



T. C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇEVİRİMİÇİ SİSTEM PARAMETRELERİNİN KESTİRİMİ

109688

EKREM DÜVEN

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

109688

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA - 2001


T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

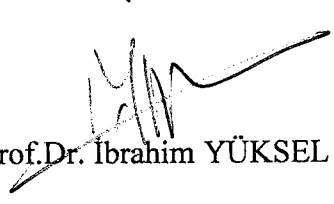
ÇEVİRİMİÇİ SİSTEM PARAMETRELERİNİN KESTİRİMİ


EKREM DÜVEN

YÜKSEK LİSANS TEZİ
ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 16.01.2001 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.


Yrd.Doç.Dr. Murat TÜRE
(Danışman)


Prof.Dr. İbrahim YÜKSEL


Yrd.Doç.Dr. Osman H. KOÇAL

ÖZET

Karşılaştığımız olaylar ya da durumlar karşısında ; bunun bir sonraki adımda bize yansıyacak olan sonuçlarını bilebilmek veya kestirebilmek insanlar için önemli bir problem teşkil etmektedir . Bu problemin üstesinden gelinebildiği ölçekte geleceğe dönük planlar yapmak ve bunlar doğrultusunda uygun adımlar atmak mümkün olmaktadır . Hem sosyal hem de teknik alanda karşımıza çıkan bu problemin teknik alandaki yansıması , ele alınan sistemlerin kontrol edilebilirliği şeklinde ortaya çıkmaktadır . Eğer üzerinde çalıştığımız sistemin bir sonraki aşamada nasıl bir davranış göstereceğini bilebiliyorsak , onu kontrol etmemiz çok kolay olmaktadır . Bunun içinse; o sistemi tanımak yani onu modelleyebilmek gerekmektedir . Modelini yarattığımız sistemin parametrelerini elde ederek , onun tüm davranışlarına hükmetmemiz mümkündür .

İşte bu çalışmada ; sistem belirleme parametrelerini elde ederek onu tanımak amacıyla kullanılan : “Sistem Belirleme Metot’ ları” incelenmektedir . Çalışma içerisinde ilk olarak sistem belirleme metotları ayrıntılarıyla ele alınarak incelenmiştir . Bir sonraki bölümde de kestirim işleminde kullanılan giriş-çıkış bilgilerinin elde edildiği sistem üzerinde durulmuş , sistemin oluşturulması ve çalışmasından bahsedilmiştir . Ardından gelen son bölümde ise gerçek bir sistemden elde edilen veriler yardımıyla gerçekleştirilen on-line kestirim işlemi ve sonuçları verilmiştir . Bu bölümde aynı zamanda elde edilen sonuçların yorumlaması da yer almaktadır .

ABSTRACT

Estimating the results of the events and situations, which will have an effect on us at the next step of our way, is a real problem for us. Planning the future and progressing step by step through this plan can be possible only if the problem of these types has been solved. The technical reflection of this problem, which occurs in either technical area or social area, can be defined as the controllability of the system, which would be examined. If we know how the system will behave at the next step, we can easily control the system. For this aim, we must define the system. By obtaining the parameters of the system that we modelled, we can rule the system's behaviour.

In this study, "System Identification Methods" which is used for obtaining for system identification parameters and for defining a system, is examined. Firstly, system identification methods are examined in detail. After that, working of a system and obtaining its parameters are discussed where the system gives us input - output knowledge for our estimation process. Lastly, the results of an on-line estimating process, which uses a real system to obtain input – output knowledge, are given. Also, the comments about these results take place in this chapter.

İÇİNDEKİLER

Simgeler Dizini	v
Şekiller Dizini	viii
Çizelgeler Dizini	xi
1. Giriş	1
2. KURAMSAL BİLGİLER	3
2.1. Sistem Belirlemenin Tanımı Ve Tarihsel Gelişimi	3
3. MATERYAL ve YÖNTEM	9
3.1. Sistem Belirleme Metotları	9
3.1.1. Klasik Metotlar	9
3.1.1.1. Frekans Cevabı Metodu	9
3.1.1.2. Darbe Cevabı ile Kestirim (Birim Basamak Cevabı Yöntemi)	10
3.1.1.3. Dekonvolüsyon Metodu	12
3.1.1.4. Korelasyon (İlişki) Metodu	14
3.1.1.5. Darbe cevabından yararlanılarak transfer fonksiyonunun bulunuşu	16
3.1.2. Modern Metotlar	18
3.1.2.1. Çevrimdışı Metotlar	18
3.1.2.1.1. En Küçük Kareler (Least Squares) Metodu	18
3.1.2.1.2. Maksimum Benzerlik Yöntemi	22
3.1.2.1.3. Genelleştirilmiş En Küçük Kareler Yöntemi	26
3.1.2.1.4. Yardımcı Değişkenler (Instrumental Variables) Metodu	29
3.1.2.2. Çevrimiçi Metotlar	30
3.1.2.2.1. En Küçük Kareler Metodu'nun Çevrimiçi Algoritması	30
3.1.2.2.2. Sistem Belirleme Parametrelerinin Yakınsaması _ Uygunluğu	32
3.1.2.2.3. En Küçük Kareler Metodunun Uygunluk Analizi	33
3.1.2.2.4. Ağırlıklı En Küçük Kareler Metodu (Çevrimiçi WLS)	35
3.1.2.2.5. Genelleştirilmiş E.K.K. (Çevrimiçi)	39
3.1.2.2.6. Yardımcı Değişkenler Metodu (Çevrimiçi)	41

3.1.2.2.7. Ardışıl İlişki (Correlation) Metodu	43
3.1.2.2.8. Maksimum Benzerlik (Maksimum Likelihood)Metodu (Çevrimiçi)	45
3.1.2.2.9. Stokastik Yaklaşım	48
3.1.3. Çevrimiçi Kestirim İşleminde Farklı Metotların Karşılaştırılması	50
3.1.3.1. Sinha ve Sen'in Uygulaması	50
3.1.3.2. Isermann ve Arkadaşları'nın Uygulaması	53
3.2. Sistem Belirleme İçin Kullanılan Deney Düzenegi	60
3.2.1. PC Ünitesi	64
3.2.2. DAC/ADC Kartı	67
3.2.3. Sayıcı Kartı	68
3.2.4. Enkoder Sinyallerini Değerlendirme Kartı	70
3.2.5. Veri Dönüşüm Ünitesi	73
3.2.6. Sürücü – BLH75B	75
3.2.7. Motor _ML3450A-10_	77
4. ARAŞTIRMA SONUÇLARI ve TARTIŞMA	79
4.1. Deney Düzenegi	79
4.1.1. Giriş – Çıkış Verilerinin Elde Edilmesi	80
4.1.2. Sistem Modelinin Belirlenmesi	81
4.1.3. Kestirim İşleminin Gerçekleştirilmesi	83
4.1.4. Farklı Değerlerden Elde Edilen Kestirim Değerleri	86
4.1.5. Sonuçların Değerlendirilmesi ve Tartışma	92
Kaynaklar	95
Ek' ler	98
Teşekkür	119
Özgeçmiş	120

KISALTMALAR VE SİMGELER

LS	Least Squares En Küçük Kareler
GLS	Generalized Least Squares Genelleştirilmiş En Küçük Kareler
WLS	Weighted Least Squares Ağırlıklı En Küçük Kareler
IV	Instrumental Variables Yardımcı Değişkenler
ML	Maximum Likelihood Maksimum Benzerlik
STA	Stochastic Approximation Stokastik Yaklaşım
COR	Correlation Analysis Korelasyon Analizi
u_k	Ayrık zaman için giriş vektörü
x_k	Ayrık zaman için gürültüsüz çıkış vektörü
n_k	Ayrık zaman için sistem çıkışına etkiyen filtrelenmiş gürültü vektörü
v_k	Ayrık zaman için sistem çıkışına etkiyen beyaz ya da pseudo random gürültü vektörü
y_k	Ayrık zaman için üzerine gürültü eklenmiş çıkış işareti
$G(s)$, $H(s)$, $T(s)$	Sistemin sürekli zaman transfer fonksiyonu
$G(z)$, $H(z)$, $T(z)$	Sistemin ayrık zaman transfer fonksiyonu
θ_k	Gerçek parametre vektörü
θ_{kp}	Tahmin edilen parametre vektörü
φ_k	Durum vektörü
e_k	Hata vektörü
ϕ_k	Birleştirilmiş durum vektörü
ε_k	Birleştirilmiş hata vektörü
ψ_k	Yardımcı parametre vektörü

$\phi_{yx}(\theta)$	Çapraz ilişki (cross correlation) fonksiyonu
$\phi_{aux}(k)$	Yardımcı değişkenler metodu için Young' ın filtre fonksiyonu
$\delta(\theta-\tau)$	Delta-dirac fonksiyonu
PDF	Probability Density Function
EKK	En Küçük Kareler
SISO	Single Input Single Output Tek Giriş Tek Çıkış
$E[x y]$	Beklenen değer fonksiyonu
σ	Standart sapma fonksiyonu
η	Gürültü / sinyal oranı
PRBS	Pseudo Random Binary Signal
DAC	Digital to Analog Converter Dijital-Analog Çevirici
ADC	Analog to Digital Converter Analog-Dijital Çevirici
PC	Personal Computer
OSC	Oscillator Osilatör
TTL	Transistor-Transistor Logic
I/O	Input/Output
CPU	Central Processing Unit Merkezi İşlem Birimi
LSB	Least Significant Bit Düşük Değerli Bit
MSB	Most Significant Bit Yüksek Değerli Bit
A/D	Analog/Digital
D/A	Digital/Analog
CLK	Clock
kHz	KiloHertz
PWM	Pulse Width Modulation Darbe Genişlik Modülasyonu

AOP	Encoder'den elde edilen işaret
BOP	Encoder'den elde edilen işaret
ZOP	Encoder'den elde edilen işaret
rpm	Revolution Per Minute



ŞEKİLLER DİZİNİ

Şekil 2.1	Sistem modelini göstermek amacıyla kullanılan genel blok diyagram	4
Şekil 3.1	Sürekli zamanda en genel haliyle bir sistemin blok diyagramı	9
Şekil 3.2	Frekans cevabı metodu için sistem belirlemede kullanılan Bode Diyagramları	10
Şekil 3.3	Girişine birim basamak şeklinde bir işaret uygulanmış ikinci dereceden bir sistem için elde edilen çıkış işaretinin grafiği ve grafik üzerinde frekans cevabı metodu ile sistem kestirimi için bilinmesi gerekli değerlerin gösterimi	11
Şekil 3.4	Korelasyon metodu için girişine beyaz gürültü uygulanan sistemin çıkışındaki işaretin süperpozisyon yöntemi ile elde edilmesi	15
Şekil 3.5	Ayrık zaman domeninde , bir sistemin en genel anlamda blok diyagramı (gürültüsüz halde)	18
Şekil 3.6	Ayrık zaman domeninde , çıkışına gürültü etkileyen bir sistemin en genel anlamda blok diyagramı	19
Şekil 3.7	Yardımcı Değişkenler Metodu için ; model belirlemede kullanılan , bir sistemin en genel anlamda blok diyagramı	29
Şekil 3.8	Ağırlıklı En Küçük Kareler metodu için ; çıkışına gürültü etkileyen bir ayrık zaman sisteminin blok diyagramı	35
Şekil 3.9	Yardımcı Değişkenler Metodu' nun çevrimiçi tipi için kullanılan ; ayrık zaman sistemi blok diyagramı	42
Şekil 3.10	Sistem Belirleme Metotlarının karşılaştırılması amacıyla Isermann ve arkadaşları tarafından kullanılan örnek test sisteminin ayrık zaman blok diyagramı	53
Şekil 3.11	Sistem Belirleme işleminin örnek bir sistem üzerinde uygulamasını gerçeklemek amacıyla ; sistem kestirimi için gerekli olan giriş- çıkış bilgilerinin elde edildiği test düzeneğinin (gerçek sistem) genel devre şeması	61
Şekil 3.12	DAC/ADC kartını kullanarak , PC ile sistem arasındaki haberleşme prosedürünün nasıl gerçekleştirildiğini gösteren blok diyagram	67

Şekil 3.13	Motor üzerindeki encoderden elde edilen ve kuvvetlendirilerek ,TTL formatına uydurulan sinyaller :	71
	a. <u>İleri</u> yöndeki motor hareketi için sinyallerin birbirine göre konumu	
	b. <u>Geri</u> yöndeki motor hareketi için sinyallerin birbirine göre konumu	
Şekil 3.14	Şekil 3.13.' deki sinyalleri değerlendirmek amacıyla tasarlanan <i>Encoder sinyalleri değerlendirme kartı</i> ' nın lojik devre şeması	73
Şekil 3.15	DAC/ADC kartı ile motor sürücüsü arasındaki gerilim seviyesi kaydırılması ve Encoder sinyallerinin TTL formatına dönüştürülmesi işlevini gerçekleştirmesi için tasarlanan Veri Dönüşüm Ünitesinin devre şeması	74
Şekil 3.16	Motor sürücü ünitesinin test düzeneği üzerindeki konumu	75
Şekil 3.17	Sürücü üzerindeki encoder sinyalleri kuvvetlendiricisinin çıkışından elde edilebilen konum sinyalleri	76
Şekil 3.18	Motor ve kendisinden önce gelen sürücü devresinin iç bağlantılarını gösteren genel devre şeması	77
Şekil 3.19	Motorun test düzeneği üzerindeki fiziksel bağlantısını ve motor üzerine bağlı olan elemanları gösteren şekildir .	78
Şekil 4.1	Kestirimin işleminin gerçekleştiği deney düzeneğinin genel blok diyagramı olarak gösterimi	79
Şekil 4.2	Kestirim işleminin gerçekleştirilmesi amacıyla sisteme uygulanan çeşitli giriş işaretlerini gösteren şekildir.	80
Şekil 4.3	Girişine sinüzoidal işaret ($f=0.4$ Hz) uygulanan 2. derece sistem modeli için gerçek çıkış ve parametrelerden elde edilen çıkış işaretlerinin karşılaştırılmasını gösteren şekildir.	87
Şekil 4.4	Girişine sinüzoidal işaret ($f=0.4$ Hz) uygulanan 2. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	87
Şekil 4.5	Girişine sinüzoidal işaret ($f=0.4$ Hz) uygulanan 3. derece sistem modeli için gerçek çıkış ve parametrelerden elde edilen çıkış işaretlerinin karşılaştırılmasını gösteren şekildir.	88

Şekil 4.6	Girişine sinüzoidal işaret ($f=0.4$ Hz) uygulanan 3. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	88
Şekil 4.7	Girişine sinüzoidal işaret ($f=0.4$ Hz) uygulanan 2. ve 3. dereceden modeller için elde edilen parametrelerden bulunan çıkış işaretlerinin karşılaştırılmasını gösteren şekildir .	89
Şekil 4.8	Sisteme uygulanan farklı işaretler için elde edilen çıkış işaretlerini temsil etmesi amacıyla verilen şekildir .	90
Şekil 4.9	Girişine sinüzoidal işaret ($f=0.5$ Hz) uygulanan 2. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	90
Şekil 4.10	Girişine sinüzoidal işaret ($f=1$ Hz) uygulanan 2. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	91
Şekil 4.11	Girişine sinüzoidal işaret ($f=1.66$ Hz) uygulanan 2. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	91
Şekil 4.12	Kare dalga formundaki işaret ($f=5$ Hz) uygulanan 2. derece sistem modeli için elde edilen parametre değerlerini gösteren şekildir .	92

ÇİZELGELER DİZİNİ

Çizelge 2.1	Sistem Belirleme işleminin tarihsel gelişimini gösteren tablo	8
Çizelge 3.1	Maksimum benzerlik Metodunun iteratif çözüm yöntemini gösteren işlem akış şeması	25
Çizelge 3.2	Genelleştirilmiş En Küçük Kareler Metodu için kullanılan algoritmayı gösteren işlem akış şeması	28
Çizelge 3.3	Sistem belirleme metotlarının karşılaştırılması amacıyla Sinha ve Sen tarafından yapılan çalışmada ; 2.dereceden bir sistem için elde edilen parametre kestirim sonuçlarının , gerçek parametre değerlerine yakınsaması açısından incelendiği tablo	51
Çizelge 3.4	Sistem belirleme metotlarının karşılaştırılması amacıyla Sinha ve Sen tarafından yapılan çalışmada ; 2.dereceden bir sistem için elde edilen parametre kestirim sonuçlarının , gerektirdikleri işlem süresi açısından incelendiği tablo	52
Çizelge 3.5	Sistem belirleme metotlarının karşılaştırılması amacıyla Isermann ve arkadaşları tarafından yapılan çalışmada ; kullanılan 3 sistem içerisinde <i>Sistem III</i> için elde edilen kestirim sonuçları ve hata değerlerinin incelendiği tablo	57
Çizelge 3.6	Sistem belirleme metotlarının karşılaştırılması amacıyla Isermann ve arkadaşları tarafından yapılan çalışmada ; yapılan karşılaştırma işleminin genel sonuçlarının verildiği tablo	57
Çizelge 3.7	Sistem Belirleme Metotlarının örnek bir sistem üzerinde uygulamasını gerçekleştirmek amacıyla oluşturulan test düzeneği için genel işlem akış diyagramı	62
Çizelge 3.8	Test düzeneği üzerindeki PC üzerinde yürütülen kontrolör programının genel işlem akış diyagramı	65
Çizelge 3.9	Düzenekteki motor üzerinde yer alan encoderden elde edilen konum sinyallerini değerlendirmek için oluşturulan durum-geçiş tablosu	72
Çizelge 4.1	Kestirim sonucunda elde edilen parametre değerlerinin karşılaştırılma amacıyla bir arada verildiği çizelge	93

1. Giriş

Sistem belirleme denildiğinde ; “ ele alınan sistemin , çalışması esnasında elde edilen bilgileri (giriş ve çıkış büyüklükleri , gürültü vb.)’ nden modelinin belirlenmesi işlemi ” anlaşılmaktadır . Bu belirleme işlemi ele alınan modelin parametrelerinin kestirimi ile gerçekleştirilmektedir . Kestirim için birçok farklı metot kullanılabilir . Burada modelden kastedilen ; sistem belirleme tanımında adı geçen giriş-çıkış büyüklükleri yardımıyla oluşturulan bir matematiksel eşitlikler bütünlüğüdür ya da takımıdır . Ancak hangi metot kullanılırsa kullanılsın , elde edilen sonuçlar yani bulunan model parametreleri bize gerçek sistemin davranışları hakkında bilgi vermektedir . Bulunan parametrelerin doğruluğu ölçüsünde de sistemin davranışlarını kontrol etmek mümkün olmaktadır , ki bu da kontrol teorisinin özünü oluşturmaktadır . O halde en hızlı bir biçimde , en doğru şekilde model parametrelerini bulan yani optimum bir çözüm sunan metodun seçilmesi önemli bir hal almaktadır .

Bu çalışmada , en genel anlamda sistem belirleme metotları incelenmiş ve bu metotların örnek bir sistem üzerinde uygulaması gerçekleştirilmiştir . Bu amaçla bir test düzeneği de oluşturulmuştur . Takip eden bölümlerdeki konuları bölüm numaralarına göre bir sıralayacak olursak;

Bölüm 2’ de sistem belirlemenin tanımı yapılmakta ve tarihsel gelişiminden bahsedilmektedir . Ayrıca oluşturulacak olan modelin tipleri verilmiş ve sistem belirleme metotlarının sınıflandırılması yapılmaktadır .

Bölüm 3.1’ de ise Bölüm 2’ de sınıflandırılması yapılan sistem belirleme metotları teker teker ele alınmış , çalışma prensipleri ve algoritmalarından bahsedilmiştir . Ayrıca bahsedilen metotların bir sistem üzerindeki karşılaştırmalı sonuçları da bölüm sonuna eklenerek , metotların birbirine göre olan üstünlüklerinden bahsedilmiştir .

Bölüm 3.2’ de de sistem belirleme işleminin gerçek bir sistem üzerindeki örnek uygulamasından bahsedilmiştir . Bu amaçla oluşturulan test düzeneği , üzerinde yer alan

tüm komponentleriyle birlikte incelenerek , her bir parçanın ya da bloğun ne işe yaradığı ve nasıl çalıştığı ayrıntılarıyla anlatılmıştır . Aynı zamanda kontrol ve kestirim amacıyla düzenek üzerinde yürütülen bilgisayar programlarının da açıklaması yapılmıştır .

Bölüm 4' te ise ; eldeki giriş ve çıkış bilgilerinden sistem belirleme işleminin nasıl gerçekleştirildiği , model seçiminin nasıl yapıldığı ve kullanılan yöntem açıklanmıştır . Bunlara göre elde edilen sonuçlar verilmiş ve bu sonuçların değerlendirilmesi yapılmıştır .

Çalışmanın sonunda yer alan Ek' lerde ise ; çalışmada kullanılan test düzeneği üzerinde kontrolör ve kestirici görevlerini gören programların kaynak kodlarıyla , sistem üzerindeki tüm elemanların teknik bilgileri verilmektedir .



2. KURAMSAL BİLGİLER

2.1. Sistem Belirlemenin Tanımı ve Tarihsel Gelişimi

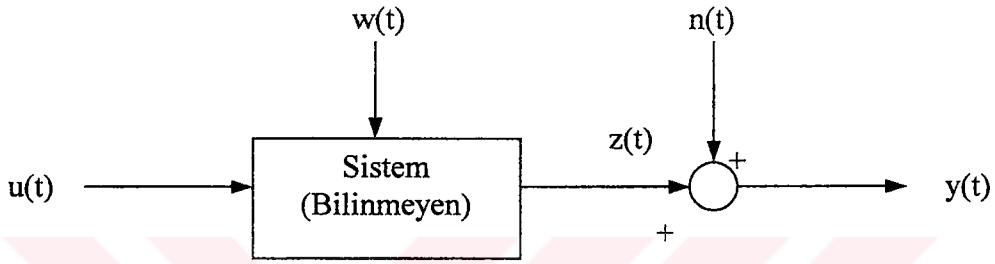
Günlük hayatta şu ya da bu şekilde sürekli karşımıza çıkan birçok sistemin bir şekilde kontrolümüz altında çalışması arzu edilir . Buna kısaca sistemin kontrol edilebilmesi denilebilir . Bu işlevin gerçekleştirilmesi içinde öncelikle sistemin tanınması yani modellenmesi gereklidir . Tekniğin gelişip her alanda kendine geniş yer bulmasıyla , bahsettiğimiz sistemler daha karmaşık formlara girmiş , bu da model kestirimini daha önemli kılmıştır . 20. yüzyılın ikinci yarısında bu problem, özellikle adaptif (uyumlu) kontrolün gelişmesiyle daha çekici bir hal almıştır . Bu çalışmalar çok geniş bir alanı , kimya endüstrisinden – havacılığa kadar geniş bir yelpazeyi kapsamaktadır . Sözü edilen bu geniş yelpazenin her bir alanında , ele alınan sistemin modelini , basit matematiksel eşitliklerle ifade ederek sistemin davranışlarını anlamak , sistemin kestirimi ve kontrolü için gerçekleştirilecek ilk ve en önemli adım olmaktadır.

Sistem belirleme iki farklı şekilde gerçekleştirilebilir : Bunlardan ilkinde ; belirli sayıda ölçülebilir fiziksel neden – etki (inputs) ve bunların oluşturduğu yine belirli sayıda ölçülebilir fiziksel sonuç – tepki (outputs)' nin elde edilebildiği bir durum söz konusudur . Bu durumda giriş ve çıkış bilgilerinden (değerlerinden) , genelde nonlineer kısmi diferansiyel denklemlerden oluşan matematiksel modeller oluşturulur . (Denklem takımının temelini oluşturmak içinse temel fizik kanunlarından faydalanılır. Eldeki giriş ve çıkış bilgileri sayesinde de bu eşitliklerin son hali bulunur.) Sistem belirlemenin bu tipi iyi bir mühendislik bilgisi gerektirmektedir . Çalışmada kullanılan ve sonraki bölümlerde de ele alıp bahsedilecek yöntemler de bu tipe girmektedir .

Sistem belirlemenin diğer tipinde ise ; sistem için ölçülebilir sonuç ya da etkileri tam olarak belirleyebildiğimiz halde , bunların nedenlerini tam olarak belirleyemediğimiz bir durum söz konusudur . Bu duruma örnek birkaç olay verecek olursak ; bir ülkeye düşen yıllık yağış miktarı , bir nehrin yıllık ortalama debisi , bir süpermarketin stok durumu vb. verilebilir. Bu tür sistemlerde sonuçları (outputs) tam olarak sırası ile bilebilmemize rağmen (bu sıra bize zaman içerisinde bir seri

oluşturabilmektedir.) nedenlerini (inputs) çok fazla sayıda oldukları için kesin olarak gözlemleyemeyiz . Buna rağmen bu sistemlere ait bir model geliştirmek , geleceğe dönük planlar yapabilmek açısından çok önemlidir .

Önceki paragraflarda sistem modelini belirlemek için sistemin çalışması esnasında elde edilen bilgilerin kullanıldığı belirtilmişti . Şimdi bunu bir diyagram ile grafik olarak gösterecek olursak ;



Şekil 2.1. Sistem modelini göstermek amacıyla kullanılan genel blok diyagram

Burada ;

$u(t) \rightarrow$ m boyutlu giriş vektörünü

$z(t) \rightarrow$ p boyutlu çıkış vektörünü

$w(t) \rightarrow$ giriş bozucu vektörünü

$n(t) \rightarrow$ gözlem gürültüsü vektörünü

$y(t) \rightarrow$ p boyutlu ölçülen çıkış vektörünü göstermektedir.

Sonuç olarak sistem belirleme : kaydedilen $u(t)$ ve $y(t)$ değerleri yardımıyla bilinmeyen sistem modelini bulmaktır .

Bu model tanıyımı ilerde detaylı açıklaması yapılacağı üzere ; En Küçük Kareler (Least Squares) metodunun kullanımı ile başlamıştır. Bu metot daha sonra Jones (1966) tarafından recursive hale getirilmiştir. Aynı zamanlarda Aström ve Bohlin (1966) tarafından Ağırlıklı En Küçük Kareler (Weighted Least Squares) Metodu geliştirilmiştir. Bu çalışmalara paralel olarak çevrim dışı uygulamalar için Aström ve Bohlin (1965) Maksimum Benzerlik Metodu' nu ortaya koymuştur. Yapılan

çalışmaların bir gelişmesi olarak ; En Küçük Kareler metodunun beyaz gürültü haricinde de yakınsayabilmesi için , Clarke (1967) tarafından Genelleştirilmiş En Küçük Kareler (Generalized Least Squares) Metodu sunulmuştur. Daha sonra bu metot Hasting-James ve Sage (1969) tarafından çevrim içi çalışmalar için recursive hale sokulmuştur. Devam eden yıllarda Söderström (1974) ve bir çok araştırmacı tarafından bu konuda çeşitli çalışmalar yapılmıştır. Gertler ve Banyasz (1974) tarafından Maksimum Benzerlik Metodu çevrim içi uygulamalar için kullanılabilir bir şekle sokulmuştur.

Modeli belirlenmek istenen sistem ile ilişkisiz fakat aynı girişe sahip gürültüsüz bir sistemin birleştirilme temeline dayanan Yardımcı Değişkenler (Instrumental Variables) Metodu' nu ise Wong ve Polak (1967) ortaya koymuş ve metot üzerinde Young (1970)' ın yaptığı güncelleme filtresi ilavesi ile kararsızlık problemi ortadan kaldırılmıştır . Stokastik sistemler üzerine Robbins ve Monro (1951) tarafından başlatılan çalışmalar ve yaklaşımlar , daha sonra bazı araştırmacılar tarafından deterministik sistemlerin tanıyımı için bir yaklaşım olarak (Saridis and Stein (1968)) sunulmuştur. Daha sonra bunlar Kwanty (1972) ve Saridis (1974)' in sunduğu güncellemeler ile kullanılabilir hale getirilmiştir.

Sistem ve model tanıyımı için, çalışılan ortamlara uygun çeşitli geliştirmeler ile bir çok farklı metot üretilmesine rağmen , bunların birçoğu yukarıdaki paragraflarda bahsedilen temel metotlardan türetilmiştir. Tüm bu metotlar ayrık zaman modeli tanıyımı için geliştirilmiştir. Sürekli zaman modeli tanıyımı üzerine ise ; yakın zaman sürecinde bu metotların uyumu (ayrık zaman metotları) ve bazı yeni metotların geliştirilme çalışmaları başlamış ve devam etmektedir. Ancak bu projede bu metotlar kullanılmadığından sürekli zaman kestirimi hakkındaki ayrıntılar verilmeyecektir.

Sistem belirlemedeki temel problem sistemi tahmin edeceğimiz model yapısının seçimidir . Model aşağıdaki tiplerden birisi olabilir :

- i. Doğrusal zamanla değişmeyen – Adi doğrusal diferansiyel denklem eşitliği

- ii. Doğrusal zamanla değişen – Adi doğrusal diferansiyel denklem eşitliği
- iii. Doğrusal fakat dağıtılmış parametrelili – Kısmi diferansiyel denklem eşitliği
- iv. Doğrusal olmayan – Nonlineer diferansiyel denklem eşitliği

Pratikte birçok sistem modeli kısmi türevli ve nonlineerdir . Fakat basitliğinden dolayı lineer modeller kullanılmaktadır . Çünkü birçok durumda bu modeller sistemi tanımakta yeterli olacaktır . Bu durumda modelin geçerli olduğu sınırlar çok önemlidir . Uygun bir doğrusal modelin seçiminde parametrelerin belirlenmesi özenle yapılmalıdır .

Sistem belirlemedeki bazı problemleri şöyle tanımlayabiliriz :

- i. Doğrusal modelin derecesinin belirlenmesi
- ii. Modelin doğruluğunun belirlenmesi için uygun kriterlerin seçimi
- iii. Model parametrelerinin tahmininin doğruluğunu maksimum yapacak (yapabilecek) giriş işaretinin tasarımının yapılması

Halihazırdaki birçok sistem sürekli zaman tipinde olmasına karşın , kestirimde sayısal bilgisayar uygulamaları için ayrık zaman modelinin kullanımı tercih edilir . Böylece ayrık zaman model parametreleri kolaylıkla belirlenebilir ve örnekleme süresi uygun ise ayrık zaman modelinden sürekli zaman modeline geçilebilir.

Sistem belirleme işlemi çevrimiçi ve çevrimdışı olmak üzere iki farklı türe daha bölünebilir . Çevrimdışı tip dediğimizde akla ; “sistemden elde edilen çok fazla miktarda giriş ve çıkış bilgisinin bilgisayar ya da herhangi bir yolla toplanıp depolanması , daha sonra bu verilerin değerlendirilerek tanımlanan matematiksel eşitlik takımlarında en uygun parametre değerlerinin hesaplanması” gelmektedir . Çevrimdışı tanıyımında işlem zamanından bağımsız olarak , istenilen metodu seçmek için geniş bir esneklik söz konusudur . Bu da bize parametrelerin kestiriminde büyük bir doğruluk sağlayacaktır .

Birçok kontrol uygulamasında , özellikle adaptif kontrolde sistemin çok kısa bir zaman zarfında tanıyımı çok önemlidir . Aşağıdaki şartları sağlayan tanıyım işlemi on-line (çevrim içi) olarak tanımlanmaktadır :

- i. Özel bir giriş işaretine ihtiyaç yoktur .
- ii. Tüm bilgilerin saklanmasına gerek yoktur .
- iii. Her bir örnekleme anından sonra parametre kestirimi gerçeklemek için recursive bir algoritma kullanılmalıdır .
- iv. Model ayarlama için gerekli toplam hesaplama süresi , örnekleme süresinden küçük olmalıdır .

Büyük olasılıkla çevrimiçi metotla elde edilen parametre değerleri off-line (çevrimdışı) metotla bulunan değerler kadar doğru olamayabilir . Ancak pratikte çevrimdışı sistemdeki kadar veri toplamak için zaman harcanmayacak ve az sayıda veri ile yeterli doğrulukta sonuca ulaşıncaya kadar çalışılacaktır .

Çevrimiçi ve çevrimdışı sistemler için birçok metot kullanılmaktadır . Bunları sınıflandıracak olursak :

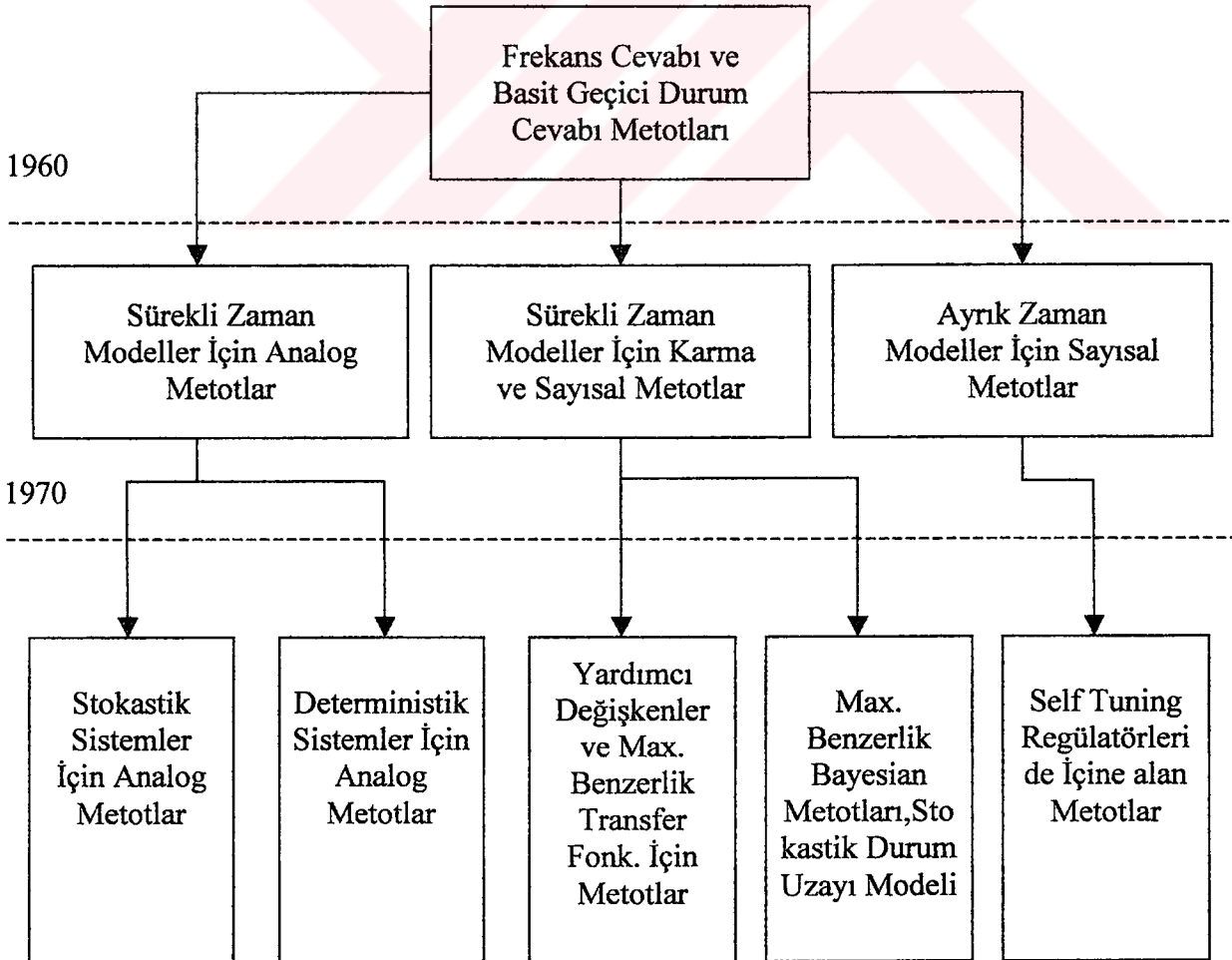
- 1) Klasik Metotlar (Çoğunlukla çevrimdışı)
 - i) Frekans cevabı metodu
 - ii) Adım cevabı metodu
 - iii) Dekonvolüsyon metodu ile darbe cevabı metodu
 - iv) Korelasyon fonksiyonu ile tanıyım metodu
- 2) Hata Eşitliği Yaklaşımı
 - i) En Küçük Kareler Metodu (Least Squares)
 - ii) Genelleştirilmiş En Küçük Kareler Metodu
 - iii) Maksimum Benzerlik Metodu
 - iv) Minimum Varyans Metodu
 - v) Eğim (Gradient) Metodu

3) Model Ayarlamalı Teknikler

- i) En Küçük Kareler Metodu (Recursive)
- ii) Genelleştirilmiş En küçük Kareler Metodu (Recursive)
- iii) Yardımcı Değişkenler Metodu
- iv) Bootstrap Metodu
- v) Maksimum Benzerlik Metodu (Recursive)
- vi) Korelasyon Metodu (Recursive)
- vii) Stokastik Yaklaşım

Bu metotların tarihsel gelişimini göstermek içinse şöyle bir çizelge vermemiz mümkündür:

Çizelge 2.1. Sistem belirlemenin tarihsel gelişimi



3. MATERYAL VE YÖNTEM

3.1. Sistem Belirleme Metotları

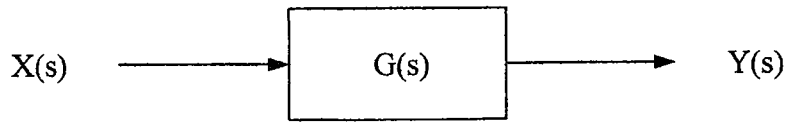
Takip eden bölümlerde sistem belirleme metotlarının birçoğu ele alınarak incelenecektir . Bu metotları klasik ve modern olmak üzere iki ana bölüme ayırmak mümkün olmaktadır .

3.1.1. Klasik Metotlar

Aşağıda yer alan metotların klasik olarak adlandırılmasının nedeni , modern metotlara nazaran çok daha uzun süredir biliniyor ve kullanılıyor olmalarıdır .

3.1.1.1. Frekans Cevabı Metodu

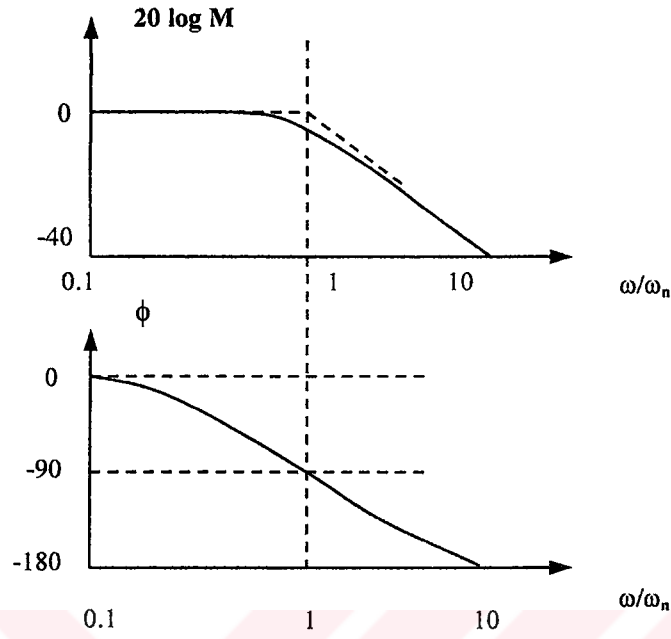
Lineer sistemlerin kestiriminde frekans cevabının kullanımı temelde Bode Diyagramı'na dayanmaktadır . Bu metotta sisteme sinüs şeklinde bir giriş işareti uygulanmakta ve kalıcı durum şartlarının oluşması beklenmektedir . Bu anda genlik oranları , yani kazanç ve giriş-çıkış arasındaki faz farkı ölçülür . Yapılan işlemlere , sinüzoidal işaretin frekansını ilgilenilen frekans aralığında değiştirerek devam edilmektedir . Aşağıdaki şekildeki gibi bir sistemi en genel ve basit anlamda gösterirsek ve sistemin transfer fonksiyonu $G(s)$ olmak üzere :



Şekil 3.1. Sürekli zamanda en genel haliyle bir sistemin blok diyagramı

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = M(j\omega).e^{j\phi(\omega)} \quad (3.1)$$

yazabiliriz . Bu denklemde M : kazancı , ϕ ise faz farkını göstermektedir . $M(\omega)$ ve $\phi(\omega)$ 'yı ω 'ya göre logaritmik olarak çizersek Bode Eğrileri' ni elde etmiş oluruz . Aşağıdaki şekilde örnek birer Bode eğrisi görülmektedir .



Şekil 3.2. Frekans cevabı metodunda kullanılan Bode Diyagramları

Bu eğrileri kullanarak sistemin doğru akım kazancı ve kırılma frekansları bulunabilmektedir . Doğruluğu arttırmak için eğrileri mümkün olduğunca hassas çizmek, bunun içinde mümkün olabildiğince farklı frekansta ölçüm almak gerekmektedir .

Bu metot sadece lineer zamanla değişmeyen sistemlerde ve off-line kestirimde kullanılabilir . Belirlenecek olan sistem birinci dereceden olduğunda bilinmeyen parametreleri bulmak kolay olmaktadır . Ancak yüksek mertebeden sistemler için bu oldukça zor gerçekleşmektedir .

3.1.1.2. Darbe Cevabı ile Kestirim (Birim Basamak Cevabı Yöntemi)

Sistem belirleme amacıyla kullanabileceğimiz en basit giriş işareti birim basamak girişidir . Teoride basit bir anahtarlama ile elde edilebilmesine rağmen , pratikte ideal bir darbe elde etmek mümkün değildir . Buna rağmen darbenin yükselme süresi , kestirim için gerekli olan en yüksek frekanstan yeterince küçük tutulabiliyorsa

doğruluğu yüksek olmaktadır . Çevrimdışı bir metot olmasına rağmen çevrimiçi kestirimde de kullanılabilir .

Eğer sistem modeli 1. dereceden ise iki bilgiye ihtiyaç vardır :

- i) Darbe girişe karşı steady-state (kararlı hal) cevabı
- ii) Zaman sabiti (darbenin genliğinin %10' dan %90' a çıkma süresi)

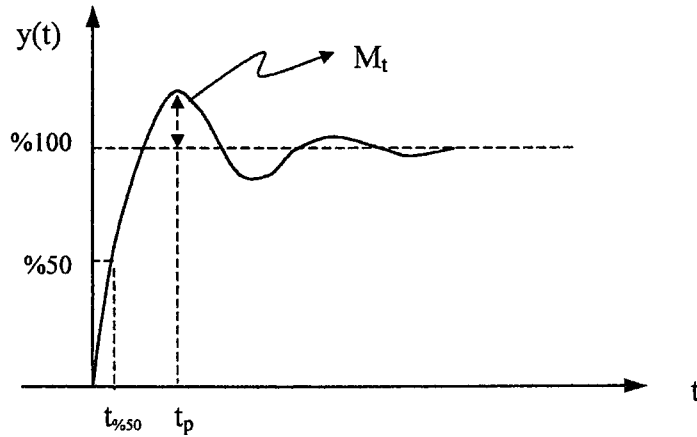
İkinci dereceden sistemler içinse (iki kutuplu) iki durum söz konusudur :

- i) İki kutbunda reel
- ii) İki kutbunda kompleks eşlenik çifti olması durumu

Sistemi belirleyebilmek için aşağıdaki ölçümlere ihtiyaç duyulmaktadır :

- i) Kararlı hal cevabı
- ii) Maksimum aşım miktarı
- iii) İlk maksimuma ulaşma süresi
- iv) Kararlı hal cevabı genliğinin %50' sine ulaştığı zaman

Bu değerleri bir çıkış işareti-zaman grafiği üzerinde gösterecek olursak ;



Şekil 3.3. İkinci dereceden bir sistem için birim basamak giriş için elde edilen çıkış işareti

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.2)$$

$$y(t) = 1 - \frac{1}{\beta} e^{-\xi\omega_n t} \cdot \cos(\omega_n \beta t - \phi) \quad (3.3)$$

Burada :

$$\beta = \sqrt{1 - \xi^2} \quad (3.4)$$

$$\phi = \text{Arctg} \frac{\xi}{\beta} \quad (3.5)$$

$$M_t = e^{-\xi \frac{\pi}{\beta}} = e^{-\xi \frac{\pi}{\sqrt{1-\xi^2}}} \quad (3.6)$$

$$t_p = \frac{\pi}{\beta \cdot \omega_n} = \frac{\pi}{\sqrt{1-\xi^2} \cdot \omega_n} \quad (3.7)$$

ve sonuçta :

$$y(t) = 1 - \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \cdot \cos(\omega_n \sqrt{1-\xi^2} t - \text{Arctg}(\frac{\xi}{\sqrt{1-\xi^2}})) \quad (3.8)$$

3.1.1.3. Dekonvolüsyon Metodu

Eğer giriş işareti basit darbe fonksiyonundan farklı ise sistemin giriş ve çıkışı arasında aşağıdaki gibi bir konvolüsyon integrali ilişkisi mevcuttur :

$$y(t) = \int_0^t x(\tau) \cdot w(t - \tau) \cdot d\tau \quad (3.9)$$

Burada ; $w(\tau)$ sistemin darbe cevabıdır . İşte yukarıdaki bağıntıda görüldüğü üzere ; giriş ve çıkış işaretlerinden darbe cevabının bulunması işlemine dekonvolüsyon adı verilmektedir . Darbe cevabı ($w(\tau)$) bir kere bulunduktan sonra sistemin transfer fonksiyonu yukarıdaki metotlardan birisiyle bulunabilir . Ancak sürekli zamanda bu

işlemi gerçekleştirmek zor olduğundan örnekleme zaman aralıkları eşit alınarak giriş ve çıkış işaretleri için aşağıdaki yaklaşımları yazabiliriz .

$$\left. \begin{array}{l} x(t) = x(nT) \\ y(t) = y(nT) \end{array} \right\} nT < t < (n+1)T, \quad n = 0,1,2,\dots \quad (3.10)$$

ve konvolüsyon integralinden :

$$\begin{aligned} y(T) &= x(0).w(0).T \\ y(2T) &= [x(0).w(T) + x(T).w(0)].T \\ &\dots\dots\dots \\ y(nT) &= T \sum_{i=1}^{n-1} x(iT).w(nT-iT-T) \end{aligned} \quad (3.11)$$

yazılabilir . Bu eşitlikleri matrissel formda yazacak olursak :

$$\tilde{y} = T.X.\tilde{w} \quad (3.12)$$

$$\tilde{y} = \begin{bmatrix} y(T) \\ \cdot \\ \cdot \\ y(nT) \end{bmatrix}, \quad \tilde{w} = \begin{bmatrix} w(0) \\ w(T) \\ \cdot \\ w(nT-T) \end{bmatrix}, \quad X = \begin{bmatrix} x(0) & 0 & \dots & 0 \\ x(T) & x(0) & & 0 \\ \cdot & & & \\ x(nT-T) & \cdot & \cdot & x(0) \end{bmatrix} \quad (3.13)$$

$x(0) \neq 0$ olmak şartı ile X matrisinin alt üçgen olma özelliğinden yararlanılarak W darbe cevabı aşağıdaki şekilde bir recursive algoritma olarak yazılabilir :

$$W(nT) = \frac{1}{x(0)} \left[\frac{1}{T}.y(nT+T) - \sum_{i=1}^n x(nT-iT) \right] \quad (3.14)$$

Bu denklemde sonuçların makul olabilmesi için T ' nin yeterince küçük olması gerekmektedir . İşlemleri kolaylaştırmak için giriş işaretini birim basamak fonksiyonu alırsak , aynı $u(iT)=1$ ise ;

$$W(nT) = \frac{1}{T} \cdot y(nT + T) - \underbrace{\sum_{i=0}^{n-1} w(iT)}_{h_n} \quad (3.15)$$

$$W(nT) = \frac{1}{T} \cdot y(nT + T) - h_n \quad (3.16)$$

ve

$$h_n = h_{n-1} + W(nT - T) \quad (3.17)$$

olmak üzere :

$$W(nT) = \frac{1}{T} \cdot [y(nT + T) - y(nT)] \quad (3.18)$$

3.1.1.4. Korelasyon (İlişki) Metodu

Bu metotta sisteme random bir işaret uygulanmasını temel alır . Eğer işaret beyaz gürültü ise bu çok kolay olmaktadır . Ancak pratikte beyaz gürültü oluşturmak mümkün değildir . Onun yerine tanıyım bölgesindeki frekanslarda spektral güç yoğunluğu sabit tutularak bir yaklaşıklık yapılır .

Doğrusal zamanla değişmeyen bir sistem için giriş-çıkış ifadesi :

$$y(t) = \int_0^{\infty} w(t - \tau) \cdot x(\tau) \cdot d\tau = \int_0^{\infty} w(\tau) \cdot x(t - \tau) \cdot d\tau \quad (3.19)$$

Giriş ve çıkış arasındaki cross correlation (çapraz ilişki) ise :

$$\phi_{yx}(\theta) = E[y(t) \cdot x(t - \theta)] = \int_0^{\infty} w(\tau) \cdot \phi_{xx}(\theta - \tau) \cdot d\tau \quad (3.20)$$

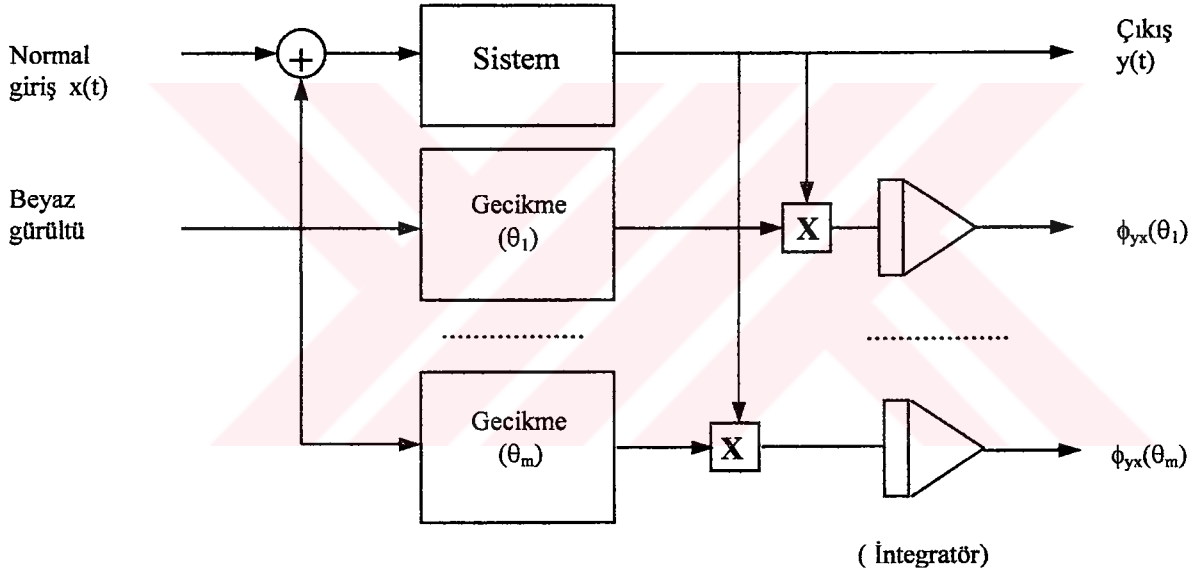
giriş beyaz gürültü olduğunda $\phi_{xx}(\theta - \tau) = \delta(\theta - \tau)$ (3.21)

olacağından ;

$$\phi_{yx}(\theta) = \int_0^{\infty} w(\tau) \cdot \delta(\theta - \tau) \cdot d\tau \quad \text{ise} \quad (3.22)$$

$$\phi_{yx}(\theta) = w(\theta) \quad (3.23)$$

Görüldüğü üzere : $\phi_{yx}(\theta)$, $t=0$ anındaki darbe cevabı ile aynı olmaktadır . Beyaz gürültü yerine pseudo-random gürültü (oto korelasyonu yaklaşık olarak delta-dirac fonksiyonudur .) kullanılabilir . Tanıyım için bu işaretle normal girişi süperpozisyon ile on-line olarak kullanmak mümkündür . Sistem lineer olduğundan çıkışların toplamı girişlerin toplamına bağlı olacaktır . Şöyle ki ;



Şekil 3.4. Korelasyon metodu için çıkışın süperpozisyon ile elde edilmesi

Burada beyaz gürültü ile normal giriş ve çıkış arasında bir ilişki yoktur . Bundan dolayı çapraz ilişki impulse cevabı olarak kabul edilir ve değeri :

$$\phi_{yx}(\theta) = \lim_{T \rightarrow \infty} \frac{1}{2\Pi} \left[\int_{-T}^T y(t) \cdot x(t - \theta) \cdot dt \right] \quad (3.24)$$

3.1.1.5. Darbe Cevabından Yararlanılarak Transfer Fonksiyonunun Bulunuşu

Eğer bir sistemin darbe cevabı bulunabiliyorsa (önceki bölümlerdeki gibi) bundan yararlanılarak önce ayrık zaman transfer fonksiyonu bulunup , ardından da sürekli zaman modeline geçilebilmektedir . Bunun için iki önemli kabul yapılmaktadır . Bunlardan birincisi sistemin darbe cevabının tam olarak bilinebilmesi (gürültüden bağımsız) , ikincisi de sistemin mertebesinin önceden biliniyor olmasıdır . Ayrık zaman modelinin , sürekli zaman modelinin özelliklerini tam olarak taşıyabilmesi için örnekleme periyodu (T) şu şartı sağlamalıdır :

$$P_k.T < 0.5 \quad (3.25)$$

Bu şart Nyquist Kriteri'ne göre yazılmıştır ve burada P_k , $G(s)$ ' in kutuplarından orijine en uzak olanıdır . Ayrık zaman domeninde n . Dereceden bir transfer fonksiyonunu aşağıdaki gibi gösterebiliriz :

$$H(z) = \frac{a_0 + a_1.z^{-1} + a_2.z^{-2} + \dots + a_n.z^{-n}}{1 + b_1.z^{-1} + b_2.z^{-2} + \dots + b_m.z^{-m}} \quad \text{yada} \quad (3.26)$$

$$H(z) = W_0 + W_1.z^{-1} + W_2.z^{-2} + \dots + W_m.z^{-m} \quad (3.27)$$

bu iki eşitliği karşı karşıya yazıp her iki taraftaki z ' lerin katsayılarını eşitlersek ;

$$a_0 + a_1.z^{-1} + \dots + a_n.z^{-n} = W_0 + (W_1 + b_1.W_0).z^{-1} + \dots + (W_n + \sum_{i=1}^n b_i.W_{n-i}).z^{-n} + \dots$$

$$\dots + (W_m + \sum_{i=1}^m b_i.W_{m-i}).z^{-m} \quad (m > n) \quad (3.28)$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ b_1 & 1 & 0 & \dots & 0 \\ b_2 & b_1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_n & b_{n-1} & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} W_0 \\ W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} \quad (3.29)$$

Burada b_i katsayılarını bildiğimizi varsayarsak a_i katsayılarını kolaylıkla bulabiliriz . b_i ' leri bulmak içinse $z^{-(n+1)}$, den z^{-2n} , e kadar olan terimleri göz önüne alarak şu bağıntı elde edilebilir :

$$\begin{bmatrix} W_1 & W_2 & \cdot & \cdot & W_n \\ W_2 & W_3 & \cdot & \cdot & W_{n+1} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ W_n & W_{n+1} & \cdot & \cdot & W_{2n-1} \end{bmatrix} \begin{bmatrix} b_n \\ b_{n-1} \\ \cdot \\ \cdot \\ b_1 \end{bmatrix} = \begin{bmatrix} -W_{n+1} \\ -W_{n+2} \\ \cdot \\ \cdot \\ -W_{2n} \end{bmatrix} \quad (3.30)$$

Buradan b_i katsayıları bulunur ve bir önceki ifadede yerine konulursa a_i parametreleri bulunabilir . Böylece sistemin ayrık zaman transfer fonksiyonu elde edilmektedir .

Sürekli zaman modeline geçiş içinse , ters z dönüşümü almak yeterlidir . z dönüşümünün özelliklerinden şunları yazabiliriz :

$$z = e^{sT} \quad \text{yada} \quad s = \frac{1}{T} \cdot \ln z \quad (3.31)$$

Dönüşüm işlemi şu sırayla gerçekleştirilebilir :

i) $(H(z) / z)$ ' in kısmi kesirlere ayırımı yapılır .

$$\frac{H(z)}{z} \rightarrow T(z) = \frac{A(z)}{B(z)} = \frac{C(z)}{D(z)} + \frac{E(z)}{F(z)} + \dots$$

ii) Her bir ifadenin ters z dönüşümü yapılarak $G(s)$ ifadesi bulunabilir .

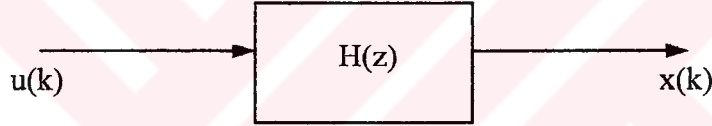
$$T(s) = Z^{-1} \{T(z)\} = Z^{-1} \left\{ \frac{A(z)}{B(z)} \right\} = Z^{-1} \left\{ \frac{C(z)}{D(z)} \right\} + Z^{-1} \left\{ \frac{E(z)}{F(z)} \right\} + \dots$$

3.1.2. MODERN METOTLAR

3.1.2.1. Çevrimdışı Metotlar

Bu bölümde tek giriş ve tek çıkışlı (SISO) sistemler için giriş-çıkış verilerinden, doğrusal model parametrelerinin kestirim (çevrimdışı) metotları ele alınacaktır . Bunun için öncelikle modelin derecesi bilinmelidir ve giriş-çıkış verileri düzenli bir şekilde elde edilebilmelidir . Bu metotlarla sadece doğrusal ayrık zaman parametreleri elde edilmektedir . Elde edilen parametrelerden daha sonra sürekli zaman modeline geçiş yapılabilmektedir .

3.1.2.1.1. En Küçük Kareler (Least Squares) Metodu



Şekil 3.5. Ayrık zaman domeni için bir sistemin blok diyagramı

Blok diyagramı yukarıda verilen devrenin giriş-çıkış ifadesi aşağıda verilmektedir :

$$\frac{x(z)}{u(z)} = H(z) = \frac{a_0 + a_1.z^{-1} + a_2.z^{-2} + \dots + a_m.z^{-m}}{1 + b_1.z^{-1} + b_2.z^{-2} + \dots + b_n.z^{-n}} \quad (3.32)$$

Transfer fonksiyonu şeklinde elde edilen yukarıdaki eşitliği , fark eşitliği olarak yazacak olursak ;

$$x_k = \sum_{i=0}^m a_i . u_{k-i} - \sum_{i=1}^n b_i . x_{k-i} \quad (3.33)$$

Burada ele alınan model doğrusal olup giriş-çıkış değerleri belirli bir T örnekleme periyoduyla elde edilmektedir . Bu verilerden faydalanılarak sistemin

parametreleri $a_0, a_1, \dots, a_m, b_1, b_2, \dots, b_n$ bulunabilir. Şöyle ki ; örneklenmiş verilerden bilinmeyen parametre sayısı kadar ($p=m+n+1$) denklem oluşturursak ve bunları matrisel bir formda yazarsak :

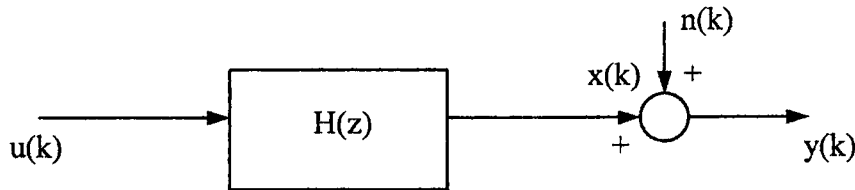
$$\underbrace{\begin{bmatrix} u_k & u_{k-1} & \cdot & u_{k-m} & -x_{k-1} & -x_{k-2} & \cdot & -x_{k-n} \\ u_{k+1} & u_k & & u_{k-m+1} & -x_{k-2} & -x_{k-1} & & -x_{k-n-1} \\ \cdot & & & \cdot & \cdot & \cdot & & \cdot \\ u_{k+p+1} & u_{k+p+2} & \cdot & u_{k+p-m-1} & -x_{k+p-2} & \cdot & \cdot & -x_{k+p-n-1} \end{bmatrix}}_{\phi_{\sim k}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ a_m \\ b_1 \\ \cdot \\ b_n \end{bmatrix}}_{\theta_{\sim}} = \underbrace{\begin{bmatrix} x_k \\ x_{k+1} \\ \cdot \\ x_{k+p+1} \end{bmatrix}}_{x_{\sim k}} \quad (3.34)$$

ise $\phi_k \cdot \theta = x_k$ (3.35)

Eğer ϕ_k matrisi tekil değilse , yani tersinde $\Delta \neq 0$ ise parametre matrisi :

$$\theta = \phi_k^{-1} \cdot x_k \quad (3.36)$$

şeklinde elde edilebilir . Eğer giriş ve çıkış büyüklükleri tam doğru olarak elde edilebiliyorsa , yani sisteme gürültü karışmıyorsa , yukarıdaki ifadeden parametreleri kolaylıkla bulmamız mümkündür . Ancak pratikte sisteme daima gürültü karıştığından bazı düzenlemeler yapılmaktadır . Sistemi gürültünün eklendiği haliyle yeniden gösterecek olursak :



Şekil 3.6. Çıkışına gürültü eklenmiş ayrık bir sistemin blok diyagramı

Şekilde n_k ile gösterilen işaret çıkıştaki ölçüm gürültüsünü göstermektedir . (3.33)' teki x_k ifadesini $y_k = x_k + n_k$ ifadesinde yerine koyarsak ;

$$y_k = \sum_{i=0}^m a_i \cdot u_{k-i} - \sum_{i=1}^n b_i \cdot y_{k-i} + v_k \quad (3.37)$$

denklemini elde ederiz . y_k ' yı matrissel formda ifade etmek istersek ;

$$\left. \begin{aligned} \varphi_k^T &= [u_k \quad u_{k-1} \quad \cdot \quad u_{k-m} \quad -y_{k-1} \quad -y_{k-2} \quad \cdot \quad -y_{k-n}] \\ \theta_k^T &= [a_0 \quad a_1 \quad \cdot \quad a_m \quad b_1 \quad b_2 \quad \cdot \quad b_n] \\ \varepsilon_k &= y_k - \varphi_k^T \cdot \theta \\ Y_p^T &= [y_k \quad y_{k+1} \quad \cdot \quad \cdot \quad y_{k+p-1}] \\ \phi^T &= [\varphi_1^T \quad \varphi_2^T \quad \cdot \quad \cdot \quad \varphi_p^T] \end{aligned} \right\} \quad (3.38)$$

Burada ;

- φ_k^T durum vektörü
- θ_k^T parametre vektörü
- ε_k hata vektörü
- Y_p^T ölçülebilen çıkış vektörü
- ϕ^T birleştirilmiş durum vektörü ' dür .

Sistem çıkışına gürültü de karıştığından sistem parametrelerinin belirlenebilmesi için $p = (m+n+1)$ ' den daha fazla sayıda eşitliğe ihtiyaç vardır .

N adet ε hata denklemini alt alta yazıp matrissel forma sokacak olursak ; p bilinmeyenli N adet denklem elde edilmiş olur .

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \cdot \\ \varepsilon_N \end{bmatrix} \quad (3.39)$$

Karesel hata fonksiyonu :

$$v(\theta) = \frac{1}{2} \sum_{t=1}^N \varepsilon_t^2 = \frac{1}{2} \cdot \varepsilon^T \cdot \varepsilon \quad (3.40)$$

şeklinde tanımlanır ve E.K.K. algoritmasında bu fonksiyonun değerini θ' ya göre minimum yapacak şekilde belirlenir . Denklemi minimum yapan değeri bulmak için aşağıdaki prosedür izlenir :

$$v(\theta) = \frac{1}{2} \cdot \varepsilon^T \cdot \varepsilon \quad \text{ve} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{bmatrix} = \begin{bmatrix} y_1 - \phi_1^T \cdot \theta \\ y_2 - \phi_2^T \cdot \theta \\ \vdots \\ y_N - \phi_N^T \cdot \theta \end{bmatrix} = Y - \phi \cdot \theta \quad \text{iken ;} \quad (3.41)$$

$$\begin{aligned} v(\theta) &= \frac{1}{2} (Y - \phi \cdot \theta)^T \cdot (Y - \phi \cdot \theta) \\ (1) \quad v(\theta) &= \frac{1}{2} (Y^T - (\phi \cdot \theta)^T) (Y - \phi \cdot \theta) \\ (2) \quad v(\theta) &= \frac{1}{2} (Y^T - \phi^T \cdot \theta^T) (Y - \phi \cdot \theta) \\ v(\theta) &= \frac{1}{2} (Y^T \cdot Y - Y^T \cdot \phi \cdot \theta - \theta^T \cdot \phi^T \cdot Y - \theta^T \cdot \phi^T \cdot \phi \cdot \theta) \end{aligned} \quad (3.42)$$

NOT : (1) ve (2) nolu denklemlerdeki ara işlemler aşağıdaki matematiksel ifadeler kullanılarak gerçekleştirilmiştir .

$$\begin{aligned} (A+B)^T &= A^T + B^T \\ (A \cdot B)^T &= B^T \cdot A^T \end{aligned}$$

Denklemi minimum yapan değeri bulmak için , θ' ya göre türev alıp sifıra eşitlersek :

$$\frac{d}{d\theta} v(\theta) = \frac{1}{2} [-\phi^T \cdot Y - \phi^T \cdot Y + \phi^T \cdot \phi \cdot \theta + \phi^T \cdot \phi \cdot \theta] \quad (3.43)$$

NOT : yukarıdaki ifade $\frac{d}{d\theta} A.\theta = A^T$ ve $\frac{d}{d\theta} \theta^T .A = A$ özellikleri kullanılarak elde edilmiştir .

$$0 = -2.\phi^T .Y + 2.\phi^T .\phi.\theta \quad \text{olmak üzere}$$

$$\hat{\theta} = (\phi^T .\phi)^{-1} .(\phi^T .Y) \quad (3.44)$$

elde edilebilir . Bu sonucu daha da genelleştirecek olursak :

$$\theta_k = \left[\sum_{n=1}^k \phi_n \phi_n^T \right]^{-1} \cdot \left[\sum_{n=1}^k \phi_n .y_n \right] \quad (3.45)$$

denklemini elde etmek mümkündür . Bu ifade sayesinde recursive bir kestirim yapmak mümkün olmaktadır .

3.1.2.1.2. Maksimum Benzerlik Yöntemi

Şekil 3.6' da olduğu gibi $\{v_k\}$ ' yi bilinmeyen θ parametresine bağlı , gelişigüzel bir ayrık değişken işaret olarak kabul edelim . Buradan da olasılık yoğunluk fonksiyonu' nu_ pdf _ “ $f(v;\theta)$ ” hesaplayabildiğimizi görürüz . n adet birbirinden bağımsız v_1, v_2, \dots, v_n gözleminin alındığını kabul edip , bunlardan θ ' nın en iyi kestirimini yapmaya çalışalım . Bunun için bir benzer fonksiyon tanımlayalım . Benzer fonksiyon genellikle v_i ' nin bağıl olasılık yoğunluk fonksiyonu olarak tanımlanır . Tüm v_i ' lerin birbirinden ilişkisiz olduğunu kabul edersek :

$$L(v_1, v_2, \dots, v_n; \theta) = f(v_1; \theta).f(v_2; \theta) \dots f(v_n; \theta) \quad (3.46)$$

olur. eşitliğin sağ tarafı olasılık yoğunluk fonksiyonlarının çarpımıdır . Burada L' yi maksimum yapmaktansa ($\log L$)' yi maksimum yapmak daha kolaydır . Böylece θ ' nın maksimum benzerlik kestirimi ($\hat{\theta}_{ML}$) aşağıdaki ifadeyle bulunabilir :

$$\frac{\partial}{\partial \theta} (\log L) \Big|_{\theta=\theta_{ML}} = 0 \quad (3.47)$$

Bu ifadeyi sistem tanımına uyguladığımızda ; $\{v_i\}$ dizisini sıfır ortalamalı Gaussian gürültü olarak ve u_i ' den ilişkisiz kabul ettiğimizde , benzer fonksiyonunu şöyle tanımlayabiliriz :

$$L[y_p; \theta] = (2\pi\sigma^2)^{-p/2} \cdot \exp \left[-\frac{1}{2} \cdot \frac{(y_p - A_p \theta)^T \cdot (y_p - A_p \theta)}{\sigma_{ML}^2} \right] \quad (3.48)$$

$$\log L[y_p; \theta] = -\frac{p}{2} \cdot \log 2\pi - \frac{p}{2} \cdot \log \sigma^2 - \frac{1}{2} \cdot \frac{(y_p - A_p \theta)^T \cdot (y_p - A_p \theta)}{\sigma_{ML}^2} \quad (3.49)$$

Bu denklemde θ ve σ ' ya göre kısmi türev alacak olursak :

$$\frac{\partial}{\partial \theta} \log L[y_p; \theta] = -\frac{1}{2\pi\sigma_{ML}^2} \left[-A_p^T \cdot (y_p - A_p \theta) - A_p \cdot (y_p^T - \theta^T \cdot A_p^T) \right] = 0 \quad (3.50)$$

$$\Rightarrow \frac{A_p^T \cdot A_p \cdot \theta - A_p^T \cdot y_p}{\sigma_{ML}^2} = 0 \quad (3.51)$$

ve

$$\frac{\partial}{\partial \sigma_{ML}} \log L[y_p; \theta] = -\frac{p}{2} \cdot \frac{1}{\sigma_{ML}^2} \cdot 2\sigma_{ML} - \frac{1}{2} \cdot 2\sigma_{ML} \cdot \frac{(y_p - A_p \theta)^T \cdot (y_p - A_p \theta)}{\sigma_{ML}^4} = 0$$

$$\Rightarrow -\frac{p}{2\sigma_{ML}^2} + \frac{\theta}{2\sigma_{ML}^2} = 0 \quad (3.52)$$

(3.51) denkleminde ve :

$$\theta = (y_p - A_p \cdot \theta_{ML}^{\wedge})^T \cdot (y_p - A_p \cdot \theta_{ML}^{\wedge}) = y_p^T \cdot y_p \quad \text{'den} \quad (3.53)$$

$$\theta_{ML} = (A_p^T \cdot A_p)^{-1} \cdot (A_p \cdot y_p) \quad (3.54)$$

Bu sanki bir özel durumu yansıtmaktadır . (yani sıfır ortalamalı beyaz gürültü gibi) Asıl gerçekçi durum olan gürültülünün beyaz gürültüden farklı olduğu durumu göz önüne alacak olursak :

$$v(k) = \sum_{i=1}^r c_i \cdot z_{k-i} + z_k \quad (3.55)$$

Burada $\{z_k\}$, giriş işareti $\{u_i\}$ ' dan ilişkisiz , sıfır ortalamalı beyaz gürültü dizisidir . Görüldüğü üzere v_k artık bir beyaz gürültü olamaz . Bu durumda $\{v_i\}$ 'nin kovaryans matrisi :

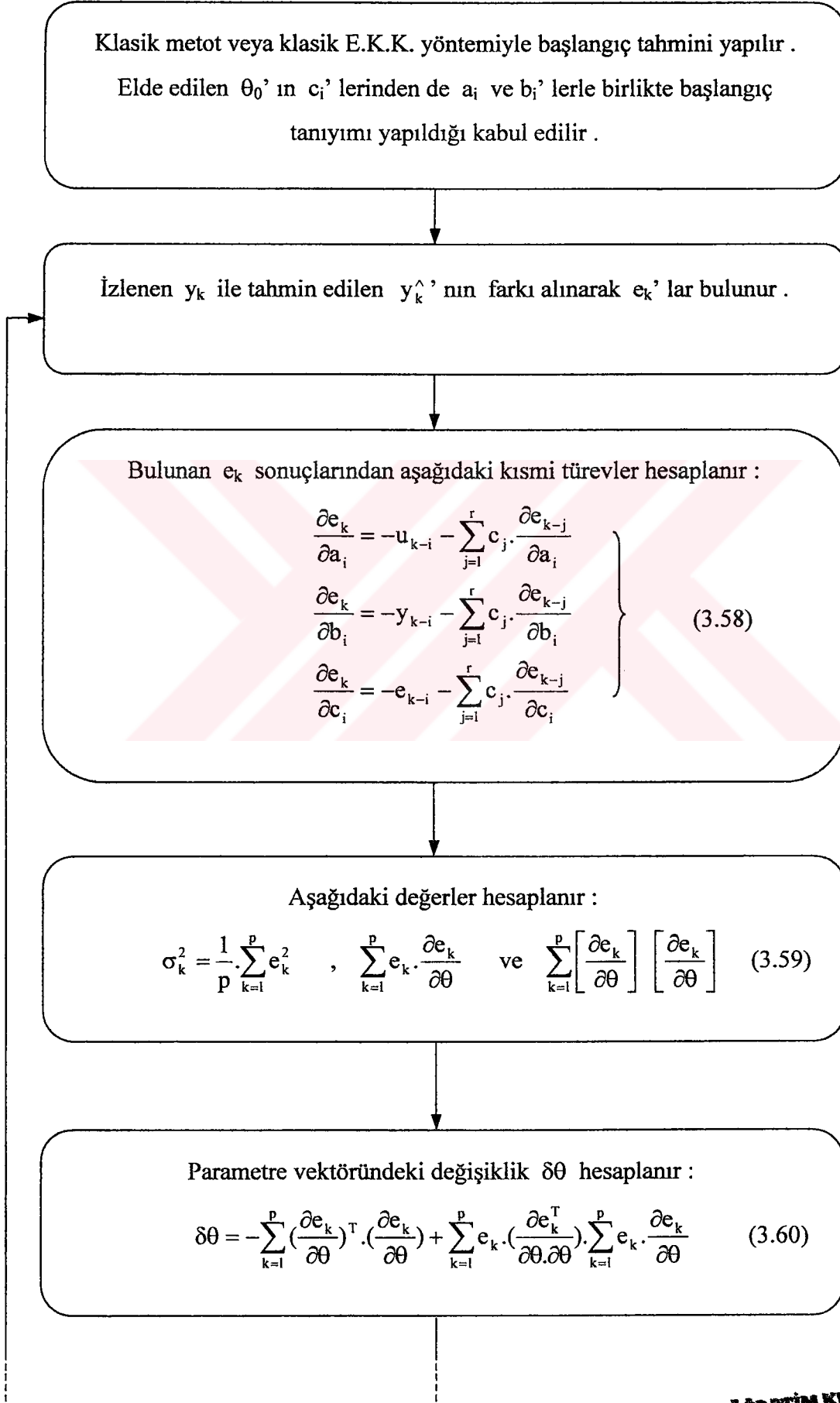
$$R = E[v_p \cdot v_p^T] \quad (3.56)$$

ve logaritmik benzerlik fonksiyonumuzda ;

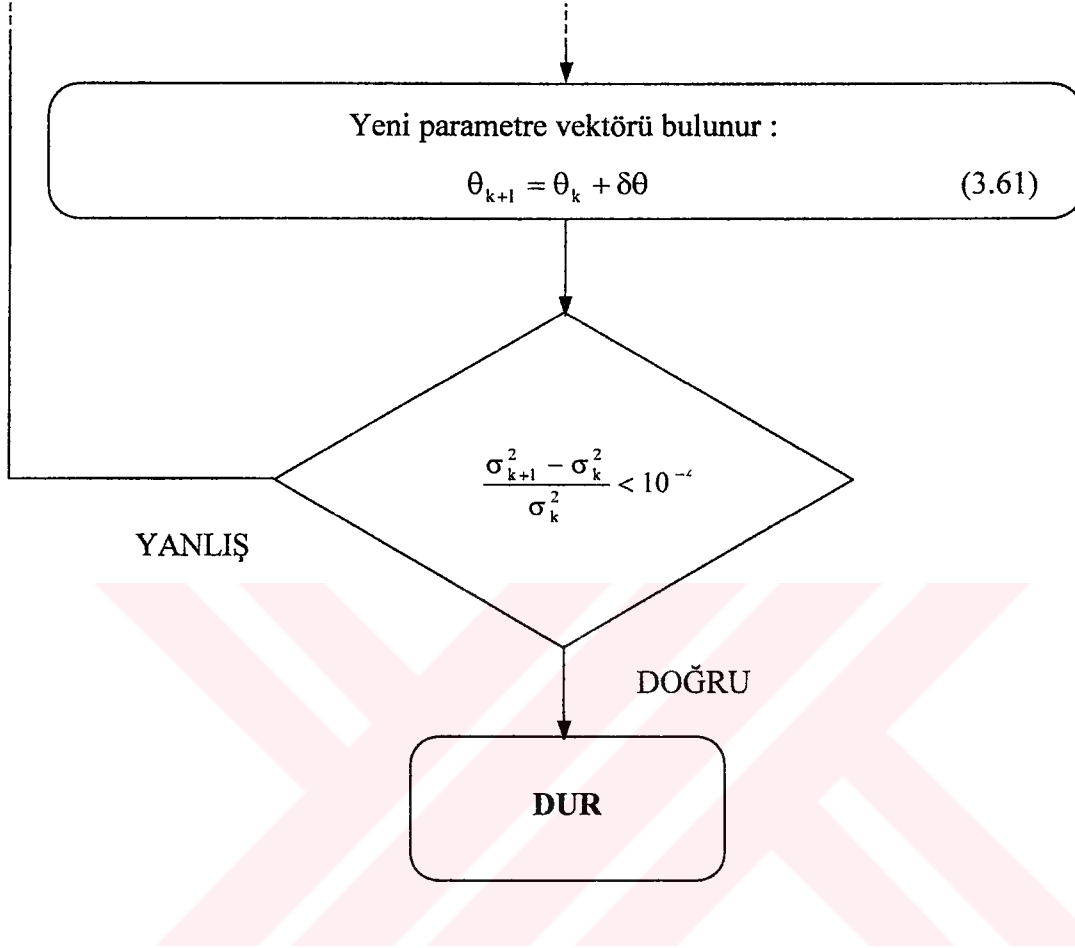
$$\log L[y_p; \theta] = -\frac{p}{2} \cdot \log 2\pi - \frac{1}{2} \cdot \log(\det R) + \frac{1}{2} \cdot (y_p - A_p \cdot \theta)^T \cdot R^{-1} \cdot (y_p - A_p \cdot \theta) \quad (3.57)$$

olur . Yukarıdaki ifade bir önceki benzerlik fonksiyonuna çok benzemesine rağmen (R^{-1} ve $\det R$) terimi yüzünden bir çözüm elde edilemez . Bunu gidermek için Aström ve Eykhoff' un (1971) geliştirdiği iteratif çözüm şeması kullanılmaktadır . Bu şema akış diyagramı şekline aşağıdaki çizelgede görülmektedir :

Çizelge 3.1. Maksimum Benzerlik Metodunun İteratif Çözüm Şeması



Çizelge 3.1. (Devam) Maksimum Benzerlik Metodunun İteratif Çözüm Şeması



3.1.2.1.3. Genelleştirilmiş En Küçük Kareler Yöntemi

Bu yöntem de hatanın karesinin ortalama değerini minimize edecek sistem ve gürültü parametrelerini kestiren iteratif bir yöntemdir . Gürültü modelinin parametrelerinden oluşan yardımcı parametre vektörü :

$$\psi = [c_1 \quad c_2 \quad \dots \quad c_r]^T \quad (3.62)$$

ve gürültü için modelimizi de ;

$$B_p = \begin{bmatrix} v_{k-1} & v_{k-2} & \cdot & v_{k-r} \\ v_k & v_{k-1} & \cdot & v_{k-r+1} \\ \cdot & \cdot & \cdot & \cdot \\ v_{k+p-2} & v_{k+p-3} & \cdot & v_{k-r+p-1} \end{bmatrix} \quad (3.63)$$

olarak alırsak : Hsia (1976)'nın geliřtirdiđi algoritma ařađıdaki gibidir :

$$\left. \begin{aligned} y_p &= A_p \cdot \theta + v_p \\ v_p &= B_p \cdot \psi + z_p \end{aligned} \right\} \quad (3.64)$$

$$y_p = \begin{bmatrix} A_p & B_p \end{bmatrix} \begin{bmatrix} \theta_p \\ \psi_p \end{bmatrix} + z_p \quad (3.65)$$

Böylece E.K.K. metodunu ařađıdaki gibi uygulamak mümkündür :

$$\begin{bmatrix} \theta_p \\ \psi_p \end{bmatrix} = \begin{bmatrix} A_p^T \cdot A_p & A_p^T \cdot B_p \\ B_p^T \cdot A_p & B_p^T \cdot B_p \end{bmatrix}^{-1} \begin{bmatrix} A_p^T \\ B_p^T \end{bmatrix} \cdot y_p \quad (3.66)$$

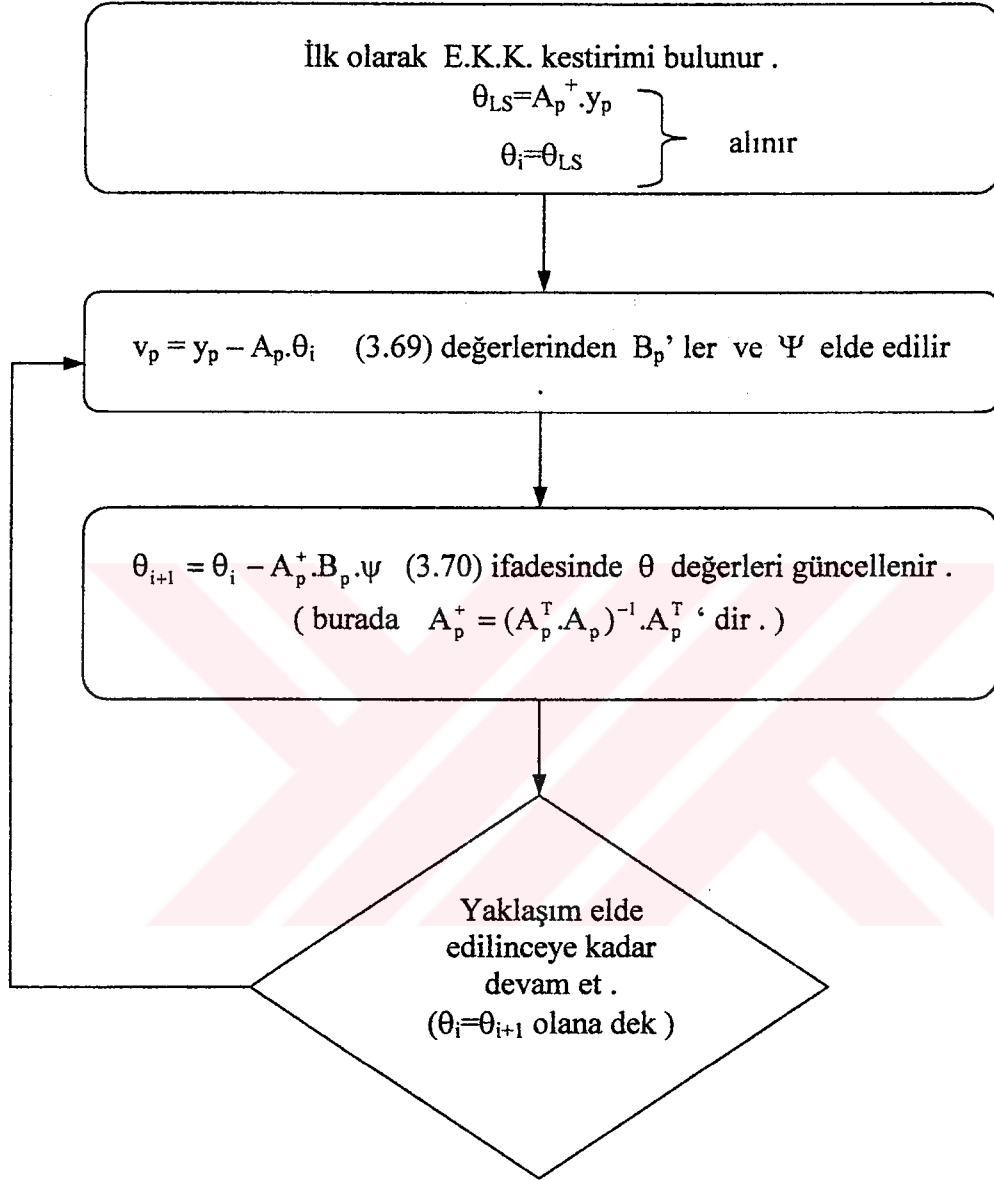
Buradan :

$$\theta_p = (A_p^T \cdot A_p)^{-1} \cdot (A_p^T \cdot y_p) - (A_p^T \cdot A_p)^{-1} \cdot (A_p^T \cdot B_p \cdot \psi_p) \quad (3.67)$$

$$\psi_p = (B_p^T \cdot M \cdot B_p)^{-1} \cdot B_p^T \cdot (I - A_p \cdot (A_p^T \cdot A_p)^{-1} \cdot A_p^T) \cdot y_p \quad (3.68)$$

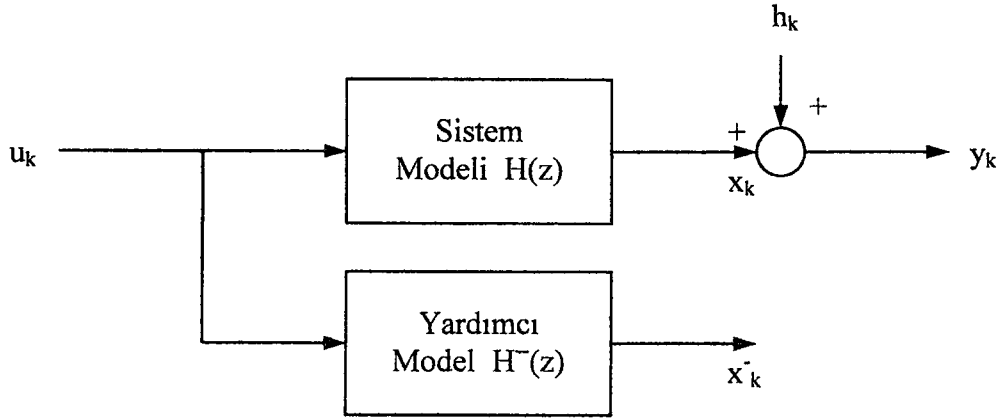
θ_p eřitliđinde sađdaki ilk terim görüldüđü üzere klasik E.K.K. tahminidir . İkinci terim ise bu tahmini düzelten kısımdır . Bu algoritmayı gerçeklemek için ařađıdaki řemadaki adımlar takip edilebilir :

Çizelge 3.2. Genelleştirilmiş E.K.K. için kullanılan algoritmanın akış diyagramı



Bu metotta daha fazla gelişme yaratılmayınca kadar Ψ ve θ değerleri güncelleştirilmeye devam edilir . Ancak sinyal / gürültü oranının düşük olduğu lokal minimumların olduğu zamanlar bu metot optimum çözüm için yakınsamayabilir .

3.1.2.1.4. Yardımcı Değişkenler (Instrumental Variables) Metodu



Şekil 3.7. Yardımcı Değişkenler Metodu için ; bir sistemin genel blok diyagramı

Bu metot daha az hesap yapılan ve daha az karmaşık olan , nispeten klasik E.K.K. kadar basit bir metottur . Tabii ki basitlik doğruluğu düşürmektedir . (3.64) eşitliğini yeniden yazacak olursak :

$$y_p = A_p \cdot \theta + v_p \quad (3.71)$$

A_p ile aynı boyutta bir Z matrisi oluşturalım . Öyle ki aşağıdaki koşulları sağlasın .

$$\begin{aligned} E[Z^T \cdot v_p] &= 0 \\ E[Z^T \cdot A_p] &= Q \end{aligned} \quad (3.72)$$

Burada Q nonsingüler bir matristir . Ardından y_p ifadesinin her iki yanını Z^T ile çarpacak olursak :

$$Z^T \cdot y_p = Z^T \cdot A_p \cdot \theta + Z^T \cdot v_p \quad (3.73)$$

Buradan parametre vektörü θ :

$$\theta_{IV} = (Z^T \cdot A_p)^{-1} \cdot (Z^T \cdot y_p) \quad (3.74)$$

şeklinde elde edilir . Z yardımcı değişkenler matrisini $Z = W \cdot A_p$ olarak tanımladığımızda yöntem Ağırlıklı E.K.K. olmaktadır . Ancak z_i ' leri belirlerken istenen iki şart mevcuttur . Bunlar ; z_i ' lerin gürültüden ilişkisiz ve A_p ile ilişkili olmasıdır . Bunun için en uygunu girişi gerçek giriş ile aynı , fakat çıkışı gürültüsüz olan bir sistemdir . Buna uygun olan Z matrisi şöyle verilebilir :

$$Z = \begin{bmatrix} u_k & u_{k-1} & \cdot & u_{k-m} & -X_{k-1}^- & -X_{k-2}^- & \cdot & -X_{k-m}^- \\ u_{k+1} & u_k & \cdot & u_{k-m+1} & -X_k^- & \cdot & \cdot & -X_{k-m+1}^- \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_{k+p+1} & u_{k+p+2} & \cdot & u_{k+p-m-1} & -X_{k+p-2}^- & \cdot & \cdot & -X_{k+p-m-1}^- \end{bmatrix} \quad (3.75)$$

3.1.2.2. ÇEVİRİMİÇİ METOTLAR

3.1.2.2.1. En Küçük Kareler Metodu'nun Çevrimiçi Algoritması

Çevrimdışı E.K.K.' dan bahsederken ele aldığımız :

$$\theta = (\varphi^T \cdot \varphi)^{-1} \cdot (\varphi^T \cdot Y) \quad (3.76)$$

algoritması bütün veriler sistemden toplandıktan sonra kullanılmaktadır . Çevrimiçi kestirim için bu ifade ile eşdeğer olan aşağıdaki ifade kullanılarak ; çevrimiçi E.K.K. türetilebilir .

$$\hat{\theta}_k = \underbrace{\left[\sum_{n=1}^k \varphi_n \cdot \varphi_n^T \right]^{-1}}_{P_k} \cdot \left[\sum_{n=1}^k \varphi_n \cdot y_n \right] \quad (3.77)$$

Bu denklemde soldaki ilk ifadeye (tersi alınmış haliyle) P_k dersek ;

$$P_k^{-1} = \sum_{n=1}^k \varphi_n \varphi_n^T \quad (3.78)$$

ve bunu recursive olarak yazacak olursak ;

$$P_k^{-1} = \underbrace{\sum_{n=1}^{k-1} \varphi_n \varphi_n^T}_{P_{k-1}^{-1}} + \varphi_k \varphi_k^T \quad (3.79)$$

$$P_k^{-1} = P_{k-1}^{-1} + \varphi_k \varphi_k^T \quad (3.80)$$

(3.77)' deki ifadeyi de aşağıdaki gibi düzenlediğimizde :

$$\theta_k = P_k \cdot \left[\sum_{n=1}^{k-1} \varphi_n \cdot y_n + \varphi_k \cdot y_k \right] \quad (3.81)$$

Yine (3.77) ifadesinden :

$$\theta_{k-1} = \underbrace{\left[\sum_{n=1}^{k-1} \varphi_n \cdot \varphi_n^T \right]^{-1}}_{P_{k-1}^{-1}} \cdot \left[\sum_{n=1}^{k-1} \varphi_n \cdot y_n \right]$$

olmak üzere
$$\sum_{n=1}^{k-1} \varphi_n \cdot y_n = P_{k-1}^{-1} \cdot \theta_{k-1} \quad (3.82)$$

elde edilir ve bu ifadeyi (3.81)'de yerine yazacak olursak ;

$$\theta_k = P_k \cdot [P_{k-1}^{-1} \cdot \theta_{k-1} + \varphi_k \cdot y_k] \quad (3.83)$$

bulunur . Bu ifadede (3.80) bağıntısından elde edilen P_{k-1}^{-1} ifadesini yazarsak :

$$\theta_k = P_k \cdot [(P_k^{-1} - \varphi_k \varphi_k^T) \cdot \theta_{k-1} + \varphi_k \cdot y_k] \quad (3.84)$$

$$\theta_k = \theta_{k-1} + P_k \cdot \varphi_k \cdot (y_k - \varphi_k^T \cdot \theta_{k-1}) \quad (3.85)$$

bulunabilir . Ancak görüldüğü üzere denklemin son halinde P_k ifadesi bulunmaktadır . Bu da demektir ; ifade içerisinde $\varphi_n \varphi_n^T$ çarpım toplamının tersi , çarpım olarak gelecektir . Matris tersi almak, algoritmayı gerçekleştireceğimiz bilgisayar programları için güç bir olaydır . Bu sorunu gidermek için aşağıdaki gibi bir matris tersi algoritması ile yaklaşık bir sonuç elde edilmektedir .

Matrisel bir denklemi aşağıdaki formda yazabiliyorsak ;

$$P_2^{-1} = P_1^{-1} + H^T \cdot R^{-1} \cdot H \quad (3.86)$$

$$\Rightarrow P_2 = P_1 - P_1 \cdot H^T \cdot (H \cdot P_1 \cdot H^T + R)^{-1} \cdot H \cdot P_1 \quad (3.87)$$

(3.80) denklemindeki P_k^{-1} ifadesine bakacak olursak (3.86) denklemine benzediğini görmekteyiz . O halde bu algoritma' dan P_k değerini şöyle yazabiliriz :

$$P_k = P_{k-1} - P_{k-1} \cdot \varphi_k \cdot (\varphi_k \cdot P_{k-1} \cdot \varphi_k^T)^{-1} \cdot \varphi_k^T \cdot P_{k-1} \quad (3.88)$$

Düzenleyecek olursak :

$$P_k = P_{k-1} - \frac{P_{k-1} \cdot \varphi_k \cdot \varphi_k^T \cdot P_{k-1}}{1 + \varphi_k^T \cdot P_{k-1} \cdot \varphi_k} \quad (3.89)$$

Görüldüğü gibi bu algoritma sayesinde matris tersi işlemi ortadan kaldırılmaktadır. Böylece program içerisinde kolaylıkla uygulanabilmektedir .

3.1.2.2.3. Sistem Belirleme Parametrelerinin Yakınsaması _ Uygunluğu

Sistem belirlemede , tahmin edilen parametrelerin ($\hat{\theta}_p$) geçen belirli bir zaman zarfında , gerçek parametre değerlerine (θ) yakınsaması istenir . Bu özelliğe uygunluk (consistency) adı verilmektedir . Parametrelerin bulunmasında önemli bir

özelliğidir . θ ile $\hat{\theta}_p$ arasında aşağıdaki gibi bir özellik sağlanıyorsa parametreleri tahminleri uygundur .

$$\lim_{p \rightarrow \infty} E[(\hat{\theta}_p - \theta)] = 0 \quad (3.90)$$

$E[(\hat{\theta}_p - \theta)]$, iki değer arasındaki farkın beklenen değerinin göstermektedir . Eğer bu eşitlik tüm $p > m+n+1$ değerleri için sağlanmıyor , aralarında bir fark kalıyorsa . yani ;

$$\lim_{p \rightarrow \infty} E[(\hat{\theta}_p - \theta)] = R \quad (3.91)$$

denklemindeki gibi sağ tarafa bir R reel sayısı geliyorsa bu parametre tahminleri doğru değildir . Oluşan bu farkın , kullanılan yonteme bağıli olarak mümkün olan en küçük değerde olması , parametre belirlemenin başlıca amaçlarından birisidir .

Yapılan tahminlerin uygunluğunu birçok faktör etkileyebilmektedir . Bunlardan birkaçını sıralayacak olursak :

- Model seçimi
- Parametre belirleme metodu
- Sistem özellikleri
- Gürültünün niteliği vs...

3.1.2.2.3. En Küçük Kareler Metodunun Uygunluk Analizi

En Küçük Kareler yaklaşımıyla parametrelerin minimum halini bulan denklemi yeniden yazacak olursak :

$$\hat{\theta}_N = \left[\sum_{k=1}^N \varphi_k \varphi_k^T \right]^{-1} \left[\sum_{k=1}^N \varphi_k \cdot y_k \right] \quad (3.92)$$

ve
$$y(k) = \varphi(k) \cdot \theta + e(k) \quad (3.93)$$

Burada : θ : doğru (gerçek) parametre vektörü
 $e(k)$: k anında çıkışa ilave edilen ölçme gürültüsü' dür .

çıkış ifadesini ilk ifadede yeniden yazacak olursak ;

$$\hat{\theta}_N = \left[\sum_{k=1}^N \varphi_k \cdot \varphi_k^T \right]^{-1} \cdot \left[\sum_{k=1}^N \varphi_k \cdot (\varphi_k \cdot \theta + e_k) \right] \quad (3.94)$$

$$\Rightarrow E[\hat{\theta}_N] = \underbrace{\left[\frac{1}{N} \cdot \sum_{k=1}^N \varphi_k \cdot \varphi_k^T \right]^{-1}}_{R(N)} \cdot \left[\frac{1}{N} \cdot \sum_{k=1}^N \varphi_k \cdot (\varphi_k \cdot \theta + e_k) \right]$$

$$\Rightarrow E[\hat{\theta}_N] = R_{(N)}^{-1} \cdot R_{(N)} \cdot \theta + R_{(N)}^{-1} \cdot \left[\frac{1}{N} \cdot \sum_{k=1}^N \varphi_k \cdot e_k \right]$$

$$\Rightarrow E[\hat{\theta}_N] = \theta + R_{(N)}^{-1} \cdot \left[\frac{1}{N} \cdot \sum_{k=1}^N \varphi_k \cdot e_k \right] \quad (3.95)$$

$\hat{\theta}_N$ denkleminin çözümünün olabilmesi için $R_{(N)}$ matrisinin determinantının sıfırdan farklı olması gereklidir . Ya da başka bir anlatımla $R_{(N)}$ matrisi nonsingüler (tekil olmayan) matris olmalıdır . Bu özellik birçok durumda sağlanır . İstenilen bu ilk özellik sağlandıktan sonra parametrelerin gerçek değere yakınsaması için denklemin sağ tarafında yer alan , $R_{(N)}^{-1} \cdot \left[\frac{1}{N} \cdot \sum_{k=1}^N \varphi_k \cdot e_k \right]$ ifadesinin sıfır olması gereklidir . Az önceki durumun aksine bu özellik birçok durumda sağlanmaktadır .

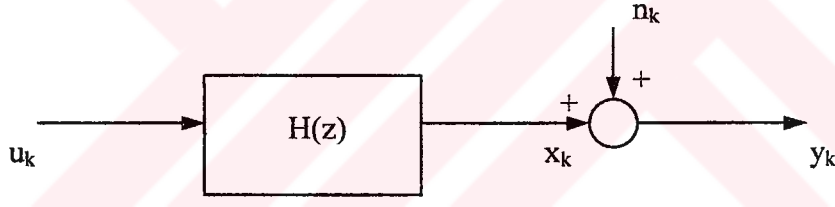
Yapılan tahminlerin uygun olabilmesi için gerekli şartları yazacak olursak ; $E[\varphi_k \cdot \varphi_k^T]$ denkleminin nonsingüler olması ve $E[\varphi_k \cdot e_k] = 0$ eşitliğinin sağlanması şartı ortaya çıkar . $E[\varphi_k \cdot \varphi_k^T]$ ' nin nonsingüler olmasının gerçekleşmediği durumlar şunlardır :

- Giriş , sistemi sürekli olarak uyarıyor olabilir .
- Elde edilen veriler gürültüsüz ve transfer fonksiyonu ortak çarpanlardan ibaret olabilir.
- Giriş , çıkıştan doğrusal bir geribesleme ile elde edilebilir .

$E[\varphi_k \cdot e_k] = 0$ denklemi ise ancak e_k 'nin beyaz gürültü olması halinde doğrudur . Bu şart altında e_k bütün eski verilerden ilişkisiz olacağı gibi aynı zamanda φ_k ile de ilişkisiz olur ve $E[\varphi_k \cdot e_k]$ değeri sıfır olacaktır .

Eğer e_k beyaz gürültü değil ise ; e_k 'nin eski değerleri ile ve dolayısıyla φ_k 'nin eski değerleriyle ilişkili olacağından $E[\varphi_k \cdot e_k]$ değeri sıfır olmayacaktır . Sonuç olarakta parametre tahminleri uygunluğunu yitirecektir .

3.1.2.2.2. Ağırlıklı En Küçük Kareler Metodu (Çevrimiçi WLS)



Şekil 3.8. Ağırlıklı E.K.K. için ayrık zaman sistem modeli blok diyagramı

Ele alınan sistem modeli yukarıdaki gibi olduğunda çevrimdışı E.K.K. için kullandığımız giriş-çıkış ilişkisi aşağıdaki gibi olmaktadır :

$$y_k = \sum_{i=0}^m a_i \cdot u_{k-i} - \sum_{i=1}^n b_i \cdot y_{k-i} + v_k \quad (3.96)$$

$$v_k = n_k + \sum_{i=1}^n b_i \cdot n_{k-i} \quad \text{olmak üzere}$$

$$y_k = \phi_k^T \cdot \theta + v_k \quad \text{olur.} \quad (3.97)$$

Burada ;

$$\phi_k^T = [u_k \quad u_{k-1} \quad \dots \quad u_{k-m} \quad -y_{k-1} \quad -y_{k-2} \quad \dots \quad -y_{k-n}] \quad (3.98)$$

$$\theta = [a_0 \quad a_1 \quad \dots \quad a_m \quad b_1 \quad b_2 \quad \dots \quad b_n]^T \quad (3.99)$$

olmaktadır . Sistemden p adet örnek aldığımızı varsayarsak :

$$y_p = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+p-1} \end{bmatrix} \quad A_p = \begin{bmatrix} \phi_k^T \\ \phi_{k+1}^T \\ \vdots \\ \phi_{k+p-1}^T \end{bmatrix} \quad v_p = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+p-1} \end{bmatrix} \quad (3.100,3.101,3.102)$$

yukarıdaki eşitliklerden çevrimdışı ağırlıklı E.K.K.' yı yazacak olursak ;

$$\theta_p = (A_p^T \cdot W_p \cdot A_p)^{-1} \cdot (A_p^T \cdot W_p \cdot y_p) \quad (W = \text{diag}[w_1 \quad w_2 \quad \dots \quad w_p]) \quad (3.103)$$

ve bunlara bir veri grubu daha eklersek ;

$$A_{p+1} \stackrel{\Delta}{=} \begin{bmatrix} A_p \\ \phi_{k+p}^T \end{bmatrix} \quad y_{p+1} \stackrel{\Delta}{=} \begin{bmatrix} y_p \\ y_{k+p} \end{bmatrix} \quad W_{p+1} \stackrel{\Delta}{=} \begin{bmatrix} W_p & 0 \\ 0 & \tilde{W}_{p+1} \end{bmatrix} \quad (3.104,3.105,3.106)$$

Bu durumda; $\theta_{p+1} = (A_{p+1}^T \cdot W_{p+1} \cdot A_{p+1})^{-1} \cdot (A_{p+1}^T \cdot W_{p+1} \cdot y_{p+1}) \quad (3.107)$

halini alacaktır . θ_{p+1} ile θ_p arasındaki ilişkiyi çıkartmak için eşitliğe $(p+1)$. verileri ilave olarak yazacak olursak ;

$$\theta_{p+1}^{\wedge} = (A_p^T \cdot W_p \cdot A_p + \phi_{k+p}^T \cdot W_{p+1} \cdot \phi_{k+p}^T)^{-1} \cdot (A_p^T \cdot W_p \cdot y_p + \phi_{k+p}^T \cdot W_{p+1} \cdot y_{k+p}) \quad (3.108)$$

$$P_p = (A_p^T \cdot W_p \cdot A_p)^{-1} \quad (3.109) \text{ tanımlamasını yaparak} \quad \theta_p^{\wedge} = P_p \cdot A_p^T \cdot W_p \cdot y_p \quad (3.110)$$

$$\mathbf{P}_{p+1}^{-1} = \mathbf{P}_p^{-1} + \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \boldsymbol{\phi}_{k+p}^T \quad (3.111)$$

$$\begin{aligned} &= \mathbf{P}_{p+1} (\mathbf{A}_p^T \cdot \mathbf{W}_p \cdot \mathbf{y}_p + \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \mathbf{y}_{k+p}) \\ &= \mathbf{P}_{p+1} (\underbrace{\mathbf{P}_p^{-1} \cdot \mathbf{P}_p \cdot \mathbf{A}_p^T \cdot \mathbf{W}_p \cdot \mathbf{y}_p}_{\hat{\boldsymbol{\theta}}_p} + \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \mathbf{y}_{k+p}) \\ &= \mathbf{P}_{p+1} (\mathbf{P}_p^{-1} \cdot \hat{\boldsymbol{\theta}}_p + \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \mathbf{y}_{k+p}) \end{aligned} \quad (3.112)$$

(3.112) 'deki \mathbf{P}_p^{-1} teriminin yerine (3.111) denkleminde bulunan \mathbf{P}_p^{-1} ' i koyarsak ;

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{p+1} &= \mathbf{P}_{p+1} [(\mathbf{P}_{p+1}^{-1} - \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \boldsymbol{\phi}_{k+p}^T) \cdot \hat{\boldsymbol{\theta}}_p + (\boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot \mathbf{y}_{k+p})] \\ &= \hat{\boldsymbol{\theta}}_p + \mathbf{P}_{p+1} \cdot \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \cdot (\mathbf{y}_{k+p} - \boldsymbol{\phi}_{k+p}^T \cdot \hat{\boldsymbol{\theta}}_p) \end{aligned}$$

$$\Rightarrow \hat{\boldsymbol{\theta}}_{p+1} = \hat{\boldsymbol{\theta}}_p + \mathbf{K}_{p+1} \cdot \tilde{\mathbf{y}}_{k+p} \quad (3.113)$$

Burada ;

$$\mathbf{K}_{p+1} \stackrel{\Delta}{=} \mathbf{P}_{p+1} \cdot \boldsymbol{\phi}_{k+p} \cdot \mathbf{W}_{p+1} \quad (3.114)$$

$$\tilde{\mathbf{y}}_{k+p} = \mathbf{y}_{k+p} - \boldsymbol{\phi}_{k+p}^T \cdot \hat{\boldsymbol{\theta}}_p \quad (3.115)$$

\mathbf{K}_{p+1} : Ağırlıklı E.K.K. kazanç matrisi

$\tilde{\mathbf{y}}_{k+p}$: Ölçülen değer ile tahmini çıkışın farkı

Görüldüğü üzere $\hat{\boldsymbol{\theta}}_{p+1}$, $\hat{\boldsymbol{\theta}}_p$ ' den elde edilmiş bir algoritmadır . Yalnız \mathbf{K}_{p+1} ' in hesaplanması \mathbf{P}_{p+1} ' e , o da $(\mathbf{A}_{p+1}^T \cdot \mathbf{W}_{p+1} \cdot \mathbf{A}_{p+1})^{-1}$ ' e bağlı olduğundan yine bir matris tersi söz konusudur . Matris tersi önermesi'ni kullanarak bunu gidermek istersek ;

(Matris tersi önermesi : eğer $\mathbf{P}_2^{-1} = \mathbf{P}_1 + \mathbf{H}^T \cdot \mathbf{R} \cdot \mathbf{H}^{-1}$ ise ;

$$\mathbf{P}_2 = \mathbf{P}_1 - \mathbf{P}_1 \cdot \mathbf{H}^T \cdot (\mathbf{H} \cdot \mathbf{P}_1 \cdot \mathbf{H}^T + \mathbf{R})^{-1} \cdot \mathbf{H} \cdot \mathbf{P}_1 \text{ 'dir . })$$

$$\mathbf{P}_{p+1} = \mathbf{P}_p - \frac{(\mathbf{P}_p \cdot \boldsymbol{\phi}_{k+p}) \cdot (\mathbf{P}_p \cdot \boldsymbol{\phi}_{k+p}^T)}{\boldsymbol{\phi}_{k+p}^T \cdot \mathbf{P}_p \cdot \boldsymbol{\phi}_{k+p} + \mathbf{W}_{p+1}^{-1}} \quad (3.116)$$

$$K_{p+1} = \frac{P_p \cdot \Phi_{k+p}}{\Phi_{k+p}^T \cdot P_p \cdot \Phi_{k+p} + w_{p+1}^{-1}} \quad (3.117)$$

$$\theta_{p+1}^{\wedge} = \theta_p^{\wedge} + \frac{P_p \cdot \Phi_{k+p}^T}{\Phi_{k+p}^T \cdot P_p \cdot \Phi_{k+p} + w_{p+1}^{-1}} \cdot (y_{k+p} - \Phi_{k+p}^T \cdot \theta_p^{\wedge}) \quad (3.118)$$

Algoritmanın çalışabilmesi için gerekli şart A_p ' nin kare matris olabildiği $p=(m+n+1)$. adımdaki θ_p ve P_p değerlerinin bilinmesi gereklidir . Bunun için $p=(m+n+1)$. adıma kadar olan veriler toplanıp (3.103)' teki ifade ile (klasik E.K.K.) başlangıç kestirimi yapılır . Ancak sonuçta yine $(m+n+1)*(m+n+1)$ boyutlu matrisin (A_p matrisi) tersinin alınması problemi ortaya çıkacaktır .

Bunu gidermek için ; satır sayısının sütun sayısından az olduğu pseudo-inverse matris yapısı kullanılarak klasik E.K.K.' dan matris tersi almaksızın sonuç elde edilebilir . $p < m+n+1$ için kullanılacak ana eşitlikler aşağıdaki gibidir :

$$\theta_{p+1}^{\wedge} = \theta_p^{\wedge} + \frac{Q_p \cdot \phi_p}{\phi_p^T \cdot Q_p \cdot \phi_p} \cdot (y - \phi_p^T \cdot \theta_p) \quad (3.119)$$

$$Q_{p+1} = Q_p - \frac{(Q_p \cdot \phi_p) \cdot (Q_p \cdot \phi_p)^T}{\phi_p^T \cdot Q_p \cdot \phi_p} \quad (3.120)$$

ve buradan da :

$$P_{p+1} = Q_p - \frac{(P_p \cdot \phi_p) \cdot (Q_p \cdot \phi_p)^T + (Q_p \cdot \phi_p) \cdot (P_p \cdot \phi_p)^T}{\phi_p^T \cdot Q_p \cdot \phi_p} + \frac{(Q_p \cdot \phi_p) \cdot (Q_p \cdot \phi_p)^T + (1 + \phi_p \cdot P_p \cdot \phi_p)}{(\phi_p^T \cdot Q_p \cdot \phi_p)^2} \quad (3.121)$$

bu ifadeler aşağıdaki başlangıç koşulları için geçerlidir :

$$Q_0=1 , P_0=0 , \theta_0=0$$

$p > m+n+1$ içinse aşağıdaki eşitlikler geçerlidir . (3.116 ve 3.118 ' deki ifadeler de $W=I$, $w_{p+1}=1$ ve $k=0$ alınarak) :

$$\hat{\theta}_{p+1} = \hat{\theta}_p + \frac{P_p \cdot \phi_p}{\phi_p^T \cdot Q_p \cdot \phi_p} \cdot (y_p - \phi_p^T \cdot \hat{\theta}_p) \quad (3.122)$$

$$P_{p+1} = P_p - \frac{(P_p \cdot \phi_p) \cdot (P_p \cdot \phi_p)^T}{1 + \phi_p^T \cdot Q_p \cdot \phi_p} \quad (3.123)$$

3.1.2.2.5. Genelleştirilmiş E.K.K. (Çevrimiçi)

Bir önceki kısımda çevrimdışı E.K.K.' dan bahsederken gürültüyü sıfır ortalamalı beyaz gürültü olarak kabul etmiştik . Buna göre aşağıdaki eşitlikleri yazabiliriz :

$$\begin{aligned} \hat{e}_k &= y_k - \left[\sum_{j=0}^m a_j \cdot u_{k-j} - \sum_{j=0}^n b_j \cdot y_{k-j} \right] \\ &= y_k - \phi_k^T \cdot \theta_k \end{aligned} \quad (3.125)$$

Buna göre ; eğer gürültü belirtildiği gibi sıfır ortalamalı ise hata sifira yaklaşacaktır . Ancak gürültünün beyaz gürültü olmayıp , beyaz gürültüden elde edilmiş farklı bir gürültü olduğunu kabul edersek hata için aşağıdaki eşitlik yazılabilir :

$$\hat{e}_k = -f_1 \cdot \hat{e}_{k-1} - f_2 \cdot \hat{e}_{k-2} \dots \dots \dots - f_s \cdot \hat{e}_{k-s} + w_k \quad (3.126)$$

Burada w_k beyaz gürültü dizisinin bir elemanı olarak kabul edilir . Buna göre gürültü parametreleri için model :

$$\frac{E_1(z)}{W(z)} = \frac{1}{1 + f_1 \cdot z^{-1} + f_2 \cdot z^{-2} + \dots \dots \dots + f_s \cdot z^{-s}} \quad (3.127)$$

olur . (3.126)' daki eşitliği matrissel formda yazacak olursak :

$$\underbrace{\begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ e_k \end{bmatrix}}_{E_k} = \underbrace{\begin{bmatrix} -e_0 & -e_1 & \cdot & -e_{1-s} \\ -e_1 & -e_0 & \cdot & -e_{2-s} \\ \cdot & \cdot & \cdot & \cdot \\ -e_{k-1} & -e_{k-2} & \cdot & -e_{k-s} \end{bmatrix}}_{G_k} \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \cdot \\ f_s \end{bmatrix}}_{\Psi_k \text{ (yardımcı parametre vektörü)}} + \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ w_k \end{bmatrix}}_{\Psi_k} \quad (3.128)$$

$k > s$ olduğu durumlar için Ψ yardımcı parametre vektörü' nün E.K.K. tahmini :

$$\hat{\Psi}_k^+ = G_k^+ \cdot e_k \quad (3.128)$$

Burada G_k^+ , G_k ' nın pseudoinverse' üdür . Ψ_k için ; gürültülü sistem giriş ve çıkışı u_k ve y_k aşağıdaki şekilde filtrelenir :

$$u_k^+ = u_k + \sum_{j=1}^s \hat{f}_j \cdot u_{k-j} \quad (3.129)$$

$$y_k^+ = y_k + \sum_{j=1}^s \hat{f}_j \cdot y_{k-j} \quad (3.130)$$

A_k yerine , filtrelenmiş verilerden elde edilen A_k^* matrisi E.K.K. metoduna uygulanır . Bulunan algoritma aşağıdaki gibi olacaktır .

$$\hat{\theta}_{k+1} = \theta_k + \frac{P_k \cdot \phi_{k+1}^* \cdot (y_{k+1} - \phi_{k+1}^* \cdot \theta_k)}{1 + \phi_{k+1}^{*T} \cdot P_k \cdot \phi_{k+1}^*} \quad (3.131)$$

$$\hat{P}_{k+1} = \hat{P}_k - \frac{(P_k \cdot \phi_{k+1}^*) \cdot (P_k \cdot \phi_{k+1}^*)^T}{1 + \phi_{k+1}^{*T} \cdot P_k \cdot \phi_{k+1}^*} \quad (3.132)$$

$$\hat{\psi}_{k+1} = \hat{\psi}_k + \frac{R_k \cdot g_{k+1} \cdot (e_{k+1} - g_{k+1}^T \cdot \psi_k)}{1 + g_{k+1}^T \cdot R_k \cdot g_{k+1}} \quad (3.133)$$

$$R_{k+1} = R_k - \frac{(R_k \cdot g_{k+1}) \cdot (R_k \cdot g_{k+1})^T}{1 + g_{k+1}^T \cdot R_k \cdot g_{k+1}} \quad (3.134)$$

Burada :

$$A_{k+1}^* = \begin{bmatrix} A_k^* \\ \phi_{k+1}^{*\top} \end{bmatrix} \quad \text{ve} \quad G_{k+1} = \begin{bmatrix} G_k \\ g_{k+1}^\top \end{bmatrix} \quad (3.135, 3.136)$$

$$\phi_k^{*\top} = [u_k^* \quad u_{k-1}^* \quad \dots \quad u_{k-m}^* \quad -y_{k-1}^* \quad -y_{k-2}^* \quad \dots \quad -y_{k-n}^*] \quad (3.137)$$

$$g_k^\top = [-e_k \quad -e_{k-1} \quad \dots \quad -e_{k+1-s}] \quad (3.138)$$

Algoritmanın başlayabilme şartı , ana parametreler için $(m+n+1)*(m+n+1)$, yardımcı parametreler için $s*s'$ lik matrislerin oluşturulması , yani bu kadar ölçüm alınmasıdır .

3.1.2.2.6. Yardımcı Değişkenler Metodu (Çevrimiçi)

Önceki kısımda çevrimdışı hali verilen metot için ;

$$y_k = A_k \cdot \theta + v_k \quad (3.139)$$

olduğu bilinmekte idi . Burada eğer $E[v_k] = 0$ ise E.K.K. metodu kullanılabilir . (3.139) eşitliğinin her iki yanını öyle bir W_k^\top terimi ile çarpalım ki:

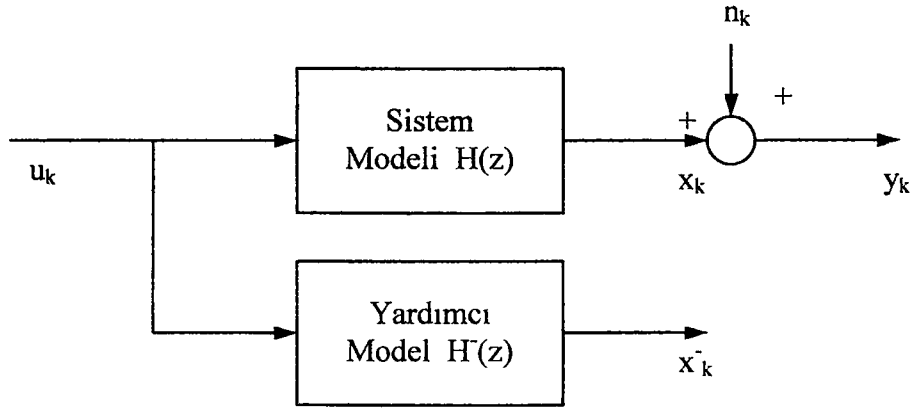
$$W_k^\top \cdot y_k = W_k^\top \cdot A_k \cdot \theta + W_k^\top \cdot v_k \quad (3.140)$$

$$\left. \begin{array}{l} E[W_k^\top \cdot v_k] = 0 \\ E[W_k^\top \cdot A_k] \rightarrow \text{nonsingüler (tekil olmayan) matris} \end{array} \right\} \text{ olmalıdır .}$$

Buradan :

$$\theta_k = (W_k^\top \cdot A_k)^{-1} \cdot (W_k^\top \cdot y_k) \quad (3.141)$$

W_k matrisi yardımcı değişkenler matrisi adını taşımaktadır ve metodumuzdaki ana problem bu matrisi oluşturmaktır . Bunun için Wong ve Polak (1967) tarafından geliştirilen yöntem şöyledir :



Şekil 3.9. Yardımcı Değişkenler Metodu için sistem modelinin blok diyagramı

Görüldüğü üzere burada yardımcı modele , gerçek model ile aynı giriş $\{u_k\}$ uygulanmaktadır . Böylece elde edilen h_k çıkış parametreleri , giriş $\{u_k\}$ ile ilişkili ancak , gürültü $\{n_k\}$ ve dolayısıyla $\{v_k\}$ ile ilişkisiz olmaktadır . Böylece W_k matrisini oluşturursak ;

$$W_k = \begin{bmatrix} u_0 & u_1 & \dots & u_{1-m} & -h_0 & -h_1 & \dots & -h_{1-n} \\ u_1 & u_0 & \dots & u_{2-m} & -h_1 & -h_0 & \dots & -h_{2-n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ u_{k-1} & u_{k-2} & \dots & u_{k-m} & -h_{k-1} & -h_{k-2} & \dots & -h_{k-n} \end{bmatrix} \quad (3.142)$$

$$w_{k+1} = [u_k \quad u_{k-1} \quad \dots \quad u_{k-m} \quad -h_k \quad -h_{k-1} \quad \dots \quad -h_{k+1-n}] \quad (3.143)$$

olur . Bölüm 3.1.2.2.4. ' teki prosedürden :

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \frac{P_k \cdot w_{k+1} \cdot (y_{k+1} - \phi_{k+1} \cdot \theta_k)}{1 + \phi_{k+1}^T \cdot P_k \cdot w_{k+1}} \quad (3.144)$$

$$P_{k+1} = P_k - \frac{(P_k \cdot w_{k+1}) \cdot (P_k \cdot w_{k+1})^T}{1 + \phi_{k+1}^T \cdot P_k \cdot w_{k+1}} \quad (3.145)$$

olduğu bilinmektedir . Burada algoritmayı başlatabilmek için $\alpha = (m+n+1)$ adet ölçüm almak gereklidir .

Buna göre :

$$\theta_{\alpha} = (W_{\alpha}^T \cdot A_{\alpha}) \cdot (W_{\alpha}^T \cdot y_{\alpha}) \quad (3.146)$$

$$P_0 = (W_{\alpha}^T \cdot y_{\alpha})^{-1} \quad (3.147)$$

olur .Young' in (1970) metoduna göre ; yardımcı model güncellenmeden önce alçak geçiren filtre ve zaman gecikmesi kullanılır . Bu işlem öylesine gerçekleştirilmelidir ki ; yardımcı model parametreleri her an e_k değeri ile ilişkisiz olsun . Bahsedilen alçak geçiren filtre modeli de aşağıdaki gibidir :

$$\phi_{aux}(k) = (1 - v) \cdot \phi_{aux}(k - 1) + v \cdot \theta(k) \quad (3.148)$$

Bu denklemde , kestirimin kararsızlığından korunmak için v değeri 0.03 ile 0.05 arasında seçilmelidir .

3.1.2.2.7. Ardışıl İlişki (Correlation) Metodu

Eğer giriş dizimiz $\{u(k)\}$ gelişigüzel bir işaret ise bunun otokorelasyonu (yani kendi kendisiyle korelasyonu) *ergodicity** kabul edilerek _ :

$$\phi_{uu}(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N+1} \cdot \sum_{k=0}^N u(k) \cdot u(k - \tau) \quad (3.149)$$

giriş ve çıkış arasındaki korelasyon ise :

$$\phi_{uy}(\tau) = \lim_{N \rightarrow \infty} \frac{1}{N+1} \cdot \sum_{k=0}^N u(k - \tau) \cdot y(k) \quad (3.150)$$

çıkış işareti için $y(k) = x(k) + n(k)$ (3.151) yazılır ve giriş işareti ile gürültü ilişkisiz kabul edilirse , aşağıdaki eşitlik yazılabilir :

* ergodicity = frekans ortalaması ile zaman ortalamasının aynı değerde olması

$$\phi_{uy}(\tau) = \phi_{uu}(\tau) \quad (3.152)$$

yukarıdaki denklem , bu metodun anahtarı olmaktadır . Konvolüsyon eşitliği ;

$$\phi_{uy}(\tau) = \sum_{\nu=0}^{\infty} g(\nu) \cdot \phi_{uu}(\tau - \nu) \quad (3.153)$$

Darbe cevabı

Bu eşitliği matrissel formda yazacak olursak ;

$$\begin{bmatrix} \phi_{uu}(-M) & \dots & \phi_{uu}(-M-1) \\ \phi_{uu}(1-M) & \dots & \phi_{uu}(1-M-1) \\ \vdots & \vdots & \vdots \\ \phi_{uu}(M) & \dots & \phi_{uu}(M-1) \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(l) \end{bmatrix} = \begin{bmatrix} \phi_{uy}(-M) \\ \phi_{uy}(1-M) \\ \vdots \\ \phi_{uy}(M) \end{bmatrix} \quad (3.154)$$

Bu eşitliği pseudoinverse algoritma kullanarak çözersek , darbe cevabının bileşenlerini ($g(0)$, $g(1)$, , $g(l)$) bulabiliriz . Görüldüğü üzere darbe cevabı l gibi sonlu bir sayı için kesilmiştir ki bu da metodun zayıf yönüdür ve doğruluğu etkiler . l ' nin çok büyük seçilmesi halinde de hesap miktarı artacağından işlem süresi uzayacaktır . Bu yüzden optimum bir l değeri seçilmelidir . Diğer bir sorun ise N ' nin sonsuz ya da çok büyük olamamasıdır . Bu güçlüğü aşmak için aşağıdaki recursive algoritma kullanılabilir .

$$\phi_{uy}(\tau, k) = \phi_{uy}(\tau, k-1) + \frac{1}{k+1} \cdot [u(k-\tau) \cdot y(k) - \phi_{uy}(\tau, k-1)] \quad (3.155)$$

Yukarıdaki denklemde k , örnek sayısını göstermektedir . u_k ve y_k ' nin ortalamaları sıfırdır .

Elde edilen darbe cevaplarından sistemin ayırık zaman transfer fonksiyonunu elde etmek için klasik metotlardan faydalanılabilir .

3.1.2.2.8. Maksimum Benzerlik (Maksimum Likelihood) Metodu (Çevrimiçi)

Bir önceki kısımda ele aldığımız , Aström ve Bohlin (1966) tarafından geliştirilen çevrimdışı maksimum benzerlik metodu yüksek seviyeli gürültüler için bile iyi sonuçlar vermesine karşın , çevrimiçi uygulamalar için uygun değildir . Gentler ve Banyasz (1974) tarafından önerilen çevrimiçi metot aşağıdaki gibi işlemektedir .

Bu algoritmada sistem , Şekil 3.6. ' daki gibi düşünülerek aşağıdaki giriş-çıkış ilişkisi yazılabilir :

$$y_k = \frac{G(z^{-1})}{1 + H(z^{-1})} \cdot u_k + \frac{1}{(1 + H(z^{-1})) \cdot (1 + D(z^{-1}))} \cdot w_k \quad (3.156)$$

Burada G , H ve D , z^{-1} ' in polinomlarıdır . Yeni parametre vektörü ise :

$$\theta = \begin{bmatrix} g \\ - \\ h \\ - \\ d \\ - \end{bmatrix} \quad (3.157)$$

Yukarıda g,h,d polinom katsayılarıdır . (3.156) denkleminde gürültü değeri çekilirse ;

$$w_k = [1 + D(z^{-1})] \cdot \{ [1 + H(z^{-1})] \cdot y_k - G(z^{-1}) \cdot u_k \} \quad (3.158)$$

ve bu eşitlikte w_k ' nın değişik parametrelere göre kısmi türevleri alınırsa :

$$\left. \begin{aligned} \frac{\partial w_k}{\partial g_j} &= -[1 + D(z^{-1})] \cdot z^{-j} \cdot u_k = -u_{k-j}^F \\ \frac{\partial w_k}{\partial h_j} &= [1 + D(z^{-1})] \cdot z^{-j} \cdot y_k = y_{k-j}^F \\ \frac{\partial w_k}{\partial d_j} &= z^{-j} \cdot \{ [1 + H(z^{-1})] \cdot y_k - G(z^{-1}) \cdot u_k \} = -\mu_{k-j} \end{aligned} \right\} \quad (3.159)$$

Buradaki u_k^F , y_k^F , μ_k değişkenleri aşağıdaki gibi tanımlanmaktadır :

$$u_k^F \stackrel{\Delta}{=} [1 + D(z^{-1})].z^{-j}.u_k \quad (3.160)$$

$$y_k^F \stackrel{\Delta}{=} [1 + D(z^{-1})].z^{-j}.y_k \quad (3.161)$$

$$\mu_k \stackrel{\Delta}{=} G(z^{-1}).u_k - [1 + H(z^{-1})].y_k \quad (3.162)$$

Yukarıdaki türevler hareketli ortalama filtrelemesi ve kaydırmalarla elde edilmektedir .

Gürültünün 1. türevlerini bir vektörde toplarsak :

$$v_k = \begin{bmatrix} -u_{n+1}^F \\ y_n^F \\ -u_n \end{bmatrix} \quad (3.163)$$

ve burada ;

$$u_{n+1}^F = \begin{bmatrix} u_k^F \\ u_{k-1}^F \\ \cdot \\ u_{k-n}^F \end{bmatrix} \quad y_n^F = \begin{bmatrix} y_{k-1}^F \\ \cdot \\ \cdot \\ y_{k-n}^F \end{bmatrix} \quad u_n = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \cdot \\ u_{k-n} \end{bmatrix} \quad (3.164, 3.165, 3.166)$$

' dir . Sıfır olmayan ikinci dereceden türevler ise :

$$\left. \begin{aligned} \frac{\partial^2 w_k}{\partial g_j \partial d_m} &= -z^{-(j+m)}.u_k = u_{k-j-m} \\ \frac{\partial^2 w_k}{\partial h_i \partial d_m} &= z^{-(j+m)}.y_k = -y_{k-j-m} \end{aligned} \right\} \quad (3.167)$$

olmaktadır . İkinci türevler de çift kaydırma ile elde edilmektedir . Bu türevlerin oluşturduğu matris ise :

$$W_k = \begin{bmatrix} 0 & x_k \\ x_k^T & 0 \end{bmatrix} \quad (3.168)$$

$$x_k \stackrel{\Delta}{=} \begin{bmatrix} -u_{(k-1)n,n+1} \\ y_{(k-2)n,n} \end{bmatrix} \quad (3.169)$$

şeklindedir .

Sonuçta ;

$$\theta_k = \theta_{k-1} - R^{-1}(k, N, \theta_{k-1}) \cdot q(k, n, \theta_{k-1}) \quad (3.170)$$

$$q(k, n, \theta_{k-1}) \stackrel{\Delta}{=} \left[\frac{\partial}{\partial \theta} (W_n^T \cdot W_n) \right]_{\theta=\theta_{k-1}}^T \quad (3.171)$$

$$R(k, N, \theta_{k-1}) \stackrel{\Delta}{=} \left[\frac{\partial}{\partial \theta} \cdot q(k, N) \right]_{\theta=\theta_{k-1}} \quad (3.172)$$

yazılabilir . Burada N , önceki N adet ardışıl gelen değerleri göstermektedir . Kısaltarak tekrar yazacak olursak :

$$\theta_k = \theta_{k-1} - R^{-1}(k; k-1, N) \cdot q(k; k-1, N) \quad (3.173)$$

$$q(k; k-1, N) = q(k-1; k-2, N-1) + v(k) \cdot W(k) \quad (3.174)$$

$$R(k; k-1, N) = R(k-1; k-2, N-1) + v(k)^T \cdot v(k) + W(k) \cdot w(k) \quad (3.175)$$

Eski ölçümlerin etkisini azaltmak için filtre fonksiyonu $\lambda \in [0,1]$ kullanılırsa :

$$q(k; k-1, N) = \lambda \cdot q(k-1; k-2, N-1) + v(k) \cdot w(k) \quad (3.176)$$

$$R(k; k-1, N) = \lambda \cdot R(k-1; k-2, N-1) + v(k)^T \cdot v(k) + W(k) \cdot w(k) \quad (3.177)$$

ve matris tersi önerme' si kullanılarak ta ;

$$R^{-1}(k; k-1, N) = R^{-1}(k-1; k-2, N) \cdot [I - W(k) \cdot w(k) \cdot R^{-1}(k; k-1, N)] \quad (3.178)$$

$$\Rightarrow R^{-1}(k; k-1, N) \stackrel{\Delta}{=} \frac{1}{\lambda} \cdot R^{-1}(k-1; k-2, N-1) - \frac{R^{-1}(k-1; k-2, N-1) \cdot v_k \cdot v_k^T \cdot R^{-1}(k-1; k-2, N-1)}{\lambda + v_k^T \cdot R^{-1}(k-1; k-2, N-1) \cdot v_k} \quad (3.179)$$

bulunur . Bu ifade (3.173) 'de yerine konularak yeni θ_k değeri bulunmaktadır .

3.1.2.2.9. Stokastik Yaklaşım

Metodun en büyük avantajı gürültünün istatistiği hakkında bir ön bilgiye ihtiyaç duymamasıdır . Uygulamada bazı metotlardan daha hızlıdır . Ancak daha kompleks işlemler gerektirmektedir .

Metot ilk olarak $h(x)=0$ eşitliğinin kökünün bulunması için basit gradient metodunun algoritması geliştirilerek tanımlanmıştır . Buna göre :

$$x_{i+1} = x_i + K_i \cdot h(x_i) \quad (3.180)$$

Burada K_i algoritmanın yakınsamasını sağlayan kesin durum belirten bir reel sayılar dizisidir . Gürültülü ortamda $h(x_i)$ yerine $z(x_i) = h(x_i) + v_i$ konulduğunda bir belirsizlik oluşmaktadır . v_i sıfır ortalamalı rasgele gürültü , $h(x)$ ise z' in x üzerinde regresyon fonksiyonudur .

$$E[z | x] = \int_{-\infty}^{\infty} z \cdot p(z | x) \cdot dx = h(x) \quad (3.181)$$

Gürültülü durumda (3.180) ifadesini kullanamayız . Bunun yerine ifade de $h(x)$ yerine $z(x)$ kullanabiliriz . Robbins ve Monro' ya (1951) göre aşağıdaki şartlar sağlanıyorsa x_i gerçek çözüme yakınsayacaktır :

$$\lim_{i \rightarrow \infty} K_i = 0 \quad \text{ve} \quad \sum_{i=1}^{\infty} K_i = \infty \quad \text{ve} \quad \sum_{i=1}^{\infty} K_i^2 < \infty \quad (3.182)$$

Bu koşulları sağlamak için gerekli olan en basit ifade :

$$K_i = \frac{\alpha}{\beta + i} \quad \left. \vphantom{K_i} \right\} \quad \begin{array}{l} \alpha, \beta \rightarrow \text{sabit} \\ \alpha > 0, \beta \geq 0 \end{array} \quad (3.183)$$

Eğer bu şartlar sağlanıyorsa ;

$$|h(x_i)| = a \cdot |x - \hat{x}| + b \quad a, b > 0 \quad (i) \quad (3.184)$$

$$u_{i+1} = u_i - K_i \cdot \frac{d\theta(u_i)}{du_i} \quad (ii) \quad (3.185)$$

Gürültülü durumda θ yerine ϕ ($\phi = \theta(u) + \xi$) alınır . Şöyle ki ;

$$\frac{d\phi(u_i)}{du_i} \cong \frac{\phi(u_{i+\Delta} - u_i) - \phi(u_{i-\Delta} - u_i)}{2 \cdot \Delta u_i} \quad (3.186)$$

$$x_{n+1} = x_n + \gamma_{n+1} \cdot \left\{ \underbrace{f(r_1, r_2, \dots, r_n)}_{\text{Gözlemlerin skaler}} - x_n \right\} \quad (3.187)$$

K_i (n+1). gözlem n. tahmini değer

Buradan tanıyım algoritması ise :

$$\hat{g}_{k+1} = \hat{g}_k + \gamma_{k+1} \cdot u_{k+1} \cdot [y_{k+1} - u_{k+1}^T \cdot \hat{g}_k] \quad (3.188)$$

$$g = [g_1 \quad g_2 \quad \dots \quad g_n] \quad (3.189)$$

$$u_k = [u_{k-1} \quad u_{k-2} \quad \dots \quad u_{k-n}]^T \quad (3.190)$$

$$\gamma_{k+1} = \frac{1}{k+1} \quad k = 1, 2, \dots \quad (3.191)$$

olmak üzere ;

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \frac{\gamma}{k+1} \cdot \frac{(\phi_{k+1}^T \cdot \theta_k - y_{k+1}) \cdot \phi_k}{\|\phi_k\|^2} \quad (3.192)$$

(E.K.K. ile stokastik yaklaşım)

3.1.3. Çevrimiçi Kestirim İşleminde Farklı Metotların Karşılaştırılması

Bu kısımda ise ele aldığımız metotların birbirlerine üstünlükleri üzerine yapılan karşılaştırılmalı çalışmaların bir özeti sunulmaktadır . İlk olarak Sinha ve Sen (1975) tarafından yapılan çalışma ele alınacaktır . Ardından da Isermann , Baur , Bamberger , Kneppo ve Siebert (1974) tarafından gerçekleştirilen çalışmadan bahsedilecektir .

3.1.3.1. Sinha ve Sen' in Uygulaması

Sinha ve Sen (1975) tarafından yapılan bu çalışmada karşılaştırma için önce örnek bir sistem ele alınmış ve bu sistemin modeli aşağıdaki gibi verilmiştir :

$$x_{k+1} = a_1 \cdot u_k + a_2 \cdot u_{k-1} - b_1 \cdot x_k - b_2 \cdot x_{k-1} \quad (3.193)$$

$$y_k = x_k + \alpha \cdot n_k \quad (3.194)$$

yukarıdaki y ifadesinde α , gürültü / sinyal oranını temsil etmektedir . Bu sistemden elde edilen 500 adet giriş çıkış bilgisinden çevrimdışı kestirim sonucunda aşağıdaki sonuçlar elde edilmiştir :

$$a_1 = 0.079$$

$$a_2 = 0.047$$

$$b_1 = -0.975$$

$$b_2 = 0.223$$

Giriş işareti u_k ' ya etki eden giriş gürültüsü , sıfır ortalamalı Gaussian gürültü kabul edilmiştir . Çıkışa etki eden gürültü ise aşağıda görülen bağıntıdaki gibi kabul varsayılmıştır :

$$n_{k+1} = 0.8 \cdot n_k + 0.7 \cdot \omega_k \quad (3.195)$$

Burada ; ω_k girişten ilişkisiz beyaz gürültü olmaktadır . α katsayısı ise farklı gürültü / sinyal oranlarını sağlamak için (%20 , %40 ve %100 oranlarını sağlamakta) kullanılmaktadır . Bunun için gerekli olan değerleri 0.04 , 0.08 ve 0.20' dir .

Eldeki bu verilerle , aşağıdaki çevrimiçi yöntemler kullanılmıştır .

- i) Recursive En Küçük Kareler
- ii) Genelleştirilmiş En Küçük Kareler
- iii) Yardımcı Değişkenler
- iv) İlişki (Korelasyon)
- v) Stokastik Yaklaşım
- vi) E.K.K. ile stokastik yaklaşım

2. dereceden bu sistem için farklı gürültü / sinyal oranlarında elde edilen kestirim sonuçları aşağıdaki çizelgede görülmektedir . Çizelgenin ilk satırında ; tüm verilerin toplanmasının ardından yapılan çevrimdışı kestirimin sonuçları verilmiştir . Bu sonuçlar doğruluk açısından bir referans olarak kullanılmıştır .

Çizelge 3.3. 2.dereceden bir sistem için elde edilen parametre kestirim sonuçları

Metot	Gürültü/sinyal oranı	a ₁	a ₂	b ₁	b ₂
Çevrimdışı kestirim sonuçları		0.079	0.047	-0.975	0.223
En Küçük Kareler	%20	0.0772	0.0483	-0.944	0.191
	%40	0.0756	0.0514	-0.890	0.133
	%100	0.0716	0.0543	-0.804	0.121
Genelleştirilmiş En Küçük Kareler	%20	0.0775	0.0445	-0.982	0.235
	%40	0.0775	0.0423	-0.988	0.253
	%100	0.0695	0.0385	-0.979	0.275
Yardımcı Değişkenler	%20	0.0772	0.0475	-0.954	0.203
	%40	0.0755	0.047	-0.951	0.193
	%100	0.0707	0.0452	-0.925	0.171
Korelasyon	%20	0.0705	0.0321	-0.952	0.354
	%40	0.069	0.0325	-0.935	0.336
	%100	0.064	0.0336	-0.879	0.275
Stokastik Yaklaşım	%20	0.079	0.052	-0.895	0.155
	%40	0.082	0.054	-0.883	0.154
	%100	0.087	0.058	-0.848	0.153
E.K.K. ile stokastik yaklaşım	%20	0.0771	0.0459	-0.974	0.2201
	%40	0.0765	0.0447	-0.975	0.2158
	%100	0.0729	0.0418	-0.966	0.1776

Görüldüğü üzere tüm metotlar düşük gürültü / sinyal oranlarında iyi sonuçlar vermiştir . Ancak gürültü / sinyal oranı %100 yapıldığında ; metotların hepsi aynı ölçüde doğru sonuçlar vermediği görülmüştür . Bu noktada dikkat edilmesi gereken diğer bir önemli noktada kestirim işlemi için gerekli olan zamandır . Çizelge 3.4.' de ise her bir metot için gerekli olan kestirim süresi ve gerekli iterasyon sayısı verilmiştir :

Çizelge 3.4. Parametre kestiriminde kullanılan metotların işlem süresi açısından karşılaştırılmaları

Metot	İşlem süresi (E.K.K.' ya göre normalize edilmiş)	Her bir iterasyondaki işlem sayısı
En Küçük Kareler	1	66
Genelleştirilmiş En Küçük Kareler	4.12	146
Yardımcı Değişkenler	1.47	81
Korelasyon	1.47	82
Stokastik Yaklaşım	0.7	16
E.K.K. ile stokastik yaklaşım	3.41	95

Çizelge 3.4. 'da verilen işlem süreleri En Küçük Kareler metodunun işlem süresine göre normalize edilmiştir (Metodun işlem süresi / E.K.K. işlem süresi) . Böylece farklı çalışma ortamlarından (farklı computer'lar) ileri gelen rakamsal farklılıkların önüne geçilmiştir .

Çizelge 3.3. ve Çizelge 3.4. ele alınıp incelendiğinde aşağıdaki sonuçlar çıkartılmıştır :

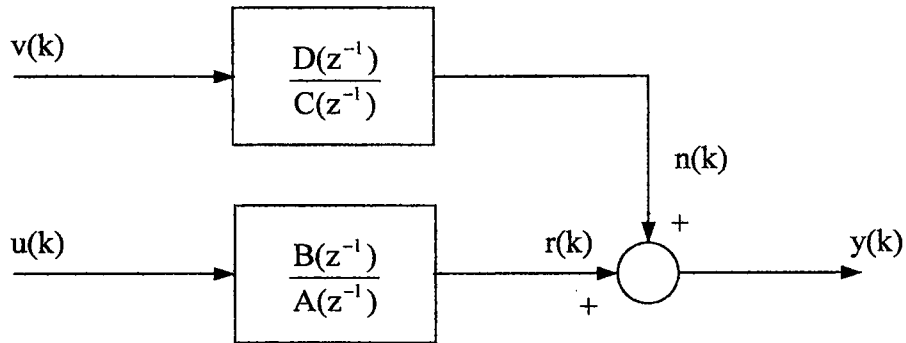
- E.K.K. algoritması ele alınan metotlar içerisinde düşük gürültü / sinyal oranları için en etkili olanıdır .Stokastik yaklaşımdan sonra en küçük işlem süresine sahiptir (Bakınız Çizelge 3.4.) . Aynı zamanda gerçek değerlere kısa sürede

yeterince yakınsamaktadır . Ancak gürültü arttıkça (>%15-%20) parametre kestirim işleminin uygunluğu azalmaktadır .

- Stokastik yaklaşım kısa işlem süresine rağmen . gerçek değerlere yakınsaması açısından çok yavaştır .
- Korelasyon metodu da yüksek gürültü / sinyal oranları haricinde oldukça verimlidir . Ancak kestirim işleminin başlayabilmesi için yeterli sayıda bilgi toplanmasını gerektirmektedir . Bu yüzden iyi bir off-line metot olarak kabul edilebilir , ancak on-line uygulamalar için uygun değildir . Özellikle gürültü sabit bir seyir sergilemiyorsa ...
- Metotların birçoğuyla tatmin edici kestirim sonuçları elde edilebilmesine rağmen genele bakıldığında şu yorumda bulunulabilir : model parametrelerinin belirlenmesinde E.K.K.'nın , yardımcı parametrelerin belirlenmesinde ise stokastik yaklaşımın kullanılması en akılcı ve verimli çözüm olmaktadır .

3.1.3.2. Isermann ve Arkadaşları'nın Uygulaması

Isermann ve arkadaşları' nın (1974) karşılaştırma amacıyla kullanılan sistemin ayrık zaman blok diyagramı aşağıdaki gibidir :



Şekil 3.10. Isermann tarafından karşılaştırma amacıyla kullanılan sistemin blok diyagramı

Burada ele alınan sistem lineer zamanla değişmeyen sistem olarak alınmıştır . Karşılaştırma işlemleri için üç farklı sistem kullanılmıştır . Bunlar :

a) İkinci dereceden osilasyon yapan sistem :

$$G_1(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (3.196)$$

Bu sistem için kullanılan gerçek parametre değerleri ise ;

$$a_1 = -1.5 , a_2 = 0.7 , b_1 = 1.0 , b_2 = 0.5 \text{ 'tir.}$$

ve örnekleme periyodu 2 saniyedir .

b) İkinci dereceden nonminimum phase sistem :

$$G_2(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}} \quad (3.197)$$

Bu sistemde kullanılan gerçek parametre değerleri ;

$$a_1 = -1.425 , a_2 = 0.496 , b_1 = -0.102 , b_2 = 0.173 \text{ 'tür.}$$

ve örnekleme periyoduda yine 2 saniyedir .

c) Üçüncü dereceden alçak geçiren gecikmeli sistem :

$$G_3(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3}} \quad (3.198)$$

Burada kullanılan gerçek değerler ise :

$$a_1 = -1.5 , a_2 = 0.705 , a_3 = -0.100 , b_1 = 0.065 , b_2 = 0.048 , b_3 = -0.008 \text{ ve } d = 1 \text{ 'dir.}$$

örnekleme süresi ise bu kez 4 saniyedir .

II ve III nolu sistemler Isermann tarafından oluşturulup kullanılmıştır . I nolu sistem ise **Aström ve Bohlin** , **Hastings-James ve Sage** , **Gustavsson ve Gentil** tarafından diğer çalışmalarda kullanılmış bir sistemdir . II ve III nolu sistemler ,

sürekli bir sistemden sıfırıncı dereceden tutucu ile elde edilmiştir . I nolu sistem ise özellikle oluşturulmuştur .

Çıkış işareti $r(k)$ ' nin üzerine , sıfır ortalamalı beyaz gürültüden (" $v(k)$ ") , gürültü filtresi $G_v(z^{-1})$ ile elde edilen $n(k)$ işareti bindirilmektedir . Bu ayrık zaman gürültü filtresinin transfer fonksiyonu aşağıdaki gibi verilmiştir :

$$G_v(z^{-1}) = \frac{D(z^{-1})}{C(z^{-1})} = \frac{d_1 \cdot z^{-1}}{1 + c_1 \cdot z^{-1} + c_2 \cdot z^{-2}} \quad (3.199)$$

Görüldüğü üzere bu fonksiyon G_1 , G_2 ve G_3 ' ten farklıdır . Çünkü kullanılan filtre genel bir filtredir . Her üç sistem için , gürültü modelinin parametreleri aşağıdaki gibidir:

$$\left. \begin{array}{l} c_1 = -1.027 \quad , \quad c_2 = 0.264 \\ d_1 = 0.014 \cdot \gamma \end{array} \right\} \text{ sistem I ve III için}$$

$$\left. \begin{array}{l} c_1 = -0.527 \quad , \quad c_2 = 0.0695 \\ d_1 = 0.0117 \cdot \gamma \end{array} \right\} \text{ sistem III için}$$

γ burada gürültü / sinyal oranına bağlı bir katsayıdır . Gürültü / sinyal oranı ise ;

$$\eta = \frac{\sqrt{n^2(k)}}{K \cdot u_0} \quad (3.200)$$

şeklinde tanımlanmıştır . Bu denklemde K kazanç katsayısını göstermektedir . u_0 içinse aşağıdaki gibi bir şart belirtilmiştir :

$$- u_0/2 \leq u(k) \leq u_0/2 \quad (u_0 = 1) \quad (3.201)$$

Görüldüğü üzere giriş işaretinin genliği belirli bir değerde sınırlandırılmıştır . Yapılan testler esnasında iki farklı gürültü / sinyal oranı ($\eta = 0.1$ ve $\eta = 0.2$) kullanılmıştır .

İşlem süresi olarakta 350 , 1400 ve 7000 sn. gibi uzun süreler seçilmiştir .
karşılaştırma yapılırken aşağıdaki hatalar sürekli olarak gözlenmiştir :

a) Bağıl parametre hatasının kareleri toplamının karekökü :

$$\delta_{\Sigma 1} = \left\| \frac{\Delta\theta_i}{\theta_i} \right\| = \left[\sum_{i=1}^p \left[\frac{\Delta\theta_i}{\theta_i} \right]^2 \right]^{1/2} \quad (3.202)$$

b) Parametre hatasının kareleri toplamının , gerçek parametre değerlerinin kareleri toplamına oranının karekökü :

$$\delta_{\Sigma 2} = \frac{\|\Delta\theta_i\|}{\|\theta_i\|} = \left[\frac{\sum_{i=1}^p \Delta\theta_i^2}{\sum_{i=1}^p \theta_i^2} \right]^{1/2} \quad (3.203)$$

c) Darbe cevabı hatasının kareleri toplamının , gerçek darbe cevabının kareleri toplamına oranının karekökü :

$$\delta_g = \left[\frac{\Delta g^2(k)}{g^2(k)} \right]^{1/2} = \left[\frac{\sum_{k=0}^M \Delta g^2(k)}{\sum_{k=0}^M g^2(k)} \right]^{1/2} \quad (3.204)$$

Karşılaştırma amacıyla aşağıdaki metotlar teste tabi tutulmuştur :

- Least Squares (En Küçük Kareler)
- Generalized Least Squares (Genelleştirilmiş En Küçük Kareler)
- Instrumental Variables (Yardımcı Değişkenler)
- Stochastic Approximation (Stokastik Yaklaşım)
- Correlation Analysis (İlişki Metodu)

Yapılan testler sonucunda elde edilen sonuçlara ilişkin çizelgelerden birkaçı aşağıda verilmiştir :

Çizelge 3.5. Isermann'ın çalışmasında ; III nolu sistem için kestirim sonuçları

Sistem III	a_1	a_2	a_3	b_1	b_2	b_3	$\delta_{\Sigma 1}$	$\delta_{\Sigma 2}$	δ_g
Gerçek Değerler	-1.500	0.705	-0.100	0.065	0.048	-0.008	-	-	-
Yardımcı Değişkenler	-0.941	-0.038	0.155	0.064	0.084	0.026	1.918	0.337	0.011
Stokastik Yaklaşım	-1.490	0.689	-0.093	0.065	0.048	-0.007	0.065	0.0001	0.005
İlişki Metodu	-1.205	0.322	0.026	0.065	0.068	0.010	0.967	0.091	0.005

Çizelge 3.6. Isermann'ın çalışması sonucunda elde edilen genel sonuçlar

	Giriş Sinyali	Performans				İşlem Süresi (%)		Genel Güvenilirlik		Ön tanımlı değerler
		$\sigma_{\delta g}$		$\sigma_{\delta y}$		Sistem I + II	III	%		
η		0.1	0.2	0.1	0.2	-	-	0.1	0.2	-
İlişki Metodu	Gürültüsüz PRBS*	0.08	0.17	0.11	0.25	100	100	100	100	I
Yardımcı Değişkenler	Gürültüsüz PRBS	0.09	0.16	0.13	0.21	153	236	90	80	$P(0), \gamma$
Stokastik Yaklaşım	Beyaz Gürültülü PRBS	0.10	0.22	0.13	0.25	106	108	90	85	$\zeta_1, \zeta_2, a, b, I$
Genelleştirilmiş En Küçük Kareler	Gürültüsüz PRBS	0.18	0.31	0.31	0.72	194	287	95	90	$P(0), Q(0)$ v
En Küçük Kareler	Gürültüsüz PRBS	0.29	0.44	-	-	145	230	100	100	$P(0)$

* PRBS : Pseudo Random Binary Signal

Çizelge 3.5.' te "marathon runs" adı verilen çok uzun süreli kestirim işleminin sonuçları verilmiştir . kestirim metotlarından Stokastik yaklaşım gerçek değerlere yakın sonuçlar vermiştir. İlişki metodu ve yardımcı değişkenler metodunun parametre hatası $\delta_{\Sigma 1}$, darbe cevabı hatası δ_g ve kazanç katsayısı K' nin hatasına göre çok büyük gerçekleşmiştir . Burada küçük darbe cevabı hatalarının bile çok farklı kestirim sonucuna yol açtığı görülmektedir .

Çizelge 3.6.' da ise ele alınan 5 çevrimiçi metodun sonuçları topluca verilmiştir. Çizelge üzerinde ilk olarak kullanılan giriş işaretinin tipi verilmiştir . Bir sonraki sütunda yer alan performans kriteri ise çalışmanın en önemli noktası olmaktadır . Bu kriter oluşturulurken sistemin darbe cevabı hatasının standart sapması ($\sigma_{\delta g}$) ve kapalı çevrim çıkış hatasının standart sapması ($\sigma_{\delta y}$) göz önünde tutulmuştur . Bu değerler hesaplanırken 1400 saniyelik işlem süresi esas alınmıştır . $\sigma_{\delta g}$ için I , II ve III nolu sistemlerden elde edilen değerlerin ortalaması , $\sigma_{\delta y}$ içinse II ve III nolu sistemlerden elde edilen değerlerin ortalaması performans değeri olarak kabul edilmiştir . Kestirim işlemi için harcanan işlem süresi içinde ; 1000 adet giriş-çıkış bilgisi ile yapılacak kestirim için harcanan süre tutulmuştur . Farklı kestirim ortamları (farklı computerlar gibi) ' ndan alınacak farklı sonuçların önüne geçmek için bu işlem süreleri ilişki metodunun işlem süresine göre normalize edilip (ilişki metodu işlem süresi = %100) çizelgede sunulmuştur . Bir sonraki sütunda ise metotların genel güvenilirliği (overall reliability) verilmiştir . Bu kriter oluşturulurken de ; metotların 30 adet kestirim işlemi sonucunda başarılı oldukları kestirim sayısı % olarak verilmektedir .

Çizelgelerin incelenmesi sonucunda aşağıdaki genel sonuçlara ulaşılmıştır :

- Genel lineer sistemlerde ilişki metodu diğer metotlara göre birçok avantajlar sağlamaktadır . Kısa işlem süresinde yüksek bir performans göstermektedir ve %100 genel güvenilirlik oranına sahiptir . Ancak bu düşük işlem süresi gürültü beyaz gürültü olduğunda mümkün olmaktadır . Ön tanımlı değeri sadece bir tanedir .

- Yardımcı Değişkenler metoduda en az ilişki metodu kadar iyi bir performans göstermektedir . Ancak işlem zamanı uzun ve genel güvenilirliği düşüktür . Başlangıç matrisi ve filtre faktörü ön tanımlı değerleridir .
- Stokastik yaklaşım ise uzun kestirim sürelerinde iyi sayılabilecek bir performans göstermektedir . işlem süresi neredeyse ilişki metodu kadar düşüktür . Ancak genel güvenilirliği diğer metotlara göre azdır ve 5 adet ön tanımlı değere ihtiyacı vardır .
- Genelleştirilmiş En Küçük Kareler Metodu genel olarak kötü bir performans sergilemiştir . işlem süresi en uzun olan metottur . Buna rağmen genel güvenilirliği , yardımcı değişkenler ve stokastik yaklaşımınkine yakındır .
- En Küçük Kareler metodu karşılaştırma işlemindeki en kötü performans sonucunu vermiştir . Bununla birlikte küçük işlem sürelerinde (örneğin 300 saniye) diğer tüm metotlarla aynı sonuçları vermiştir . işlem süresi yardımcı değişkenler metodunkine kadar kısadır . ve genel güvenilirliği %100' dür . Ön tanımlı değer olarak sadece başlangıç matrisine ihtiyaç duymaktadır .

Tüm test aşamalarında bilinmeyen parametre sayısı , metotların etkinliğini etkilediği gözlenmiştir . 2. dereceden olan I ve II nolu sistemlerde (her birinde 4' er adet bilinmeyen parametre mevcuttur .) giriş-çıkış davranışı darbe cevabı şeklindedir ve tüm parametreler iyi bir yakınsamayla doğrulukla elde edilebilmektedir . 3. dereceden olan III nolu sistemde ise (6 adet bilinmeyen parametre mevcuttur .) parametreleri yakınsaması çok yavaş olmaktadır .

Buradaki ilginç bir noktada ; aynı ön tanımlı değere sahip tüm metotların birbirlerine göre çok farklı olmalarına rağmen , çok uzun kestirim sürelerinde aynı performansla ($\sigma_{\delta g}$, $\sigma_{\delta \Sigma 1}$, $\sigma_{\delta \Sigma 2}$, $\sigma_{\delta y}$) aynı kestirim değerlerini vermesidir .

3.2. Sistem Belirleme İçin Kullanılan Deney Düzenegi

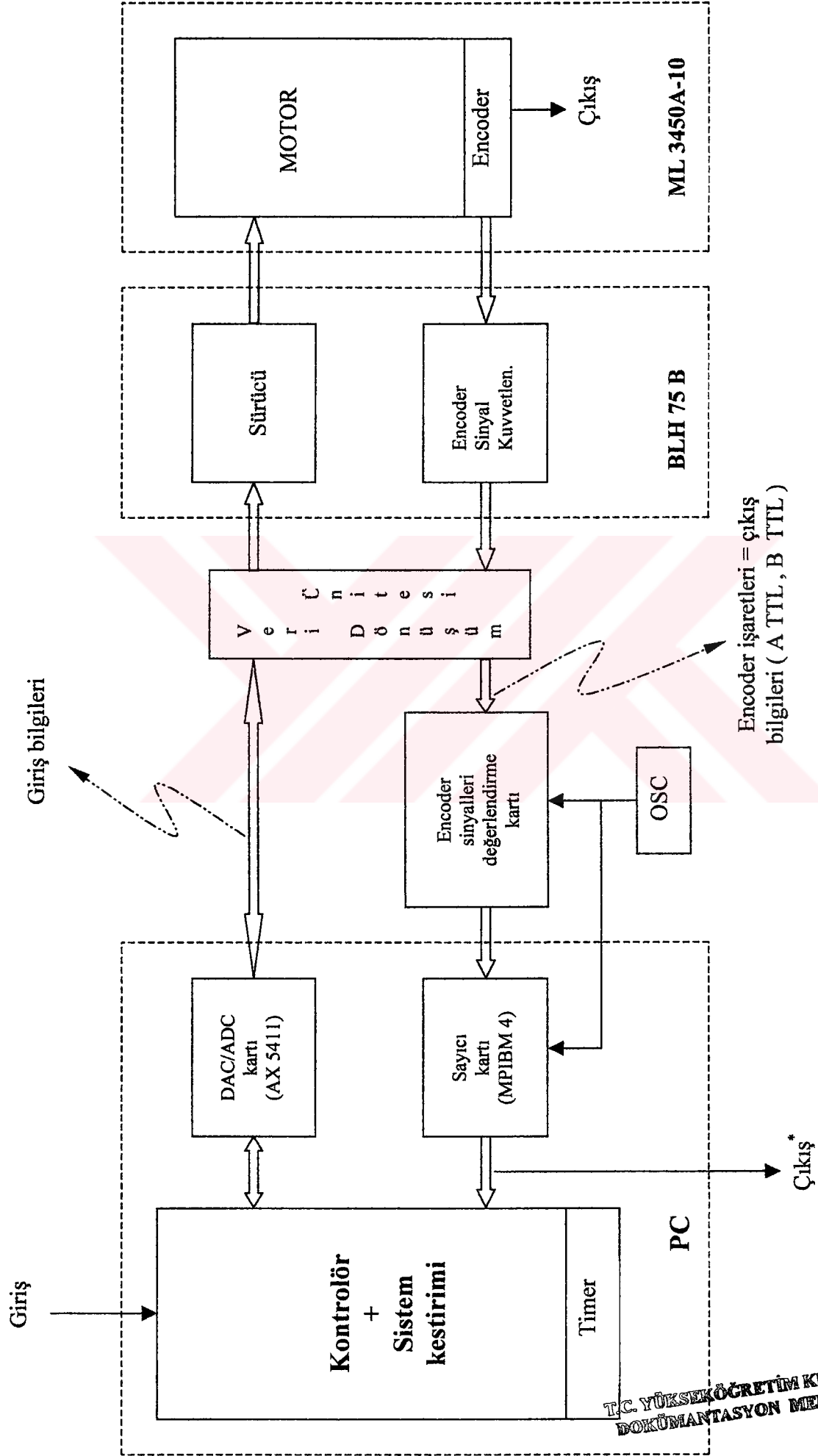
Bu bölüme kadar ele alınan konularda , birçok off-line ve on-line metodun incelemesi ve karşılaştırılması yapılmıştır . Bu bölümde ise sistem belirleme işleminin gerçek bir sistem üzerinde nasıl uygulandığından bahsedilecektir . Uygulamanın gerçekleştirilmesi amacıyla özel bir test düzenegi hazırlanmıştır .

Şekil 3.11' de genel devre şeması görülen test düzeneginde , modeli belirlenecek sistem olarak , bir motor (brushless AC servo motor) seçilmiştir . Motoru sürmek amacıyla uygun bir servo motor sürücüsü kullanılmıştır . Motora giriş bilgilerini göndermek için bir DAC/ADC kartından , elde edilen çıkış bilgilerini alabilmek içinse bir sayıcı kartından faydalanılmıştır . Alınan işaretleri istenen formlara sokmak için iki adet özel kart tasarlanmıştır . Ayrıca sistemde kontrolör görevini üstlenmek (kontrolör programını çalıştırmak) ve sistem belirleme işlemlerini (programlarını) çalıştırmak için bir adet PC kullanılmıştır . Bu PC aynı zamanda üzerinde , biraz önce bahsedilen DAC/ADC ve sayıcı kartlarını da barındırmaktadır .

Test düzeneginde elde edilen giriş ve çıkış bilgileri yine düzenegin üzerindeki PC' de depolanmakta ve tüm sistem belirleme işlemi yine burada gerçekleştirilmektedir. Elde edilen model parametrelerine göre yeni giriş işareti üretilmekte , böylece motorun yani sistemin davranışları kontrol altında tutulabilmektedir .

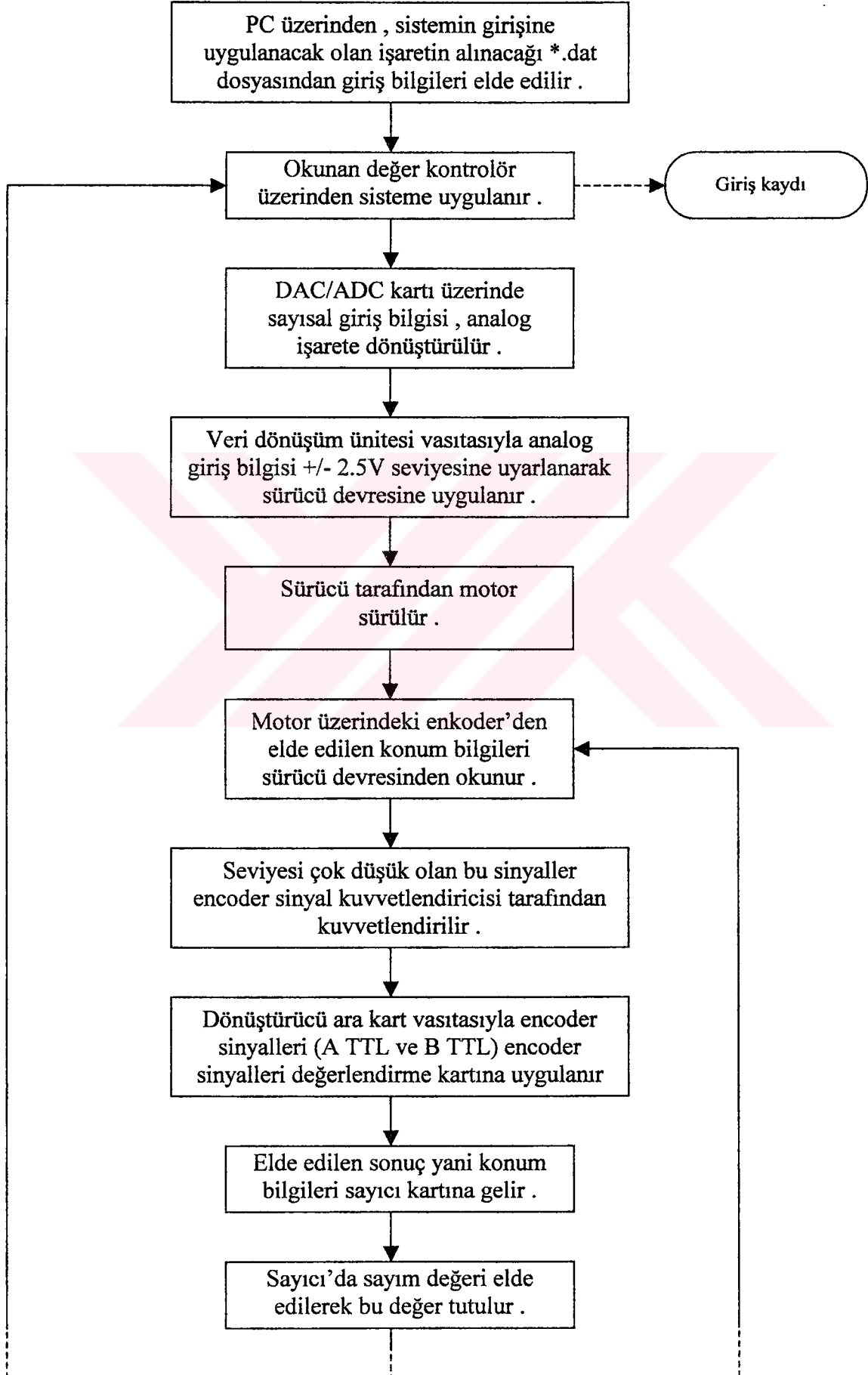
İlerleyen bölümlerde bu test düzeneginin nasıl oluşturulduğu , işlem akışının nasıl gerçekleştiği ve her bir bloğun nasıl çalıştığı ile üstlendiği fonksiyonlar ele alınacaktır .

Aşağıdaki ilk şekilde kullanılan test düzeneginin genel devre şeması , ardından gelen Çizelge 3.7' de de sistemin işlem akış diyagramı verilmektedir :

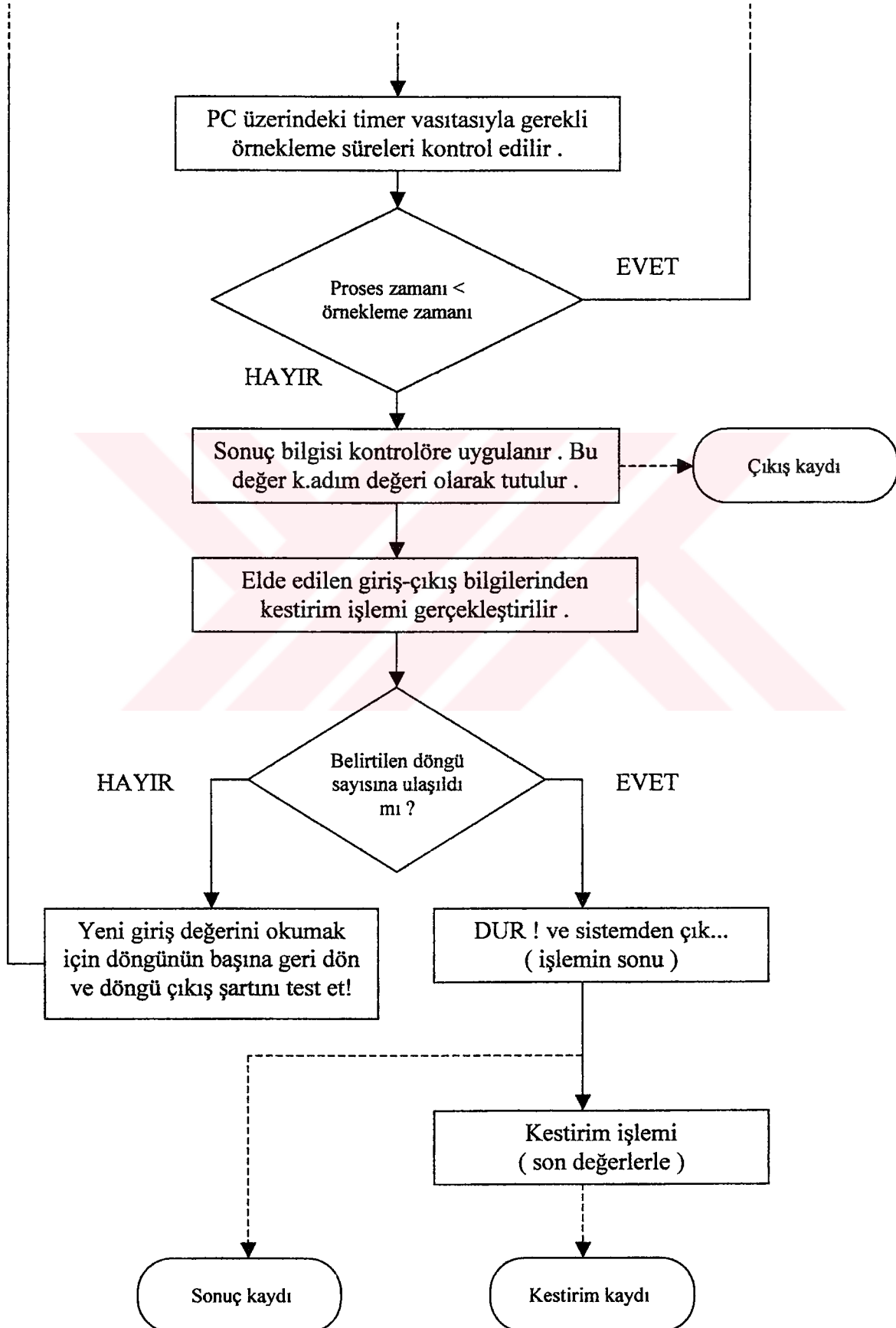


Şekil 3.11. Sistem Belirleme işleminin uygulamasının gerçekleştirildiği test düzeninin genel devre şeması

Çizelge 3.7. Kestirim işleminde kullanılan test düzeneği için işlem akış diyagramı



Çizelge 3.7. (Devam) Kestirim işleminde kullanılan test düzeneği için işlem akış diyagramı



Sistemde yer alan tüm komponentleri aşağıdaki gibi sıralayabiliriz :

1. PC Ünitesi (Kontrolör)
2. DAC/ADC Kartı (AX 5411 DAC/ADC Board <Axiom>)
3. Sayıcı Kartı (MPIBM 4 Multifunction I/O Card <Bytronic>)
4. Veri Dönüşüm Ünitesi
5. Encoder sinyalleri değerlendirme kartı
6. Servo motor sürücü (BLH75B <Digiplan>)
7. Motor_Brushless servo motor_ (ML3450A-10 <Daedal>)

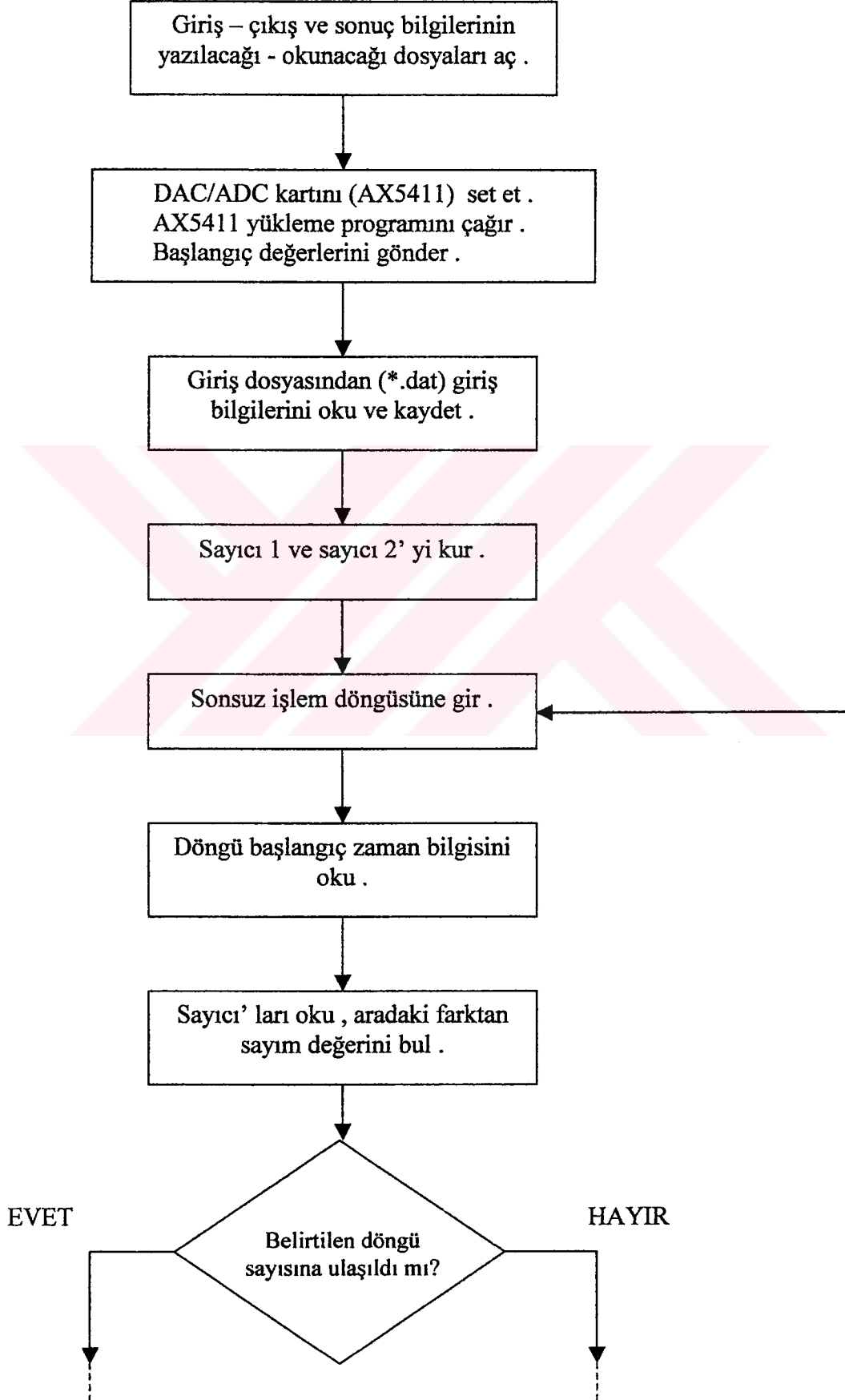
Bu elemanları sırasıyla ele alıp inceleyecek olursak :

3.2.1. PC Ünitesi

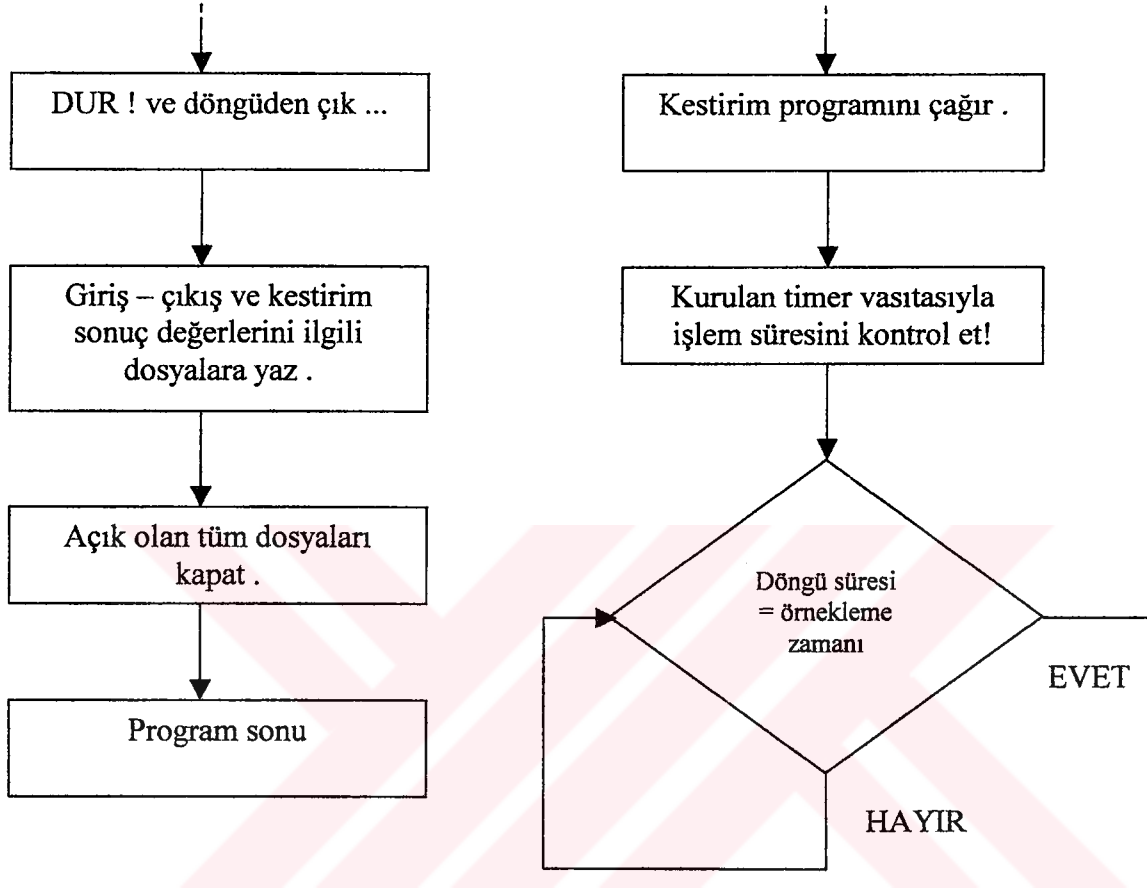
Bu test düzeneğinde PC' nin üstlendiği görev : kontrolör işlevini üstlenmek ve DAC/ADC kartı ile sayıcı kartını üzerinde barındırmaktır . Ayrıca CPU üzerinden alınan clock sinyalleri kullanılarak sistemdeki örnekleme zamanını kontrol etmek amacıyla bir dahili timer oluşturulmuştur . Çalışmada kullanılan PC , 486 DX4-100 işlemcisine sahip sıradan bir computer olmasına rağmen tüm işlemler için (kontrolör programlarının yürütülmesi ve I/O kartlar vasıtasıyla motorun sürülmesi) yeterli işlem gücünü ve hızını sağlamaktadır .

Burada kontrolör ile kastedilen , C dilinde hazırlanmış ve bilgisayar üzerinde yürütülen bir programdır . Kaynak kodu Ek 1' de verilen bu program için proses akış şeması aşağıdaki çizelgede (Çizelge 3.8.) görüldüğü gibidir :

Çizelge 3.8. PC üzerinde yürütülen kontrolör programının işlem akış diyagramı



Çizelge 3.8. (Devam) PC üzerinde yürütülen kontrolör programının işlem akış diyagramı



Çizelge 3.8’ de de görüldüğü üzere PC bloğu aynı zamanda giriş ve çıkış bilgilerinin üzerinde tutulduğu ve kestirim işleminin gerçekleştirildiği bir ünite de olmaktadır . Kestirim işlemi yine kontrolör gibi bir bilgisayar programının yürütülmesinden ibarettir ve bir alt program olarak ana kontrol programının içerisine çağırılmaktadır . Aynı zamanda Ek 1’ deki kaynak kodda görülen ;

```

outp(67,0);
tcount1=inp(64);
tcount2=inp(64);
tcount_first=(tcount2*256)+tcount1;
  
```

komut parçacığıyla CPU’ dan alınan clock sinyalleri kullanılarak bir “ timer ” elde edilmektedir . Bu işlem için öncelikle 67 nolu adrese “0” yazılarak sayıcı kilitlenmekte

(tutulmakta), peşisıra gelen komutlardada 64 nolu adresten 16 bitlik count değerinin önce düşük değerli bitleri (LSB) , sonra da yüksek değerli bitleri (MSB) okunmaktadır . Son satırdada okunan bu bilgilerden sayım değeri elde edilmektedir .

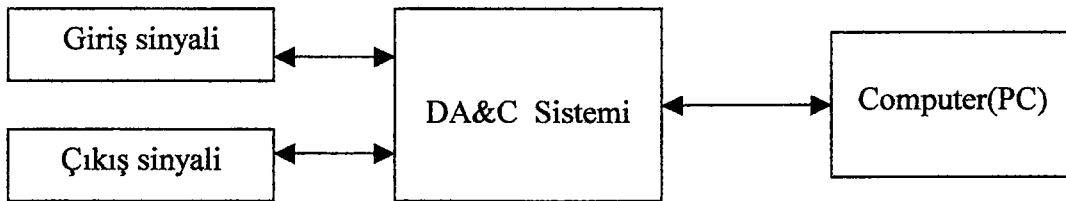
Bu küçük bloğu proses zamanını kontrol etmek istediğimiz program bloğunun başına ve sonuna koyarsak ; elde edilen iki *tcount* değerinin farkından , bir count değerinin ne kadar süre işgal ettiğini bilmek kaydıyla , geçen süreyi hesaplayabiliriz. Proses zamanına *t* dersek :

$$t = \frac{tcount_last - tcount_first}{\Delta} \quad (3.205)$$

Burada Δ belirttiğimiz gibi iki count değeri arasında geçen süredir _ki CPU' nun kendi bilgilerinden ya da basit bir ölçümle bulunabilmektedir .

3.2.2. DAC/ADC Kartı

Bir PC' nin , DA&C sistemi aracılığıyla prosten bilgi toplama prosedürünü aşağıdaki şekil ile gösterebiliriz :



Şekil 3.12. PC ile sistem arasındaki haberleşme prosedürü

Görüldüğü üzere ; DA&C sistemi , PC ile proses arasında bir “uygun formlara dönüştürme mekanizması” görevini üstlenmektedir .

Test düzeneğinde kullanılan DA&C sistemi ise Axiom Tech. firmasının ürettiği AX5411 DAC/ADC kartıdır . Bu cihaz 12 bit çözünürlükte çalışan 2 adet D/A

kanalına , 16 adet analog giriş ve 24' er adet dijital giriş ve çıkışa sahiptir . Üzerinde mevcut olan jumperlar vasıtasıyla A/D ve D/A çalışma aralıkları seçilebilmektedir . Çalışmamızda kullanılan A/D aralığı +/- 5V ve D/A aralığı ise 0-10V ' tur . Ancak motorumuzun her iki yöndede dönmesi arzulandığından veri dönüşüm ünitesi vasıtasıyla D/A çalışma aralığı olan 0-10V seviyesi +/- 2.5V seviyesine çekilmektedir . Böylece her iki yöndede dönüş sağlanmaktadır . (- işaretler için farklı , + işaretler için farklı <zıt yönde> dönüş sağlanmaktadır .) Bu durumda D/A çıkışından elde edilebilen en küçük gerilim (12 bit çalıştığımızı gözönünde bulundurarak ; $2^{12} = 4096 = \text{full DxA}$ çıkış değeri) ;

$$\text{D/A output} = \frac{1}{4096} \cdot 10 \cong 0.0024 \text{ V} \quad (3.206)$$

olmaktadır . Veri dönüşüm ünitesi vasıtasıyla çıkış aralığını +/- 2.5V değerine çektiğimize göre : 0-4095 aralığındaki 2048 değeri 0 V' a karşılık gelecektir ki , bu da motorun hiçbir yöne dönmemesi anlamına gelmektedir .

Kart ana program içerisinde *ax5411.h* kütüphanesiyle tanıtılmakta ve yine bu kütüphanede yer alan *ax5411* altprogramı ile sürülmektedir . Kartın programa tanıtılması (initialization) ve D/A çevirici olarak kullanılması için (tek kanal veya çift kanal) gerekli olan program parçacıkları ve prosedürler ile kartın kullanıcı arabiriminin pin bağlantıları Ek 2' deki bilgilerde verilmiştir .

3.2.3. Sayıcı Kartı

Düzenlerimizde sayıcı olarak kullandığımız kart , Bytronic firmasının ürettiği MPIBM4 Multifunction I/O kartıdır . Üzerinde üç adet 16 bitlik sayıcı bulunmaktadır . Sayıcı / Timer işlemcisi olarak 8253 kullanılmaktadır . Bu üç adet birbirinden bağımsız sayıcı 5 MHz' e kadar hızlarda geriye doğru sayım yapabilmektedir . Kart kendi üzerinde 4 MHz' lik bir osilatör bulundurmaktadır . Ancak çalışmamda bu osilatör devre dışı bırakılmıştır .

Kartın 40 pinlik bir kullanıcı arabirim soketi mevcuttur . Soketin pin bağlantıları Ek 3' te görülmektedir . Bu kartı sistemimize uygun olarak adapte etmek için aşağıdaki bağlantılar gerçekleştirilmektedir :

- İlk olarak 11 ve 12 nolu pinler kısa devre edilerek internal clock işareti devre dışı bırakılır .
- Encoder sinyalleri değerlendirme kartına uyguladığımız osilatör (harici) işaretini bu kez 3 clock girişinden istenilen ikisine bağlanılır (örneğin 3 ve 4' e) .
- Encoder sinyalleri değerlendirme kartından elde edilen ileri ve geri sinyalleride bu sayıcı' ların gate' lerine bağlanılır .

Böylece ileri ya da geri sinyallerinden birisi lojik 1 olduğunda sayıcı' lardan ilgili olan geriye doğru saymaya başlar ve bu değer hafızada tutulur . İleri ya da geri sinyallerinden yalnızca birisi lojik 1 olabileceğinden , diğerinin lojik 0 olması ikinci sayıcı' nın değerinin set edilen değerde kalması demektir . Bu iki değer farkından +/- polaritesi ile birlikte konum bilgisi elde edilmektedir .

Bu işlemler kontrolör programı içerisinde aşağıdaki program parçacıkları ile sağlanmaktadır . İşlemlerin başlangıcında öncelikle sayıcı' larımızın kurulması ve set edilmesi gerekmektedir . Bunun için her bir sayıcı için aşağıdaki ifade kullanılır :

```

1   cnt0_start=50000;
2   cnt_high=cnt0_start/256;
3   cnt_low=cnt_start-(cnt0_high*256);
4   outp(modecnt_id,modecn0);
5   outp(count0_id,cnt0_low);
6   outp(count0_id,cnt0_high);

```

Görüldüğü gibi ilk satırda 0 nolu sayıcı' ya 50000 başlangıç değeri atanmıştır . Sayıcı 16 bitlik , ancak işlemler 8 bit üzerinden gerçekleştirildiği için 2. ve 3. satırlarda başlangıç değerinin düşük ve yüksek seviyeli bitleri (LSB ve MSB) elde edilmektedir . 4. satırda ise *modecnt_id* adresli porta *modecn0* kontrol kelimesi yazılmaktadır .

Sistemimizde sayıcı kartının base adresi 640 decimal' dir. Bu durumda *Modecnt_id* 640+3 olmaktadır . Base adress +0 , +1 , +2 nolu adreslerde sayıcı' ların adresleridir . *modecn0* ile belirtilen sayı ise 643 nolu adrese yazılacak kontrol kelimesidir . 8 bitten oluşan bu kelimenin alabileceği değerler Ek 3' te görülmektedir . Örneğin kontrol kelimesine 48 yazarsak bu şu anlama gelmektedir : 0 nolu sayıcı seçilmiş ve bu sayıcı mode 0'da önce LSB , ardından MSB okunacak-yüklenecek şekilde ayarlanmıştır . Okunacak-yüklenecek değer binary formunda olacaktır .

Aynı şekilde ; elde edilen count değerinin okunup değerlendirilmesi gerekmektedir . Bu işlem şu program parçasıyla gerçekleştirilmektedir :

```

1   outp(modecnt_id,0);
2   ccount1=inp(count0_id);
3   ccount2=inp(count0_id);
4   ccount=(ccount2*256)+ccount1;

```

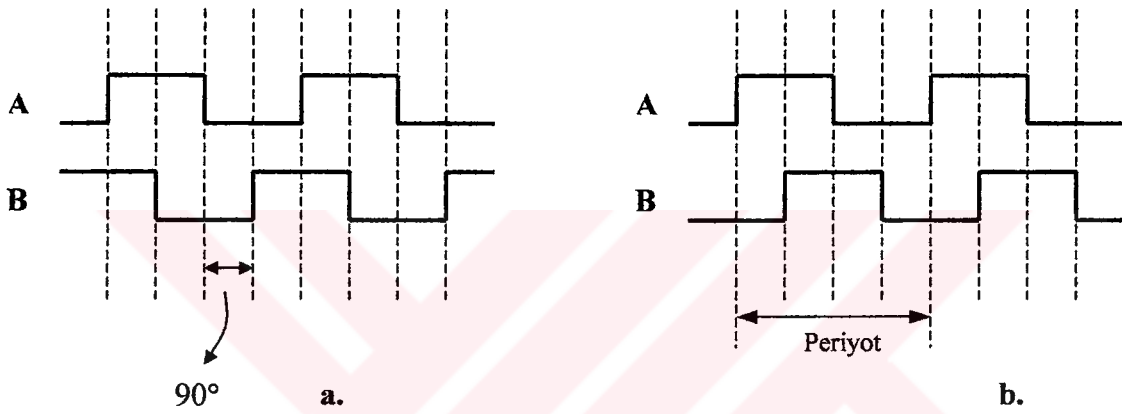
modecnt_id (643) adresine kontrol kelimesi olarak 0 yazarsak 0 nolu sayıcı kilitlenmektedir . Ardından 2. ve 3. satırlarda önce LSB sonrada MSB değeri okunmakta ve 4. satırda da bu değerlerden 16 bitlik sayım değeri elde edilmektedir . Diğer sayıcı içinde aynı bloktan kullanılarak ikinci count değeri elde edilmektedir . Bu iki count değeri arasındaki farktan gerçek sonuç bilgisine ulaşılmaktadır .

3.2.4. Enkoder Sinyallerini Değerlendirme Kartı

Bu ara kartın kullanılmasındaki amaç , sürücü üzerindeki kuvvetlendirici tarafından kuvvetlendirilen enkoder sinyallerinden dönüş yönü ve konum bilgisini elde etmektir . Bu işlem için kullanılan iki adet sinyal mevcuttur . A ve B adı verilen bu iki sinyal arasında 90° lik bir faz farkı vardır . Ayrıca kartı kullanmamızdaki ikinci bir amacımızda bir periyotta üretilen darbe sayısını 1' den 4' e çıkarmaktır . Böylece elde edilen çözünürlükte artırılmış olmaktadır .

Sürücü içerisindeki enkoder sinyal kuvvetlendiricisinden elde edilen işaretler RS485 formatındadır . Oysa düzeneğimizde kullanılan sistemler TTL girişlere sahiptirler . Bu yüzden RS 485 formatındaki A ve B sinyalleri , veri dönüşüm ünitesi ve üzerindeki PS 75176 BN entegresi vasıtasıyla TTL formatına dönüştürülmektedir .

Dönüşüm sonrası A ve B sinyallerinin durumu aşağıdaki hallerden birisi gibi olmaktadır :



Şekil 3.13. a. ve b. Motor çıkışından elde edilen TTL formatındaki encoder sinyalleri

Yukarıdaki şekillerden Şekil 3.13.-a ileri yönde hareket durumunda elde edilen , Şekil 3.13.-b ise geri yönde hareket durumunda elde edilen sinyallerin konumunu göstermektedir .

Şekiller dikkatle incelendiğinde ; B ve A sinyallerinin 00 , 10 , 11 , 01 , 00 , şeklinde bir sıra ile değişim gösterdiği zaman ileri yönde bir hareketin , 00 , 01 , 11 , 10 , 00 , şeklinde bir değişim gösterdiğinde ise geri yönde bir hareketin söz konusu olduğu görülmektedir . İleri yönde harekette ileri fonksiyonunu 1 , geri fonksiyonunu 0, geri yönde harekette de ileri fonksiyonunu 0 , geri fonksiyonunu 1 alırsak ve bunun dışındaki tüm işaret değişimlerinde ileri ve geri fonksiyonlarını 0 kabul edersek ; aşağıdaki gibi bir durum-geçiş çizelgesi oluşturmamız mümkündür .

Çizelge 3.9. Encoder sinyallerini değerlendirmek için oluşturulan durum-geçiş çizelgesi

Girişler		Önceki Durum		Sonraki Durum		İleri	Geri	D _B	D _A
B	A	Q _B	Q _A	Q _B	Q _A				
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	0
0	0	1	1	0	0	0	0	0	0
0	1	0	0	0	1	1	0	0	1
0	1	0	1	0	1	0	0	0	1
0	1	1	0	0	1	0	0	0	1
0	1	1	1	0	1	0	1	0	1
1	0	0	0	1	0	0	1	1	0
1	0	0	1	1	0	0	0	1	0
1	0	1	0	1	0	0	0	1	0
1	0	1	1	1	0	1	0	1	0
1	1	0	0	1	1	0	0	1	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	1	1	0	1	1	1
1	1	1	1	1	1	0	0	1	1

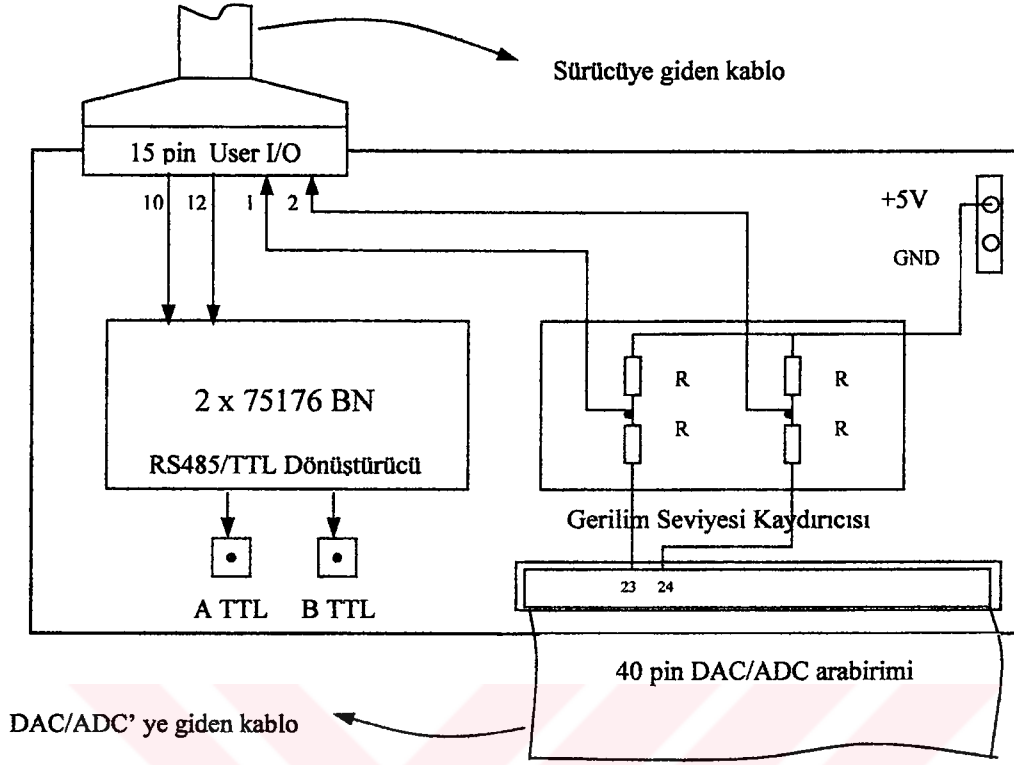
KAYNAK : Armağan (1998) , Bilgisayar uyumlu enkoder sinyalleri sayıcı kartı, Bursa.

Bu çizelgeyi Carnaugh diyagramları ile gerçekleyip , çözecek olursak ; (Bakınız Ek-4)

$$\text{İleri} = (A \oplus Q_B) \cdot (B \otimes Q_A)$$

$$\text{Geri} = (B \oplus Q_A) \cdot (A \otimes Q_B)$$

fonksiyonları elde edilmektedir . Bu sonuçlarla gerekli devreyi lojik kapılar kullanarak gerçekleyecek olursak aşağıdaki sistem elde edilmektedir :



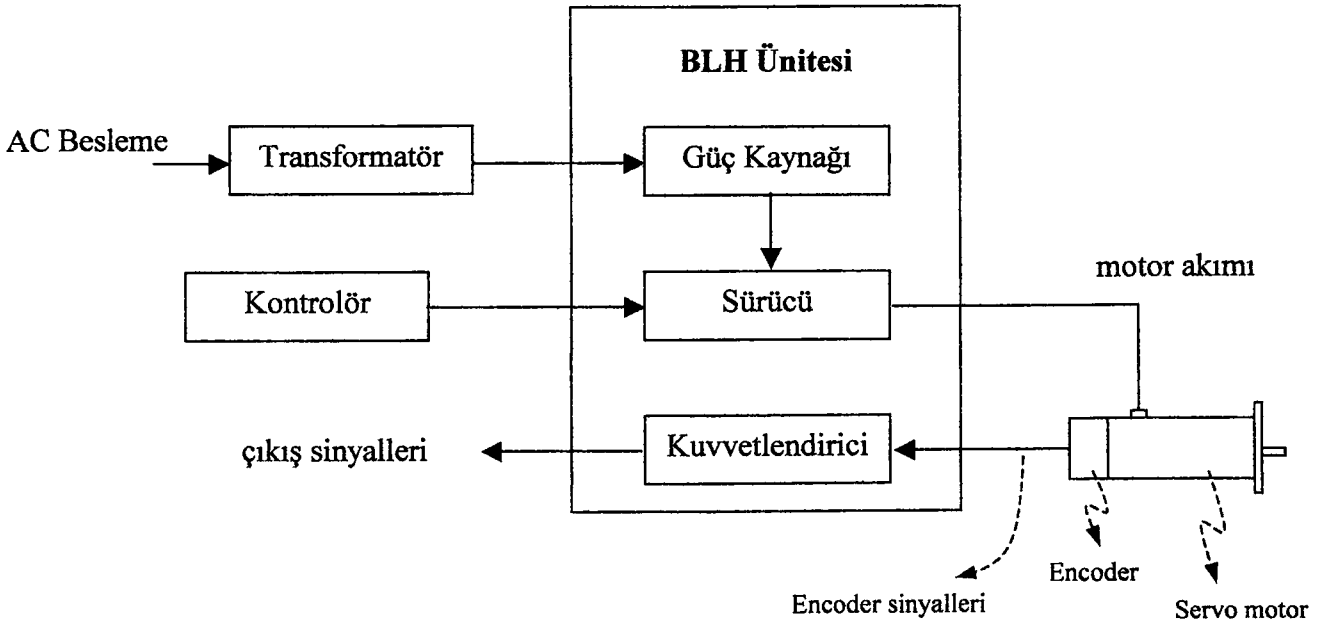
Şekil 3.15. Test düzeneğinde kullanılan Veri Dönüşüm Ünitesinin devre şeması

Yukarıdaki devrede görüldüğü üzere sürücünün kullanıcı arabirim socketinin 10 ve 12 nolu pinlerinden alınan A ve B encoder sinyalleri RS485/TTL dönüştürücü bloğunda A TTL ve B TTL sinyallerine dönüştürülmektedir. Bu blok içerisinde iki adet 75176 BN entegresi mevcuttur. Entegreye ilişkin katalog bilgileri Ek 7' de verilmektedir. Sürücüden gelen işaretler bir direnç üzerinden geçirilerek, akım formundan gerilim formuna sokulmaktadır. Yani akım şeklindeki işareten kare dalga elde edilmektedir. Seviyesi çok düşük olan bu işaret 75176 BN entegresi tarafından TTL formatına dönüştürülmektedir. Elde edilen TTL formatındaki işaretler kart üzerinden encoder sinyalleri değerlendirme kartına uygulanmaktadır. Aynı zamanda DAC/ADC kartından elde ettiğimiz analog değer (ki bu değer 0-10V arasındadır.), seviyesini kaydırarak $\pm 2.5V$ aralığına çekilmektedir. Bu işlem için kullanılan gerilim seviyesi kaydırıcısı ise bir gerilim bölücüdür. Alt ucu kart üzerindeki +5V' a bağlı olan eşit değerli gerilim bölücünün üst ucunu DAC çıkışından alınan işarete bağladığımızda, orta uçtan alınan işaret $\pm 2.5V$ aralığında değişmektedir.

Elde edilen yeni değerler sürücü arabirim soketinin 1 ve 2 nolu 'analog command input' pinlerine uygulanmaktadır . Bu command işaretleri Şekil 3.18.'deki *command+* ve *command-* işaretleridir . Yani sürücüye uygulanan işaret bu iki işaret arasındaki fark kadar olmaktadır . Ancak bu çalışmada D/A kanallarından sadece birisi kullanılmıştır . Bu durumda diğer kanalın işareti (V_2) her zaman sıfır değerinde kalacaktır . Böylece sürücüye uygulanan işaret ($V_2 - V_1$)' den ; V_2 değeri sıfır olduğundan , her zaman $\pm V_1$ değeri yani $\pm 2.5V$ aralığında değişecektir .

3.2.6. Sürücü – BLH75B

BLH75B sürücüsü , brushless (fırçasız) servo motorları sürmek için kullanılan, darbe genişlik modülasyonu (PWM) çalışan bir cihazdır . Cihazın girişine analog bir işaret uygulanmaktadır . Bu işaret bir servo kontrolör tarafından (sistemimizde bu , PC tarafından üretilmektedir.) sağlanmaktadır . Uygulanan bu kontrol işaretiyle sürücü devresi , motorun istenen çalışma şartını sağlayan akımı üretmektedir . Motor üzerinden encoder vasıtasıyla elde edilen bilgi bir geribesleme ile tekrar sürücüye girmektedir . Bu sinyal encoder sinyal kuvvetlendiricisi ile kuvvetlendirilerek , çıkış olarak dışarıya verilmektedir . Tüm bunları bir şekil üzerinde toparlayacak olursak :



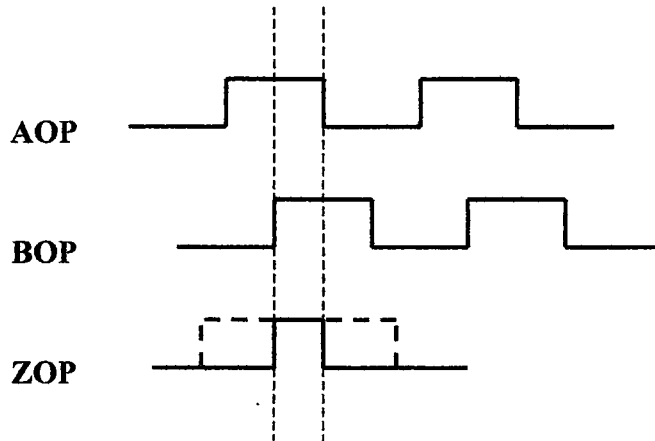
Şekil 3.16. Test düzeneğindeki sürücü ünitesinin devre üzerindeki konumu

Sürücü cihazı , üzerindeki jumper seti sayesinde istenilen çeşitli çalışma formlarında kullanılabilir . Tüm bu çalışma formlarına ilişkin jumper konumları ve çizelgesi Ek 5’ te verilmektedir . Bu çalışmada sürücü Ek 5’ teki formlardan “Torque Amp” modunda kullanılmıştır .

Adından da anlaşılacağı üzere bu modda motora uygulanan tork miktarı kontrol edilmektedir . Yani motoru istediğimiz şekilde kontrol edebilmek için , üzerine uyguladığımız güç miktarının şiddeti ve yönü kullanılmaktadır .

Sürücünün AC şebekesi ve motor bağlantısı şeması ile üzerindeki pin bağlantılarını gösteren şekiller Ek 5’ te görülmektedir . Bu şekillere bakıldığında sürücünün 25 pinlik bir motor geribeslemesi soketine ve 15 pinlik bir kullanıcı I/O soketine sahip olduğu görülmektedir . Bu soketlerin her bir pin’ inin adı , fonksiyonu ve uygulanacak işaret tipleri Ek 5’ teki çizelgelerde görülmektedir . Ayrıca yine sürücü üzerinde denge , kazanç , zaman sabiti ve damping ayarlarının yapılmasını sağlayan potansiyometreler mevcuttur .

Ek 5’ teki kullanıcı I/O soketi pin çizelgesinde görüldüğü üzere sürücüden çıkan üç adet encoder çıkışı mevcuttur . Bunlar AOP , BOP ve ZOP işaretleridir . Bu üç sinyali bir arada gösterecek olursak ;

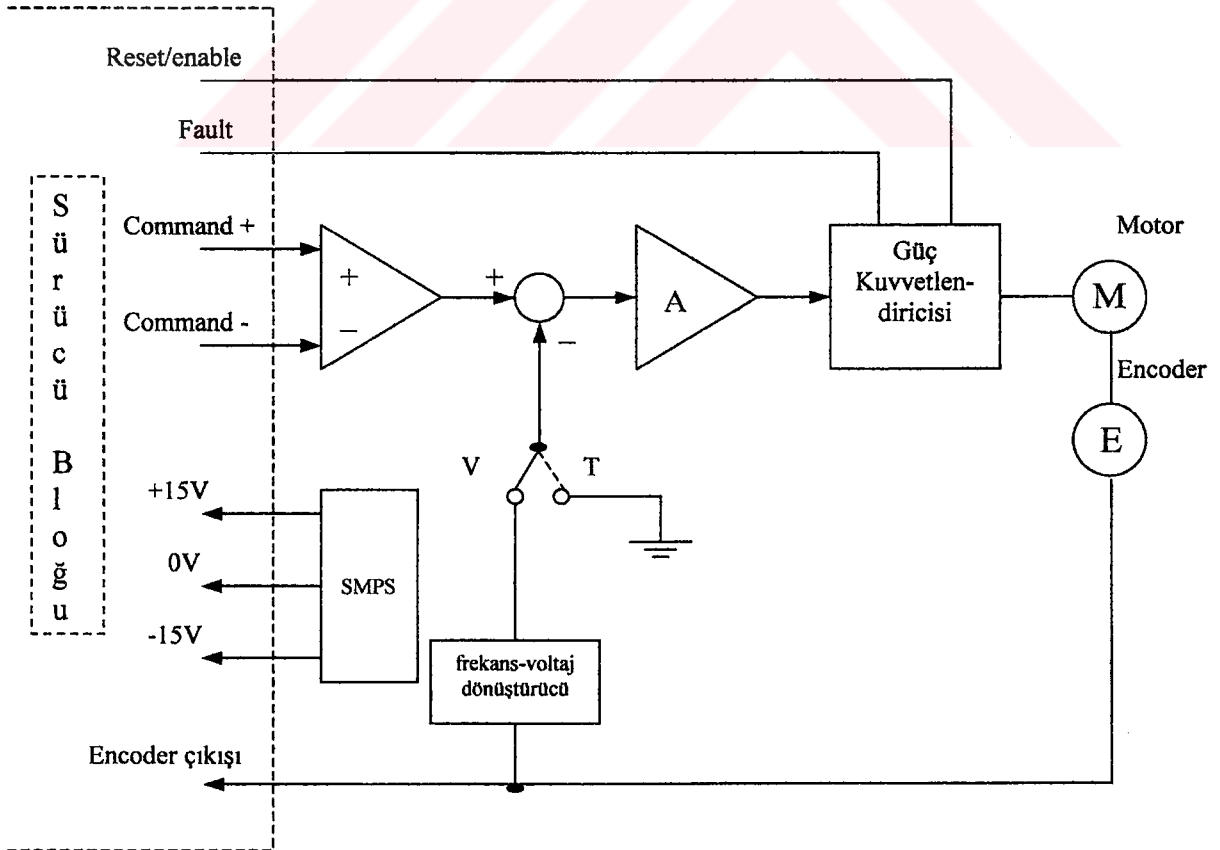


Şekil 3.17. Sürücü devresinin çıkışından elde edilebilen işaretler

AOP işaretinin BOP' tan önce gelmesi saat yönünde bir dönüş olduğunu göstermektedir . ZOP işareti ise AOP işaretinin $\frac{1}{4}$ ' lük bir periyodunda sadece AOP ve BOP' un her ikisinin de yüksek değerde (lojik 1) olması durumunda lojik 1 olmakta, diğer tüm durumlarda lojik 0 değerini almaktadır . Ancak bu çalışmada bahsedilen işaretlerden sadece AOP ve BOP kullanılmıştır .

3.2.7. Motor _ML3450A-10_

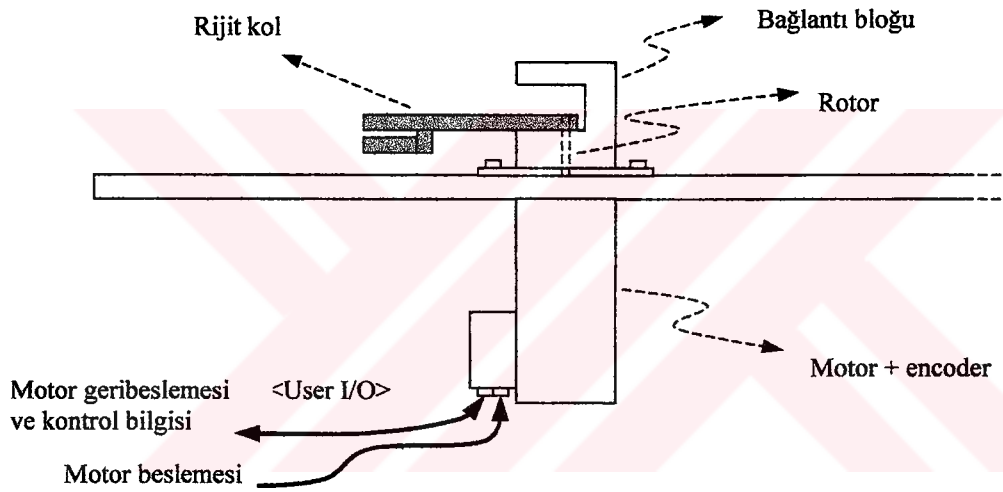
Çalışmada kullanılan motor (ML3450A-10 Daedal) , bir fırçasız (brushless) servo motordur . İki adet analog komut girişiyle (command + , command -) çalışmaktadır . Tork kontrolü ve hız kontrolü olmak üzere iki farklı çalışma modunda kullanılabilir . İçerisinde gömülü halde bir artan (incremental) encoder bulunmaktadır . Bu encoderden elde edilen işaret bir frekans-voltaj dönüştürücüsü ile hız kontrolü için sisteme geri besleme yapılabilir . Motor ve öncesindeki sürücüsünü bir şekil üzerinde gösterecek olursak :



Şekil 3.18. Test düzeneğindeki motor ve sürücüsünün bağlantı şeması

Ele aldığımız bu motor 13 Nm tork üretebilmekte ve 7200 rpm devire kadar çıkabilmektedir . Motor , command girişlerine uygulanan gerilimlerin farkı ölçüsünde harekete geçmektedir. Örneğin V_1 girişine $+V$, V_2 girişine $-V$ uygulandığında motor üzerine 2V kadar bir gerilim uygulanmış olmaktadır . Motor hakkındaki daha fazla ve ayrıntılı bilgi Ek 6' daki çizelgelerde görülmektedir .

Deney düzeneğinde motor yatay bir düzleme (bir masaya) dik olarak monte edilmiştir . Rotoruna , üzerine kol vb. bağlamak için bir bağlantı bloğu ve temsili bir yük olarak rijit bir kol bağlanmıştır . Bu bağlantıları aşağıdaki şekil üzerinde görmek mümkündür :



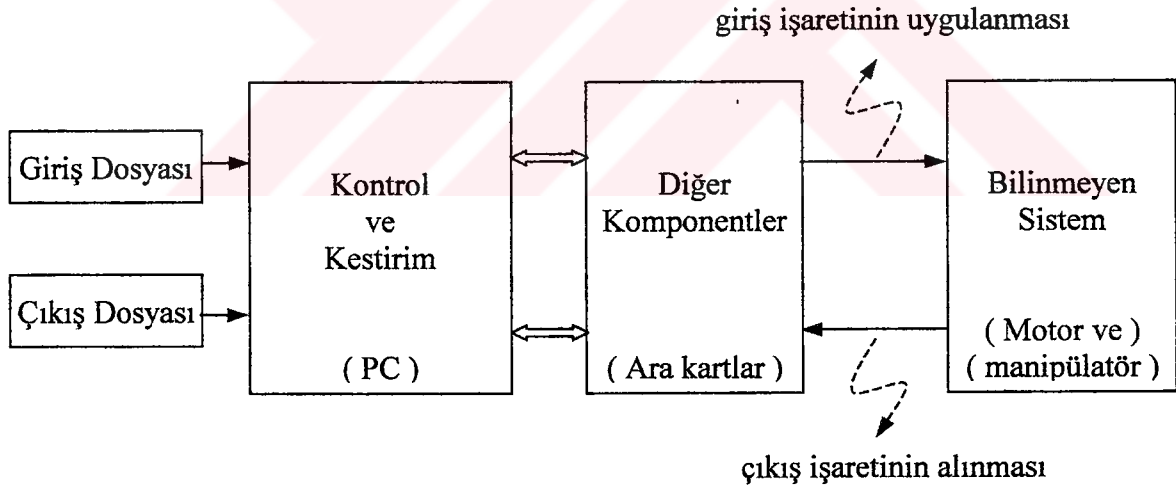
Şekil 3.19. Motorun deney düzeneği üzerindeki bağlantısı

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Deney Düzenegi

Bu bölüm içerisinde ; elde edilen sistem verilerinden (giriş – çıkış bilgileri) sistem belirleme işleminin nasıl gerçekleştirildiğinden , parametre kestiriminde hangi yöntemin uygulandığından ve model seçiminin nasıl gerçekleştirildiğinden bahsedilecektir . Ayrıca bölüm sonunda elde edilen sonuçların değerlendirilmesi de yapılmaktadır .

Kestirim işleminin gerçek bir sistem üzerinde uygulamasını gözlemlemek amacıyla bir test düzenegi oluşturulmuştur . Bölüm 3.2' de ayrıntılarıyla ele alınmış olan bu test düzenegini , kestirim işleminin nasıl gerçekleştirildiğini göstermek amacıyla genel bir blok diyagram halinde verecek olursak ;



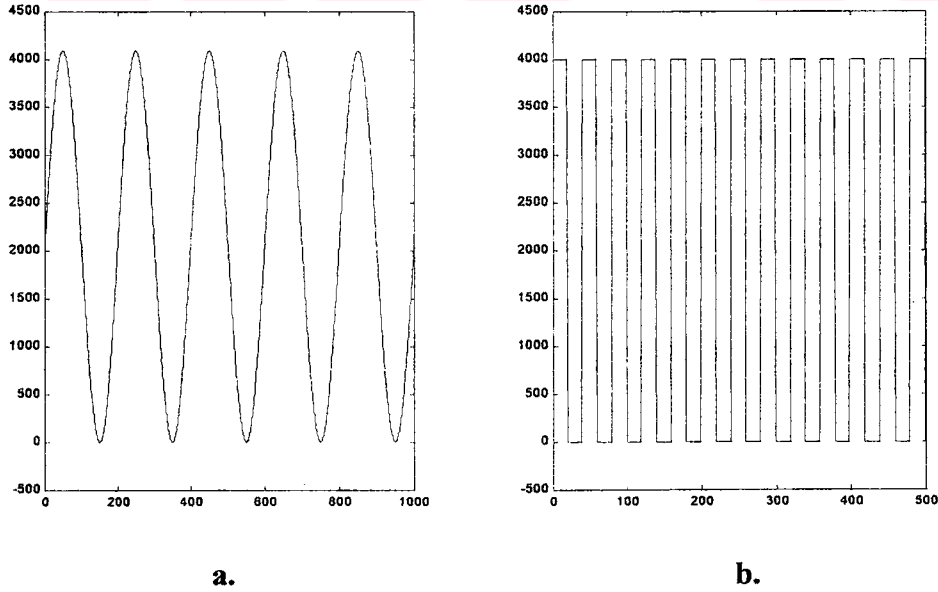
Şekil 4.1. Kestirimin gerçekleştirildiği deney düzeneginin genel blok diyagramı

Şekilden de görüleceği üzere test düzeneginde kontrol ve kestirimi işlemlerini yürütmek için bir PC kullanılmaktadır . Giriş – çıkış bilgilerinin toplanması amacıyla kullanılan çeşitli ara kartlar mevcuttur . Bilinmeyen sistem olarak bir motor ve sürücüsü kullanılmıştır .

Bir kestirim işleminin gerçekleştirilebilmesi için gerekli olan şeylerden ilki , sistemin karakteristiklerini belirleyeceğimiz giriş ve çıkış bilgileridir . Diğeri ise sistemi temsil edecek olan modelin belirlenmesidir . Bunun ardından da eldeki verilerle kestirim işlemi gerçekleştirilmekte ve bulunan parametreler yardımıyla model oluşturularak sistemin davranışı belirlenebilmektedir . Şimdi bu işlemleri sırasıyla ele alacak olursak :

4.1.1. Giriş – çıkış verilerinin elde edilmesi

Bu çalışmada kullanılan tüm giriş bilgileri test düzeneğinden ayrı bir ortamda (bilgisayar yardımıyla) üretilip , bir data dosyası haline getirilmektedir . Bu dosya input.dat ismiyle saklanmaktadır . Elde edilen bu giriş dosyası daha sonra Bölüm 3' te ayrıntılarıyla ele alınmış olan gerçek bir sistem üzerine uygulanmaktadır . Böylece sistem üzerinden elde edilen çıkış bilgisi , giriş bilgisi ile birlikte çevrimiçi kestirim işleminde kullanılmaktadır . Aynı zamanda bu veriler yine bir data dosyası haline getirilerek output.dat adı altında saklanmaktadır . Burada ayrık zamanda çalışıldığından dolayı tüm veriler belirli örneklemelemlerle elde edilmektedir . Aşağıdaki şekilde iki adet örnek giriş işareti görülmektedir :



Şekil 4.2-a. ve b. Kestirim işleminin gerçekleştirildiği sisteme uygulanan çeşitli giriş işaretleri

Sistem belirleme amacıyla kullanılan giriş işareti Şekil 4.2-a ve b' de görüldüğü üzere sinüzoidal ya da kare dalga formunda olabilmektedir . Kestirim işlemi ayrık zamanda gerçekleşeceğinden bu işaretler belirli ve istenilen sayıdaki adım uzunluğundadır . İşaretlerin genlik değerleri ise DAC/ADC kartının çözünürlüğü olan 2^{12} değerinin izin verdiği $0 - 4095$ aralığında olmalıdır . Bu değer daha sonra analog gerilim seviyesine dönüştürülmektedir . İşaretlerin frekansları da isteğe bağlı olarak değiştirilebilmektedir . Bu değer sistemi uyarıya yetecek kadar büyük olmalı ancak sistemi kararsız yapacak kadarda büyük tutulmamalıdır .

4.1.2. Sistem Modelinin Belirlenmesi

Bölüm 3' te ayrık zaman sistemler için transfer fonksiyonu aşağıdaki gibi tanımlanmıştır :

$$H(z) = \frac{a_0 + a_1.z^{-1} + a_2.z^{-2} + \dots + a_m.z^{-m}}{1 + b_1.z^{-1} + b_2.z^{-2} + \dots + b_m.z^{-m}} \quad (4.1)$$

Bu fonksiyonda ; işlem kolaylığı açısından $a_0 = 0$ alınmıştır . Yapılan deneyler esnasında sistem için 2. ve 3. derece olmak üzere iki farklı model kullanılmıştır . Deney düzeneğimizde bilinmeyen sistem olarak bir fırçasız servo motorun kullanıldığı belirtilmiştir . Motorların matematiksel modeli için kullanılan durum denklemleri ve bunların vektör-matris şeklindeki düzenlemeleri aşağıdaki gibi gösterilebilmektedir (Kuo,1995)

$$\frac{di_a(t)}{dt} = \frac{1}{L_a}.e_a(t) - \frac{R_a}{L_a}.i_a(t) - \frac{1}{L_a}.e_b(t) \quad (4.2)$$

$$T_m(t) = K_i.i_a(t) \quad (4.3)$$

$$e_b(t) = K_b.\frac{d\theta_m(t)}{dt} = K_b.\omega_m(t) \quad (4.4)$$

$$\frac{d^2\theta_m(t)}{dt^2} = \frac{1}{J_m}.T_m(t) - \frac{1}{J_m}.T_L(t) - \frac{B_m}{J_m}.\frac{d\theta_m(t)}{dt} \quad (4.5)$$

$$\begin{bmatrix} \frac{di_a(t)}{dt} \\ \frac{d\omega_m(t)}{dt} \\ \frac{d\theta_m(t)}{dt} \end{bmatrix} = \begin{bmatrix} -R_a & -K_b & 0 \\ L_a & L_a & 0 \\ K_i & -B_m & 0 \\ J_m & J_m & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ L_a \\ 0 \\ 0 \end{bmatrix} \cdot e_a(t) - \begin{bmatrix} 0 \\ 1 \\ J_m \\ 0 \end{bmatrix} \cdot T_L(t) \quad (4.6)$$

Bu denklemde ;

i_a = motorun armatür akımı	L_a = motorun armatür endüktansı
R_a = motorun armatür direnci	e_a = motora uygulanan gerilim
T_L = yük momenti	T_m = motor momenti
ω_m = rotorun açısal hızı	θ_m = rotorun açısal konumu
B_m = viskoz sürtünme katsayısı	K_i = moment sabiti
K_b = zıt elektromotor sabiti	J_m = motorun eylemsizlik momenti
e_b = zıt elektromotor gerilimi	

Tüm bu ifadeler doğru akım motorları için geçerlidir . Deney düzeneğimizde kullandığımız motor ise bir AC motordur . Buna rağmen motorun üreticisine ait kullanım kılavuzlarında da motorun modellenmesi için yukarıdaki doğru akım eşitlikleri kullanılmaktadır . (4.6)'daki matrissel ifadeden anlaşılacağı gibi bir doğru akım motoru 2. dereceden bir model ile temsil edilebilmektedir . Ancak motorun içerisinde bulunduğu çalışma şartlarında etkin olan birçok farklı etkenden dolayı sistemin derecesi artabilmektedir . Bu çalışmada hem ikinci hem de üçüncü dereceden bir model kullanılarak sistem üzerinde böyle bir durum olup olmadığı kontrol edilmiştir . Böylece hangi modelin daha doğru olduğunu bulmamız , aynı sistemi en doğru haliyle belirleyebilmemiz mümkün olmaktadır . Bu modelleri ayrı ayrı ele alacak olursak ;

i. 2. dereceden sistem modeli

2. dereceden modeli elde etmek için (4.1) denkleminde $m = 2$, $n = 2$ alınarak:

$$H_{M1}(z) = \frac{a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}{1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}} \quad (4.7)$$

transfer fonksiyonu elde edilecektir . Böylece bulmamız gereken 4 adet bilinmeyen parametremiz (a_1 , a_2 , b_1 , b_2) olmaktadır .

ii. 3. dereceden sistem modeli

3. dereceden sistem modelini elde etmek içinse (4.1) denkleminde $m = n = 3$ almak yeterlidir . Böylece aşağıdaki ayrık zaman transfer fonksiyonu elde edilmektedir :

$$H_{M2}(z) = \frac{a_1.z^{-1} + a_2.z^{-2} + a_3.z^{-3}}{1 + b_1.z^{-1} + b_2.z^{-2} + b_3.z^{-3}} \quad (5.8)$$

Görüldüğü gibi bu denkleminde parametre sayısı 6' ya çıkmıştır . Bu da beklendiği üzere işlem yükünün artmasına sebep olmaktadır .

İlerleyen bölümlerde her iki model içinde kestirim işlemi gerçekleştirilecek ve hangisinin daha iyi sonuçlar verdiği gözlenecektir . Bu gözlem sonucuna göre modellerden birisi seçilerek bu model için kestirim yapılacaktır .

4.1.3. Kestirim işleminin gerçekleştirilmesi

Şekil 4.1' de görüleceği üzere test düzeneği üzerinde kontrol ve kestirim işlemleri bir PC üzerinde gerçekleştirilmektedir . Kestirim işlemi , kaynak kodu Ek 1' de görülen ana kontrol programının içerisinde bir alt program olarak yer almaktadır . Alt program içerisinde kullanılan sistem belirleme metodu ise recursive En Küçük Kareler metodudur . E.K.K. metodunun tercih edilmesinin nedeni ; sisteme kolayca uygulanabilmesi (kullanılan bilgisayar programlama dilinde uygulaması güç olan matematiksel fonksiyonlar içermemesi , değişken sayısının çok olmaması vb.) ve hızlı bir biçimde yeterli bir yaklaşıklık sunarak sonuç üretebilmesidir .

2.dereceden sistem modelinde 4 adet parametre söz konusu olduğundan , E.K.K. metodunu uygularken ϕ durum matrisi iterasyon içerisinde aşağıdaki gibi oluşturulmaktadır . (u giriş vektörü , y çıkış vektörüdür .)

$$\varphi_k^T = [u_{k-1} \quad u_{k-2} \quad -y_{k-1} \quad -y_{k-2}] \quad (4.9)$$

Matris içerisinde kullanılan çıkış işareti y değişkenlerinin önüne “-” işaretinin gelmesinin nedeni aşağıdaki denklem takımının çözümüyle açıklanabilir :

$$\frac{y(z)}{u(z)} = H_{MP}(z) = \frac{a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}{1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2}} \quad (4.10)$$

$$z^0 \cdot y(z) + b_1 \cdot z^{-1} \cdot y(z) + b_2 \cdot z^{-2} \cdot y(z) = a_1 \cdot z^{-1} \cdot u(z) + a_2 \cdot z^{-2} \cdot u(z) \quad (4.11)$$

halini alır . Bu denklemde z^{-1} bir tutucuyu temsil ettiğine göre denklemi aşağıdaki formda yazmak mümkündür :

$$y_k + b_1 \cdot y_{k-1} + b_2 \cdot y_{k-2} = a_1 \cdot u_{k-1} + a_2 \cdot u_{k-2} \quad (4.12)$$

düzenleyecek olursak ;

$$y_k = a_1 \cdot u_{k-1} + a_2 \cdot u_{k-2} - b_1 \cdot y_{k-1} - b_2 \cdot y_{k-2} \quad (4.13)$$

denklemi elde edilmektedir . İşte bu da 2. dereceden modelimizi temsil etmektedir . Görüldüğü üzere b katsayılarının önündeki “-” işareti durum matrisine de yansımaktadır .

NOT : Bölüm 3’ teki E.K.K. metodu incelendiğinde durum matrisinin normalde u_k terimi ile başladığı görülmektedir . Ancak çalışmada a_0 terimi 0 alındığı için bu terim otomatik olarak ortadan kalkmaktadır .

Recursive E.K.K. algoritmasında kullanılacak olan P başlangıç matrisi içinde ön değer olarak kolay bir matris olan birim matris seçilmektedir .

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

ikinci dereceden model için verilen bu matris modelin derecesine göre artırılabilir. Parametre vektörünün başlangıç değeri içinde rasgele birtakım sayılardan oluşan aşağıdaki gibi bir matris tanımlanmıştır .

$$\theta = \begin{bmatrix} 0 \\ 0 \\ 0.05 \\ -0.05 \end{bmatrix} \quad (4.15)$$

3. dereceden sistem modeli içinse E.K.K. uygulamasında ϕ durum matrisi iterasyon içerisinde aşağıdaki gibi tanımlanmaktadır :

$$\phi_k^T = [u_{k-1} \quad u_{k-2} \quad u_{k-3} \quad -y_{k-1} \quad -y_{k-2} \quad -y_{k-3}] \quad (4.16)$$

Görüldüğü üzere 3. derecede değişen , sadece (k-3). terimlerin eklenmesidir . Bu işlem aynı şekilde P başlangıç matrisine ve θ parametre matrisine de uygulanarak aşağıdaki başlangıç koşulları elde edilmektedir .

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \theta = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0.05 \\ -0.05 \\ 0 \end{bmatrix} \quad (4.17, 4.18)$$

Ek 1' deki kaynak kodunda görülen kestirim altprogramında yukarıdaki değişkenler uygulanarak θ parametre matrisi güncellenmektedir . ve elde edilen sonuçlar gözlem amacıyla sürekli para*.dat isimli dosyalara kaydedilmektedir . Bir sonraki bölümde bu sonuçların sunumu gerçekleştirilecektir .

4.1.4. Farklı Deneylemlerden Elde Edilen Kestirim Değerleri

Deney amacıyla sisteme Şekil 4.2-a ve b' deki gibi sinüzoidal ve kare dalga formunda giriş işaretleri uygulanmıştır . Uygulanan işaretler input.dat dosyasından okunmaktadır ve boyutu istenildiği kadar büyük tutulabilmektedir . Ancak bilgisayar üzerindeki uygulamalarda hafızaya atanabilecek sayı sınırlı olduğundan 4000 eleman değerinin aşmamaktadır . Sistemde örnekleme periyodumuz 25 ms' dir . Sonuç olarak 800 adet örneklenmiş veriden kestirim işlemi gerçekleştirilmiştir .

2. ve 3. dereceden model için sonuçların etkinliğini görmek için parametre değerleri (4.13) denkleminde yerine konularak bir çıkış işareti elde edilmektedir . Bu işaret ile kaydettiğimiz çıkış işareti karşılaştırılarak bir doğruluk testi yapılabilmektedir . Aşağıdaki şekillerde 2. ve 3. derece modeller için parametrelerden elde edilen çıkış işaretleri ile sistemden alınan çıkış işaretlerinin karşılaştırıldığı grafikler ve parametrelerin iterasyonlar boyunca aldığı değeri gösteren grafikler verilmektedir :

Şekil 4.3' te ; 0.4 Hz' lik sinüzoidal giriş işareti (1 nolu sinüzoidal işaret) uygulanan sistemden , 2. dereceden model için alınan çıkış işareti ile parametrelerden elde edilen çıkış işaretinin karşılaştırılması , Şekil 4.4' te ise bu giriş işareti için elde edilen parametrelerin izlediği yol verilmektedir .

Şekil 4.5' te ise yine 1 nolu sinüzoidal işaretin uygulandığı sistemden , 3. dereceden model için alınan çıkış işareti ile parametrelerden elde edilen çıkış işaretinin karşılaştırılması , Şekil 4.6' da ise yine bu model için elde edilen parametre değerlerinin izlediği yol verilmektedir .

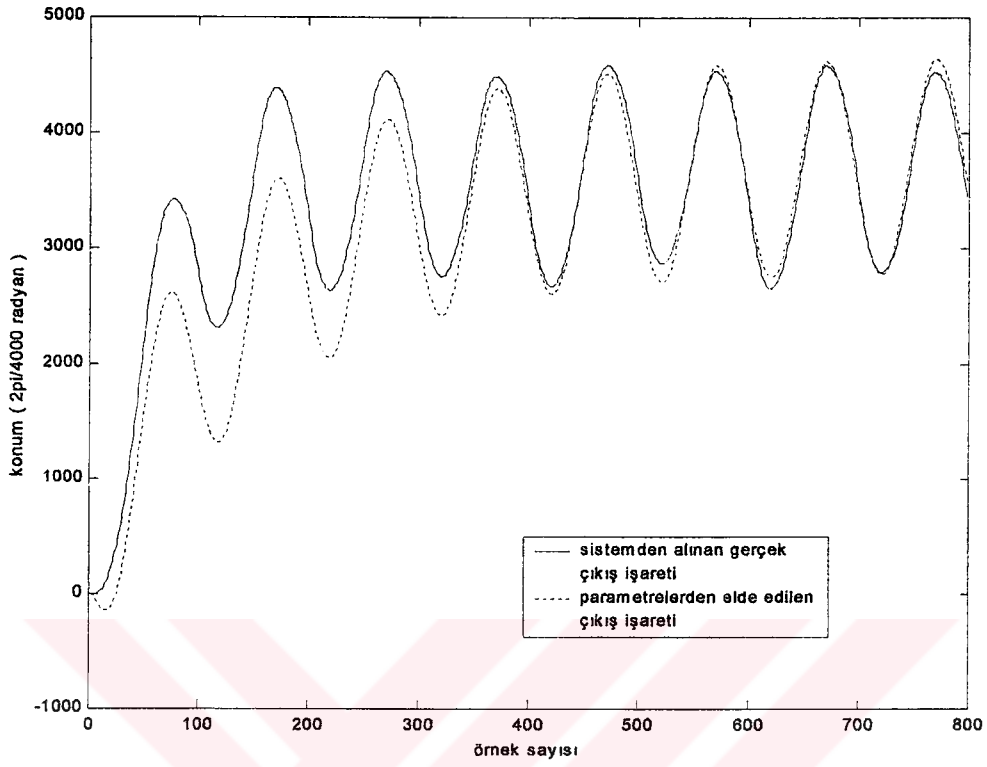
Şekil 4.7' de ise 1 nolu sinüzoidal işaret uygulanan sistemden , 2. derece ve 3. derece modeller için parametrelerden elde edilen çıkış işaretlerinin bir arada karşılaştırma amacıyla verildiği grafik görülmektedir .

Şekil 4.9' de 0.5 Hz' lik sinüzoidal giriş işareti (2 nolu sinüzoidal işaret)

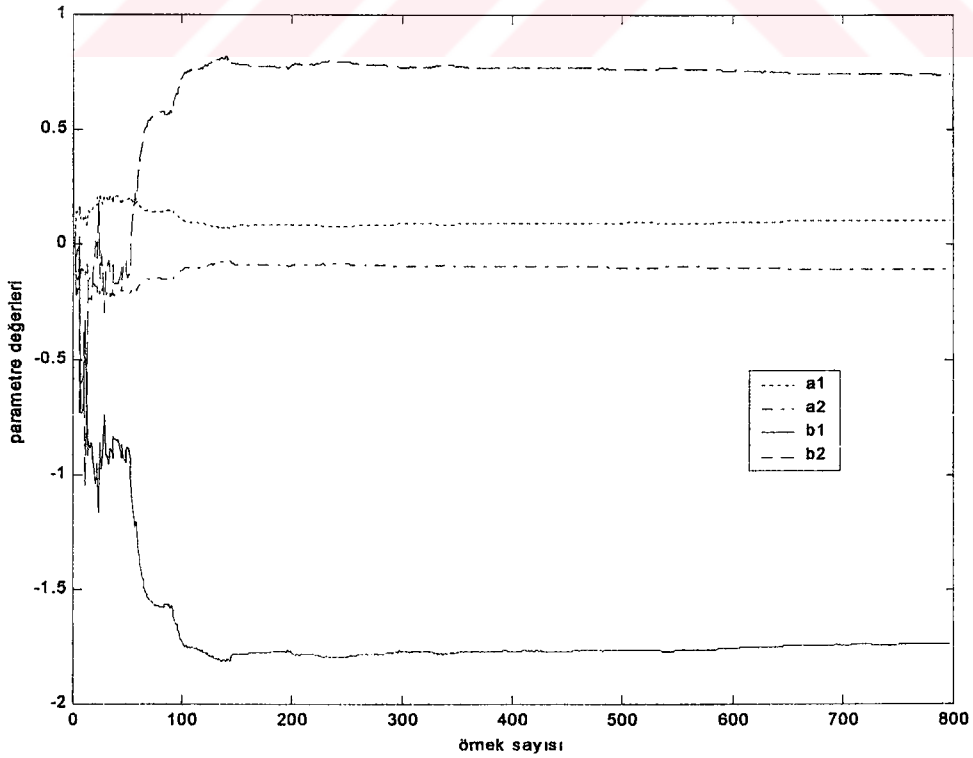
Şekil 4.10' da 1 Hz' lik sinüzoidal giriş işareti (3 nolu sinüzoidal işaret)

Şekil 4.11' da 1.66 Hz' lik sinüzoidal giriş işareti (4 nolu sinüzoidal giriş işareti)

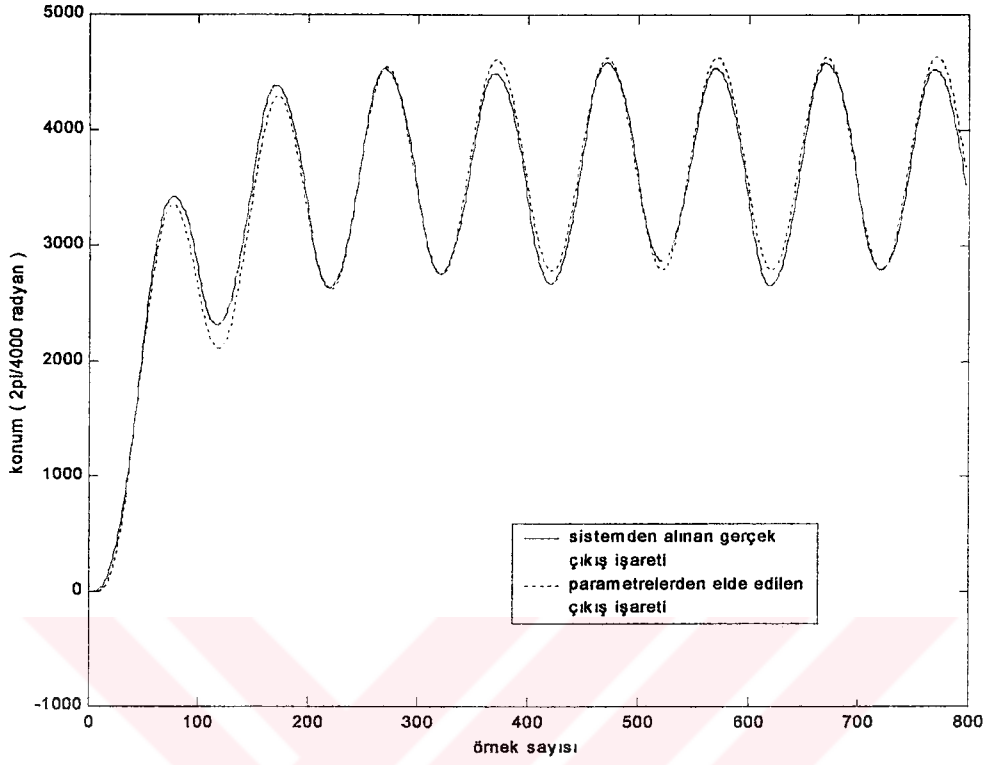
Şekil 5.12 ' de de 5 Hz' lik kare dalga formundaki giriş işareti için elde edilen parametre değerlerinin izlediği yol verilmektedir .



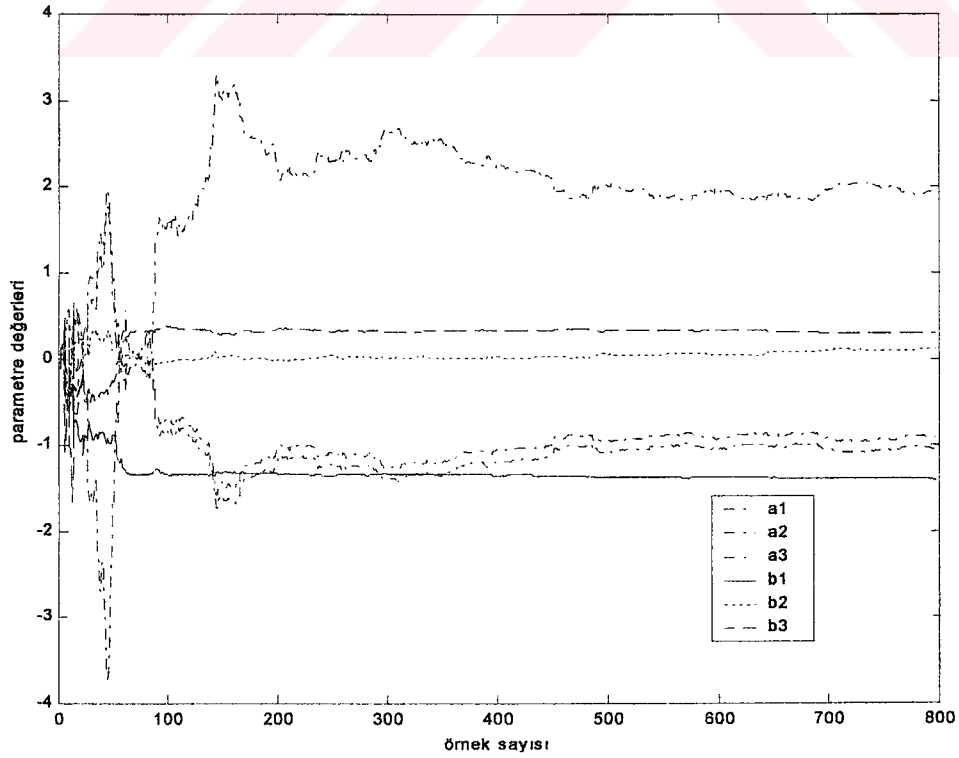
Şekil 4.3. 2. dereceden model için elde edilen çıkış işaretlerinin karşılaştırılması



Şekil 4.4. 2. dereceden model için elde edilen parametreler

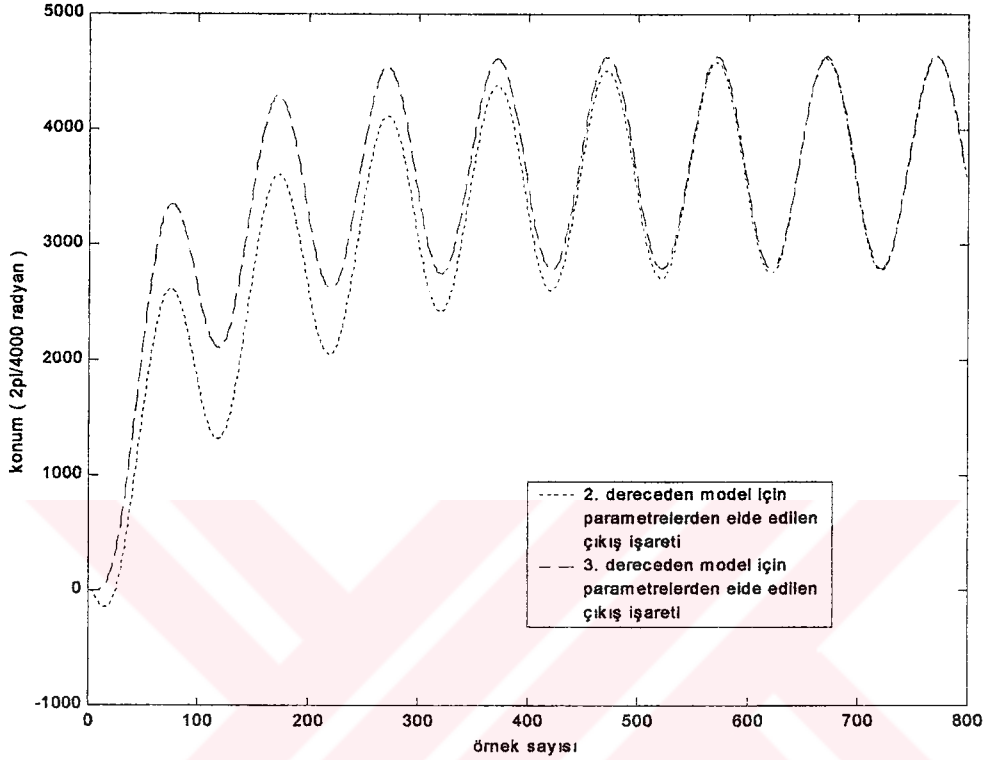


Şekil 4.5. 3. dereceden model için elde edilen çıkış işaretlerinin karşılaştırılması



Şekil 4.6. 3. dereceden model için elde edilen parametreler

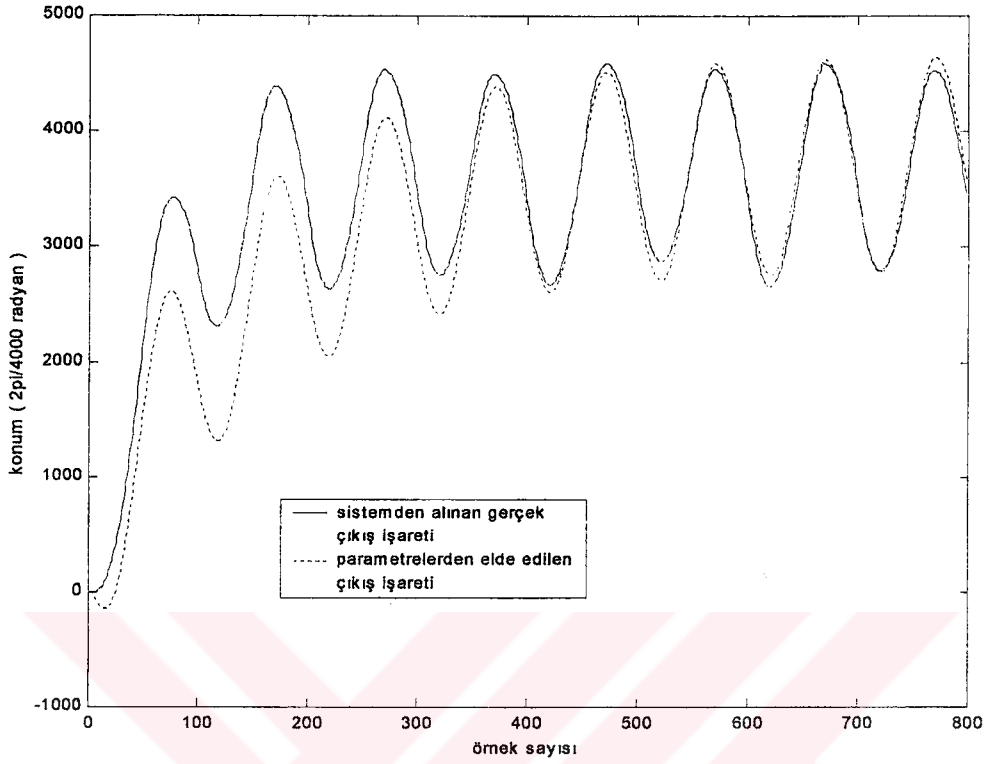
2. ve 3. derece modeller için , kestirim sonucunda elde edilen parametreler kullanılarak bulunan çıkış işaretlerini aynı grafik üzerinde gösterecek olursak ;



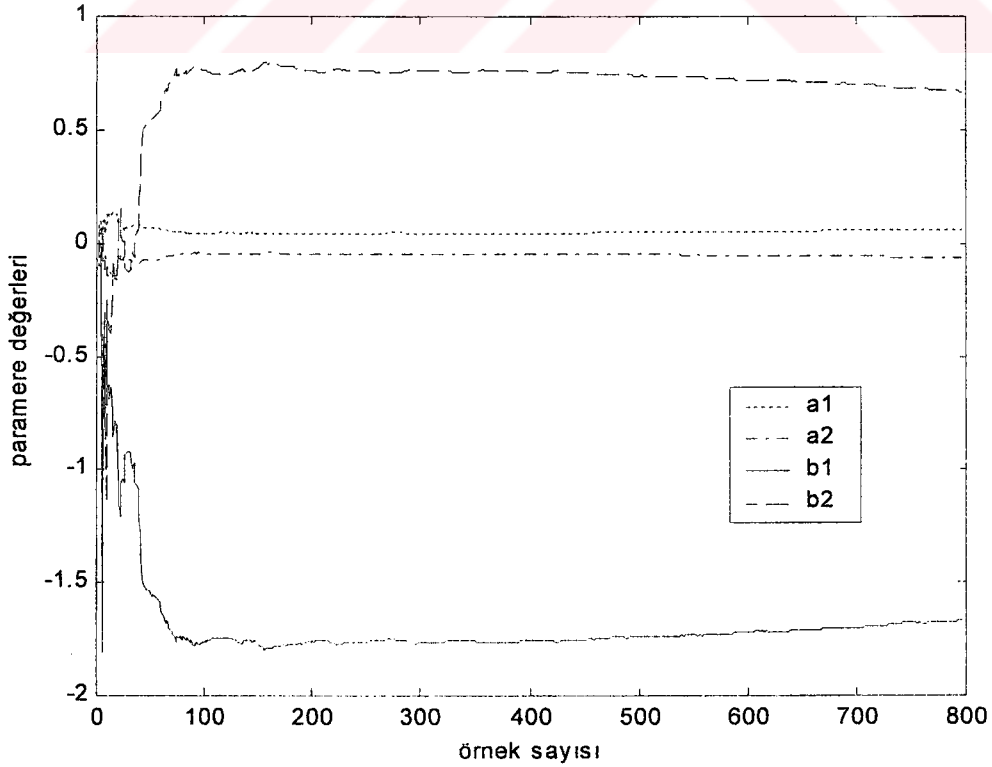
Şekil 4.7. 2. ve 3. dereceden modeller için elde edilen parametrelerden bulunan çıkış işaretlerinin karşılaştırılması

Görüldüğü üzere işaretler parametrelerin belirli bir değere oturduğu bir noktada neredeyse tamamen birbirini tutmaktadır . Bu bize göstermektedir ki sistem aslında 3. derecedendir . Aksi halde 3. derece model için elde edilen işaret yakınsamazdı . Ancak eklenen yeni 3. parametrelerin etkisi çok küçük olduğundan ikiside aynı sonuçları vermektedir . Çıkış işaretlerindeki büyük yaklaşıklıktan dolayı 3. derece model yerine 2. derece model kullanmamızda bir sorun olmayacaktır . Bu noktadan sonraki tüm kestirim işlemlerinde model olarak ; (4.13) denklemiyle verilen 2. dereceden model kullanılacaktır .

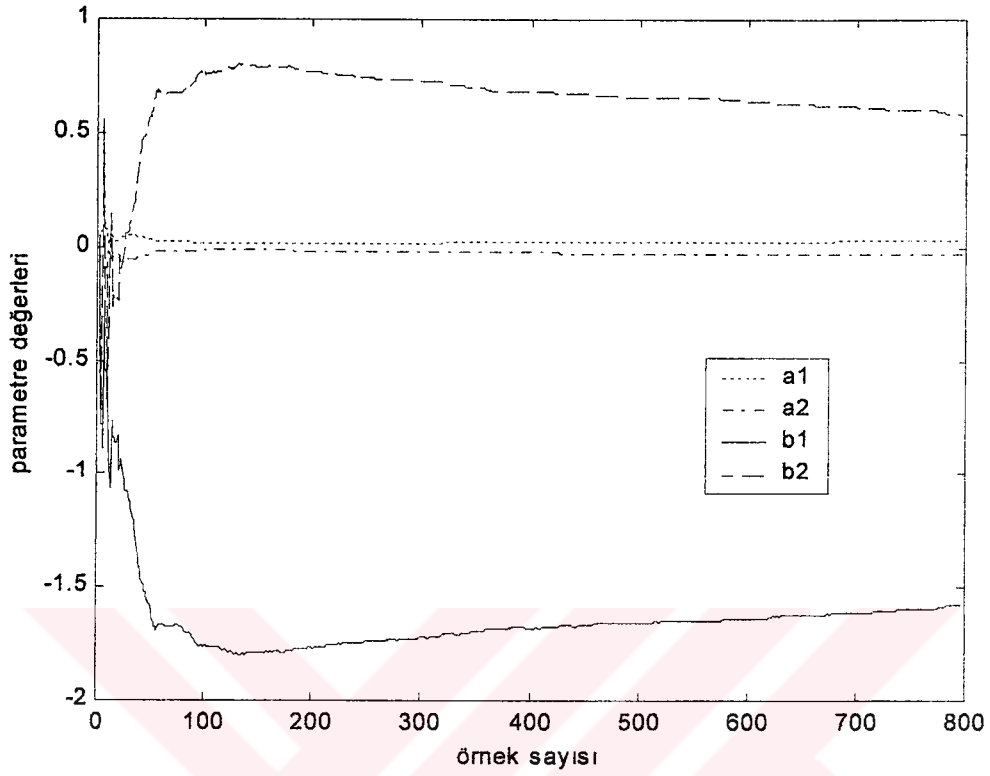
Uygulanan sinüzoidal işaretin frekansını değiştirilerek , elde edilen örnek bir çıkış işareti ve parametre değerlerinin grafiksel gösterimi ise aşağıdaki şekillerde görülmektedir :



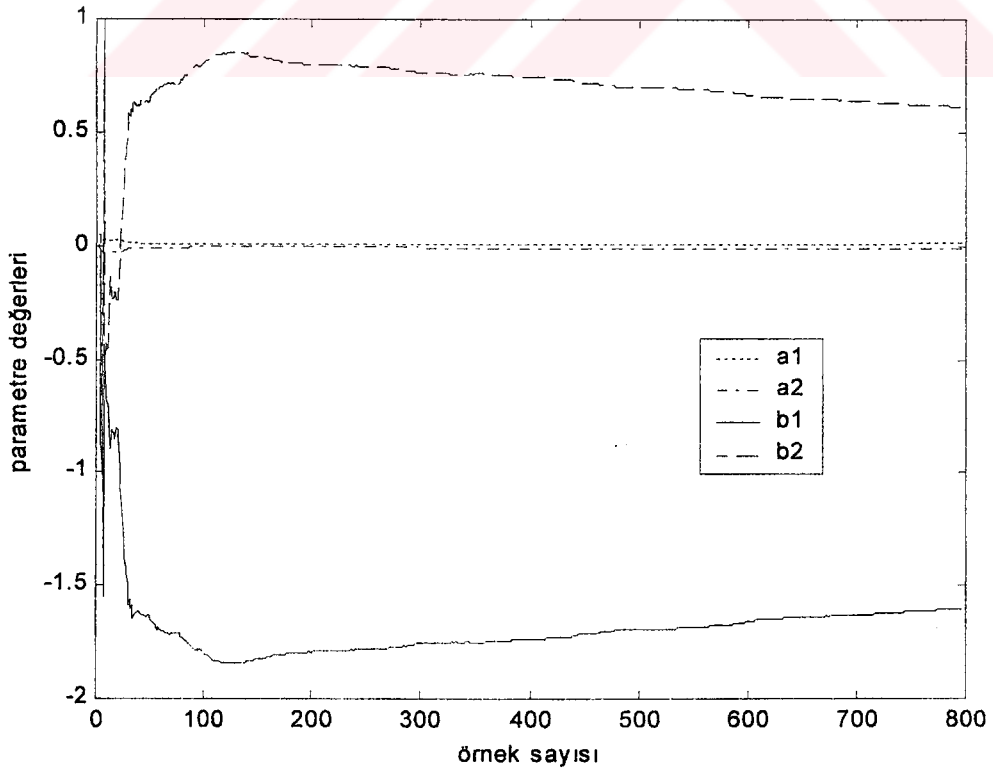
Şekil 4.8 Sisteme uygulanan farklı işaretler için elde edilen örnek çıkış işaretleri



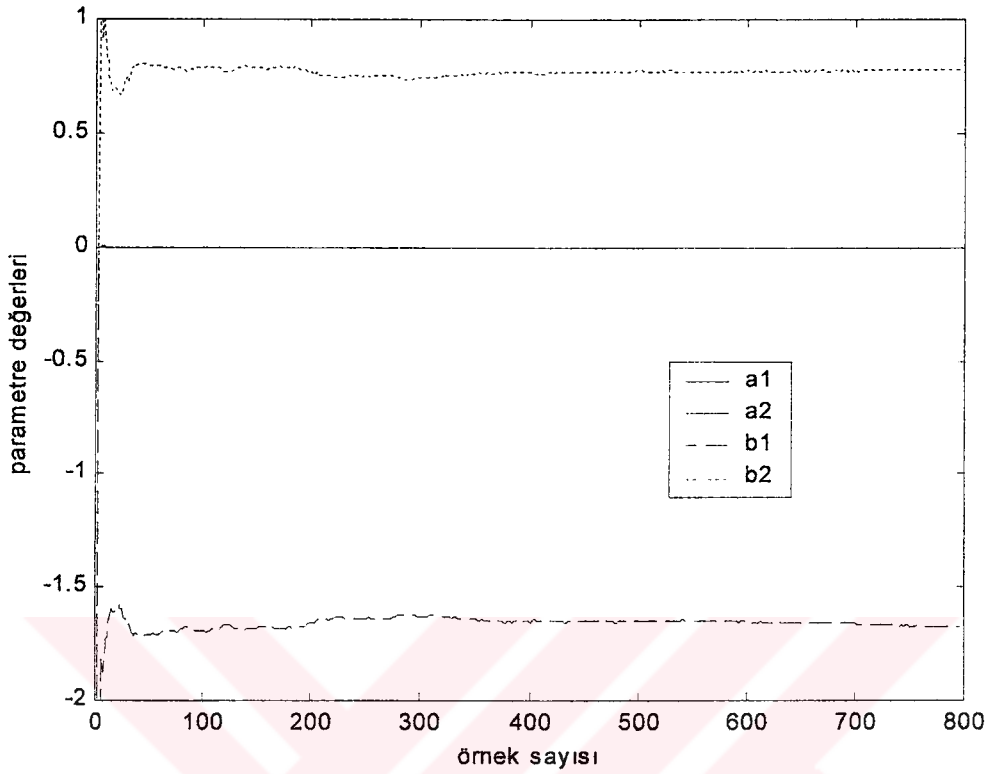
Şekil 4.9. $f=0.5$ Hz'lik sinüzoidal giriş işaretleri için elde edilen parametre değerleri



Şekil 4.10 $f=1$ Hz'lik sinüzoidal giriş işareti için elde edilen parametre değerleri



Şekil 4.11 $f=1.66$ Hz'lik sinüzoidal giriş işareti için elde edilen parametre değerleri



Şekil 4.12 Kare dalga formundaki giriş işareti için elde edilen parametre değerleri

4.1.5. Sonuçların Değerlendirilmesi ve Tartışma

Sistem belirleme amacıyla , 4 farklı sinüzoidal (farklı frekanslarda) ve bir adet de kare dalga olmak üzere toplam beş giriş işareti kullanılmıştır . Tüm bu deneyler sonucunda kayıtları tutulan giriş – çıkış bilgilerinden , kontrol amacıyla birde çevrimdışı E.K.K. kullanılarak kestirim işlemi gerçekleştirilmiştir . Çevrimdışı kestirim için kullanılan giriş işareti bir beyaz gürültüdür . Diğer giriş işaretlerinde olduğu gibi dış ortamda üretilerek sisteme uygulanmaktadır . Bu işaret içinde uygulanan örnekleme periyodu 25 ms' dir . Buna göre elde edilen bütün parametre değerlerini aşağıdaki çizelgede (Çizelge 4.1) topluca görmek mümkündür :

Çizelge 4.1 Kestirim sonuçlarının toplu halde gösterimi

	a_1	a_2	b_1	b_2
Off – line kestirim sonuçları	-0.0032	0.0019	-1.9220	0.9251
1 nolu sinüzoidal giriş (f=0.4 Hz) uygulanan sistem için çevrimiçi kestirim sonuçları	-0.1053	0.1084	-1.7396	0.7413
2 nolu sinüzoidal giriş (f=0.5 Hz) uygulanan sistem için çevrimiçi kestirim sonuçları	-0.0617	0.0643	-1.6661	0.6690
3 nolu sinüzoidal giriş (f=1 Hz) uygulanan sistem için çevrimiçi kestirim sonuçları	-0.0314	0.0335	-1.5725	0.5787
4 nolu sinüzoidal giriş (f=1.66 Hz) uygulanan sistem için çevrimiçi kestirim sonuçları	-0.0106	0.0122	-1.6039	0.6136
Kare dalga formunda giriş (f=5 Hz) uygulanan sistem için çevrimiçi kestirim sonuçları	0.0020	0.0006	-1.6722	0.7761

Tüm kestirim işlemleri 25ms örnekleme ve 800 iterasyon için gerçekleştirilmiştir .

Yukarıdaki çizelge incelendiğinde aşağıdaki sonuçlara ulaşmak mümkündür ;

- i. çevrimdışı kestirim sonuçlarına yani ; doğru kabul edebileceğimiz parametre değerlerine en yakın sonuçların üretildiği adımın , kare dalga formunda giriş işaretinin uygulandığı ve 4 nolu sinüzoidal giriş işaretinin uygulandığı adımlar olduğu görülmektedir . Bunun sebebi işaretlerin frekanslarına bakıldığında kolaylıkla anlaşılmaktadır . Yüksek frekansa sahip giriş işaretinde ; birim ölçüm süresine düşen zaman aralığında sisteme uygulanan enerji miktarı bir önceki adıma göre daha fazla değişmektedir . Böylece sistemin davranışını belirleyen tüm fonksiyonlarının etkin olarak ortaya çıkması sağlanmaktadır . Ancak burada dikkat edilmesi gerekli olan bir konuda : giriş işaretinin frekansının sistemi uyardırmaya yetmeyecek kadar büyük seçilmemesidir . Aksi takdirde Bölüm

3.1.2.2.2' de belirttiğimiz üzere kestirim işlemi ya hiç gerçekleştirilememekte yani sistem tahrik edilememekte ya da hatalı sonuçlara sebep olmaktadır .

- ii. Çizelgeden de görüleceği üzere elde edilen parametre değerleri , gerçek olduğu kabul edilen parametre değerlerine (çevrimdışı kestirim ile bulunan) belirli bir değere kadar yakınsamaktadır . Bunun nedeni sistem üzerine etkiyen gürültünün beyaz gürültü değilde , başka bir gürültü olmasıdır . Bu gürültü ölçüm gürültüsüdür . Motor üzerindeki encoderden alınan konum bilgisi geribeslemesi , Bölüm 3.2.3' te ayrıntılarıyla ele alındığı gibi iki farklı sayıcıda sayılmaktadır . Deney amacıyla kurduğumuz devrede encoderlerden birisinden gelen sinyalin hatalı değer vermesi yüzünden sayıcılardan birisi saymaması gereken durumlarda da saymakta , bu yüzden elde edilen konum bilgisi de hatalı olmaktadır . Büyük genlikli çıkış işaretleri için oluşan bu hata değeri pek önem oluşturmaz iken , küçük genlikli işaretler için durum değişmektedir . Motorun sadece bir dönüş yönünde gerçekleştiği ve oluşumuna dair bir bilgi olmadığı için hata modelide belirlenememektedir .

Kestirim işlemi sonucunda elde edilen parametre değerleri ile model yapısı oluşturulabilmektedir . Böylece sistemin davranışlarını kontrol etmek artık mümkündür. Bunun sonucunda çalışmanın bir sonraki aşaması ; bu sisteme bir kontrolör eklenmesi olacaktır . Parametreleri , bu çalışmada ele alınan yöntemlerle bulunan bir sistemde , (k+1). adım için k. ve daha önceki adımlarda bulunan değerler kullanılarak kontrolör ayarlanabilmektedir . Bu sayede sistemin davranışını istediğimiz forma sokmak mümkün olacaktır .

KAYNAKLAR

ARMAĞAN, O. 1998. *Bilgisayar uyumlu enkoder sinyalleri sayıcı kartı*. Lisans Tezi (Yayınlanmamış), Bursa. 60s.

ASTRÖM, K.J. and T. BOHLIN. 1966. *Numerical identification of linear dynamic systems from normal operating records*. IFAC – Symposium of self adaptive control systems, Teddington, pp.96-110 .

ASTRÖM, K.J. and P. EYKOFF. 1971. *System identification – a survey*. Automatica, Vol 7, pp.123-162.

AXIOM, 1994 edition. *AX5411 General purpose DA&C board user's manual*. Axiom Technology Co. Ltd.

BYTRONIC. *MPIBM84 multifunction I/O card user's manual*. Bytronic control&training systems.

CLARKE, D.W. 1967. *Generalized-least squares estimation of the parameters of a dynamic model*. IFAC Symposium – Identification in Automatic Control Systems, pp.3-17.

DAEDAL, 1995 edition. *Daedal, manual&motorized positioning systems, 1995 catalog*. Parker Motion&Control.

DIGIPLAN, 1996 edition. *BLH/BLX Brushless servo drives user guide*. Digiplan Division, London.

GERTLER, J. and C. BANYASZ. 1974. *A recursive (on-line) maximum likelihood identification method*. IEEE transactions on automatic control, Vol AC-19, No 6, pp.816-820 .

SINHA, N.K. and A. SEN. 1975. *Critical evaluation of on-line identification methods*. Proceedings of the Institution of Electrical Engineers – Control&Science. IEE, Vol 122, No 10, pp.1153-1158 .

SÖDERSTRÖM, T. 1974. *Convergence properties of the generalized least squares identification method*. Automatica, Vol 10, pp.617-626 .

WONG, K.Y. and E. POLAK. 1967. *Identification of linear discrete time systems using on instrumental variables method*. IEEE transactions on automatic control, Vol AC-12, pp.707-719 .

YOUNG, P.C. 1970. *An instrumental variable method for real-time identification of a noisy process*. Automatica, Vol 6, pp.271-287 .

EK'LER



EK-1 (Kontrolör program kodu)

```

// *****
// giriş – çıkış ve sonuç kayıtlarının yazılacağı dosyaların açılması ve hata kontrolü
// *****

if ((fptr1=fopen("datainp.dat","w"))==NULL)
{   printf("datainp.dat dosyası oluşturulamadı !\n");
    return 1;
}
if ((fptr2=fopen("dataoutp.dat","w"))==NULL)
{   printf("dataoutp.dat dosyası oluşturulamadı !\n");
    return 1;
}
if ((fid1=fopen("para1.dat","w"))==NULL)
{   printf("para1.dat dosyası oluşturulamadı !\n");
    return 1;
}
if ((fid2=fopen("para2.dat","w"))==NULL)
{   printf("para2.dat dosyası oluşturulamadı !\n");
    return 1;
}
if ((fid3=fopen("para3.dat","w"))==NULL)
{   printf("para3.dat dosyası oluşturulamadı !\n");
    return 1;
}
if ((fid4=fopen("para4.dat","w"))==NULL)
{   printf("para4.dat dosyası oluşturulamadı !\n");
    return 1;
}

// *****
// Kestirim işleminde kullanılmak üzere üretilen giriş işaretinin harici giriş
// dosyasından
// okunması ve hafızaya alınması .
// *****

fidx=fopen("input.dat","r");

for(i=0;i<4001;i++)
{   fscanf(fidx,"%f",(yenigirdi+i));
}

// *****
// DAC/ADC kartının initialize edilmesi ve hata kontrolü
// Burada dio[0] kartın base adres'ini , dio[1] IRQ no' sunu , dio[2] DMA
// channel' ını göstermektedir .
// ax5411 ise yükleyici altprogramdır .
// *****

```

EK-1 (Kontrolör program kodu)

```

fun=INIT ;
dio[0]=IOPORT;
dio[1]=3;
dio[2]=3;
flag=ax5411(fun,dio,ary1,ary2);

if(flag!=0)
{   printf("DRIVER INITIALIZATION FAILED ! , flag=%d",flag);
    exit(1);
}

// *****
// Motora harekete başlamadan önce tamamen durması için analog 0V' un karşılığı
// olan 2040 değeri gönderilmektedir .
// *****

fun=DAC_ONE;
dio[0]=0;
dio[1]=2040;
flag=ax5411(fun,dio,ary1,ary2);

// *****
// Sonsuz işlem döngüsüne giriliyor . Bu döngü içerisinde her bir hareket toplamı
// için gerekli olan çıkış şartlarının testi yapılmaktadır .
// *****

while(1)
{
    p=0;
    printf("\n\n\n Kontrol değerini giriniz : "); // döngüden çıkış şartının testi =
    scanf("%d",&step); // 0 değeri için döngüden çıkılır.
    if(step==0) // diğer değerler için içeride kalınır.
    {   fun=DAC_ONE;
        dio[0]=0;
        dio[1]=2040;
        flag=ax5411(fun,dio,ary1,ary2);
        break;
    }
}

// *****
// Counter' ların kurulması ( Burada counter0 ve counter2 kurulmaktadır .)
// *****

cnt0_start=50000;
cnt0_high=cnt0_start/256;
cnt0_low=cnt0_start-(cnt0_high*256);
outp(modecnt_id,modecn0);

```


EK-1 (Kontrolör program kodu)

```

outp(count0_id,cnt0_low);
outp(count0_id,cnt0_high);

cnt2_start=50000;
cnt2_high=cnt2_start/256;
cnt2_low=cnt2_start-(cnt2_high*256);
outp(modecnt_id,modecn2);
outp(count2_id,cnt2_low);
outp(count2_id,cnt2_high);

old_data=0;

// *****
// İkinci sonsuz işlem döngüsüne giriliyor . Bu döngü içerisinde ise tüm kontrol ve
// kestirim işlemleri gerçekleştirilmektedir .
// *****

while(s<4000) // Burada 4000 iterasyon sayısını
{ // temsil etmektedir .
    s++;
    outp(67,0); // İşlemin başlangıç zamanının
    tcount1=inp(64); // okunması
    tcount2=inp(64);
    tcount_first=(tcount2*256)+tcount1;

    outp(modecnt_id,0); // Counter0 ve counter2' den o
    ccount1=inp(count0_id); // andaki sayım değerleri okunması
    ccount2=inp(count0_id);
    ccount=(ccount2*256)+ccount1;
    cdiff=ccount;
    outp(modecnt_id,128);
    ccount1=inp(count2_id);
    ccount2=inp(count2_id);
    ccount=(ccount2*256)+ccount1;

    new_data=cdiff-ccount; // konum bilgisinin elde edilmesi
    out[p]=new_data;

// *****
// Aşağıdaki kontrolör kısmında gerekli yeni giriş değeri hesaplanmaktadır .
// kullandığımız kontrol metodu PD kontrolüdür .
// *****

y=yenigirdi[s];
if(y>4096) y=4096;
if(y<0) y=0;

```

EK-1 (Kontrolör program kodu)

```

old_data=new_data;
in[p+1]=y;

fun=DAC_ONE; // yeni giriş işaretinin motora
dio[0]=0; // gönderilmesi
dio[1]=y;
flag=ax5411(fun,dio.ary1,ary2);

in_old2=in_old1; // kestirim için gerekli olan
in_old1=in_new; // bilgilerin hazırlanması
in_new=y;
out_old2=out_old1;
out_old1=out_new;
out_new=new_data;

// *****
// Kestirim adlı alt program ile sistem belirleme amacıyla kullanılacak metot
// çağrılmaktadır . Burada kullanılan metot E.K.K. olmakla birlikte
// istenildiği takdirde aynı giriş parametreleriyle farklı bir metotta
// uygulanabilmektedir .
// *****

kestirim(in_old1,in_old2,out_old1,out_old2,out_new);

p++;
tdiff=200;
while(tdiff<5500) // Örnekleme süresinin kontrolü
{
    outp(67,0);
    tcount1=inp(64);
    tcount2=inp(64);
    tcount_last=(tcount2*256)+tcount1;
    tdiff=tcount_first-tcount_last;
}
if(kbhitt()!=0) break; // Döngü sonu kontrolü ve çıkış
}

// *****
// Döngü sonucunda her bir adım için elde edilen ve hafızada tutulan giriş – çıkış
// değerleri ve kestirim sonuçlarının hepsi ilgili dosyalara yazılmaktadır .
// *****

for(m=1;m<p;m++)
{
    fprintf(fp1,"%d \n",in[m]);
    fprintf(fp2,"%f \n",out[m]);

    fprintf(fid1,"%f \n",PARAM[0][m]);
    fprintf(fid2,"%f \n",PARAM[1][m]);
}

```

EK-1 (Kontrolör program kodu)

```

        fprintf(fid3,"%f\n",PARA[2][m]);
        fprintf(fid4,"%f\n",PARA[3][m]);
    }

} // İkinci sonsuz döngünün sonu

fclose(fptr1);
fclose(fptr2);
fclose(fid1); // Programın başında açılan
fclose(fid2); // dosyaların kapatılması
fclose(fid3);
fclose(fid4);

return 0;

} // Main döngüsünün sonu

// %%%%%%%%%%% Kestirim Alt Programı %%%%%%%%%%%
// *****
// Kestirim alt programı aşağıda görülmektedir .
// Program içerisinde kullanılan metot ; Recursive En Küçük Kareler metodu' dur .
// Ancak istenildiği takdirde , altprogramın giriş parametrelerine bağlı kalmak
// koşuluyla başka metotlarda kullanılabilir .
// *****

float kestirim(float in1,float in2,float in3,float in4,float in5)
{
    float a,b,c,d,gecici_top;
    float x,Q[4];
    float P_PART_1_1[4],P_PART_1_2[4],P_PART_1[4][4];
    float P_PART_2_1,P_PART_2,P_PART_3[4][4];
    float TETA_PART_1[4],TETA_PART_2_1,TETA_PART_2;
    float TETA_PART_3[4];
    int i,j,k;

    c=(in1*5.0/2048.0)-5.0;
    a=(in2*5.0/2048.0)-5.0;
    d=(in3*3.1416/1000);
    d*=-1;
    b=(in4*3.1416/1000);
    b*=-1;

    Q[0]=c;
    Q[1]=a;
    Q[2]=d;
    Q[3]=b;

```

EK-1 (Kontrolör program kodu)

```

for(i=0;i<4;i++)
{
    gecici_top=0;
    for(j=0;j<4;j++)
        gecici_top+=*((P+i)+j)*((Q+j));
    *(P_PART_1_1+i)=gecici_top;
}

for(i=0;i<4;i++)
{
    gecici_top=0;
    for(j=0;j<4;j++)
        gecici_top+=*(Q+j)*((P+j)+i);
    *(P_PART_1_2+i)=gecici_top;
}

for(i=0;i<4;i++)
    for(k=0;k<4;k++)
        *((P_PART_1_1+i)+k)=*(P_PART_1_1+i)*((P_PART_1_2+k));

gecici_top=0;

for(i=0;i<4;i++)
    gecici_top+=*(P_PART_1_2+i)*((Q+i));

P_PART_2_1=gecici_top;
P_PART_2=1+P_PART_2_1;

for(i=0;i<4;i++)
    for(j=0;j<4;j++)
        *((P_PART_3+i)+j)=*((P_PART_1+i)+j)/P_PART_2;

for(i=0;i<4;i++)
    for(j=0;j<4;j++)
        *((P+i)+j)=*((P+i)+j)-*((P_PART_3+i)+j);

for(i=0;i<4;i++)
{
    gecici_top=0;
    for(j=0;j<4;j++)
        gecici_top+=*((P+i)+j)*((Q+j));
    *(TETA_PART_1+i)=gecici_top;
}

gecici_top=0;

for(j=0;j<4;j++)
    gecici_top+=*(Q+j)*((TETA+j));

TETA_PART_2_1=gecici_top;

```

EK-1 (Kontrolör program kodu)

```
x=(in5*3.1416/1000);  
  
TETA_PART_2=x-TETA_PART_2_1;  
  
for(i=0;i<4;i++)  
    *(TETA_PART_3+i)=*(TETA_PART_1+i)*TETA_PART_2;  
  
for(i=0;i<4;i++)  
    *(TETA+i)=*(TETA+i)+(*(TETA_PART_3+i));  
  
for(i=0;i<4;i++)  
    (*(PARA+i)+p)=*(TETA+i);  
  
return 0;  
}
```



EK-2 (Ax5411 DAC/ADC Kartı bilgileri)

Name	Pin	Pin	Name
AI0	1	2	AI8
AI1	3	4	AI9
AI2	5	6	AI10
AI3	7	8	AI11
AI4	9	10	AI12
AI5	11	12	AI13
AI6	13	14	AI14
AI7	15	16	AI15
AGND	17	18	AGND
+12V	19	20	-12V
	21	22	
DA0	23	24	DA1
AGND	25	26	AGND
DO0	27	28	DI0
DO1	29	30	DI1
DO2	31	32	DI2
DO3	33	34	DI3
DGND	35	36	DGND
DO4	37	38	DI4
DO5	39	40	DI5
DO6	41	42	DI6
DO7	43	44	DI7
+5VP	45	46	+12VP
EXTRG	47	48	
DGND	49	50	DGND

a

AX5411 (CNA) Pin Assignment

Name	I/O	Function
AIx	Input	Channel x Analog Input
DAX	Output	D/A Channel x Voltage Output
DIx	Input	Channel x Digital Input
DOx	Output	Channel x Digital Output
+ 12V	Source	Analog Power
- 12V	Source	Analog Power
AGND	Ground	Analog Ground
DGND	Ground	Digital Ground
+5VP	Source	PC +5V
+12VP	Source	PC +12V
-12VP	Source	PC -12V
EXTRG	Input	External trigger

Name	Pin	Pin	Name
DI0	1	2	DI1
DI2	3	4	DI3
DI4	5	6	DI5
DI6	7	8	DI7
DI8	9	10	DI9
DI10	11	12	DI11
DI12	13	14	DI13
DI14	15	16	DI15
DGND	17	18	DGND
+5VP	19	20	+12VP

CN1 (Digital Input) Pin Assignment

Name	Pin	Pin	Name
DO0	1	2	DO1
DO2	3	4	DO3
DO4	5	6	DO5
DO6	7	8	DO7
DO8	9	10	DO9
DO10	11	12	DO11
DO12	13	14	DO13
DO14	15	16	DO15
DGND	17	18	DGND
+5VP	19	20	+12VP

b

CN2 (Digital Output) Pin Assignment

Name	I/O	Function
DIx	Input	Channel x Digital Input
DOx	Output	Channel x Digital Output
DGND	Ground	Digital Ground
+5VP	Source	PC +5V
+12VP	Source	PC +12V

EK-2 (Ax5411 DAC/ADC Kartı bilgileri)**(2) Function INIT : Initialization****Purpose:**

1. Setup base address according to dio[0].
2. Setup interrupt level according to dio[1].
3. Setup DMA level according to dio[2].
4. Check the presence of ax5411 hardware.
5. Disable all external, interrupt, sft trg function.
6. Set timer 1KHz.

This function, needed to be executed only once, must be executed before any other functions.

Parameter:

```
.Entry
fun      - INIT (or 0).
dio[0]   - I/O port.
dio[1]   - IRQ level from 2 to 7.
dio[2]   - DMA channel (1 or 3).
dio[3]   - Not used.
dio[4]   - Not used.
dio[5]   - Not used.
*ary1    - Not used.
*ary2    - Not used.
```

.Return

```
flag = 0 - No error.
2      - Function No. out of range.
3      - I/O port setting error (100H - 3FFH).
4      - IRQ level setting out of range.
5      - DMA channel setting out of range.
6      - Hardware fail.
```

Example:

```
#include      "ax5411.h"
int          fun, flag, IOPORT=0x300;
unsigned int dio[7];
int far     *ary1;
int far     *ary2;

fun = INIT;      /* Select function */
dio[0] = 0x300;  /* Set io port   */
dio[1] = 3;      /* IRQ level     */
dio[2] = 3;      /* DMA level     */
flag = ax5411(fun, dio, ary1, ary2);
if (flag != 0) printf(" Initialization failed");
```

EK-2 (Ax5411 DAC/ADC Kartı bilgileri)

(17) Function DAC_ONE : 1 D/A conversion single channel

Purpose:

This function performs single channel 1 D/A conversion D/A output full scale equalizes +5 or +10VDC according to JP2 setting.

DA output = out-date/4096 * DA-full-scale

Parameter:**.Entry**

fun - DAC_ONE (or 15).
dio[0] - Channel NO. 0 or 1 .
dio[1] - Out data 0 - 4095 .
dio[2] - Not used.
dio[3] - Not used.
dio[4] - Not used.
dio[5] - Not used.
*ary1 - Not used.
*ary2 - Not used.

d

.Return

flag = 0 - No error.
1 - Driver not initialized.
2 - Function No. out of range.
21 - D/A data setting out of range(0 to 4095).
20 - D/A channel NO. setting out of range(0 to 1).

Example:

```

fun = DAC_ONE; /* DAC single channel */
dio[0] = 0; /* channel 0 */
dio[1] = 1024; /* out-date */
flag=ax5411(fun,dio,ary1,ary2);
if (flag != 0) printf (" DAC error !,flag=%d\n",flag);

```

(18) Function DAC_TWO : 1 D/A conversion, both of channels

Purpose:

This function performs D/A conversion of tow channels simuteniously. D/A output full scale equalizes +5 or +10VDC according to JP2 setting.

D/A output = out-date/4095 * DA-full-scale

Parameter:**.Entry**

fun - DAC_TWO (or 16).
dio[0] - Channel 0 out data 0 - 4095 .
dio[1] - Channel 1 out data 0 - 4095 .
dio[2] - Not used.
dio[3] - Not used.
dio[4] - Not used.
dio[5] - Not used.
*ary1 - Not used.
*ary2 - Not used.

e

.Return

flag = 0 - No error.
1 - Driver not initialized.
2 - Function No. out of range.
21 - Output data setting out of range(0 to 4095).

Example:

```

fun = DAC_TWO; /* DAC two channel */
dio[0] = 1024; /* Channel 0 +1.25 V DC */
dio[1] = 1024; /* Channel 1 +1.25 V DC */
flag=ax5411(fun,dio,ary1,ary2);
if (flag != 0) printf (" DAC error !,flag=%d\n",flag);

```


EK-3 (MPIBM84 Sayıcı Kartı bilgileri)

The counter modes are selected by writing to the control word register (ba address +3), according to the following tables.

D7	SC1	SELECT COUNTER
D6	SC0	
D5	RL1	READ/LOAD
D4	RL0	
D3	M2	MODE
D2	M1	SELECT
D1	M0	
D0	BDC	BINARY

SC1	SC0	FUNCTION
0	0	SELECT COUNTER 0
0	1	SELECT COUNTER 1
1	0	SELECT COUNTER 2
1	1	ILLEGAL

RL1	RL0	FUNCTION
0	0	LATCH COUNTERS
1	0	READ/LOAD MOST SIGNIFICANT BYTE ONLY
0	1	READ/LOAD LEAST SIGNIFICANT BYTE ONLY
1	1	READ/LOAD LSB FIRST, THEN MSB

M2	M1	M0	FUNCTION
0	0	0	MODE 0
0	0	1	MODE 1
0	1	0	MODE 2
0	1	1	MODE 3
1	0	0	MODE 4
1	0	1	MODE 5

BCD	FUNCTION
0	BINARY COUNTER 16-BITS
1	BINARY CODED DECIMAL (BCD) COUNTER (4 DECADES)

EK-3 (MPIBM84 Sayıcı Kartı bilgileri)**The 40 way IDC Header Pin Allocations**

FUNCTION	PIN No.		FUNCTION
CLK1	1	2	INTERNAL CLOCK
CLK2	3	4	CLK0
OUT1	5	6	GATE1
GATE2	7	8	OUT0
OUT2	9	10	GATE0
DISABLE INTERNAL CLK	11	12	0V
5V	13	14	5V
PORT C0	15	16	C7
C1	17	18	C6
C2	19	20	C5
C3	21	22	C4
B0	23	24	B7
B1	25	26	B6
B2	27	28	B5
B3	29	30	B4
A0	31	32	A7
A1	33	34	A6
A2	35	36	A5
A3	37	38	A4
OV	39	40	OV

EK-4 (Encoder sinyalleri değerlendirme kartı bilgileri)

Tablo 4.3. ' te A TTL ve B TTL işaretleri için oluşturulan durum – geçiş tablosu görülmektedir . Bu tablo oluşturulurken devre gerçekleştirilmesi esnasında D latch kullanılacağı varsayılmış , buna göre D_B ve D_A sütunları da eklenmiştir. İleri ve geri fonksiyonlarını elde etmek için sistemi çözümlememiz gerekmektedir . Bu amaçla Carnaugh diyagramlarından faydalanılarak D_A , D_B , ileri ve geri fonksiyonları bulunmaktadır . Şöyle ki ;

 D_B için:

		Q_B				
		00	01	11	10	
00		0	0	0	0	}
01		0	0	0	0	
11	}	1	1	1	1	
10		1	1	1	1	
		Q_A				

 D_A için:

		Q_B				
		00	01	11	10	
00		0	0	0	0	}
01		1	1	1	1	
11	}	1	1	1	1	
10		0	0	0	0	
		Q_A				

Bu diyagramlardan gerekli eliminasyonlar sonrası aşağıdaki sonuçlara ulaşmak mümkündür :

$$D_B = B$$

$$D_A = A$$

İleri için :

		Q_B				
		00	01	11	10	
00					1	}
01		1				
11	}		1			
10				1		
		Q_A				

Geri için :

		Q_B				
		00	01	11	10	
00			1			}
01				1		
11	}				1	
10		1				
		Q_A				

EK-4 (Encoder sinyalleri değerlendirme kartı bilgileri)

Bu diyagramlarla ilgili eşitlikleri yazacak olursak :

$$\begin{aligned} \text{İLERİ} &= A'.B'.Q_B.Q_A' + A'.B'.Q_A'.Q_B' + A.B.Q_A.Q_B' + A'.B.Q_A.Q_B \\ &\Rightarrow \text{İLERİ} = B.Q_A.(A.Q_B' + A'.Q_B) + B'.Q_A'.(A.Q_B' + A'.Q_B) \\ &\Rightarrow \text{İLERİ} = \underbrace{(A.Q_B' + A'.Q_B)}_{A \oplus Q_B^*} \cdot \underbrace{(B.Q_A + B'.Q_A')}_{B \otimes Q_A^*}^* \\ &\Rightarrow \text{İLERİ} = (A \oplus Q_B).(B \otimes Q_A) \end{aligned}$$

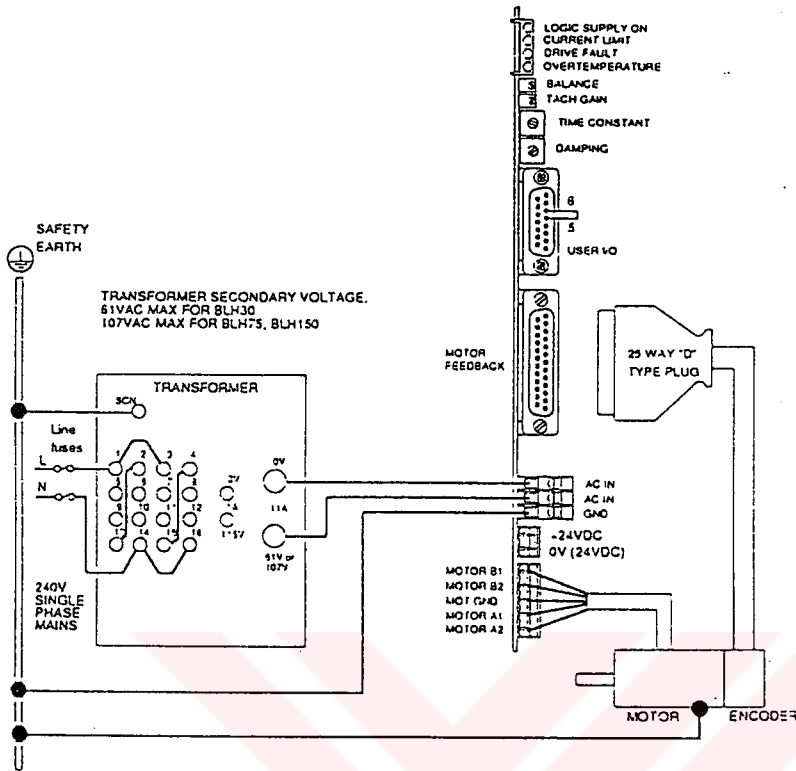
$$\begin{aligned} \text{GERİ} &= A'.B'.Q_A.Q_B' + A.B'.Q_A.Q_B + A.B.Q_A'.Q_B' + A'.B.Q_A'.Q_B' \\ &\Rightarrow \text{GERİ} = A'.Q_B'.(B'.Q_A + B.Q_A') + A.Q_B'.(B'.Q_A + B.Q_A') \\ &\Rightarrow \text{GERİ} = \underbrace{(B'.Q_A + B.Q_A')}_{B \oplus Q_A} \cdot \underbrace{(A.Q_B' + A'.Q_B)}_{A \otimes Q_B}^* \\ &\Rightarrow \text{GERİ} = (B \oplus Q_A).(A \otimes Q_B) \end{aligned}$$

sonuçlarına ulaşılmaktadır . Bu fonksiyonlara göre , lojik elemanlarla gerçekleştirilen devre Şekil 4.4. ' te verilmiştir .

* $X.Y' + X'.Y \rightarrow \text{XOR} \rightarrow \oplus$
 $X.Y + X'.Y' \rightarrow \text{XNOT} \rightarrow \otimes$

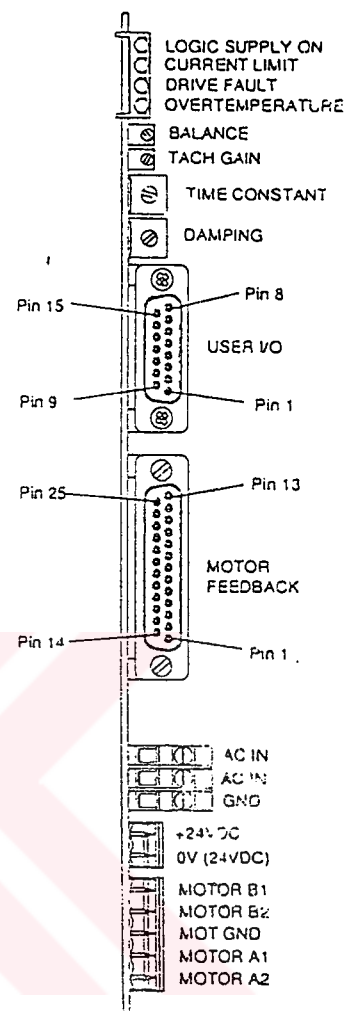
* $(X.Y' + X'.Y)' = (X'+Y).(X+Y)'$ dür . Bu ifadeye $X=A$ ve $Y=Q_B$ dersek ;
 $A \oplus Q_B = (A.Q_B' + A'.Q_B)$ ise $(A \oplus Q_B)' = (A.Q_B' + A'.Q_B)' = (A'+Q_B).(A+Q_B)' = (A.Q_B + A'.Q_B') = A \otimes Q_B$ olduğu görülmektedir . Bu da bize Şekil 4.4. teki devrenin sağ kısmındaki NOT kapılarını açıklamaktadır

EK-5 (Brushless servo motor sürücüsü bilgileri)



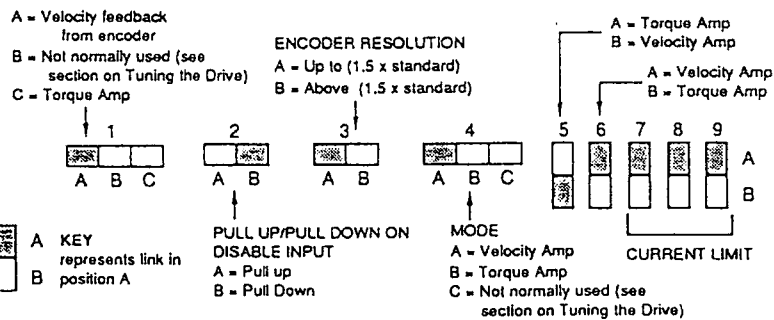
Pre-installation Test Configuration (240V AC Mains Supply)

a



Connectors and Indicators

b



Jumper Link Setting Schematic

Factory Default Link Positions	1	2	3	4	5	6	7	8	9
BLH	A	B	A	A	B	A	A	A	A
BLHX	C	A	A	B	A	B	A	A	A

Factory Default Jumper-Link Positions

c

EK-5 (Brushless servo motor sürücüsü bilgileri)

Motor Feedback Connector Pin Functions

ML1620 Motor Connector Pin Functions at the Drive End

Pin Number	Function	Signal Type	Lead Colour
1	Screen	R	Drain Wire
2	0V	E	Bk
3	5V	D	Rd
4			Not connected
5			Not connected
6	$\overline{\text{MOT}}$	F	Rd
7	MOT	F	Gn
8	Z-	C	Rd
9	Z+	C	W
10	B-	C	Bk
11	B+	C	Or
12	A-	C	Bk
13	A+	C	Brn
14 to 17			Not connected
18	$\overline{\text{A3}}$	C	Bk
19	A3	C	W
20	$\overline{\text{A2}}$	C	Bk
21	A2	C	Gn
22	$\overline{\text{A1}}$	C	Bk
23	A1	C	Bl
24	$\overline{\text{A0}}$	C	Bk
25	A0	C	Y

d

Motor Feedback Connector Pin Functions

User I/O Connector Pin Functions

Pin	Signal Name	Function	Signal Type
1	V2	Analogue velocity input	L
2	V1	Analogue velocity input	L
3	-15v	Reference voltage	O
4	GND	Ground	Q
5	$\overline{\text{RST}}$	Reset/Disable	P
6	+15V	Reference voltage	O
7	Not used		
8	Not used		
9	$\overline{\text{FT}}$	Fault	N
10	AOP *	A output from incremental encoder	M
11	$\overline{\text{AOP}}$	A output	M
12	BOP *	B output from incremental encoder	M
13	$\overline{\text{BOP}}$	B output	M
14	ZOP **	Z output from incremental encoder	M
15	$\overline{\text{ZOP}}$	Z output	M

e

Key to Signal Types

- A +24v supply to drive
- B 0V of 24V supply to drive
- C Differential encoder input
- D Encoder supply voltage
- E Encoder supply 0V
- F Motor overtemperature input
- G AC input from mains transformer
- H Ground for mains transformer screen
- I Motor supply
- J Motor Ground
- L Analogue velocity demand signal
- M Differential output from incremental encoder
- N Open collector output
- O Reference voltage out (10mA max.)
- P Active low control input
- Q Logic supply ground

EK-6 (Brushless servo motorun<ML3450A-10> bilgileri)**BL Series Brushless Servo System**

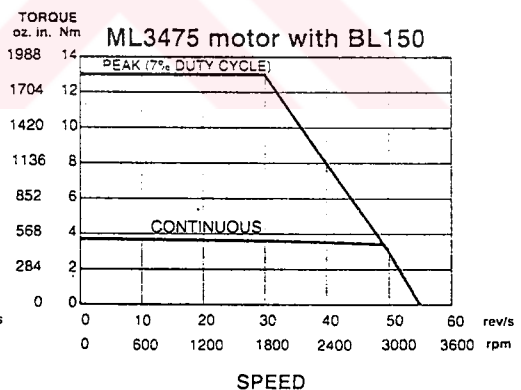
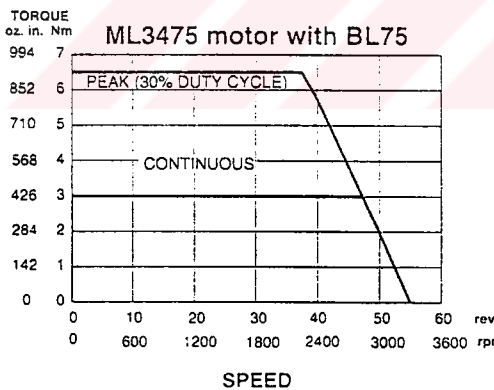
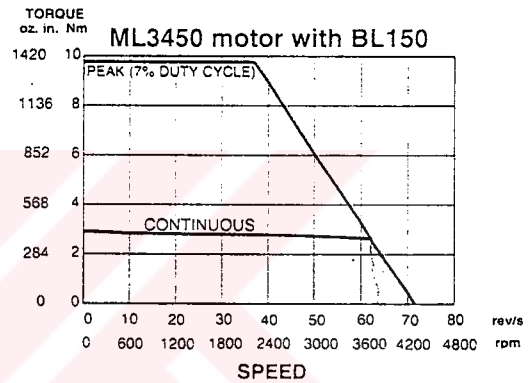
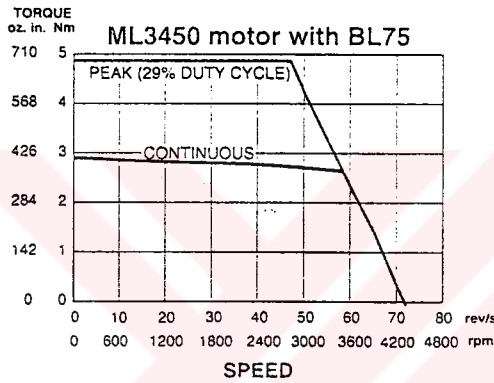
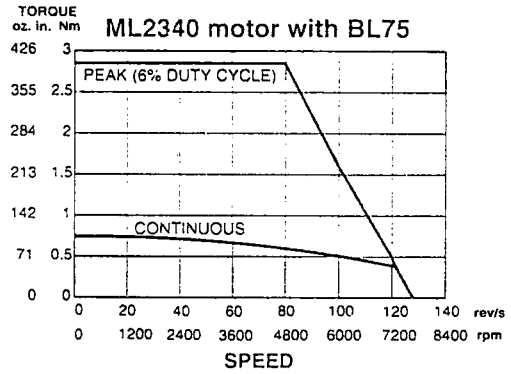
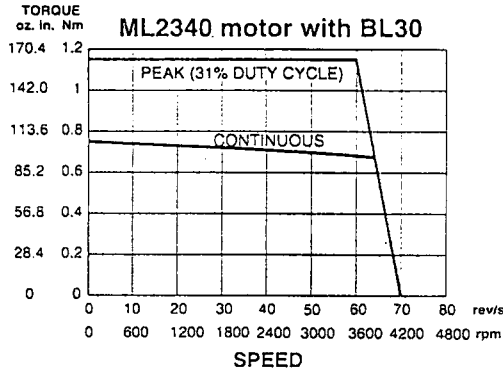
BL Drive Specifications

Parameter	Value			
Performance:				
Bandwidth	Torque amplifier >2.5 KHz			
Formfactor	1.01			
Gains	Velocity mode 50,500,5000. Torque mode 6.6.			
Deadband	Zero			
Drift	10uV/°C			
Speed/torque	Curves located on page D26			
Encoder	Maximum frequency pre-quadrature 100KHz (from motor encoder)			
Current control	Re-circulating PWM, at 20KHz			
		BL30	BL75	BL150
Continuous current	A	3	7.5	15
Peak current	A	6	15	30
AC input voltage	VAC	61	107	107
DC bus voltage	VDC	85	150	150
BL drive weight	lbs (kg)	1.76 (0.8)	3.74 (1.7)	6.16 (2.8)
Motor compatibility				
ML2340		X	X	
ML3450			X	X
ML3475			X	X
Amplifier:				
Configuration	Factory default is velocity mode, jumper configurable for torque mode			
Peak current	Clamped by jumper configuration			
AC Power Input:				
Voltage	61-107 VAC, requires transformer			
Frequency	50-60Hz			
Inputs:				
Connector	15 pin D-type socket (user I/O)			
Analog command	±10V differential analog input. Input impedance 44KΩ			
Reset/disable	Jumper configurable for normally closed contact to +15V, or normally open contact to GND			
Outputs:				
Connector	15 pin D-type socket (user I/O)			
Drive fault	Active low. NPN open collector. Emitter coupled to GND. Maximum off-state voltage 40V. Maximum current sink 80mA. On state voltage of 0.2V at 80mA.			
Encoder ±15V	See encoder output specifications ±15V auxiliary power supply at 10mA max			
Encoder Outputs:				
Type	Buffered from motor encoder, available for use with servo controller			
Connector	15 pin D-type socket (user I/O)			
Electrical	Pre-quadrature A, B with Z channel. Differential TTL line driver. 100KHz maximum frequency.			
Diagnostics:				
LED	Over-temperature, drive fault and current limit			
Output	See drive fault specifications			
Environmental:				
Drive	Storage	-40°F to 185°F (-40°C to 85°C)		
	Operation	32°F to 122°F (0°C to 100°C)		
	Humidity	0-95% non-condensing		
Motor	Storage	-40°F to 212°F (-40° to 100°C)		
	Operation	32°F to 212°F (0°C to 100°C)		
	Humidity	0-95% non-condensing		

EK-6 (Brushless servo motorun<ML3450A-10> bilgileri)

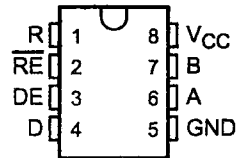
BL Series Brushless Servo System

Torque/Speed Curves



Motor Data		ML2340	ML3450	ML3475
Rotor inertia	oz-in ² (kg-cm ²)	1.53 (0.28)	8.75 (1.6)	13.12 (2.4)
Electrical time constant	mS	1.50	3.00	3.00
Mechanical time constant	mS	0.92	0.61	0.76
Thermal time constant	min	20	35	40
Torque constant	Nm/A (pk) oz-in/A (pk)	0.22 92	0.38 158	0.42 175
Encoder resolution	lines/rev counts/rev	500 2000	1000 4000	1000 4000
Weight	lbs kg	4.35 1.95	10.6 4.85	13.6 6.2
Minimum operating temperature	°F (°C)	32° (0°)	32° (0°)	32° (0°)
Maximum operating temperature	°F (°C)	212° (100°)	212° (100°)	212° (100°)

- Bidirectional Transceivers
- Meet or Exceed the Requirements of ANSI Standards TIA/EIA-422-B and TIA/EIA-485-A and ITU Recommendations V.11 and X.27
- Designed for Multipoint Transmission on Long Bus Lines in Noisy Environments
- 3-State Driver and Receiver Outputs
- Individual Driver and Receiver Enables
- Wide Positive and Negative Input/Output Bus Voltage Ranges
- Driver Output Capability . . . ± 60 mA Max
- Thermal Shutdown Protection
- Driver Positive and Negative Current Limiting
- Receiver Input Impedance . . . 12 k Ω Min
- Receiver Input Sensitivity . . . ± 200 mV
- Receiver Input Hysteresis . . . 50 mV Typ
- Operate From Single 5-V Supply

D OR P PACKAGE
(TOP VIEW)

description

The SN65176B and SN75176B differential bus transceivers are monolithic integrated circuits designed for bidirectional data communication on multipoint bus transmission lines. They are designed for balanced transmission lines and meet ANSI Standards TIA/EIA-422-B and TIA/EIA-485-A and ITU Recommendations V.11 and X.27.

The SN65176B and SN75176B combine a 3-state differential line driver and a differential input line receiver, both of which operate from a single 5-V power supply. The driver and receiver have active-high and active-low enables, respectively, that can be connected together externally to function as a direction control. The driver differential outputs and the receiver differential inputs are connected internally to form differential input/output (I/O) bus ports that are designed to offer minimum loading to the bus when the driver is disabled or $V_{CC} = 0$. These ports feature wide positive and negative common-mode voltage ranges, making the device suitable for party-line applications.

The driver is designed for up to 60 mA of sink or source current. The driver features positive and negative current limiting and thermal shutdown for protection from line-fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 150°C. The receiver features a minimum input impedance of 12 k Ω , an input sensitivity of ± 200 mV, and a typical input hysteresis of 50 mV.

The SN65176B and SN75176B can be used in transmission-line applications employing the SN75172 and SN75174 quadruple differential line drivers and SN75173 and SN75175 quadruple differential line receivers.

The SN65176B is characterized for operation from -40°C to 105°C and the SN75176B is characterized for operation from 0°C to 70°C .



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

 **TEXAS
INSTRUMENTS**

POST OFFICE BOX 855303 • DALLAS, TEXAS 75265

Copyright © 1999, Texas Instruments Incorporated

SN65176B, SN75176B

DIFFERENTIAL BUS TRANSCEIVERS

SLLS101B – JULY 1985 – REVISED JUNE 1999

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IK}	Input clamp voltage	$I_I = -18$ mA				-1.5	V
V_O	Output voltage	$I_O = 0$		0		6	V
$ V_{OD1} $	Differential output voltage	$I_O = 0$		1.5	3.6	6	V
$ V_{OD2} $	Differential output voltage	$R_L = 100 \Omega$,	See Figure 1	$1/2 V_{OD1}$ or 2‡			V
		$R_L = 54 \Omega$,	See Figure 1	1.5	2.5	5	V
V_{OD3}	Differential output voltage	See Note 4		1.5		5	V
$\Delta V_{OD} $	Change in magnitude of differential output voltage§	$R_L = 54 \Omega$ or 100Ω , See Figure 1				± 0.2	V
V_{OC}	Common-mode output voltage					+3 -1	V
$\Delta V_{OC} $	Change in magnitude of common-mode output voltage§					± 0.2	V
I_O	Output current	Output disabled, See Note 5	$V_O = 12$ V			1	mA
			$V_O = -7$ V			-0.8	
I_{IH}	High-level input current	$V_I = 2.4$ V				20	μ A
I_{IL}	Low-level input current	$V_I = 0.4$ V				-400	μ A
I_{OS}	Short-circuit output current	$V_O = -7$ V				-250	mA
		$V_O = 0$				150	
		$V_O = V_{CC}$				250	
		$V_O = 12$ V				250	
I_{CC}	Supply current (total package)	No load	Outputs enabled		42	70	mA
			Outputs disabled		26	35	

† The power-off measurement in ANSI Standard TIA/EIA-422-B applies to disabled outputs only and is not applied to combined inputs and outputs.

‡ All typical values are at $V_{CC} = 5$ V and $T_A = 25^\circ\text{C}$.

§ $\Delta|V_{OD}|$ and $\Delta|V_{OC}|$ are the changes in magnitude of V_{OD} and V_{OC} , respectively, that occur when the input is changed from a high level to a low level.

¶ The minimum V_{OD2} with a $100\text{-}\Omega$ load is either $1/2 V_{OD1}$ or 2 V, whichever is greater.

NOTES: 4. See ANSI Standard TIA/EIA-485-A, Figure 3.5, Test Termination Measurement 2.

5. This applies for both power on and off; refer to ANSI Standard TIA/EIA-485-A for exact conditions. The TIA/EIA-422-B limit does not apply for a combined driver and receiver terminal.

switching characteristics, $V_{CC} = 5$ V, $R_L = 110$ k Ω , $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
$t_d(OD)$	Differential-output delay time	$R_L = 54 \Omega$	See Figure 3		15	22	ns
$t_t(OD)$	Differential-output transition time				20	30	ns
t_{pZH}	Output enable time to high level	See Figure 4			85	120	ns
t_{pZL}	Output enable time to low level	See Figure 5			40	60	ns
t_{pHZ}	Output disable time from high level	See Figure 4			150	250	ns
t_{pLZ}	Output disable time from low level	See Figure 5			20	30	ns



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

TEŞEKKÜR

Yüksek Lisans tez çalışmalarım esnasında her konuda yaptığı katkılarından dolayı danışmanım sayın Yrd. Doç Dr. Murat TÜRE' ye ve tezimin yazımı sırasında birçok teknik konuda bana yardımcı olan araştırma görevlisi arkadaşım Gökhan YENİKAYA' ya teşekkürü bir borç bilirim . Aynı zamanda çalışmamın deneysel kısmını gerçekleştirdiğim ; Makine Mühendisliği Bölümü , Makine Teorisi ve Dinamiği Laboratuvarı'ndan sorumlu öğretim üyesi ve öğretim elemanlarına da teşekkürlerimi sunuyorum .



ÖZGEÇMİŞ

Ekrem Düven , 15.05.1977 tarihinde Çanakkale ilinin Biga ilçesinde doğdu. İlk ve orta öğrenimini aynı yerde tamamladı. Lisans öğrenimini Uludağ Üniversitesi Mühendislik Mimarlık Fakültesi Elektronik Mühendisliği Bölümü'nde 1998 yılında bitirdi. Aynı yıl Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilimdalı'nda Yüksek Lisans öğrenimine ve aynı üniversitenin Mühendislik Mimarlık Fakültesi Elektronik Mühendisliği Bölümü'nde araştırma görevlisi olarak çalışmaya başladı. Halen bu bölümde araştırma görevlisi olarak çalışmaya devam etmektedir .

