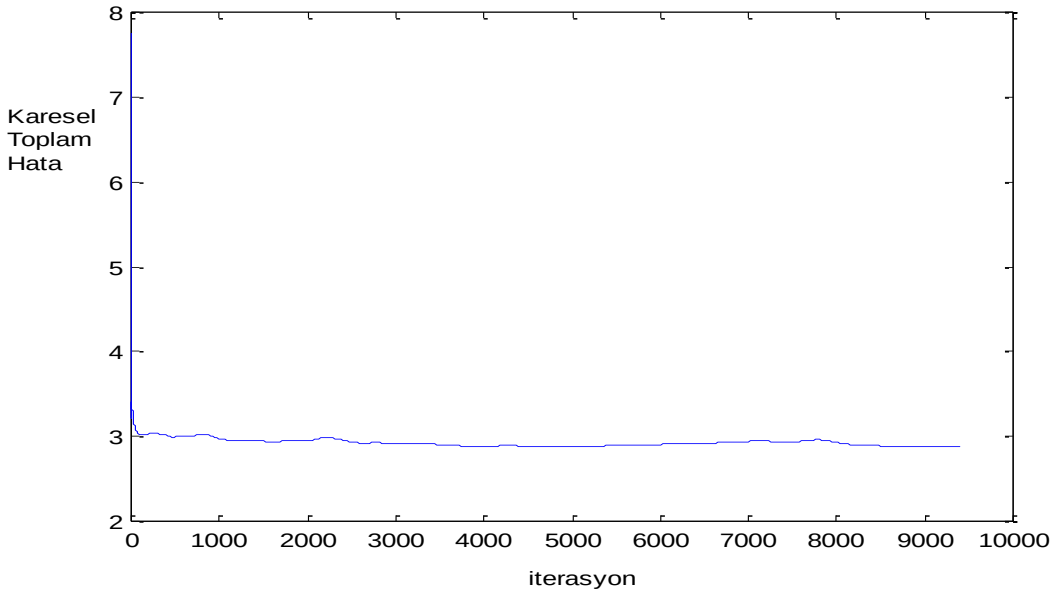
Zaman	Kalman Filtresi			YSA tek gizli kat. yapı			YSA çift gizli kat. yapı		
	x	y	z	x	y	z	x	y	z
2	-0.20	-4.82	0.06	-0.70	-0.57	-2.73	-0.57	-0.47	-1.74
5	-6.34	-2.90	3.23	-2.28	0.42	-0.08	-1.87	0.34	-0.05
10	-2.99	-4.25	-1.26	-0.13	-0.43	-2.9	-0.1	-0.35	-1.85
20	-3.58	-1.39	0.28	-1.26	0.18	0.29	-1.03	0.15	1.84
30	-1.03	-3.22	-2.61	0.61	0.14	-0.71	0.5	0.11	-0.45
40	-2.05	-2.83	-2.16	-0.28	-0.61	-0.94	-0.23	-0.5	-0.59
50	-0.15	-1.38	-6.45	0.6	-0.51	-4.08	0.48	-0.42	-2.6
80	-0.06	-0.09	-5.13	-0.67	0.42	-1.38	-0.55	0.34	-0.87
100	1.39	-0.86	-6.13	0.2	0.63	-1.37	0.16	0.52	-0.86
Düzeltme Oranı(%)				43.2	62.5	58.5	53.5	69.3	73.6

Şekil 6.5'te sistemde kullanılan iki gizli katmana sahip YSA'nın eğitim performansını gösteren grafik görülmektedir. Karesel hata ne kadar az olursa eğitim o kadar iyi olmuştur denilebilir. İdeal olarak karesel hata toplamı başlangıç değerinin 0.1 katı olmalıdır. Şekil 6.6.'da ise tek gizli katmana sahip YSA'nın karesel hata grafiği görülmektedir. Bu grafikte tek gizli katmana sahip YSA diğerine daha uzun eğitilmiş olmasına rağmen onun kadar iyi bir grafiğe sahip değildir.

Teorik olarak gizli katmanın görevi giriş ile çıkış arasındaki nonlineerliği sağlamaktır. Normalde bu görevi yerine getirmek için tek gizli katman yeterlidir. Ancak yapılan deneyler sonucu tek gizli katmanın yeterli performansı sağlayamadığı görülmüştür. Bu nedenle daha iyi bir sonuç elde edebilmek için gizli katman sayısını arttırmak gerekir.



Şekil 6.5. İki gizli katmana sahip YSA'nın karesel hata grafiği



Şekil 6.6. Tek gizli katmana sahip YSA'nın karesel toplam hata grafiği

7. SONUÇ

1980'li yıllarda çok katmanlı yapay sinir ağlarının öğrenme algoritmalarının ortaya çıkması bu konuya olan ilginin çok hızlı bir şekilde artmasına sebep olmuştur. Özellikle çok popüler olan geriye yayılım algoritması bir çok mühendislik alanında etkin bir şekilde kullanılmaktadır. Yapay sinir ağlarının nonlineer yapıları, öğrenme yetenekleri, genelleme yapabilmeleri onları güçlü kılan unsurlardır. Bu özellikleri sayesinde YSA'ları, sinyal işleme, sistem tanıma, kontrol gibi alanlarda, problemlerin çözümünde çok büyük kolaylık sağlamaktadır.

Kalman Filtresi 1960'da ortaya atılmasından sonra birçok dinamik sistemde başarı ile uygulanmıştır. Çok boyutlu sistemlerde kolay uygulanabilir olması, bilgisayarda algoritmasının basit olarak yazılabilmesi gibi özelliklerinden dolayı dinamik sistemlerde oldukça kullanım alanı bulmuştur. Ancak Kalman Filtresini kısıtlayan bir özellik ise lineer olmayan sistemlerde veya sistem bozuntusu büyük olan sistemlerde doğruluğunun önemli derecede düşmesidir. Bu durumda bu bozuntulu sönümleyecek ek bir sisteme ihtiyaç vardır. Bu problemin çözümüne YSA'nın nonlineer yapısını kullanacak bir bakış açısından bakabiliriz.

Bu çalışmada Kalman Filtresinin bozuntulu ortamda çalışması durumunda bozuntusunun YSA'ların öğrenme kabiliyeti ile onlara öğretilebileceği ve YSA'nın öğrendiği bu hatayı Kalman Filtresinin sonucunu düzeltmek için kullanılabileceği gösterildi.

Her ne kadar bir YSA'nın performansını karşılaştıracak kriterler daha geliştirilmesede yapılan deneyler ile YSA'ların performans karşılaştırılması yapılabilir. Gizli katman sayısı, başlangıçta random olarak atanan ağırlıklar, öğrenme ve momentum katsayısı YSA'nın performansını etkileyebilecek faktörlerdir.

Bu çalışmada YSA'nın öğrenme kabiliyetini karşılaştırmak için aldığım kriter gizli katman sayısıdır. Gizli katman sayısı tek ve çift olarak aldığım YSA'ları aynı eğitim kümeleri ile eğittim. Ağırlıkların başlangıç değeri random olarak alındığı için başlangıç değerlerinden kaynaklanan bir fark olmasın diye bu eğitimi her iki YSA'ya birkaç kez uyguladım. Sonuçlar şunu göstermiştir ki, tek gizli katmana sahip YSA'nın öğrenme performansı belli bir değerde sabit kalıyor ve bir artış gözlenmiyordu. Ancak çift gizli katmana sahip YSA daha iyi bir eğitim performansı göstermiştir.

YSA'nın öğrenme performansını arttıracak bir husus daha fazla veri ile daha uzun bir eğitim yaptırmaktır. Bu çalışmada kullandığı yazılım Matlab olduğu için, hızını diğer dillere göre yavaş olması ve ek olarak çalıştığım bilgisayarın hızınında yavaş olması YSA'ya fazla veri ve uzun bir eğitim süreci sonuçu iyi bir eğitim sonuçuna ulaşmayı engellemiştir. C veya Delphi programlama dillerinde yazılacak olan bir program ile daha çok veri daha kısa sürede YSA'na sunulabilir.

KAYNAKLAR

[1] **ZURADA, J., M**., Introduction to Artifical Neural Systems, West Publishing Company, 1992

[2] **KOHONEN, T.,** An Intoduction to Neural Computing, Neural Networks, Vol. 1, pp 3-16,1988

[3] **McCULLOCH W. S., PITTS W. A.,** A Logical Calculus of the Ideas Immanent in Nervous Activity, Bull. Math. Biophysics, Vol 9, pp 127-147, 1947

[4**] HEBB, D. O**. The Organization of Behavior: A Neuropsychological Theory, NewYork:Wiley, pp 62, 1949

[5] **ROSENBLATT, F**., The Perceptron: A probabilistic model for information storage and organization in the brain, Phsychological Review, 65, pp 386-408, 1958

[6] **MINSKY, M., PAPERT, S**., Perceptrons, MIT Press, 1969

[7] **HOPFİELD, J.,** Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Academy of Sci., 79, pp 2554-2558, 1982

[8] **HOPFİELD, J.,** Neurons with graded response have collective computational abilities like those of two state neurons, Proc. Nat. Academy of Sci., 81, pp 3088-3092, 1984

[9] **HOPFİELD, J.,** Neurons with graded response have collective computational abilities like those of two state neurons, Proc. Nat. Academy of Sci., 81, pp 3088-3092, 1984

[10] **RUMELHART, D.E., McCLELLAND, J.L**., Paralel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, 1986

[11] **RİEDMİLLER, M.,** Advanced Supervised Learning in Multilayer Perceptrons From Backpropagation to Adaptive Learning Algoritms, Jornal of Computer Standars and Interfaces, 5, Special issue on Neural Networks, 1994

[12] **SİMPSON, P. K.,** Artifical Neural Systems, Pergamon Press, 1990

[13] .**GÜZELİŞ, C.,** Genelleştirilmiş Hücresel Yapay Sinir Ağları, Elektrik Mühendisliği 5. Kongresi, Trabzon, pp 7-12, 1993

[14] . **NARENDRA, K., S., ve PARTHASARATHY, K.,** Identification and Control Dynamic Systems Usind Neural Network, IEEE Trans. Neural Networks, vol. 1., pp 4-27,1990

[15] **HAYKIN., S.,** Neural Networks, Prentice Hall, New Jersey, 1999

[16] **SORENSON, H. W.,** Least-squares Estimation: From Gauss to Kalman, IEEE Spectrum, Vol. 7, pp 63-68,1970

[17] **KALMAN, R. E.,** A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering,Vol. 82D, pp35-45, 1960

[18] **GREWAL, M.S., ANDEWS A.P.,** Kalman Filtering Theory and Practice, Printice Hall, Englewood Cliffs, New Jersey, 1993

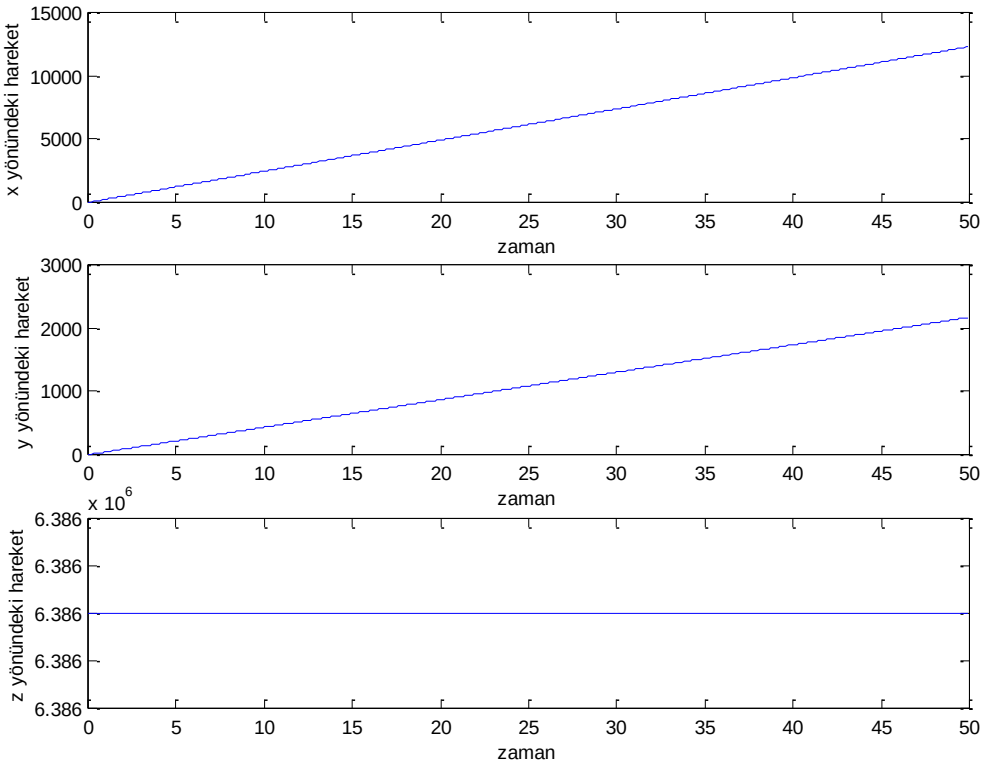
[19] **SUN, P., MARKO, K**., Optimal learning rate for training time lagged recurrent neural networks with the extended Kalman filter algoritm, IEEE International Conference on Neural Networks-Conference Proceeding, Vol. 2, pp 1287-1292,1998

[20] **BOAGHE, O.M**., Theoretical and pactical considerations over neural network trained

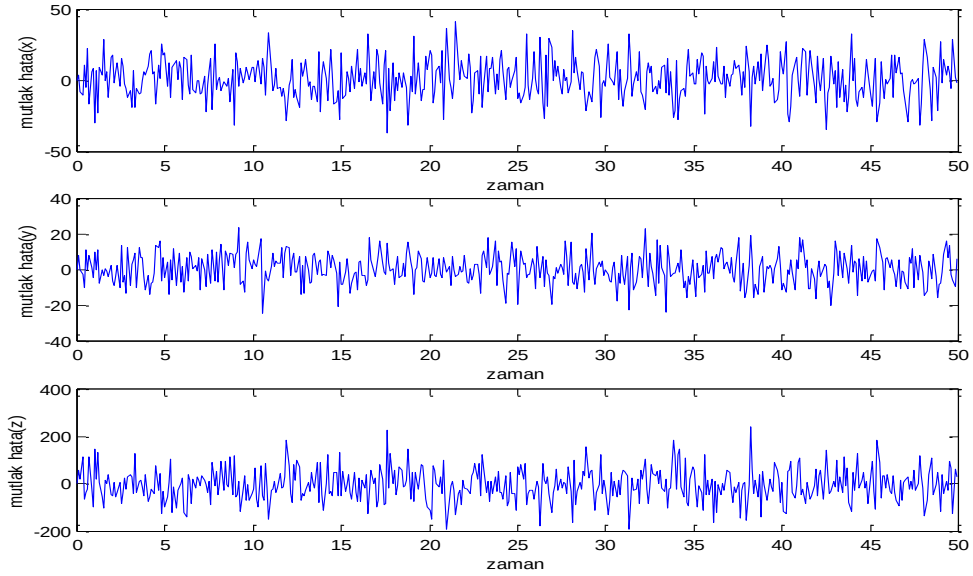
with Kalman Filter, Neural Network World, Vol. 6, pp 507-513, 1996
[21] **HORTON, M. P.,** Real-time identification of missile aerodynamics using a linearized Kalman filter aided an artifical neural network, IEE Proceedings: Control Theory and Applications, Vol. 144, pp 299-308, 1997
[22] **GAO. F., WANG, F.M.,** Neural network based nonlinear controller using an extended Kalman filter, Industrial and Engineering Chemistry Research,Vol. 38, pp 2345-2349, 1999
[23] **NAPOLİTANO, M. R. WINDON D. A., CASANOVA, J.L.I., MARIOSİL V., G.,** IEEE Transaction on Control Systems Technology, Vol. 6, pp 596-611, 1998
[24] **VAIDEHI, V., CHİTRA, N. C., M. KRİSHNAN, C. N.,** Neural Network aided Kalman Filtering for multitarget tracking applications, Computers and Electrical Engineering, Vol. 27, pp 217-228, 2001
[25] **İNCE, C.D.,** Dinamik sistemlerin GPS ve Kalman Filtresi ile anlık olarak izlenmesi, Doktora Tezi, İTÜ Fen Bilimleri Enstitüsü, 1999
[26] **WELCH, G., BISHOP, G**. An Introduction to Kalman Filter, http://www.cs.unc.edu/~welch, 2001
[27] **HACIYEV, Ç.,** 1999. Radyo navigasyon, İstanbul : İTÜ
[28] **ZHANG F., KHAWAJA B., MAHMOOD F., YUEZU VE SUN, XİANFANG,** 2000. GPS/INU/DM integre konum ve navigasyon teknolojileri akıllı taşıma sistemlerinde kullanımının uygulanması. Beijing Hangkong Hangtian Daxue Xuebao/Journal of Beijing University of Aeronautics and Astronautics c **26** n 3 Jun 2000. s 299-302.

[29] **COLLINSON, R. P. G.,** Introduction to Avionics,Chapman & Hall, 1996
[30] **PARKİNSON. B. W.,** Global Positioning System: Theory and Applications Vol1, American Institute of Aeronautics and Astronautics Inc.,1996
[31] **COSKUN, M. Z.,** GPS' in Teorik Esasları ve Türkiye'deki Uygulama Alanları, 1990
[32] **KAPLAN, D. E.,**Understanding GPS Principles and Applications, Artech House,1996
[33] **WASSERMAN, P. D, SCHWARTZ, T,** Neural Networks-Part 1,IEEE Expert, Vol. 2, No 4,pp 10-12, Winter 1987
[34] **WASSERMAN, P. D, SCHWARTZ, T,** Neural Networks-Part 2,IEEE Expert, Vol. 3, No 1,pp 10-15, Spring 1988

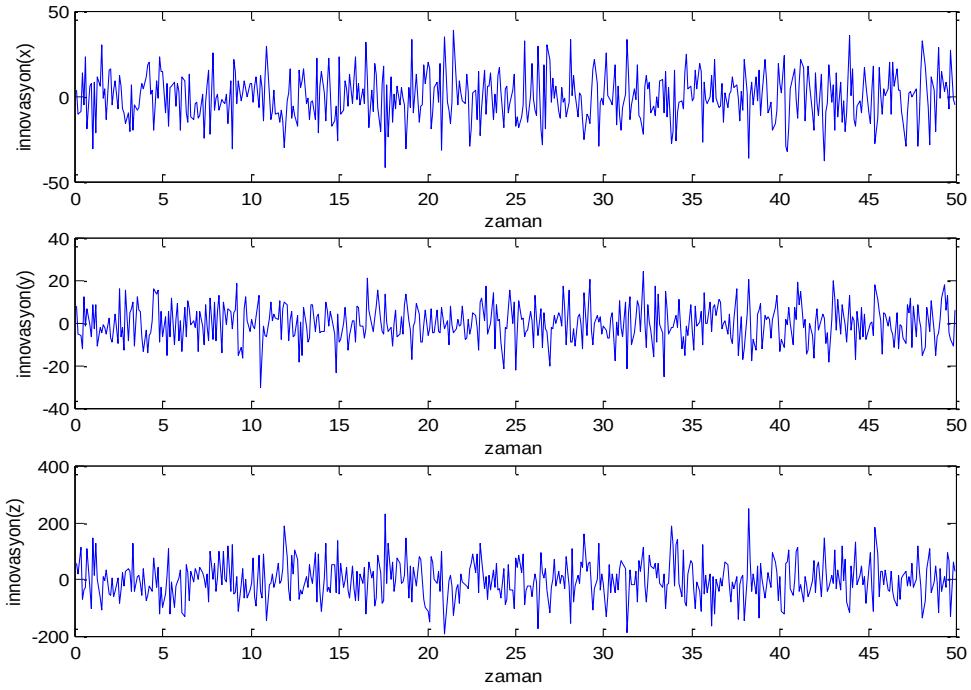
EK-1



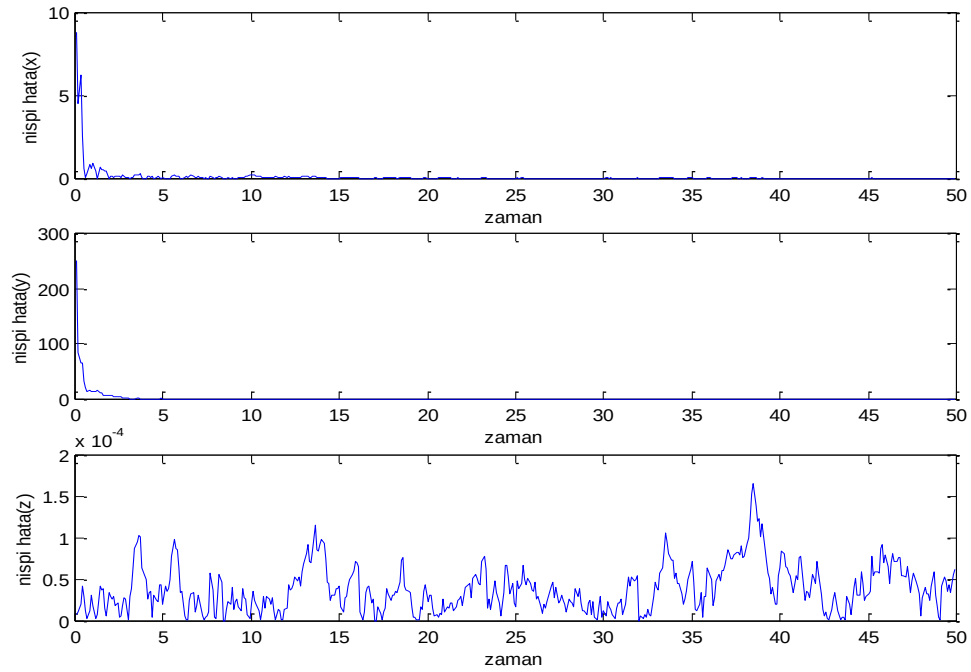
Şekil E.1. Uçağın simülasyonu sonucu x,y,z yönünde hareket grafiği



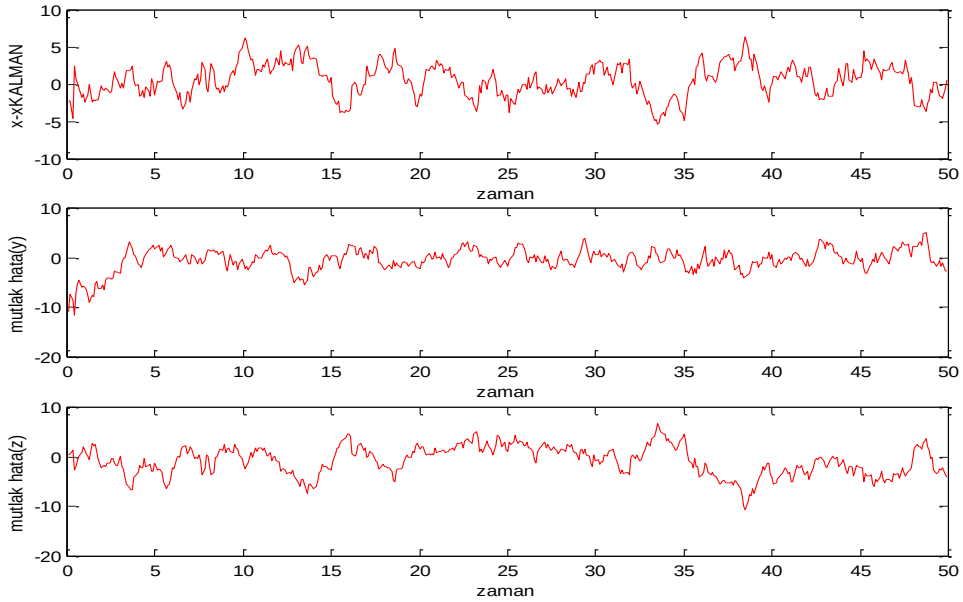
Şekil E.2. GPS sonuçlarının mutlak hatasını gösteren grafik



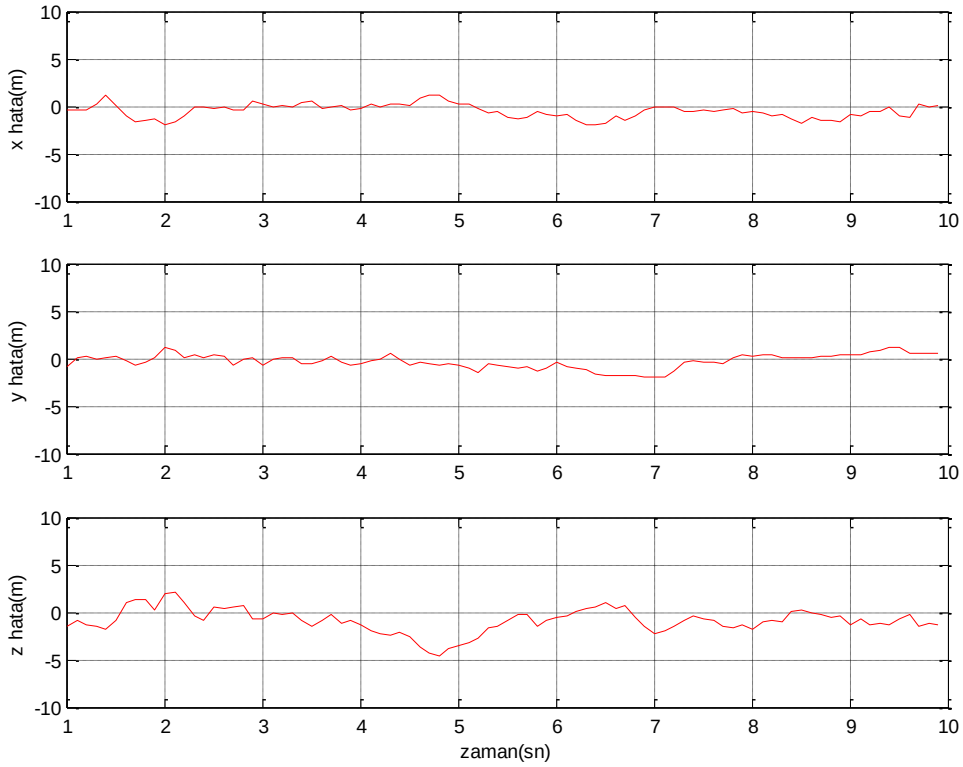
Şekil E.3. Kalman Filtresi'nin innovasyon süreç grafiği



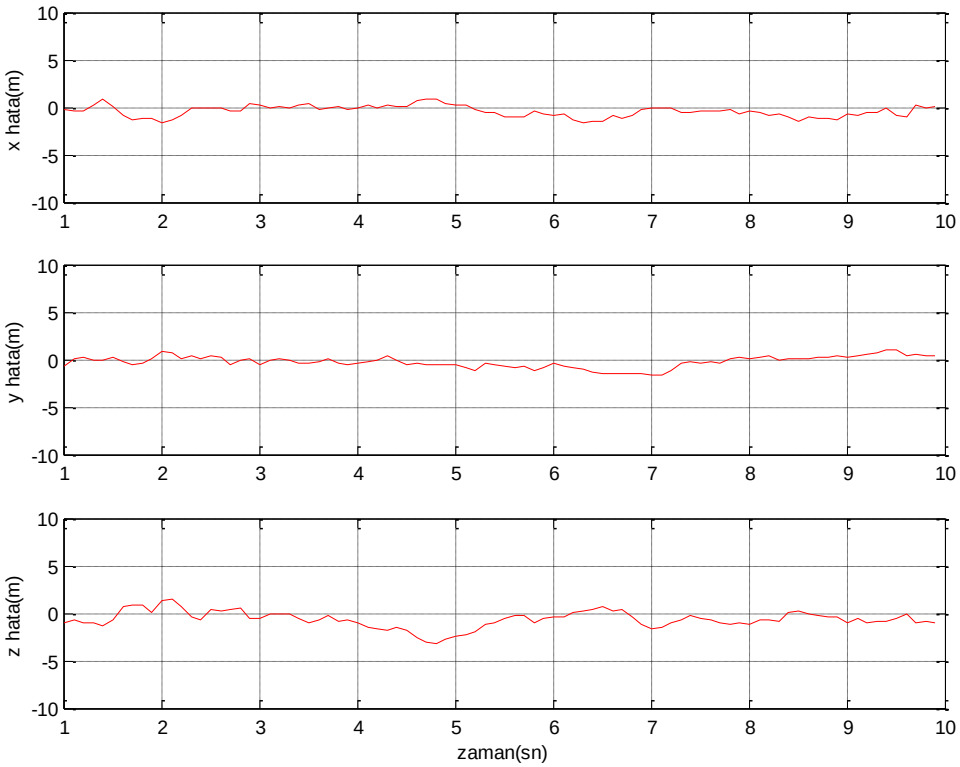
Şekil E.4. Kalman Filtresi'nin nispi hata grafiği



Şekil E.5. Kalman Filtresi'nin mutlak hata grafiği



Şekil E.6. Tek katmanlı YSA'nın mutlak hata grafiği



Şekil E.7. Çift katmanlı YSA'nın mutlak hata grafiği

EK-2

```
clear;
load ysaegt;
yd(:,1)=ysa(:,1)/10;%giriþler
yd(:,2)=ysa(:,2)/5;%giriþler
yd(:,3)=ysa(:,3)/5;%giriþler
yd(:,4)=ysa(:,4)/100;%giriþler
yd(:,5)=ysa(:,5)/50;%giriþler
yd(:,6)=ysa(:,6)/500;%giriþler
yd(:,7)=ysa(:,7)/0.5;%giriþler
yd(:,8)=ysa(:,8)/1;%giriþler
yd(:,9)=ysa(:,9)/0.03;%giriþler

ud(:,1)=hata(:,1)/10;%çýkýþlar
ud(:,2)=hata(:,2)/10;%çýkýþlar
ud(:,3)=hata(:,3)/20;%çýkýþlar


xp=yd;
d=ud;
orsay=200;
norsay=[9;7;5;3];

W1=randn(norsay(2),norsay(1)+1);WW1=W1;DW1=zeros(norsay(2),norsay(1)+1);
W2=randn(norsay(3),norsay(2)+1);WW2=W2;DW2=zeros(norsay(3),norsay(2)+1);
W3=randn(norsay(4),norsay(3)+1);WW3=W3;DW3=zeros(norsay(4),norsay(3)+1);


Emax=0.0001;E=0;ET=0;eta=0.1;alfa=0.9;
h = waitbar(0,'Program çalýþýyor.Ýþlem için bekleyin...');
tic
for l=1:35000

   for i=1:orsay
      x=xp(i,:)';
      x=[x;1];
      net1=W1*x;
      clear y1;clear y1tur;

      for j=1:norsay(2)
         y1(j,1)=(2/(1+exp(-net1(j))))-1;
         y1tur(j,1)=2*exp(-net1(j))/(1+exp(-net1(j)))^2;
      end;
      y1=[y1;1];
      y1tur=[y1tur;0];

      net2=W2*y1;
      clear y2;clear y2tur;
      for j=1:norsay(3)
         y2(j,1)=(2/(1+exp(-net2(j))))-1;
         y2tur(j,1)=2*exp(-net2(j))/(1+exp(-net2(j)))^2;
      end;
      y2=[y2;1];
      y2tur=[y2tur;0];


      net3=W3*y2;
      clear y3;clear y3tur;
      for j=1:norsay(4)
         y3(j,1)=(2/(1+exp(-net3(j))))-1;
         y3tur(j,1)=2*exp(-net3(j))/(1+exp(-net3(j)))^2;
      end;

      e=d(i,:)'-y3(:,1);
      for j=1:norsay(4)
         E=0.5*e(j)^2+E;
      end
```

```
        delta3=e.*y3tur(:,1);

        for j=1:norsay(3)+1
            delta2(j)=(W3(:,j)'*delta3(:))*y2tur(j);
        end

         for j=1:norsay(2)+1
            delta1(j)=(W2(:,j)'*delta2(1:norsay(3))')*y1tur(j);
        end


        for j=1:norsay(4)
           for k=1:norsay(3)+1
              W3(j,k)=WW3(j,k)+eta*delta3(j)*y2(k)+alfa*DW3(j,k);
              DW3(j,k)=W3(j,k)-WW3(j,k);
              WW3(j,k)=W3(j,k);
           end
        end

        for j=1:norsay(3)
           for k=1:norsay(2)+1
              W2(j,k)=WW2(j,k)+eta*delta2(j)*y1(k)+alfa*DW2(j,k);
              DW2(j,k)=W2(j,k)-WW2(j,k);
              WW2(j,k)=W2(j,k);
           end
        end

        for j=1:norsay(2)
           for k=1:norsay(1)+1
              W1(j,k)=WW1(j,k)+eta*delta1(j)*x(k)+alfa*DW1(j,k);
              DW1(j,k)=W1(j,k)-WW1(j,k);
              WW1(j,k)=W1(j,k);
           end
        end

    end


    ee(l)=E;

    if mod(l,1)==0
        [l ee(l) ]
    end
       if E>=Emax
       E=0;
       ET=0;
    else
       break
    end
   save egtm055 W1 W2 W3 ee l eta alfa orsay norsay
   waitbar(l/35000)

end
toc
close(h)
msgbox('iþlem tamam')
```