

75183

UYARLAMALI MODEL KONTROL

YÜKSEK LİSANS TEZİ
Mak.Müh. Erdoğan GÜLYAZ

75183

Tezin Enstitüye Verildiği Tarih : 30 Eylül 1998

Tezin Savunulduğu Tarih : 3 Kasım 1998

Tez Danışmanı : Doç.Dr. N.Aydın HIZAL

Diğer Jüri Üyeleri : Prof.Dr. Ahmet KUZUCU

Doç.Dr. Levent GÜVENÇ

İSTANBUL TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EYLÜL 1998

ÖNSÖZ

Bu çalışmada bana her konuda yardımcı olan sayın hocam Doç.Dr. N.Aydın HIZAL'a, verdikleri destekle eğitimimi bitirmemi sağlayan değerli iş arkadaşlarıma ve sevgili aileme teşekkürü borç bilirim.

Erdoğan GÜLYAZ

İSTANBUL, 1998



İÇİNDEKİLER

ÖNSÖZ	ii
SEMBOL LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	ix
BÖLÜM 1.GİRİŞ	1
BÖLÜM 2.UYARLAMALI SİSTEMLER	4
2.1 Açık Çevrimli ve Kapalı Çevrimli Uyarlamalı Sistemler	5
2.2 Uyarlamalı Kontrol	7
BÖLÜM 3. UYARLAMALI LİNEER BİRLEŞTİRİCİLER	8
3.1 Giriş Sinyali ve Ağırlık Vektörleri	8
3.2 İstenen Cevap ve Hata	10
3.3 Gradyen ve Minimum MSE	12
3.4 Dik İnme Yöntemi(<i>Steepest Descent</i>) ile Gradyen Arama	13
3.5 En Küçük Ortalama Kareler (LMS) Algoritması	13
3.6 Ayarsızlık	14
BÖLÜM 4. UYARLAMALI MODELLEME	17
4.1 Transversal Filtre	17
4.2 Uyarlamalı Modelleme	19
BÖLÜM 5. UYARLAMALI MODEL KONTROL	28
5.1 İleri Zamanlı Hesaplama	29
5.2 Direkt Programlama	32

5.3 Uyarlamalı Model Kontrol	33
BÖLÜM 6. SONUÇ	36
KAYNAKLAR	37
EK A Matlab Programları	38
EK B Uyarlamalı Modelleme ve Kontrol Örnekleri	42
ÖZGEÇMİŞ	46



SEMBOL LİSTESİ

ξ	: Ortalama kare hata
Λ	: Özdeğer matrisinin köşegeni
∇_k	: Gradyen
ε_k	: Hata sinyali
λ	: Özdeğer
μ	: Yakınsama faktörü
d_k	: Sistemin cevabı ve istenen cevap
Q	: Özdeğer matrisi
R	: Giriş vektörü korelasyon matrisi
r_k	: Referans giriş
s	: Laplace ifadesi
T_s	: Örnekleme periyodu
w_n	: Doğal frekans
$w_0, w_1, \dots, w_L$	: Ağırlık vektör elemanları
W_k	: Ağırlık vektörü
W^*	: Optimum ağırlık vektörü
$x_0, x_1, \dots, x_L$	: Giriş vektör elemanları
X_k	: Model ve sistemin giriş vektörü
y_k	: Modelin cevabı

ŞEKİL LİSTESİ

Şekil 2.1	Uyarlamalı kontrol sistemi	4
Şekil 2.2	Açık ve kapalı çevrimli uyarlamalı kontrol genel şeması	6
Şekil 2.3	Bir uyarlamalı kontrol sisteminin blok diyagramı	7
Şekil 3.1	Uyarlamalı lineer bir birleştiricinin genel formu	8
Şekil 3.2	Tek girişli uyarlamalı lineer birleştirici	9
Şekil 3.3	İkinci dereceden kuadratik bir performans yüzeyi	11
Şekil 3.4	İki ağırlık değeri için dik inme yöntemi gösterimi	13
Şekil 4.1	Tek girişli ve tek çıkışlı dinamik bir sistemin modellenmesi	17
Şekil 4.2	Ayrık zamanlı transversal filtre ile modelleme	18
Şekil 4.3	Uyarlamalı modelleme	19
Şekil 4.4	Köklerin geometrik yeri	20
Şekil 4.5	Ayrık zamanlılaştırılmış sistemin impuls cevabı	20
Şekil 4.6	Filtre edilmiş rasgele giriş sinyal üreticisi	21
Şekil 4.7	Sistem ve uyarlamalı modelin, $\mu=0.01$ için çeşitli basamak giriş değerlerine cevapları	22
Şekil 4.8	Sistem ve uyarlamalı modelin, $\mu=0.001$ için çeşitli basamak giriş değerlerine cevapları	23
Şekil 4.9	Sistem ve uyarlamalı modelin, $\mu=0.08$ için rasgele giriş değerlerine cevapları	24
Şekil 4.10	Sistem ve uyarlamalı modelin, ilk 200 adımda $\mu=0.05$ için rasgele giriş değerlerine cevapları	24
Şekil 4.11	Farklı μ değerleri için ϵ hataların zamana bağlı değişimi	25
Şekil 4.12	Adaptasyon sürecinde hatanın karesi ϵ^2	25
Şekil 4.13	Farklı uzunlukta seçilen ağırlık vektörleri için ortalama kare hata ϵ^2 'nin değişimi	26
Şekil 4.14	İlerleyen adımlarda ağırlıkların değişimi	27
Şekil 5.1	Uyarlamalı model kontrol sisteminin blok diyagramı	28
Şekil 5.2	'İleri zamanlı hesaplama' kutusunun detayı	31

Şekil 5.3	Sistemin, direkt programlama yöntemiyle elde edilen blok diyagramı	32
Şekil 5.4	Sistemin ve uyarlamalı modelin, $k=4$ için rasgele giriş değerlerine cevapları (500. Adımda kontrol başlıyor)	34
Şekil 5.5	Farklı μ değerleri için ortalama kare hata ε^2 'nin değişimi	35



ÖZET

Klasik kapalı çevrimli kontrol sistemleri, farklı çevre şartlarında sistem parametrelerinde değişiklik yaratmadığı sürece tasarımı kolay ve düşük maliyetinden dolayı tercih edilir. Ancak değişken çevre şartlarının sistem parametrelerini büyük ölçüde değiştirdiği ve giriş değerlerinin önceden kestirilemediği uygulamalarda klasik kontrol tamamen yetersiz kalmaktadır. Özellikle robotik, radar, iletişim ve biyomedikal cihazlar gibi ileri teknoloji uygulamalarında son yıllarda tercih edilen uyarlamalı kontrol sistemleri üzerinde sürekli çalışmalar yapılmakta ve son zamanlarda yapay sinir ağlarıyla entegrasyonu konusunda da ciddi ilerlemeler kaydedilmektedir. Canlı organizmaların değişken çevre şartlarında dengelerini korumaları prensibini kendisine çıkış noktası olan bu sistem yapı itibarıyla da bir öğrenen sistemdir.

Bu tez çalışmasında uyarlamalı sistemlerin genel tanıtımı yapılmış, bu alan içinde önemli bir yere sahip olan uyarlamalı modelleme ve buna bağlı olarak uyarlamalı model kontrol detaylı olarak incelenmiştir. İkinci dereceden bir sistem, MATLAB programlama dili kullanılarak önce modellemesi yapılmış ve rasgele verilen giriş değerlerine karşılık sistemin ve modelin performansları değerlendirilmiştir. Modellemede, gerçek sistem çıkışı ile model çıkışı arasındaki farkı minimum yapabilmek için LMS (en küçük kareler) algoritması kullanılmıştır. Tasarım sırasında bazı parametreler değiştirilerek modelin üzerindeki etkisi incelenmiştir. Kontrol için, sistem direkt programlama yöntemiyle yeniden yapılandırılmış ve ileri zamanlı hesaplama yöntemiyle sistemin ve modelin kontrolü sağlanmıştır. Sistemin uyarlamalı modellemesi ve kontrolüyle ilgili yapılan bütün simülasyonlar grafiklerle verilmiştir.

SUMMARY

ADAPTIVE MODEL CONTROL

Adaptive plant modeling or plant identification is an important function for all adaptive control systems. In the practical world, the plant to be controlled may be unknown and possibly time variable. Firstly, it is assumed that the plant is continuous, stable or stabilized, linear and time invariant. A discrete-time adaptive modeling system samples the plant input and output and automatically adjusts its internal parameters to produce a sampled output which is a close match to the samples of the plant output when the samples of the plant input are used as the input to the adaptive model. When the plant and its model produce similar output signals, the adaptive impulse response is a good representation of the plant impulse response. In fact, the discrete-time adaptive model is a model of the samples of the impulse response of the plant. The basic idea is illustrated in Fig.1, where all signals and systems are considered to be sampled. [1,2,5]

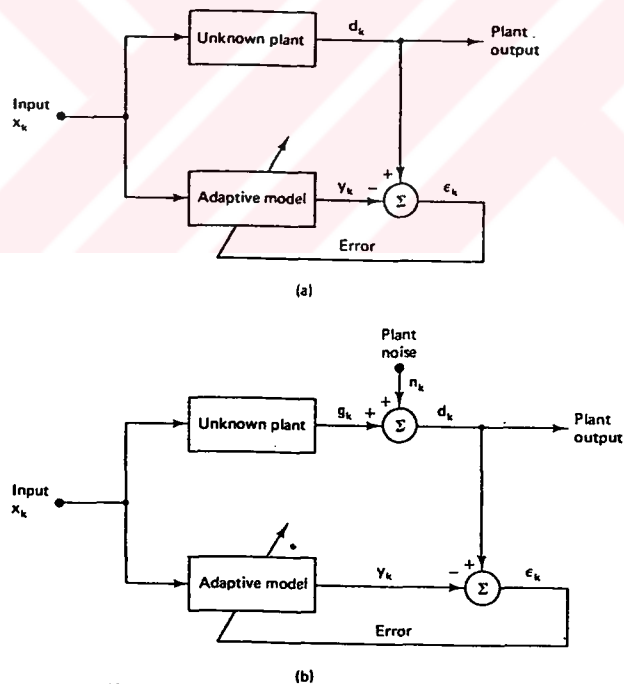


Fig. 1 Adaptive Modeling [3]

An adaptive filter (transversal filter) used in modeling, that is, imitating the behaviour of physical dynamic systems which may be regarded as unknown. The form of the adaptive filter to be considered as a tapped delay line, variable weights whose input signals are the signals at the delay-line taps, a summer to add the weighted signals, and an adaptation process that automatically seeks an optimal impulse response by adjusting the weights. Fig. 2 illustrates the adaptive filter which is casual and has a finite impulse response (FIR) [5].

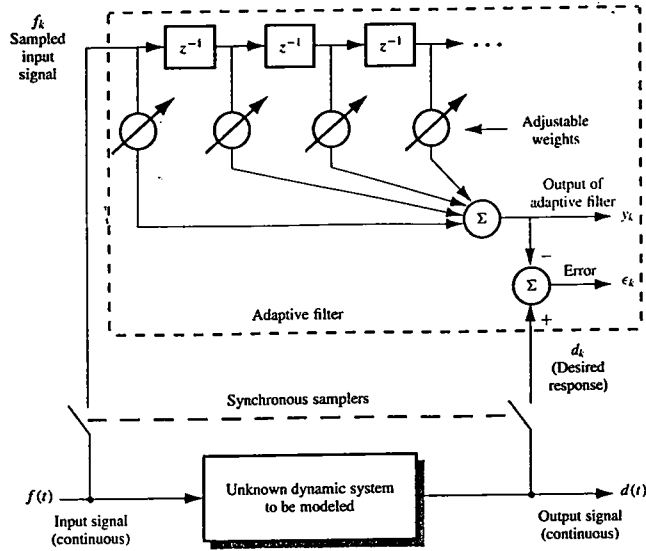


Fig.2 Modeling a plant by transversal filter [5]

There is an input signal vector with elements $x_0, x_1, \dots, x_L$, a corresponding set of adjustable weights, $w_0, w_1, \dots, w_L$, a summing unit, and a single output signal, y . Input output relation is formulated as follows.

$$y_k = \sum_{i=0}^L w_{ik} x_{k-i} \quad (1.1)$$

$$\mathbf{W}_k = [w_{0k} \ w_{1k} \ \dots \ w_{Lk}]^T \quad (1.2)$$

$$y_k = \mathbf{X}_k^T \mathbf{W}_k = \mathbf{W}_k^T \mathbf{X}_k \quad (1.3)$$

Adaptive algorithm is,

$$\epsilon_k = d_k - y_k = d_k - \mathbf{X}_k^T \mathbf{W} = d_k - \mathbf{W}^T \mathbf{X}_k \quad (1.4)$$

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu \epsilon_k \mathbf{X}_k \quad (1.5)$$

The output signal, y_k , is simply subtracted from the desired signal, d_k , to produce the error signal, ϵ_k . The objective of the modeling process is to minimize the error signal ϵ_k so that for the same input signals the model and the plant generate the same outputs. In order to minimize the error signal, LMS (least-mean-squares) algorithm is used which uses a special estimate of the gradient that is valid for adaptive filter. Formula (1.5) is obtained from this algorithm by using steepest descent method which is used for gradient searching.

With a stationary input and a stationary plant, the LMS algorithm is known to be convergent when the convergence factor μ is chosen in the range,

$$1/\lambda_{\max} > 1/\text{tr}\mathbf{R} > \mu > 0 \quad (1.6)$$

where $\mathbf{R}$ is the input autocorrelation matrix, $\lambda_{\max}$ is its largest eigenvalue.

System to be modeled is chosen as a second order simple system and its transfer function is given below.

$$\frac{1}{s^2 + s + 1} \quad (1.7)$$

The sampling period is determined according to settling time of the impulse response of discretized plant. T_s equals to 0.1 second which is small enough. As the settling time is about 10 seconds the number of weights is calculated as $10/0.1=100$. Therefore the weight vector with 100 elements used as an adaptive filter in this study.

By using MATLAB programming language which is very common in control applications, the above system is modeled and for each 200 iteration cycle three different step input values is entered both the plant and the model and their outputs are illustrated in Fig. 3.

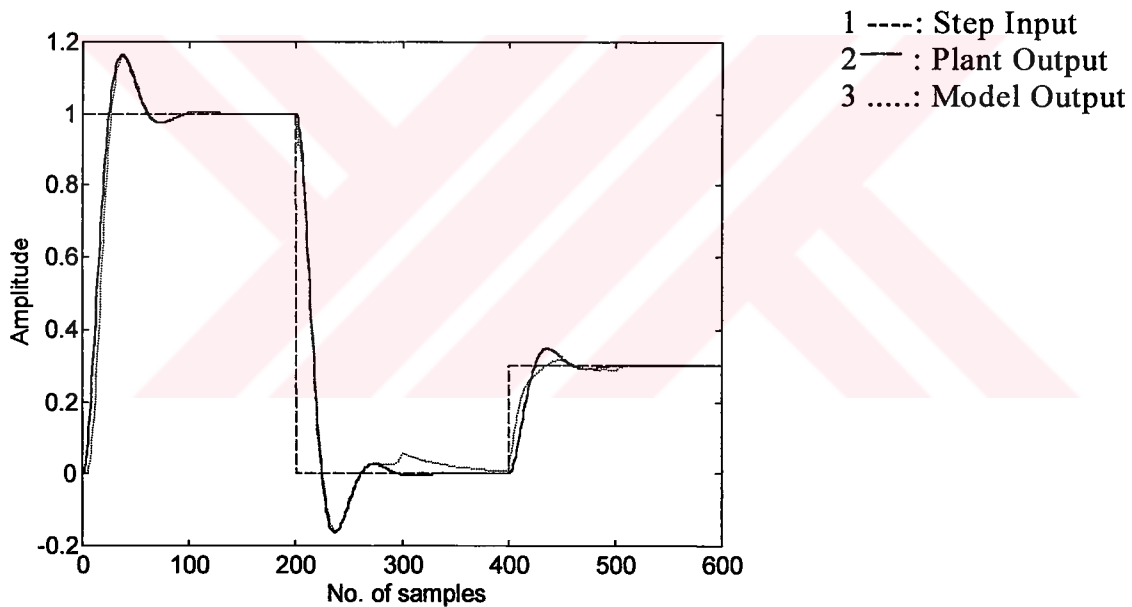


Fig.3 Comparison of model output with plant output over 600 iteration cycle with several step input values, $\mu=0.01$

For the smaller convergence factor , convergent would not be satisfactory.

In adaptive control systems, the input signals usually can not be foreseen. Therefore, a random input signal generator is programmed in this study. The random signals are firstly passed through an second order filter in order to soften their sharp peaks. After this filtered random output is entered to the plant and the model. Similar comparison between the model and the plant output is illustrated in Fig. 4. It is noticed that the model can follow the plant perfectly from the beginning as the μ is chosen correctly. If it had been chosen smaller, there would be lack of following the plant at the beginning but it

would follow after a few hundreds steps. For the bigger value of $\mu=0.18$ it is observed that model becomes unstable and for the smaller value the error increases in predictable manner.

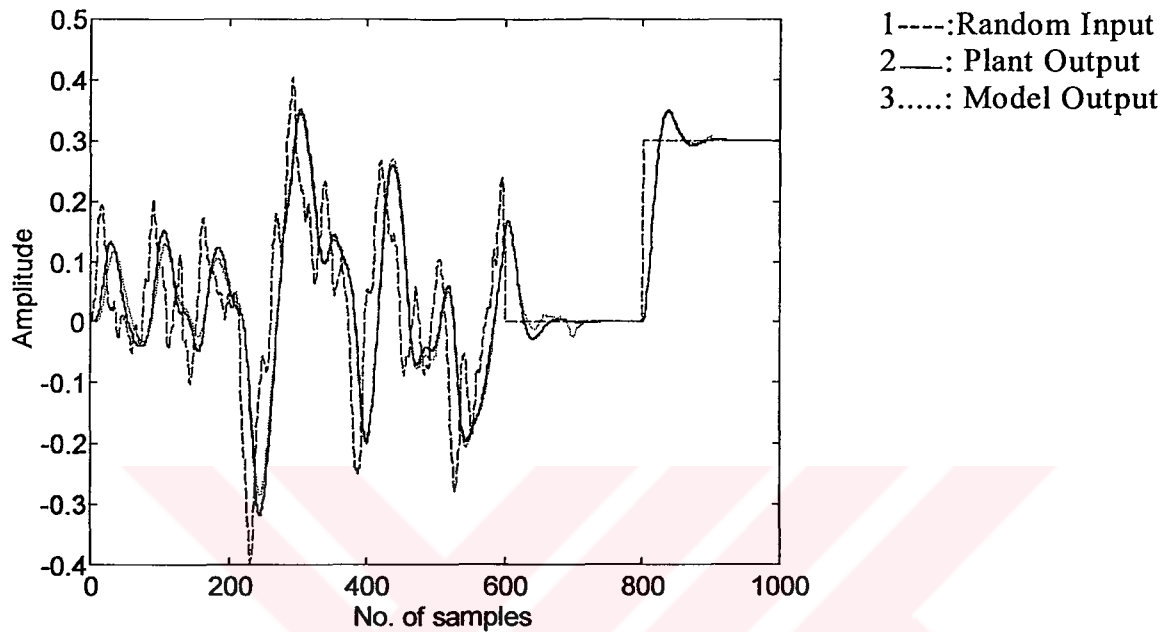


Fig.4 Comparison of model output with plant output with random input signals, $\mu=0.08$

A block diagram of the adaptive control system is shown in Fig.5. The functions labelled “forward-time calculation” is the inverse of the model in reality. This box generates x_k from the reference command input r_k and from the weight vector W_k and the input vector X_k .

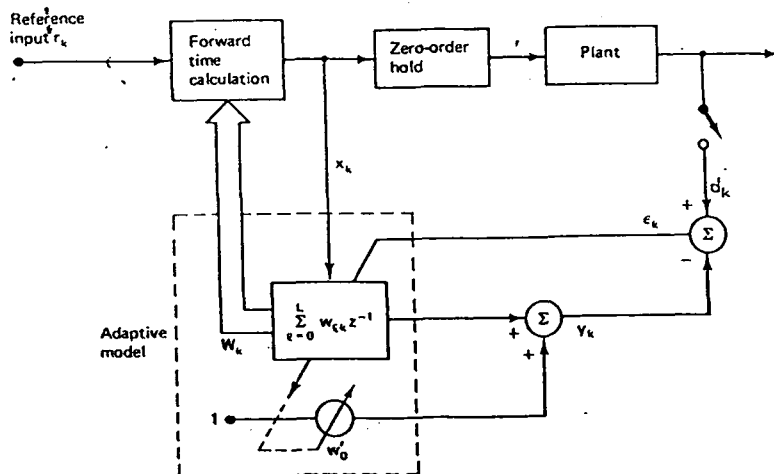


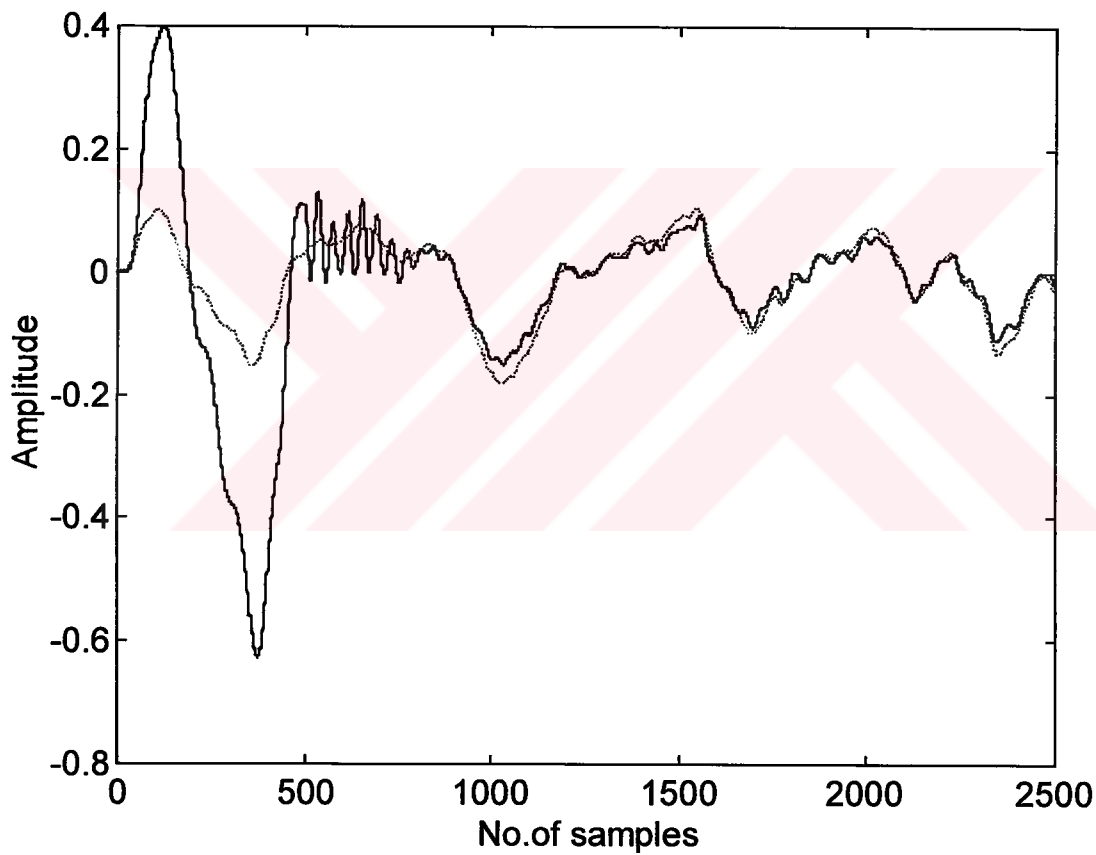
Fig.5 Block Diagram of Adaptive Model Control System [3]

Assuming that $E[\varepsilon^2]$ has been driven close to zero by the adaptive process, the objective of this box is to derive x_k from r_k such that r_k and y_k are equal. If r_k and y_k are equal, then (with ε_k^2) the plant output g_k (equal to desired response d_k) will be close to r_k , which is the ultimate goal for this study.

If it is considered that there are delays in the plant such as transport lag, the x_k can be written as follows, with y_k equal to r_k

$$x_k = 1/w_{2k} [r_k - w_{0k} - \sum_{n=3}^L w_{nk} x_{k+2-n}] \quad (1.8)$$

Before using this formula in MATLAB program, the plant is programmed in direct form.



1----: Input
2—: Plant output

Fig.6 Tracking performance: A comparison of the plant output and the model output (Control starts at 500.step)

BÖLÜM 1

GİRİŞ

Günlük dilde “uyarlamalı” sözcüğü, değişen çevre şartlarına uyum sağlayabilir anlamında kullanılmaktadır. Uyarlamalı bir regülatör, dinamik bir sistemin davranış değişikliklerine ve bozucu dış etkenlere karşı kendini ayarlayıp istenen kontrolü sağlamaktadır. Adaptasyon, değişen çevre şartları altında fizyolojik dengelerini korumak zorunda olan canlı organizmalar için temel bir özelliktir. Bu nedenle uyarlamalı sistemlerin dizaynında bu yaklaşım kullanılmıştır [1,2].

Özellikle son yıllarda bu konuya olan ilgi artmış ve robotik, uzay araçları, uçak, radar, biyomedikal cihazlar ile diğer kontrol uygulamalarında kendine özel bir yer edinmiştir. Geri beslemeli çoğu kontrol uygulamalarında parametrelerin, dizayn değerlerinden ufak sapmaları normal çalışma şartları altında sorun yaratmaz. Fakat sistem parametreleri değişen çevre şartlarından dolayı çok büyük değişiklikler gösterirse sistem bazı şartlarda tatmin edici bazı şartlar altında da tamamen yetersiz cevaplar verecektir. Bazı durumlarda sistem parametrelerindeki bu değişim kararsızlığa da neden olabilmektedir [3].

Uyarlamalı bir kontrol, sürekli ve otomatik olarak sistemin dinamik karakteristiklerini ölçer (transfer fonksiyonu veya durum denklemleri gibi) ve istenen dinamik karakteristiklerle kıyaslar, farkları genelde kontrol parametrelerini ayarlamak için kullanır veya değişen çevre şartlarından bağımsız olarak en uygun performansı korumak için gerekli olan giriş sinyallerini oluşturur.

Sistemin uyarlamalı olabilmesi için kendi kendini organize edebilmesi gerekir. Eğer sistem parametre ayarlamaları sadece çevreden elde edilen direkt ölçümlerle yapılıyorsa bu uyarlamalı kontrol değildir [2].

Parametreleri farklı set değerlerinden oluşan bir sistem ele alınsın ve bu sistemin tüm set değerleri için iyi bir performans göstermesi istensin. Bu istek formüle edilir edilmez en uygun kontrol problemi hemen önemini yitirir.

Sonuç olarak uyarlamalı sistemlerin karakteristikleri özetlenecek olursa:

1. Sabit olamayan değişken çevre şartlarına ve değişen sistem ihtiyaçlarına göre kendisini ayarlayabilir.
2. Uyarlamalı sistem programlanması, belli bir süre çalıştırılarak sisteme öğretim yöntemi ile yapılabilir.
3. Bu nedenle uyarlamalı sistemlerde, kendi kendilerine dizayn edildikleri için çok hassas analizlere gerek yoktur.
4. Belli öğretim aşamasından sonra bildikleriyle gelecekte olacakları tahmin edebilirler.
5. Belli sınırlar çerçevesinde kendi kendilerini onarabilirler. Bu da bazı iç aksaklıklarına rağmen görevlerini sürdürebilmeleri demektir.
6. Genelde, zamana bağlı değişken parametrelili lineer olmayan sistemler olarak tanımlanırlar.
7. Çoğunlukla uyarlamasız sistemlere göre analiz açısından çok daha kompleksler fakat giriş sinyali bilinmez ve zamana göre değişirken sistem performansını artırmak için yeterince imkan sunar [3].

Tüm bu anlatılanlardan da anlaşıldığı gibi uyarlamalı kontrol, yaygın klasik kontrolden çok daha ileri düzeyde bir uygulama olup yapı olarak öğrenen bir sistem olmasıyla da yeni ufuklar açmıştır. Bu çalışmanın konusu olan uyarlamalı model kontrol yaklaşımı gibi, tek girişli tek çıkışlı (SISO) ve çok girişli çok çıkışlı (MIMO) uyarlamalı ters model kontrol, lineer olmayan uyarlamalı ters kontrol, yapay sinir ağlarına bağlı lineer olmayan uyarlamalı filtreler gibi pek çok araştırma konuları üzerinde son senelerde halen çalışılmaktadır. Gelişen yüksek teknolojiye bağlı tüm çalışmalarda (robotik, uydu, radar vs.) kaçınılmaz olarak kullanılan bu tekniğin diğer büyük özelliği de gürültüyü iptal edebilir olmasıdır.

Bu çalışmada öncelikle basit bir karakteristiği olan bir sistem ele alınmıştır. Kontrol uygulamalarında sıkça kullanılan programlama dili MATLAB kullanılarak bu sistemin bir modeli oluşturulmuş ve rasgele sisteme verilen giriş değerlerine karşılık sistemin ve oluşturulan modelin verdiği cevaplar kıyaslanmıştır. Daha sonra da bu sistemin kontrolü için ileri zamanlı hesaplama tekniği kullanılarak kontrol sağlanmıştır. Tüm bu çalışmalar MATLAB ortamında programlanmış olup, adım adım grafiklerle destekli olarak yapılan simülasyonlar yorumlarıyla birlikte sunulmuştur.

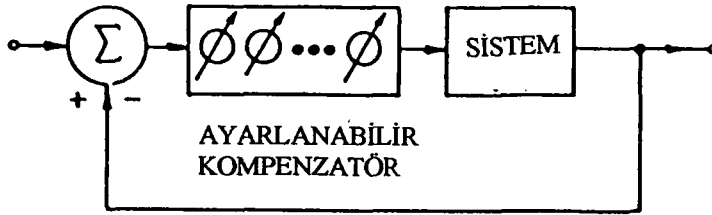


BÖLÜM 2

UYARLAMALI SİSTEMLER

Uyarlamalı sistemlerin esas özellikleri zamana göre değişen ve kendi kendini ayarlayabilir olmalarıdır. Bir sistem dizayn edilirken olabilecek tüm giriş değerlerine karşı alınacak çıkış değerleri öngörülür. Sistem çalıştırılmaya başlandıktan sonra , gerçek sistemin çıkışı ile modelin veya ideal sistemin çıkışı arasındaki hata sistemin performansını değerlendirmek için bir kriter olarak seçilirse bundan sonraki aşama bu hatayı en aza indirmektir.

Uyarlamalı bir sistemin basit bir diyagramı Şekil 2.1’de verilmektedir. Bu sistem düşünce olarak basittir fakat adaptasyon açısından verimsiz ve uğraşılması zordur. Kompensatör, gerçek zamanda uygun bir çıkış alınabildiğinde veya sistemi çalıştıran sinyale karşılık kompensatör giriş sinyali elde edilebildiğinde kolayca uyarlanabilir. Buna rağmen sistem bilinmediğinde bu sinyali elde etmek çoğunlukla zordur. Zaten sinyal elde edilebilir olsa kompensatör ve geri besleme çevrimi gereksiz olurdu [4].



Şekil 2.1 Uyarlamalı kontrol sistemi [4]

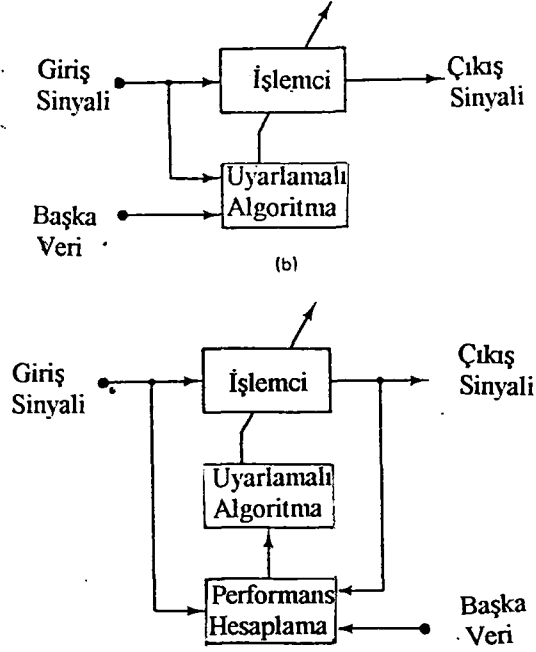
Bu durumda şöyle bir yaklaşım geliştirilmiştir. Amaç ortalama kare olarak hatayı minimum yapmak olsun. Gradyen bileşenleri, kompenzator ayarlarını değiştirerek ölçülebilir. Ortalama kare hata, bir gradyen metodu olan dik inme yöntemi kullanılarak minimum yapılabilir. Bu noktada karşılaşılan iki zorluk vardır. Ayar değiştirme tekniği ne olursa olsun, kompenzator ayar değişimleri ve sistem parametre değişimleri her bir adımda ölçülebilmesi için sistemin rejime girmesi beklenmelidir. Bununla birlikte gradyenin başarıyla ölçüldüğü varsayılın, ortalama kare hata performans fonksiyonu irregular ve parabolik olmayıp birden fazla minimum veya maksimum içeriyor olabilir [4].

Bu tür zorlukların üstesinden gelebilecek yaklaşım, uyarlamalı model kontrol tekniğidir ve bu çalışmanın esasını oluşturmaktadır. Öncelikle sistemin bir modeli oluşturulur ve uyarlamalı proses kullanılarak model sürekli güncellenir. Sonra bu model kullanılarak, gelecekte istenilen model çıkışını verecek modele giriş değerini belirlemek için ileri zamanlı analiz yapılır böylece de model iyi bir şekilde kontrol edilir. Daha sonra aynı kontrol gerçek sisteme uygulanır ve eğer model sisteme benzer şekilde davranırsa gerçek sistemin çıkışı yakın bir şekilde kontrol edilir. Sistemin kontrolü bir bakıma açık çevrimdir fakat aslında uyarlamalı proses (ileri zamanlı hesaplama) sayesinde çevrim kapatılmıştır.

2.1 Açık Çevrimli ve Kapalı Çevrimli Uyarlamalı Sistemler

Açık çevrimli uyarlamalı prosesde, giriş değerlerinin veya çevre karakteristiklerinin ölçümü yapılır ve bu bilgi bir formüle veya bir algoritmaya uygulanır, sonuçlarıyla da uyarlamalı sistemin ayarlanması yapılır. Kapalı çevrim uyarlamalı sistemde ise ölçülen sistem performansını uygun hale getirmek için bu ayarlamalardan ve sonuç bilgilerinden faydalanılır [3].

Açık ve kapalı çevrimli uyarlamalı sistemlerin gösterimi Şekil 2.2’de verilmiştir.



Şekil 2.2 Açık ve kapalı çevrimli uyarlamalı kontrol genel şeması [3]

Tasarım sırasında belli faktörler kapalı ve açık çevrim seçimini belirler. Giriş sinyallerinin elde edilebilirliği, toplam hesap kapasitesi ve bunu sağlayacak bilgisayar sistemi gibi unsurlar mevcuttur. Kapalı çevrimli kontrol, analitik sentez prosedürünün olmadığı veya bilinmediği uygulamalarda çalışılabilir olduğu için avantajlıdır. Örneğin sistem lineer olmadığı veya zamana bağlı olduğunda, giriş değerleri sabit olmadığı vs. Ayrıca kapalı çevrim, fiziksel sistem bileşenlerinin değişken veya yeterince bilinmediği durumlarda da efektif olarak kullanılabilir. Performans geri beslemesi kullanılarak sistem güvenilirliği sürekli gelişir.

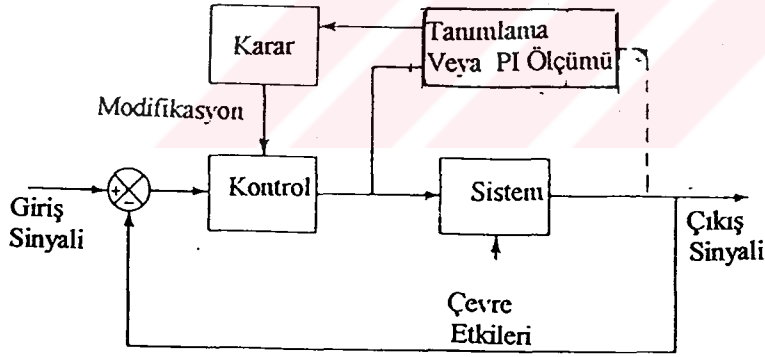
Kapalı çevrimli uyarlamalı kontrol, yinede bazı zorluklarla karşılaşabilmektedir. Bazı durumlarda performans fonksiyonunun birden fazla minimumu veya maksimum olabilir. Kapalı çevrimli adaptasyon prosesi, kapalı çevrimli bir kontrol sistemi gibi olduğunda kararsız olabilir ve yakınsama yerine uzaklaşabilir. Tüm bu ihtimallere rağmen, adaptasyon uygulamalarında performans geri beslemeli kontrol tekniği en çok kullanılanıdır ve bu çalışmada da bu teknik kullanılmıştır.

2.2 Uyarlamalı Kontrol

Uyarlamalı bir regülatör aşağıdaki üç fonksiyondan oluşur:

1. Sistemin dinamik karakteristiklerini tanımlama
2. Sistem tanımlamaya bağlı karar verme
3. Karar vermeye bağlı değişiklik yapma [2]

Zamana göre rasgele parametrelerin değişmesi veya değişken çevre şartlarının dinamik karakteristikleri üzerindeki etkisi nedeniyle sistem yeteri kadar bilinmiyorsa başlangıçta tanımlama, karar verme ve değişiklik prosedürleri performans ölçütünü minimum veya maksimum yapmak için yeterli olamayacaktır. Bundan dolayı, sistem parametrelerinin nasıl hızlı değiştiğine bağlı olarak sürekli veya belli zaman aralıklarında bu prosedürleri yapmak gerekir. Sistem içindeki tahmin edilemeyen değişiklikleri telafi etmesi için bu kendi kendini tasarlama, genellikle uyarlamalı kontrol sistemin bir yönüdür.



Şekil 2.3 Bir uyarlamalı kontrol sisteminin blok diyagramı [2]

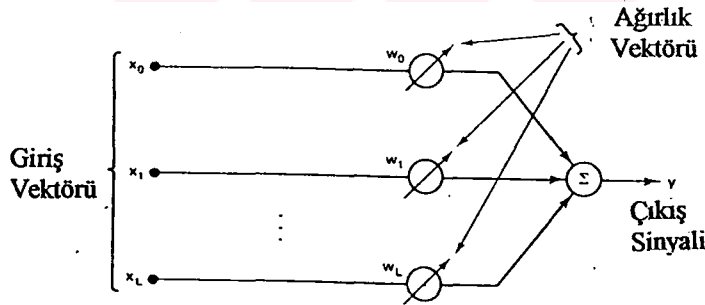
Uyarlamalı kontrol sistemine ait blok diyagramı Şekil 2.3’de verilmiştir. Burada sistem tanımlanmış ve performans ölçütü sürekli veya periyodik olarak ölçülmüştür. Bir kere bunlar yapıldıktan sonra performans ölçütü uygun değerler ile kıyaslanır ve tahrik sinyalini nasıl değiştireceğine dair karar verilir. Sistem kapalı çevrimli olduğu için parametre ayarlamaları geri besleme sayesinde yapılır.

BÖLÜM 3

UYARLAMALI LİNEER BİRLEŞTİRİCİLER

Uyarlamalı lineer birleştiriciler, uyarlamalı kontrol sistemlerindeki en temel taşlardan biridir. Esas olarak zamana bağlı ve tekrarlamasız (nonrecursive) bir dijital birleştiricidirler [3].

Şekil 3.1’de bu birleştiricinin genel bir formu verilmiştir. Giriş vektörü elemanları $x_0, x_1, \dots, x_L$ ve buna karşılık gelen ağırlık vektörleri $w_0, w_1, \dots, w_L$ olup çıkan toplam çıkış y ’dir. Birleştiricinin lineer olmasının sebebi, sabit bir değer olan ağırlık vektörleri için çıkış giriş bileşenlerinin lineer bir kombinasyonu olmasıdır. Fakat , operasyon sırasında ağırlık değerleri giriş değerlerinin fonksiyonu olarak değişeceğinden, artık çıkış girişin lineer bir fonksiyonu olmayacaktır.



Şekil 3.1 Uyarlamalı lineer bir birleştiricinin genel formu [3]

3.1 Giriş Sinyali ve Ağırlık Vektörleri

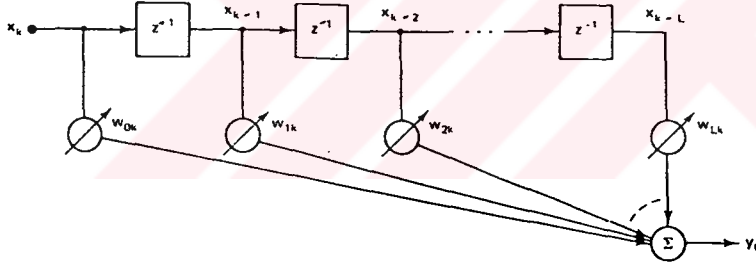
Şekil 3.1’deki giriş vektörü iki farklı şekilde olabilir. Birincisi, bunlar farklı kaynaktan gelen $L+1$ adet simultane sinyaller olabilir. İkincisi ise x_0-x_L aynı

kaynaktan gelen $L+1$ adet ardışık sinyaller olabilir. Bu iki ayrı durum sırasıyla çoklu giriş (multiple input) ve tek giriş (single input) olarak adlandırılacaktır. Bu çalışmada tekli giriş ele alındığından, giriş vektörü aşağıdaki gibi ifade edilebilir.

$$\mathbf{X}_k^T = [x_k \ x_{k-1} \ \dots \ x_{k-L}]^T \quad (3.1)$$

Bu gösterimde T transpoze olduğundan gerçekte $\mathbf{X}_k$ bir sütun vektördür ve k zaman indeksi ifadesi olarak kullanılmaktadır.

Tek giriş olması durumunda uyarlamalı işlemci Şekil 3.2’ de gösterildiği gibi bir gecikme elemanı ve lineer birleştiriciyle birlikte oluşturulabilir ve buna uyarlamalı lineer birleştirici denir. Uyarlamalı modelleme ve kontrol uygulamalarında bunun çok geniş uygulama alanları vardır ve bu çalışmada de bundan faydalanılmıştır.



Şekil 3.2 Tek girişli uyarlamalı lineer birleştirici [3]

Giriş ve çıkış arasında, aşağıdaki bağıntı elde edilir.

$$y_k = \sum_{l=0}^L w_{lk} x_{k-l} \quad (3.2)$$

$$\mathbf{W}_k = [w_{0k} \ w_{1k} \ \dots \ w_{Lk}]^T \quad (3.3)$$

$$y_k = \mathbf{X}_k^T \mathbf{W}_k = \mathbf{W}_k^T \mathbf{X}_k \quad (3.4)$$

3.2 İstenen Cevap ve Hata

Geri beslemeli uyarlanma sürecinde, ağırlık vektörleri, y_k çıkış değeri istenen cevap değerine mümkün olduğu kadar yakın olacak şekilde ayarlanır. Bu çıkış ile istenen cevap karşılaştırılıp aralarındaki hata değeri bulunur ve buna göre bu değeri en aza indirmek için ağırlık vektörleri optimize edilip ayarlanır. Genel olarak bu işlem, en küçük kareler yöntemi kullanılarak yapılır. Şimdilik istenen cevap d_k 'nın bir şekilde elde edilebilir olduğu kabulü yapılırsa hata sinyali ε_k basitçe aşağıdaki gibi elde edilir [3].

$$\varepsilon_k = d_k - y_k \quad (3.5)$$

Her k zaman aralığında $\mathbf{X}$ 'in değiştiği ve $\mathbf{W}$ 'nin sabit kaldığı varsayılırsa hata,

$$\varepsilon_k = d_k - \mathbf{X}_k^T \mathbf{W} = d_k - \mathbf{W}^T \mathbf{X}_k \quad (3.6)$$

olarak elde edilir. Hatanın karesi alındığında,

$$\varepsilon_k^2 = d_k^2 + \mathbf{W}^T \mathbf{X}_k \mathbf{X}_k^T \mathbf{W} - 2d_k \mathbf{X}_k^T \mathbf{W} \quad (3.7)$$

olur. Ortalama kare hata MSE (Mean-square error) ifadesi aşağıdaki gibi yazılır.

$$\text{MSE} = \xi = E[\varepsilon_k^2] = E[d_k^2] + \mathbf{W}^T E[\mathbf{X}_k \mathbf{X}_k^T] \mathbf{W} - 2E[d_k \mathbf{X}_k^T] \mathbf{W} \quad (3.8)$$

$\mathbf{R}$ olarak bir kare matris tanımlansın.

$$\mathbf{R} = E[\mathbf{X}_k \mathbf{X}_k^T] = E \begin{bmatrix} x_{0k}^2 & x_{0k}x_{1k} & x_{0k}x_{2k} & \dots & x_{0k}x_{LK} \\ x_{1k}x_{0k} & x_{1k}^2 & x_{1k}x_{2k} & \dots & x_{1k}x_{LK} \\ . & . & . & & . \\ . & . & . & & . \\ x_{Lk}x_{0k} & x_{Lk}x_{1k} & x_{Lk}x_{2k} & \dots & x_{Lk}^2 \end{bmatrix} \quad (3.9)$$

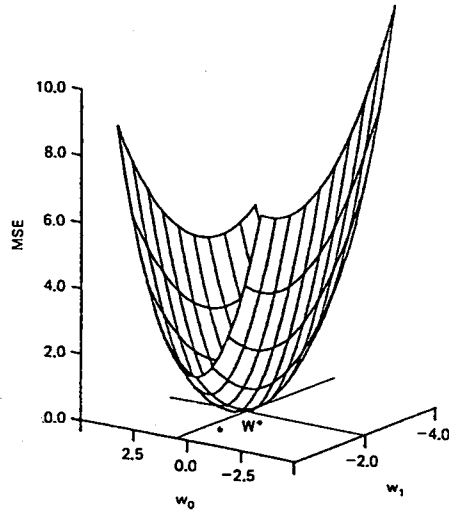
Bu matris “giriş korelasyon matrisi” olarak adlandırılmıştır. Benzer şekilde P sütun matrisi tanımlanabilir.

$$P = E[d_k \mathbf{X}_k] = E[d_k x_{0k} \quad d_k x_{1k} \quad \dots \quad d_k x_{Lk}]^T \quad (3.10)$$

(3.8) ifadesi yeniden düzenlenirse, aşağıdaki ifade elde edilir.

$$MSE = \xi = E[\epsilon_k^2] = E[d_k^2] + \mathbf{W}^T \mathbf{R} \mathbf{W} - 2 \mathbf{P}^T \mathbf{W} \quad (3.11)$$

Bu ifadeden açıkça anlaşılabileceği gibi ortalama kare hata ξ , giriş ve istenen cevap sabit iken W ağırlık vektör bileşenlerinin kuadratik bir fonksiyonudur. Şekil 3.3’de tipik bir ikinci dereceden MSE fonksiyonunun grafiksel gösterimi verilmiştir. Dikey eksen MSE ‘yi ifade ederken yatay eksenler iki ağırlık değerini gösterir. Çanak şekilli kuadratik hata fonksiyonu veya performans yüzeyi bu bağlamda bir paraboloiddir (ikiden fazla ağırlık değeri için hiperparaboloid olur). Yukarı doğru konkav olmak zorundadır çünkü aksi takdirde ağırlık ayarlandığında, gerçek fiziksel sinyallerde imkansız olan negatif MSE elde edilirdi.



Şekil 3.3 İkinci dereceden kuadratik bir performans yüzeyi [3]

Çanağın en altındaki $\mathbf{W}^*$, optimum ağırlık veya minimum MSE'dir. Kuadratik performans fonksiyonunda böyle sadece tek bir optimum nokta vardır.

3.3 Gradyen ve Minimum MSE

Bir çok uyarlamalı proses uygulamalarında, ağırlık vektörünü oluşturmada performans yüzeyindeki minimumu bulmada gradyen metodu kullanılır. MSE'nin gradyeni $\nabla(\xi)$ veya kısaca ∇ olarak gösterilir ve (3.11) ifadesinin diferansiyeli alınırsa aşağıdaki sütun vektör elde edilir.

$$\nabla = \partial \xi / \partial \mathbf{W} = [\partial \xi / \partial w_0 \quad \partial \xi / \partial w_1 \quad \dots \quad \partial \xi / \partial w_L]^T \quad (3.12)$$

$$= 2\mathbf{R}\mathbf{W} - 2\mathbf{P} \quad (3.13)$$

Minimum MSE'yi elde etmek için $\mathbf{W}$ ağırlık vektörü gradyenin sıfır olduğu $\mathbf{W}^*$ olarak ayarlanırsa:

$$\nabla=0=2\mathbf{R}\mathbf{W}^* - 2\mathbf{P} \quad (3.14)$$

$\mathbf{R}$ vektörünün tekil olmadığı varsayılırsa, optimum $\mathbf{W}^*$ ağırlık vektörü, bazen Wiener ağırlık vektörü olarak da adlandırılır, aşağıdaki gibi elde edilir.

$$\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P} \quad (3.15)$$

Şimdi minimum MSE (3.11) de $\mathbf{W}$ yerine $\mathbf{W}^*$ konularak bulunur.

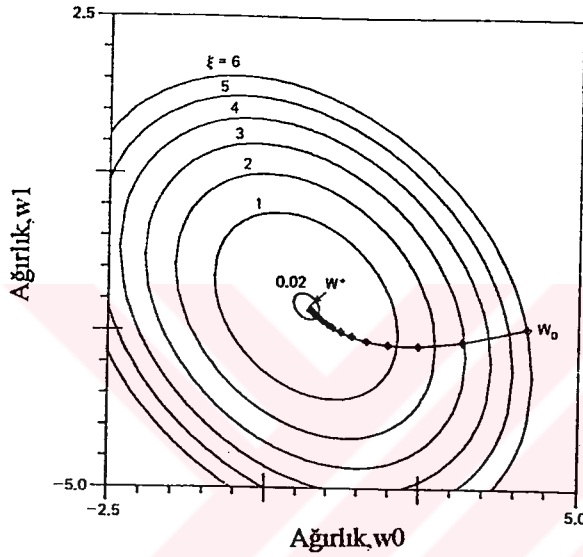
$$\xi_{\min} = E[d_k^2] + \mathbf{W}^{*T}\mathbf{R}\mathbf{W}^* - 2\mathbf{P}^T\mathbf{W}^* \quad (3.16)$$

(3.15)'den $\mathbf{P} = \mathbf{R}\mathbf{W}^*$ olur ve buradan (3.16) aşağıdaki gibi yazılır.

$$\begin{aligned} \xi_{\min} &= E[d_k^2] + \mathbf{W}^{*T}\mathbf{P} - 2\mathbf{P}^T\mathbf{W}^* \\ &= E[d_k^2] + \mathbf{P}^T\mathbf{W}^* - 2\mathbf{P}^T\mathbf{W}^* \\ \xi_{\min} &= E[d_k^2] - \mathbf{P}^T\mathbf{W}^* \end{aligned} \quad (3.17)$$

3.4 Dik İnme Yöntemi (*Steepest Descent*) ile Gradyen Arama

Gradyen arama, Newton metodu, Southwell metodu ve Rasgele arama metodları gibi bir çok yöntemle aranabilir. Uyarlamalı sistem uygulamalarında genelde dik inme yöntemi kullanılır. Ağırlık vektöründeki değişimler, performans yüzeyinin negatif gradyeni boyunca yönlendirilir. Şekil 3.4’de iki ağırlık değeri için dik inme yönteminin izlediği yol görülebilir [3].



Şekil 3.4 İki ağırlık değeri için dik inme yöntemi gösterimi [3]

Tanımından, dik inme yöntemi aşağıdaki algoritmayla ifade edilebilir.

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu \nabla_k \quad (3.18)$$

μ yakınsama faktörü sabit bir değer olup adım büyüklüğünü ifade eder.

3.5 En Küçük Ortalama Kareler (LMS) Algoritması

Dik inme yöntemi, performans yüzeyinin minimum noktasına ulaşmak için kullanılmıştır. Bu yöntem her iterasyon adımı bir kestirmeye ihtiyaç duyar. Burada en küçük ortalama kareler (Least Mean Squares) yönteminin üzerinde durulacaktır. LMS algoritması, uyarlamalı lineer filtreler için geçerli olan özel bir gradyen kestirme yöntemi kullanır. Kullanım ve hesabındaki kolaylık LMS algoritmasının önemini artırmaktadır. Giriş vektörü $\mathbf{X}_k$ olan uyarlamalı sistem

eğer lineer bir birleştirici ise ve istenen cevap d_k her iterasyon adımında elde edilebiliyorsa, LMS algoritması uyarlamalı sistem uygulamalarındaki en iyi seçimdir.

LMS algoritmasını geliştirebilmek için ϵ_k^2 'nin kendisi ξ_k 'nın bir kestirmesi (estimate) olarak alınır ve her iterasyonda aşağıdaki gradyen kestirmesi oluşur.

$$\begin{aligned}\nabla_k &= [\partial \epsilon_k^2 / \partial w_0 \quad \partial \epsilon_k^2 / \partial w_1 \quad \dots \quad \partial \epsilon_k^2 / \partial w_L]^T \\ &= 2\epsilon_k [\partial \epsilon_k / \partial w_0 \quad \partial \epsilon_k / \partial w_1 \quad \dots \quad \partial \epsilon_k / \partial w_L]^T \\ &= -2\epsilon_k \mathbf{X}_k\end{aligned}\quad (3.19)$$

(3.18)'den

$$\begin{aligned}\mathbf{W}_{k+1} &= \mathbf{W}_k - \mu \nabla_k \\ &= \mathbf{W}_k + 2\mu \epsilon_k \mathbf{X}_k\end{aligned}\quad (3.20)$$

Bu yukarıdaki LMS algoritmasındaki μ ifadesi daha öncede bahsedildiği gibi adaptasyonun hızını ve kararlılığını belirleyen yakınsama faktörüdür. Her iterasyonda ağırlıkların değişimi, mükemmel olmayan gradyen kestirmeden kaynaklandığı için sistemin gürültülü olması beklenir ve bu nedenle de performans yüzeyinin dik inme doğrusunu takip edemeyecektir. Pratik sistemlerde LMS algoritması kare, diferansiyel ve ortalama almadan kullanılabilir.

3.6 Ayarsızlık

Görüldüğü üzere uyarlamalı dijital filtre, uyarlamalı bir algoritmayla kontrol edilen bir gecikme zincirinden ve ayarlanabilir ağırlıklardan oluşmaktadır. Sabit giriş değerleri için ortalama kare hata MSE, dijital filtre çıkışı ile dışarıdan elde edilen istenen cevap arasındaki fark olup ağırlıkların kuadratik bir fonksiyonudur ve minimum sabit bir noktası olan bu paraboloid gradyen tekniği kullanılarak çözülebilir. Gradyen kestirme prosesinde adaptasyon hızıyla ve ağırlıkların sayılarıyla orantılı olarak bir gürültü etkisi oluşur. Bu etki optimum performanstan sapmanın bir ölçüsü olarak boyutsuz bir ifade

olan *ayarsızlık (misadjustment)* olarak gösterilir. Dik inme yöntemine bağlı LMS algoritması, giriş korelasyon matrisi özdeğerlerinin birbirlerine yakın veya eşit olduklarında, ayarsızlık ve adaptasyon hızı anlamında verimliliğin teorik limitlerine yaklaşmış olur. Özdeğerler tamamen farklı olduğunda ($\lambda_{\max}/\lambda_{\min} > 10$) LMS'e yakın bir algoritma olan fakat Newton metoduna bağlı bir yaklaşım bu teorik limitlere çok yaklaşmaktadır. Uyarlamalı lineer filtre için (3.11) ifadesine (3.15) ve (3.17) konulacak olursa aşağıdaki bağıntı elde edilir [3,5].

$$\xi = \xi_{\min} + (\mathbf{W} - \mathbf{W}^*)^T \mathbf{R} (\mathbf{W} - \mathbf{W}^*) \quad (3.21)$$

$\mathbf{V} = (\mathbf{W} - \mathbf{W}^*)$ olarak tanımlanırsa;

$$\xi = \xi_{\min} + \mathbf{V}^T \mathbf{R} \mathbf{V} \quad (3.22)$$

Ve buradan gradyen ifade alınır;

$$\nabla = 2\mathbf{R}\mathbf{V} \quad (3.23)$$

$\mathbf{Q}$ özdeğer matrisi ve Λ bu matrisin köşegen matrisi olarak tanımlanırsa $\mathbf{R}$ için aşağıdaki ifade yazılabilir.

$$\mathbf{R} = \mathbf{Q} \Lambda \mathbf{Q}^{-1} = \mathbf{Q} \Lambda \mathbf{Q}^T \quad (3.24)$$

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_p, \dots, \lambda_n] \quad (3.25)$$

(3.22) tekrar bu eşitliklere göre tekrar yazılacak olursa;

$$\xi = \xi_{\min} + \mathbf{V}^T \mathbf{Q} \Lambda \mathbf{Q}^{-1} \mathbf{V} \quad (3.26)$$

$\mathbf{V}$ 'nin transforme bir versiyonu aşağıdaki gibi tanımlansın.

$$\mathbf{V}' = \mathbf{Q}^{-1} \mathbf{V} \text{ ve } \mathbf{V} = \mathbf{Q} \mathbf{V}' \quad (3.27)$$

(3.20) ifadesinde (3.23), (3.24) ve (3.27) ifadesi kullanıldığında;

$$\mathbf{V}'_{k+1} - (\mathbf{I} - 2\mu\Lambda) \mathbf{V}'_k = 0 \quad (3.28)$$

Bu homojen vektör fark eşitliğinde basit bir geometrik çözüm getirildiğinde,

$$\mathbf{V}'_0 = \mathbf{W}'_0 - \mathbf{W}^* \quad (3.29)$$

ve $\mathbf{V}'_0$ ilk şartlar olarak tanımlanırsa,

$$\mathbf{V}'_k = (\mathbf{I} - 2\mu\mathbf{\Lambda})\mathbf{V}'_0 \quad (3.30)$$

elde edilir. (3.30) ifadesinin kararlılığı için, $\lambda_{\max}$ $\mathbf{R}$ 'nin en büyük özdeğeri olarak tanımlanır.

$$1/\lambda_{\max} > \mu > 0 \quad (3.31)$$

LMS algoritması için bir çok yakınsama denemeleri senelerce yapılmıştır. Her birisi yakınsamaları bazı kabuller altında gerçekleştirmiştir. (3.31) ağırlık vektörünün yakınsamasında yeterli gibi gözükse de ağırlık vektörünün uyumsuzluk durumlarında yetersizdir. Aşağıdaki gibi daha kuvvetli bir şart gereklidir.

$$1/\lambda_{\max} > 1/\text{tr}\mathbf{R} > \mu > 0 \quad (3.32)$$

Yukarıdaki ifadede $\text{tr}\mathbf{R}$ ağırlıklara giden tüm sinyallerin ortalama kareleridir.

$$\text{tr}\mathbf{R} = \sum (\mathbf{\Lambda}'\text{'nin köşegen elemanları}) \quad (3.33)$$

Ayarsızlık ifadesi M ;

$$M = \mu \text{tr}\mathbf{R} = n / N \quad (3.34)$$

n : Ağırlık Sayısı

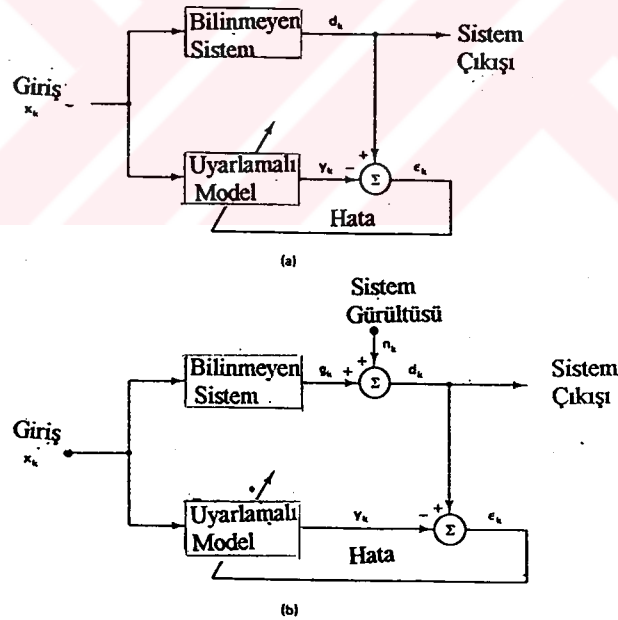
N : Bağımsız çalışma örneklerinin sayısı

Bu bağıntı, verilen veri örnekleri için ağırlık vektör büyüklüğünü seçmede kullanılır. Özdeğerler eşit olduğu zaman N rejime girme zamanına (settling time) tekabül etmektedir. Bazı uygulamalarda %10 ayarsızlık kabul edilebilmektedir.

BÖLÜM 4

UYARLAMALI MODELLEME

Bir veya çok girişli ve bir veya çok çıkışlı “kara kutu” olarak tanımlanabilecek karakteristiği bilinmeyen bir dinamik sistemin modellenmesinde, bu sistemin davranışlarını taklit edecek uyarlamalı bir birleştirici kullanılabilir. Tek giriş ve tek çıkışlı bir dinamik sisteminin modellenmesinin diyagramı Şekil 4.1’de verilmiştir. Hem bilinmeyen sistem hem de birleştirici aynı giriş değeriyle beslenmektedir. Uyarlamalı birleştirici, kendi çıkış değeri ile bilinmeyen sistemin çıkış değerinin birbirine uymasını için kendi kendini ayarlar. İki çıkış arasındaki hatayı minimum etmek için daha önce bahsedilen LMS algoritması kullanılır [3].

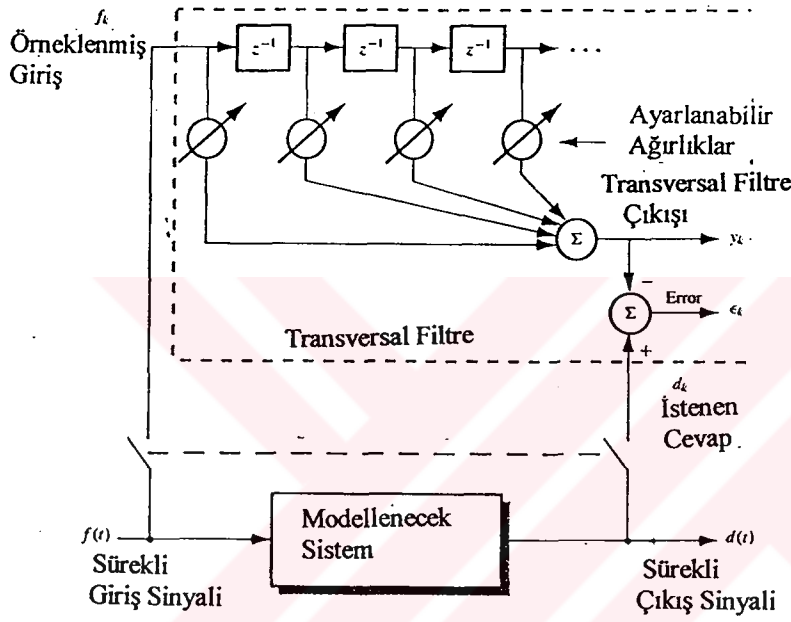


Şekil 4.1 Tek girişli ve tek çıkışlı dinamik bir sistemin modellenmesi [3]

4.1 Transversal Filtre

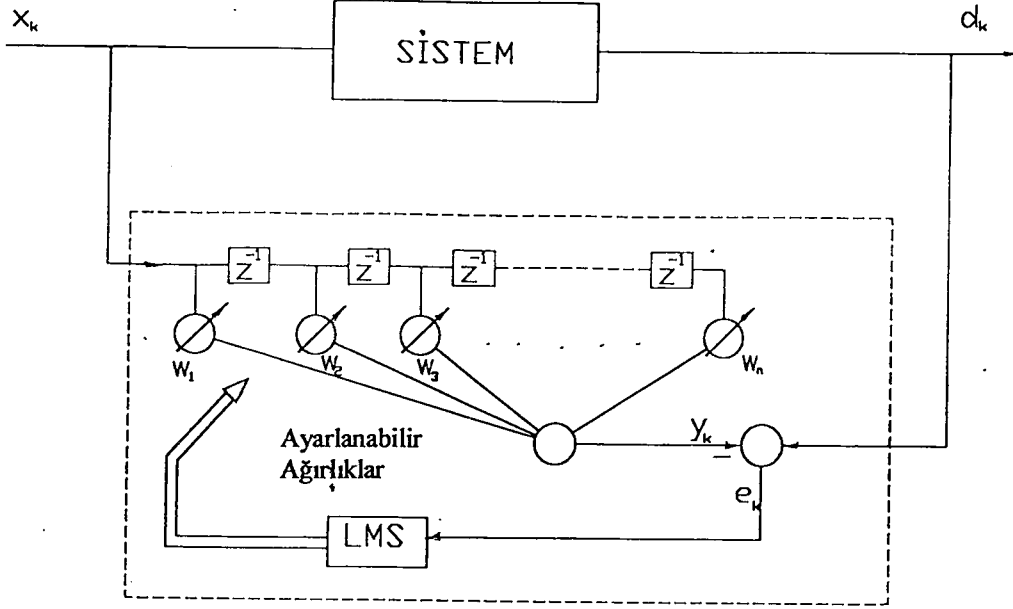
Zamana bağlı lineer dinamik sistemler, impuls cevaplarıyla karakterize edilirler. Kararlı sistemler için impuls cevabı sıfıra gidecek ve sonrada

kesilebilecektir. Bu sonuç, sonlu impuls cevabı kısaca FIR (Finite Impuls Response) olarak veya transversal filtre olarak adlandırılır. Transversal filtre değişken ağırlıklardan oluşur ve giriş sinyali gecikmeli hat (tapped delay-line) üzerindeki sinyaldir. Ağırlıklandırılmış sinyaller bir toplayıcıda toplanır ve adaptasyon prosesi ağırlıkları ayarlayarak otomatik olarak optimum cevabı arar. Şekil 4.2 bilinmeyen bir sistemin modellenmesinde kullanılabilecek filtrenin yapısını göstermektedir.



Şekil 4.2 Ayırık zamanlı transversal filtre ile modelleme [5]

Giriş sinyallerine ek olarak ϵ hata sinyalini oluşturabilmek için istenen cevap d_k 'nın sağlanması gerekmektedir. Burada bilinmeyen sistemin çıkışı istenen cevap d_k 'yı vermektedir. Ağırlıklar MSE'yi minimum yapacak şekilde otomatik bir algoritmayla ayarlanır. Ağırlıklar yakınsadığı zaman ve hata yeterince küçük olduğunda, birleştiricinin impuls cevabı bilinmeyen sistemin örneklenmiş cevabına uyacaktır. Dolayısıyla aynı giriş cevabına karşılık birleştirici ve sistem aynı çıkışı vereceklerdir.



Şekil 4.3 Uyarlamalı modelleme

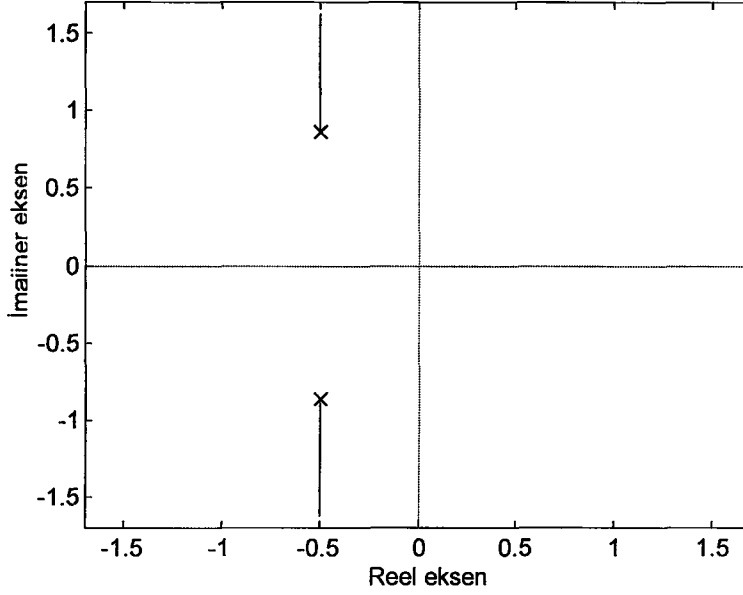
4.2 Uyarlamalı Modelleme

Uyarlamalı modelleme simülasyonu yapabilmek için otomatik kontrol uygulamalarında sıklıkla kullanılan MATLAB'den faydalanılmıştır. Bilinmeyen sistemin uyarlamalı modellenmesi bu çalışmada yapılmış ve sonuçları grafiksel olarak verilmiştir. Burada simülasyonu gerçekleştirebilmek için sistem önceden tanımlanacak fakat uyarlamalı algoritma için hala bilinmeyen olarak kalacaktır.

Bu simülasyon için ilk olarak aşağıdaki ikinci mertebeden transfer fonksiyonuna sahip bir sistem modellenecek sistem olarak ele alınmıştır.

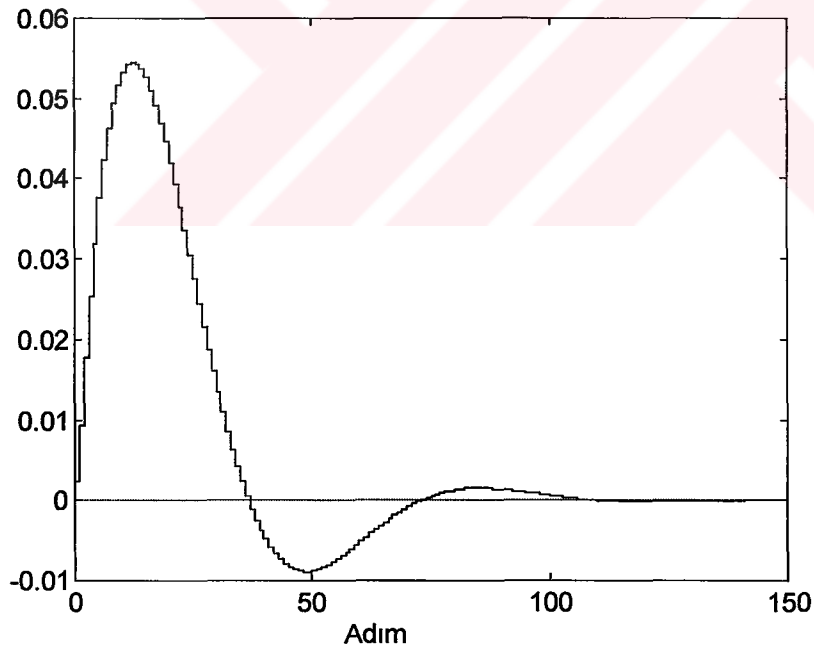
$$\frac{1}{s^2+s+1} \quad (4.1)$$

Karakteristiği bilindiği için öncelikle sistemin kararlı olup olmadığına Root-Locus yöntemi kullanılarak bakılabilir.



Şekil 4.4 Köklerin geometrik yeri

Şekil 4.4’de de görüldüğü gibi seçilen sistem kararlıdır. Yukarda seçilmiş olan sistemin ayrık zamanlı impuls cevabı aşağıdaki gibidir.



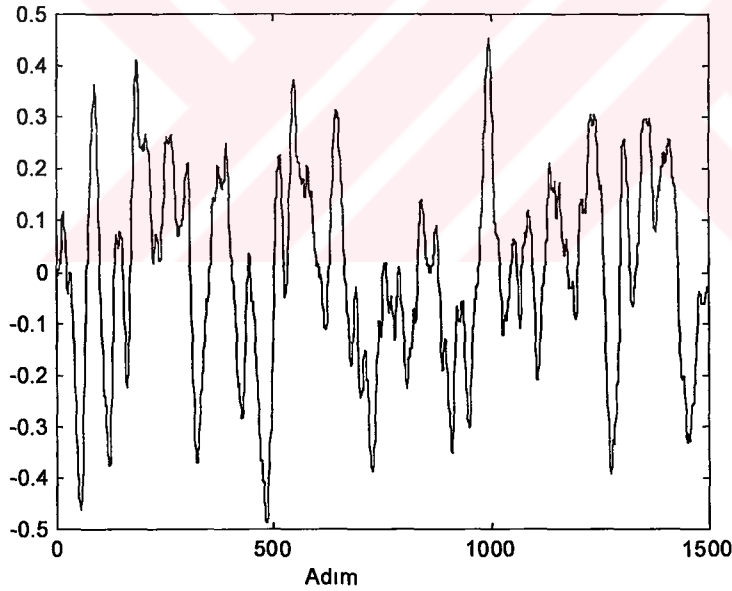
Şekil 4.5 Ayrık zamanlılaştırılmış sistemin impuls cevabı

Örnekleme periyodu, sistemin sürekli zamandaki impuls cevabına bakılarak $T_s=0.1$ sn. olarak yeterince küçük seçilmiştir. Sistem yaklaşık 100 örnekleme zamanında yani $100 \cdot 0.1 = 10$ saniyede rejime girdiği için ağırlık vektör sayısı

$10/0.1 = 100$ olarak seçilmiştir. Daha sonra örnekleme periyodu , ağırlık sayıları değiştirilerek model performansı gözlemlenecektir.

Uyarlamalı modellemede giriş sinyalleri genelde rasgele (random input) olup modelin, sistemin bu girişlere karşı verdiği çıkış değerleri takip etmesi beklenir. Bunun için öncelikle uygun bir sinyal kaynağı oluşturmak gerekmektedir. Bunun için MATLAB’de , ikinci dereceden bir sistemden geçirilerek yumuşatılmış rasgele giriş değerleri üretilmesi için *filtrand* adlı bir fonksiyon EK A’da verildiği gibi yazılmıştır.

Burada m , w_n ve T_s parametrik olup istenilen değerler girilebilmektedir. Rasgele üretilen x değeri klasik anlamda bir sönüm ve titreşim elemanından oluşan bir sisteme girilmekte ve böylelikle modellemede kullanılacak giriş sinyalleri üretilmektedir. Bu giriş sinyal üreticisinin *filtrand(1500,1,0.1)* değerleri için çıkışı aşağıdaki gibidir.



Şekil 4.6 Filtre edilmiş rasgele giriş sinyal üreticisi

EK A’da verilen MATLAB’de hazırlanan program sayesinde (4.1)’de verilen sistemin, uyarlamalı modellemesi yapılmıştır. Uyarlamalı modelleme Şekil 4.3’de gösterildiği gibi kurulmuş olup, ağırlıkların başlangıç değerleri sıfır alınmıştır. Ağırlıklar giriş sinyalinin bir fonksiyonu olmadığı için

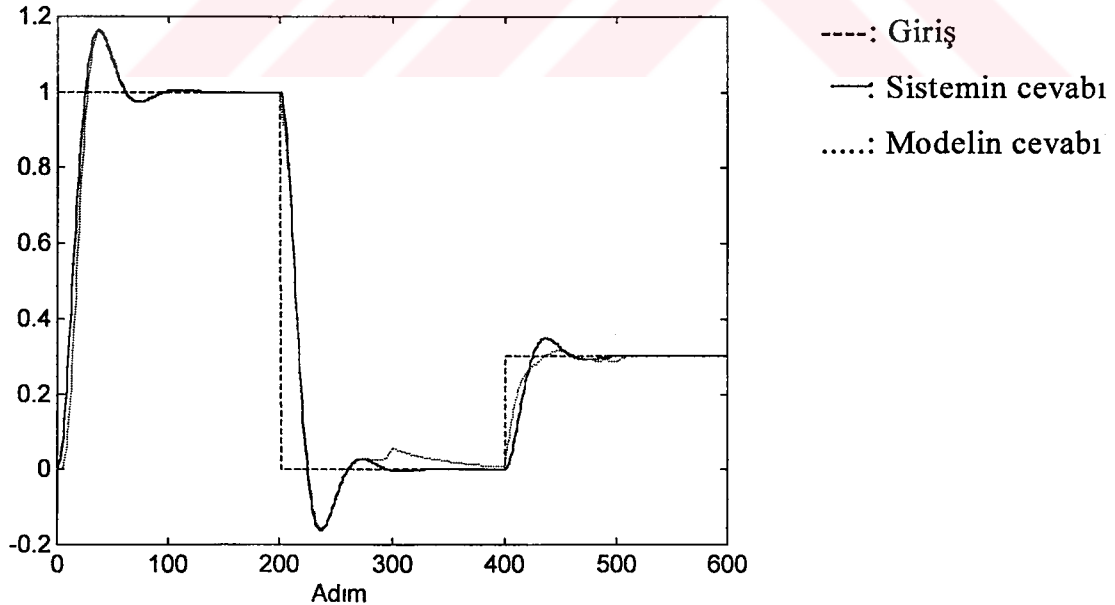
uyarlamalı model lineerdir. Uyarlamalı proses otomatik olarak ağırlıkları ayarlar öyle ki verilen giriş sinyalleri için model, sistemin örneklenmiş çıkışına uygun en iyi minimum ortalama kare hatayı sağlar.

Yakınsama faktörü μ adaptasyon açısından önemli bir etken olup (3.32) ve (3.33) 'de belirtilen kriterler doğrultusunda seçilir. Bunun için öncelikle R matrisi, giriş sinyali rasgele giriş sinyali üreticisinden çıkan x giriş sinyallerine göre düzenlenir. MATLAB'de oluşturulan 100×100 'lük bu kare matrisin özdeğer vektörü son elemanı hariç diğer tüm elemanları sıfır olup bu son elemanda program bir çok defa çalıştırılarak $\lambda_{\max}$ değerinin 1 ile 10 arasında değiştiği gözlemlenmiştir. Giriş değerleri birlerden oluşan bir dizi verildiğinde $\lambda_{\max}=100$ olarak bulunur.

Bu her iki farklı giriş sinyali yaklaşımında, uyarlamalı modellemenin yakınsama faktörü μ aşağıdaki gibi olur.

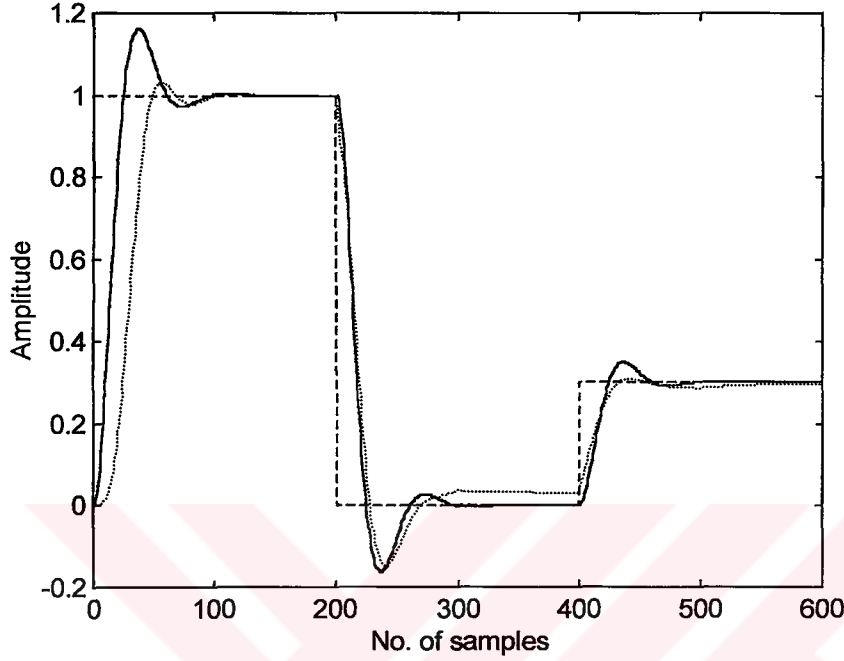
$$0.1 > \mu > 0 \quad (\text{Rasgele giriş değerleri için}) \quad (4.3)$$

$$0.01 > \mu > 0 \quad (\text{Basamak giriş=1 için}) \quad (4.4)$$



Şekil 4.7 Sistem ve uyarlamalı modelin, $\mu=0.01$ için çeşitli basamak giriş değerlerine cevapları

Yakınsama faktörü birinci kıyaslamada $\mu=0.01$ olarak sınırda seçilmiştir. Daha küçük bir yakınsama değeri $\mu=0.001$ olarak seçilirse sistem ve model çıkışları aşağıdaki gibi olur.

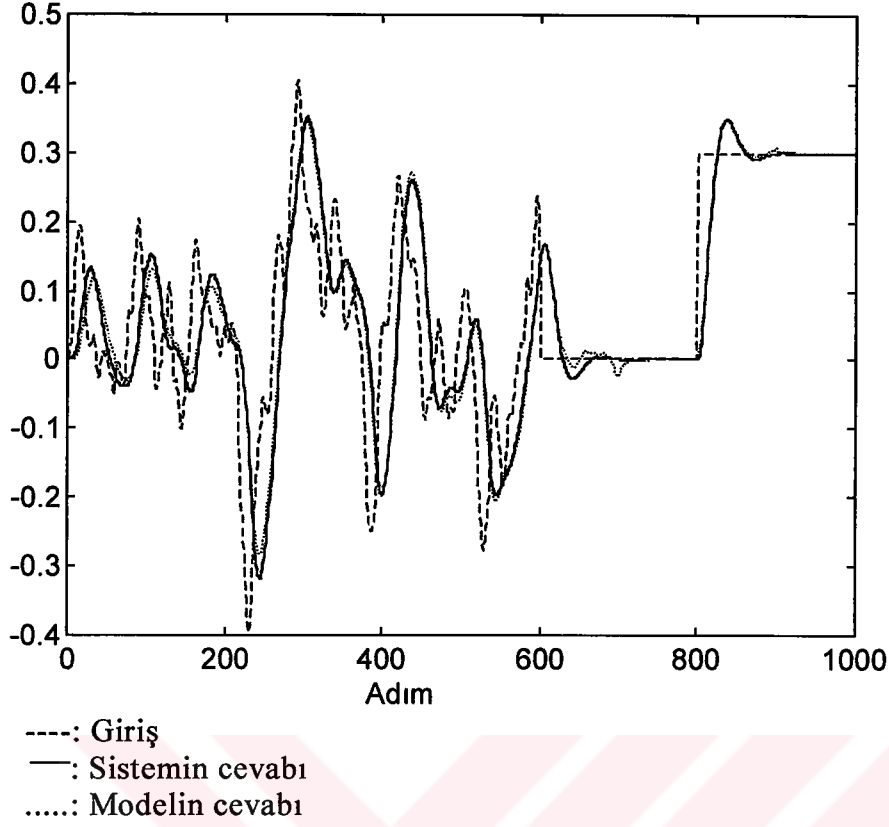


----: Giriş
 —: Sistemin cevabı
: Modelin cevabı

Şekil 4.8 Sistem ve uyarlamalı modelin, $\mu=0.001$ için çeşitli basamak giriş değerlerine cevapları

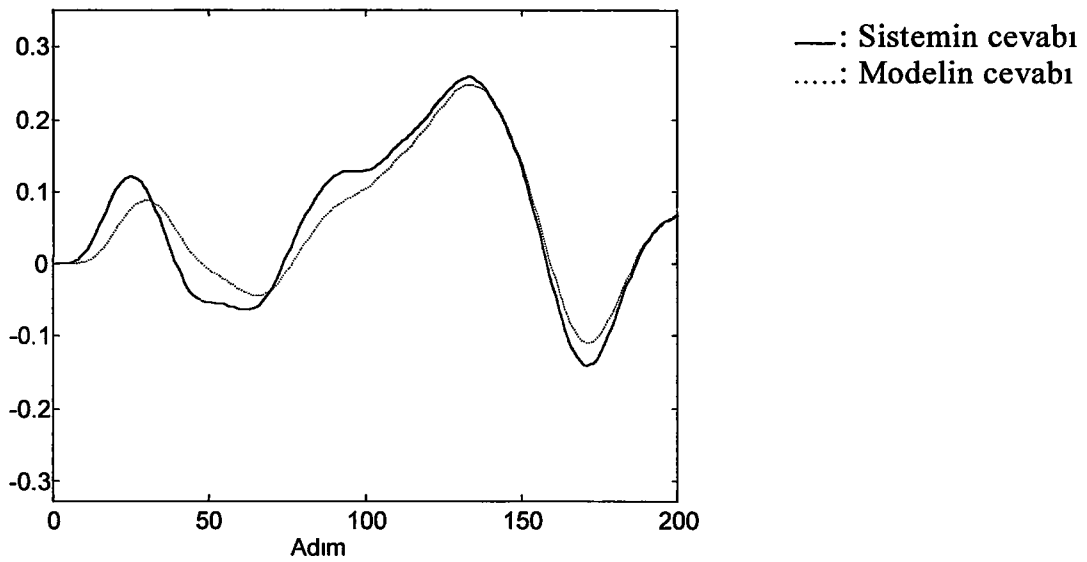
Görüldüğü gibi daha küçük μ değerleri için salınım azalmış fakat ε hata artmıştır. Yakınsama faktörü μ büyük seçildiği takdirde de model yakınsayamayacak ve kararsız olacaktır.

Şekil 4.6'da verilen rasgele sinyaller, ayrık zamanlılaştırılmış sisteme ve modele giriş olarak verilsin ve 1000 adımda benzer inceleme yapılsın. Burada sisteme 600. ile 800. adımlar arasında sıfır sabit girişleri ve 800. ile 1000. adımlar arasında da 0.3 sabit girişleri uygulanmıştır. Yakınsama faktörü μ , 4.3 ifadesi esas alınarak $\mu=0.08$ seçilmiş ve ağırlık sayısı daha önce de hesaplandığı gibi 100 olarak alınmıştır.



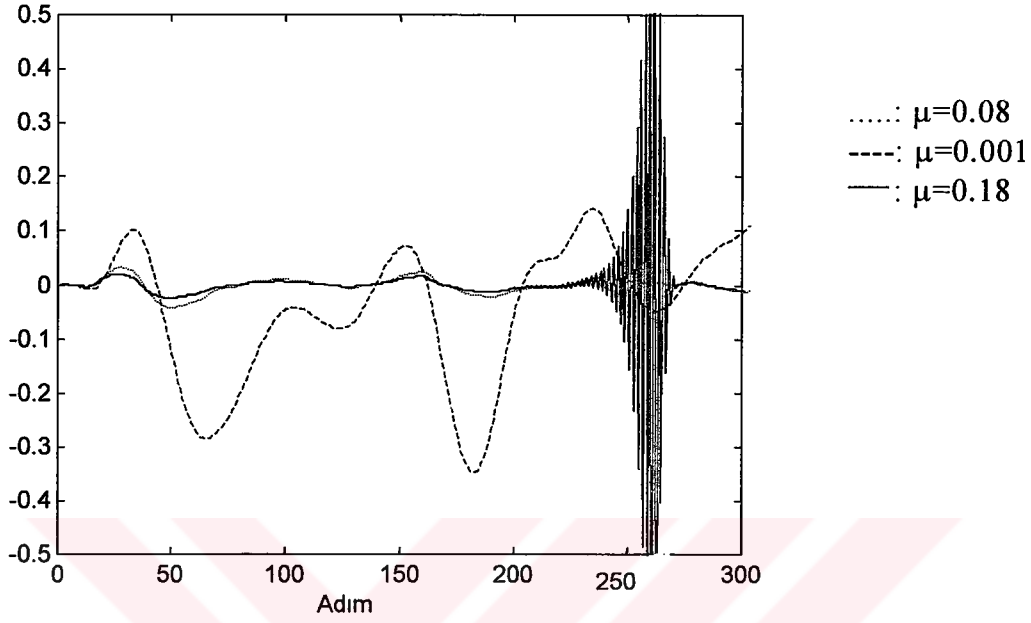
Şekil 4.9 Sistem ve uyarlamalı modelin, $\mu=0.08$ için rasgele giriş değerlerine cevapları

İlk 200 adımda model çıkışının sistem çıkışı ile karşılaştırılması için $\mu=0.08$ seçilirse model çıkışının zamanla nasıl sistemin çıkışını takip ettiği görülür.



Şekil 4.10 Sistem ve uyarlamalı modelin, ilk 200 adımda $\mu=0.05$ için rasgele giriş değerlerine cevapları

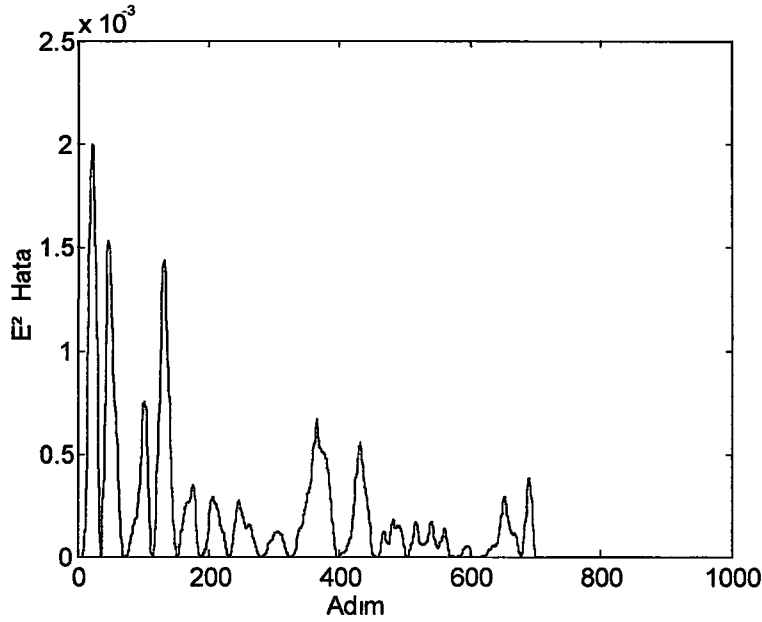
Şekil 4.11 üç farklı μ değeri için ε hatalarının 300 adımda nasıl değiştiğini grafiksel olarak göstermektedir.



Şekil 4.11 Farklı μ değerleri için ε hataların zamana bağlı değişimi

Yukarıdaki şekilde de görüldüğü gibi μ 'nın $\lambda_{\max}=0.1$ 'den büyük değeri 0.18 için ε hata değeri zaman zaman çok büyük boyutlara ulaşmaktadır. Yine aynı şekilde $\mu=0.001$ için ise ε hata istenildiği gibi minimum edilememiştir.

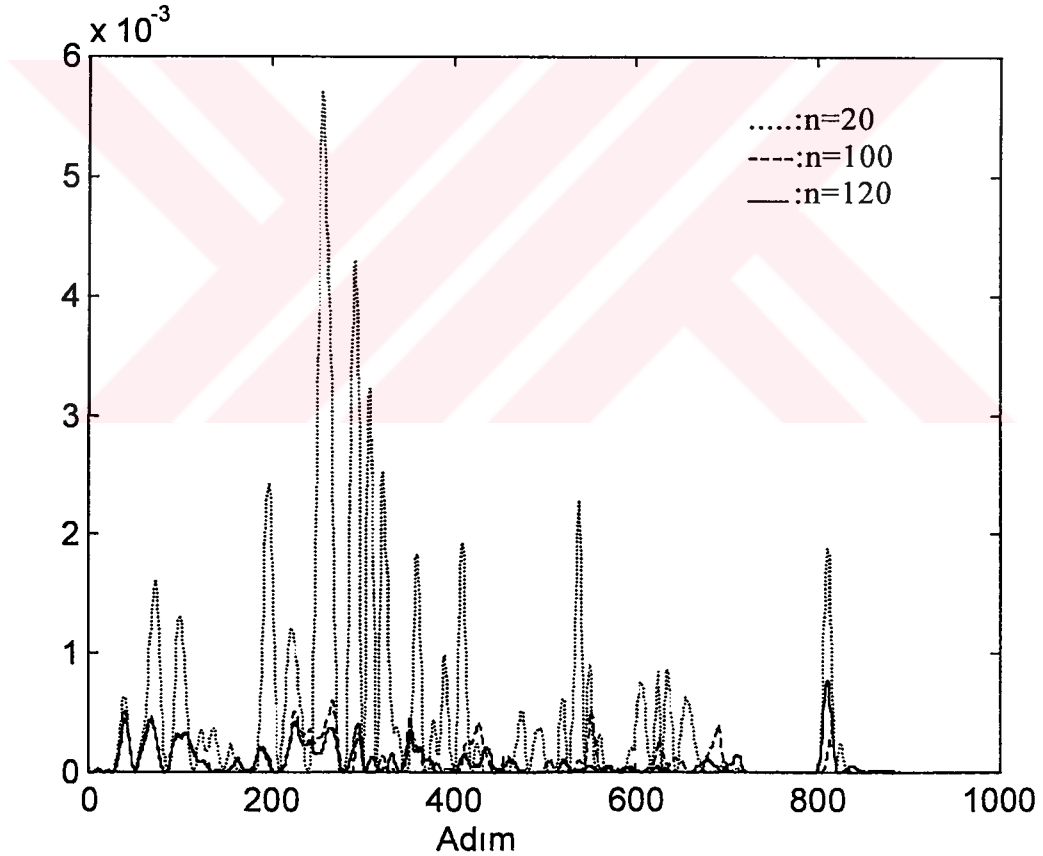
Bu üç μ arasında en iyi performansı veren $\mu=0.08$ için ε^2 aşağıdaki gibidir.



Şekil 4.12 Adaptasyon sürecinde hatanın karesi ε^2

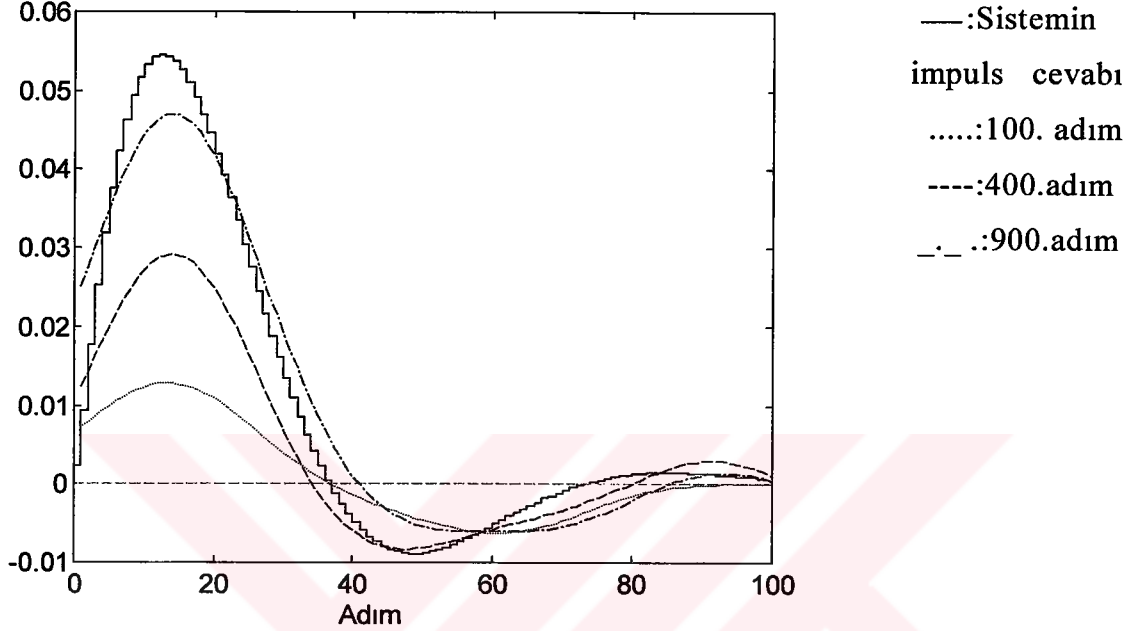
Yukarıdaki Şekil 4.12’de toplam 1000 adımdan oluşan adaptasyon sürecinde ϵ^2 ‘nin LMS algoritması kullanılarak nasıl zamanla minimum olduğu açıkça görülebilir.

Modelleme aşamasında seçilen ağırlık vektörünün uzunluğu, model performansını etkileyen bir diğer faktördür. Şekil 4.13’de değişik uzunluktaki ağırlık vektörleri için ortalama kare hataların ϵ^2 zamanla değişimi verilmiştir. Şekilden de anlaşılabacağı gibi ağırlık eleman sayısı küçük seçildiğinde ϵ^2 hata büyük olurken, eleman sayısı gereğinden fazla uzun seçildiğinde modelin başlangıçta iyi yakınsadığı fakat zamanla kararsızlığa gittiği gözlenmiştir.



Şekil 4.13 Farklı uzunlukta seçilen ağırlık vektörleri için ortalama kare hata ϵ^2 'nin değişimi

Gecikmeli zincirin ağırlıklarının değişimleri belli adımlarda Şekil 4.14’de de gösterildiği gibi kaydedilmiştir ve sistemin Şekil 4.5’de verilen impuls cevabını oluşturduğu gözlemlenmiştir.



Şekil 4.14 İlerleyen adımlarda ağırlıkların değişimi

Şekilden de anlaşıldığı gibi zamanla ağırlıkların değerleri sistemin impuls cevabının formunu almaktadır ve 900. adımda bu daha belli olmaktadır. Başlangıçta bütün ağırlıklar sıfır olarak seçildiği için 100. adımda henüz ağırlık değerleri küçüktür. Buradan şöyle bir önemli sonuç çıkmaktadır.

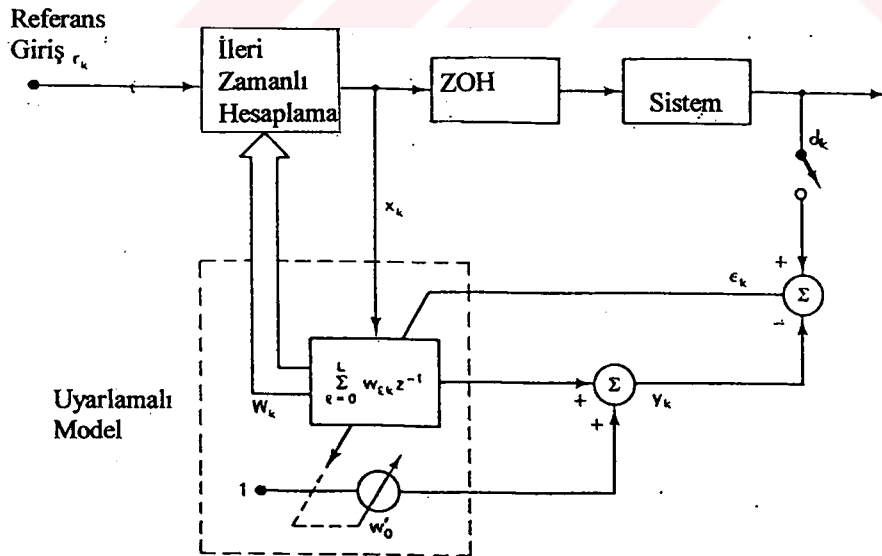
Model ağırlıkları uyarlama prosesi süresince sürekli otomatik olarak ayarlanmaktadır ve zamanla tamamen sistem ile herhangi bir giriş değeri için aynı cevabı verecek yapıya ulaşır bu da uyarlamalı modelleme de ulaşılması istenen noktadır.

BÖLÜM 5

UYARLAMALI MODEL KONTROL

Bir önceki bölümde uyarlamalı modelleme ve bu modelin nasıl kurulduğu detaylı olarak anlatılmıştır. Burada da yapı olarak bir transversal filtre olan uyarlamalı modelin ve sistemin nasıl kontrol edildiği anlatılacaktır.

Uyarlamalı kontrol sisteminin blok diyagramı Şekil 5.1'de verilmiştir. 'İleri zamanlı hesaplama' olarak adlandırılan kutu detaylı olarak anlatılacaktır. Normal uygulamalarda bu kontrol sisteminde gerekli olan veri toplayıcıları ve görüntüleyicileri gibi gerekli cihazlar burada gösterilmemiştir. ZOH sıfırıncı mertebeden tutucu, 0.1 sn. zaman aralıklarıyla örneklenen x_k değerini tutar. Bu her bir zaman aralığında uyarlamalı model güncellenir ve yeni giriş değeri hesaplanır.



Şekil 5.1 Uyarlamalı model kontrol sisteminin blok diyagramı [3]

Uyarlamalı model burada bir önceki bölümde açıklandığı gibi 100 ağırlıktan oluşan bir transversal filtredir. Ön bir ağırlık değeri w_0 , henüz giriş değerleri hesaplanmamışken model çıkışının sistem çıkışına uygun olması için konulmuştur. Uyarlamalı model burada direkt bir model olarak çalışmaktadır. Uyarlamalı proses, ağırlıkları her adımda öyle ayarlar ki verilen giriş sinyalleri için ZOH ve sistemin oluşturduğu kombinasyonunun örneklenmiş çıkışına en uygun ortalama kare hatayı hesaplar. Bunun için Bölüm 3'te detaylı olarak anlatılan LMS algoritması kullanılır.

5.1 İleri Zamanlı Hesaplama

Şekil 5.1'deki sistem uygun bir şekilde çalıştırılırsa sistem referans giriş değeri r_k 'yi takip edecektir. Sistem kontrol değeri x_k , 'ileri zamanlı hesaplama' kutusundan çıkmaktadır. Bu kutu x_k 'yi, referans giriş değeri r_k ve ağırlık vektörü W_k ile giriş vektörü X_k 'dan bulmaktadır [4].

Varsayalım ki uyarlamalı proses ile $E[\epsilon_k^2]$ sifıra çok yakın olarak elde edilmiş olsun. 'İleri zamanlı hesaplama' fonksiyonunun amacı, r_k 'dan öyle bir x_k elde etmektir ki r_k ve y_k eşit olsun. Eğer r_k ve y_k eşit olursa (ϵ_k^2 yeterince küçük olacak şekilde) o zaman sistem çıkışı d_k , r_k 'ya çok yakın olacak ki buda asıl ulaşılması istenen hedeftir. Böylece giriş ve çıkış değerleri birbirlerinin tersi olduğundan 'ileri zamanlı hesaplama' esas olarak uyarlamalı modelin tersi olur. Sistem bir gecikme içerdiğinden, ters model ileriye görmelidir [3].

Ters model şu şekilde oluşturulur. W_k ve w_0 ağırlıkları her bir adımda LMS algoritmasıyla yeniden hesaplanır. Az önce bahsedildiği gibi y_k ve r_k eşit olarak kabul edilirse k. adımı için aşağıdaki eşitlik yazılabilir.

$$y_k = w_0' + \sum_{n=0}^L w_{nk} x_{k-n} \quad (\text{Uyarlamalı Model}) \quad (5.1)$$

Böylece,

$$x_k = 1/w_0 [r_k - w_0' - \sum_{n=1}^L w_{nk} x_{k-n}] \quad (\text{İleri Zamanlı Hesaplama}) \quad (5.2)$$

Bu şekildeki bir ters çevirme işlemi için x_k 'nin sistemi tahrik edebilecek kadar yeterli bir seviyede olduğu varsayılmıştır. Aksi takdir de 'Dither Sinyali' sistem girişine eklenmelidir. Dither yetersiz veya sabit olmayan girişli sistemler için geliştirdiği yöntemde rasgele bir girişi sisteme vererek giriş değeri yeterliliğini arttırır ancak bu yöntemin dezavantajı, dikkatli bir seçim yapılmadığında sistemin gürültü mertebesini arttırmasıdır.

(5.2)'de verildiği gibi x_k bu şekilde seçilirse model çok iyi bir şekilde kontrol edilecek ve aynı giriş değerini gerçek sisteme uygulayarak $E[r_k - d_k]$ 'nin ortalama kare hata $E[\epsilon_k^2]$ 'ye eşit olması sağlanacaktır.

Öncelikle w_{0k} 'nin sıfıra yakın olmadığı kabul edilmek zorunda. Gerçekte sistemde transport ilerlemesi gibi gecikmeler olduğunda w_0 küçük ve gürültülü olur bu da (5.2)'deki x_k 'yi w_0 'la bölme olduğu için çok büyük ve hatalı yapabilir. Bu nedenle transport gecikmesi dikkate alınarak bu durum değiştirilmelidir.

Bu değişiklik, uyarlamalı modelin ilk birkaç ağırlığını sıfır yapmaktan başlar. Sıfır yapılan ağırlıkların sayısı sistemin transport gecikmesi ile ilgili olup önceden elde edilebilir. Örneğin ilk iki ağırlığın w_{0k} ve w_{1k} 'nin sıfır olarak alındığı varsayılırsa o andaki ve daha önceki giriş değerleri x_k , x_{k-1} 'in çıkışta bir etkisi olmayacak fakat x_{k-2} , x_{k-3} ,, x_{k-L} bir etki yapacaktır. Model çıkışının r_k 'ya eşit olmasını sağlayacak giriş değerini seçmek için oluşturulan formüller yeniden düzenlenirse;

$$y_k = w_{0k}' + \sum_{n=2}^L w_{nk} x_{k-n} \quad (5.3)$$

Bu sonuç sayesinde (5.2) deki formülden x_{k-2} hesaplanabilir fakat asıl bilmek istenen x_k 'dir x_{k-2} değildir. Bu nedenle (5.3)'deki ifade gelecek içinde iki birim kaydırılsın.

$$y_{k+2} = w_{0,k+2}' + \sum_{n=2}^L w_{n,k+2} x_{k+2-n} \quad (5.4)$$

Şimdi ağırlıkların yavaş değiştiği kabul edilsin böylece şimdiki ağırlık değerleri gelecekteki ağırlık değerleri yerine kullanılabilir. Sonra yine y_k , r_k 'ya eşit olacak şekilde;

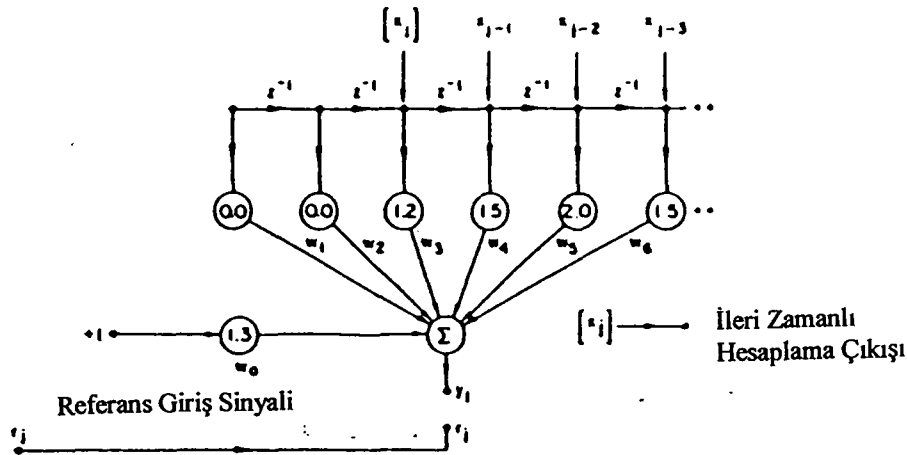
$$x_k = 1/w_{2k} [r_{k+2} - w_{0k}' - \sum_{n=3}^L w_{nk} x_{k+2-n}] \quad (5.5)$$

Bu sonuç, referans giriş değerinin gelecekteki iki birimini bilmeyi gerektirir. Bazen bu değerleri elde etmek mümkündür ve (5.5) ifadesi kontrol için kullanılabilir. Sadece o andaki r_k değeri bilinebiliyorsa (5.5) aşağıdaki gibi düzenlenir.

$$x_k = 1/w_{2k} [r_k - w_{0k}' - \sum_{n=3}^L w_{nk} x_{k+2-n}] \quad (5.6)$$

Yukarıda (5.6) ifadesini kullanmak model çıkışının referans girişini iki birim gecikmeyle takip etmesine neden olur. Her bir adımda, y_k 'nın r_k 'ya eşit olmasını sağlayacak şekilde x_k değeri hesaplanacaktır. Sistem girişi x_k olduğu için, örneklenmiş çıkış d_k , modelin sisteme uymadaki başarısıyla orantılı olarak r_{k-2} 'ye yakınsayacaktır. Cevaptaki bu gecikme, sistemin içindeki transport gecikmesinin kaçınılmaz bir sonucudur.

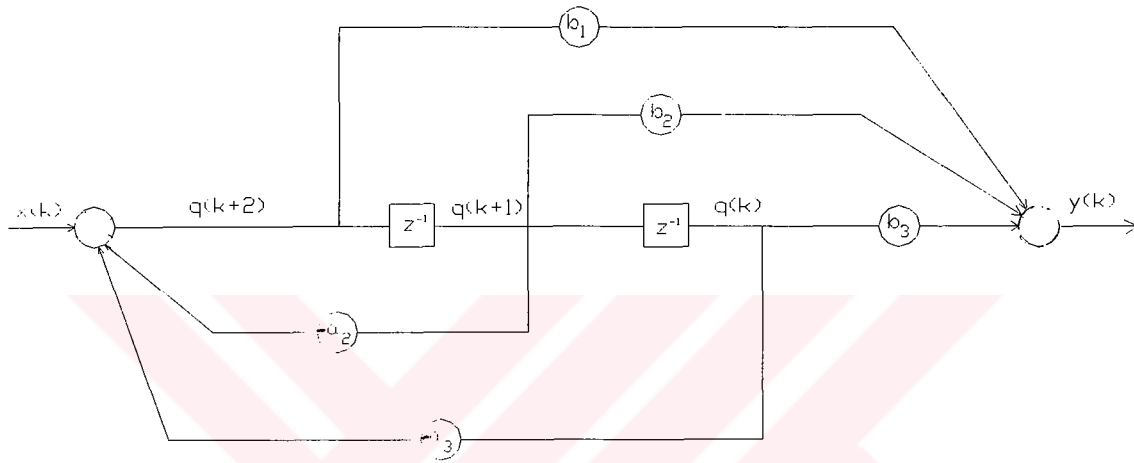
'İleri zamanlı hesaplama' kutusunun detayı Şekil 5.2'de verilmiştir.



Şekil 5.2 'İleri zamanlı hesaplama' kutusunun detayı [4]

5.2 Direkt Programlama

Uyarlamalı model kontrolünün MATLAB ile programının ve matematiksel analizinin yapılabilmesi için sistemin direkt programlama yöntemiyle dijital filtre yapısına getirilmesi gerekir. Bu çalışmada örnek olarak seçilen sistemin direkt programlama yöntemi ile blok diyagram yapısı Şekil 5.3’de verildiği gibi oluşur.



Şekil 5.3 Sistemin, direkt programlama yöntemiyle elde edilen blok diyagramı

Sistemin ayrık zamanlı transfer fonksiyonu $G(z)$ olarak tanımlanırsa aşağıdaki ifade elde edilir.

$$G(z) = N(z) / D(z) = (b_1 z^2 + b_2 z + b_3) / (a_2 z + a_3) \quad (5.7)$$

Bu çalışmada örnek olarak seçilen sistem (4.1) ifadesindeki ikinci mertebe sistem göz önüne alınırsa, ayrık zamanlı transfer fonksiyonunun katsayıları Matlab’de aşağıdaki gibi kolayca elde edilebilir.

```
Num=[1];
Den=[1 1 1];
Ts=0.1;
[Numd,Dend]=c2dm(Num,Den,ts,'tustin');
```

Böylece b_1, b_2, b_3, a_2, a_3 katsayıları aşağıdaki gibi elde edilir.

$$b_1=0.0024 ; b_2=0.0048 ; b_3=0.0024 ; a_2= -1.8955 ; a_3=0.9050 ;$$

Şekil 5.3'deki blok diyagramından aşağıdaki bağıntılar oluşturulur.

$$G(z)= N(z) / D(z) = (b_1z^2+b_2z+b_3) / (a_2z+a_3)$$

$$Y(z) = G(z).X(z)= N(z). X(z) / D(z) \quad (5.8)$$

$$y(k+2)+a_2y(k+1)+a_3y(k)= b_1x(k+2)+b_2x(k+1)+b_3x(k) \quad (5.9)$$

İlk şartlar sıfır olsun ve $Q(z) =X(z) / D(z)$ olarak tanımlansın;

$$x(k)= q(k+2) + a_2q(k+1) + a_3q(k)$$

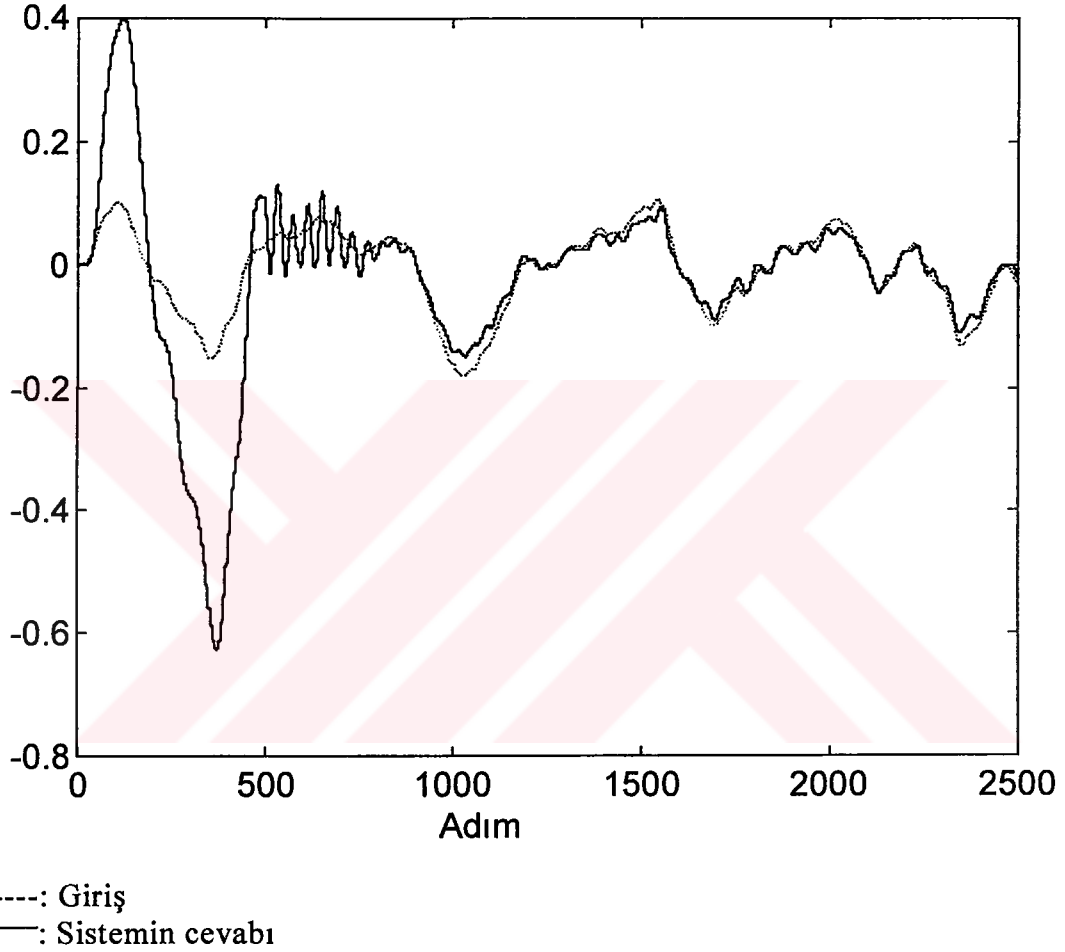
$$y(k)= b_1q(k+2) + b_2q(k+1) + b_3q(k) \quad (5.10)$$

Sonuç olarak sistemin, direkt programlama yöntemiyle yapısı bu şekilde oluşur. Benzer şekilde Matlab'de programı yapıldığında uyarlamalı model kontrolünü gerçekleştirmek için gerekli olan ileri zamanlı hesaplama çalıştırılabilir. Matlab program detayı EkA'da verilmiştir.

5.3 Uyarlamalı Model Kontrol

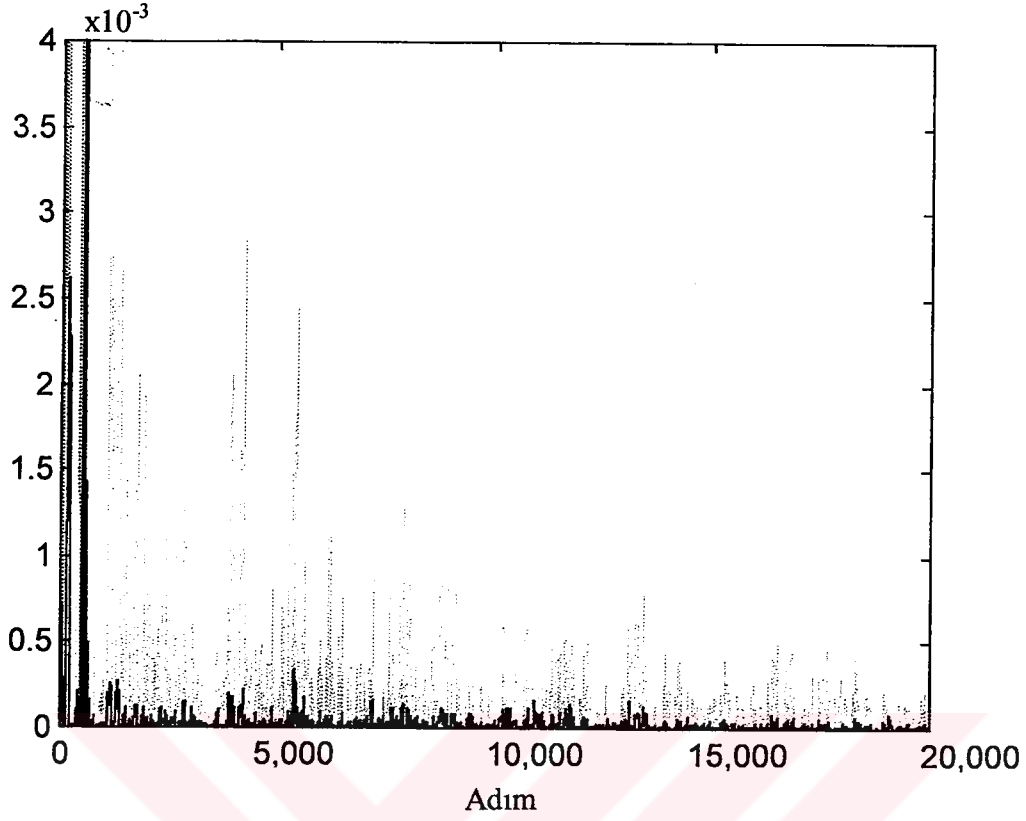
İleri zamanlı hesaplama ve direkt programlama yönteminin detaylı olarak anlatılmasıyla, uyarlamalı model kontrolünü gerçekleştirmek için gerekli alt yapı oluşturulmuştur. Bölüm 4'de uyarlamalı model programı yapılmış ve rasgele giriş değerlerine, sistemin ve modelin verdiği cevaplar incelenmiştir. Burada da sistem direkt programlama yöntemiyle programlanacak, referans giriş değerinden sonra sisteme Şekil 5.1'de gösterildiği gibi 'İleri zamanlı hesaplama' kutusu eklenecektir. Bu şekilde uyarlamalı model kontrolü yapılacaktır. Burada sistemin önüne $G_1(s) = k / (as+b)$ transfer fonksiyonlu bir model eklenmiştir. Bunun nedeni, sistemin rasgele giriş değerlerine karşı verdiği cevap böyle bir kontrol uygulamasında kararsız olabilir. Yukarıdaki bu model, sistem cevabında bir gecikmeye neden olur fakat kararsızlığı ortadan

kaldırır. Öncelikle a ve b katsayıları 1 olarak ve $k=4$ olarak seçildiğinde izleme performansı incelenmek istensin. Şekil 5.4’de verilen grafikte 500. adıma kadar kontrol devreye girmemiş ve Bölüm.4’deki gibi sadece modelleme çalıştırılmış ve 500. adımdan sonra kontrol devreye girmiştir.



Şekil 5.4 Sistem ve uyarlamalı modelin, $k=4$ için rasgele giriş değerlerine cevapları.(500.adımda kontrol başlıyor)

Yukarıdaki şekilde de görüldüğü gibi sistem çıkışı ile referans giriş değeri 500. adıma kadar k kazanç değerinden dolayı çok farklıdır. Kontrol çalışmaya başladığında ise sistem çıkışı, referans girişi iyi bir şekilde takip ettiği gözlenmektedir.



.....: $\mu=0.01$

—: $\mu=0.08$

Şekil 5.5 Farklı μ değerleri için ortalama kare hata ϵ^2 'nin değişimi.

Şekil 5.5.'de iki farklı yakınsama faktörü için ortalama kare hataların değişimi verilmiştir. Yakınsama faktörü değeri μ küçük seçildiğinde hatanın, optimum μ değerine göre nasıl arttığı grafikte anlaşılmaktadır. Bunun yanında kontrol 500. adımda başladığında hatada önceleri büyük değerlere ulaşılmış fakat zamanla, bu yine minimum mertebelerinde seyretmiştir.

BÖLÜM 6

SONUÇ

Bu çalışmada, uyarlamalı model kontrolün genel yapısı ve LMS algoritması ile gerçek sistem ve model arasındaki hatanın minimum edilmesi yöntemine dayalı modelleme tekniği üzerinde durulmuştur. Ele alınan bir örnek sistem için bu teknik kullanılarak MATLAB’de program yazılıp model performansı ile ilgili simülasyonlar sunulmuştur. Modelleme sürecinde karşılaşılan zorluklar, uyarlamalı sistemin performansını etkileyen parametrelerin genel etkileri incelenmiştir. Pratik olarak bilinmeyen sistemlerin kontrolü için sıkça kullanılan uyarlamalı model kontrolünün bu anlamda önemini incelemek için giriş değerleri rasgele seçilmiştir. Uyarlamalı kontrol sistemlerinin öğrenen sistemlere yakınlığından dolayı bu çalışma ayrıca bu yöndeki çalışmalara da ışık tutacağı düşünülmektedir.

KAYNAKLAR

- [1] ASTRÖM,K.J.,WITTENMARK,B.,(1989) Adaptive Control
Addison-Wesley Publishing Company,USA
- [2] OGATA,K., (1990), Modern Control Engineering,
Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [3] WIDROW,B.,STEARNS, S.D.,(1985), Adaptive Signal Processing,
Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [4] WIDROW,B., (1971), Adaptive Model Control Applied To Real-Time
Blood-Pressure Regulation , Journal of Pattern Recognition and
Machine Learning, Plenum Press, pp. 310-324
- [5] WIDROW,B.,WALACH,E., (1996), Adaptive Inverse Control,
Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [6] OGATA,K.,(1987), Discrete-Time Control Systems,
Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [7] ETTER, M.D.,(1997) Engineering Problem Solving with Matlab
Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [8] GRACE A.,LAUB,A.J., (1990) Control System Toolbox for use with
Matlab, The Mathworks,Inc., Natwick
- [9] STEARNS,S.D.,DAVID,R.A.,(1996), Signal Processing Algorithms in
Matlab, Englewood Cliffs,NJ:Prentice-Hall,Inc.
- [10] HIZAL,N.A.,(1998), Improved Adaptive Model Control,
ARI,Springer-Verlag,Kabul edildi.
- [11] HIZAL,N.A.,(1998), Gain Scheduling Adaptive Model Control,
Turkish Journal of Engineering and Enviromental Sciences,
TÜBİTAK, Gönderildi.

EK A

1) Filtre Edilmiş Rasgele Sinyal Üreticisi

```
function y=filtrand(m,wn,Ts)
% y=filtrand(m,wn,Ts)
% m = y'nin uzunluğu
% wn = İkinci dereceden filtrenin doğal frekansı
x=randn(1,m);
y=lsim(1,[1/wn/wn 2*.707/wn 1],x,0:Ts:(m-1)*Ts);
```

2) Uyarlamalı Modelleme

```
w0=0;
num=[1];
den=[1 1 1];
ts=0.1;
u=0.08;
[numd,dend]=c2dm(num,den,ts,'tustin');
%w=dimpulse(numd,dend,100);
w=zeros(1,100);
x=zeros(1,99);
r=filtrand(600,1,0.1);
r=[r; zeros(200,1);0.3*ones(200,1)];
d=dlsim(numd,dend,r);
plot(d,'m');hold on;
```

```
for i=1:1000
x=[r(i) x(1:99)];
y(i)=x*w'+w0;
e=d(i)-y(i);
w=w+(2*u*e*x);
w0=w0+(2*u*e);
end
```

```
plot(y,'c:')
plot(r,'g:')
xlabel('Adım');
hold off
figure(gcf)
%axis([0 200 -inf inf])
```

3) Uyarlamalı Modellemede Farklı μ 'ler için Hata Kareleri

```

w10=0;
w20=0;
w30=0;
num=[1];
den=[1 1 1];
ts=0.1;
u1=0.1;
u2=0.001;
u3=0.18;
[numd,dend]=c2dm(num,den,ts,'tustin');
w1=zeros(1,100);
w2=zeros(1,100);
w3=zeros(1,100);
x=zeros(1,99);
r=filtrand(600,1,0.1);
r=[r; zeros(200,1);0.01*ones(200,1)];
d=dlsim(numd,dend,r);
%plot(y,'m');hold on;
for i=1:1000
    x=[r(i) x(1:99)];
    y1(i)=x*w1'+w10;
    y2(i)=x*w2'+w20;
    y3(i)=x*w3'+w30;
    e1(i)=d(i)-y1(i);
    e2(i)=d(i)-y2(i);
    e3(i)=d(i)-y3(i);
    w1=w1+(2*u1*e1(i)*x);
    w10=w10+(2*u1*e1(i));
    w2=w2+(2*u2*e2(i)*x);
    w20=w20+(2*u2*e2(i));
    w3=w3+(2*u3*e3(i)*x);
    w30=w30+(2*u3*e3(i));
end
plot(e1,'c:');hold on;
plot(e2,'m')
plot(e3,'g');hold off;
xlabel('Adım');
ylabel('E Hata')
figure(gcf)
pause;
axis([0 300 -0.5 0.5])

```

4) Ağırlıkların Zamanla Değişimi

```

w0=0;
num=[1];
den=[1 1 1];
ts=0.1;
u=0.0035;
[numd,dend]=c2dm(num,den,ts,'tustin');
w=zeros(1,100);
x=zeros(1,99);
r=filtrand(600,1,0.1);
r=[r; zeros(200,1);0.3*ones(200,1)];
d=dlsim(numd,dend,r);

for i=1:1000
    x=[r(i) x(1:99)];
    y(i)=x*w'+w0;
    e=d(i)-y(i);
    w=w+(2*u*e*x);
    w0=w0+(2*u*e);
    if i==100
        plot(w,'c:');hold on;dimpulse(numd,dend,100);
        end;
    if i==400
        plot(w,'r:');hold on;
        end;
    if i==900
        plot(w,'m:');hold on;
        end;
end

xlabel('Adim');
ylabel('Ağırlıklar')
hold off
figure(gcf)

```

5) Uyarlamalı Model Kontrol

```

w0=0;
num=[8];
den=[1 1 1];
num1=[1];
den1=[1 1];
ts=0.1;
u=0.08;
n=2;
q=zeros(1,n+1);
w=0.01*ones(1,100);
%w=dimpulse(numd,dend,100)
w=[0 w(1:99)];
x=zeros(1,100);
[numd,dend]=c2dm(num,den,ts,'tustin');
[num1d,den1d]=c2dm(num1,den1,ts,'tustin');
r=filtrand(9502,0.15,0.1);
r=[r; zeros(200,1);0.2*ones(200,1)];

for i=1:9900;
    p=r(i);
    if i>1800
        m=(r(i)-w0-x(2:99)*w(3:100)')/w(2);
        p=dlsim(num1d,den1d,m);
    end;
    x=[p x(1:99)];
    q(1)=p-(dend(2)*q(2)+dend(3)*q(3));
    d(i)=numd(1)*q(1)+numd(2)*q(2)+numd(3)*q(3);
    q=[0 q(1:n)];
    y(i)=x*w'+w0;
    j=y(i);
    e=d(i)-y(i);
    w=w+(2*u*e*x);
    w0=w0+(2*u*e);
end;

plot(r,'g');hold on;
plot(d,'m');hold on;plot(y,'c:');
figure(gcf)
pause;
axis([8000 9900 -0.2 0.2])

```

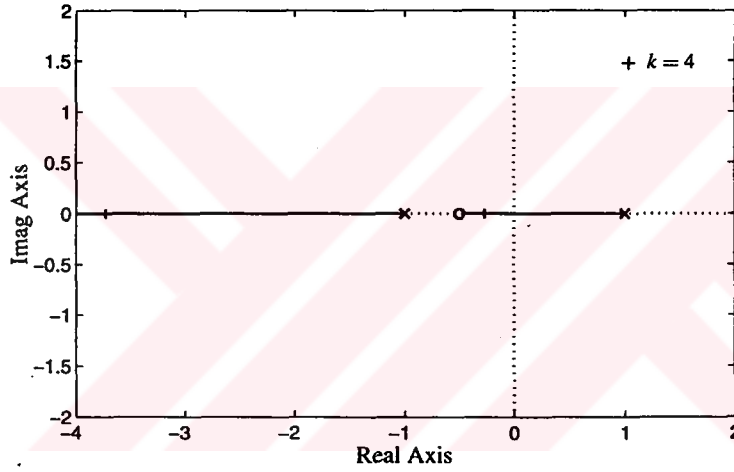
EK B

Uyarlamalı Model ve Kontrol Örnekleri

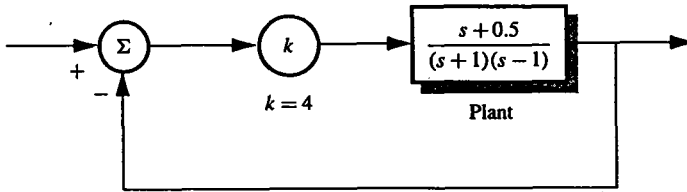
Kontrol edilecek sistemin transfer fonksiyonu;

$$\frac{s + 0.5}{(s + 1)(s - 1)}$$

Yukarıdaki sistem kararsız bir sistem olup öncelikle bir geri besleme ile kararlı hale getirilir. Köklerin geometrik yeri Şekil B.1’de verilmiştir.

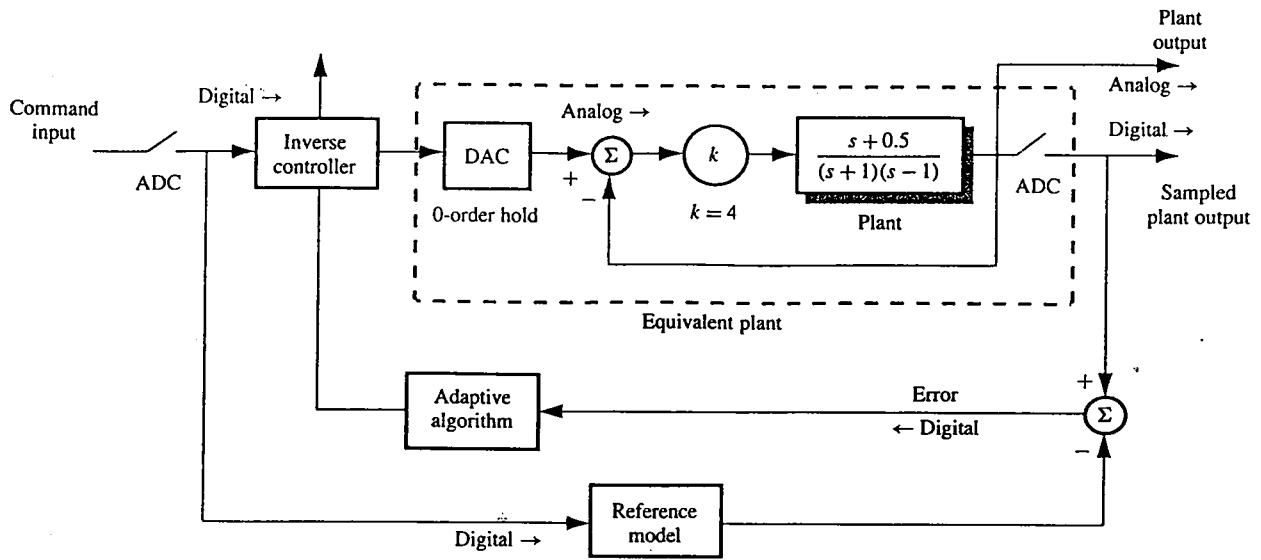


Şekil B.1 Oransal geri beslemeli sistemin köklerinin geometrik yeri [5].

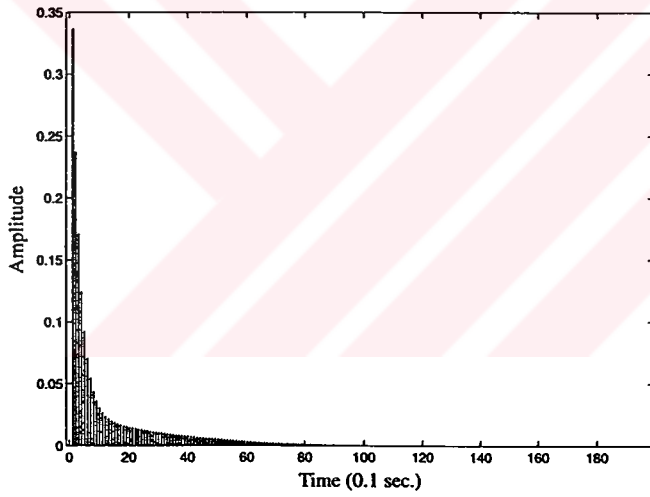


Şekil B.2 Oransal geri besleme ile kararlı hale getirilmiş sistem [5].

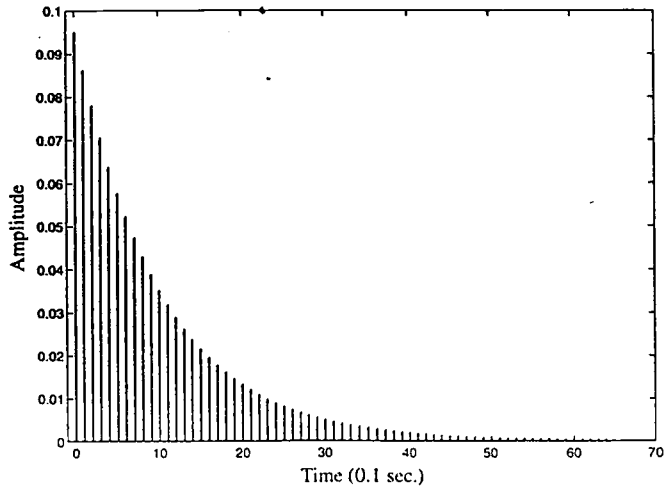
Bu örnekte ağırlık vektörü daha öncede olduğu gibi 100 elemanlı olarak seçilmiştir. Sistemin modellenmesi ve yapılan simülasyonlar adım adım verilmiştir.



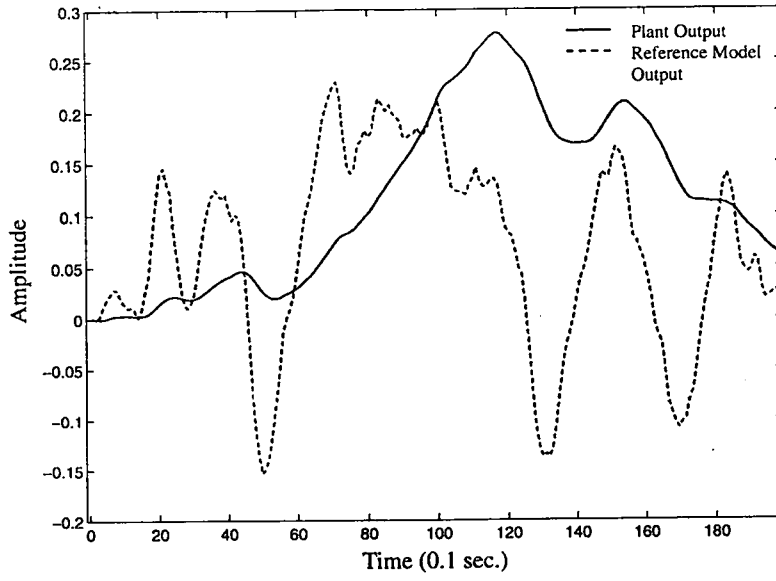
Şekil B.3 Sistemin uyarlamalı ters model kontrol diyagramı [5].



Şekil B.4 Ayırık zamanlı sistemin impuls cevabı [5].

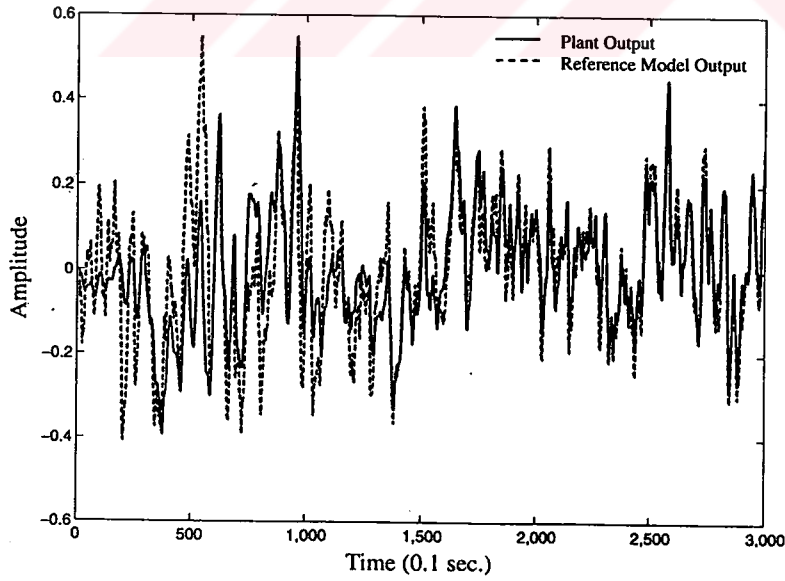


Şekil B.5 Modelin impuls cevabı [5].



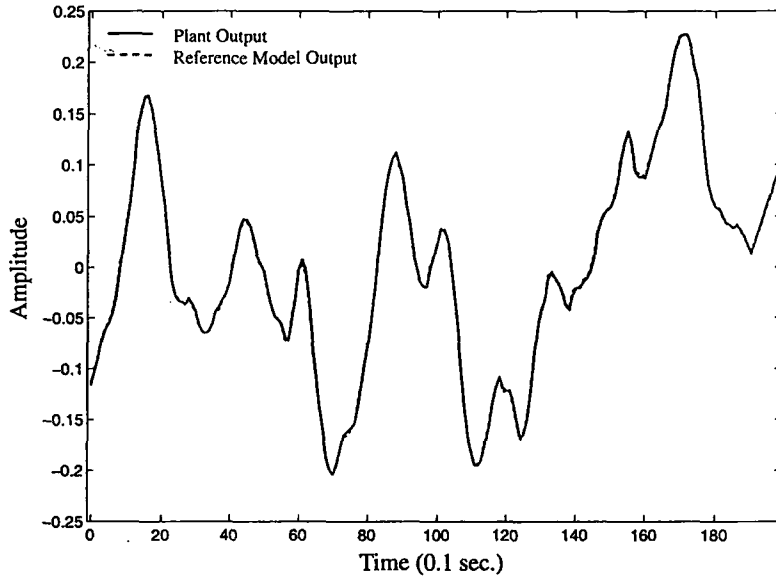
— : Sistemin cevabı
 : Modelin cevabı

Şekil B.6 İlk 200 adımda sistemin ve modelin cevapları [5].



— : Sistemin cevabı
 : Modelin cevabı

Şekil B.7 3000 adım boyunca sistemin ve modelin cevapları [5].



— : Sistemin cevabı
 : Modelin cevabı

Şekil B.8 100,000 adımın son 200 adımında sistemin ve modelin cevapları [5].

ÖZGEÇMİŞ

10.06.1972 tarihinde Sivas'ta doğdu. Lise öğrenimini İstanbul Atatürk Fen Lisesinde yaptı. 1989 yılında İTÜ Makina Fakültesi Makina Mühendisliği Bölümüne girdi. 1993'de Makina Mühendisi ünvanını aldı. Aynı yıl İTÜ Fen Bilimleri Enstitüsünün Makina Anabilim dalının Robotik programında yüksek lisans öğrenimine başladı.

