İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BİR SUALTI ARACININ ATALETSEL ÖZELLİKLERİNİN ALGILANMASI YOLUYLA YÖRÜNGE TAKİBİ

YÜKSEK LİSANS TEZİ

Müh. Mesut Ufuk ALTUNKAYA

Anabilim Dalı: DİSİPLİNLERARASI

Programı: MEKATRONİK MÜHENDİSLİĞİ

AĞUSTOS 2006

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BİR SUALTI ARACININ ATALETSEL ÖZELLİKLERİNİN ALGILANMASI YOLUYLA YÖRÜNGE TAKİBİ

YÜKSEK LİSANS TEZİ

Müh. Mesut Ufuk ALTUNKAYA

518031018

Tezin Enstitüye Verildiği Tarih : 16 Ağustos 2006

Tezin Savunulduğu Tarih : 16 Ağustos 2006

Tez Danışmanı : Y.Doç.Dr. Serhat İKİZOĞLU (İ.T.Ü.)

Jüri Üyeleri : Y.Doç.Dr. Levent OVACIK (İ.T.Ü.)

Prof.Dr. Serhat ŞEKER (İ.T.Ü.)

AĞUSTOS 2006

ÖNSÖZ

Bu çalışmada bir sualtı aracının yörüngesinin takibi amacıyla kullanılması düşünülen ataletsel algılayıcıların verilerinin işlenerek gerçeğe en yakın konum bilgilerine ulaşılmaya çalışılmıştır.

Başta, çalışmam sırasında mümkün olan her türlü yardımı bana sunan, verdiği bilgi ve önerilerle bana yol gösteren danışman hocam Sn. Serhat İkizoğlu olmak üzere, çalışmanın temelini oluşturan ataletsel algılayıcıyı bana temin edip maddi ve manevi destekte bulunan tüm Baykar Makina San. ekibine, sualtı aracının yapımını gerçekleştiren Sn. Ercan Akdemir ve Sn. Ahmet Karagöz'e, yardımlarını esirgemeyen diğer iki değerli hocam Sn. Fikret Gürgen ve Sn. Demir Sindel'e, ayrıca Enstrümantasyon Laboratuvarı'ndaki asistan arkadaşlarla, İTÜ-GİDB Fakültesi'ndeki değerli büyüklerime ve bu süreç içinde bana sürekli destek olan ailem ve tüm arkadaşlarıma teşekkürlerimi sunarım.

Ağustos, 2006 Mesut Ufuk Altunkaya

İÇİNDEKİLER

KISALTMALAR v
TABLO LİSTESİ vı
ŞEKİL LİSTESİ vıı
SEMBOL LİSTESİ ıx
ÖZET x
SUMMARY xv

1. GİRİŞ 1

2. ATALETSEL ALGILAYICILAR 3
2.1. Ataletsel Algılayıcı Hataları 3
2.1.1. Statik Hata 3
2.1.2. Ölçek Katsayısı 4
2.1.3. Eksen Kaçıklığı 4
2.1.4. Gürültü 5
2.2. Hata Giderme Yöntemleri 5
2.2.1. Kestirim 5
2.2.2. Fark Alma 6
2.2.3. Ölçekleme 6
2.3. Hatanın Modellenmesi 6
2.4. Projede Kullanılan Algılayıcı 7
2.5. Algılayıcıdan İvme Verilerinin Okunması ve Değerlendirilmesi 8

3. ALGILAYICI VERİLERİNİN FİLTRELENMESİ 13
3.1. Temel Filtreleme Yöntemleri 13
3.1.1. Kayan Ortalama 13
3.1.2. Eğri Atama 14
3.2. LMS Algoritması 16
3.2.1. LMS Filtre Uygulaması 18
3.2.2. RLS Filtre Uygulaması 21
3.3. Yapay Sinir Ağları ile Filtreleme 25
3.3.1. Geri Yayılım Algoritmalı MLP Sinir Ağları 26
3.3.2. Elman Sinir Ağı 28
3.3.3. Radyal Temel Fonksiyonlu Sinir Ağları 29
3.3.4. Farklı Aşamalarda Filtreleme 32
3.3.5. Parçalı Yapay Sinir Ağları ile Filtreleme 34
3.4. LMS ile YSA Sonuçlarının Karşılaştırılması 36

4. KONUM TAKİP SİMÜLASYONU 38
4.1. Üç Eksende Hareket Takibi 38

4.1.1. Deniz Araçlarının Hareketleri 38
4.1.2. Eksen Takımları 39
4.2. Simulink Arayüzü ile Oluşturulan Program 41

5. SUALTI ARACI 43
5.1. İnsansız Sualtı Araçları 43
5.2. Gerçeklenen Sualtı Aracı 47
5.2.1. Aracın Yapısı 47
5.2.2. Elektronik Sistem ve Araç Kumandası 49

6. SONUÇLAR VE TARTIŞMA 53
6.1. Yapılan Çalışmanın Sonuçları 53
6.2. Geleceğe Dönük Öneriler 54

KAYNAKLAR 55

ÖZGEÇMİŞ 58

KISALTMALAR

GPS : Global Positioning System
INS : Inertial Navigation System
LMS : Least Mean Square
FIR : Finite Impulse Response
RLS : Recursive Least Square
YSA : Yapay Sinir Ağı
IMU : Inertial Measurement Unit
MLP : Multi Layer Perceptron
GYA : Geri Yayılım Algoritması
GD : Gradient Descent
GDM : Gradient Descent with Momentum Backpropagation
GDX : Gradient Descent with Adaptive Learning Rate Backpropagation
RBP : Levenberg-Marquardt Backpropagation
RBP : Resilient Back Propagation
RTF : Radyal Temel (Radial Basis) Fonksiyonu
RBE : Exact Radial Basis Net
GRNN : Generalized Regression Neural Net
PYSA : Parçalı Yapay Sinir Ağları
RBP : Resilient Back Propagation
UUV : Unmanned Underwater Vehicle
ROV : Remotely Operated Vehicle
AUV : Autonomous Underwater Vehicle
PWM : Pulse Width Modulation

TABLO LİSTESİ

Sayfa No

Tablo 2.1 : Hata modelleme yöntemlerinin karşılaştırılması……. 7

ŞEKİL LİSTESİ

			Sayfa No
Şekil 2.1	**:**	Microstrain 3DM-G Algılayıcı…….....	8
Şekil 2.2	**:**	Algılayıcı parametreleri…….....	8
Şekil 2.3	**:**	Uygulanan yöntemi akış diyagramı…….....	9
Şekil 2.4	**:**	İşlenmemiş ivme verileri…….....	10
Şekil 2.5	**:**	Algılayıcı tarafından işlenmiş ivme verileri… ……..	10
Şekil 2.6	**:**	Sabit algılayıcı için ivme, hız ve konum değişimi verileri…….	11
Şekil 2.7	**:**	Hareketli algılayıcı için ivme hız ve konum değişimi verileri...	11
Şekil 3.1	**:**	İvme bilgilerinden hareketle konum belirlenmesine ilişkin Simulink şeması.....…….	13
Şekil 3.2	**:**	Kayan ortalaması alınmış gürültülü ivme verisi….……	14
Şekil 3.3	**:**	10'uncu dereceden eğri atama sonucu.....……	14
Şekil 3.4	**:**	Gürültülü sinyale spline eğrisi atanması...……	15
Şekil 3.5	**:**	Trapez sinyale eğri atanması……..	15
Şekil 3.6	**:**	Adaptif (uyarlamalı) filtreleme işlemi...…….	16
Şekil 3.7	**:**	LMS filtre modeli.....……...	18
Şekil 3.8	**:**	LMS blok parametreleri.....…….	19
Şekil 3.9	**:**	Gerçek sinyal (koyu renk) (m/s^2) ile filtre çıkışı (açık renk).....	20
Şekil 3.10	**:**	LMS hata verisi……..	20
Şekil 3.11	**:**	Gerçek konum değişimi (metre) ve LMS'in hatası.....……...	21
Şekil 3.12	**:**	RLS filtreleme modeli.....……....	23
Şekil 3.13	**:**	RLS filtre çıkış hatası…….	23
Şekil 3.14	**:**	Gerçek konum bilgisi ve RLS'in hatası……..	24
Şekil 3.15	**:**	LMS ve RLS'in karşılaştırılması...…….	24
Şekil 3.16	**:**	Gerçek (gürültüsüz) ivme verisi...…….	26
Şekil 3.17	**:**	GYA'lı MLP mimarisi.....…….	26
Şekil 3.18	**:**	Farklı etkinlik fonksiyonları..…….	27
Şekil 3.19	**:**	Resilient Backpropagation ile eğitim sonucu...…….	28
Şekil 3.20	**:**	Elman ağının mimarisi…….	28
Şekil 3.21	**:**	Elman ile MLP'nin karşılaştırılması (m/s^2)……...…….	29
Şekil 3.22	**:**	RTF Algoritması.....……	29
Şekil 3.23	**:**	Generalized Regression algoritmasıyla yapılan eğitim sonucu..	30
Şekil 3.24	**:**	Simulinkte hazırlanan YSA filtreleme modeli....…….	31
Şekil 3.25	**:**	Elman ile RTF'nin ivme çıkış hatalarının karşılaştırılması...….	31
Şekil 3.26	**:**	Gerçek konum değişimi verisi.....…….	31
Şekil 3.27	**:**	Elman ile RTF'nin konum bilgilerinin karşılaştırılması.....….	32
Şekil 3.28	**:**	Gürültülü ve filtrelenmiş hız verilerinin hataları.....….	33

Şekil 3.29	**:**	Konum bilgisi olarak filtrelenmiş ve gürültülü sinyallerin hataları.....	33
Şekil 3.30	**:**	Gürültülü ve filtrelenmiş konum verilerinin hataları.....….	33
Şekil 3.31	**:**	İvme, hız ve konum bilgilerinin filtrelenmesiyle elde edilen konum verilerinin hataları.….....	34
Şekil 3.32	**:**	Parçalı filtrelemede RTF ağları arası anahtarlama düzeneği..…	35
Şekil 3.33	**:**	Parçalı RTF ağının (koyu renk veri filtre çıkışı) ve gürültülü sinyalin konum değişimi hataları.....	35
Şekil 3.34	**:**	LMS ile YSA'nın ivme verilerinin gerçek ivme verisinden farkları.....	36
Şekil 3.35	**:**	LMS ile YSA'nın konum verilerinin hatları.....….	36
Şekil 4.1	**:**	Sualtı aracının yaptığı doğrusal ve açısal hareketler.....	38
Şekil 4.2	**:**	Eksenler etrafında dönüşler....……	39
Şekil 4.3	**:**	Üç eksenli Simulink takip modeli..…….....……	41
Şekil 4.4	**:**	Animasyonlu konum takibi.....…….……..	42
Şekil 5.1	**:**	Kontrol yüzeylerinin etkisi.....……..	44
Şekil 5.2	**:**	Ticari bir ROV modeli....……....	44
Şekil 5.3	**:**	Bir AUV örneği.....……...	46
Şekil 5.4	**:**	a) Robot Yengeç, b) Robot Balık, c) Robot Yunus....……..	46
Şekil 5.5	**:**	Tasarlanan sualtı aracı.....……	48
Şekil 5.6	**:**	Sualtı aracının bitmiş hali .....……..	49
Şekil 5.7	**:**	Kullanılan PWM motor sürücü devresi.....	50
Şekil 5.8	**:**	Alıcıdan çıkan PWM sinyali.....	51
Şekil 5.9	**:**	Alıcıdan çıkan PWM sinyali ile motor sürme.....	52

SEMBOL LİSTESİ

g	:	Yerçekimi ivmesi
f	:	İvme
b_f	:	Statik hata
e	:	gürültü
S	:	Ölçek katsayısı
μ	:	Adım büyüklük parametresi
$e^2(n)$	:	Ortalama kare hata
R	:	Korelasyon matrisi
λ_{max}	:	R korelasyon matrisinin aldığı en büyük özdeğerdir (eigenvalue)
$g(n)$	:	uyarlılık katsayısı
u	:	Öteleme hızı
v	:	Sürüklenme hızı
w	:	Dalıp-çıkma hızı
p	:	Yalpa hızı
q	:	Baş-kıç vurma hızı
r	:	Savrulma hızı
Φ	:	Yalpa açısı
θ	:	Baş-kıç vurma açısı
ψ	:	Savrulma açısı

BİR SUALTI ARACININ ATALETSEL ÖZELLİKLERİNİN ALGILANMASI YOLUYLA YÖRÜNGE TAKİBİ

ÖZET

Sualtında üç eksende hareket edecek bir aracın yörünge takibinin yapılmasının planlandığı bu çalışmada üç eksenli ivmeölçer yardımıyla aracın X, Y, Z eksenlerindeki ivmelerinin belirlenip bu değerlerin integrasyonu ile aracın bu yönlerdeki hızlarının, elde edilen değerlerin ikinci kez integrasyonuyla da aracın konumunda meydana gelen değişimlerin incelenmesi öngörülmüştür.

Bu amaçla öncelikle projede kullanılmasına karar verilen üç eksenli ataletsel algılayıcı ile eksenel ivme ölçümleri yapılmıştır. Yerçekimi ivmesine göre iyi ayarlanmış bir ivme ölçerin matematiksel modeli kısaca şu şekilde tanımlanabilir:

$$I_f = f + b_f + e \qquad \textbf{(1)}$$

Burada f, b_f, e sırasıyla ivme, statik hata ve gürültüyü göstermektedir.

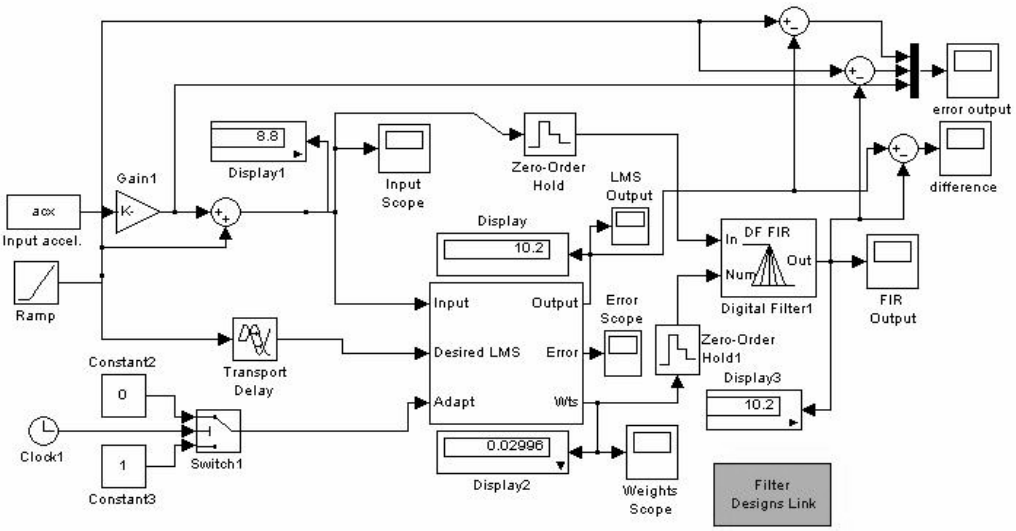
Gerçek ivme değerlerinin elde edilebilmesi için eldeki verilerin filtrelenmesi gerekmektedir. Uygulanan ilk yöntemler olan kayan ortalama ve eğri atama sonucu istenilen verilere ulaşılamamıştır. Ayrıca kullanılan algılayıcıdan alınan veriler sayısal olduğu için (okuma yapıldığı andaki değer 16 bit'lik veri olarak verilmektedir) herhangi bir frekans bilgisi içermemektedir. Bu nedenle frekans alanında işlem yapılması imkansızdır ve alçak geçiren filtre gibi klasik filtreler kullanılamaz. Bu tür durumlarda pratikte uyarlamalı (adaptif) algoritmalara sahip filtrelerden faydalanılmaktadır.

Sistemin, gerçeklenmesi durumunda farklı araçlar için kullanılabilir olması istenildiğinden filtreleme işleminin de model bağımsız olmasına karar verilmiştir. Bu nedenle Kalman Filtresi gibi stokastik yöntemler bu çalışmada uygulanmamıştır. İlk olarak En Küçük Ortalama Karesi (LMS- Least Mean Square) yöntemi denenmiştir. LMS algoritması en dik inme (steepest decent) yöntemini kullanan uyarlamalı bir filtredir. LMS, en küçük ortalama kare hatayı sağlayacak şekilde ağırlık vektörüne başarılı düzeltmeler uygular. En dik inme yönteminden *w(n)* ağırlık vektörü şu şekil formüle edilir:

$$w(n+1) = w(n) + \frac{1}{2}\mu[-\nabla(E\{e^2(n)\})] \qquad \textbf{(2)}$$

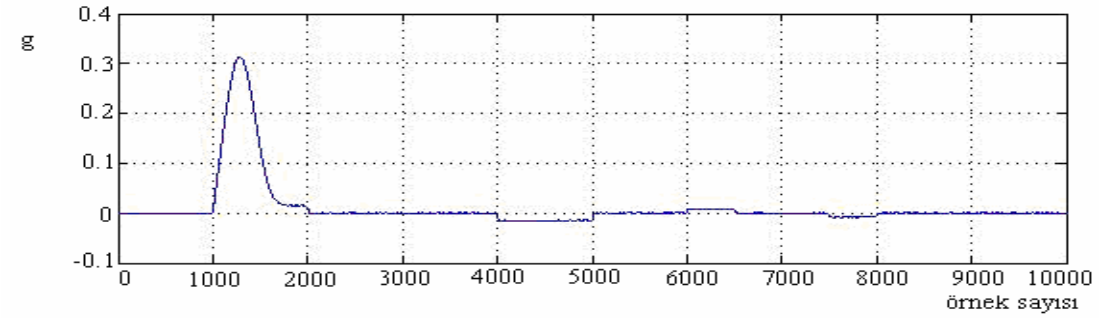
Burada μ adım büyüklük parametresidir ve LMS algoritmasının yakınsama karakteristiğini kontrol eder. $e^2(n)$ ise $y(n)$ çıkışı ile referans sinyal arasındaki ortalama kare hatadır. LMS algoritması n=0 da keyfi atanmış bir $w(0)$ değeri ile başlatılır. Ağırlık vektörünün başarılı bir şekilde düzeltilmesi sonuçta en küçük ortalama kare hatanın elde edilmesini sağlar.

LMS algoritmasını kullanarak Simulink ortamında hazırlanan filtreleme modeli Şekil 1'de gösterilmiştir.



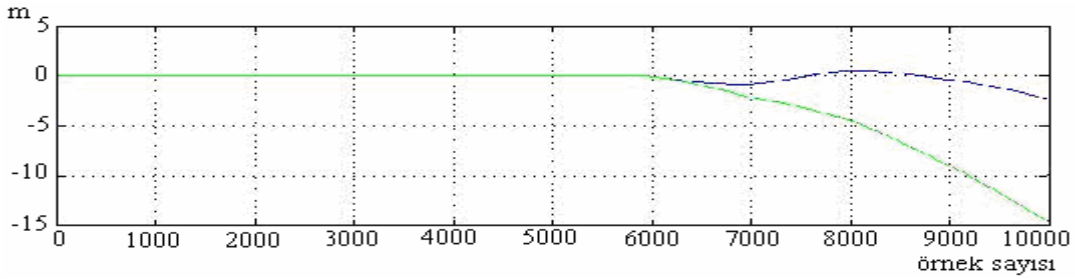
Şekil 1 : LMS filtre modeli

Burada algılayıcı çıkışından elde edilen gürültülü sinyal bozucu olarak kullanılmış ve hazırladığımız bilinen, "gerçek" sinyale eklenerek filtre girişine verilmiştir. Diğer yandan "gerçek" sinyal filtreye istenen sinyal olarak girilmektedir. Simulink'in LMS filtre bloğu, temel olarak giriş ve istenen sinyalleri karşılatırarak sürekli ağırlık değişimi yapmakta ve istenen sinyale yakınsamaktadır. LMS'in hatası Şekil 2'de verilmiştir. Burada ilk 2000 değerde görülen büyük hata filtrenin ağırlıklarının yeni düzenlenmeye başladığı sırasında oluşmuştur. Daha sonra LMS'in kendini toparlayarak hatayı azalttığı görülmektedir.



Şekil 2 : LMS hata verisi

Daha sonra LMS'e göre daha hızlı yakınsayan Yinelemeli En Küçük Kare (RLS-Recursive Least Square) yöntemi denenmiştir. Çalışma prensipleri birbirine benzeyen RLS filtreleme çıkışı ile LMS filtreleme çıkışından elde edilen konum değişimi bilgilerindeki hataların karşılaştırılması ise Şekil 3'teki gibidir. Burada koyu renk olan eğri LMS'e, daha açık renk olan eğri RLS'e aittir.

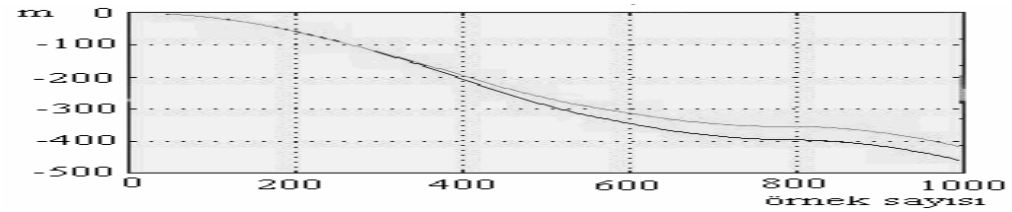


Şekil 3 : LMS ve RLS'in karşılaştırılması

Bu sonuçlara dayanarak, yapılan ivme verisinin filtrelenmesi çalışmasında LMS'in RLS'e oranla çok daha başarılı olduğunu belirtmek gerekir. Bunun nedeni büyük ölçüde daha önce de belirtildiği üzere Kalman filtresi gibi model bağımlı olan RLS'in, çalışmadaki model yapısıyla uygunluk sağlayamamasıdır.

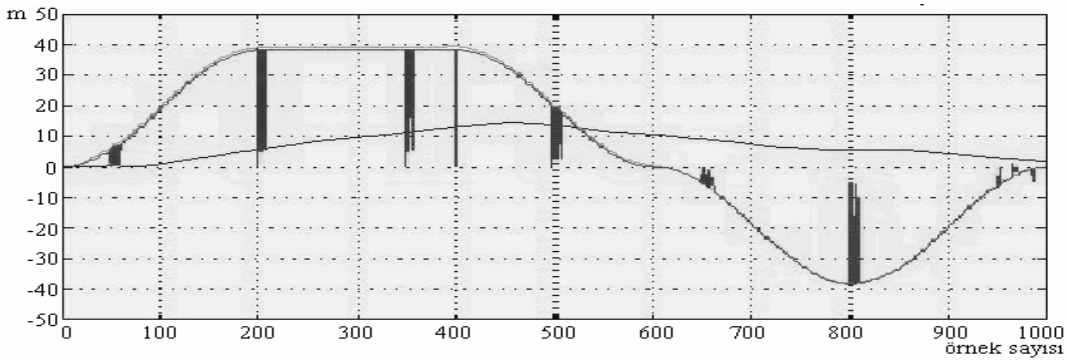
Uyarlamalı filtreleme yöntemlerinden günümüzde sıklıkla tercih edilenlerden birisi de Yapay Sinir Ağları (YSA) kullanımıdır. Ortam değişimleri için YSA'nın tekrar eğitilmesi gerekebilir fakat bu Kalman filtrelerinde uygulandığı gibi modelin baştan hazırlanması gibi bir işlem yerine kısa süreli olarak YSA'nın otomatik olarak yapacağı bir ağırlık düzenlemesidir. İlk olarak ileri beslemeli ağ (feedforward network) mimarisinde değişik tipteki çok katmanlı perseptron (Multi Layer Perceptron-MLP) "geri yayılım (backpropagation) algoritmaları" (GYA'lar) denenmiştir. Çalışmada sırasıyla Gradient Descent (GD), Gradient Descent with Momentum Backpropagation (GDM), Gradient Descent with Adaptive Learning Rate Backpropagation (GDX), Levenberg-Marquardt Backpropagation (LM) ve Resilient Back Propagation (RBP) eğitim yöntemlerini kullanan sinir ağı modelleri karşılaştırılmıştır. Modellerin hepsinde algılayıcıdan alınan aynı gürültülü ivme değeri dizisine aynı tip trapezoid sinyal eklenerek eğitim yapılmıştır. Bunların içinde Resilient Back Propagation ve ardından Levenberg-Marquardt algoritmaları en iyi sonuçları (gerçek sinyale en iyi yakınsamayı) vermişlerdir.

Bundan sonra MLP ile karşılaştırılmak üzere farklı YSA mimarilerinin arayışına geçilmiş ve Radyal Temel (Radial Basis) Fonksiyonlu (RTF) sinir ağının incelenmesine karar verilmiştir.



Şekil 4 : MLP ile RTF'nin konum bilgilerinin karşılaştırılması

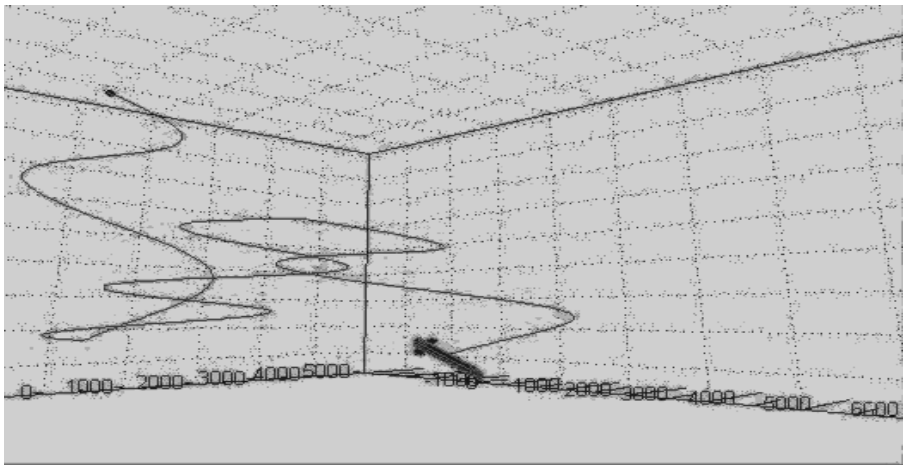
Şekil 4'te de görüleceği gibi RTF sinir ağı (açık renk eğri) MLP sinir ağlarından (koyu renk eğri) daha iyi sonuçlar vermektedir. RTF, gerek bu karşılaştırmanın sonucu gerekse de öğrenme hızı göz önüne alındığında sistemimiz için uygun YSA modeli olarak seçilmiştir. Ayrıca bu model kullanılarak filtrelemenin prosesin farklı aşamalarında yapılması denenmiş ve ivme, hız ve konum değişimi verilerine uygulanan filtrelemeler karşılaştırılmıştır (Şekil 5).



Şekil 5 : İvme, hız ve konum bilgilerinin filtrelenmesiyle elde edilen konum verilerinin hataları

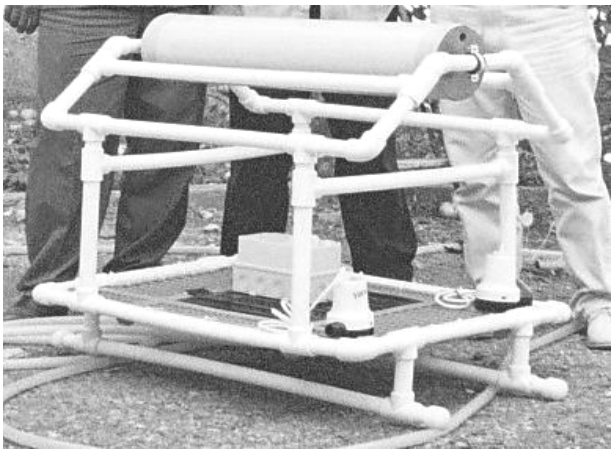
Bu karşılaştırmaya göre filtreleme işleminin ivme verisine (ortadaki koyu eğri) uygulanmasının hız (en açık rek eğri) ve konum değişimi (hız eğrisine benzeyen daha koyu eğri) filtrelemelerinden daha az hataya sahip olduğuna karar verilmiştir. Son olarak da sistemin değişik durumlara kolay uyum sağlayabilmesi açısından parçalı yapay sinir ağlarının kullanımı denenmiştir.

Çalışmanın filtreleme ile ilgili kısmından sonra elde edilen verileri kullanarak araç takibini gerçekleştirmek üzere kullanılacak, grafik arayüz de Simulink kullanılarak hazırlanmıştır (Şekil 6).



Şekil 6 : Konum takip arayüzü

Son olarak da deneylerde kullanılmak üzere bir sualtı aracı tasarlanmış ve imal edilmiştir (Şekil 7).



Şekil 7 : Sualtı aracının gövdesi

Araç gövdesi üzerinde aracın üç eksendeki hareketini sağlayacak itici motorlar, balast tankı ve motor sürücü devreleri ile aküleri taşıyan sızdırmaz kutular bulunmaktadır. Bunun yanı sıra aracın sudaki testlerinin ardından ataletsel ölçüm sistemi de araca yerleştirilecektir. Aracın kumandası radyo frekansı kullanan model uçak kumandasıyla gerçekleştirilmektedir.

TRAJECTORY TRACKING OF AN UNDERWATER VEHICLE BY SENSING ITS INERTIAL PROPERTIES

SUMMARY

The project, will involve measuring the acceleration of an underwater vehicle in X, Y, Z directions with a tri-axial accelerometer. Then the axial velocities are calculated by integrating the data once; and the vehicle's position changes are calculated afterwards by integrating twice.

For this purpose, first the accelerations of the vehicle are measured in three dimensions. The mathematical model of a truly calibrated accelerometer due to the gravitational force is:

$$I_f = f + b_f + e \quad \textbf{(1)}$$

Here f, b_f, e gives acceleration, sensor bias and noise respectively.

These errors (bf, e) should be filtered to get the real acceleration data. The desired results could not be received using the methods "moving average" and "curve fitting". On the other side, as the data given by the sensor are digitized (16 bit data), they do not carry any frequency information. Therefore it is not possible to do any application in the frequency domain or use classical filters like low pass, etc. In such cases adaptive filtering methods are used in practice.

As the system is wished to be applicable to different types of vehicles, it is decided to make the system model-free. For that, stochastic modeling methods like Kalman Filtering are not ued in this study. The first method applied is the least Mean Square (LMS) method. LMS is an adaptive filter that uses the steepest descent method. LMS does succesful corrections to the weight vector in order to achieve least mean square error. Using the steepest descent method the weight vector *w(n)* is formulated as follows:

$$w(n+1) = w(n) + \frac{1}{2}\mu[-\nabla(E\{e^2(n)\})] \quad \textbf{(2)}$$

Here μ is the step size parameter and controls the convergence characteristic of the LMS algorithm. $e^2(n)$ is the mean square error between the *y(n)* output and the reference signal. LMS algorithm is started using a *w(0)* value at n=0 that is appointed arbitrarily. After that by correcting the weight vector successfully, the least mean square error is obtained.

The LMS filtering model set by using the Simulink program is presented in figure 1.

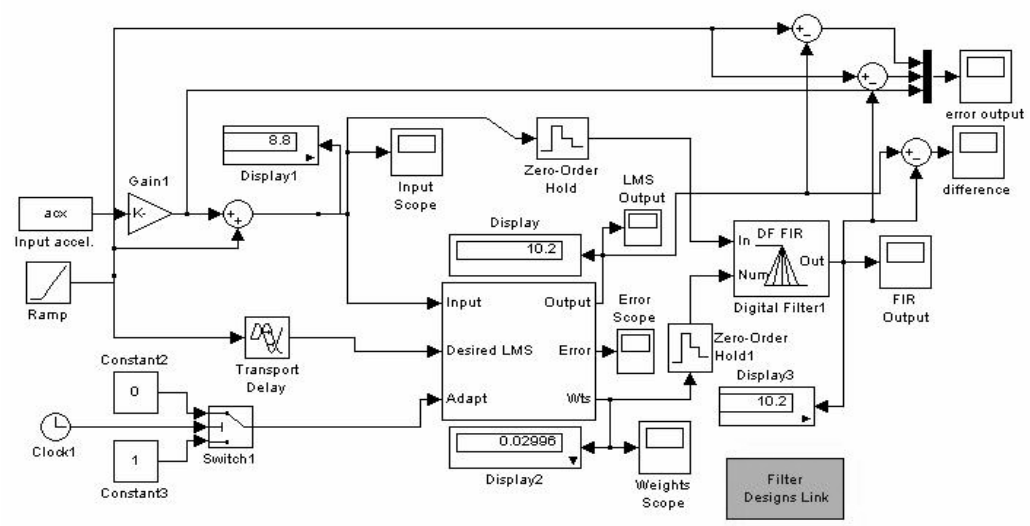


Figure 1 : LMS filtering model

In this model the noisy data obtained from the sensor is used as a disturbance and it is added to the known, "real" signal that is set by the user, in order to be used as the input. The "real" signal is fed to the filter through the other port as the desired signal. The LMS filter block of Simlink compares the input and the desired signal for changing the weight values to converge to the desired signal. Figure 2 shows the error of the LMS filter. In the graphic, the large error seen in the first 2000 samples, is occured during the beginning of the weight setting process of the filter and it is clearly seen that LMS reduces the error afterwards.

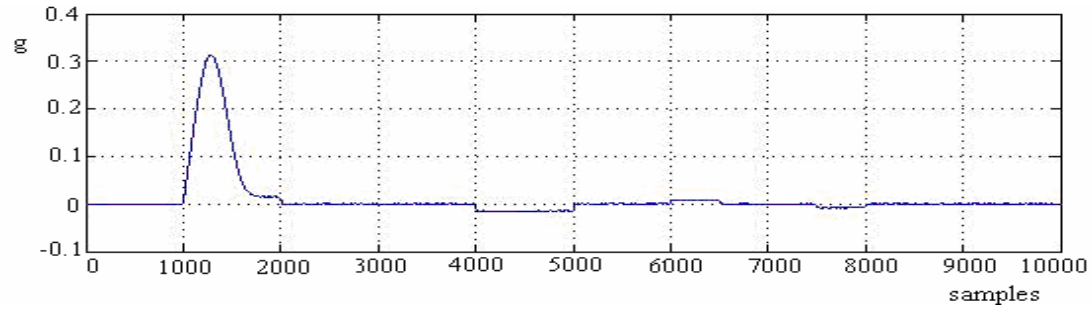


Figure 2 : LMS error data

Later on, another method, The Recursive Least Square (RLS) method is tested which converges faster than LMS. In figure 3, the displacement error comparison obtained of the filtering outputs of the two methods (RLS and LMS), that have similar working principles, is given. Here the dark curve shows the error of the LMS filter after 5500 samples an the lighter curve belongs to RLS.

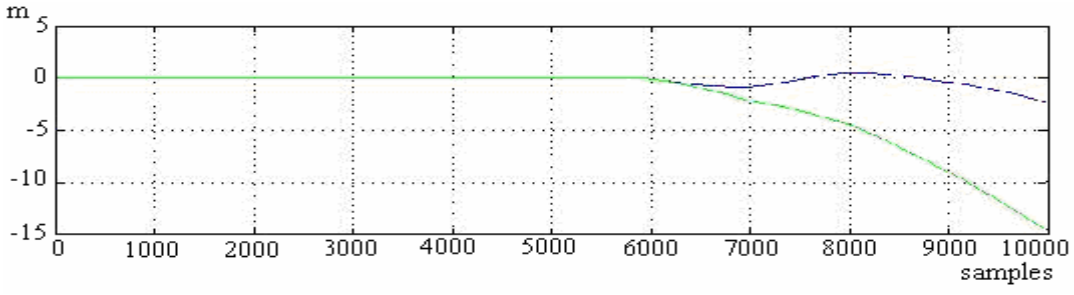


Figure 3 : Comparison of LMS and RLS

Upon these results it can be said that LMS is more accurate that RLS for such a filtering process. The main reason of this result is, as previously described, due to the model dependent nature of RLS just like Kalman filter.

One of the mostly preferred adaptive filtering methods is using a Neural Network (NN) system. Although the NN may need to be re-trained for different environments, this would not be as complex as creating a new model like done for Kalman filtering; it would be a weight forming done by NN itself in a short time. For NN, different types of Multi Layer Perceptron (MLP) Back-Propagation (BP) algorithms are tried first. At the study, networks that use Gradient Descent (GD), Gradient Descent with Momentum Backpropagation (GDM), Gradient Descent with Adaptive Learning Rate Backpropagation (GDX), Levenberg-Marquardt Backpropagation (LM) and Resilient Back Propagation (RBP) training methods are compared respectively. For all of the models the training is done by the combination of the same noisy data measured by the sensor and the trapezoidal signal. Upon these the best results are given by the Resilient Back Propagation and afterwords by the Levenberg-Marquardt algorithms.

After these results, the literature is search for a different NN architecture to compare with MLP, and Radial Basis Function (RBF) network is chosen to be tried.

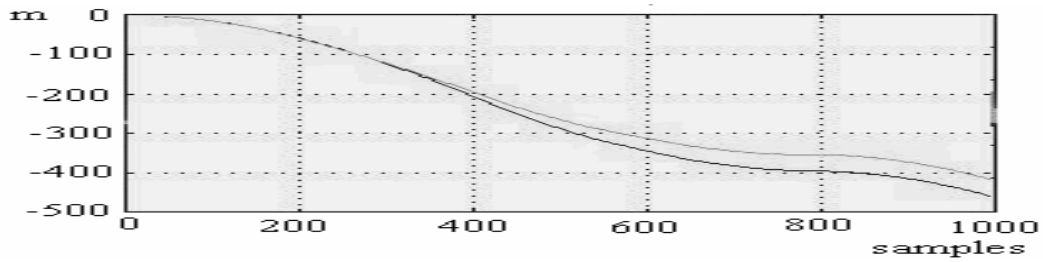


Figure 4 : Comparison of MLP and RTF's displacement data errors

As it is shown in figure 4, RBF (lighter curve) network results a better filtering than MLP (dark curve). RBF is chosen as the proper NN for our system both due to its convergence speed and the comparison. After that filtering process is applied to different stages of the data: acceleration, velocity and displacement (Figure 5).

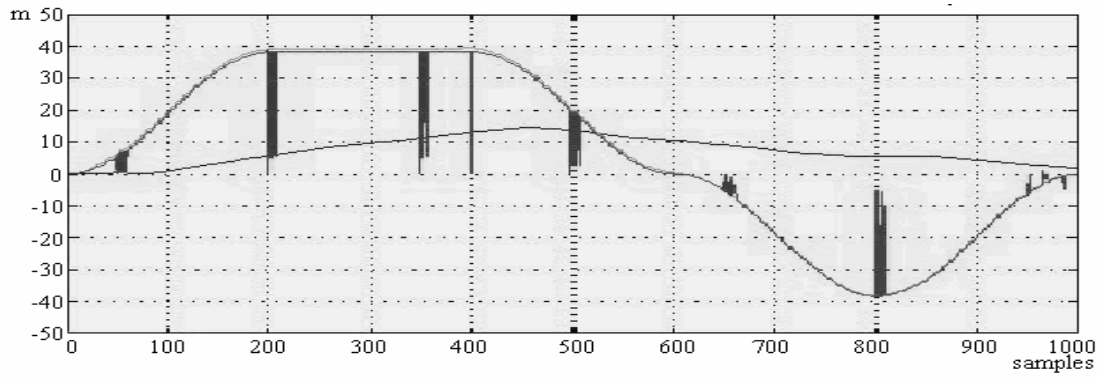


Figure 5 : Comparison of position data obtained from acceleration, velocity and position filtering

Result of the comparison is that filtering the acceleration (darkest curve in the middle), has less error in displacement than both the velocity (lightest curve) and the position (dark curve that looks like velocity error) filtering. As the last part of filtering a Partial Neural Network architecture is designed.

After completing the data filtering process of the study, a grafical interface (Figure 6) is programmed using Simulink in order to achieve the position tracking of the underwater vehicle by means of the position data derived.

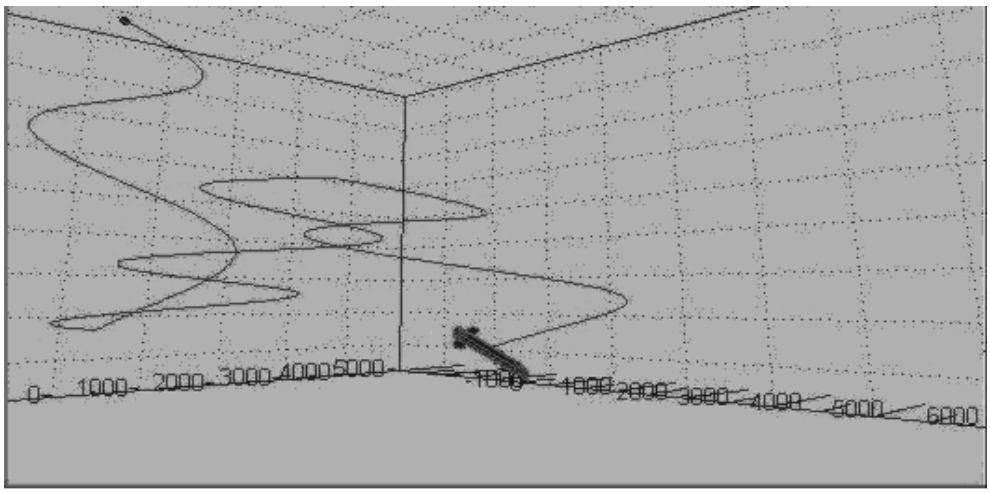


Figure 6 : Position tracking interface

As a latest stage of the study an underwater vehicle that is planned to be used in the experiments is designed and constructed (Figure 7).

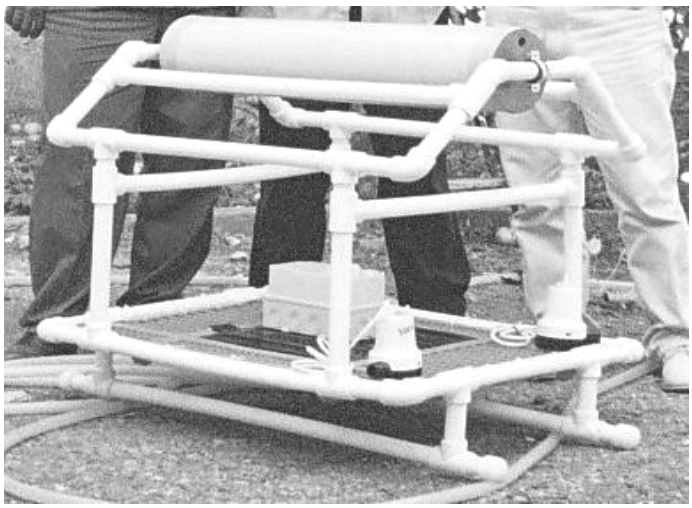


Figure 7 : The body of the underwater vehicle

The vehicle has the thruster motors that will drive it in three dimensions, a ballast tank and water-proof housings that carry the motor drive circuits and the batteries. Apart from that, the inertial mesurement system will be also added, after the first water tests of the vehicle. The vehicle is controlled by a radio control unit used for model planes.

1. GİRİŞ

Özellikle ülkemizde, mobil robotlar ve konum takibi üzerine yapılan çalışmaların çoğu kara araçları üzerinedir. Genelde tekerlekli olan bu araçların konum değişimleri, tekerleğin devir sayısına göre yer değiştirmenin hesaplanması veya çevredeki cisimlere olan uzaklığın değişiminin ultrasonik, kızılötesi gibi çeşitli sensörler yardımıyla hesaplanmasıyla yapılmaktadır [1]. Kara araçlarından sonra en çok incelenen tip ise hava araçlarıdır. Bunlar hareket serbestlikleri sebebiyle sualtı araçlarına benzemekle birlikte, yapılan çalışmalarda konum belirlemede kullanılan yöntemlerin genelde yeryüzeyi taraması, basınç ölçümleri ve en çok da küresel konumlandırma sistemi (GPS - Global Positioning System) olduğu görülmektedir [2, 3].

Halbuki günümüzde daha çok uzay araçlarının konum belirlemesinde kullanılan ve aracın ataletsel özelliklerine dayanarak çalışan ivme ölçerli sistemin [4], kullanım alanlarının uzayla sınırlı kalmayıp başta havacılık sektörü olmak üzere çeşitli alanlarda çalışmaların devam ettiği bilinmektedir [5]. Tokyo Metropolitan Üniversitesinden Yoshikazu Mori, ivmeölçerli sistemlerin faydaları arasında kolay kurulabilmelerini ve ulaşımı zor veya kapalı alanlarda da konum tesbitine olanak sağlamalarını saymaktadır [6].

İvmeölçerlerden elde edilen ivme verilerinin iki kez entegrasyonu sonucunda oluşan hataların süre uzadıkça eklenerek artması ve konum değerlerinin hesabında önemli hatalara neden olması önemli bir sorundur[7]. Liu ve Pang'ın iki eksenli bir robot kol ile yaptığı testler sonucu 8 m/s^2'lik bir ivme için 40 santimetrelik bir mesafede 1.08 santimetrelik hata oluştuğu ve ivme arttıkça hatanın da büyüdüğü görülmüştür [8]. Kalman filtresi ile kendi atadıkları düzeltme katsayılarını sisteme geri besleme olarak girerek hatayı minimize etmişlerdir.

Bu tip ivmeölçerli sistemlerin sualtı araçlarının hız ve konumunun belirlenmesinde kullanılması faydalı görülmektedir. Genelde sualtı araçları için basınç ölçümüyle derinlik tespiti ve sonar kullanımı önemli olmakla birlikte, son zamanlarda GPS'in

sualtı araçları için de kullanımı çalışmalarına ağırlık verilmiştir [9, 10]. Bu tip uygulamalarda ataletsel sensörlerin oluşturabilecekleri hataları ortadan kaldırmak için GPS sistemleriyle geri besleme yapılması önerilmektedir. Ancak bu sistemlerin de sürekli uydu bağlantısı gerektirmesi bazı sıkıntılar oluşturabilmektedir. COVELIA projesinde [11] konum takibi için GPS yerine sonardan faydalanılması önerilmektedir. Uydularla bağlantının derin sularda kesintiye uğrayacağının belirtildiği çalışmada ayrıca askeri çalışmalar gibi gizliliğin ön planda olduğu bazı durumlarda bu tip bilgi alışverişlerinin tehlike yaratabileceğine de değinilmektedir. Bunun yerine önerilen sonar sistemiyle deniz tabanının taranarak, gerçekleşen kayma miktarına dayalı konum değişiminin hesaplanmasının daha sağlıklı olacağı belirtilmektedir.

Yurt dışında ivmeölçerler yardımıyla sualtı araçlarının konum takibi konusundaki çalışmalara son birkaç yıldır ağırlık verilmekle birlikte [12, 13], yapılan çalışmalar incelendiğinde ayrık sistemlerin kullanıldığı ve ivmeölçerin yanında yine GPS veya sonardan oluşan ek bir algılayıcı ile sistemin desteklendiği görülmektedir.

Bu çalışmada, içinde üç eksenli ivmeölçer, jiroskop ve pusula bulunan tek parça algılayıcıdan [14] alınan çoklu verinin işlenerek, elde edilen sonuçlarla bilgisayar ortamında aracın yörünge takibinin yapılması planlanmaktadır. 3 eksendeki ivme değişimlerinin ve yine 3 eksende aracın yapacağı açısal hareketlerin tek bir algılayıcı tarafından alınmasının avantajı, tümleştirme nedeniyle hata oranının azaltılmış olmasıdır. Kullanılacak olan algılayıcı ünitesi normalde gerçek zamanlı veri işlemesi sırasında yapılan filtreleme sonucu oluşan faz kaymasını kendi içinde bulanan devre yardımıyla giderebilmektedir [11].

2. ATALETSEL ALGILAYICILAR

Bir cismin konumunun belirlenmesinde kullanılan iki yöntem vardır: O cismin başka bir noktaya göre bağıl konumunun belirlenmesi veya direkt olarak kesin konumunun belirlenmesi (örn: GPS kullanımı). Çoğunlukla kullanılan bağıl konum belirleme sistemlerinden biri de ataletsel seyir sistemidir (INS - Inertial Navigation System). Bu sistem genellikle[1] açısal değişimleri ölçen jiroskop ve ivmelenmeleri ölçen ivmeölçerden oluşur. İvmeölçerden elde edilen değerlerin iki kez entegrasyonu ile konumdaki değişim miktarları hesaplanır. İvme ölçerlerin küçük, ucuz olmak ve çok eksenli kullanım gibi birçok avantajları vardır [8].

Fakat yapılarına ve duyarlılıklarına bağlı olarak ölçümlerde önemli hatalar oluşturabilmektedirler.

2.1 Ataletsel Algılayıcı Hataları

Ataletsel algılayıcıların ölçüm sonuçlarını etkileyen hatalar şu sınıflara ayrılabilirler [15]:

Algılayıcının statik hatası (sensor bias), ölçek katsayısı (scale factor), eksen kaçıklığı (axes misalignment), gürültü.

2.1.1 Statik Hata

Jiroskop ve ivmeölçerler için statik hata, algılayıcının "sıfır" giriş değerinde, yani durağan halde çıkışında ölçülen değerdir. Jiroskop için bu hata (°/h) (rad/s) ile belirtilirken, ivmeölçer için birim (g) veya (m/s^2)dir. Statik hata, genelde "ofset" denen deterministik bölüm ve "kayma" denen rastlantısal bölümden oluşur. Bunun dışında algılayıcının statik hatasını etkileyen iki özellik daha vardır: Bunlardan ilki statik hata asimetrisidir (bias asymmetry). Algılayıcının pozitif ve negatif değerler için statik hatasında meydana gelen değişim olan bu özelliğin dışında, statik hata

[1] Bazı sistemler dünyanın manyetik alanına göre konum belirlemede kullanılan manyetometre de içerir.

dengesizliği (bias instability) denilen bir özellik daha vardır. Bu da statik hatada, belirli ortalama zaman aralıklarında hesaplanan rastlantısal değişimlerdir.

Ölçeklenmemiş bir ivmeölçerde oluşan statik hata, hız hesabında zamanla (t), konum değişimi hesabında ise zamanın karesiyle (t^2) orantılı olarak değişen bir hata değeri oluşturur.

$$v = \int b_f dt = b_f t$$
$$p = \int v dt = \int b_f t dt = 1/2\, b_f t^2 \qquad \textbf{(2.1)}$$

Burada v, b_f, p sırasıyla hız, statik hata ve konum değişimini gösterir.

2.1.2 Ölçek Katsayısı

Ölçek katsayısı, giriş-çıkış arasında genellikle en küçük kareler metoduyla uyarlanan bir doğrunun eğimi olarak belirlenir. Bu doğrunun hatası ölçekleme ile saptanabilir. Statik hatada olduğu gibi ölçek katsayısı için asimetri, ölçülen pozitif ve negatif girişlerin arasındaki farktır. Buradan giriş-çıkış fonksiyonunun sıfır girişte süreksiz olabileceği sonucuna varılabilir [16]. Ölçek katsayısının sıcaklık başta olmak üzere çeşitli etkenlerle değişimi, 'tekrarlanabilirlik' ya da 'ölçek katsayısı kararlılığı' olarak adlandırılır ve parça/milyon (ppm-parts per million) ile gösterilir.
Bir ivmeölçer için ölçek katsayısı şu formülle verilebilir:

$$I_f = S_T f + S_E f \qquad \textbf{(2.2)}$$

Burada S_T gerçek ölçek katsayısı, S_E ise hata oranıdır.

Statik hatada olduğu gibi bu hata miktarı da zamanla değişebilmektedir.

2.1.3 Eksen Kaçıklığı

Algılayıcının montaj hatasından kaynaklanır. Bu durumda her bir eksen diğer iki eksendeki ölçümlerden etkilenir.

Eksen kaçıklığının giderilmesi için yapılacak en temel uygulama sabit ivmeölçer için z ekseninde okunacak yerçekimi ivmesine (g) göre ayarlama yapmaktır. Bu durumda z ekseninde 1g değeri okunurken, x ve y eksenlerinde "0" değeri okunmalıdır.

$$f_x = g.\sin\theta_x$$
$$f_y = g.\sin\theta_y \qquad \textbf{(2.3)}$$

Eğer x ve y eksenlerinde sıfırdan farklı bir değer okunuyorsa bu, algılayıcının eksenlerinin gerçek eksen takımını göstermediğini ve algılayıcının yerleştirilmesinde belli bir açıyla kayıklık olduğunu belirtir.

2.1.4 Gürültü

Algılayıcının kendi iç yapısından kaynaklanan veya çevrede bulunan diğer bir elektronik cihazın ölçüm işaretine karışmasıyla oluşan ek işarettir.

İvmeölçerler ile yapılan ölçümlerde elde edilen verilerin içinde gerçek değerlerin yanında yukarıda sayılan statik hata, doğrusal ve doğrusal olmayan ölçekleme hataları, eksen kaçıklığı, gürültüler ve yerçekimi ivmesinin etkileri de bulunmaktadır. Yapılan hareketlerin doğru gözlenebilmesi için alınan verideki hataların mümkün olduğunca yok edilmesi gerekmektedir.

2.2 Hata Giderme Yöntemleri

Oluşan statik ve ölçekleme hatalarının giderilmesi için kullanılan başlıca üç temel yöntem vardır [15]:

Kestirim (Estimation), Fark Alma (Differencing) ve Ölçekleme (Calibration).

2.2.1 Kestirim

Yerçekimi ivmesine göre çok iyi ayarlanmış bir ivmeölçerin matematiksel modeli kısaca şu şekilde tanımlanabilir:

$$I_f = f + b_f + e \tag{2.4}$$

Burada f, b_f, e sırasıyla ivme, statik hata ve gürültüyü göstermektedir. İvmeölçeri yerel yerçekimi ivmesine göre ayarladığımızda yapılan ölçüm değeri

$$I_f = -g + b_f + e \tag{2.5}$$

olur. Buradan statik hata:

$$b_f = (I_f + g) + e \tag{2.6}$$

olarak öngörülebilir.

Statik hatanın tahmini değeri yerel yerçekimi ivmesinin doğru öngörüsüne bağlıdır. Burada gürültü değerinin küçük kabul edildiğini unutmamak gerekir.

2.2.2 Fark Alma

Bilinen iki yerel yerçekimi değeri için yapılan ölçümlerde elde edilen sonuçların farkı alınarak statik hatanın giderilmesi sağlanabilir [15].

2.2.3 Ölçekleme

Genellikle laboratuvar ortamında yapılan bu işlemde cihaz ölçümleri bilinen bilgilerle karşılaştırılarak gerekli katsayı ayarlamaları yapılır. Ölçekleme sırasında cihaz, pozisyonu kesin olarak bilinen sabit bir tabla üzerine yerleştirilerek algılayıcı yukarı ve aşağı dönük olacak şekilde ölçümler yapılır.

$$I_{f\,yukar} = b_f - (1+S)g \tag{2.7}$$

$$I_{f\,asagı} = b_f + (1+S)g \tag{2.8}$$

Burada b_f, S, g sırasıyla statik hata, ölçek katsayısı ve yerçekimi ivmesini göstermektedir. Buradan z-ekseni için statik hata ve ölçek katsayısı değerleri şu şekilde elde edilir:

$$b_f = (I_{f\,yukarı} + I_{f\,asagı})/2 \tag{2.9}$$

$$S = (I_{f\,yukarı} - I_{f\,asagı} - 2g)/2g \tag{2.10}$$

Uzun süreli uygulamalarda sağlıklı sonuç elde edebilmek için hatanın modellenmesi ve ardından da filtrelenmesi gerekir.

2.3 Hatanın Modellenmesi

Algılayıcı verisinin doğruluğunun kontrolü için kullanılan modelleme ve filtreleme yöntemlerinde yapılan, önceden hesaplanan hata modelinin alınan ölçümlerden çıkan değerlerle farkını alarak mümkün olduğunca gerçeğe yakın sonuçlar elde etmeye çalışmaktır. Bu uygulamalardan en yaygını Kalman Filtresidir. Haiying Hou [15], Kalman Filtresi ve diğer bazı modelleme yöntemlerini karşılaştırmıştır (Tablo 2.1).

Tablo 2.1 : Hata modelleme yöntemlerinin karşılaştırılması

	Oto-korelasyon	**Spektral Güç Yoğunluğu**	**Varyans Tekniği**	**Adaptif Kalman Filtresi**
Faydaları	-Periyodik olan ve olmayan işaretlerin analizi için uygundur. -Fourier dönüşümü Spektral güç yoğunluğu ile uyumluluk sağlar.	-Periyodik olan ve olmayan işaretlerin analizi için uygundur. -Fourier dönüşümü otokorelasyonla uyumluluk sağlar.	-Verinin zaman domeninde ayrıntılı anlaşılabilmesi -Farklı hatalara yol açan parametrelerin kolayca belirlenmesi	-Sinyalin içindeki yüksek frekanslı bilgileri yakalayabilme -Durumlara kendiliğinden adapte olabilme
Sınırlamalar	-Öncelikle gürültü giderme yöntemi uygulanmalıdır. -Kabul edilebilir sonuç için uzun süreli işaret gereklidir	-Frekans ortalaması yöntemi gerekir. -Parametre çıkarımı karmaşıktır.	-Sonuçlar işlemin yapısının anlaşılmasına bağlıdır. -Kestirim doğruluğunu artırmak için uzun süreli veri gerekir.	-Hesapların karmaşıklığı gittikçe artar. -Olası ciddi hataların bulunması algoritmanın uygulanmasını sınırlar.

Genellikle kontrol çevrimleri için kullanılan Kalman Filtresi kullanımlarında filtrenin parametrelerinin doğru ayarlanmasına dikkat etmek gerekir.

Liu ve Pang'ın [8] iki eksenli bir robot kol ile yaptığı testlerde oluşan yaklaşık 2 mg statik hata nedeniyle dakikada 35.32 m'lik bir pozisyon hatası hesaplamışlardır. Kalman filtresi ile kendi atadıkları düzeltme katsayılarını sisteme geri besleme olarak girerek 8 m/s^2'lik bir ivme için 40 santimetrelik bir mesafede (-)1.08 santimetrelik minimum hata değerine ulaşmışlardır. 10 m/s^2 ivme için ise sonuç (+)1.55 cm olmuştur. Buradan ivme arttıkça hatanın da büyüdüğü görülmüştür.

2.4 Projede Kullanılan Algılayıcı

Microstrain [14] firması tarafından üretilen 3DM-G ataletsel ölçüm ünitesi (Şekil 2.1), bünyesinde 3 ivmeölçer, 3 jiroskop, 3 manyetometre ve 1 sıcaklık algılayıcısı bulundurmaktadır. Bu sayede ünitenin 3 eksendeki ivme, açısal hareket ve manyetik alan değişimleri okunabilmektedir. Ayrıca içinde bulunan bir mikroişlemci sayesinde

okuduğu değerleri istenildiğinde filtreleyerek daha optimal çıkışlar verebilmektedir. Sıcaklık algılayıcısı yardımıyla ünite içindeki sıcaklık sürekli kontrol edilmekte ve sıcaklığa bağlı olarak oluşan hatalar minimuma indirilmektedir.

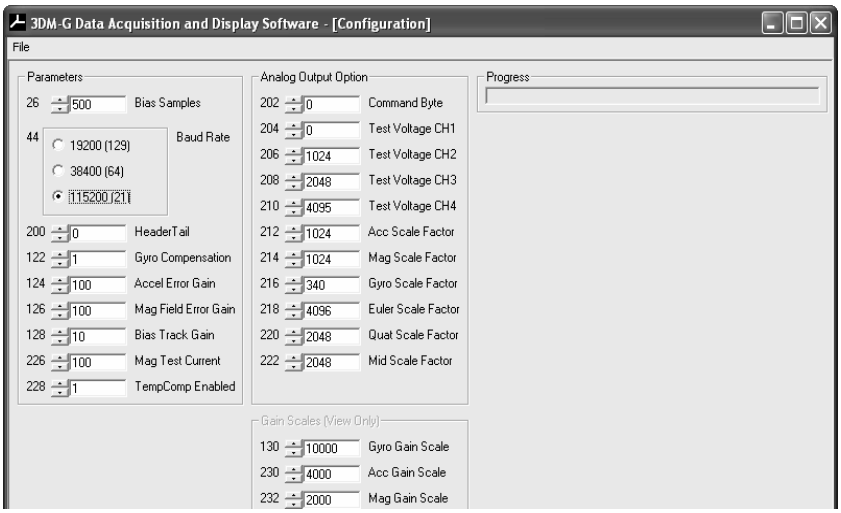


Şekil 2.1 : Microstrain 3DM-G Algılayıcı

Kendi kullanıcı ara yüzüne sahip olan üniteden veri okuması, sürekli veya kullanıcı isteğiyle olmak üzere iki şekildedir. Kullanıcı isteği durumundayken 76 Hz veri çıkışına sahip ünitenin istenildiğinde gerek veri çıkış frekansı, gerekse de algılayıcı parametreleri Şekil 2.2'de görülen kullanıcı ara yüzü kullanılarak ayarlanabilmektedir.

2.4 Algılayıcıdan İvme Verilerinin Okunması ve Değerlendirilmesi

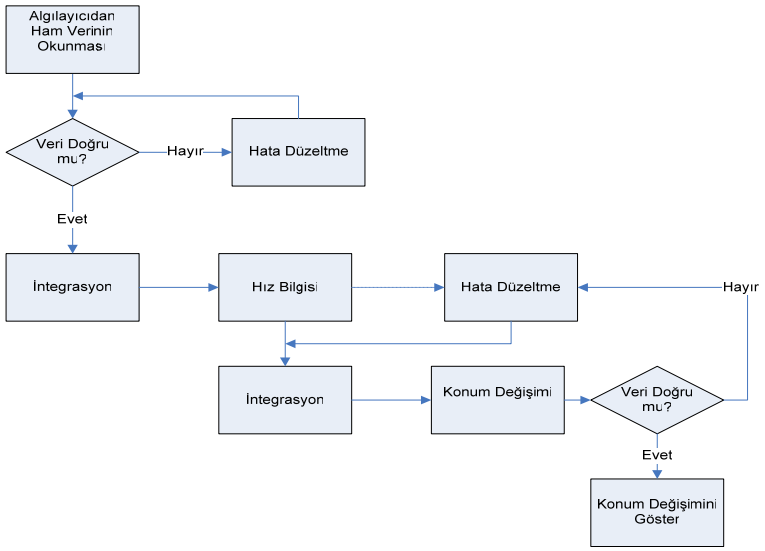
Matlab kullanılarak yapılan veri okuması işleminde öncelikle sayısal çıkış veren algılayıcı ünitesinden hangi tip veriyi isteyeceğimizi seçmemiz gerekir. Algılayıcı isteğe bağlı olarak 'ham' veri, 'filtre edilmiş' veri, 'Euler açılarına dönüştürülmüş' veri gibi çeşitli aktarımlar yapabilmektedir.



Şekil 2.2 : Algılayıcı parametreleri

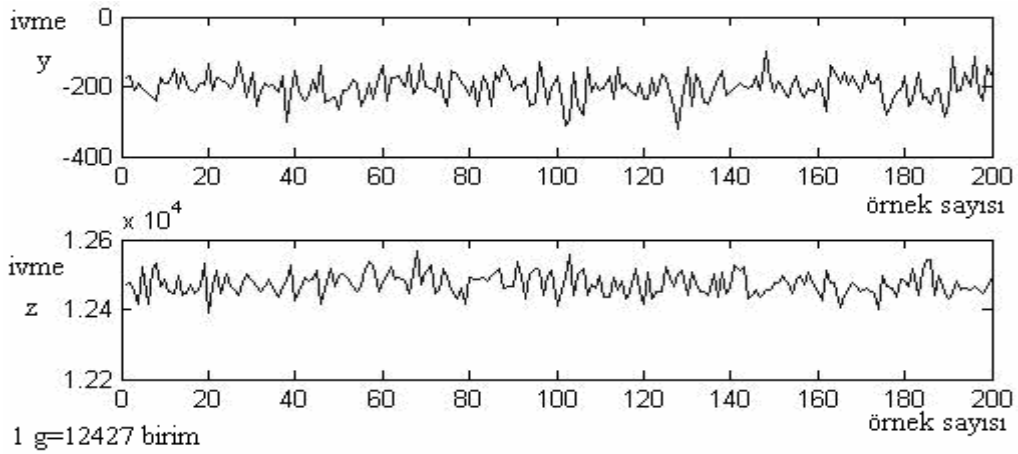
İstenilen veri tipi seçildikten sonra bu verilerin açısal değişim, ivme gibi belirli birimlere dönüştürülmesi için gerekli Matlab kodunun yazılması gerekir. Kullanılan bilgisayarın performansına bağlı olarak Matlab'in veri okuma hızı da düşmektedir (yapılan ölçümlerde saniyede 70 defa veri okuması yapılmıştır). Ölçüm süresi yerine algılayıcıdan alınan örnekleme hızı kontrol edilebildiği için veri aktarım hızı ölçüm süresinin belirlenmesinde dikkat edilmesi gereken bir unsurdur.

Belirli örnekleme hızları için yapılan ölçümler sonunda elde edilen ivme, açısal değişim ve manyetik alan değişim değerleri bir dosyaya yazılmakta, ayrıca her bir eksen için bu değişimler grafiksel olarak alınmaktadır. Daha sonra bu dosyadaki veriler işlenerek hataların giderilmesi ve doğru konum değişimlerinin elde edilmesi öngörülmüştür. Şekil 2.3'te çalışmada uygulanan yöntemin akış diyagramı görülmektedir.



Şekil 2.3 : Uygulanan yöntemin akış diyagramı

İlk olarak algılayıcıdan sabit konumda ham ivme verileri alınmıştır. Algılayıcının sayısal olarak verdiği çıkış değerleri 0 ile 65535 arasında değişmektedir ki bu da analog çıkış olarak 0–5 V aralığına karşı gelmektedir. Bu değerlerin 0'dan 32768'e kadar olan kısmı negatif yöndeki ivmelenmeleri, 32768 ile 65535 arasındakiler ise pozitif yöndeki ivmelenmeleri göstermektedir. Bu nedenle okunan değerlerden 32768 çıkartılarak elde edilen veriler mutlak ivme değeri olarak göz önüne alınmıştır.

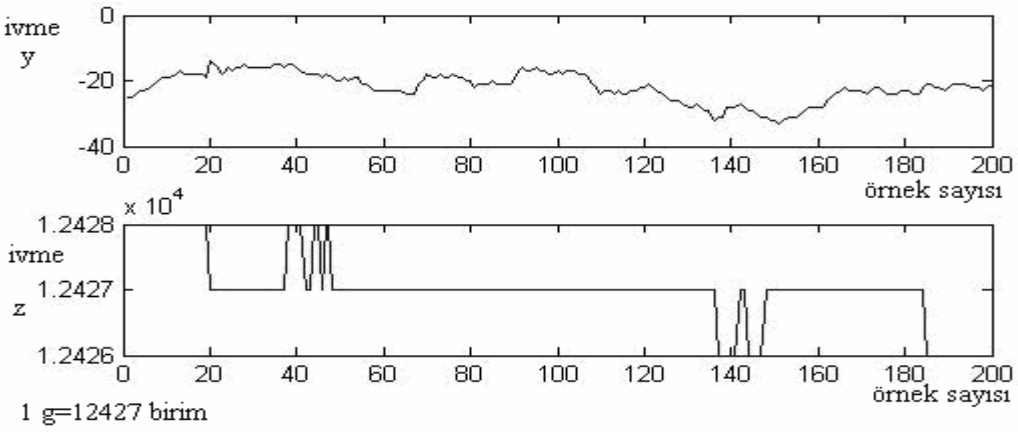


Şekil 2.4 : İşlenmemiş ivme verileri

Şekil 2.4'te de görüldüğü gibi 3 eksende (yukarıdan aşağıya sırasıyla x, y, z) elde edilen ivme verilerinde sabit değer okunması beklenirken statik hatalar tespit edilmiştir. Burada yatay eksen "örnek sayısını" gösterirken, dikey eksen ölçeklenmemiş ivme değerlerini vermektedir. Ayrıca sistemde sürekli bir gürültü olduğu da gözlemlenmiştir.

Bundan sonra algılayıcının kendi iç işlemcisi tarafından filtrelediği işlenmiş veriler alınarak incelenmiştir (Şekil 2.5).

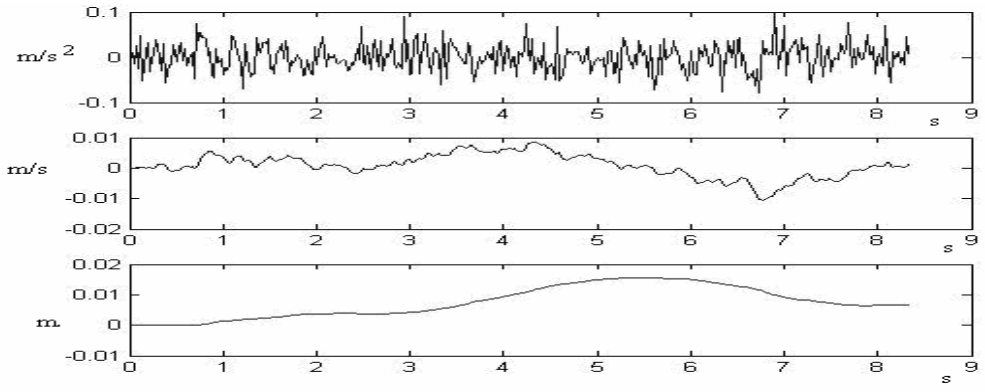
Burada statik hataların büyük oranda giderildiği görülmüştür. Ancak bu verilerin konum belirlenmesi amacına yönelik olarak kullanılmasında filtre parametrelerinin seçiminin kümülatif hatalar getirdiğinin deneysel olarak belirlenmesi sonucu, yapılacak işlemlerde ham verilerin kullanılmasına, ve hataların kendimiz tarafından yazılacak algoritmalar ile azaltılmasına karar verilmiştir.



Şekil 2.5 : Algılayıcı tarafından işlenmiş ivme verileri

Algılayıcı sabit konumda iken:

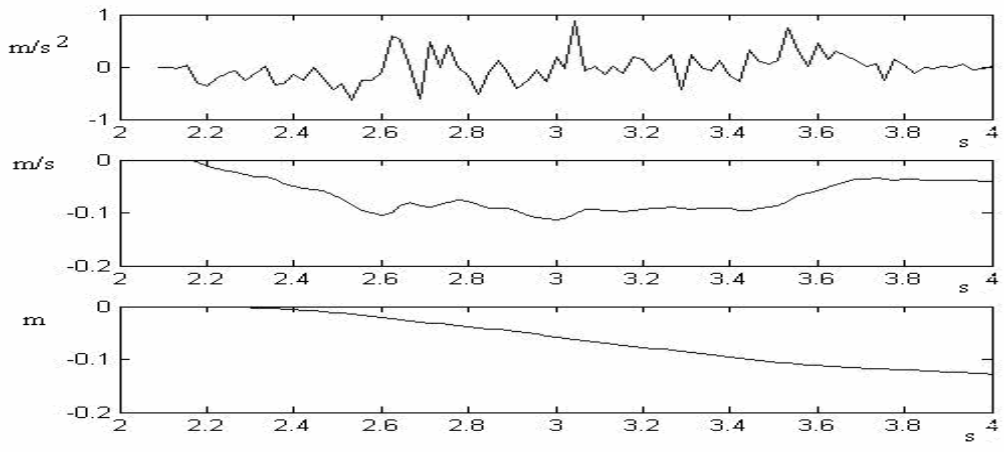
İlk okunan ham ivme verilerinin[2] iki kez integrasyonu sonucunda elde edilen konum değişimi grafiklerinde (Şekil 2.6) sıfır değeri beklenirken yaklaşık 10 saniye sonunda -statik hata değerinin ivme bilgisinden çıkarılmasının ardından- elde edilen konum değişimi grafiklerinde konumdaki hatanın gürültü ve sıcaklığın etkisiyle 1-2cm mertebesinde olduğu görülmüştür.



Şekil 2.6 : Sabit algılayıcı için ivme, hız ve konum değişimi verileri

Algılayıcı hareket ettirilerek yapılan ölçümler:

Algılayıcının x ekseni doğrultusunda 10cm ileri hareket ettirilmesi sonucunda elde edilen konum değişimi grafiği bize algılayıcının yaklaşık 12,8cm ilerlediğini göstermektedir. Oluşan bu hatanın temeli kaynağın gürültüsüdür. (Şekil 2.7).



Şekil 2.7 : Hareketli algılayıcı için ivme, hız ve konum değişimi verileri

[2] Ölçümler sadece x ekseni için yapılmaktadır. Ayrıca bu ölçümlerde algılayıcının eksen takımının doğru olduğu varsayılmaktadır.

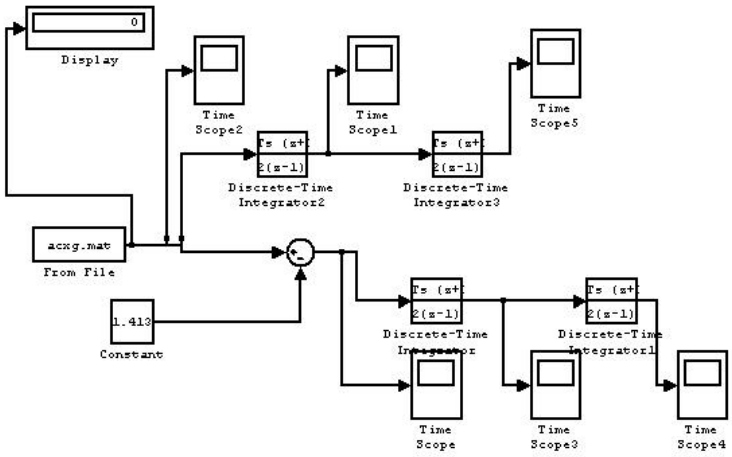
Sistemdeki bu gürültünün giderilmesinde uygulanacak filtreleme yönteminin seçimi için dikkat edilmesi gereken en önemli husus filtreleme sırasında cisimde meydana gelecek ani ivmelenmelerin de kaybedilmemesidir.

Hata minimizasyonu için incelenecek çeşitli filtreler yanında, okunan ivme değerinde herhangi bir filtreleme yapmadan tüm değerlerin alınması ve filtreleme işinin hız değerleri üzerinde yapılması seçeneği de düşünülmektedir. Buradaki düşünce aracın hızında çok ani artış ve azalmaların peş peşe olmayacağı, bir başka deyişle aracın sarsılmadan, teklemeden sakin hızlanma ve yavaşlama hareketleriyle ilerleyeceğinin varsayılmasıdır.

3. ALGILAYICI VERİLERİNİN FİLTRELEMESİ

3.1 Temel Filtreleme Yöntemleri

Filtreleme işlemleri için, daha önceden Matlab yardımıyla algılayıcıdan alınan verilerin aktarıldığı dosyaları giriş bilgisi olarak kullanan bir Simulink şeması kullanılmaktadır. Buradaki amaç kod yazımıyla uğraşmadan Simulink'in içinde bulunan kütüphanesini ve komut bloklarını kullanarak hızlı ve kolayca testler yapmaktır.

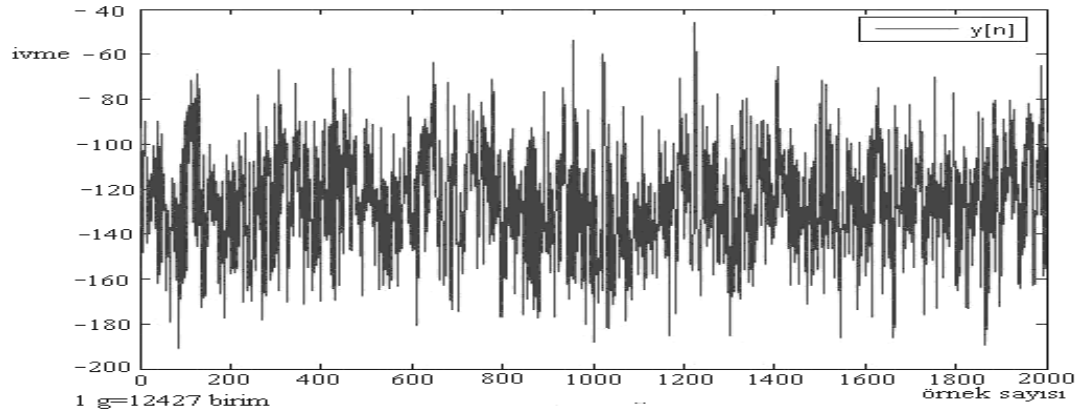


Şekil 3.1 : İvme bilgilerinden hareketle konum belirlenmesine ilişkin Simulink şeması

Şekil 3.1'de ivme değerlerinin, statik hatalarının giderilmesinin ardından entegre edilmesine ilişkin Simulink şeması görülmektedir. Burada ayrıca işlenmiş ve işlenmemiş sonuçların karşılaştırılması yapılmaktadır.

3.1.1 Kayan Ortalama

Gürültüyü giderme amaçlı yapılan ilk uygulama ivme verisinin kayan ortalamasını almak olmuştur. Gürültülü ivme verisinin belli bir değer etrafında (sabit konum için sıfır değeri etrafında) salınım yaptığı ve pozitif ile negatif yönlerdeki gürültü değerlerinin birbirini götüreceği öngörülerek yapılan işlemde sıfır ivme değeri için kullanılan gürültülü sinyalden Şekil 3.2'deki sonuç elde edilmiştir.

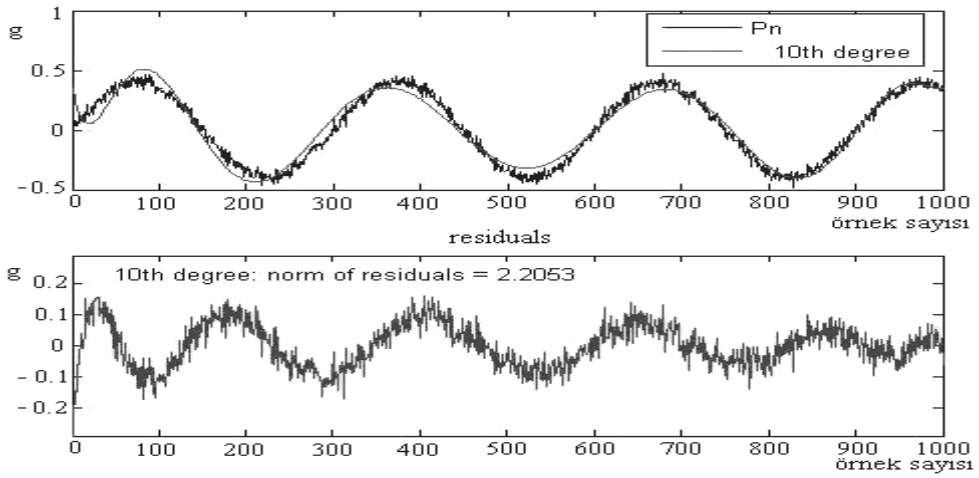


Şekil 3.2 : Kayan ortalaması alınmış gürültülü ivme değeri

Her ne kadar bu işlem sonunda gürültü azaltılmış olsa da istenilen sonuç elde edilememiştir.

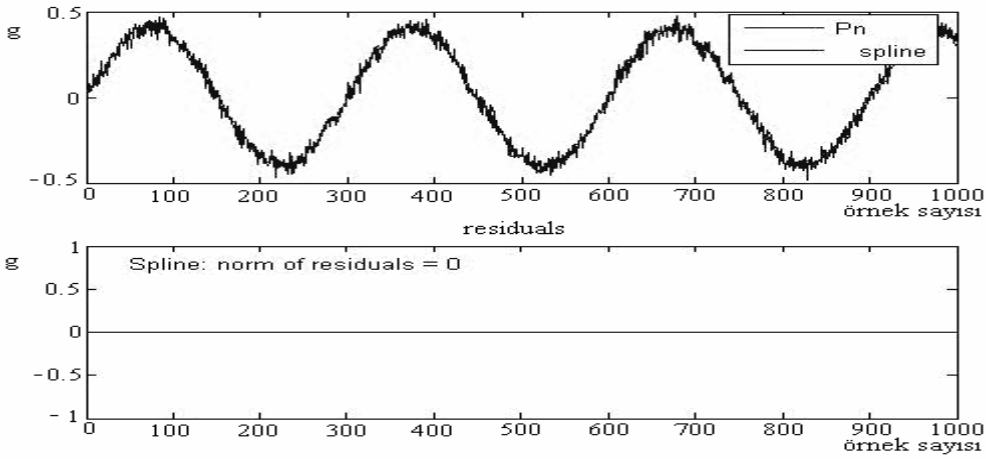
3.1.2 Eğri Atama

İkinci olarak ivme değerlerinin belli bir düzende olacağı varsayımından yola çıkarak eğri atama yöntemi ile gürültünün giderilmesi yoluna gidilmiştir. Bunun için Matlab'in "Basic Fitting" arayüzü kullanılmıştır. Burada, algılayıcıdan sabit konumda alınmış ivme verilerini tamamen gürültü kabul edip kendi hazırladığımız bir sözde "gerçek" ivme verisi ile birleştirerek oluşturduğumuz gürültülü sinyale çeşitli eğriler atayarak filtrelemeye çalışılmıştır.



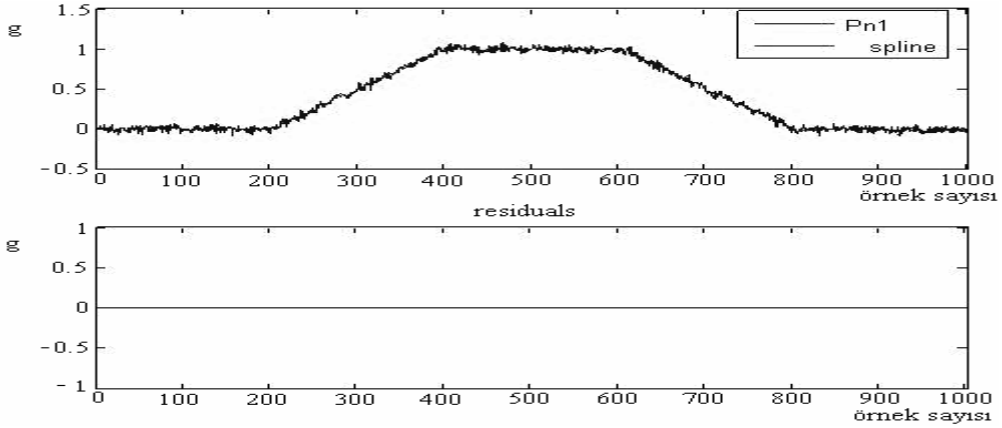
Şekil 3.3 : 10'uncu dereceden eğri atama sonucu

İlk olarak gürültülü bir sinüs eğrisine eğri uydurulmaya çalışılmış onuncu dereceden bir polinom ancak Şekil 3.3'te görüldüğü kadar sinyale uyabilmiştir. Alttaki grafik giriş eğrisi ile atanan eğri arasındaki hatayı göstermektedir. Daha sonra spline ataması yapılmış, bu durumda iki sinyal üstüste oturmuştur (Şekil 3.4).



Şekil 3.4 : Gürültülü sinyale spline eğrisi atanması

Ancak bu da beklediğimiz sonuç değildir; zira istenen sinyalin kopyalanması değil gürültünün giderilmesidir. Bu sonuç üzerine "gerçek" sinyal değiştirilerek trapez şeklinde bir sinyal denenmiştir.

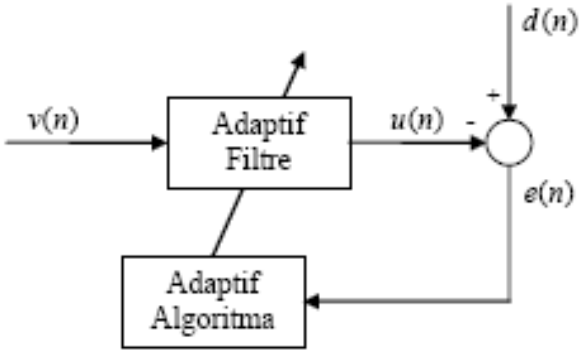


Şekil 3.5 : Trapez sinyale eğri atanması

Trapez şeklindeki sinyale spline dışında herhangi bir eğri atanamamış, bu atamadan ise bir öncekine benzer sonuç elde edilmesi (Şekil 3.5) ile eğri atama uygulamasından vazgeçilmiştir.

Bundan sonra çalışmada, literatürde yer alan benzer problemlerin çözümlerinden faydalanma yoluna gidilmiştir. Bu bağlamda çeşitli problemler incelendiğinde durumumuza en yakın örnek olarak ses sinyallerine uygulanan filtreleme işlemi göze çarpmaktadır. Akustik filtrelemesinde de bir gerçek sinyal ve bunun üzerine binen bir gürültü sinyali bulunmaktadır. Bu alanda yapılan uygulamalarda çoğunlukla gürültünün frekansına göre alçak geçirgen filtre vb. yöntemler ile filtreleme yapılmaktadır.

Ancak kullanılan algılayıcıdan alınan veriler sayısal olduğu için (okuma yapıldığı andaki değer 16 bit'lik veri olarak verilmektedir) herhangi bir frekans bilgisi içermemektedir. Bu nedenle frekans alanında işlem yapılması imkansızdır ve bu tip klasik filtreler kullanılamaz. Akustik filtrelemesinde de sistemimizde olduğu gibi belirli bir karekteristiğe ve sabit bir frekans aralığına sahip olmayan gürültülerin filtrelenmesi gibi sorunlarla karşılaşılmaktadır. Bu tür durumlarda uyarlamalı (adaptif) algoritmalara sahip filtrelerden faydalanılmaktadır. Bu algoritmalar Şekil 3.6'da görüldüğü gibi filtre parametrelerini iteratif olarak ayarlamak için kullanılır.



Şekil 3.6 : Adaptif (uyarlamalı) filtreleme işlemi

Burada *v(n)* adaptif filtrenin ayrık-zamanlı giriş sinyali, *u(n)* adaptif filtrenin tahmin edilen parametreler kullanılarak hesaplanan çıkış sinyali, *d*(*n*) ise adaptif filtrenin takip etmesi istenen sinyaldir. Parametre ayarlama algoritmalarıyla, tahmin edilen parametreleri kullanarak hesaplanan *u*(*n*) sinyali ile adaptif filtrenin izlemesi istenen *d*(*n*) sinyali arasındaki hatanın karesinin beklenen değerini minimum yapan parametre tahminleri iteratif olarak hesaplanır [17].

Bu tip filtreler içinde en çok kullanılan yöntemlerden biri de En Küçük Ortalama Karesi (Least Mean Square - LMS) algoritmasını kullanan filtrelerdir.

3.2 LMS Algoritması

LMS algoritması 1959 yılında Widrow ve Hoff tarafından uyarlamalı doğrusal eleman olarak bilinen bir desen tanıma makinası çalışmaları sırasında bulunmuştur (Adaline [18, 19]). LMS algoritması en dik inme (steepest decent) yöntemini kullanan uyarlamalı bir filtredir. LMS, en küçük ortalama kare hatayı sağlayacak şekilde ağırlık vektörüne başarılı düzeltmeler uygular. En dik inme yönteminden *w(n)* ağırlık vektörü şu şekil formüle edilir:

$$w(n+1) = w(n) + \frac{1}{2}\mu[-\nabla(E\{e^2(n)\})] \quad \textbf{(3.1)}$$

Burada μ adım büyüklük parametresidir ve LMS algoritmasının yakınsama karakteristiğini kontrol eder. $e^2(n)$ ise $y(n)$ çıkışı ile referans sinyal arasındaki ortalama kare hatadır:

$$e^2(n) = [d^*(n) - w^h x(n)]^2 \quad \textbf{(3.2)}$$

Yukarıdaki ağırlık güncelleme eşitliğindeki gradyan vektörü şu şekilde hesaplanabilir:

$$\nabla(E\{e^2(n)\}) = -2r + 2Rw(n) \quad \textbf{(3.3)}$$

En dik inme yöntemindeki en büyük problem r ve R değerlerinin hesabı ile ilgilidir. Diğer yandan LMS bunu, kovaryans matrisleri gerçek değerleri yerine anlık değerlerini kullanarak sadeleştirmektedir.

$$R(n) = x(n)x^h(n) \quad \textbf{(3.4)}$$

$$r(n) = d^*(n)x(x) \quad \textbf{(3.5)}$$

Buradan ağırlık güncellemesi şu denklem ile verilebilir:

$$\begin{aligned} w(n+1) &= w(n) + \mu x(n)[d^*(n) - x^h(n)w(n)] \\ &= w(n) + \mu x(n)e^*(n) \end{aligned} \quad \textbf{(3.6)}$$

LMS algoritması n=0 da keyfi atanmış bir $w(0)$ değeri ile başlatılır. Ağırlık vektörünün başarılı bir şekilde düzeltilmesi sonuçta en küçük ortalama kare hatanın elde edilmesini sağlar.

Buradan LMS algoritması aşağıdaki denklemlerle özetlenebilir;

Çıkış, $y(n)=w^h x(n)$ **(3.7)**

Hata, $e(n)=d^*(n)-y(n)$ **(3.8)**

Ağırlık, $w(n+1)=w(n)+\mu x(n)e^*(n)$ **(3.9)**

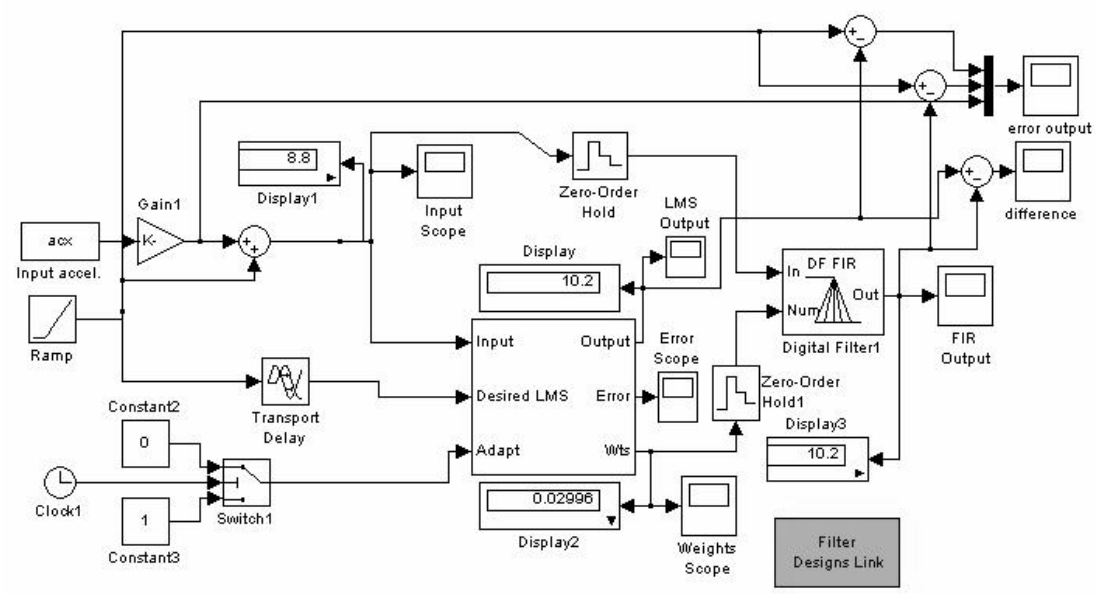
Adım büyüklüğü parametresi veya yakınsama faktörü olarak adlandırılan μ LMS algoritmasının temelini oluşturur. Keyfi atanmış bir ağırlık değeri ile başlatılan LMS algoritmasının,

$$0<\mu<1/\lambda_{max} \quad \textbf{(3.10)}$$

bağıntısına göre yakınsadığı ve kararlı kaldığı görülmektedir. Burada λ_{max} R korelasyon matrisinin aldığı en büyük özdeğerdir (eigenvalue). Algoritmanın yakınsaması R korelasyon matrisinin özdeğerlerinin yayılımı ile ters orantılıdır. R'nin özdeğerlerinin geniş bir yayılımı olduğunda yakınsama yavaşlayabilir. Korelasyon matrisinin özdeğerlerinin yayılımı, matrisin en büyük özdeğerinin en küçük özdeğere oranıyla belirlenir. Eğer μ çok küçük seçilirse algoritma yavaş yakınsar. Büyük bir μ değeri yakınsama hızını arttırırken en düşük değer etrafında kararlılığı azaltabilir; bu nedenle μ değerinin seçiminde hassas davranılmalıdır.

3.2.1 LMS Filtre Uygulaması

LMS algoritmasını kullanarak Simulink ortamında hazırlanan filtreleme modeli Şekil 3.7'de gösterilmiştir.

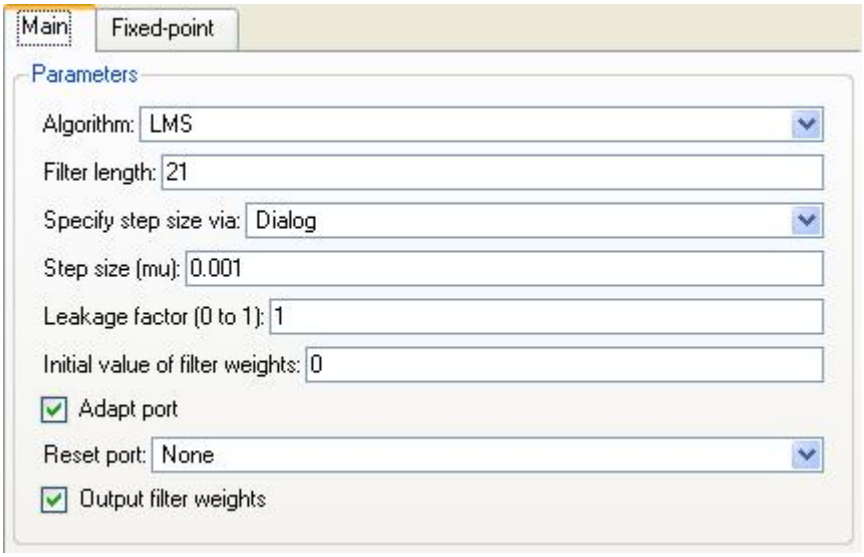


Şekil 3.7 : LMS filtre modeli

LMS filtresini öğrenebilen sonlu darbe yanıtlı (FIR-Finite Impulse Response) bir filtre olarak tanımlamak yanlış olmaz. Zira Şekil 3.7'de de görüldüğü üzere bir dijital FIR filtreye, işlenecek sinyalin dışında bu sinyali işlemek üzere filtrenin ihtiyaç duyduğu ve kullanıcı tarafından atanan ağırlıklar da girilmektedir. Diğer taraftan ilk

etapta bir öğrenim adımına sahip olan LMS filtre ağırlıklarını kendi atar ve en uygun sonucu elde edene kadar ağırlıkları değiştirmeye devam eder. Sistemin çalışma şekline LMS bloğunu inceleyerek başlayabiliriz. Şekil 3.7'de görüldüğü gibi bloğa (resmin ortasında üzerinde LMS yazan blok) bir "input" girişi bulunmaktadır, buradan filtreye işlenmesini istediğimiz gürültülü sinyal girilir. Bunun altındaki ikinci girişten ise filtre çıkışının yakınsamasanı istediğimiz sinyal girilir. Filtre belirli adım sayında bu iki sinyale göre ağırlık değiştirerek istenilen sinyale yakınsamaya çalışır. İyi bir yakınsama için yakınsanmak istenen gerçek sinyalin belirli miktarda geciktirilmiş olarak filtreye sokulması gerekir (Şekil 3.7'de istenen sinyalin LMS bloğuna girmeden önce geciktirildiği görülmektedir.). Sinyalin ne kadar geciktirileceğine dair bağıntı şu şekildedir:

Yakınsanmak istenen sinyal, *n* adımlı bir LMS filtresi için *(n-1)/2* defa geciktirilmiş olarak filtreye girilmelidir. LMS bloğunun parametrelerine bakacak olursak (Şekil 3.8) filtrenin 21 adımlı olduğu görülmektedir. Bu nedenle sinyali 10 adım geciktirerek LMS bloğuna girmeliyiz.

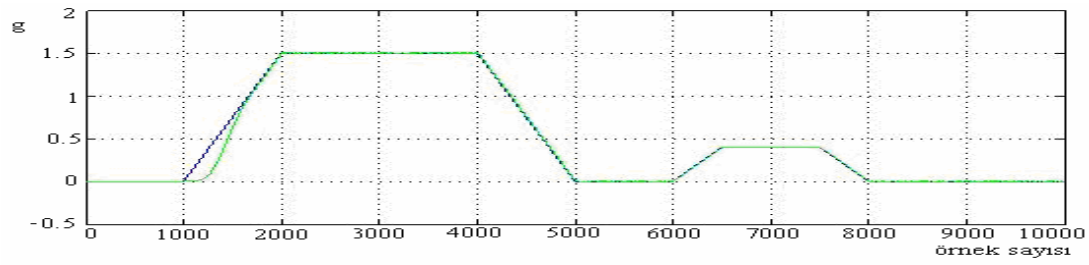


Şekil 3.8 : LMS blok parametreleri

Ayrıca μ (adım büyüklük parametresi) de bu menüden ayarlanabilmekte; istenirse de μ'ye sabit bir sayı atamak yerine dışarıdan değişken bir sinyal vs. de verilebilmektedir. Sızıntı çarpanı (leakage factor) ise programın LMS algoritmasını çalıştırırken kullanacağı bellek kapasitesini, bir bakıma değerlerin ne kadarını aklında tutacağını belirler. Bu parametrenin sıfır olması durumunda program hafıza kullanmaz ki bu durumda filtre yakınsamakta güçlük çeker. Bunun dışında filtrenin

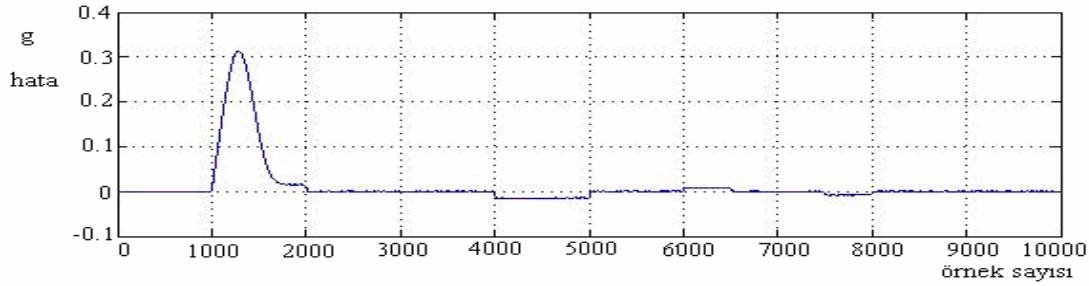
başlangıç ağırlığının atanması ve ağırlıkların gösterilip gösterilmemesi yine buradan seçilebilmektedir. “Adapt port” ile gösterilen seçenek filtrenin duyarlılığını bizim belirlememizi sağlar. Bu şık seçiliyle LMS bloğuna bir giriş daha oluşur (bkz. Şekil 3.7). Buradan sıfır veya sıfırdan farklı bir değer girilebilir. Girilen değer sıfırdan farklı olduğu sürece filtre ağırlıklarını yenilemeye devam eder; sıfır giriş olduğunda ise son ağırlık değerlerinde sabitler. Bir başka değişle eğitim sona ermiş, öğrenilen şekilde uygulamaya geçilmiştir. “Reset port” ise filtre ağırlıklarının sıfırlanmasını sağlar.

Şekil 3.9’da sırasıyla sisteme girilen gerçek sinyal (koyu renk) ile filtre çıkışı (açık renk) verilmiştir.



Şekil 3.9 : Gerçek sinyal (koyu renk) (m/s^2) ile filtre çıkışı (açık renk)

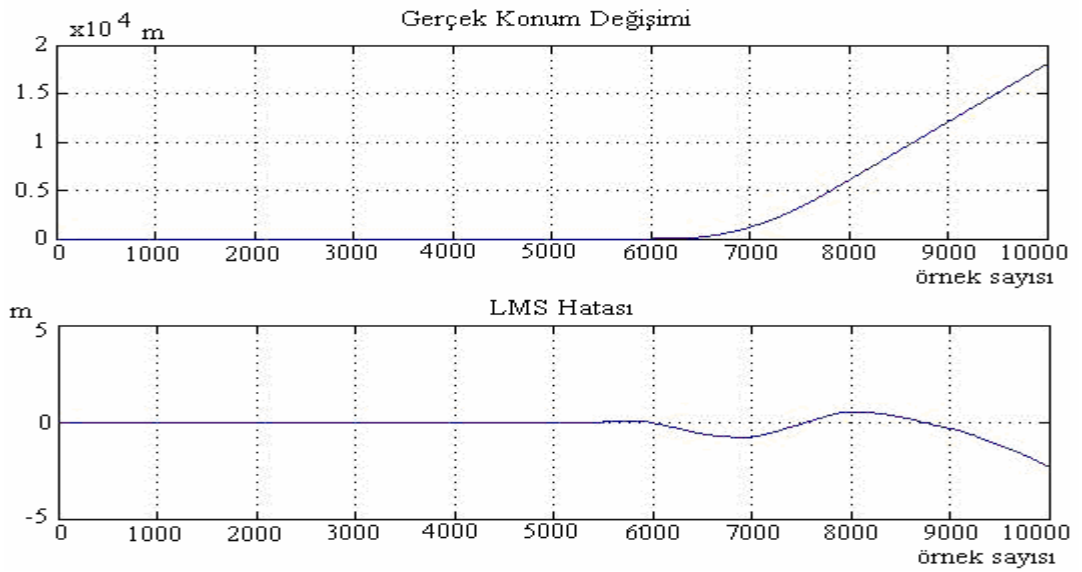
Şekil 3.9’da görülen çıkışa sahip LMS’in hatası Şekil 3.10’da verilmiştir. Burada ilk 2000 değerde görülen büyük ve değişken hata filtrenin ağırlıklarının yeni düzenlenmeye başladığı sırasında oluşmuştur. Daha sonra LMS’in kendini toparlayarak hatayı azalttığı görülmektedir.



Şekil 3.10 : LMS hata verisi

Bu hata miktarının konum değişimine yansıması Şekil 3.11’deki gibi olmaktadır. Şekil 3.11’de üstte görülen eğri gerçek konum değişimini verirken filtre çıkışının iki kez entegrasyonu ile elde edilen konum değişimi bilgisinin gerçek konum değişiminden farkı altta verilen eğridir. İlk 5000 değer eğitim sırasında kullanıldığı

için konum grafiklerinde bu kısımlar gösterilmemiş, hareketin 5000'den sonra başlamış olduğu kabul edilmiştir.



Şekil 3.11 : Gerçek konum değişimi (metre) ve LMS'in hatası

LMS algoritması basitliği ve belirgin performansı ile en genel kullanılan uyarlamalı algoritmalardan biridir. Tekrarlamalı bir algoritma olmasından dolayı zamana bağlı değişimi yüksek olan sinyal ortamlarında kullanılabilir. Farklı sinyal durumları için kararlı ve sağlam bir performansa sahiptir. Yine de bazı diğer algoritmalarla karşılaştırıldığında çok da hızlı bir yakınsamaya sahip olmadığı söylenebilir. Bu algoritmaların başında Yinelemeli En Küçük Kare (RLS-Recursive Least Square) gelmektedir.

3.2.2 RLS Filtre Uygulaması

RLS ile LMS algoritmaları kullanılarak yapılan filtreleme işlemlerinin sonuçları karşılaştırıldığında şu hükümler ortaya çıkmıştır [19]:

- RLS algoritmasının yakınsama hızı LMS'inkinden daha fazladır.
- RLS, giriş vektörünün korelasyon matrisinin özdeğer yayılımına karşı hassas değildir.
- Katsayılar ortalamada yakınsayabilir, bu nedenle RLS asimptotik olarak yansızdır. Öbür yandan LMS'in yanlılığı bulunmakta ve katsayılar optimal değerlere yakınsayamamaktadır.

- RLS'in istikrarlı halde ortalama kare hata düzensizliği LMS'inkinden oldukça azdır.

Buna rağmen yapılan karşılaştırma çalışmalarının çoğunda LMS algoritmasının RLS'e göre daha iyi bir takip özelliğine sahip olduğu görülmüştür. Bunun nedeni LMS'in model bağımsız bir yapıya sahipken RLS'in model bağımlı olmasıdır [21]. Bir bakıma denilebilir ki RLS algoritması Kalman filtresinin özel bir halidir.

RLS'in amacı ağırlıklı hataların karelerinin toplamını en aza indirmektir. Hata fonksiyonu,

$$E[e^2(m)]) = E\{[x(n) - a_k y(n)]^2\} \quad \textbf{(3.11)}$$

ile gösterilir. RLS algoritması yinelemeli bir işlemle o anki ağırlıklarını geçmiş ağırlıklarından hesap eder. RLS'in ağırlık güncelleme deklemi,

$$w(n) = w(n-1 + g(n)e^*(n) \quad \textbf{(3.12)}$$

olarak verilir. Burada * kompleks birleşiği (complex conjugate) temsil ederken *w(n)* adım ağırlıklarının Nx1 sütun vektörü, *g(n)* uyarlılık katsayısı ve *e(n)* de hata sinyalidir. Hata sinyali e(n),

$$e(n) = y(n) - w^H(n-1)x(n) \quad \textbf{(3.13)}$$

İle verilir. Uyarlılık katsayısı

$$g(n) = \frac{Px(n)}{Px(n)X(n)^H} \quad \textbf{(3.14)}$$

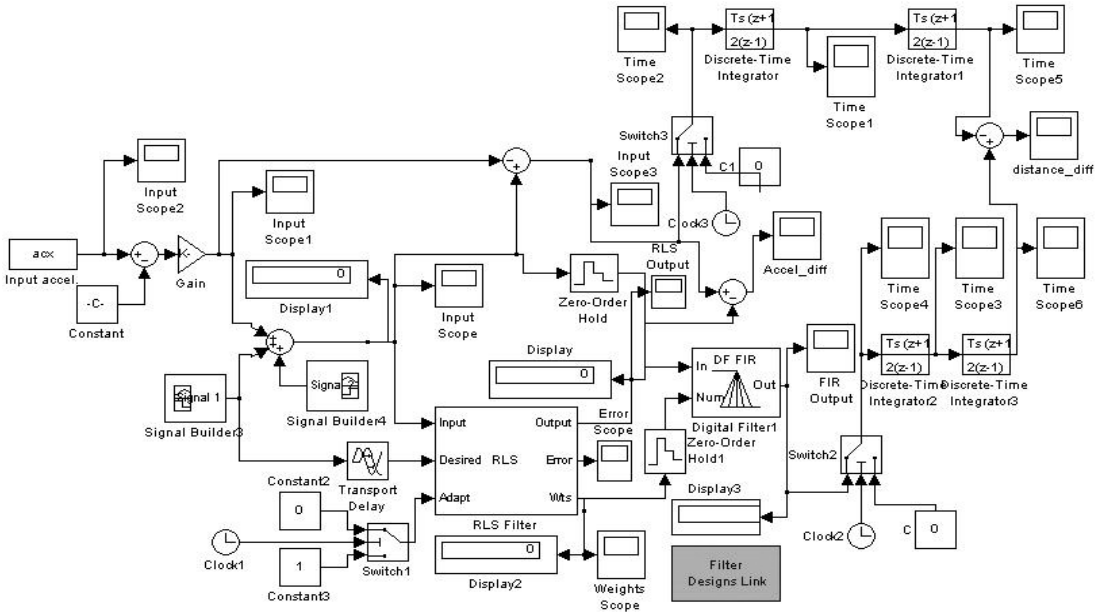
ile verilirken $P=R^{-1}(n)\ x(n)$'in ters kovaryans matrisidir. RLS algoritması $R^{-1}(n)$ değerlerini güncellerken,

$$P = \frac{P(I - g(n)x(n)^H)}{\lambda} \quad \textbf{(3.15)}$$

Denklemine göre önceki ters ve o anki örnekleri kullanır. Burada λ geçmiş verinin eksponansiyel ağırlığı için kullanılan birim değere çok yakın küçük bir skaler sayıdır.

Aynı zamanda unutma çarpanı olarak adlandırır. $()^H$ ise Hermitian'ı (i'inci satırı ve j'inci sütunu ile j'inci satırı ve i'inci sütünundaki her bir çift elemanı kompleks birleşik olan matrisi) göstermektedir.

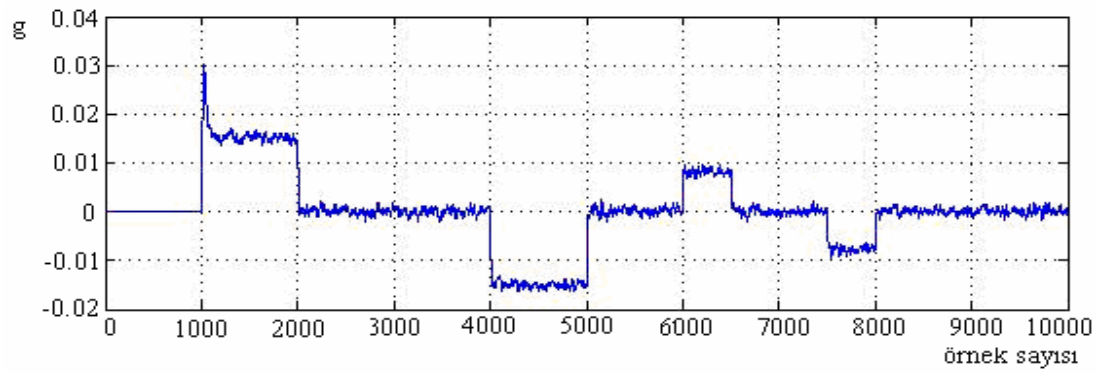
Çalışmanın bu kısmında, LMS algoritması ile uygulanan filtreleme düzeneğine benzer bir düzenek RLS için kurulmuş (Şekil 3.12) ve sonuçlar karşılaştırılmıştır.



Şekil 3.12 : RLS filtreleme modeli

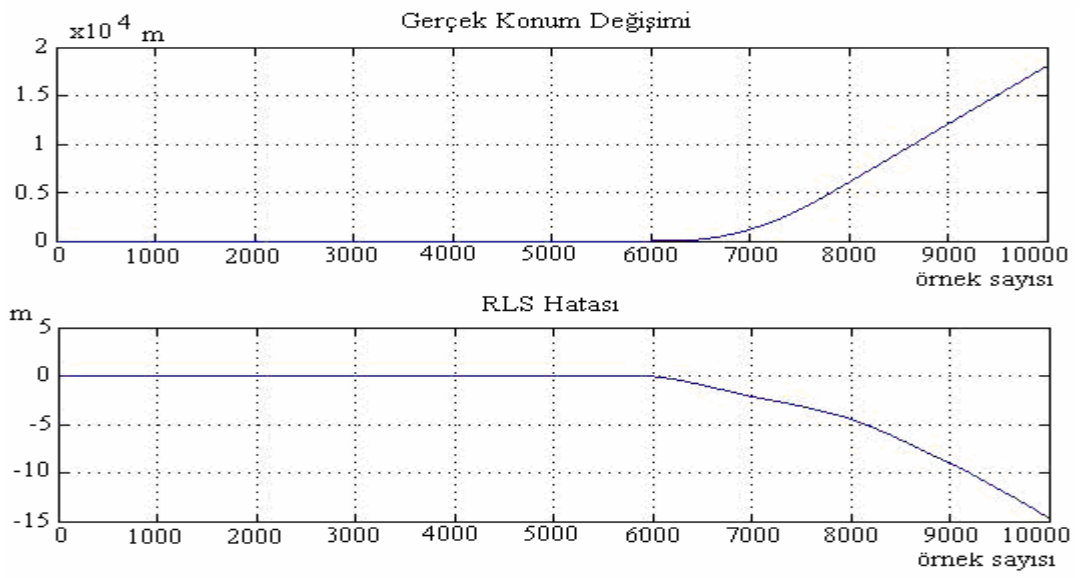
Her ne kadar RLS algoritması LMS'ten farklı olsa da Simulink blokları yapı olarak birbirine çok benzemektedir. Aralarındaki farklar: RLS'te μ yoktur, onun yerine filtre boyu belirlenir; sızıntı çarpanı yerine de λ unutma çarpanı bulunmaktadır.

RLS filtreye de LMS uygulamasında kullanılan gürültülü ve temiz sinyaller girildiğinde çıkışta Şekil 3.13'te görülen hataya sahip sinyal elde edilir.



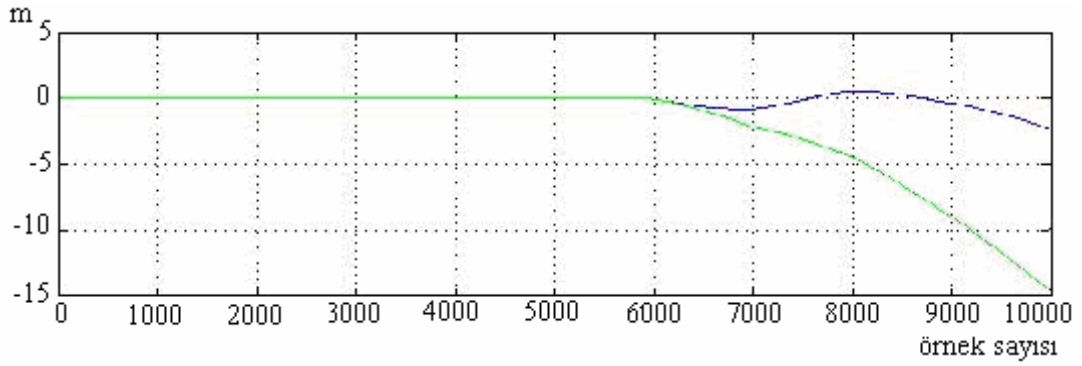
Şekil 3.13 : RLS filtre çıkış hatası

Gerçek konum değişimi ile bu çıkışın iki kez entegrasyona sokulmasıyla elde edilen konum değişimindeki hata yukarıdan aşağıya sırasıyla Şekil 3.14'te verilmiştir.



Şekil 3.14 : Gerçek konum bilgisi ve RLS'in hatası

RLS filtreleme çıkışı ile LMS filtreleme çıkışından elde edilen konum değişimi bigilerindeki hataların karşılaştırılması ise Şekil 3.15'teki gibidir. Burada koyu renk olan eğri LMS'e, daha açık renk olan eğri RLS'e aittir.



Şekil 3.15 : LMS ve RLS'in karşılaştırılması

Bu sonuçlara dayanarak yapılan ivme verisinin filtrelenmesi çalışmasında LMS'in RLS'e oranla çok daha başarılı olduğunu belirtmek gerekir. Bunun nedeni büyük ölçüde daha önce de belirtildiği üzere Kalman filtresi gibi model bağımlı olan RLS'in, çalışmadaki model yapısıyla uygunluk sağlayamamasıdır.

Stokastik modelleme yöntemleri de denilen Kalman ve benzeri yöntemlerde önce hata modellemesi yapılır, daha sonra bu hata sistematik olarak sinyalden çıkartılarak filtreleme gerçekleştirilir [15]. Buna bağlı olarak modelleme sırasında ve Kalman filtresinin katsayılarının belirlenmesinde çok dikkatli olunması gerekmektedir.

Kalman filtresi gibi model bağımlı yöntemlerin getirdiği hassasiyet ve güçlükler göz önüne alındığında filtreleme işlemini model bağımsız hale getirerek sistemin çevreye bağlı kalmadan değişik yapılarda ve ortamlarda kullanılabilir kılınması öngörülmüştür. Örneğin belirli bir sualtı aracı için tasarlanmış bir sistem herhangi bir yapısal değişiklik yapılmadan formu, yapısı değişik başka bir sualtı aracında ya da tamamen ortam değiştirerek bir kara taşıtında veya bir uçakta da kullanılabilir olmalıdır.

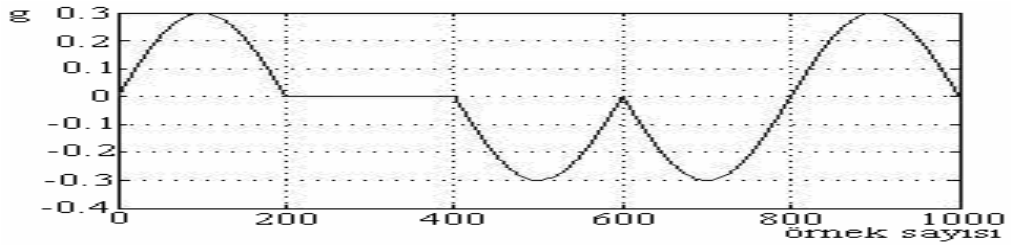
Uyarlamalı filtreleme yöntemlerinden günümüzde sıklıkla tercih edilenlerden birisi de Yapay Sinir Ağları (YSA) kullanımıdır. Ortam değişimleri için YSA'nın tekrar eğitilmesi gerekebilir fakat bu Kalman filtrelerinde uygulandığı gibi modelin baştan hazırlanması gibi bir işlem yerine kısa süreli olarak YSA'nın otomatik olarak yapacağı bir ağırlık düzenlemesidir.

3.3 Yapay Sinir Ağları ile Filtreleme

YSA, model bağımsız yapısı, farklı yapılara uyum sağlayabilmesi gibi pek çok özelliğiyle kullanım alanını genişletmekte, gün geçtikçe daha çok tercih edilmektedir. Yörünge takibinde kullanım örneklerine sıklıkla rastlamamıza rağmen, daha çok INS/GPS entegrasyonunda kullanıldıkları görülmektedir [22], [23]. Bu uygulamalarda YSA aracın gerçek konumunun takibi ve GPS'in kullanılamadığı durumlarda yörüngeyi tutturabilmek için son GPS verisine göre INS verisinin yakınsamasında kullanılmaktadır. Sualtı araçlarıyla ilgili bir örnekte ise YSA'nın araç kontrolü için algılayıcı ünitelerin birlikte çalışmasını sağlamada kullanıldığı görülmektedir [24]. Bu çalışmada ise YSA, GPS gibi yardımcı bir cihaz kullanmadan doğrudan yörünge takibinde kullanılan ataletsel ölçüm cihazı (IMU-Inertial Measurement Unit) çıkışının filtrelenmesinde kullanılmaktadır.

İvme verilerinde görülen hataların giderilmesi için kullanılan yapay sinir ağı sistemine, algılayıcı tarafından hareketsiz konumda (a=0, v=0, p=0) okunan ivme değerine eklenen bilinen bir sinyal (Şekil 3.16) ile yaratılan yapay hareket ivme verisi girilmiştir. Sistemi eğitmek için girilen "doğru" veri ise bizim sonradan eklediğimiz sinyalin tek olarak verilmesidir. Bu eğitim sonucu makul derecede iyi bir sonuç elde edilmesine mütakip eğitilmiş olan ağa, kontrol verisi girilerek çıkıştaki sonuç incelenmiştir. Burada, daha önce algılayıcıdan alınan 2000 değerlik ölçümün

ilk 1000'i ile oluşturulan kısım eğitim grubu, ikinci 1000'i ile oluşturulan kısım ise kontrol grubu olarak kullanılmıştır.

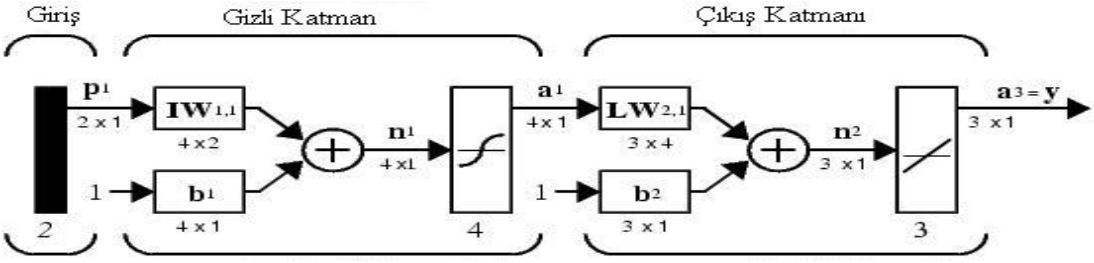


Şekil 3.16 : Gerçek (gürültüsüz) ivme verisi

İlk olarak ileri beslemeli ağ (feedforward network) mimarisinde değişik tipteki çok katmanlı perseptron (Multi Layer Perceptron-MLP) "geri yayılım (backpropagation) algoritmaları" (GYA'lar) denenmiştir.

3.3.1 Geri Yayılım Algoritmalı MLP Sinir Ağları

Daha önce Adaline mimarisi ile kullanılan LMS veya Delta kuralının geliştirilmiş ve çok katmanlı yapıya uyarlanmış hali olan geri yayılım algoritması, MLP yapısı üzerinde çeşitli problemlerin çözümünde kullanılmaktadır .Basitçe GYA, ağırlıkları hata türev (delta) vektörlerine göre atayan bir öğrenim kuralıdır. GYA hataları geriye doğru çıkıştan girişe azaltmaya çalışmasından dolayı geri yayılım ismini almıştır. Bu kural ayrıca "genelleştirilmiş delta kuralı" diye de anılmaktadır. İleri beslemeli ağlar çoğunlukla sigmoid nöronlar taşıyan bir veya daha fazla gizli katman ve lineer (doğrusal) nöronlara sahip çıkış katmanına sahiptir. Doğrusal olmayan etkinlik (activation) fonksiyonlarına sahip nöronlardan oluşan çoklu katmanlar sinir ağının, giriş ve çıkış vektörleri arasındaki doğrusal ve doğrusal olmayan ilşkileri öğrenmesini sağlar. Şekil 3.17'de en çok kullanılan MLP mimarilerinden biri gösterilmektedir.



Şekil. 3.17 : GYA'lı MLP mimarisi

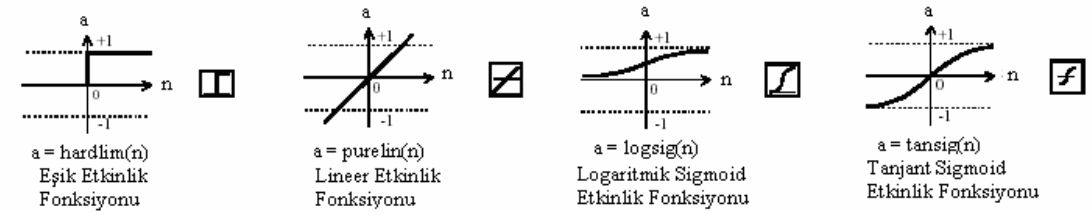
Burada p_1 giriş vektörü, gizli katmanda $w_{1.1}$ ağırlığı ile çarpıldıktan sonra b_1 yanlılık terimi (bias) ile toplanarak katmanın etkinlik fonksiyonu olan tanjant sigmoid fonksiyona (tansig) sokulur. a_1 gizli katmanın transfer fonksiyonunu göstermektedir.

$$a_1 = \tan sig\left(IW_{1.1}\ p_1 + b_1\right) \qquad \textbf{(3.16)}$$

Gizli katmandan çıkan a_1 değeri çıkış katmanında yeni bir $w_{2.1}$ ağırlığı ile çarpılarak b_2 yanlılık terimi ile toplanır ve çıkış katmanının doğrusal etkinlik fonksiyonundan (purelin) geçerek a_2 değerini alır ki bu, sinir ağının da çıkışı olması nedeniyle y olarak da adlandırılır.

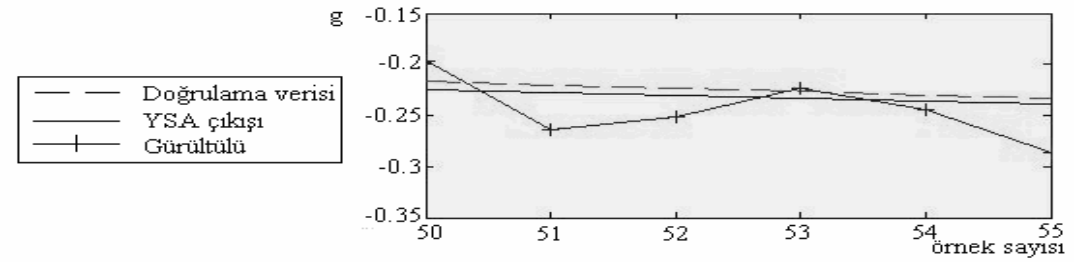
$$a_2 = purelin\left(LW_{2.1}\ a_1 + b_2\right) \qquad \textbf{(3.17)}$$

Nöronlardaki etkinlik fonksiyonları çok çeşitli yapılarda olabilirler. En çok kullanılan etkinlik fonksiyonları Şekil 3.18'de verilmiştir [25].



Şekil 3.18 : Farklı etkinlik fonksiyonları

Sinir ağlarının eğitimi için kullanılan çeşitli geri yayılım algoritmaları bulunmaktadır. Çalışmada sırasıyla Gradient Descent (GD), Gradient Descent with Momentum Backpropagation (GDM), Gradient Descent with Adaptive Learning Rate Backpropagation (GDX), Levenberg-Marquardt Backpropagation (LM) ve Resilient Back Propagation (RBP) eğitim yöntemlerini kullanan sinir ağı modelleri karşılaştırılmıştır. Modellerin hepsinde algılayıcıdan alınan aynı gürültülü ivme değeri dizisine aynı tip trapezoid sinyal eklenerek eğitim yapılmıştır. Bunların içinde Resilient Back Propagation ve ardından Levenberg-Marquardt algoritmaları en iyi sonuçları (gerçek sinyale en iyi yakınsamayı) vermişlerdir. Şekil 3.19'da verilen yakınlaştırılmış (0-500 arası olan değerlerde verilen farkın daha iyi görülebilmesi için 50-55 değerleri arası alınmıştır) grafikte görüldüğü üzere eğitim sonrası test verilerine göre yapılan filtrelemenin sonucunda Resilient Backpropagation algoritmasına sahip YSA çıkışının gürültülü sinyale göre oldukça iyi bir şekilde gerçek sinyale yakınsadığı görülmektedir.



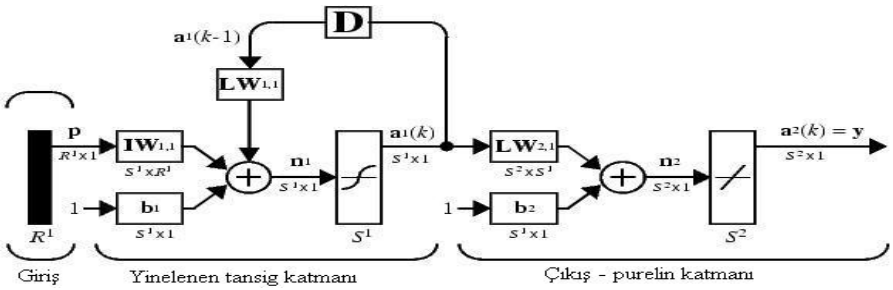
Şekil 3.19 : Resilient Backpropagation ile eğitim sonucu

En iyi sonucu vererek bundan sonra kullanılacak algoritma olarak seçilen Resilient Backpropagation, sigmoid transfer fonksiyonunun uç noktalarında oluşan küçük eğimlerin hesaplamalara kattığı olumsuz etkiyi gideren bir eğitim algoritmasıdır [26]. Bu algoritmanın en büyük avantajı ağırlıkların güncellenmesi sırasında hatanın türevinin yerine bu türevin işaretini kullanarak hatayı minimum yapacak ağırlık değerlerinden fazla uzaklaşmamasıdır. Bu yaklaşım sinir ağının performansını iyileştirmektedir.

Seçilen eğitim algoritmasına sahip YSAnın çıkışının daha da iyileştirilmesi için MLP kullanan farklı mimariler denenmiştir.

3.3.2 Elman Sinir Ağı

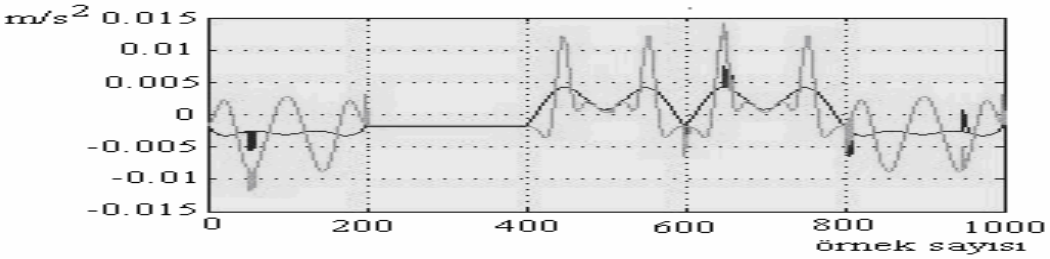
MLP'nin performansını arttırmak için kullanılabilecek yöntemlerden biri de Elman sinir ağıdır. Elman sinir ağı genel olarak iki katmanlı bir sinir ağı modelidir. Bu modelde birinci katman (gizli katman) çıkışından bu katmanın girişine geciktirilmiş bir geri besleme bulunmaktadır.Bu yinelenen bağlantı Elman sinir ağının zamanla değişen değerlerinin tespit edebilmesini sağlar. İki katmanlı böyle bir Elman sinir ağının yapısı şekil 3.20'de gösterilmektedir.



Şekil. 3.20 : Elman Ağının mimarisi

Şekil 3.20'de de görüldüğü gibi Elman sinir ağı gizli (yinelenen) katmanında tansig fonksiyonlu nöronlara, çıkış katmanında ise purelin fonksiyonlu nöronlara sahiptir. Bu kombinasyonun özelliği şudur ki bu etkinlik fonksiyonlarına sahip iki katmanlı

sinir ağları her fonksiyona (sonlu sayıda süreksizliğe sahip) yakınsayabilmektedirler. Buradaki tek gereksinim gizli katmanda yeterli sayıda nöron bulunmasıdır.



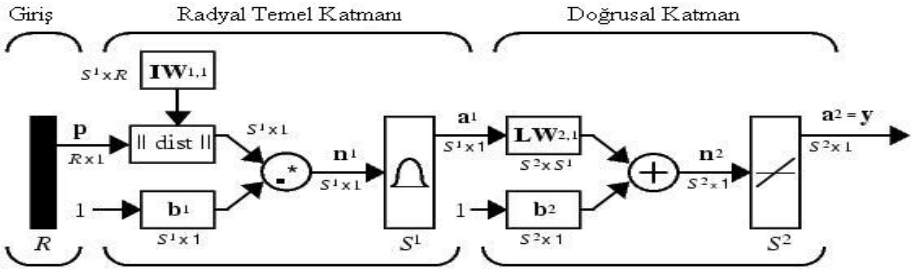
Şekil 3.21 : Elman ile MLP'nin karşılaştırılması (m/s^2)

Şekil 3.21'de Elman sinir ağı ile geleneksel GYA sinir ağının eğitimleri sonucunda hatalarının karşılaştırması verilmiştir. Koyu renk çizgi ile gösterilen Elman sinir ağının 1000 örneklik bir veri girişi için daha iyi sonuç verdiği görülmektedir.

Bundan sonra MLP ile karşılaştırılmak üzere farklı YSA mimarilerinin arayışına geçilmiş ve Radyal Temel (Radial Basis) Fonksiyonlu (RTF) sinir ağının incelenmesine karar verilmiştir.

3.3.3 Radyal Temel Fonksiyonlu Sinir Ağları

MLP sinir ağlarına alternatif olarak düşünülen RTF ağları ile eğitime geçmeden önce bu ağları biraz tanıyalım. RTF çok boyutlu uzayda eğri uydurma yaklaşımıdır. RTF ağlarının eğitimi çok boyutlu uzayda eğitim verilerine en uygun yüzeyi bulma problemidir [27]. RTF ağları standart MLP ağlarından daha fazla nörona ihtiyaç duyabilirler ama çoğunlukla eğitim süreleri standart MLP ağlarının eğitim sürelerinden daha kısadır.



Şekil 3.22 : RTF Algoritması

Şekil 3.22'de de gösterildiği gibi RTF'ler radyal temel ve doğrusal olmak üzere iki katmandan oluşur. Tipik bir RTF ağının 0 giriş için maksimum çıkışı 1 değeridir. Ağırlık vektörü (w) ile giriş vektörü (p) arasındaki uzaklık azaldıkça çıkış büyür. Buradan hareketle, radyal temelli bir nöronun p girişinin w ağırlık vektörüne özdeş

olduğunda 1 çıkışı veren bir yakınlık detektörü gibi çalıştığı söylenebilir. Yanlılık terimi *b* ise kullanılan "radbas" nörounun hassasiyetini oluşturur.

$$a_i 1 = radbas\left(\left\|_i IW_{1.1}\ \ p\right\| b_i 1\right) \tag{3.18}$$

$$a_2 = purelin\left(LW_{2.1}\ \ a_1 + b_1\right) \tag{3.19}$$

RTF ile MLP'yi karşılaştırdığımızda şu maddeler öne çıkar:

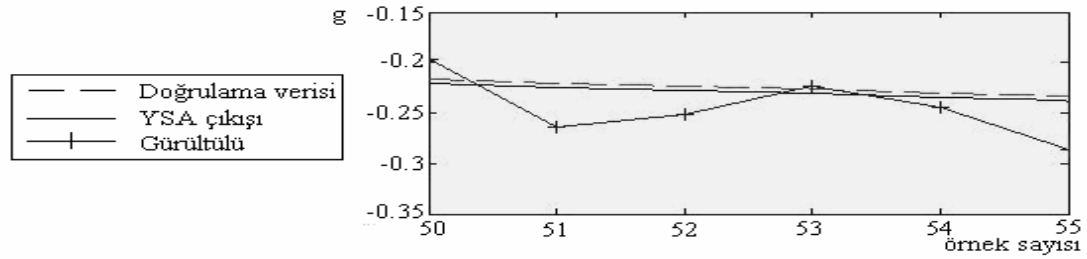
- RTFler tek gizli katmana sahipken MLPler çoklu gizli katmandan meydana gelebilmektedir.

- RTF katmanlarının tamamı genelde doğrusal olmayan yapıdayken, MLP'de sadece gizli katmanlar doğrusal olmayan yapıdadırlar; çıkış katmanları ise doğrusaldır.

- RTFlerde hesap düğümleri genel iken MLPlerin gizli ve çıkış katmanlarında temel farklılıklar gösterir.

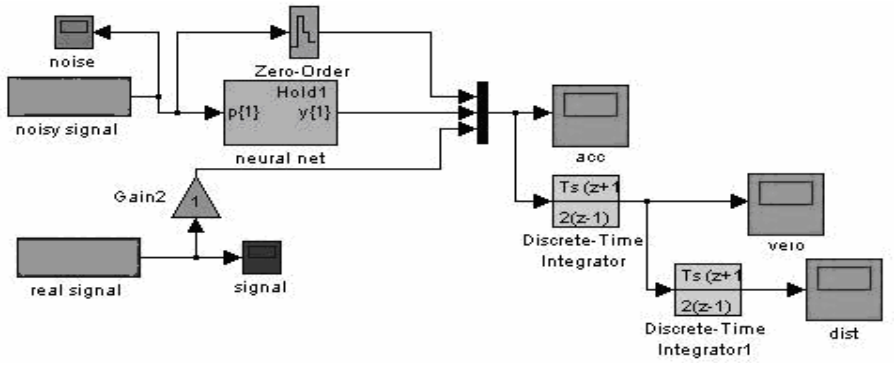
- RTFler evrensel yakınsamaya sahip oldukları için iyi ekstrapolasyonu (dış değer bulması) iyidir. Halbuki MLPlerde yakınsama yerel olduğu için hızlı öğrenir ama ekstrapolasyonda başarılı değildir.

Çalışmada Radial Basis Function Net (RTF), Exact Radial Basis Net (RBE) ve Generalized Regression Neural Net (GRNN) mimarileri eğitilerek karşılaştırılmıştır. Bu karşılaştırma sonucunda Generalized Regression Neural Network en iyi sonucu vermiştir (Şekil 3.23).



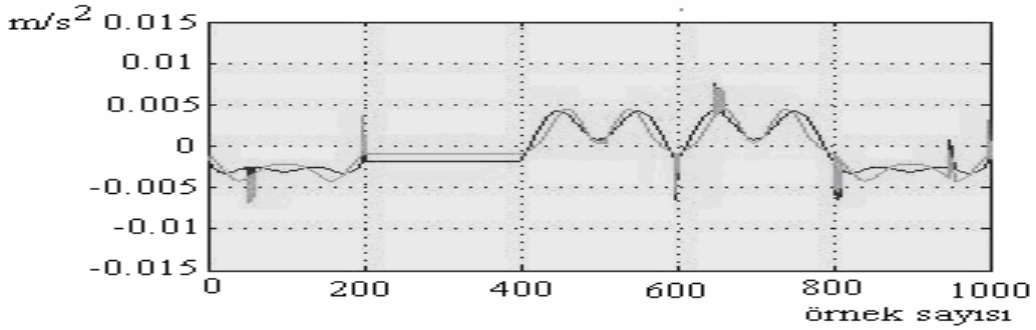
Şekil 3.23 : Generalized Regression algoritmasıyla yapılan eğitim sonucu

Bundan sonra en iyi sonuçları veren bu iki YSA modelinin (Elman ve RTF) birbirleriyle karşılaştırılmasının yapılması aşamasına geçilmiştir. Bunun için bu modeller filtre elemanı olarak Simulink'te hazırlanan konum takip modeline yerleştirilerek performansları karşılaştırılmıştır (Şekil 3.24).



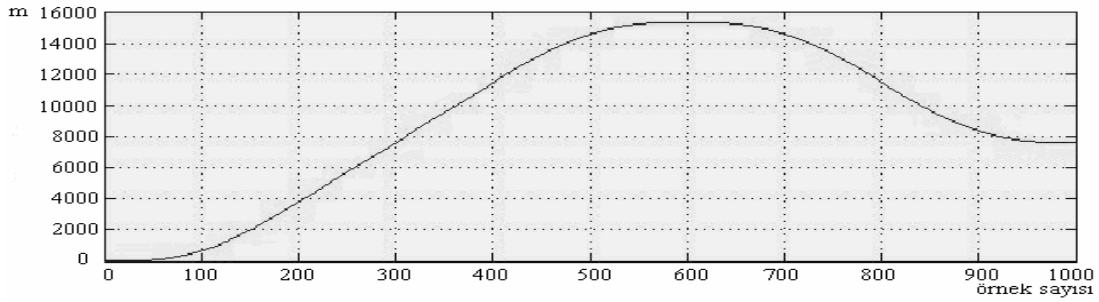
Şekil 3.24 : Simulinkte hazırlanan YSA filtreleme modeli

Bu modelde kendi hazırladığımız sinyal "gerçek" ivme verisini oluştururken, aynı sinyalin algılayıcının durağan halinde alınan çıkış verisiyle birleştirilmesiyle elde edilen "gürültülü" ivme verisi ise filtreye (YSA'ya) giriş sinyali olarak verilmiştir. Şekil 3.25'te Elman ve RTF sinir ağlarından çıkan ivme değerlerinin (m/s^2) hatalarının karşılaştırılması verilmiştir. Burada koyu renk çizgi Elman ağını gösterirken, açık renk çizgi de RTF ağının hata miktarını göstermektedir.



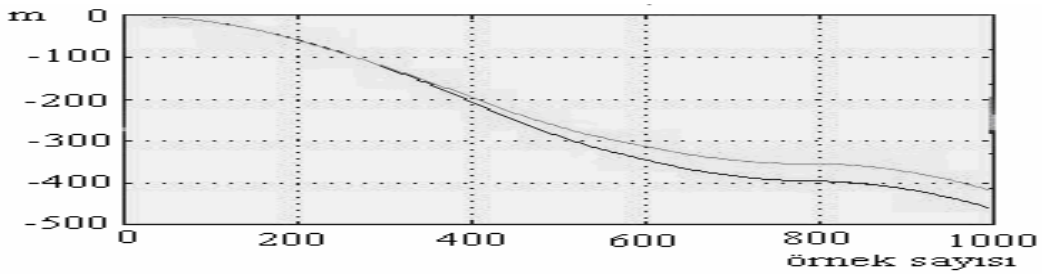
Şekil 3.25 : Elman ile RTF'nin ivme çıkış hatalarının karşılaştırılması.

"Gerçek" ivme verisinin iki kez entegrasyonu ile elde edilen konum değişimi (m) ise Şekil 3.26'da verilmektedir.



Şekil 3.26 : Gerçek konum değişimi verisi

YSA çıkışlarının ikişer defa entegrasyonu ile elde edilen konum değişimi verilerindeki hata miktarlarının karşılaştırması ise Şekil 3.27'de verilmiştir. Yine burada da koyu renk çizgi Elman ağını gösterirken, açık renk çizgi de RTF ağının hata miktarını göstermektedir.



Şekil 3.27 : Elman ile RTF'nin konum bilgilerinin karşılaştırılması

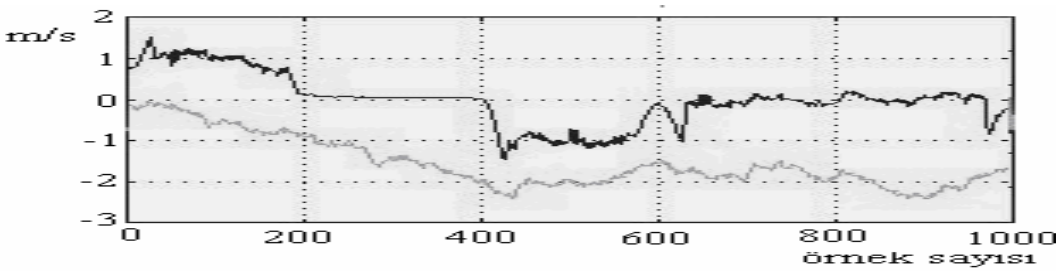
Sistemimiz aracın konum değişiminin mümkün olan en iyi şekilde takibi üzerine olduğu için burada konum değişimindeki hata miktarlarını göz önüne almak gerekmektedir. Şekil 3.27'nin tekrar incelenmesiyle de görüleceği gibi RTF sinir ağı GYA sinir ağlarından daha iyi sonuçlar vermektedir. RTF, gerek bu karşılaştırmanın sonucu gerekse de öğrenme hızı göz önüne alındığında sistemimiz için uygun YSA modeli olarak seçilmiştir.

3.3.4 Farklı Aşamalarda Filtreleme

Uygun YSA algoritmasının seçiminin ardından gerçek konum bilgisine en iyi yakınsamayı sağlamak için farklı yöntemler aranmaya başlanmıştır. Bunların ilki filtreleme işlemini ivme verisinde değil de hız verisinde uygulamaktır.

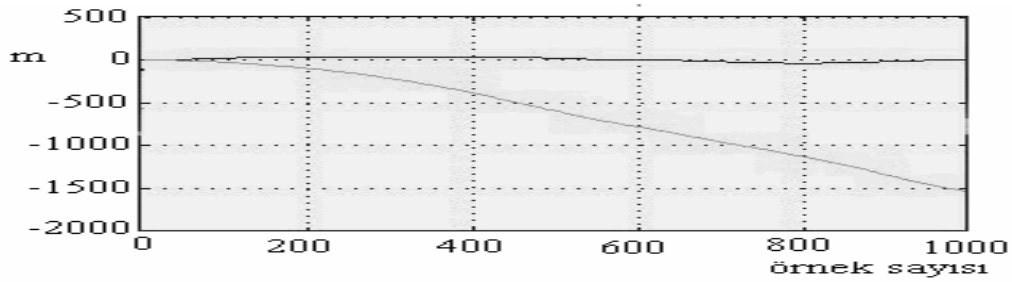
Hız verisinin filtrelenmesi :

İvme verisinde filtreleme sırasında oluşabilecek kayıpların konum değişimi verisine etkisini incelemek için filtreleme işlemi bu adımda ivme yerine elimizdeki gürültülü sinyali bir defa entegre ederek elde ettiğimiz hız verisine uygulanmıştır. (Bu işleme ikinci aşama filtreleme ismi verilmiştir.) Burada YSA'ya referans oluşturacak "gerçek hız" verisi olarak da hazırladığımız, bilinen ivme verisinin bir defa entegre edilmiş hali kullanılmaktadır. Şekil 3.28'te verilen grafikte gürültülü hız verisi ile YSA çıkışının hatalarının karşılaştırması gösterilmiştir. Burada koyu renk ile gösterilen YSA çıkışında hatanın azaltıldığı görülmektedir.



Şekil 3.28 : Gürültülü ve filtrelenmiş hız verilerinin hataları

Hız verilerinin entegrasyonu ile elde edilen konum değişimi verileri arasındaki hata karşılaştırması Şekil 3.29'da gösterilmektedir.

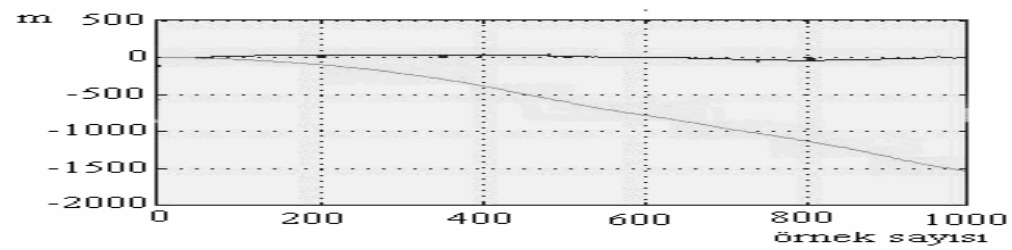


Şekil 3.29 : Konum bilgisi olarak filtrelenmiş ve gürültülü sinyallerin hataları

Yine koyu renkle gösterilen ve YSA çıkışından elde edilen konum değişimi verisinden anlaşıldığı üzere hız verisinin filtrelenmesi ile konum değişimindeki hata büyük oranda giderilebilmektedir.

Konum verisinin filtrelenmesi :

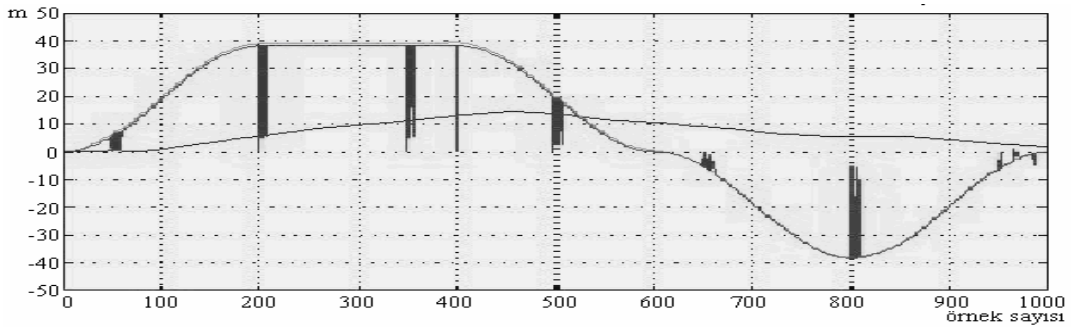
Konum verisinin filtrelenmesinde (Bu işleme ise üçüncü aşama filtreleme ismi verilmiştir.) gürültülü ve gerçek ivme verilerinin ikişer kez entegrasyonu ile elde edilen gürültülü ve gerçek konum değişimi verilerinin kullanımıyla eğitilen yapay sinir ağının çıkışının hatası ile gürültülü verinin hataları karşılaştırıldığında Şekil 3.30'da de görüldüğü gibi YSA'nın (koyu renk) büyük oranda fitrelemeyi gerçekleştirdiği görülmektedir.



Şekil 3.30 : Gürültülü ve filtrelenmiş konum bilgilerinin hataları

Farklı aşamalarda yapılan filtrelemelerin karşılaştırılması :

Her üç kademede yapılan filtrelemelerde YSA modellerinin gürültüyü büyük oranda giderdikleri görülmüştür. Sırada bu üç uygulamanın birbirleriyle karşılaştırılması vardır ki bu sayede algılayıcı çıkışına hangi aşamada müdahele edeceğimiz belli olacaktır. Şekil 3.31'de neredeyse üstüste binmiş gibi duran iki hata verisi 2. ve 3. aşama filtrelemelerin sonuçlarıdır. Burada daha koyu olan 3. aşama filtreleme hatasında bazı bölgelerde kararsızlıklar bulunmasına rağmen eğrinin 2. aşama filtrelemeden değer olarak daha düşük olduğu görülmektedir. Yine de iki filtreleme de ivme verisine uygulanan filtrelemeden kötü sonuçlar vermişlerdir.



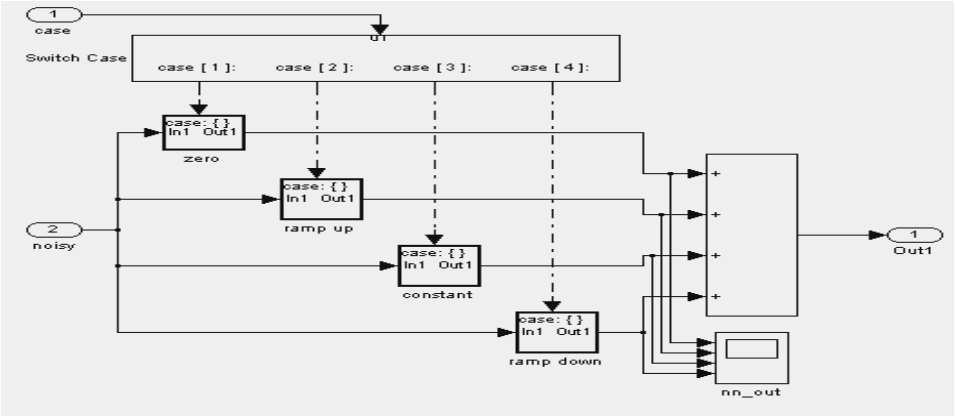
Şekil 3.31 : İvme, hız ve konum bilgilerinin filtrelenmesiyle elde edilen konum verilerinin hataları

Bu sonuçların ışığında filtreleme işleminin doğrudan ivme değerine yapımasında karar kılınmıştır. Fakat yapılan bu filtreleme işlemlerinde eğitim ve kontrol için kullanılan sinyallerin karakteristiklerinin benzer olması bu YSA modellerinin sadece verilen tipteki sinyalleri öğrenmesini sağlamaktadır. Eğer aracın hareketi sırasında YSA'ya eğitilen veri dizisinden farklı bir yapıda bir ivmelenme gerçekleşirse ne olur? Böyle bir durumda sistem büyük oranda bu farklı ivme verisini filtreleyemeyecektir. Bu sorunu çözmenin yolu Parçalı Yapay Sinir Ağları (PYSA-Piecewise Neural Network) kullanmaktır.

3.3.5 Parçalı Yapay Sinir Ağları ile Filtreleme

Parçalı yapay sinir ağları basitçe farklı eğitilmiş YSA'lardan oluşan birer topluluktur. Bu çalışmada PYSA'yı oluşturan her bir YSA farklı karakteristikteki bir ivme girişine göre eğitilmiştir. PYSA'nın yapısında bulunan bir anahtarlama düzeneğiyle gelen ivme sinyali karakteristiğine uygun eğitilmiş YSA'ya yönlendirilmekte, daha sonra bu YSA'ların çıkışları tekrar birleştirilerek en iyi şekilde filtrelemenin

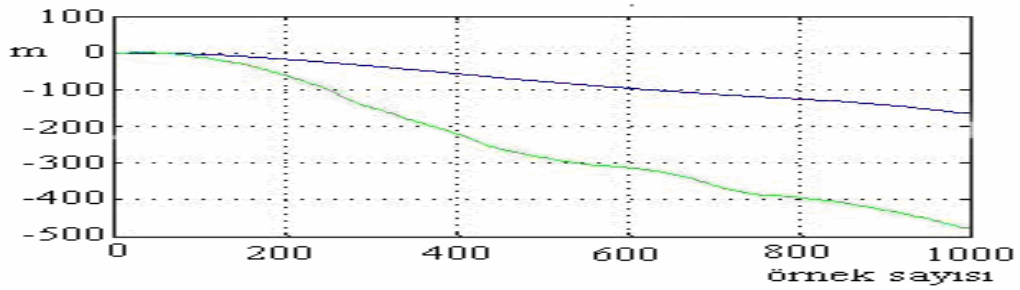
gerçekleşmesi sağlanmaktadır. Şekil 3.32'te PYSA'nın anahtarlama düzeneği gösterilmektedir.



Şekil 3.32 : Parçalı filtrelemede RTF ağları arası anahtarlama düzeneği

Burada PYSA'yı oluşturan alt sinir ağları da görülmektedir. Bu YSA'lar sırasıyla ivme verisinin "sıfır" girişi, "yükselen rampa" formu, "sıfırdan farklı sabit değer"girişi ve "alçalan rampa" formundaki giriş için eğitilmiş 4 farklı sinir ağıdır.

Şekil 3.33'te bu yeni yapıdaki filtreleme ünitesinin çıkışında oluşan hata miktarı (koyu renk) gösterilmektedir. Şekil 3.33'te görülen ikinci eğri ise filtrelenmemiş gürültülü sinyalin hata miktarını vermektedir.

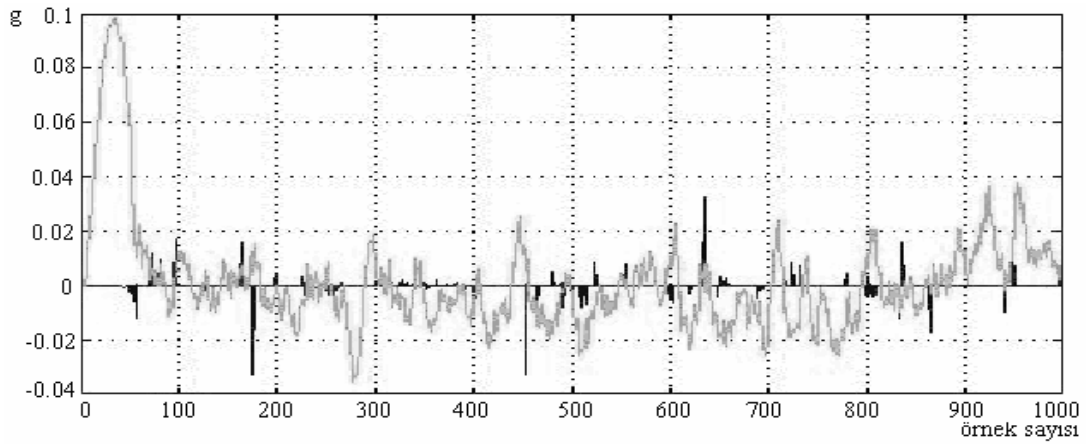


Şekil 3.33 : Parçalı RTF ağının (koyu veri filtre çıkışı) ve gürültülü sinyalin konum değişimi hataları

Çalışmanın bu bölümünde yapay sinir ağlarının filtrelemede kullanılabilirliği incelenmiştir. Elde edilen olumlu sonuçlar yapay sinir ağlarının sinyal işlemede kullanılabileceğini göstermektedir. Son olarak tasarlanan PYSA'nın modüler yapısı ve farklı giriş dizilerine kolaylıkla uyum sağlayabilmesi yapay sinir ağlarının filtreleme alanındaki kullanışlılığını göstermektedir.

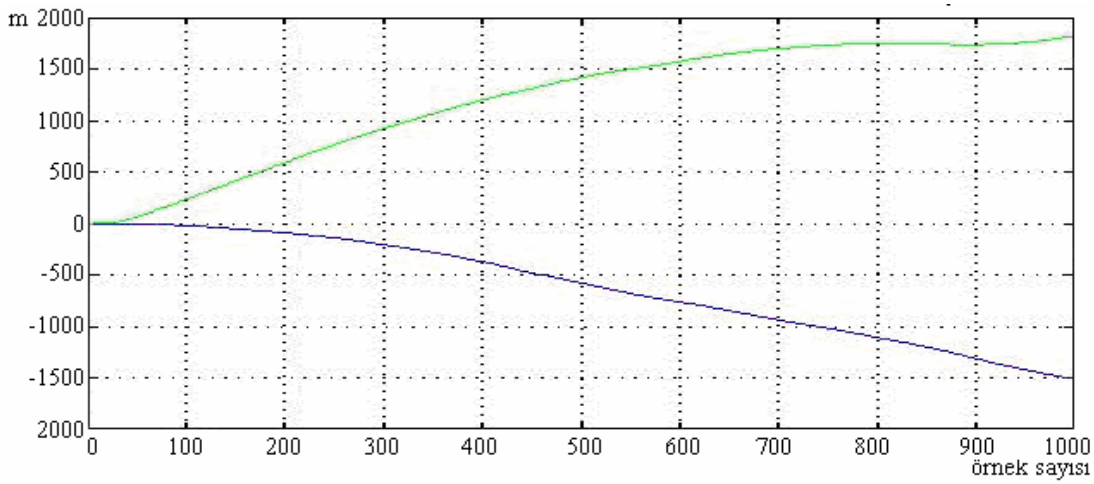
3.4 LMS ile YSA Sonuçlarının Karşılaştırılması

Son olarak LMS algoritmasıyla parçalı YSA modelinin karşılaştırması yapılmıştır. İki filtreleme yöntemi de aynı koşullar altında, aynı sinyal girişi ile test edilmiştir. Buna göre filtrelerin çıkışında oluşan ivme verilerinin gerçek ivme verisinden farkları Şekil 3.34'te verildiği gibidir. Burada açık renk eğri LMS'e aitken koyu renk olan YSA filtresinin hatasıdır.



Şekil 3.34 : LMS ile YSA'nın ivme verilerinin gerçek ivme verisinden farkları

Şekil 3.34'ten de görüldüğü gibi LMS filtresi ilk 100 değer civarında öğrenmeye yeni başladığı için hatası büyük çıkmaktadır. Ondan sonra belli bir ölçüde toparlasa da hatasını YSA kadar düşük seviyelere indirememektedir. Filtre çıkışlarındaki bu hatalardan konum değişimine geçecek olursak karşımıza Şekil 3.35'te verilen sonuç çıkar.



Şekil 3.35 : LMS ile YSA'nın konum verilerinin hataları

Burada yine açık renk eğri LMS verisini göstermektedir. Koyu renkteki YSA konum hatasıyla karşılaştırdığımızda hatalar ters yönlü olmasını gözardı edersek anlık değerlerde LMS'in hatasının devamlı YSA'dan fazla olduğu görülmektedir. Sonuç olarak yörünge takibi için algılayıcıdan gelen gürültülü verilerin filtrelenmesinde YSA'nın daha uygun olduğu kararına varılmıştır.

4. KONUM TAKİP SİMÜLASYONU

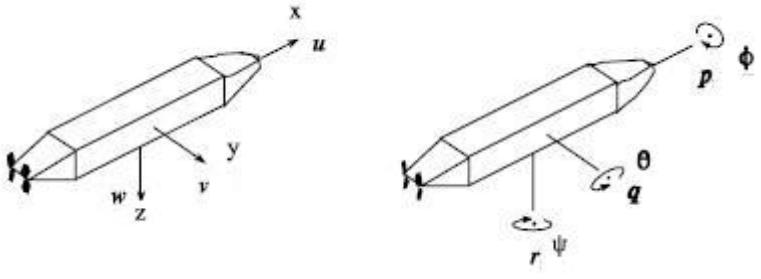
İvme verilerinin filtrelenmesinde kullanılacak yöntem belirlendikten sonra bir üst aşamaya, bu verilerle aracın takibinin gerçekleştirilmesi işlemine geçilmiştir. Şu ana kadar tek eksen üzerinde yapılan çalışma artık sualtı aracının hareket edeceği üç eksene göre düzenlenecek; ardından sualtı aracının yörünge takibinin bilgisayar ortamında gözlemlenebilmesi için kullanılacak üç boyutlu grafik arayüz programı tanıtılacaktır.

4.1 Üç Eksende Hareket Takibi

Tüm deniz araçları gibi sualtı araçları da 6 serbestlik derecesine sahiptir; ancak yüzey gemilerinde Z eksenindeki doğrusal hareket ve tüm eksen takımları etrafında gerçekleşen dönme hareketleri istem dışı gerçekleşirken sualtı araçları bu hareketleri bilinçli olarak gerçekleştirebilmektedir.

4.1.1 Deniz Araçlarının Hareketleri

Şekil 4.1'de bir denizaltının üç eksen doğrultusunda yaptığı doğrusal ve açısal hareketler gösterilmiştir.



Şekil 4.1 : Sualtı aracının yaptığı doğrusal ve açısal hareketler

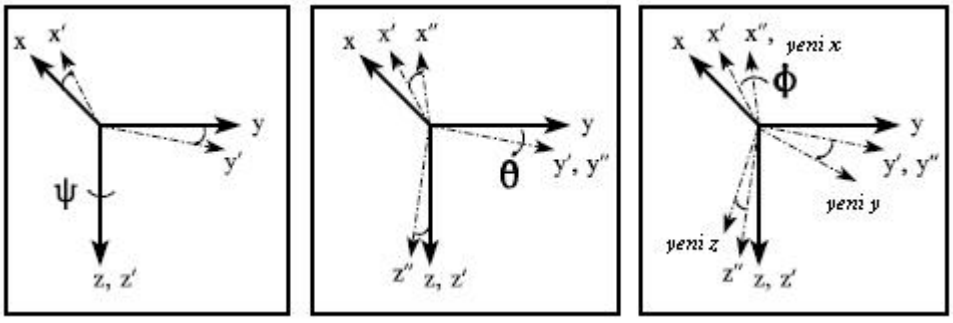
Bir deniz aracının sabit eksen takımında X ekseni doğrultusundaki hareketine boy-öteleme (surge), Y ekseni boyunca olana yan-sürüklenme (sway) ve Z ekseni boyunca olana da dalıp-çıkma (heave) denir. Bu doğrusal hareketler dışında

genellikle aracın dengesinden kaynaklanan dönme hareketleri gerçekleşir. Deniz aracının X ekseni etrafında yaptığı dönme hareketine yalpa (roll), Y ekseni etrafındakine baş-kıç vurma (pitch), Z ekseni etrafındaki dönmeye ise savrulma (yaw) denir. [28]

Aracın öteleme hızı *u*, sürüklenme hızı *v*, dalıp-çıkma hızı da *w* ile sembolize edilir. Açısal hızlar ise sırasıyla yalpa *p*, baş-kıç vurma *q*, savrulma ise *r* ile gösterilir. Aracın doğrusal ve açısal hız vektörleri aracın kendi sabit eksen takımına göre tanımlanırken, aracın konum ve yönelimi dünya eksen takımına göre tanımlanır. Bunlar, X ekseni doğrultusu Kuzey, Y ekseni doğrultusu Doğu ve Z ekseni doğrultusu Derinlik'tir. Aracın eksenler etrafında yaptığı Euler açıları ise sırasıyla yalpa için Φ, baş-kıç vurma hareketi için θ, savrulma hareketi içinse ψ'dir.

4.1.2 Eksen Takımları

Eksen takımlarının doğru bilinmesi özellikle yapılan bu tip çalışmalarda çok önemlidir. Zira algılayıcı, ölçümleri kendi eksen takımına göre hesaplamaktadır. Eğer araç üzerine yanlış yerleştirilirse örneğin araç dalarken biz onun ileri gittiğini düşünebilir, bu da yanlış ölçümden daha vahim olarak aracın kaybı gibi sonuçlar doğurabilir. Sorun sadece algılayıcı ile aracın eksen takımlarının doğru olarak çakıştırılmasıyla çözülemez. Bu durumda da aracın eksen takımının dünya eksen takımına göre ne kadar değiştiğinin bilinmesi gerekir.



Şekil 4.2 : Eksenler etrafında dönüşler

Şekil 4.2'de verilen örnekte [29] olduğu gibi aracı önce z ekseninde ψ kadar döndürdüğümüzü kabul edelim. Bu durumda yeni eksen takımımız x^{I}, y^{I}, z^{I} olur. Daha sonra y^{I} ekseni etrafında θ açılı bir dönüş yaptığımızda eksen takımımız bu sefer de x^{II}, y^{II}, z^{II} olur. Son olarak x^{II} ekseninde Φ kadar bir dönüş yaparak hareketimizi tamamladığımızda aracımızın burnu gerçek x (kuzey) doğrultusu yerine

tamamen farklı bir noktayı göstermektedir. Bu açısal değişimleri (yönelimi) hiç dikkate almadan sadece ivme bilgisinden faydalanıp aracın konum değişimini gözlemlemeye kalkarsak aracın ne kadar yol aldığını bilebiliriz ancak nerede olduğunu kesinlikle bilemeyiz. Bu nedenle eksen takımları dönüşümlerine dikkat etmemiz gerekmektedir.

Eksen takımları çevrimine geçmeden önce bilmemiz gereken en önemli husus Euler açılarının kısıtlamalarıdır [29]:

$$-\pi < \phi \leq \pi \tag{4.1}$$

$$-\frac{\pi}{2} < \theta < \frac{\pi}{2} \tag{4.2}$$

$$0 \leq \psi < 2\pi \tag{4.3}$$

Bu kısıtlamalar göz önüne alınarak dünya eksen takımına göre Euler açılarının dönüş hızları sabit eksen takımı açısal hızlarının bağıntılısı olarak şu şekilde verilir [29]:

$$\dot{\phi} = p + q\sin(\phi)\tan(\theta) + r\cos(\phi)\tan(\theta)$$

$$\dot{\theta} = q\cos(\phi) - r\sin(\phi) \tag{4.4}$$

$$\dot{\psi} = \frac{q\sin(\phi) + r\cos(\phi)}{\cos(\theta)} \tag{4.5}$$

$$\tag{4.6}$$

Eksen takımı çevrimleri matris formatında ve "direction cosines" kullanılarak verildiğinde bilgisayar açısından da daha kolay bir hesap işlemi gerçekleşmektedir.

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.7}$$

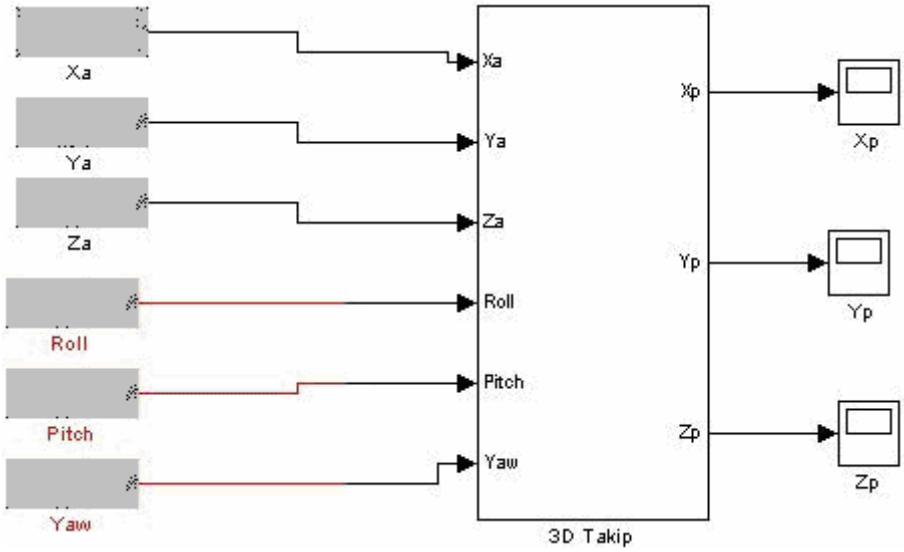
Burada X, Y, Z sabit eksen takımına ait herhangi bir vektörü (ivme, hız, kuvvet,...) temsil etmektedir. X_W, Y_W, Z_W ise dünya eksen takıma ait vektörlerdir. "Direction cosines" değerlerini Euler açıları cinsinden yazarsak [30];

$$
\begin{array}{lll}
a_1=\cos\theta\cos\psi & b_1=\sin\Phi\sin\theta\cos\psi-\cos\Phi\sin\psi & c_1=\cos\Phi\sin\theta\cos\psi+\sin\Phi\sin\psi \\
a_2=\cos\theta\sin\psi & b_2=\sin\Phi\sin\theta\sin\psi-\cos\Phi\cos\psi & c_2=\cos\Phi\sin\theta\sin\psi-\sin\Phi\cos\psi \\
a_3=-\sin\theta & b_3=\sin\Phi\cos\Phi & c_3=\cos\Phi\cos\theta
\end{array} \qquad \textbf{(4.8)}
$$

Bu denklemler yardımıyla algılayıcının kendi eksen takımına göre ölçtüğü ivme değerleri ile yine algılayıcı ünitenin içinde bulunan jiroskoplar tarafından ölçülen açısal bilgileri kullanarak sualtı aracımızın dünya eksen takımındaki gerçek hareketlerini gözlemleyebiliriz.

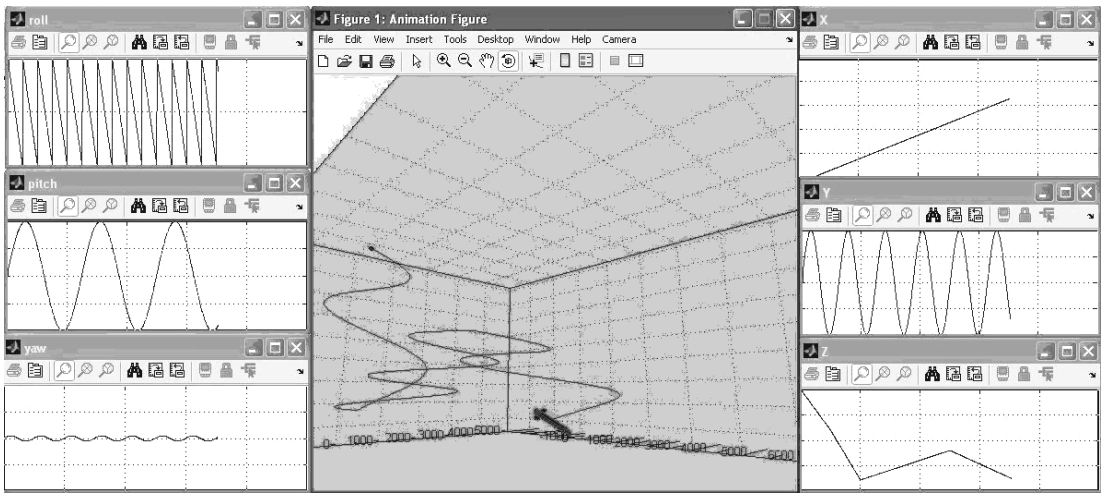
4.2 Simulink Arayüzü ile Oluşturulan Program

Üç eksenli algılayıcı ünitenin içinde bulunan ve ünitenin kendi eksen takımına göre x, y, z eksenlerinde ayrı ayrı ölçümler yapan ivmeölçerler her ne kadar benzer olsalarda tamamen aynı ölçümleri yapamazlar. Bu nedenle her bir ivmeölçer için ayrı olarak hazırlanmış toplam 3 filtreden oluşan üç eksenli filtreleme sistemini, jiroskoplardan açısal değer okuyarak bunları eksen takımı çevrimlerinde kullanan diğer bir sistem ile birleştirerek Simulink programında üç eksenli konum takip modeli oluşturulmuştur (Şekil 4.3).



Şekil 4.3 : Üç eksenli Simulink takip modeli

Ayrıca yine Simulink programı ile hazırlanan 3 boyutlu grafik animasyon da takip modeline eklenerek daha gerçekçi ve kullanışlı bir simülasyon elde edilmiştir.



Şekil 4.4 : Animasyonlu konum takibi

Şekil 4.4'de, hazırlanan modelin örnek bir çalışma için animasyonlu gösterimi verilmiştir. Burada ortadaki büyük çerçevede aracın hareketi 3 boyutlu olarak izlenebilirken sağ taraftaki küçük pencerelerden aracın hangi eksende ne kadar yol gittiği, sol taraftaki pencerelerden ise aracın 3 eksende yaptığı açısal hareketler gözlenebilmektedir.

5. SUALTI ARACI

5.1 İnsansız Sualtı Araçları

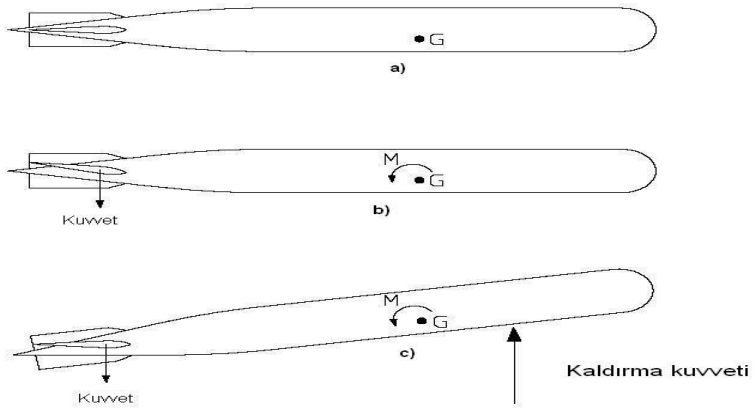
Bu tür bir çalışmada kullanılacak insansız sualtı aracının (UUV – Unmanned Underwater Vehicle) seçiminde dikkatli davranılması gerekir. Bu alanda çok çeşitli tasarımlar mevcut olmasına rağmen hepsinin yapımında ortak bir özellik bulunmaktadır :Güvenlik!

Güvenlik konusunda, her ne kadar aracın kullanımı sırasında çevreye zarar gelmemesine dikkat edilmesi gerekmekte ise de, değinilmesi gereken asıl konu aracın kendi iç güvenlidir. Bir kere araç hiçbir şekilde su geçirmeyecek şekilde imal edilmelidir. Aksi takdirde içinde bulunan tüm elektronik cihazlar bozulabilir. Su geçirmezliğin sağlanması konusunda aracın kullanılması düşünülen ortamın da büyük önemi bulunmaktadır. Derin sularda çalışılması düşünülüyorsa gövde imalatında ve bağlantı noktalarında ona göre basınca dayanıklı malzeme kullanılması gerekir.

Hobi amaçlı üretilen veya düşük maliyetli olması istenen sualtı araçlarının büyük bir çoğunluğu plexiglas, epoksi ve pvc türü malzemelerden üretilmektedir. Bu malzemeler hem ucuz olmaları hem de kalınlıklarıyla oynanarak farklı basınçlara dayanabilecek ürünler imal edilmesine olanak sağladıkları için çok tercih edilmektedir. Daha profesyonel çalışmalarda ise kompozit malzemeler, alüminyum türevi hafif metaller kullanılmaktadır. Malzeme seçimindeki diğer bir husus şekil verebilmedir. Malzemenin istenilen formu rahatça alabilmesi, kırılgan olmaması hem aracın ömrünü uzatır hem de aracın performansını etkiler.Burada araç imalatında dikkat edilmesi gereken ayrı bir konu karşımıza çıkmaktadır: Performans.

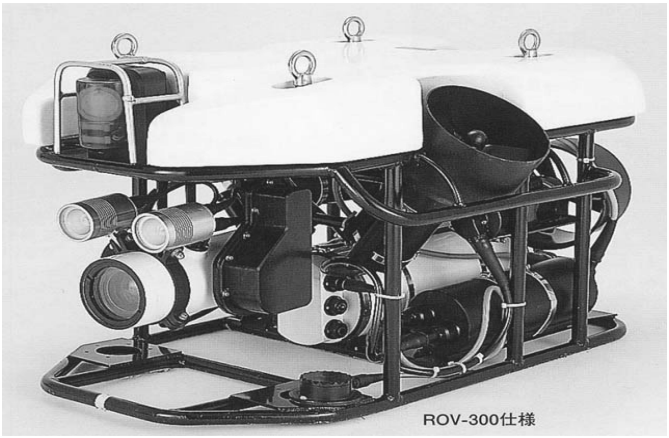
Performans ile kastedilen aracın hızlı gitmesi değil, iyi bir denizciliğe sahip olmasıdır. Zaten hız her zaman tercih edilen bir özellik değildir. Aracın istenilen görevi başarıyla yerine getirmesi, manevra kabiliyetinin iyi olması ve verilen emirlere çabuk cevap verebilmesi önemlidir. Bunların sağlanabilmesi için başta

aracın fazla dirence maruz kalmasına neden olmayacak hidrodinamik bir forma sahip olması garekir. Böylece aracın belirli bir yolu alması için harcayacağı enerji azaltılmış olur, yakıt sarfiyatı sağlanır ve motor ömrü korunmuş olur. Genelde kıçtan itki sağlayan araç dümen ve farklı yerlere konuşlandırılabilen kontrol yüzeyleri ile manevra yaparlar. Kıç kontrol yüzeylerinin, ağırlık merkezine olan uzaklıklarından dolayı denizaltının dönmesine büyük etkileri vardır. Bu kanatları Şekil 5.1b'deki gibi pozisyonladığımızda aşağıya doğru bir itme kuvveti oluşturacaklardır. Bu, sualtı aracının başının yükselmesine neden olacak bir moment (M) oluşturur. Sualtı aracı bir kere yukarıya doğru bir açı yaptı mı denizaltının gövdesi yukarıya doğru bir itme kuvveti oluşturur. Bunun sonucunda araç yukarı doğru bir açıyla yükselir. [31]



Şekil 5.1 : Kontrol yüzeylerinin etkisi

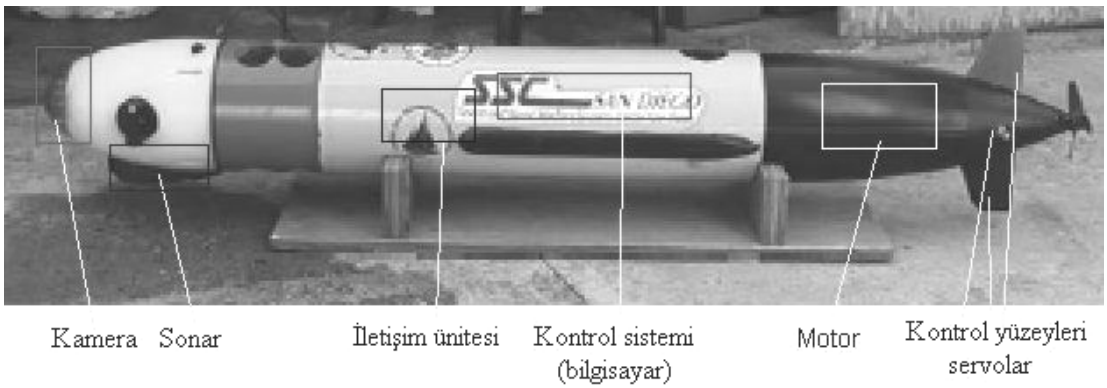
Bunun dışında daha çok uzaktan kumanda edilen sualtı araçlarında (ROV – Remotely Operated Vehicle) tercih edilen kutu biçimli bir form ve çoklu motora sahip tasarımlar da bulumaktadır (Şekil 5.2).



Şekil 5.2 : Ticari bir ROV modeli

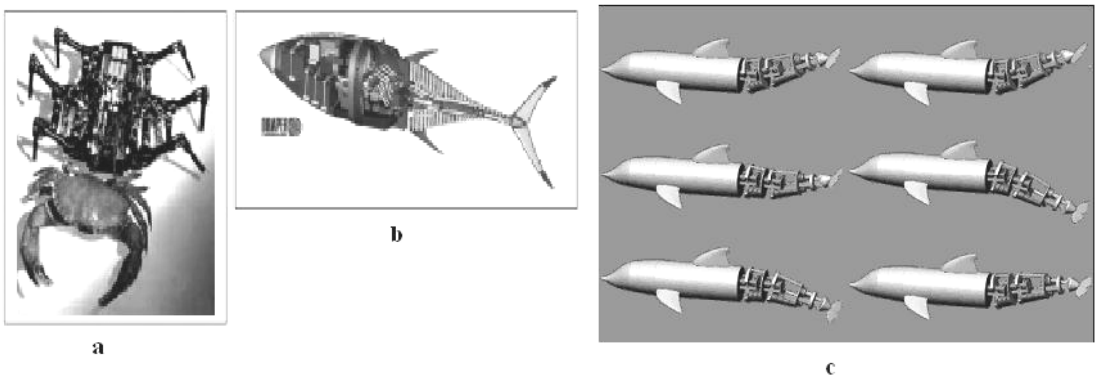
İlk olarak derin deniz araştırmalarında kullanılmaya başlayan ROV'lara olan ilgi gün geçtikçe artmaktadır. Değişik yönlerde yerleştirilmiş çoklu motorları sayesinde yüksek manevra kabiliyetine sahip bu araçların klasik tip sualtı araçlarına göre üstün yanları motorlarının sayı ve yerleştirilişlerine bağlı olarak istenildiğinde hız almaya ihtiyaç duymadan bir çok manevrayı kolaylıkla gerçekleştirebilimeleri, gelişmiş modellerde takılan robot kollar yardımıyla dalgıçların çalışamadıkları ortamlarda uzaktan kumanda ile gerek gözlem, gerekse tamir veya örnek toplama gibi işleri yapabilmeleridir. Ünlü Titanik gemisinin enkazını bulan da Argo isimli bir ROV'dur. Ancak ROV'ların da bazı olumsuz yönleri vardır. Bunların başında araç ile kullanıcısı arasındaki bağlantının kablolarla sağlanıyor olması gelir. Gerekli sinyalleri derinlere taşıması için oldukça kalın olan bu kabloların hem maliyeti yüksektir, hem de aracın gideceği mesafeye göre artan kablo boyu sorun teşkil eder. Aracın kumandası dahil bütün verilerin taşındığı bu kabloların deniz altından bir kayaya takılması, aracın pervanesine dolanması gibi durumlar kablonun kopmasına ve ROV'un denizin derinliklerinde kaybolmasına yol açabilir. Her ne kadar araç ile kullanıcı arasındaki bağlantının radyo sinyalleri gerçekleştirilebileceği düşünülse de radyo dalgalarının su içine fazla nüfuz edememesi nedeniyle bu ancak sığ sularda gerçekleşebilir. Sualtında kullanılabilen yegane iletişim şekli ses dalgalarıyladır. Ancak dar frekans bandı nedeniyle ses ile veri aktarımı da kısıtlıdır.

Bu noktada devreye başka bir sualtı aracı konsepti olan otonom araçlar (AUV – Autonomous Underwater Vehicle) girmektedir. AUV'ler içlerinde bulunan işletim sistemi ve algılayıcı ünitelerin yardımıyla dışarıdan her hangi bir yardım almadan kendi denge, derinlik ve konum kontrollerini yapabilen ve önceden verilmiş görevi yerine getirmek üzere programlanmış sualtı robotlarıdır. Günümüzde daha çok sonar yardımıyla derin deniz dibi taraması gibi bilimsel çalışmalarda kullanılar AUV'ler için önümüzdeki yıllara yönelik çok daha stratejik görevler öngörülmektedir. Bir AUV'nin yapısal elemanları basitçe şekil 5.3'te gösterilmiştir.



Şekil 5.3 : Bir AUV örneği

AUV'lerin en uç noktası ise çeşitli alanlarda türevlerini gördüğümüz, doğadaki canlıları taklit eden biyomimetik olarak adlandırılan robotların denizlerle ilgili üyeleridir. Şekil 5.4-a, b ve c'de değişik su canlılarından esinlenilerek yapılan robotlar gözükmektedir.



Şekil 5.4 : a) Robot Yengeç, b) Robot Balık, c) Robot Yunus

Şekil 5.4 a'da gösterilen robot yengeç sualtında kablo ve boru hatlarının tamirinde, batık araştırmalarında, diğer araçların zarar vermeden giremeyeceği yerlerin araştırılmasında ve yapılması planlanan madencilik çalışmalarında kullanılmak üzere tasarlanmıştır. Şekil 5.4 b'de görülen robot balık ise Geleneksel sualtı araçlarından farklı olarak gerçek balıklar gibi kuyruk hareketiyle yüzmektedir. Şekil 5.4 c'de görülen prototip ise bir adım daha ileri giderek bir yunusun yüzüş stilini örneklemektedir. İstanbul Teknik Üniversitesi Makina Fakültesi tarafından geliştirilmekte olan aracın balık benzerlerinden en büyük farkı bir memeli olan yunus gibi kuyruk bölümü yatay değil, dikey hareket ettirmesidir. Aracın pervaneli sualtı araçlarına göre üstünlükleri arasında sessiz çalışması ve suda iz bırakmaması özellikle belirtilmiştir [32].

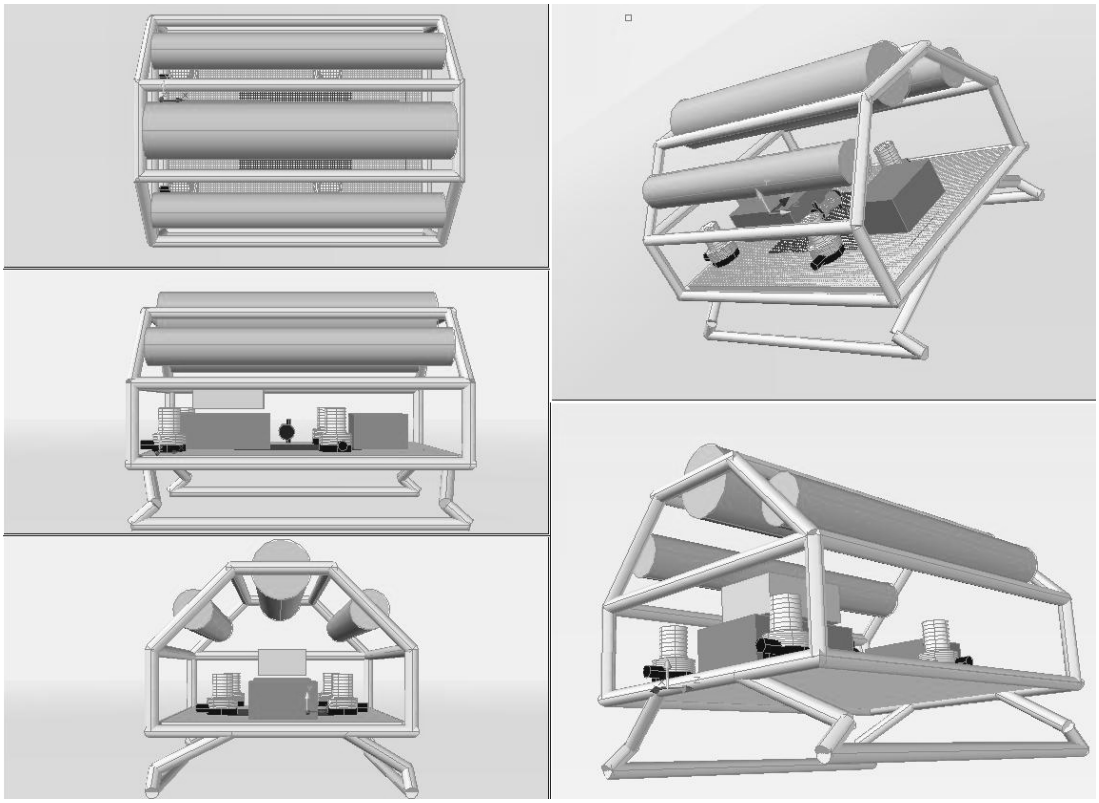
5.2 Gerçeklenen Sualtı Aracı

Bu bölümde, yapılan tez çalışmasının bir üst aşaması olan ve tasarlanan konum takip yönteminin gerçeklenmesinde kullanılacak sualtı aracı tasarımı, yapısı ve kontrolü açısından tanıtılmaktadır.

Araç tasarlanırken öncelikle boyutlarının iyi belirlenmesi gerekir. Gereğinden büyük bir araç hem hantal olacak, hem de maliyetli olacaktır. Fazla küçük bir araç kullanılması halinde ise bazı gerekli parçaların araç içine sığmama ihtimali vardır. Bunun için de aracı boyutlandırmadan önce içinde bulunması öngörülen elemanların boyutlarına ve yerleştirilişlerine dikkat edilmelidir. İlk başta kullanılması düşünülen araç, gövdesi en az iki parçadan oluşan ve konum takibinde kullanılması düşünülen IMU ve işlemci (istenilirse IMU verileri işlenmeden direkt olarak yer bilgisayarına gönderilerek bütün işlemlerin bilgisayarda yapılması sağlanabilir; ama bu durumda modemin hızına bağlı olarak veri kaybı yaşanabilir) gibi cihazların sualtı aracının kendi kontrol ünitelerinden ayrı yerleştirildiği kapalı forma sahip bir yapı idi. Fakat böyle bir aracın kapalı formundan dolayı ileride yapılması düşünülebilecek sistem geliştirmeleri sınırlayabileceği ve maliyetinin yüksek oluşu nedeniyle, Özellikle Amerikadaki hobi meraklıları tarafından tercih edilen açık bir yapıya sahip, geliştirilebilir bir platform olan ROV tarzı bir sualtı aracının [33] yapılmasına karar verilmiştir.

5.2.1 Aracın Yapısı

Kullanılacak sualtı aracının üzerinde, araç elektroniğini ve sonradan eklenecek diğer sistemlerin yerleştirileceği su geçirmez bölmelerin bulunduğu, belirli doğrultularda hareket için kullanılmak üzere ayrı motorlara sahip bir yapı tasarlanmış ve yapılan birkaç farklı model sonrasında şekil 5.5'te verilen yapıda karar kılınmıştır. Nispeten düşük maliyetli olan araç, açık bir platform üzerine yerleştirilmiş gerekli ünitelerden ve pvc borulardan imal edilmiş bir dış iskeletten oluşmaktadır.



Şekil 5.5 : Tasarlanan sualtı aracı

Aracın sevki için su jeti gibi çalışacak, 3870 litre/saat (1000galon/saat) su basma kapasitesine sahip, tamamen su içinde çalışabilen 6 tane dalgıç pompadan faydalanılması, elektronik sistemlerin ise sızdırmaz kutular içine yerleştirilmesine karar verilmiştir. Şekil 5.5'te görülen aracın üst kısmındaki silindirik, kapalı borular aracın ağırlık dengesinin sağlanmasında kullanılacak boş, "hava" tanklarıdır.

Tasarımın gerçeklenmesi doğrultusunda malzeme temini için yapılan piyasa araştırması ve testler sonucu, araçta kullanılacak pompalar, zemin malzemesi, iskelet ve tanklar için pvc borular temin edilmesine karşın sızdırmaz kutularda sıkıntı yaşanmıştır. Bulunan, birkaç farklı firmaya ait IP67 standartındaki (geçici olarak su içinde çalışabilme) kapaklı kutularla yapılan sızdırmazlık testlerinin sonunda en kaliteli kutunun dahi belli bir süre içinde su aldığı gözlemlenmiştir. Sürekli su içinde çalışabilecek IP68 olarak kabul edilen sızdırmaz kutular özel üretim oldukları için yüksek maliyeti ve uygun üretici bulunamadığından alınan IP67 kutuların yalıtılmalarına karar verilmiştir. Bu doğrultuda kutu kapaklarına kendi contaları dışına sıvı conta ile yalıtım uygulanırken, kablo giriş çıkışlarında sızdırmaz rakorlarla birlikte silikon takviyesi de kullanılmıştır.

Pvc boruların birleştirilmesi ile elde edilen iskelet yapıya hava tankınında eklenmesiyle şekil 5.6'da görülen 30 kg üzeri taşıma kapasitesine sahip su altı aracı oluşturulmuştur. Daha sonra diğer elemanlar da yerleştirilip sabitlenerek araç yapısal larak hazır hale gelmiştir.

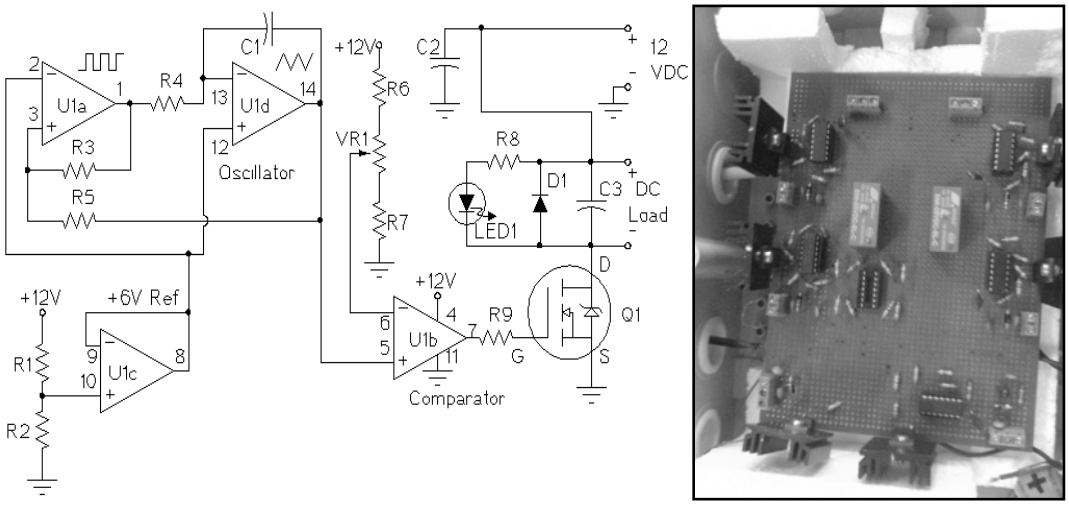


Şekil 5.6 : Sualtı aracının bitmiş hali

5.2.2 Elektronik Sistem ve Araç Kumandası

Sualtı aracının formu hazırlandıktan ve kullanılacak motorlar (pompalar) da temin edildikten sonra sıra bu motorların sürülerek aracın hareket ettirilmesine gelmektedir. Araç istenirse içine yerleştirilecek olan bir mikro-bilgisayar tarafından kontrol edilerek otonom hareket edebilir. Yine de başlangıç olarak uzaktan bir insan tarafından kontrol edilmesi daha uygun görülmektedir. Bunun için ROV'lerde genel olarak kullanılan kablo bağlantısı ile araç kumandası yöntemine uygun bir darbe genişlik modülasyonlu (PWM-Pulse Width Modulation) doğru akım (DC) motoru sürücü devresi hazırlanmıştır (Şekil 5.7). Burada darbenin genişliği motora giden gerilimi belirler, bu sayede PWM değişimi ile motor farklı hızlarda sürülebilir. Şekil

5.7'deki devre temel olarak PWM sinyalinin oluşturulduğu dört adet işlemsel kuvvetlendirici ve motoru süren bir MOSFETten oluşmaktadır.

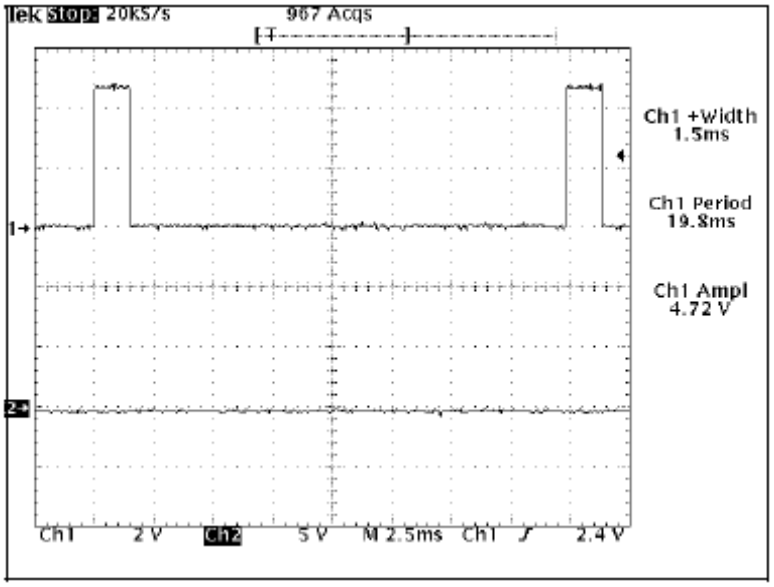


Şekil 5.7 : Kullanılan PWM motor sürücü devresi

Burada U1a tarafından oluşturulan kare dalga U1d'de üçgen dallgaya dönüştürülmektedir. U1c'de oluşturulan 6V referans akım U1d için sanal topraklama görevi görür, bu sayede sisteme eksi gerilim bağlamak gerekmez. U1d'nin çıkışındaki üçgen işaret U1b'de 6 numaralı bacaktan gelen değişken gerilim ile karşılaştırılır. Bu bacağa bağlı potansiyometre ile üçgen işaretin kesim genliği belirlenir. Üçgen işaret, 6 numaralı bacaktan gelen gerilimden yüksekse PWM çıkışı büyük, düşük ise PWM çıkışı daha dar olur. Bu sayede potansiyometre ile MOSFET'in kapısının açılıp kapanma süresi değiştirilmiş olur, bu da motoru yüksek frekanslarda açıp kapayarak sabit 12V gerilim beslemesi olmasına rağmen sanki motor besleme gerilimi değiştiriliyormuş gibi bir durum oluşturularak motorun hız kontrolü gerçekleştirilmiş olur.

Her bir motor (pompa) için yapılmış olan devreler bir kart üzerinde toplanarak sualtı aracı için kumanda ünitesi oluşturulmuştur. Ancak potansiyometre ile hız kontrolü için araç ile dışarısı arasında sürekli bir kablo bağlantısı olmak zorundadır. Halbuki aracın tamamen mobil olması karaya bağlı olmaması istenmektedir. Aynı nedenden ötürü aracın elektrik beslemesi genelde kullanıldığı gibi kablo ile suyun dışından yapılmak yerine araçüzerine yerleştirilen akülerle sağlanmaktadır. Aynı zamanda aracın çalışacağı derinliğin en fazla 3.5m (havuz derinliği / tatlı su – su yoğunluğu arttıkça radyo dalgalarının nüfuziyeti azalmaktadır) olacağı öngörüsüyle model araba ve uçaklar için kullanılan bir radyo kontrol ünitesinin yeterli olduğu düşünülmüştür.

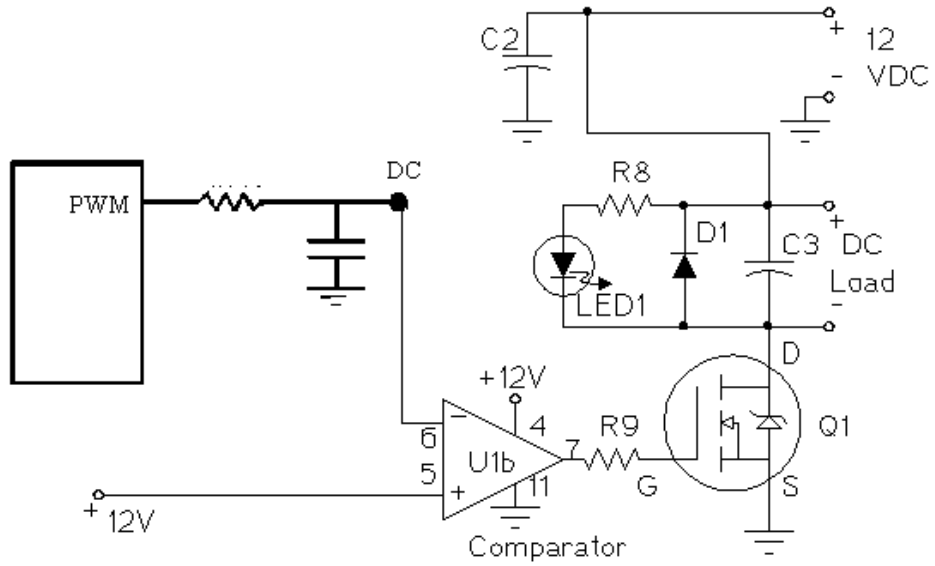
Buradan hareketle bu tip radyo alıcı, vericilerin haberleşmeleri ve alıcı üniteden motor sürmek için ne tip bir işaretin çıktığı incelenmiştir. Alıcı ünite ile buna bağlanan servo motorlar arasında üçlü kablo düzeni bulunmaktadır: Bunlar sırasıyla toprak hattı, motor beslemesi olan 5V gerilim ve motor kontrolü için sistemegirilen PWM sinyalidir. Bu bilgi doğrultusunda model radyo alıcısının PWM çıkışı elimizdeki hazır motor sürücü devresinin MOSFETine bağlanarak test edilmiş, ama olumlu bir sonuç alınamamıştır. Bunun nedeni araştırıldığında sıkıntının alıcı ile vericinin haberleşme protokolünden kaynaklandığı görülmüştür. Bu tip kumandalarda alıcı ünite her bir kanal için 20 ms'lik periyotlarla 1 ile 2 ms arasında değişen PWM sinyalleri yollamaktadır (Şekil 5.8). Normalde servo motorun kafa açısını kontrol eden bu sinyal, bizim durumumuzda minimum %5 ile maksimum %10luk bir besleme gerilimi veya çalışma döngüsü anlamına gelmektedir. Bu da kullandığımız motorların sürülmesine yetmemektedir.



Şekil 5.8 : Alıcıdan çıkan PWM sinyali

Bu durumda yapılması gereken darbe genişliğinin arttırılmasıdır. Şekil 5.9'da verilen devre kullanılarak belli oranda darbe genişliği arttırılmıştır. Alıcı çıkışındaki PWM sinyali bir RC filtreden geçirilerek doğru akıma dönüştürülmüştür. Ancak bu doğru akım üzerinde kullanılan elemanların büyüklüklerine göre değişen bir dalgalanma (gürültü) bulunmaktadır. Bu gürültülü sinyal daha çok bir üçgen dalga sinyale benzemekte, ve onu bir karşılaştırıcıdan geçirdiğimizde de yeni bir PWM sinyali oluşmaktadır. Bu yeni PWN'nin darbe genişliği gerçekPM'ye göre oldukça

büyümüştür. Yeni elde edilen PWN'nin MOSFET'in kapısına (gate) bağlanmasıyla radyo kumanda ile motorlar sürülebilmektedir.



Şekil 5.9 : Alıcıdan çıkan PWM sinyali ile motor sürme

6. SONUÇLAR VE TARTIŞMA

6.1 Yapılan Çalışmanın Sonuçları

Bu tez çalışmasında bir sualtı aracının yörünge takibinde kullanılması öngörülen ataletsel algılayıcı ünitesinin verdiği ivme verisinin işlenmesiyle gerçeğe en yakın konum değişimi değerlerine ulaşılmaya çalışılmıştır. Bu kapsamda öncelikle algılayıcı üniteden elde edilen ivme verileri incelenmiştir. Daha sonra bu verinin içinde bulunan hataların giderilmesi için çeşitli filtreleme yöntemleri denenmiştir.

Algılayıcının sayısal haberleşme özelliğinden dolayı veriye frekans düzeyinde müdahele edilememiş, bunun yerine sahip olduğu değişken gürültü düzeyi göz önüne alınarak uyarlamalı filtreleme yöntemleri denenmiştir. LMS ve çeşitli YSA algoritmaları ile yapılan testlerde umut verici sonuçlar elde edilmiş, ayrıca YSA'nın LMS'e olan üstünlüğü gösterilmiştir.

YSA algoritmaları ile yapılan filtreleme çalışmalarında bu algoritmaların eğitilen belli sinyale göre sonuçlar verdikleri gözlendiğinden farklı sinyal tiplerine göre eğitilmiş YSAlardan oluşan parçalı bir YSA modeli oluşturularak karşılaşılacak farklı verilere kolay uyum sağlayarak daha başarılı bir filtrelemenin geçekleşmesi sağlanmıştır.

Su altında hareket edecek bir aracın dış bozuculara daha az maruz kalacağı öngörüldüğünden ataletsel algılayıcılarla seyir için sualtı aracı uygun görülmüştür. Ayrıca aracın su altında 3 eksende de hareket edebilmesinin uygulama açısından da büyük avantaj sağlayacağı öngörülmüştür. Bu kapsamda, uygulamaya yönelik bir konum takip programı da Simulink ortamında geliştirilmiştir.

Bilgisayar ortamında yapılan testlerde YSA tabanlı filtreleme ile araç takibinin yapılabileceği görülmüştür. Ancak unutulmamalıdır ki gerçek ortamda denemeler gerçekleştirilmeden kesin bir hükme varmak yanlıştır. Zira yapılan çalışmada araç içi ve dışından gelebilecek bozucular (motor gürültüsü, vs...) hesaba katılmamıştır. Yapılan denemelerde oda ortamında bulunan bir algılayıcıdan elde edilen veriler

kullanılmış, algılayıcının kendi gürültüsü ve bu ortamda algıladığı düşük miktarda dış gürültü filtrelenmiştir. Yine de karakteristiği belli olmayan gürültülerin giderilmesinde uyarlamalı filtrelerin başarısı ortaya konmuş, bu tür alternatif yöntemlerle araç takiplerinin gerçekleştirilebileceği gösterilmiştir.

Son olarak da bu çalışmanın devamına yönelik olarak gerçekleme testlerinde kullanılacak bir sualtı aracı imal edilmiştir. Aracın gövdesi ve sevk sistemi hazırlanmış, motor olarak kullanılan dalgıç pompaları radyo kumanda yardımıyla sürecek elektronik devre oluşturulmuştur.

6.2 Geleceğe Dönük Öneriler

Yapılacak gerçekleme elde edilecek sonuçlar doğrultusunda sistemin geliştirilip daha kullanışlı hale getirilmesi düşünülmektedir. Yörünge takibi gerçekleştirildikten sonra bir üst adım olarak aracın otonom hareket ederek verilen görevleri gerçekleştirmesi öngörülmektedir. Başarılı sonuçların elde edilmesi ve sisteme esnek bir yapı kazandırılmasıyla bu sistem ticari bir ürün olarak da kullanılabilir. Hatta sistemin çevreden bağımsız çalışması sayesinde sistem rahatlıkla deniz dışında, kara ve hava araçlarına da uygulanabilir.

KAYNAKLAR

[1] **Yenilmez, Levent**, 2001. Mobil Robotlarda Çoklu-Sensör Veri Füzyonu Teknikleri ile Lokalizasyon ve Harita Oluşturma. *Doktora Tezi İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Bilgisayar Mühendisliği, İstanbul*

[2] **Pritchett, John E.; Pue, Alan J.**; 2000. Robust guidance and navigation for airborne vehicles using GPS/terrain aiding. *IEEE PLANS, Position Location and Navigation Symposium,* p 457-463

[3] **Basu, P; Redi, J.; Shurbanov, V.**; 2004. Coordinated Flocking of UAVS For Improved Coonectivity of Mobile Ground Nodes. *Milcom*

[4] **Tapper, Myron L.; Dana-Bashian, David J.; Hablani, H.B.**; 2002. Guidance and relative navigation for autonomous rendezvous in a circular orbit *Journal of Guidance, Control, and Dynamics,* v 25, n 3, May/June, 2002, p 553-562

[5] **Grimnes, Morten; Hoff, Lars; Halvorsen, Steinar; Elle, Ole Jakob; Alker, Hans Jorgen; Fosse, Erik**; 2004. Velocity and position approximations from left ventricular 3D accelerometer data *Bioengineering, Proceedings of the Northeast Conference, v 30, Proceedings of the IEEE 30th Annual Northeast Bioengineering Conference*, p 25-26

[6] **Mori, Yoshikazu; Nakamura, Tatsuya; Asano, Michitomo; Goto, Atsutoshi**; 2002. Position estimation for a mobile robot, using an accelerometer with impedance-type detector**.** *Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C*, v 68, n 3, p 870-876

[7] **Thong, Y.K.; Woolfson, M.S.; Crowe, J.A.; Hayes-Gill, B.R.; Challis, R.E.;** 2002. Dependence of inertial measurements of distance on accelerometer noise. *Measurement Science and Technology*, v 13, n 8, p 1163-1172

[8] **Liu, H.H.S.; Pang, G.K.H.**;2001. Accelerometer for mobile robot positioning. *IEEE Transactions on Industry Applications*, v 37, n 3, p 812-819

[9] **Zhao, Lin; Gao, Wei,** 2004. The experimental study on GPS/INS/DVL integration for AUV. *Record - IEEE PLANS, Position Location and Navigation Symposium*, *PLANS - 2004 Position Location and Navigation Symposium*, p 337-340

[10] **Boskovic, Dejan M.; Krstic, Miroslav**; 1999. Global attitude/position regulation for underwater vehicles**.** *International Journal of Systems Science*, v 30, n 9, p 939-946

[11] **Keary, A.; Phillips, B.; Robinson, H**.; 2001. COVELIA – A Correlation Sonar Velocity Sensor. *H Scientific Ltd and Marine Acoustics Ltd*

[12] **Smallwood, David A.; Whitcomb, Louis L.**; 2004. Model-Based Dynamic Positioning of Underwater Robotic Vehicles: Theory and Experiment**.** *IEEE Journal of Oceanic Engineering*, v 29, n 1, p 169-186.

[13] **Huster, A.; Frew, Eric W.; Rock, Stephen M.**; 2002. Relative position estimation for AUVs by fusing bearing and inertial rate sensor measurements**.** *Oceans Conference Record (IEEE)*, v 3, p 1864-1871

[14] **Microstrain Inc.** www.microstrain.com

[15] **Hou, Haiying,** 2004 Modeling Inertial Sensors Errors Using Allan Variance, *Department of Geomatics Engineering, The University of Calgary, Calgary, Canada*

[16] **El-Sheimy, N.**, 1996 The Development of VISAT-A Mobile Survey System for GIS Applications, Report No. 20101, *Department of Geomatics Engineering, The University of Calgary, Calgary, Canada*

[17] **Hatun, M. ve Koçal O.H.,** 2005. Adaptif Filtrelerde Gauss-Seidel Algoritmasının Stokastik Yakınsama Analizi, *Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi Cilt 10, Sayı 2* s. 87-92.

[18] **Widrow, B. and Hoff, M.E.,** 1960. Adaptive Switching Circuits, *IRE WESCON Conv. Rec.,* 4, 96-104.

[19] **Widrow, B.,** 1971. Adaptive Filters, *Aspects of Network and System Theory,* Eds. Kalman R. and DeClaris N., Holt, Rinehart and Winston, New York, s.563-587.

[20] **Gou, Hongmei.,** 2003. Adaptive Signal Processing*, ENEE 724 Project 2*.

[21] **Haykin, S. et. al.,** 1997. Adaptive Tracking of Linear Time-Variant Systems by Extended RLS Algorithms, *IEEE Transactions on Signal processing v.45, no.5 ,s. 1118-1128*

[22] **Noureldin, Aboelmagd; Osman, Ahmed, El-Sheimy, Naser** 2004 A neuro-wavelet method for multi-sensor system integration for vehicular navigation. Journal of Measurement Science and Technology, Volume 15, Issue 2, pages 404-412.

[23] **Kaygisiz, B.H.; Erkmen, A.M.; Erkmen, I**. 2003 GPS/INS enhancement using neural networks for autonomous ground vehicle applications. Intelligent Robots and Systems, (IROS 2003)

[24] **Sayyaadi, H.; Ura, T.;** 1999. Multi Iput – Multi Output System Identification of AUV system by Neural Network. OCEANS'99 MTS/IEEE. Riding the Crest into the 21st Century

[25] **Atmaca, V.,** 2005. Örme Kumaşlardaki Üretim Hatalarının Görüntü İşleme Teknikleri ile Otomatik Tespiti ve Sınıflandırılması, Y. Lisans *Tezi İTÜ Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği, İstanbul*

[26] **Riedmiller, M., and H. Braun,** 1993 "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," *Proceedings of the IEEE International Conference on Neural Networks*

[27] **Hamzaçebi, C. ve Kutay F.,** 2004. Yapay Sinir Ağları ile Türkiye Elektrik Enerjisi Tüketiminin 2010 Yılına Kadar Tahmini, Gazi Üniv. Müh.Mim.Fak. Der. cilt 19, no.3 s.227-233

[28] **Sabuncu, T.,** 1993. Gemi Hareketleri, Gemi İnşaatı ve Deniz Bilimleri Fakültesi Ofset Atölyesi, İstanbul.

[29] **Brutzman, D. P.,** 1994. A Virtual World For An Autonomous Underwater Vehicle, Thesis, Naval Postgraduate School, California.

[30] **Cooke, J.C., Zyda, M.J., Pratt, D.R. and McGhee, R.B.**, 1992 "NPSNET: Flight Simulation Dynamic Modeling using Quaternions," *PRESENCE: Teleoperations and Virtual Environments*, vol. 1 no. 4, , pp. 404-420.

[31] **Course Objectives** – Chapter 10 – Submarine and Submersibles, - www.usna.edu/NAOE/courses/en200/ch10.pdf

[32] **Doğangil, G., Özçiçek, E., Kuzucu, A.,** 2005. Bir Robot Yunusun Tasarımı ve Geliştirilmesi, Otomatik Kontrol Ulusal Toplantısı-TOK'05, İTÜ, İstanbul, 2-3 Haziran, s. 273-278

[33] **Bohm, H. ve Jensen, V.,** 2005. Built Your Own Underwater Robot, Westcoast Words, Vancouver, B.C. Kanada

ÖZGEÇMİŞ

M. Ufuk Altunkaya, 1981 yılında İstanbul'da doğmuştur. İlkokul, orta ve lise öğrenimini İstanbul'da tamamladıktan sonra 1999 yılında girdiği İstanbul Teknik Üniversitesi Gemi İnşaatı Muhendisliği bölümünden 2003 yılında mezun oldu. Aynı yıl başladığı İstanbul Teknik Üniversitesi Mekatronik Mühendisliği yüksek lisans programındaki eğitimine devam etmektedir.