**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

# BULANIK MODEL ÖNGÖRÜLÜ KONTROL

**YÜKSEK LİSANS TEZİ**
**Ezgi KÖMÜRCÜ**
**(504061114)**

**Tezin Enstitüye Verildiği Tarih : 04 Mayıs 2009**
**Tezin Savunulduğu Tarih : 05 Haziran 2009**

**Tez Danışmanı : Prof. Dr. Müjde GÜZELKAYA (İTÜ)**
**Diğer Jüri Üyeleri : Prof. Dr. İbrahim EKSİN (İTÜ)**
**Prof. Dr. Serhat ŞEKER (İTÜ)**

**HAZİRAN 2009**

## ÖNSÖZ

Yüksek lisans tez çalışması sırasında birlikte çalıştığımız ve her konuda yardımını esirgemeyen değerli danışman hocam Prof. Dr. Müjde Güzelkaya' ya, değerli hocam Prof. Dr. İbrahim Eksin' e, değerli arkadaşım Ar. Gör. Tufan Kumbasar' a, verdiği destek için TÜBİTAK-BİDEB' e, hayatım boyunca beni her zaman destekleyen aileme ve değerli arkadaşım Barış Parlak' a sonsuz teşekkürlerimi sunarım.

Mayıs 2009

Ezgi KÖMÜRCÜ

Elektrik Elektronik Mühendisi

**İÇİNDEKİLER**

**Sayfa**

**ÖNSÖZ** ..... **iii**
**İÇİNDEKİLER** ..... **v**
**KISALTMALAR** ..... **vii**
**ŞEKİL LİSTESİ** ..... **ix**
**ÖZET** ..... **xi**
**SUMMARY** ..... **xiii**
**1. GİRİŞ** ..... **1**
1.1 MPC Özellikleri ..... 1
1.2 Doğrusal Olmayan Model Öngörülü Kontrol ..... 2
**2. BULANIK MODELLEME** ..... **5**
**3. MODEL ÖNGÖRÜLÜ KONTROL (MPC)** ..... **9**
3.1 Gerileyen Ufuk Yapısı ..... 10
3.2 En İyi Girişlerin Hesaplanması ..... 13
3.3 Durum Uzayı Modeli ..... 18
3.4 Temel Denklemler ..... 19
3.4.1 Öngörülü kontrol problemlerinin çözümü ..... 20
**4. BULANIK MODEL ÖNGÖRÜLÜ KONTROL (FMPC)** ..... **27**
**5. BULANIK MODEL ÖNGÖRÜLÜ KONTROL UYGULAMALARI** ..... **31**
5.1 Benzetim Çalışmaları ..... 31
5.1.1 Doğrusal olmayan sistem bulanık modellemesi ..... 31
5.1.2 Model öngörülü kontrolör tasarımı ..... 34
5.1.3 Model öngörülü kontrolörlerin bulanık kaynaşmaları ..... 36
5.2 Gerçek Zaman Uygulaması ..... 43
**6. SONUÇ VE ÖNERİLER** ..... **53**
**KAYNAKLAR** ..... **55**
**EKLER** ..... **57**

## KISALTMALAR

**DMC** **:** Dinamik Matris Kontrolü
**FMPC** **:** Bulanık Model Öngörülü Kontrol
**GPC** **:** Genelleştirilmiş Öngörülü Kontrol
**IMC** **:** İçsel Model Kontrolü
**MAC** **:** Model Algoritmik Kontrol
**MBPC** **:** Modele Dayalı Öngörülü Kontrol
**MPC** **:** Model Öngörülü Kontrol
**NMPC** **:** Doğrusal Olmayan Model Öngörülü Kontrol
**PFC** **:** Öngörülü Fonksiyonel Kontrol
**PI** **:** Oransal ve Entegral Kontrolü
**PID** **:** Oransal, Entegral ve Türev Kontrolü
**SISO** : Tek giriş tek çıkış
**TS** **:** Takagi Sugeno

## ŞEKİL LİSTESİ

**Sayfa**

**Şekil 3.1 :** Model öngörülü kontrol şeması. ..... 10
**Şekil 3.2 :** Öngörülü kontrol: temel düşünce. ..... 12
**Şekil 4.1:** FMPC kontrolör yapısı. ..... 29
**Şekil 5.1 :** Alt sistem birim basamak cevapları. ..... 32
**Şekil 5.2 :** Bulanık kümeler ve tanımladıkları üyelik fonksiyonları. ..... 33
**Şekil 5.3 :** Doğrusal olmayan sistem modeli birim basamak cevabı. ..... 34
**Şekil 5.4 :** MPC sistemine eklenen IMC yapısı. ..... 35
**Şekil 5.5 :** Alt sistemlerin MPC ile kontrol edilmesi ..... 36
**Şekil 5.6 :** Bulanık MPC ile sistem kontrolü blok şeması ..... 36
**Şekil 5.7 :** FMPC ile doğrusal olmayan sistem kontrolü. ..... 37
**Şekil 5.8 :** FMPC ve MPC ile doğrusal olmayan sistem kontrolü. ..... 39
**Şekil 5.9 :** Alt sistemlerin PI kontrolörlerle kontrol edilmesi. ..... 34
**Şekil 5.10 :** PI kontrolörlerin kaynaşması ile doğrusal olmayan sistem kontrolü blok şeması . ..... 40
**Şekil 5.11 :** FMPC ve PI kontrolörler ile doğrusal olmayan sistem kontrolü. ..... 40
**Şekil 5.12 :** İki PI kontrolör kaynaşması ve tek PI kontrolör ile doğrusal olmayan sistem kontrolü ..... 41
**Şekil 5.13 :** FMPC ile doğrusal olmayan sistem kontrolü (giriş bozucusu etkisi). ..... 42
**Şekil 5.14 :** FMPC ile doğrusal olmayan sistem kontrolü (çıkış bozucusu etkisi) ..... 42
**Şekil 5.15 :** Ayar değeri 4.5 V (yaklaşık $38^{o}$ C) için ısıl sistem ve matematiksel model cevabı. ..... 43
**Şekil 5.16 :** Ayar değeri 7 V (yaklaşık $50^{o}$ C) için ısıl sistem ve matematiksel model cevabı. ..... 44
**Şekil 5.17 :** Isıl sistem üyelik fonksiyonları. ..... 44
**Şekil 5.18 :** Bulanık kaynaşma ile modellenen ısıl sistem basamak cevabı. ..... 45
**Şekil 5.19 :** Isıl sistem modeli FMPC kontrolü blok şeması. ..... 46
**Şekil 5.20 :** Isıl sistem modeli FMPC kontrolü. ..... 46
**Şekil 5.21 :** Isıl sistem FMPC kontrolü blok şeması. ..... 47
**Şekil 5.22 :** Isıl sistem FMPC kontrolü. ..... 47
**Şekil 5.23 :** Isıl sistem FMPC kontrolü (çıkış bozucusu etkisi). ..... 48
**Şekil 5.24 :** Isıl sistem FMPC kontrolü (giriş bozucusu etkisi). ..... 49
**Şekil 5.25 :** Isıl sistem FMPC ve PID kontrolü ..... 50
**Şekil 5.26 :** Isıl sistem FMPC ve PID kontrolü (çıkış bozucusu etkisi). ..... 51
**Şekil 5.27 :** Isıl sistem FMPC ve PID kontrolü (giriş bozucusu etkisi). ..... 51

## BULANIK MODEL ÖNGÖRÜLÜ KONTROL

## ÖZET

Son yıllarda özellikle kimya ve petrokimya endüstrilerinde, klasik kontrol yöntemlerinden farklı olarak Model Öngörülü Kontrol algoritmaları kullanılmaya başlanmıştır. Model Öngörülü Kontrol pek çok yöntemin aksine, teorik çalışmalardan çok önce endüstride kullanılmış, bu sebeple de pratikte mevcut olan pek çok probleme cevap aramıştır.

Model Öngörülü Kontrol, sıkça karşılaşılan uzun ölü zaman problemini ortadan kaldırabilmekte, çok değişkenli kontrol problemlerini ele alabilmekte ve kısıtlara yakın çalışarak, endüstriyel süreçler için çok önemli olan karlı çalışma ortamını yaratabilmektedir. Tüm bu çözümleri sunarken, kolay uygulanabilir bir yöntem olması, önerilen diğer kontrol yöntemlerine göre daha çok tercih edilmesine sebep olmuştur.

Doğrusal olmayan sistemlerin kontrolünde doğrusal bir Model Öngörülü Kontrol yapısı, istenen kontrol işlemini gerçekleştiremeyebilir. Bu sebeple literatürde doğrusal olmayan Model Öngörülü Kontrol yapıları geliştirilmiştir. Ancak bu yöntemler çok fazla hesap yükü getirmesinden dolayı endüstride kolay uygulanabilir değildir.

Doğrusal olmayan sistemlerin modellenmesi ile ilgili problemlere cevap arayan yöntemler, hesap yükünü azaltmak amacıyla, doğrusal olmayan süreçleri doğrusal alt modeller kümesi ile tanımlamış ve kontrolör tasarımını kolaylaştıran algoritmalar üretmişlerdir. Bu çalışmada bu yöntemlere alternatif bir yaklaşım olan Bulanık Model Öngörülü Kontrol yapısı önerilmiştir.

Bulanık Model Öngörülü Kontrol yaklaşımı, doğrusal olmayan bir süreci, karakteristiğine uygun doğrusal veya doğrusal-benzeri alt sistemlerin bulanık olarak kaynaşması biçiminde modellemektedir. Bulunan her alt sisteme göre doğrusal Model Öngörülü Kontrolörler tasarlayarak, modelleme için kullandığı bulanık kaynaştırmayı kontrol işareti için de uygular. Böylece doğrusal olmayan bir sistemin kontrolünü doğrusal Model Öngörülü Kontrolörler ile gerçekleştirip, hesap yükünü azaltır.

Bu çalışmada, bahsedilen Bulanık Model Öngörülü Kontrol algoritması açıklanmıştır. Önerilen yöntem, MATLAB ortamında doğrusal olmayan bir sisteme uygulanmış ve sonuçlar verilmiştir. Yöntemin giriş ve çıkış bozucuları karşısında verdiği tepkiler gözlenmiştir. Ayrıca klasik PI kontrolörlerin bulanık kaynaşması ile elde edilen bir sistem de Bulanık Model Öngörülü Kontrol yapısı ile karşılaştırılmıştır.

Çalışmada ayrıca PT326 sıcaklık deney setinin sistem davranışı gözlenmiş, ve önerilen Bulanık Model Öngörülü Kontrol yaklaşımı ile gerçek zamanlı kontrolü yapılmıştır. Gerçek zamanlı uygulamanın giriş ve çıkış bozucularına verdiği tepkiler gözlenmiş ve klasik PID kontrolör davranışı ile karşılaştırılmıştır.

## FUZZY MODEL PREDICTIVE CONTROL

## SUMMARY

During the past decade especially in chemical and petrochemical industries Model Predictive Control algorithms are widely used besides regular control methods. Different from the development of other methods, Model Predictive Control is firstly used in industry practically before the theoretical knowledge has developed. So this method tried to find answers for the problems encountered in practice.

Model Predictive Control overcomes the widely faced long dead time problems, handles multivariable control problems naturally, and allows operation closer to constraints, which leads to more profitable operation. Besides solving these problems, it is practically easier to be applied so that Model Predictive Control is preferred more than other suggested methods.

A highly nonlinear system controlled by a linear Model Predictive Controller may not give a satisfactory control solution. To solve this problem, Nonlinear Model Predictive Control methods are proposed. But it is not easy to apply these methods practically since the nonlinearity increases the computational effort.

In literature, in order to solve modelling problems, nonlinear process is defined by a set of linear submodels and algorithms decreasing the computational effort to design controllers are proposed. In this study, an alternative approach to these methods, Fuzzy Model Predictive Control approach is proposed.

According to the Fuzzy Model Predictive Control approach, a nonlinear process is described by a fuzzy convolution model that consists of a number of linear or quasi-linear subsystems. For each subsystem, a Model Predictive Controller is designed and the control effort is fuzzily merged by using the same rules used to merge subsystems. Since the approach controls nonlinear processes by using linear Model Predictive Controllers, the computational effort is decreased.

In this study the Fuzzy Model Predictive Control approach is analyzed and the algorithms are explained. The proposed approach is applied to a nonlinear system and simulated in MATLAB. The response of the system to input and output disturbances are investigated and results are given. Also a system obtained by fuzzily merged PI controllers is compared with Fuzzy Model Predictive Control approach.

In addition, the behavior of PT326 temperature experimental set is observed and Fuzzy Model Predictive Control approach is applied in real time on MATLAB. The reactions to input and output disturbance is observed and the approach is compared with a PID controller.

# 1. GİRİŞ

Modele dayalı öngörülü kontrol (MBPC) veya diğer bir adıyla model öngörülü kontrol (MPC), son yıllarda gelişen ve kontrol uygulamalarında sıkça rastlanmaya başlayan bir kontrol yöntemidir. MPC, anlık kontrol işaretlerini her örnekleme anında çevrimiçi olarak elde eder. Sistemin o anki durum değerlerini başlangıç durum değerleri olarak kabul edip, sonlu ufuklu eniyileme kontrol problemini çözer. Bulunan en iyi kontrol dizisini sisteme uygular ve bir sonraki örnekleme anında bu işlemleri tekrarlayıp kontrol işlemini gerçekleştirir [1].

## 1.1 MPC Özellikleri

Endüstriyel kontrol mühendisliği üzerinde kayda değer etkiye sahip bilinen tek gelişmiş kontrol metodolojisi öngörülü kontroldür [2]. Pek çok teknik buluşun aksine, teorik özelliklerinin anlaşılmasından çok önce, MPC ilk olarak endüstride gerçeklenmiştir. Seksenlerin ortalarına doğru, özellikle Shell'in düzenlediği iki seminerden sonra, MPC' ye duyulan akademik ilgi artmaya başlamıştır. Akademik araştırmalar sonucu MPC özelliklerinin anlaşılmasıyla birlikte, pratik ve teorik uygulamalar için güçlü bir kavramsal çatı oluşmuştur [3]. Çoğunlukla petrokimya endüstrisinde uygulanmışsa da, günümüzde, artan bir hızla, diğer süreç kontrolü alanlarında da uygulanmaktadır. MPC' nin uygulandığı alanlarda başarılı olmasının esas sebepleri şu şekilde sıralanabilir:

1. Çok değişkenli, kontrol problemlerini ele alabilmektedir.
2. Klasik kontrol ile karşılaştırıldığında, kısıtlara yakın çalışmaya izin vermektedir, ki bu genelde daha karlı çalışma anlamına gelmektedir.
3. Bu uygulamalarda kontrol güncelleme oranı görece olarak daha düşüktür. Böylece çevrimiçi hesaplamalar için gerekli zaman bulunabilmektedir [2].
4. MPC, basit dinamiği olan sistemleri kontrol edebildiği gibi, kararsız veya uzun ölü zamanlı sistemler gibi daha karmaşık süreçlere de uygulanabilmektedir.

5. Elde edilen MPC yapısı, kolayca uygulanabilen doğrusal bir kontrol kuralıdır [4].

Oluşabilecek kısıtlamaları hesaba katabilmenin pek çok önemli getirisi vardır. MPC' nin süreç kontrollerinde başarılı olmasını sağlayan ve en çok üzerinde durulan getirisi, en karlı işlemin genelde, süreç bir veya birden fazla kısıtlamada çalışıyorken elde edilmesidir. Genelde bu kısıtlar doğrudan maliyetle, çoğunlukla da enerji maliyetleriyle ilişkilidir [2].

MPC' nin yukarıda sıralanan tercih sebepleri yanı sıra bazı dezavantajları da bulunmaktadır. Kontrol kuralı doğrusal ve kolay hesaplanabilir bir yapıdadır ancak kontrol kuralının elde edilmesi, klasik PID kontrolöründen çok daha karmaşıktır. Kısıtlar göz önüne alındığında, hesaplama gereksinimi klasik sistemlere göre çok daha fazladır. Mevcut bilgisayarların işlemci hızlarının yüksek olması bu sorunu büyük ölçüde ortadan kaldırmıştır. Ancak MPC' nin en büyük dezavantajı, şüphesiz ki bir model önbilgisine bağımlılığı ve gerçek sistem ile elde edilen model arasındaki farklardan etkilenmeye açık olmasıdır [4]. Bu sorun bir içsel model kontrolör (IMC) yapısı ile ortadan kaldırılabilir.

İçsel model ifadesi, gerçek hayatta mevcut sistemin birebir aynısı değil ancak bir yaklaşık modelinin elde edilebildiğini söyler. Doğru model bulunmuş olsa bile, süreç değişkenlerinin doğru ölçümlerinin yapılamayacağını, dolayısıyla ideal bir model elde edilemeyeceğini vurgular. Bu problemleri çözmek adına IMC, sürece göre bir kontrolör yapısı bulmak yerine, süreç ve ideal olmayan modelin farkını alır ve bu değeri girilen referans değere getirmeye çalışır. Bu yapı MPC yapısına eklenecek olursa, MPC' nin gerçek sistem ve model arasındaki farklardan etkilenme durumu ortadan kalkmış olacaktır [5].

### 1.2 Doğrusal Olmayan Model Öngörülü Kontrol

Pek çok süreç çalışma noktası civarında kararlı haldedir ve doğrusal olarak ifade edilebilir. Ancak bazı süreçlerde doğrusal olmayan yapılar ortaya çıkar ve bu durum kritik bir hale dönüşerek kapalı çevrim kararlılığı etkileyen ve doğrusal olarak ifade edilemeyen bir biçime dönüşür. Yine bazı süreçlerde sistemin geçici durumu çalışma noktasına gelinceye kadar oldukça uzun bir zaman alabilir veya harmanlama süreçlerindeki gibi hiçbir zaman kararlı hale gelmeyebilir.

Böyle süreçlerde, doğrusal bir kontrol kuralı, sistemi istenen yapıya getirebilecek nitelikte olmaz. Bu nedenle kararlı bir sistem davranışı için doğrusal olmayan bir kontrolör (NMPC) yapısına ihtiyaç duyulur.

MPC algoritması içinde, doğrusal olmayan modelleri kontrol edebilecek bir yapı bulunmamaktadır. Ancak MPC genel tanımına eklenen bazı yöntemlerle, doğrusal olmayan sistemler kontrol edilmeye çalışılmaktadır. Böylece farklı doğrusal olmayan MPC (NMPC) yaklaşımları önerilmiştir. NMPC' lerin halen çözmeye çalıştıkları bazı problemler mevcuttur:

- Doğrusal olmayan sistemlerin tanımlama yöntemlerinin yetersizliğine bağlı olarak modellenememesi,
- Doğrusal olmayan süreçlerin MPC kontrolü için gerekli olan hesaplama yükünün fazlalığı,
- Doğrusal olmayan sistemlerin kararlılık ve dayanıklılık sonuçlarının azlığı [4].

Literatürde bu problemlere çözüm arayan pek çok çalışma mevcuttur. Doğrusal olmayan sistemlerin kararlılık problemlerine [6] ve dayanıklılık problemlerine çözüm bulan çalışmalar [7] yapılmıştır. Ancak bu yöntemler sadece kapalı çevrim kararlı olan sistemlere uygulanabilmektedir.

Modelleme ile ilgili karşılaşılan sorunlara çözüm bulmak amacıyla araştırmacılar farklı doğrusal olmayan modelleri kullanarak NMPC algoritmaları geliştirmişlerdir.Bu modeller genellikle birinci mertebeden sistemlere yakınsanmıştır ve pek çok endüstriyel süreç için geliştirilmesi çok zordur. Bunun yanısıra doğrusal olmayan sistem modelini kullanan bir NMPC yapısı, eniyileme problemini çözmek için çok fazla hesap yüküne sahiptir ve çevrimiçi uygulamalarda problem yaratmaktadır. Eğer doğrusal olmayan süreç bir doğrusal alt modeller kümesi ile tanımlanabilirse, model öngörülü kontrolör tasarımı ciddi anlamda kolaylaşacaktır [11].

Referans [8] ' te farklı çalışma koşullarına sahip süreçlerin kontrolü ile ilgili olarak süreç farklı çalışma rejimlerine ayrılmıştır. Ve her rejim için basit bir lokal durum uzayı modeli çıkarılmıştır. Daha sonra bu modeller bir interpolasyon yöntemi ile

birleştirilmiş ve global bir model elde edilmiştir. Bu model yapısına göre de MPC tasarlanmıştır.

Referans [9] de, referans [8]' ten farklı olarak bulanık mantık tabanlı bir yöntem uygulanmış ve doğrusal olmayan süreci doğrusal alt sistemlere ayrılmıştır. Daha sonra her alt model için bulanık mantık kullanılarak yapay bir model elde edilmiştir. Bu model kural tabanlı bir bulanık çıkarsamadır. Dolayısıyla model öngörülü kontrolör tasarımını basitleştirmek adına bir fırsat yakalanmıştır.

Referans [10] Takagi-Sugeno tip bir model kullanarak MPC algoritması geliştirmiştir. Ancak algoritmanın sürekli model güncellemesi sebebiyle kontrolör parametrelerini ayarlamakta ciddi sorunlar yaşanmıştır.

Bu çalışmada yüksek dereceli doğrusal olmayan süreçler için tasarlanan bir bulanık model öngörülü kontrolör yaklaşımından bahsedilecektir. Huang, Lou, Gong ve Edgar' ın önermiş olduğu bulanık MPC (FMPC) yaklaşımı kullanılacaktır [11]. Bu yaklaşıma göre, doğrusal olmayan sistem modeline Takagi-Sugeno modelleme metodolojisi uygulanarak bulanık model elde edilmektedir [11]. Elde edilen model ile hiyerarşik bir kontrol tasarım yaklaşımı tanıtılmaktadır. Bu çalışmada referans [11]' de önerildiği gibi doğrusal olmayan sistem modeli, bulanık doğrusal-benzeri alt sistemlerin kaynaşması şeklinde tanımlanmaktadır. Ancak kontrol yöntemi olarak, makalenin önerdiği hiyerarşik yapı yerine, tasarım yönteminin anafikrinden yola çıkılarak her alt sistem modelinin, kendine uygun MPC yapısı ile kontrol edildiği ve kontrol çıkışlarının, alt sistemlerin bulanık kaynaşma yöntemi ile kaynaştırıldığı bir kontrolör yapısı uygulanacaktır.

Bu çalışmada bulanık modelleme, MPC' nin teorik yapısı ve FMPC yaklaşımı açıklanmaktadır. Daha sonra FMPC yöntemi ile yapılan benzetim çalışmaları ve son olarak da ısıl sistem gerçek zamanlı uygulamasından bahsedilmiştir.

## 2. BULANIK MODELLEME

Bulanık mantık, bir uzman bilgisine dayanarak ifade edilmiş dilsel değişkenleri, dilsel değerleri ve dilsel kuralları nicelendirebilmek üzere kullanılan bir yöntemdir [12].

Bulanık mantık, karmaşık sistemlerin modellenmesinde sıkça kullanılmaktadır. Çıkarım kurallarının yapısına bağlı olarak, üç farklı bulanık sistem modeli bulunmaktadır.

- Dilsel bulanık modeller
- Bulanık ilintili modeller
- Takagi – Sugeno (TS) bulanık modeller [13].

Yukarıda verilen kategorilerden ilk ikisi Mamdani bulanık modelleri olarak da bilinmektedir.

Bu çalışmada doğrusal olmayan bir sisteme TS bulanık modelleme yöntemi uygulanmaktadır. Buna göre TS bulanık modelleme yapısı aşağıdaki gibi tanımlanabilir:

- İlk olarak giriş uzayı, alt uzaylara ayrıştırılır.
- Daha sonra her alt uzayda (bulanık bölgede) sistem modeli daha basit olan doğrusal veya doğrusal benzeri modellere yakınsanır.
- Daha basit hale dönüşen bu lokal modeller, klasik kontrol yöntemleri veya bu çalışmada anlatılacağı üzere MPC yöntemi ile kontrol edilebilir.
- Son olarak, alt sistem modelleri kural tabanının gerektirdiği şekilde ağırlıklı ortalama ile kaynaştırılarak, genel bulanık model yapısı oluşturulur [9].

Yukarıda izlenilen yol, referans [11]' de Bulanık Bükülme (Convolution) Modeli olarak ifade edilmektedir. Buna göre doğrusal olmayan bir sistem *p* adet alt sisteme ayrıştırılır ve her alt sistemin davranışı doğrusal veya doğrusal-benzeri olmalıdır. TS' nun modelleme yöntemi gereği, her alt istem için bir bulanık doğrusal-benzeri model

tanımlanmalıdır [9]. Bu tanımlamalar bulanık çıkarsama (fuzzy implication) olarak adlandırılır ve alt bölgelerin içinde ( $R^i$ , $i = 1,2,..,p$ )tanımlanır. Bulanık alt bölgeler $R^i$ ile gösterilir. Bu şekilde bir modelin neden sonuç ilişkisi, kontrol işareti *u*, çıkış *y* ve örnekleme anı *n* cinsinden ayrık zaman fark denklemi ile ifade edilir. Her bulanık bölge aşağıdaki gibi bir kartezyen çarpım ile gösterilir [11].

$$R^i = y(n)\times y(n-1)\times...\times y(n-m+1)\times u(n)\times u(n-1)\times...\times u(n-l+1) \qquad \textbf{(2.1)}$$

Yukarıdaki ifadede görülen *y(n-j)*, *n-j* anında ölçülen çıkış değeri, *u(n-j)*, *n-j* anında ölçülen giriş değeridir.

Bulanık çıkarsama kuralları, sembolik öncüllerden oluşan "eğer" kısmı ve sayısal sonuç ifadelerinden oluşan "o halde" kısmından oluşmaktadır. Başka bir deyişle bulanık bükülme alt modelleri aşağıdaki gibi bir yapıya sahiptir [11]:

$R^i$ : Eğer $y(n)\ A^i_0$, $y(n-1)\ A^i_1$, ….. , $y(n-m+1)\ A^i_{m-1}$ ve $u(n)\ B^i_0$, $u(n-1)\ B^i_1$, ….. , $u(n-l+1)\ B^i_{l-1}$ ise

O halde $y^i(n+1) = y(n) + \sum_{j=1}^{T} h^i_j \Delta u(n+1-j)$ dir. **(2.2)**

Yukarıda görülen ifadede

- $A^i_j$, *i*' nci bulanık çıkarsamadaki *y(n-j)* çıkışına ait bulanık kümedir.
- $B^i_j$, *i*' nci bulanık çıkarsamadaki *u(n-j)* girişine ait bulanık kümedir.
- $h^i_j$, *i*' nci bulanık çıkarsamadaki darbe yanıtı katsayılarıdır.
- *T,* model ufkudur.
- $\Delta u(n)$, *u(n)* ve *u(n-1)* arasındaki farktır [11].

Bu çalışmada, alt sistemlerin darbe yanıtları yerine, fark denklemleri kullanılmıştır. Bu değişiklik, kural tabanında herhangi bir farklılığa neden olmamış, sadece, alt sistemlerin çıkış hesaplama biçiminde bir farklılık getirmiştir.

Bütün bir bulanık bükülme modeli içinde *p* adet bulanık çıkarsama bulunmaktadır. Sistem çıkışı *y(n+1)*, bulanık çıkarsamalar tarafından hesaplanan çıkış değerlerinin ağırlıklı ortalaması alınarak bulunur [11].

$$y(n+1) = \frac{\sum_{j=1}^{p} w^j y^j(n+1)}{\sum_{j=1}^{p} w^j} \qquad \textbf{(2.3)}$$

Yukarıdaki denklemde $w^j$ terimi, $j$' nci bulanık çıkarsamanın gerçeklik derecesi, yani ateşleme çarpanıdır. Ateşleme çarpanı bulanık kümenin "eğer" kısmından yola çıkılarak aşağıdaki biçimde hesaplanır [11]:

$$w = \bigwedge_i A_i^j \bigwedge_k B_k^j \qquad \textbf{(2.4)}$$

Ateşleme çarpanlarını belirlemek üzere üçgen üyelik fonksiyonları kullanılacaktır.

$p$ adet alt sistemin ateşleme çarpanları toplamı 1'e eşit olmasından dolayı

$$\sum_{j=1}^{p} w^j = 1 \qquad \textbf{(2.5)}$$

denklemi yazılabilir.

Buna göre $y(n+1)$ için bulunan eşitlik aşağıdaki biçime dönüşür [11]:

$$y(n+1) = \sum_{j=1}^{p} w^j y^j(n+1) \qquad \textbf{(2.6)}$$

Eşitlik **(2.2)**' de görüldüğü üzere her bulanık altmodel için $y^j(n+1)$ hesaplanırken, $y^j(n)$ kullanmak yerine $y(n)$ kullanılmıştır. $n$ anında $y(n)$ bilindiğinden tahmin hatasını en aza indirgeyebilmek üzere böyle bir ifade kullanılmıştır [11].

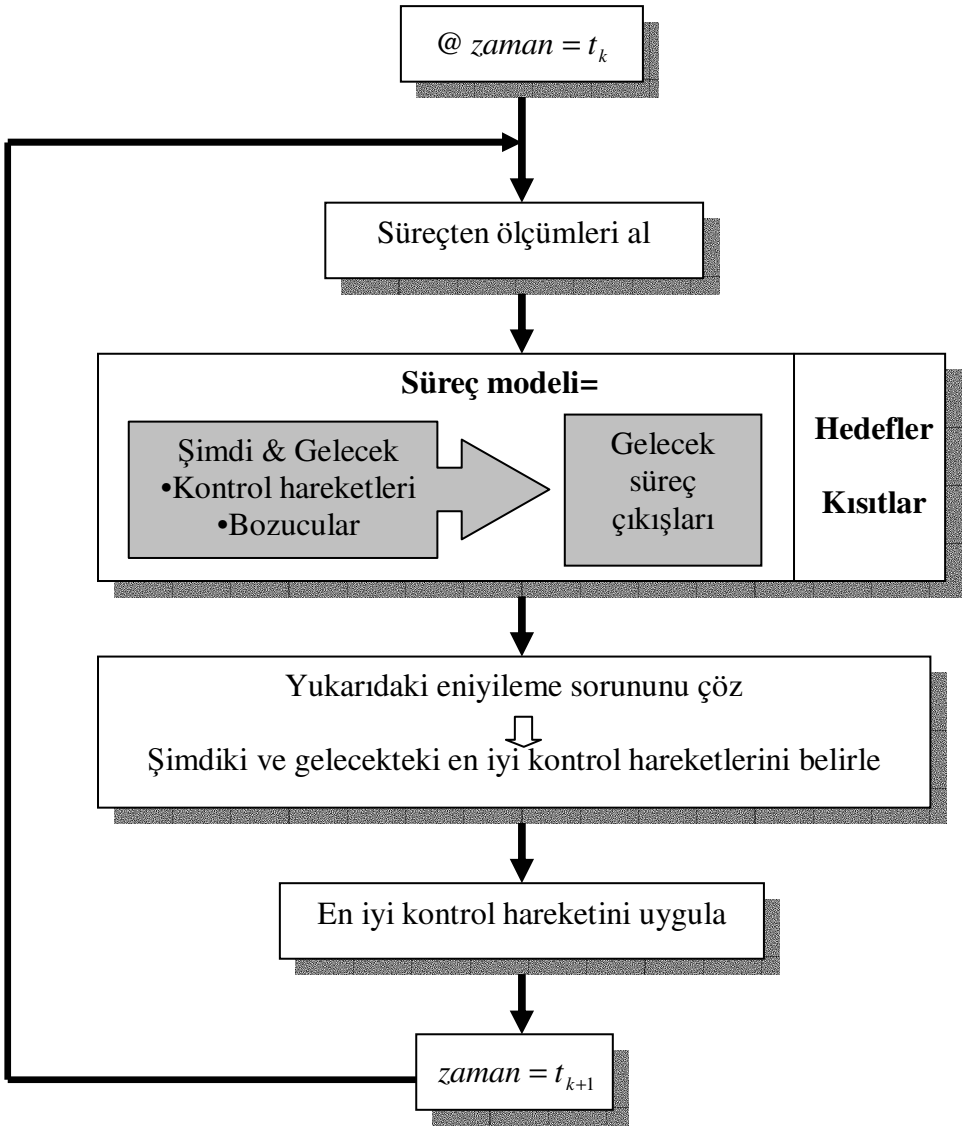
Çıkış değerleri hesaplandıktan sonra FMPC yapısına geçilecektir. Ancak bunun için öncelikle klasik MPC yapısının açıklanması gerekmektedir.

# 3. MODEL ÖNGÖRÜLÜ KONTROL (MPC)

MPC ailesine ait tüm kontrolörlerin yapısı aşağıdaki gibi bir strateji izler:

- Gelecek çıkış değerleri her *t* anında bir öngörü ufku boyunca hesaplanır. Bu öngörülen çıkışlar, geçmiş giriş ve çıkış değerlerine ve gelecekteki kontrol işaretlerine bağlı olarak bulunur.
- Gelecek kontrol işaretleri bir eniyileme yöntemi kullanılarak, süreci, referans yörüngesine mümkün olduğu kadar yakınsayacak biçimde hesaplanır. Genelde bu eniyileme yöntemi, öngörülen çıkış ve öngörülen referans yörüngesi arasındaki farkın ikinci dereceden bir fonksiyonu şeklindedir. Ve çoğunlukla kontrol artımları da başarım ölçütü içinde bulunur.
- Bulunan kontrol işaretlerinden sadece ilki sisteme gönderilir çünkü bir sonraki örnekleme anında bir sonraki çıkış değeri ölçülebilir olacaktır. Dolayısıyla yeni bir kontrol işareti hesaplaması yapılmalıdır. Bu durum gerileyen ufuk kavramı olarak bilinmektedir ve Bölüm 3.2.1' de daha ayrıntılı olarak bahsedilecektir [4].

Yukarıda anlatılan strateji Şekil 3.1' de akış diyagramı biçiminde görülmektedir.




**Şekil 3.1:** Model öngörülü kontrol şeması [3]

Şekil 3.1' de görüldüğü üzere MPC sistem davranışı karmaşık olabilmektedir. Karmaşıklığın sebebi kontrol eyleminin çevrimiçi eniyileme probleminin sonucu ile belirlenmesinden kaynaklanmaktadır [3].

### 3.1 Gerileyen Ufuk Yapısı

Öngörülü kontrolün temel düşüncesi Şekil 3.2' de görüldüğü gibidir. Burada tek giriş tek çıkışlı (SISO) bir sistemin kontrolü ele alınmıştır. Hesabın yapıldığı zamanın "*k*" anında olduğu düşünülürse sistemin çıkışı *y(k)* olarak gösterilmiş, çıkış yörüngesinin geçmişte almış olduğu değerler de verilmiştir. Ayrıca çıkışın ideal olarak takip etmesi gereken yörünge, ayar yörüngesi (setpoint) olarak gösterilmiştir. Herhangi bir *t* anında ayar yörüngesinin aldığı değer *s(t)* ile ifade edilmektedir. Ayar

yörüngesinden uzaklık referans yörüngesidir. Referans yörüngesi, o anki çıkış değeri *y(k)*' dan başlayan, -örneğin bir bozucu geldiği anda- ve sistemin ayar yörüngesine dönmek için izlediği ideal yörünge olarak tanımlanmaktadır. Genellikle referans yörüngesinin, o anki çıkış değerinden ayar değerine, üstel olarak yaklaştığı kabul edilmektedir. Cevabın hızını belirlemek üzere de üstel fonksiyonun zaman sabiti $T_{ref}$ ile ifade edilmektedir. Buna göre anlık hata ifadesi şu şekilde gösterilmiştir:

$$\varepsilon(k) = s(k) - y(k) \tag{3.1}$$

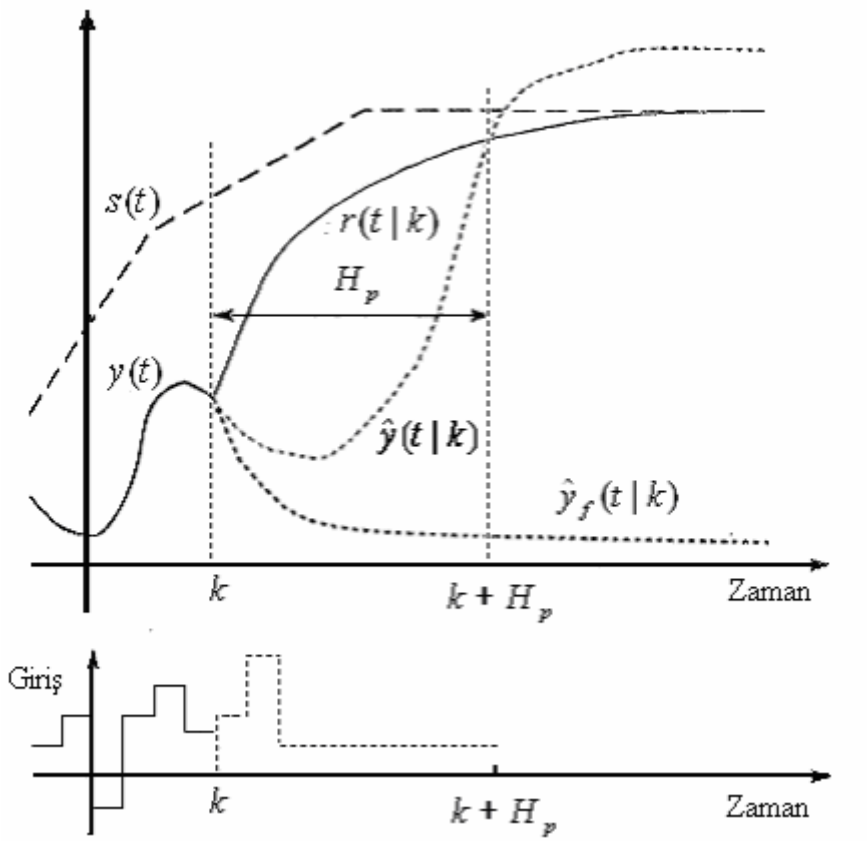
Eğer çıkış, referans yörüngesini takip edecek olursa, *i* adım sonra hata ifadesi aşağıdaki gibi olacaktır:

$$\begin{aligned} \varepsilon(k+i) &= e^{-iT_s/T_{ref}}\,\varepsilon(k) \\ \varepsilon(k+i) &= \lambda^i \varepsilon(k) \end{aligned} \tag{3.2}$$

Yukarıdaki ifadede $T_s$ örnekleme zamanıdır ve $\lambda = e^{-T_s/T^{ref}}$ tir. Böylece referans yörüngesi aşağıdaki gibi tanımlanabilir:

$$\begin{aligned} r(k+i|k) &= s(k+i) - \varepsilon(k+i) \\ r(k+i|k) &= s(k+i) - e^{-T_s/T^{ref}}\,\varepsilon(k) \end{aligned} \tag{3.3}$$

Yukarıda verilen $r(k+i|k)$ gösterimi, referans yörüngesinin *k* anındaki koşullara bağlı olduğu anlamına gelmektedir [2].




**Şekil 3.2:** Öngörülü kontrol: temel düşünce [2]

Öngörülü kontrolör, mevcut zamandan başlayıp, bir *öngörü ufku* $H_p$ boyunca sistem davranışını öngörebilmek için kullanılan içsel bir modele sahiptir. Öngörülen sistem davranışı, öngörü ufku boyunca uygulanacak varsayılan giriş yörüngesine $\hat{u}(k+i|k)$ $(i = 0,1,...,H_p - 1)$ bağlıdır ve asıl düşünce en iyi öngörülü davranışı vadeden girişi seçmektir. İçsel modelin doğrusal olduğu varsayılmaktadır. Gösterim olarak *u* yerine $\hat{u}$ seçilmiş olması, *k* anında, girişin o anki değerine değil, *k+i* anında ne olacağı ile ilgili öngörüye sahip olunduğu anlamına gelmektedir. Ayrıca giriş değeri *u(k)*' nın hangi değeri alacağına karar verirken, *y(k)* çıkış değerinin bilindiği varsayılmaktadır. Yani *y(k)* o anki giriş *u(k)* ya değil, geçmiş giriş değerlerine *u(k-1), u(k-2),…* bağlıdır.

En basit durumda, giriş yörüngesi, öngörü ufkunun sonunda ($k+H_p$ ' de) sistem çıkışını referansa getirecek şekilde seçilebilir. Şekil 3.2'de görüldüğü üzere, girişin, öngörü ufkunun ilk üç adımı boyunca değiştiği varsayılmaktadır. Bu da *kontrol ufku* $H_u$' yu ifade etmektedir. Buna göre giriş yörüngesi sonraki $H_u$ adım boyunca değişiklik gösterecek, daha sonra sabit kalacaktır. Böylece giriş yörüngesi

$\hat{u}(k|k), \hat{u}(k+1|k), ..., \hat{u}(k+H_u-1|k)$ olarak seçilir ve $\hat{u}(k+H_u-1|k) = \hat{u}(k+H_u|k) = ..., \hat{u}(k+H_p-1|k)$ olduğu varsayımı yapılır. Genellikle $H_u$, $H_p$' den daha düşük bir değerde seçilir. Şekil 3.2' de görüldüğü üzere $H_u$ = 3 olmasına bağlı olarak $\hat{u}(k+2|k) = \hat{u}(k+3|k) = ..., \hat{u}(k+H_p-1|k)$ dir ve seçilmesi gereken üç parametre $\hat{u}(k|k), \hat{u}(k+1|k), \hat{u}(k+2|k)$ dir.

Giriş yörüngesi seçildikten sonra, yapılması gereken, sisteme giriş sinyali olarak sadece ilk yörünge elemanını uygulamaktır. En basit durum için, girişin öngörü ufku boyunca değişmediği varsayılacak olursa, $H_u=1$, seçilmesi gereken sadece bir parametre vardır ki o da $\hat{u}(k|k)$' dır ve sağlanması gereken bir eşitlik vardır ki o da $\hat{y}(k+H_p|k) = r(k+H_p|k)$' dir. Bu aşamadan sonra asıl giriş sinyali olan *u(k)* , $u(k) = \hat{u}(k|k)$ eşitliği ile belirlenir ve sisteme uygulanır. Asıl giriş sinyali sisteme uygulandıktan sonra tüm döngü – çıkış ölçülmesi, öngörü işlemleri, giriş yörünge belirlenmesi - bir sonraki örnekleme aralığı için tekrarlanır: yeni bir çıkış ölçümü yapılır ve *y(k+1)* elde edilir; yeni bir referans yörüngesi $r(k+i|k+1)$ $i = 2,3,...$ tanımlanır; $k+1+i$ ( $i = 2,3,...,H_p$ ) ufku boyunca öngörüler yapılır; yeni bir giriş yörüngesi $\hat{u}(k+1+i|k+1)$ ( $i = 2,3,...,H_p-1$) seçilir; ve en sonunda bir sonraki giriş değeri sisteme uygulanır: $u(k+1) = \hat{u}(k+1|k+1)$. Öngörü ufkunun uzunluğu bir önceki döngüye göre değişmemektedir. Sadece bir örnekleme aralığı boyunca her adımda kaydırılmıştır. Bir sistemin bu şekilde kontrolü gerileyen ufuk (receiding horizon) stratejisi olarak adlandırılır.

### 3.2 En İyi Girişlerin Hesaplanması

Çoğunlukla öngörü ufku üzerinde birden çok çakışma noktası görülür, bazı durumlarda tüm noktalar $k+1$, $k+2$,..., $k+H_p$ çakışma noktası olabilir. Bu gibi durumlarda mevcut değişken sayısından daha fazla eşitlik bulunmaktadır ve genellikle kesin çözümü bulmak mümkün değildir. Açıkça görüldüğü üzere yaklaşık çözüm arayışı kaçınılmazdır. En sık başvurulan yöntem 'en küçük kareler' çözümünün bulunmasıdır. Buna göre hataların kareleri toplamı $\sum_{i \in P} [r(k+i|k) - \hat{y}(k+1|k)]^2$ en aza indirgenir. Burada *P*, çakışma noktalarına tekabül

eden $i$ indisleri kümesini göstermektedir. Eğer içsel model doğrusalsa, en küçük kareler çözümü kolayca bulunabilir, ve sonuçta doğrusal bir kontrol kuralı oluşur.

Tekrar en basit durum olan $k+H_p$ ' de tek çakışma noktası ve seçilmesi gereken tek parametre $\hat{u}(k|k)$ olduğu varsayılsın. İzlenen prosedür şu şekildedir: içsel model, ilk olarak sistemin *serbest cevabını (free response)* $\hat{y}_f(k+H_p|k)$ öngörmek için kullanılır. Serbest cevap, gelecek giriş yörüngesinin en son giriş değeri olan $u(k-1)$' de kalması durumunda çakışma noktasında elde edilen cevap olarak tanımlanır.

Sistem modeli olarak elimizde sistemin basamak cevabının - tüm geçmiş giriş değerleri ile birlikte - bulunduğu varsayılsın. $S(H_p)$, sistem modelinin, birim basamak giriş uygulandıktan $H_p$ adım sonra verdiği cevap olsun. Buna göre $k+ H_p$ anında, öngörülen çıkış ifadesi aşağıdaki gibi olur:

$$\hat{y}(k+H_p|k) = \hat{y}_f(k+H_p|k) + S(H_p)\Delta\hat{u}(k|k) \qquad \textbf{(3.4)}$$

$$\Delta\hat{u}(k|k) = \hat{u}(k|k) - u(k-1) \qquad \textbf{(3.5)}$$

Yukarıda görülen $\Delta\hat{u}(k|k)$ ifadesi, öngörülen giriş $\hat{u}(k|k)$'nin, o anki giriş $u(k-1)$'e göre değişimidir. Gerçekleştirilmek istenen

$$\hat{y}(k+H_p|k) = r(k+H_p|k) \qquad \textbf{(3.6)}$$

buna göre en uygun giriş değişimi aşağıdaki gibi verilir:

$$\Delta\hat{u}(k|k) = \frac{r(k+H_p|k) - \hat{y}_f(k+H_p|k)}{S(H_p)} \qquad \textbf{(3.7)}$$

Birden fazla çakışma noktası olması durumunda, gerçekleştirilmek istenen $i = 1,2,...,c$, için $\hat{y}(k+P_i|k) = r(k+P_i|k)$ $i = 1,2,...,c$, dir. Burada $c$ çakışma noktası sayısıdır. Buna göre aşağıdaki eşitlikleri çözebilmek için $\hat{u}(k|k)$' nın seçilmesi gerekir.

$$\begin{aligned}
\hat{r}(k+P_1 \mid k) &= \hat{y}_f(k+P_1 \mid k) + S(P_1)\Delta\hat{u}(k \mid k) \\
\hat{r}(k+P_2 \mid k) &= \hat{y}_f(k+P_2 \mid k) + S(P_2)\Delta\hat{u}(k \mid k) \\
&\vdots \\
\hat{r}(k+P_c \mid k) &= \hat{y}_f(k+P_c \mid k) + S(P_c)\Delta\hat{u}(k \mid k)
\end{aligned} \qquad \textbf{(3.8)}$$

Çoğunlukla, bu eşitlikler, 'en küçük kareler' yöntemi ile çözülür.

$$T = \begin{bmatrix} \hat{r}(k+P_1 \mid k) \\ \hat{r}(k+P_2 \mid k) \\ \vdots \\ \hat{r}(k+P_c \mid k) \end{bmatrix} \quad y_f = \begin{bmatrix} \hat{y}_f(k+P_1 \mid k) \\ \hat{y}_f(k+P_2 \mid k) \\ \vdots \\ \hat{y}_f(k+P_c \mid k) \end{bmatrix} \quad S = \begin{bmatrix} S(P_1) \\ S(P_2) \\ \vdots \\ S(P_c) \end{bmatrix} \qquad \textbf{(3.9)}$$

MATLAB' da en küçük kareler çözümü 'BACKSLASH' operatörü kullanılarak yapılır. Buna göre en iyi giriş değişimi eşitlik **(3.10)**' daki gibi hesaplanır.

$$\Delta\hat{u}(k \mid k) = S \setminus (T - y_f) \qquad \textbf{(3.10)}$$

***Örnek 1:***

Ayar değerinin sabit bir değerde *s(k+i)=3* olduğu varsayılsın. Üstel fonksiyonun zaman sabiti $T_{ref}$ = 9 s olsun. Örnekleme zamanı $T_s$ = 3 s ve iki adet çakışma noktası $P_1$ = 1, $P_2 = H_p$ = 2 gibi verilsin. Sistemin z-dönüşümü matematiksel modeli aşağıdaki gibi tanımlansın:

$$G(z) = \frac{2}{z - 0.7}$$

Geçmiş ve şu anki çıkış değerleri y(k-1) = y(k) = 2, ve en son kontrol girişi u(k-1) = 0.3 olsun. Buna göre $\hat{u}(k|k) = \hat{u}(k+1|k)$ ise , en iyi giriş değeri $\Delta\hat{u}(k|k)$ nedir?

***Çözüm:***

$\varepsilon(k) = s(k) - y(k) = 3 - 2 = 1$, ve $\lambda = \exp(-T_s / T_{ref}) = 0.7165$ dir.

Buna göre referans yörüngesi aşağıdaki gibi hesaplanır:

$$r(k+1 \mid k) = s(k+1) - \lambda\varepsilon(k) = 2.284$$
$$r(k+2 \mid k) = s(k+2) - \lambda^2\varepsilon(k) = 3 - 0.7165^2 = 2.487$$

Serbest cevabı bulabilmek için, matematiksel modeli fark denklemine dönüştürmek gerekmektedir.

$$y(k) = 0.7\,y(k-1) + 2u(k-1)$$

Serbest cevabın tanımına göre, girişin en son aldığı değerde sabit kaldığı düşünüldüğünden $u(k+1) = u(k) = u(k-1) = 0.3$ olduğu varsayılır.

$$\hat{y}_f(k+1\,|\,k) = 0.7\times 2 + 2\times 0.3 = 2.0$$
$$\hat{y}_f(k+2\,|\,k) = 0.7\times 2.0 + 2\times 0.3 = 2.0$$

Son olarak $S(1)$ ve $S(2)$' nin bulunması gerekmektedir. Bunun için fark denkleminde giriş değerleri için $u(k) = u(k+1) = 1$ ve çıkış değerleri için

$y(k) = y(k-1) = 0$ yazılması yeterli olacaktır.

$$S(1) = 0.7\times 0 + 2\times 1 = 2$$
$$S(2) = 0.7\times 2 + 2\times 1 = 3.4$$

Bu durumda aşağıdaki eşitlikler oluşturulabilir:

$$T = \begin{bmatrix} 2.284 \\ 2.487 \end{bmatrix} \quad y_f = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix} \quad S = \begin{bmatrix} 2.0 \\ 3.4 \end{bmatrix}$$

MATLAB' ın BACKSLASH operatörü kullanılırsa

$$\Delta\hat{u}(k\,|\,k) = S \setminus (T - y_f) = 0.1429$$
$$\hat{u}(k\,|\,k) = u(k-1) + \Delta\hat{u}(k\,|\,k) = 0.4429$$

Şimdi de daha karmaşık bir giriş yörüngesi olduğu varsayılsın. Girişin sonraki $H_u$ adım boyunca değiştiği düşünülsün. $k+P_i$ anında öngörülen çıkış değeri aşağıdaki eşitlik ile hesaplanmaktadır:

$$\begin{aligned}\hat{y}(k+P_i\,|\,k) = \hat{y}_f(k+P_i\,|\,k) &+ H(P_i)\hat{u}(k\,|\,k) + H(P_i-1)\hat{u}(k+1\,|\,k) + \cdots \\ &+ H(P_i - H_u + 2)\hat{u}(k+H_u-2\,|\,k) + \\ &+ S(P_i - H_u + 1)\hat{u}(k+H_u-1\,|\,k)\end{aligned} \qquad \textbf{(3.11)}$$

Yukarıdaki eşitlikte $H(j)=S(j)-S(j-1)$' dir ve sistemin $j$ adım sonraki birim darbe cevabıdır.

$$\begin{aligned}\hat{y}(k+P_i\,|\,k) = \hat{y}_f(k+P_i\,|\,k) &+ S(P_i)\Delta\hat{u}(k\,|\,k) + S(P_i-1)\Delta\hat{u}(k+1\,|\,k) \\ &+ \cdots + S(P_i - H_u + 1)\Delta\hat{u}(k+H_u-1\,|\,k)\end{aligned} \qquad \textbf{(3.12)}$$

$$y = y_f + \Theta \Delta u \tag{3.13}$$

$$y = \begin{bmatrix} \hat{y}(k+P_1 \mid k) \\ \hat{y}(k+P_2 \mid k) \\ \vdots \\ \hat{y}(k+P_c \mid k) \end{bmatrix} \quad \Delta u = \begin{bmatrix} \Delta\hat{u}(k \mid k) \\ \Delta\hat{u}(k+1 \mid k) \\ \vdots \\ \Delta\hat{u}(k+H_u-1 \mid k) \end{bmatrix} \tag{3.14}$$

$$\Theta = \begin{bmatrix} S(P_1) & S(P_1-1) & \cdots & S(1) & 0 & \cdots & 0 & 0 & \cdots & 0 \\ S(P_2) & S(P_2-1) & \cdots & \cdots & \cdots & \cdots & S(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S(P_c) & S(P_c-1) & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & S(P_c-H_u+1) \end{bmatrix} \tag{3.15}$$

$$\Delta u = \Theta \setminus \left[ T - y_f \right] \tag{3.16}$$

$\Delta U$ vektörünün ilk elemanı seçilir ve bir önceki giriş değeri ile toplanır. Sisteme uygulanan giriş oluşturulur.

$$u(k) = \Delta\hat{u}(k \mid k) + u(k-1) \tag{3.17}$$

ve bir sonraki zaman adımı için tüm hesaplama döngüsü, bir sonraki sistem çıkışı *y(k+1)'* in ölçümüyle başlayarak tekrarlanır.

***Örnek 2:***

Örnek 1 için bu kez $H_u = 2$ seçilmiştir, böylece her adımda $\hat{u}(k|k)$ ve $\hat{u}(k+1|k)$'nin en iyi değerleri hesaplanacaktır. Tek fark

$$\Theta = \begin{bmatrix} S(1) & 0 \\ S(2) & S(1) \end{bmatrix} = \begin{bmatrix} 2.0 & 0 \\ 3.4 & 2.0 \end{bmatrix}$$

Görüldüğü üzere $\Theta$ matrisi kare bir matristir ve tersi alınabilir, bu sebeple çözüm tektir ve MATLAB' ın BACKSLASH operatörü yine doğru cevapı verebilecektir:

$$\Delta u = \Theta \setminus [T - y_f] = \begin{bmatrix} 0.1420 \\ 0.0021 \end{bmatrix}$$

Buna göre sisteme uygulanacak giriş aşağıdaki gibidir:

$$u(k) = \Delta\hat{u}(k \mid k) + u(k-1) = 0.1420 + 0.3 = 0.4420$$

Görüldüğü üzere öngörülü kontrolü kullanabilmek için sistemin ayrık zaman modeline ihtiyaç duyulmamaktadır. Sistemle ilgili bilinmesi gereken, çakışma noktalarındaki sistem basamak cevapları ve bu noktalardaki serbest cevapların hesaplanabilirliğidir. Yani gerçek zamandan daha hızlı çalışabilecek bir simülasyon modeline sahip olunmalıdır. Ancak pratikte, ayrık zaman doğrusal modeli elde etmek daha kolaydır ve sürekli zaman doğrusal modele göre daha kullanışlıdır [2].

### 3.3 Durum Uzayı Modeli

Sistem modelinin doğrusallaştırılmış, ayrık zamanlı durum uzayı biçiminde verildiği kabul edilsin.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= C_y x(k) \\ z(k) &= C_z x(k) \end{aligned} \qquad \textbf{(3.18)}$$

Burada *x, n*-boyutlu durum vektörü, *u, l*-boyutlu giriş vektörü, *y,* $m_y$-boyutlu ölçülen çıkış vektörü, ve z ise $m_z$ -boyutlu kontrol edilen çıkışların vektörüdür. $y$ ve $z$ vektörlerindeki değişkenler uzun vadede üst üste geldikleri için genellikle aynı kabul edilecektir, yani tüm kontrol edilen çıkışların ölçüldüğü varsayılacaktır. Buna göre $y \equiv z$ olduğu düşünülecektir ve $C_y$ ve $C_z$ matrisleri için ortak olarak $C$, $m_y$ ve $m_z$ için ortak olarak $m$ gösterimi kullanılacaktır.

Yukarıda verildiği gibi durum uzay modelinin standart biçiminin kullanılmasının sebebi, doğrusal sistem teorisi ve kontrolü ile doğrudan bağlantılı olmasındandır.

*k* anında yapılması gerekenlerin sırası aşağıdaki gibi verilmiştir:

1. *y(k)* ölçümlerinin alınması
2. İstenen sistem girişi *u(k)'* nın hesaplanması
3. *u(k)'* nın sisteme uygulanması

Yukarıda görüldüğü üzere, *y(k)*' nin ölçülmesi ve *u(k)'* nin uygulanması arasında her zaman bir gecikme olacaktır.

### 3.4 Temel Denklemler

Öngörülü kontrolün temel denklemlerini çıkarmak için, sistem modelinin doğrusal, başarım ölçütünün ikinci mertebeden, kısıtların doğrusal eşitsizlikler biçiminde verildiğini ve her şeyin zamanla değişmeyen olduğunu varsaymak gerekmektedir. Ayrıca başarım ölçütünün sadece giriş vektörünün değişimlerini yani $\Delta u(k)$'yı cezalandırdığı kabul edilecektir.

Değer fonksiyonu *V*, öngörülen çıkışların, $\hat{z}(k+1|k)$, referans yörüngesinden, $r(k+i|k)$, sapmasını da cezalandıracaktır. Yine gösterimde belirtildiği üzere, referans yörüngesi *k* anına kadar yapılan ölçümlere dayanılarak bulunmakta; başlangıç noktası, çıkış ölçümü *y(k)* olmaktadır. Fakat başka bir sabit ayar değeri veya önceden tayin edilmiş bir yörünge de referans yörüngesi olarak tanımlanabilir. Buna göre değer fonksiyonu aşağıdaki gibi tanımlanır:

$$V(k) = \sum_{i=H_w}^{H_p} \left\| \hat{z}(k+i|k) - r(k+i|k) \right\|_{Q(i)}^2 + \sum_{i=0}^{H_u-1} \left\| \Delta\hat{u}(k+i|k) \right\|_{R(i)}^2 \qquad \textbf{(3.19)}$$

Öngörü ufkunun boyutu $H_p$ olmasına rağmen, *z*'nin *r*'den sapmasını hemen cezalandırmak şart değildir ($H_w$>1). Bunun sebebi girişin uygulanması ve bir etki görülmesi arasında genellikle bir gecikmeyle karşılaşılmasındandır. Burada verilen "$H_w$" pencere parametresi olarak adlandırılır. Her zaman $H_u \le H_p$ olduğu varsayılacaktır.

Değer fonksiyonu veya diğer bir adıyla başarım ölçütü *V*, hata vektörü $\hat{z}(k+i|k) - r(k+i|k)$'yi öngörü ufku boyunca $H_w \le i \le H_p$ aralığındaki her noktada cezalandıracaktır. Bu öngörülü kontrolde en sık karşılaşılan durumdur. Fakat hatayı, *i*' nin pek çok değerinde $Q(i) = 0$ yaparak, sadece birkaç çakışma noktasında cezalandırmak da mümkündür.

***Örnek 3:***

İki adet kontrol edilen çıkış olduğu kabul edilsin. Öngörü ufku $H_p = 3$ olsun. Birinci çıkış için tek çakışma noktası $i = 2$ 'de olsun. İkinci çıkış için $i = 2$ *ve* $i = 3$' te iki adet çakışma noktası bulunsun. İkinci çıkışın hataları daha ağır cezalandıracak olursa, bu durum aşağıdaki gibi gösterilebilir:

$$Q(1)=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\quad Q(2)=\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}\quad Q(3)=\begin{bmatrix}0 & 0\\ 0 & 2\end{bmatrix}$$

Eğer $H_w = 1$ seçilecek olursa yukarıdaki ifade doğru olur. Ancak, $H_w = 2$ alınıp, $Q(1)$ tanımlanmayacak olursa $Q(2)$ ve $Q(3)$ yukarıdaki gibi kalacaktır. Genellikle ağırlık matrisleri yukarıdaki örnekte gösterildiği gibi diyagonaldir.

Öngörü ufku $H_p$ , kontrol ufku $H_u$ , pencere parametresi $H_w$ , ağırlık matrisleri (cezalandırma matrisleri) $Q(i)$ ve $R(i)$, referans yörüngesi $r(k+i)$, kapalı çevrim sistem kontrolünü etkileyen parametrelerdir. Bu ayar parametreleri, sistemin istenen dinamiğe gelmesi için uygun değerlerde seçilmelidir.

### 3.4.1 Öngörülü kontrol problemlerinin çözümü

Kısıtsız öngörülü kontrol problemleri için en aza indirgenmek istenen değer fonksiyonu **(3.19)**' da verildiği üzere aşağıdaki gibidir:

$$V(k)=\sum_{i=H_w}^{H_p}\left\|\hat{z}(k+i|k)-r(k+i|k)\right\|_{Q(i)}^{2}+\sum_{i=0}^{H_u-1}\left\|\Delta\hat{u}(k+i|k)\right\|_{R(i)}^{2}$$

Yukarıdaki fonksiyon şu şekilde de yazılabilir:

$$V(k)=\left\|Z(k)-T(k)\right\|_{Q}^{2}+\left\|\Delta U(k)\right\|_{R}^{2} \qquad \textbf{(3.20)}$$

$$Z(k)=\begin{bmatrix}\hat{z}(k+H_w|k)\\ \vdots\\ \hat{z}(k+H_p|k)\end{bmatrix} \qquad \textbf{(3.21)}$$

$$T(k)=\begin{bmatrix}\hat{r}(k+H_w|k)\\ \vdots\\ \hat{r}(k+H_p|k)\end{bmatrix} \qquad \textbf{(3.22)}$$

$$\Delta U(k)=\begin{bmatrix}\Delta\hat{u}(k|k)\\ \vdots\\ \Delta\hat{u}(k+H_u-1|k)\end{bmatrix} \qquad \textbf{(3.23)}$$

Ağırlık matrisleri Q ve R de aşağıdaki gibi bulunabilir:

$$Q = \begin{bmatrix} Q(H_w) & 0 & \cdots & 0 \\ 0 & Q(H_w+1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q(H_p) \end{bmatrix} \quad \textbf{(3.24)}$$

$$R = \begin{bmatrix} R(0) & 0 & \cdots & 0 \\ 0 & R(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R(H_u-1) \end{bmatrix} \quad \textbf{(3.25)}$$

Ayrıca uygun $\Psi, \Upsilon$ ve $\Theta$ matrisleri için *Z(k),* eşitlik **(3.26)** de verildiği gibi hesaplanır:

$$Z(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta \Delta U(k) \quad \textbf{(3.26)}$$

$$\Psi = \begin{bmatrix} C_z & 0 & \cdots & 0 \\ 0 & C_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} A \\ \vdots \\ A^{Hu} \\ A^{Hu+1} \\ \vdots \\ A^{Hp} \end{bmatrix} \quad \textbf{(3.27)}$$

$$\Upsilon = \begin{bmatrix} C_z & 0 & \cdots & 0 \\ 0 & C_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{Hu-1} A^i B \\ \sum_{i=0}^{Hu} A^i B \\ \vdots \\ \sum_{i=0}^{Hp-1} A^i B \end{bmatrix} \quad \textbf{(3.28)}$$

$$\Theta = \begin{bmatrix} C_z & 0 & \cdots & 0 \\ 0 & C_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} B & \cdots & 0 \\ AB+B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{Hu-1} A^i B & \cdots & B \\ \sum_{i=0}^{Hu} A^i B & \cdots & AB+B \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{Hp-1} A^i B & \cdots & \sum_{i=0}^{Hp-Hu} A^i B \end{bmatrix} \quad \textbf{(3.29)}$$

Bu aşamada aşağıdaki ifade tanımlanır:

$$\varepsilon(k) = T(k) - \Psi x(k) - \Upsilon u(k-1) \quad \textbf{(3.30)}$$

$\Delta U(k) = 0$ kabul edilip, gelecek hedef yörüngesinin serbest cevaptan farkı alındığı varsayıldığında, yukarıdaki denklem, 'izleme hatası' olarak düşünülebilir. Eğer hata $\varepsilon(k)$ gerçekten 0 olursa, $\Delta U(k) = 0$ almak doğru olur. Buna göre aşağıdaki denklemler yazılabilir:

$$\begin{aligned} V(k) &= \left\| \Theta \Delta U(k) - \varepsilon(k) \right\|_Q^2 + \left\| \Delta U(k) \right\|_R^2 \\ &= \left[ \Delta U(k)^T \Theta^T - \varepsilon(k)^T \right] Q \left[ \Theta \Delta U(k) - \varepsilon(k) \right] + \Delta U(k)^T R \Delta U(k) \\ &= \varepsilon(k)^T Q \varepsilon(k) - 2\Delta U(k)^T \Theta^T Q \varepsilon(k) + \Delta U(k)^T \left[ \Theta^T Q \Theta + R \right] \Delta U(k) \end{aligned} \quad \textbf{(3.31)}$$

Yukarıdaki denklem şöyle bir biçime sahiptir:

$$V(k) = sabit_ \Delta U(k)^T G + \Delta U(k)^T H \Delta U(k) \quad \textbf{(3.32)}$$

$$G = 2\Theta^T Q \varepsilon(k) \quad \textbf{(3.33)}$$

$$H = \Theta^T Q \Theta + R \quad \textbf{(3.34)}$$

Yukarıda görülen *G* ve *H* ifadelerinin hiçbiri $\Delta U(k)$' ya bağlı değildir.

En iyi $\Delta U(k)$ değerini bulabilmek için, değer fonksiyonu *V(k)*' nın gradyanını alıp sıfıra eşitlemek gerekmektedir. Buna göre en iyi gelecek giriş hareketleri kümesi eşitlik **(3.35)**' deki gibi görülür:

$$\Delta U(k)_{opt} = \frac{1}{2} H^{-1} G \qquad \textbf{(3.35)}$$

Eğer sistemin girişlerinin sayısı $l$ ise, $\Delta U(k)_{opt}$'un ilk $l$ satırını kullanmak yeterli olacaktır.

$$\Delta u(k)_{opt} = [I_\ell, 0_\ell \ldots, 0_\ell] \Delta U(k)_{opt} \qquad \textbf{(3.36)}$$

Yukarıda görülen $I_l$, $l \; x \; l$ boyutlu birim matris, $0_l$ ise $l \; x \; l$ boyutlu sıfır matrisidir.

Eşitlik **(3.35)**'de verilen eniyileme çözümü, asla $H$ matrisinin tersini alıp bulunmamalıdır. Bunun sebebi, $\Theta$ matrisinin genelde kötü karakteristikli olmasına bağlı olarak H matrisinin de kötü karakteristikli olmasındandır.

Çözümü hesaplamanın en iyi yolu, problemi 'en küçük kareler' yöntemiyle ele almaktır. $Q \geq 0$ ve $R \geq 0$ olması sebebiyle bu iki matrisin karekökleri olan $S_Q$ ve $S_R$ matrisleri bulunabilir.

$$S_Q^T S_Q = Q \qquad S_R^T S_R = R \qquad \textbf{(3.37)}$$

Buna göre aşağıdaki vektör düşünülecek olursa

$$\begin{bmatrix} S_Q \{\Theta \Delta U(k) - \varepsilon(k)\} \\ S_R \Delta U(k) \end{bmatrix} \qquad \textbf{(3.38)}$$

Yukarıda verilen vektörün uzunluğunun karesinin yada elemanlarının karelerinin toplamının değer fonksiyonu *V(k)'* ye eşit olduğunu ve $\Delta U(k)_{opt}$' un $\Delta U(k)$'yı minimize eden değer olduğunu göstermek gerekmektedir.

$$\begin{aligned} &\left\| \begin{bmatrix} S_Q \{\Theta \Delta U(k) - \varepsilon(k)\} \\ S_R \Delta U(k) \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} S_Q \{Z(k) - T(k)\} \\ S_R \Delta U(k) \end{bmatrix} \right\|^2 \\ &= [Z(k) - T(k)]^T S_R^T S_Q [Z(k) - T(k)] + \Delta U(k)^T S_R^T S_R \Delta U(k) \\ &= \|Z(k) - T(k)\|_Q^2 + \|\Delta U(k)\|_R^2 \\ &= V(k) \end{aligned} \qquad \textbf{(3.39)}$$

Görüldüğü üzere $\Delta U(k)_{opt}$ eşitlik **(3.40)**' nın 'en küçük kareler' çözümüdür.

$$\left\| \begin{bmatrix} S_Q\{\Theta\Delta U(k) - \varepsilon(k)\} \\ S_R \Delta U(k) \end{bmatrix} \right\| = 0 \qquad \textbf{(3.40)}$$

Eşitlik **(3.40)** aşağıdaki şekilde de yazılabilir:

$$\begin{bmatrix} S_Q\Theta \\ S_R \end{bmatrix} \Delta U(k) = \begin{bmatrix} S_Q\varepsilon(k) \\ 0 \end{bmatrix} \qquad \textbf{(3.41)}$$

MATLAB gösterimi kullanılarak en iyi çözüm $\Delta U(k)_{opt}$ aşağıdaki gibi hesaplanır:

$$\Delta U(k)_{opt} = \begin{bmatrix} S_Q\Theta \\ S_R \end{bmatrix} \backslash \begin{bmatrix} S_Q\varepsilon(k) \\ 0 \end{bmatrix} \qquad \textbf{(3.42)}$$

Yukarıda verilen denklemin tek çözümü olması için, $S_Q\Theta$ matrisinin tekil olmayan ve kare matris olması gerekmektedir. Genel bir çözüm bulabilmek için **(3.30)** ve **(3.36)** numaralı eşitlikleri hatırlamak gerekir:

$$\Delta u(k)_{opt} = [I_\ell, 0_\ell, \ldots, 0_\ell] H^{-1}\Theta^T Q\varepsilon(k)$$

$$\varepsilon(k) = T(k) - \Psi x(k) - \Upsilon u(k-1)$$

Yukarıdaki çözümün her adımda değişen tek elemanı 'izleme hatası' $\varepsilon(k)$'dır.

Eğer $K_{MPC}$ değişkeni aşağıdaki gibi ifade edilecek

$$K_{MPC} = [I_\ell, 0_\ell, \ldots, 0_\ell] H^{-1}\Theta^T Q \qquad \textbf{(3.43)}$$

$K_{MPC}$ ' yi doğru hesaplamak için yine MATLAB gösterimi ile birlikte, ilk *l* satırı çekmek üzere ':' gösterimi kullanılacak olursa:

$$K_{full} = \begin{bmatrix} S_Q\Theta \\ S_R \end{bmatrix} \backslash \begin{bmatrix} S_Q \\ 0 \end{bmatrix} \qquad \textbf{(3.44)}$$

MATLAB'ın Model Öngörülü Kontrol Araç kutusuna ait *smpccon* fonksiyonu $K_s$ matrisinin hesaplanmasına yardımcı olur ve $K_s$ ile $K_{MPC}$ arasındaki ilişki aşağıdaki gibi bulunabilir:

$$K_s = K_{MPC} \begin{bmatrix} \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}, -\Psi, -\Upsilon \end{bmatrix} \qquad \textbf{(3.45)}$$

$$\Delta u(k)_{opt} = K_S \begin{bmatrix} T(k) \\ x(k) \\ u(k-1) \end{bmatrix} \quad \textbf{(3.46)}$$

Eğer tüm durum vektörünün ölçümleri mevcut değilse, bir gözleyici kullanılması gerekmektedir. Tüm kazanç matrisleri değişmeden kalacaktır. Tek değişiklik, ölçülen durum *x(k)* yerine tahmini durum değeri $\hat{x}(k|k)$' in kullanılmasıdır [2].

Önerilen FMPC yapısı içinde en küçük kareler yöntemi eniyileme yöntemi olarak kullanılmaktadır. Buna göre en iyi gelecek giriş hareketleri kümesi bir önceki giriş değeri ile toplanır ve yeni giriş değeri olarak, kontrol edilmek istenen sisteme verilir.

# 4. BULANIK MODEL ÖNGÖRÜLÜ KONTROL (FMPC)

Referans [11]' de anlatılan FMPC yapısı, klasik MPC yapısının tüm özelliklerini barındırır ve amacı yine çıkış ile referans yörüngesi arasındaki öngörü hatasını, sonraki $N_y$ adım boyunca eniyi $N_u$ adım kontrol işaretini belirleyerek, en aza indirgemektir [11].

Yukarıdaki hedef doğrultusunda eniyileme problemi aşağıdaki gibi ifade edilmiştir.

$$\min_{\Delta u(n),\Delta u(n+1),\ldots,\Delta u(n+N_u)} J(n) \tag{4.1}$$

$$J(n) = \sum_{i=1}^{N_y} \mu_i \left(\hat{y}(n+i) - y^r(n+i)\right)^2 + \sum_{i=1}^{N_u} \upsilon_i \Delta u^j(n+i)^2 \tag{4.2}$$

- $\mu_i$ ve $\upsilon_i$ sırasıyla, öngörü hatası ve kontrol artımlarının ağırlık (cezalandırma) katsayıları,
- $\hat{y}(n+i)$ i' nci adım çıkış öngörüsü,
- $y^r(n+i)$ i' nci adım referans yörüngesi,
- $\Delta u(n+i)$ i' nci adım kontrol artımıdır.

**(4.1)**' de verilen amaç fonksiyonu, **(2.2)**' de verilen *p* adet bulanık çıkarsamaya (FI) bağlıdır [11].

Denklem **(4.2)**' te verilen kontrol artımını $\Delta u(n+i)$, $i = 1,2,\ldots,N_u$ oluşturmak için öncelikle *p* adet lokal kontrol artımı $\Delta u^j(n+i)$, $i = 1,2,\ldots,N_u$, $j=1,2,\ldots,p$ elde edilir ve buradaki *p* altsistem sayısıdır. Daha sonra, hesaplanan lokal kontrol artımlarının ağırlıklı ortalaması alınır ve genel kontrol artımı hesaplanır [11].

$$\Delta u(n+i) = \sum_{j=1}^{p} \omega^j \Delta u^j(n+i) \tag{4.3}$$

Eşitlik **(4.3)**' de görülen *j*' nci kontrol artımına ait ateşleme çarpanı, *j*' nci alt modele ait ateşleme çarpanı ile birebir aynıdır. Aynı olması son derece doğaldır çünkü *j*'nci bulanık çıkarsama tarafından tahmin edilen çıkış değerinin sürece katkısı, *j*'nci kontrol çıkışının kontrol olayına katkısı ile aynı olmalıdır [11].

Yukarıdaki bilgiler ışığında eşitlik **(2.6)** ve **(4.3)**, eşitlik **(4.2)**' nin içine yerleştirilecek olursa aşağıdaki denklem elde edilir.

$$J(n)=\sum_{i=1}^{N_y}\mu_i\left(\sum_{j=1}^{p}\left(\omega^j\left(\hat{y}^j(n+i)-y^r(n+i)\right)\right)\right)^2+\sum_{i=1}^{N_u}\nu_i\left(\sum_{j=1}^{p}\omega^j\Delta u^j(n+i)\right)^2 \qquad \textbf{(4.4)}$$

Eşitlik **(4.4)**' da verilen eniyileme probleminde alt sistemler arası ilişkiler bulunduğundan, çözümü için gerekli hesap yükü oldukça fazladır. Bu hesap karmaşasını ortadan kaldırabilmek üzere, alternatif bir amaç fonksiyonu önerilmiştir [11].

Cauchy eşitsizliğine göre aşağıdaki ilişkiler doğrudur.

$$\left(\sum_{j=1}^{p}\left(\omega^j\left(\hat{y}^j(n+i)-y^r(n+i)\right)\right)\right)^2\le p\sum_{j=1}^{p}\left(\omega^j\left(\hat{y}^j(n+i)-y^r(n+i)\right)\right)^2 \qquad \textbf{(4.5)}$$

$$\left(\sum_{j=1}^{p}\omega^j\Delta u^j(n+i)\right)^2\le p\sum_{j=1}^{p}\left(\omega^j\Delta u^j(n+i)\right)^2 \qquad \textbf{(4.6)}$$

Eşitsizliklerden görüldüğü üzere hataların karelerinin ağırlıklı toplamları, orijinal amaç fonksiyonu için bir üst sınır oluşturabilmektedir. Bu da alternatif bir amaç fonksiyonu tanımına izin vermektedir [11].

$$\tilde{J}(n)=\sum_{i=1}^{N_y}\sum_{j=1}^{p}\mu_i\left(\omega^j\left(\hat{y}^j(n+i)-y^r(n+i)\right)\right)^2+\sum_{i=1}^{N_u}\sum_{j=1}^{p}\nu_i\left(\omega^j\Delta u^j(n+i)\right)^2 \qquad \textbf{(4.7)}$$

Eşitlik **(4.7)** aşağıdaki gibi de ifade edilebilir.

$$\tilde{J}(n)=\sum_{j=1}^{p}\left(\left(\omega^j\right)^2\left(\sum_{i=1}^{N_y}\mu_i\left(\hat{y}^j(n+i)-y^r(n+i)\right)^2+\sum_{i=1}^{N_u}\nu_i\Delta u^j(n+i)^2\right)\right) \qquad \textbf{(4.8)}$$

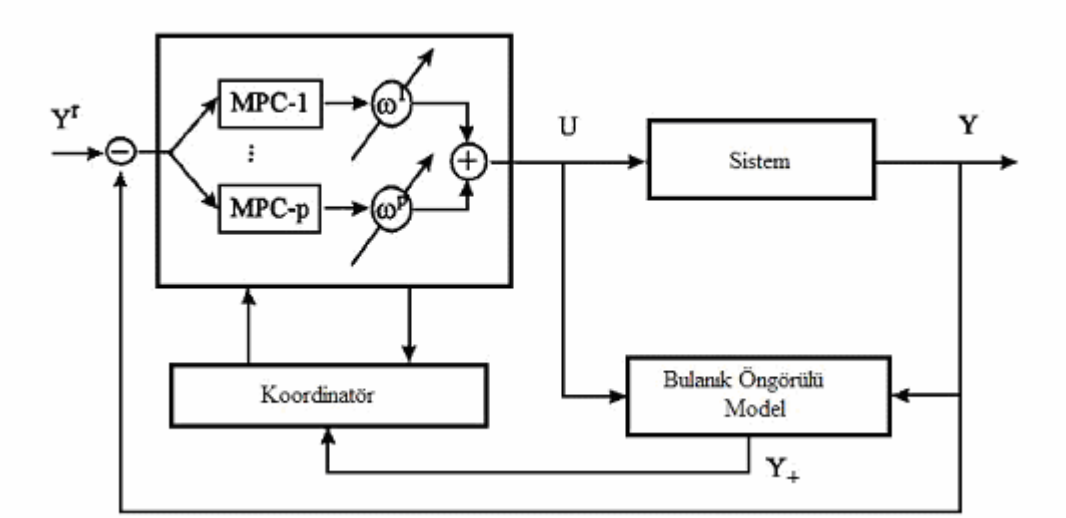
Yukarıda ifade edilen denklemler ışığında, eniyileme problemi aşağıdaki gibi yazılır.

$$\min_{\Delta u(n),\Delta u(n+1),\ldots,\Delta u(n+N_u)} \sum_{j=1}^{p} (\omega^j)^2 \tilde{J}^j(n) \tag{4.9}$$

$$\tilde{J}^j(n) = \sum_{i=1}^{N_y} \mu_i \left(\hat{y}^j(n+i) - y^r(n+i)\right)^2 + \sum_{i=1}^{N_u} \upsilon_i \left(\Delta u^j(n+i)\right)^2 \tag{4.10}$$

Önerilen alternatif amaç fonksiyonu kullanılarak, referans [11], hiyerarşik bir kontrol tasarım yaklaşımı tanımlamaktadır. Önerdiği yöntem ile, alt sistemlere ait en iyi kontrol artımlarını hesaplamakta ve **(4.53)**' de verildiği üzere bunları kaynaştırıp en iyi kontrol işaretini üretmektedir.

Ancak bu çalışmada, önerilen kontrol yapısı kullanılmış, fakat hiyerarşik yapı birebir uygulanmamıştır. Buna göre makalede önerilen FMPC kontrolör yapısı aşağıdaki gibi verilmiştir.




**Şekil 4.1:** FMPC kontrolör yapısı [11]

Şekil 4.1' de görüldüğü üzere, öncelikle sistem doğrusal-benzeri alt sistemlerin bulanık kaynaşması biçiminde modellenmiştir. Bulunan doğrusal alt sistemlere uygun MPC kontrolörler tasarlanmıştır. MPC kontrolörlerin ürettiği kontrol işaretleri, modelleme sırasında kullanılan bulanık kaynaşma metoduyla kaynaştırılmış ve kontrol işareti üretilmiştir.

# 5. BULANIK MODEL ÖNGÖRÜLÜ KONTROL UYGULAMALARI

FMPC algoritmasını incelemek amacıyla referans [11]' de verilen doğrusal olmayan sistem incelenmiştir. Verilen sistemlere uygun FMPC tasarlanmış ve önerilen yapı, klasik kontrol yöntemleri ile karşılaştırılmıştır. FMPC yapısının giriş ve çıkış bozucularına tepkileri incelenmiştir. Ayrıca doğrusal olmayan sıcaklık deney seti PT326' dan veri alınarak, FMPC yapısı uygulanmış ve sistemin performansı gözlenmiştir.

## 5.1 Benzetim Çalışmaları

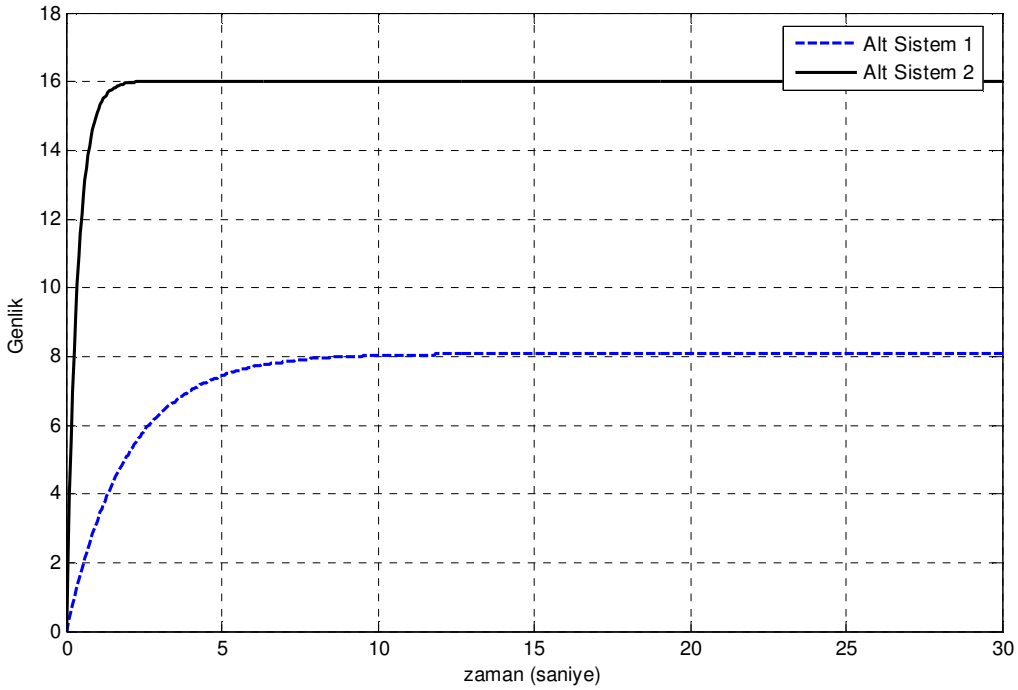
### 5.1.1 Doğrusal olmayan sistem bulanık modellemesi

Referans [11]'de verilen örnek 1' de sistem dinamiği, birim basamak cevabı olarak verilmiştir. Birim basamak cevabına bakılarak sistemin belli bölgelerde, bazı birinci mertebeden sistemlere benzeyen bir davranış sergilediği görülmüştür. Özellikle de ilk 1 saniye boyunca yavaş bir sistem cevabı olduğu ve 3. saniyeye gelindiğinde hızlı bir yükseliş yaptığı gözlenmiştir. Bu durumdan yola çıkılarak sistemin davranışı iki adet alt sistemin bulanık olarak kaynaşması biçiminde ifade edilmiştir. Bu durumda alt sistem adeti olan $p$ değişkeni 2 olarak seçilmiştir. Verilen alt sistemlerin fark denklemleri aşağıdaki gibidir.

$$y_1(k) = 0.202u(k-1) + 0.975y(k-1) \quad \textbf{(5.1)}$$

$$y_2(k) = 2.13u(k-1) + 0.867y(k-1) \quad \textbf{(5.2)}$$

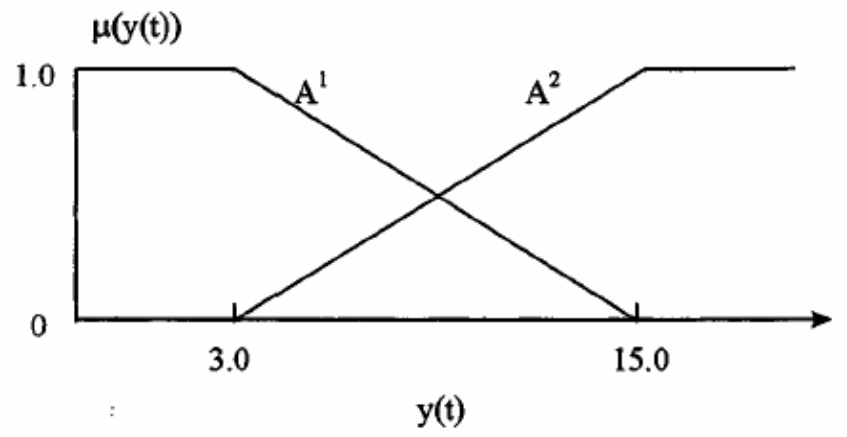
Alt sistemlerin MATLAB Simulink' te gerçeklenen birim basamak cevapları Şekil 5.1' deki gibidir.




**Şekil 5.1:** Alt sistem birim basamak cevapları

Görüldüğü üzere her iki sistem de birinci mertebeden sistemlerdir ve referans [11]' de verilen örnek 1 için tüm işlemler ayrık zamanda ve örnekleme zamanı 0.05 saniye seçilerek yapılmıştır.

Birim basamak cevabına bakıldığında, doğrusal olmayan sistemin çıkış değeri 3' e gelinceye kadar birinci alt sistem gibi davrandığı, çıkış değeri 15' i geçtiğinde ise ikinci alt sistem gibi davrandığı gözlenmiştir. 3 ile 15 değerleri arasında ise iki alt sistemin kaynaşması biçiminde bir davranış izleyeceği düşünülmüştür. Doğrusal olmayan sistemin modeli, Şekil 5.1'de verilen alt sistemlerin bulanık olarak kaynaşmaları sonucu elde edilmiştir. Bulanık kaynaşma için bulanık kümeleri tanımlamak gerekmektedir. İki alt sistem olduğuna göre iki adet bulanık küme bulunmaktadır ve bu kümeler $A^1$ ve $A^2$ Şekil 5.2'de verildiği gibi ifade edilmiştir.




**Şekil 5.2:** Bulanık kümeler ve tanımladıkları üyelik fonksiyonları

Doğrusal olmayan sistemin bulanık modeli iki adet bulanık çıkarsamadan $R^1$ ve $R^2$ meydana gelmektedir ve bu çıkarsamalar alt sistemlerin fark denklemlerine göre aşağıdaki gibi tanımlanmıştır:

$R^1$ : Eğer $y(k)$ $A^1$ ise o halde $y_1(k+1) = 0.202u(k) + 0.975y(k)$ **(5.3)**

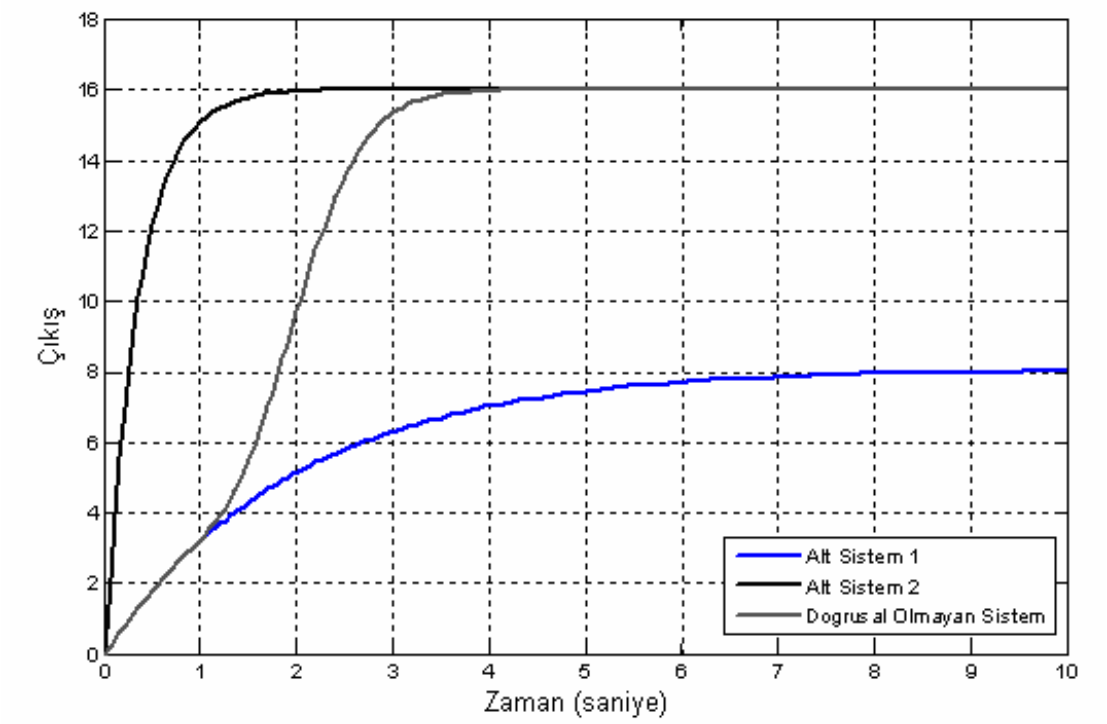
$R^2$ : Eğer $y(k)$ $A^2$ ise o halde $y_2(k+1) = 2.13u(k) + 0.867y(k)$ **(5.4)**

Yukarıda verilen bulanık çıkarımlar, ağırlıklı ortalama ile durulanacak olursa doğrusal olmayan sistem modeline ait çıkış denklemi, **(5.6)**' daki gibi olacaktır:

$$y(n+1) = \sum_{j=1}^{p} w^j y^j(n+1)$$

$$y(k+1) = w_1 y_1(n+1) + w_2 y_2(n+1) \quad \textbf{(5.5)}$$

Yukarıda anlatıldığı gibi alt sistemlerin kaynaştırılması, MATLAB Simulink ile kullanıcı tanımlı fonksiyonlara dönüştürülüp, blok içerisine gömülmüş ve sistem modelinin birim basamak giriş cevabı alınmıştır. Tasarlanan fonksiyon yazılımı Ek' te verilmiştir. Modellenen sistemim birim basamak cevabı Şekil 5.3' te verilmiştir.




**Şekil 5.3.** Doğrusal olmayan sistem modeli birim basamak cevabı

### 5.1.2 Model öngörülü kontrolör tasarımı

Referans [11]'nin önerdiği gibi her bir alt sistemin modeline uygun MPC kontrolörleri tasarlanmıştır. Bu amaçla, Bölüm 3' te anlatılan kontrolör yapısı aynen uygulanmıştır. Oluşturulan MPC yazılımı yine MATLAB' da gömülü fonksiyon bloklarına dönüştürülerek MPC kontrolörler tanımlanmıştır. MPC kontrolörlerin yazılımı oluşturulurken, referans [2]' nin "basicmpc" adlı kodundan yola çıkılmıştır. Ayrıca sistemin modelleme hatalarına karşı direncini arttırmak üzere, yine referans [2]' de önerilen IMC yapısı eklenmiştir.

Her alt sistemin modeli farklı olduğu için kontrolör yazılımları da farklıdır. Ancak yazılım, farklı bir modele çok çabuk adapte olacak şekilde geliştirilmiştir. Kullanıcı, sadece model transfer fonksiyonunu, öngörü ve kontrol ufuklarını, örnekleme zamanı, üstel fonksiyonun zaman sabitini ve ağırlık matrislerini değiştirerek istediği sistemi kontrol edebilmektedir. Tasarlanan MPC yazılımları Ek' te verilmiştir.

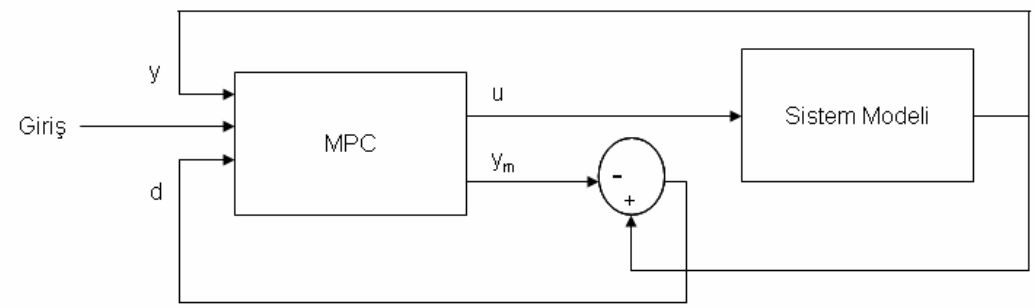
MPC kontrolörleri tasarlanırken belirlenen bazı parametreler bulunmaktadır ki bunlar sistem dinamiği üzerinde önemli rollere sahiptir. Bu uygulama için öngörü ufku $H_p=15$, kontrol ufku $H_u = 10$ seçilmiştir. Bu parametrelerin seçilmesinde bazı ayar kuralları göz önüne alınmıştır. Kontrol ufku arttıkça daha kararlı bir sonuç alınırken

hesap yükü artmaktadır. Öngörü ufku, gelecek kontrol olaylarını hesaplayarak, öngörülen hataları azaltıcı bir etkiye sahiptir. Dolayısıyla bu parametrenin büyük seçilmesi hataları azaltır ancak hesap yükünü arttırır. Öngörü ufkunun küçük seçilmesi ise daha dayanıklı bir kontrolör yapısına sebep olur fakat model hatalarına karşı hassasiyet artar [11]. Bu sebeplerden dolayı tüm çelişkiler düşünülerek yukarıda verilen değerler seçilmiştir.

Bu parametrelerden başka örnekleme zamanı $T_s = 0.05$ saniye ve referans yörüngesi için gerekli olan üstel fonksiyonun zaman sabiti $T_{ref} = 0.5$ olarak verilmiştir.

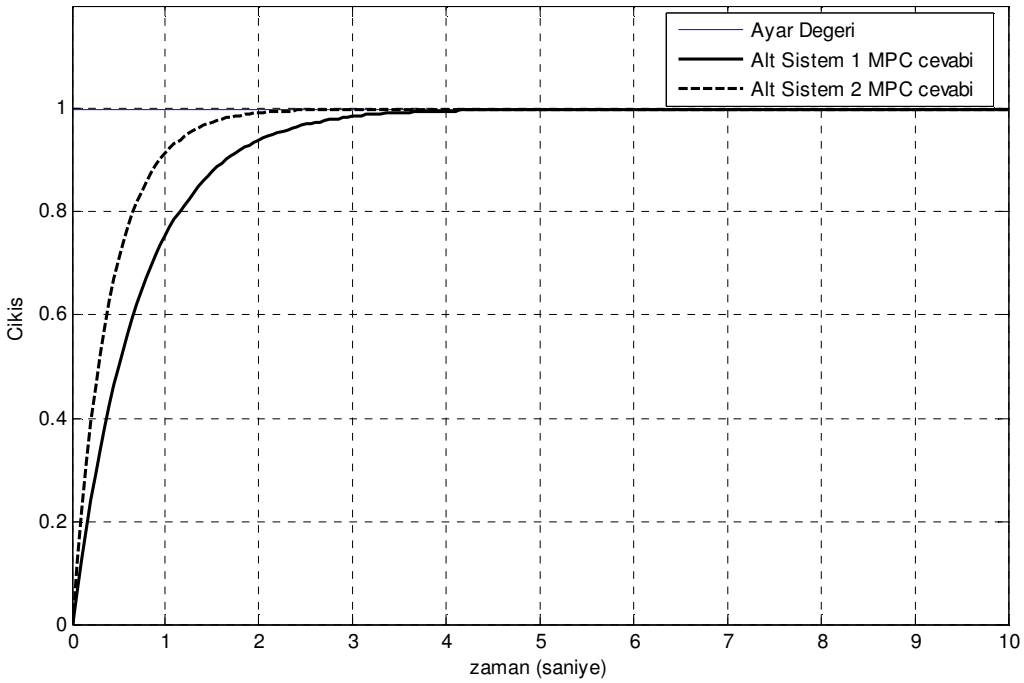
Başarım ölçütü için gerekli olan cezalandırma katsayılarının hesaplanması için sistematik bir yaklaşım bulunmamaktadır. Bazı deneysel yaklaşımlar mevcuttur [11] ve burada en uygun sonucu veren ağırlık matrisleri, birinci kontrolör için 0.3, ikinci kontrolör için 0.5 olarak seçilmiştir.

Modelleme hatalarına karşı kontrolörün direncini arttırmak üzere, referans [2]' de anlatılan IMC yapısı sisteme eklenmiştir. Her adımda sistemin ölçülen çıkış değeri ve modele göre hesaplanan çıkış değerleri arasındaki fark alınmış ve değer fonksiyonun içine eklenerek, oluşabilecek modelleme hataları ortadan kaldırılmıştır. Sisteme eklenen IMC yapısı aşağıdaki şekilde görülebilir:




**Şekil 5.4:** MPC sistemine eklenen IMC yapısı

Alt sistemlerin kendilerine uygun MPC kontrolörleri tasarlandıktan sonra, her alt sistemin kapalı çevrim kontrolleri yapılmıştır. Alınan sonuçlar aşağıdaki gibidir:

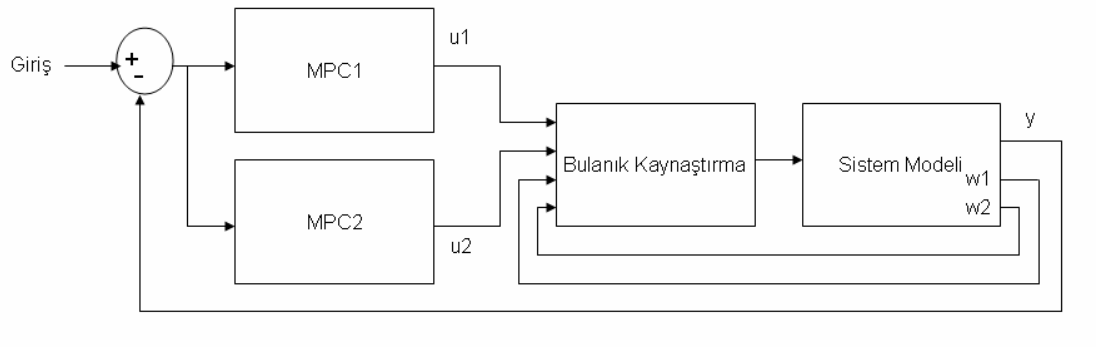



**Şekil 5.5:** Alt sistemlerin MPC ile kontrol edilmesi

### 5.1.3 Model öngörülü kontrolörlerin bulanık kaynaşmaları

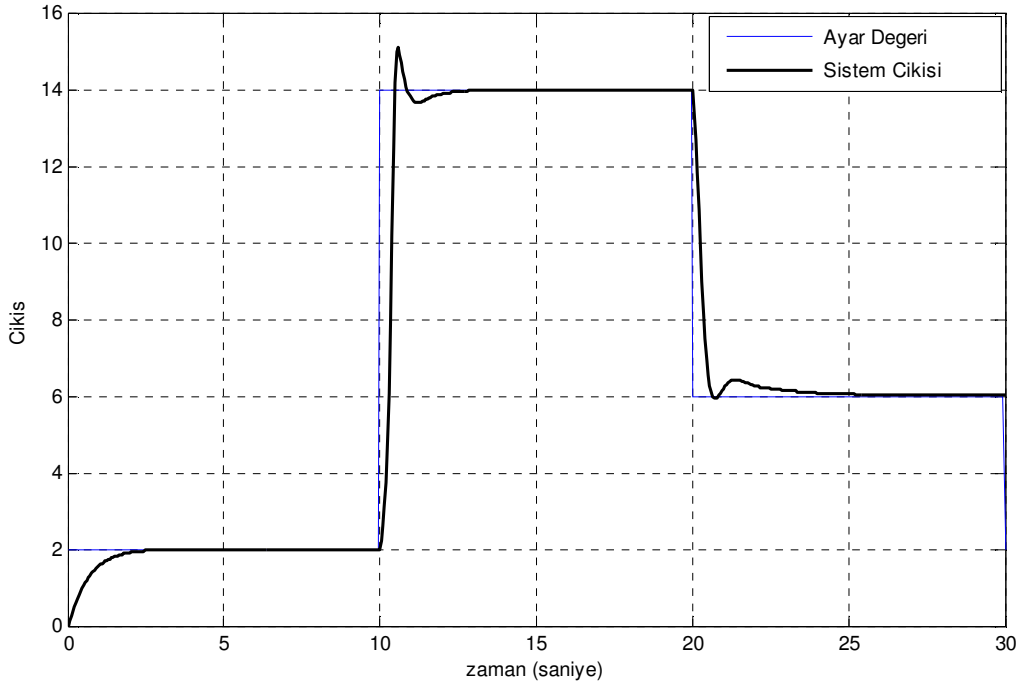
Her alt sisteme ait MPC kontrolör tasarlandıktan sonra, elde edilen iki adet kontrol çıkışı, doğrusal olmayan sisteme verilmek üzere kaynaştırılmıştır. Bu kaynaşma işlemi, iki alt sistemin bulanık olarak kaynaşması ile birebir aynıdır ve eşitlik (4.3)' te verildiği gibidir.

Bu işlem sonucu hesaplanan kontrol işareti doğrusal olmayan sisteme uygulanmıştır. Kontrol sisteminin yapısı MATLAB Simulink' te blok diyagramlar biçiminde aşağıdaki gibi tasarlanmıştır.




**Şekil 5.6:** Bulanık MPC ile sistem kontrolü blok şeması

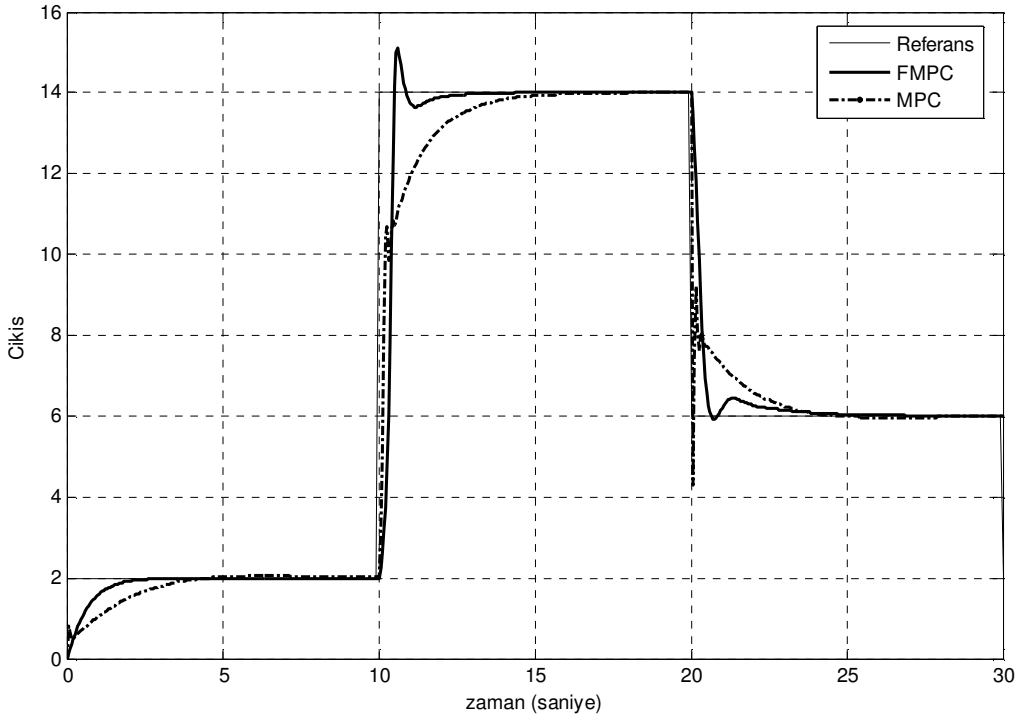
Yukarıda verilen kontrol yapısı, farklı giriş değerleri için denenmiş ve sistemin verilen referans değerlere tepkileri gözlenmiştir. FMPC yapısının doğrusal olmayan sistem kontrolü aşağıdaki şekilde verilmiştir.




**Şekil 5.7:** FMPC ile doğrusal olmayan sistem kontrolü

Şekil 5.7' de görüldüğü üzere, önerilen yöntem, ayar değeri değişikliklerine çabuk tepki verebilmiş ve sistemi kısa sürede kararlı hale geçirebilmiştir.

FMPC yapısının klasik MPC yapısına göre davranışını incelemek amacıyla, bulanık olarak kaynaştırılan sisteme MPC kontrolör uygulanmıştır. MPC kontrolörün tasarımı için gerekli olan model yapısı doğrusal olmayan sistemin dört parametre yöntemi ile ikinci mertebeden modeli çıkarılarak elde edilmiştir. Bu modele göre tasarlanan MPC kontrolör doğrusal olmayan sisteme uygulanmış ve FMPC yapısı ile karşılaştırılmıştır.



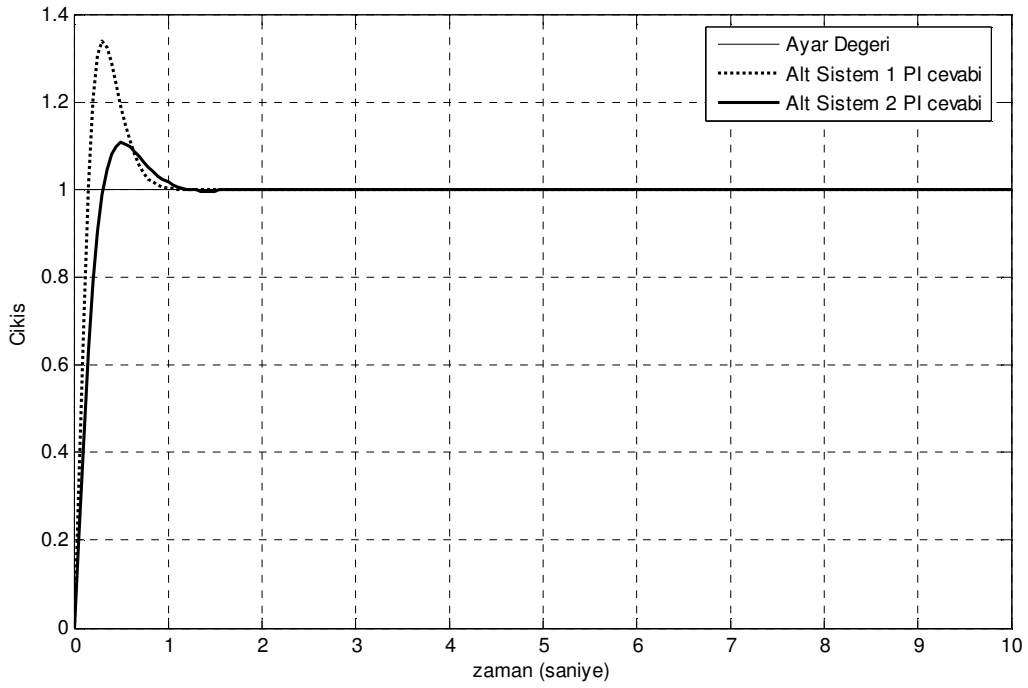

**Şekil 5.8:** FMPC ve MPC ile doğrusal olmayan sistem kontrolü

Görüldüğü üzere, doğrusal olmayan sistemin FMPC kontrolü çok daha hızlı ve salınımsız bir cevap vermektedir. Bunun sebebi MPC yapısının en uygun cevapları en uygun sistem modeli varlığında verebilmesidir.

Referans [11]' de önerilen kontrol yapısı, iki adet MPC' nin bulanık kaynaşması yerine iki adet PI kontrolörün kaynaşması şekline dönüştürülüp, önerilen sistemin klasik yöntemlere göre nasıl bir davranış sergileyeceği gözlenmiştir. Bu sebeple, her alt sistem için bir PI kontrolör tasarlanmıştır. Alt sistemler birinci mertebeden olduğu için PI kontrolörler tasarlanması uygun görülmüştür. Alt sistemlere uygun PI katsayıları, MPC' nin eniyileme yöntemi en küçük kareler olması sebebiyle, hatanın karesini minimize edecek şekilde seçilmiştir. PI katsayıları, MATLAB' ın genetik algoritması kullanılarak bulunmuştur. Buna göre elde edilen PI katsayıları aşağıdaki gibidir:

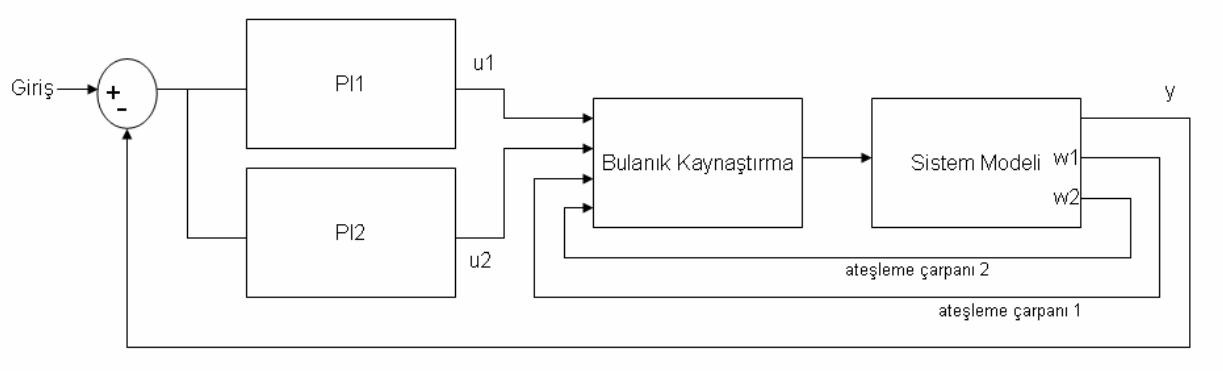
| PI Katsayıları | Alt Sistem 1 | Alt Sistem 2 |
|---|---|---|
| Oransal Katsayı | 1.67793 | 0.10012 |
| Entegral Katsayısı | 4.52156 | 0.40142 |

Alt sistemlerin kendilerine uygun PI kontrolörleri tasarlandıktan sonra, her alt sistemin kapalı çevrim kontrolleri yapılmıştır. Alınan sonuçlar aşağıdaki gibidir:



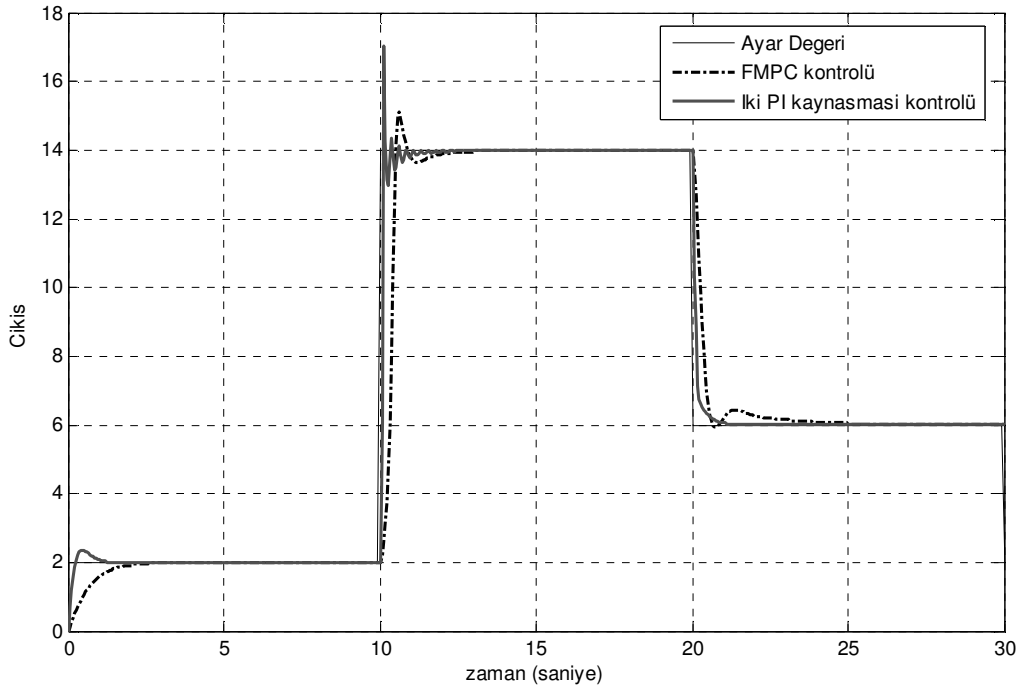

**Şekil 5.9:** Alt sistemlerin PI kontrolörlerle kontrol edilmesi

PI kontrolörler tasarlandıktan sonra, iki MPC kontrolörün kaynaştırılmasında olduğu gibi, iki PI kontrolör de bulanık olarak kaynaştırılmıştır. Kontrol işaretleri kaynaştırıldıktan sonra doğrusal olmayan sisteme verilmiştir. MATLAB Simulink' te gerçeklenen PI kontrolörlerin kaynaşması ile doğrusal olmayan sistem kontrolü şeması aşağıdaki gibidir.




**Şekil 5.10:** PI kontrolörlerin kaynaşması ile doğrusal olmayan sistem kontrolü blok şeması

Şekil 5.6' da verilen FMPC yapısı ve Şekil 5.10' da verilen iki adet PI kontrolörün bulanık kaynaşması ile doğrusal olmayan sistem kontrolü sonuçları karşılaştırılmıştır. Gözlenen sonuçlar Şekil 5.11' da verilmiştir.

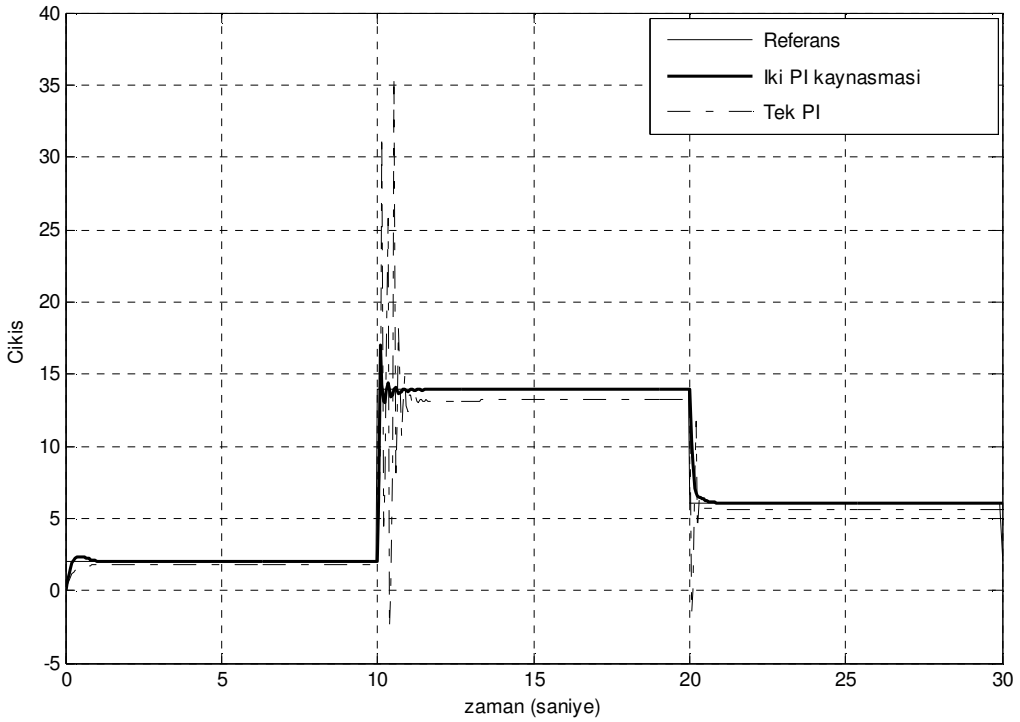



**Şekil 5.11:** FMPC ve PI kontrolörler ile doğrusal olmayan sistem kontrolü

Şekil 5.11' da görüldüğü üzere, önerilen FMPC yapısı, klasik PI kontrolörlerin kaynaşması ile oluşturulan kontrolöre göre daha az aşım yapmaktadır. Ancak burada unutulmamalıdır ki, PI parametreleri hesaplanırken hatanın karesini minimize etmeye çalışılmış, ve böylece MPC yapısı ile eşit şartlarda karşılaştırma yapılmıştır. Daha

farklı bir metotla, PI katsayıları hesaplanır ve bu kontrolörler kaynaştırılacak olursa, FMPC yapısından daha iyi bir performans sergileyeceği muhakkaktır.

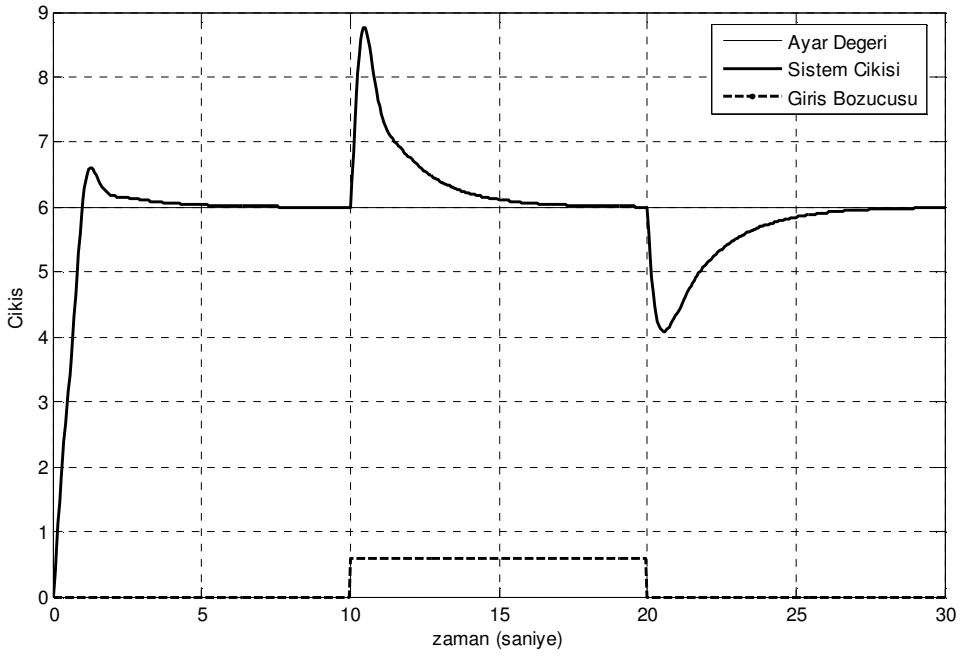
İki PI kontrolörün bulanık olarak kaynaşması sonucu elde edilen sonuçlar, tek PI kontrolör ile doğrusal olmayan sistem kontrolü ile de karşılaştırılmıştır. Buna göre sisteme en uygun PI katsayıları 1.003 ve 0.1211 olarak genetik algoritma yardımı ile hesaplanmış ve doğrusal olmayan sisteme uygulanmıştır. Alınan sonuçlar iki adet PI kontrolörün bulanık olarak kaynaşması sonucu ile karşılaştırılmıştır.




**Şekil 5.12:** İki PI kontrolör kaynaşması ve tek PI kontrolör ile doğrusal olmayan sistem kontrolü

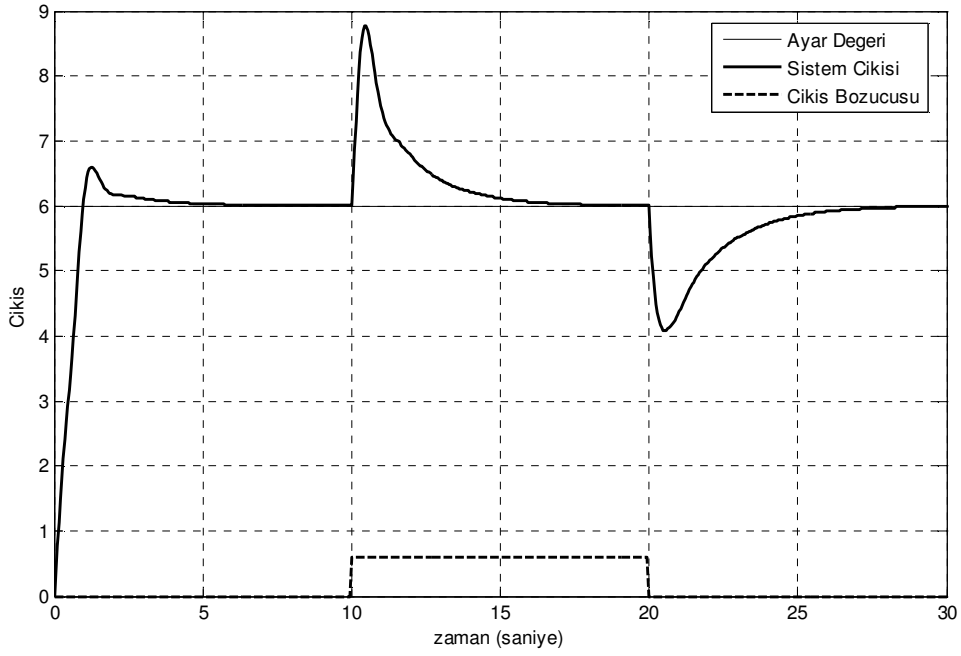
Şekilde görüldüğü üzere sistemin alt modellere ayrılması ile doğrusal olmayan sistem kontrolü daha düzgün bir performans sergilemektedir.

Önerilen FMPC yapısının, giriş ve çıkış bozucularına verdiği tepkiler de incelenmiştir. Bu amaçla ayar değeri 6 iken (çıkış değerinin 3 ile 15 arasında olması her iki alt sistemin de ateşlendiği anlamına gelmektedir) sisteme, ayar değerinin %10' u büyüklükte giriş bozucusu 10 saniyelik bir süreç boyunca eklenmiş ve sistemin davranışı gözlenmiştir. Buna göre sistem Şekil 5.13' de verildiği gibi bir davranış sergilemiştir.




**Şekil 5.13:** FMPC ile doğrusal olmayan sistem kontrolü (giriş bozucusu etkisi )

Aynı işlem bu kez sistem çıkışına uygulanmış, ve FMPC' nin çıkış bozucusuna verdiği tepki gözlenmiştir. Ayar değeri yine 6 olarak seçilmiş ve çıkışa uygulanan bozucunun genliği de yine 0.6 olarak verilmiştir. Sistemin gösterdiği davranış aşağıdaki gibidir.



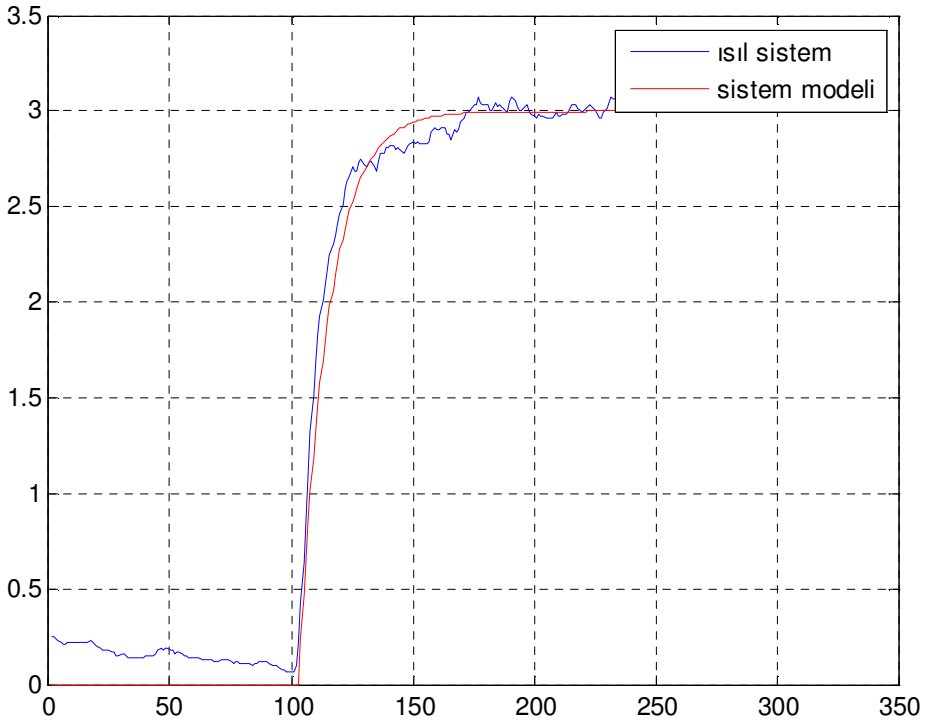

**Şekil 5.14:** FMPC ile doğrusal olmayan sistem kontrolü (çıkış bozucusu etkisi )

## 5.2 Gerçek Zaman Uygulaması

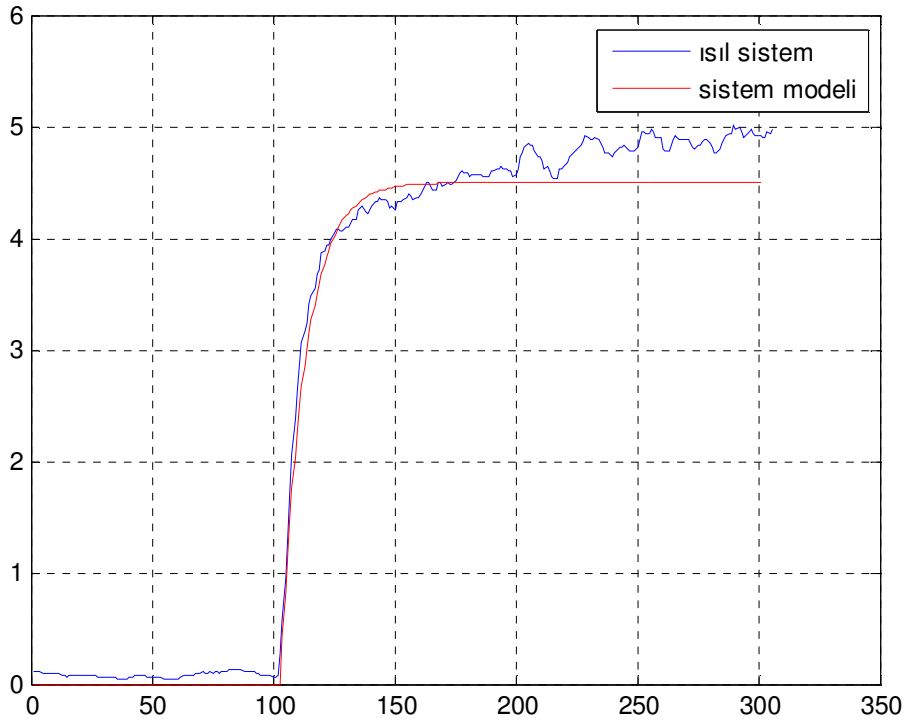
PT326 Sıcaklık deney seti, daha önce pek çok uygulama ile kontrol edilmiş ve matematiksel modelinin giriş değerine göre değiştiği gözlenmiş bir sistemdir. Bu uygulama için de iki farklı giriş değeri verilmiş ve farklı kazançlara sahip iki adet birinci mertebeden sistem verisi elde edilmiştir. Buna göre sisteme sırasıyla 4.5 ve 7 girişleri verildiğinde birinci ve ikinci sistemin matematiksel modelleri aşağıdaki gibi gözlenmiştir:

$$G_1(s) = \frac{0.6667}{1.2s+1} \cdot e^{-2s} \tag{5.6}$$

$$G_2(s) = \frac{0.6429}{s+1} \cdot e^{-2s} \tag{5.7}$$

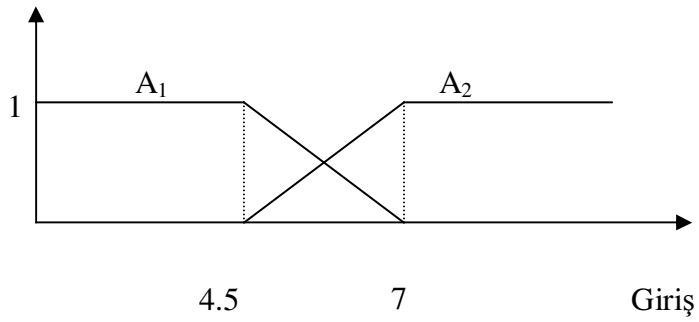



**Şekil 5.15:** Ayar değeri 4.5 V (yaklaşık 38$^{o}$ C) için ısıl sistem ve matematiksel model cevabı




**Şekil 5.16:** Ayar değeri 7 V (yaklaşık 50$^{o}$ C) için ısıl sistem ve matematiksel model cevabı

Sistemin doğrusal olmayan yapısını, eşitlik **(5.6)** ve **(5.7)**' de verilen transfer fonksiyonların bulanık kaynaşması biçiminde ifade etmek üzere, bulanık kaynaşma için gerekli olan bulanık alt kümeler ifade edilmiştir. Ancak burada, önceki bulanık kaynaştırmadan farklı olarak, bulanık ateşlemeler çıkış değerine göre değil, giriş değerine bakılarak hesaplanmaktadır. Bulanık üyelik fonksiyonları, aşağıdaki gibi elde edilmiştir.



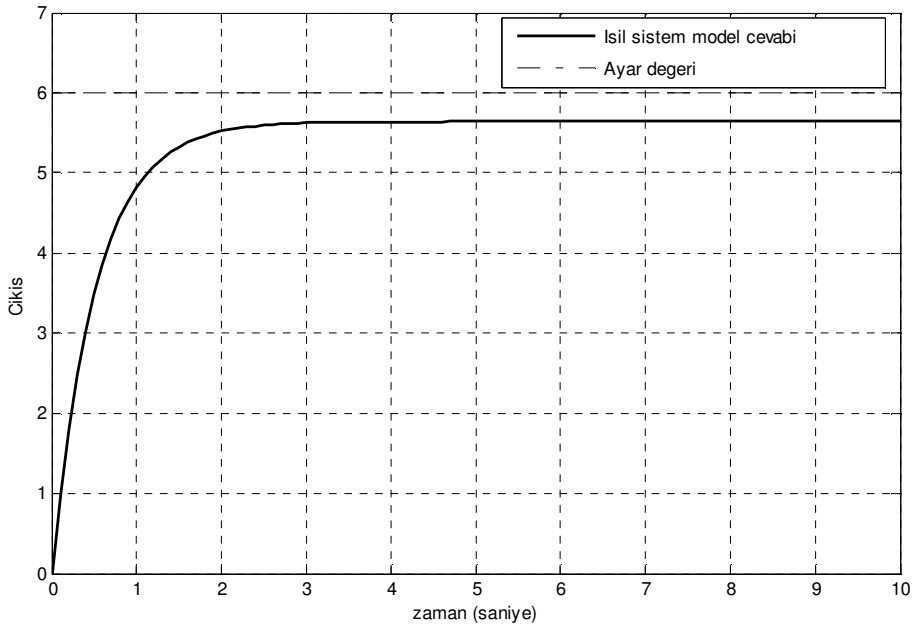

**Şekil 5.17:** Isıl sistem üyelik fonksiyonları

Alt sistemlerin ayrık zaman fark denklemlerini hesaplamak üzere örnekleme zamanı $T_s = 0.1$ *saniye* seçilmiştir. Bulunan fark denklemleri eşitlik **(5.8)** ve **(5.9)**' daki gibidir:

$$y_1(k+1) = 0.05331u(k) + 0.92y_1(k) \quad \textbf{(5.8)}$$

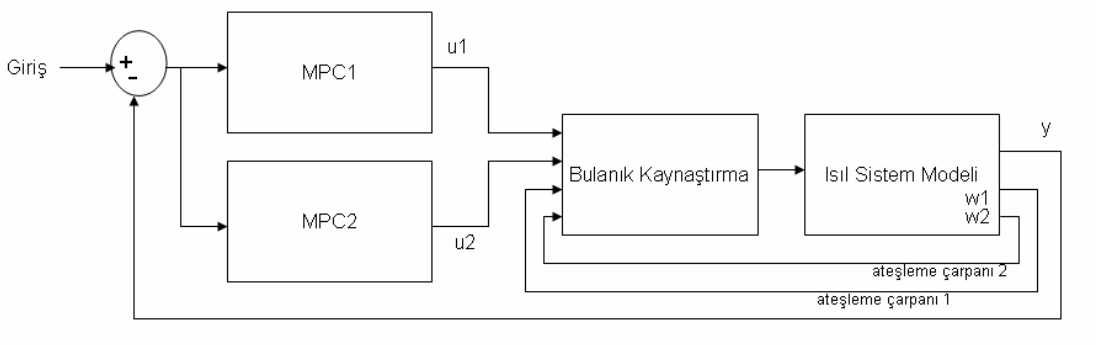
$$y_2(k+1) = 0.061181u(k) + 0.9048y_2(k) \quad \textbf{(5.9)}$$

Eşitlik **(5.8)**, **(5.9)** ve **Şekil 5.17**' de verilen üyelik fonksiyonları kullanılarak, MATLAB' da kullanıcı tanımlı fonksiyonlar tasarlanmış ve yazılım Ek' te verilmiştir. Bulanık kaynaşma sonucu elde edilen sistem modelinin ayar değeri 6 için verdiği basamak cevabı aşağıdaki gibidir: ( 6 değeri, kaynaşma işleminin 4.5 ve 7 değerleri arasında olması ve kaynaşmanın görülebilmesi için seçilmiş bir giriş değeridir.)



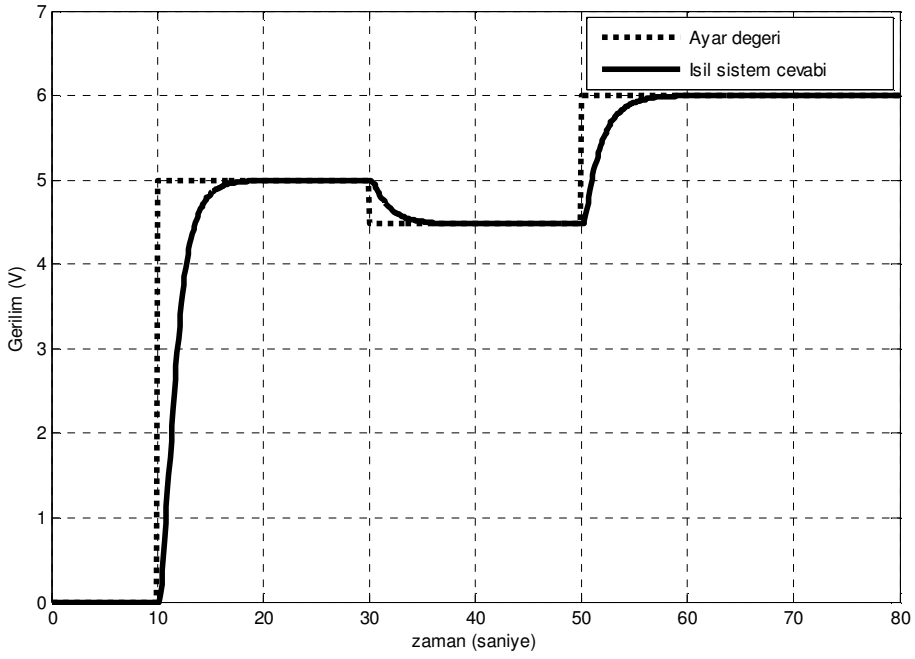

**Şekil 5.18:** Bulanık kaynaşma ile modellenen ısıl sistem basamak cevabı

Model elde edildikten sonra, alt sistemleri kontrol edebilecek ve alt sistemlerin modelini içeren MPC yapıları oluşturulmuştur. Yine MATLAB kullanıcı tanımlı fonksiyonlar tasarlanarak blok şema içine gömülmüş ve alt sistemlerin kontrolü yapılmıştır. Isıl sistem alt sistemlerine ait yazılımların daha önce Ek' te verilen alt sistem MPC yazılımlarından tek farkı, ayrık zaman transfer fonksiyonlarıdır. MATLAB Simulink' te oluşturulan kontrol blok şeması Şekil 5.19' daki gibidir:




**Şekil 5.19:** Isıl sistem modeli FMPC kontrolü blok şeması

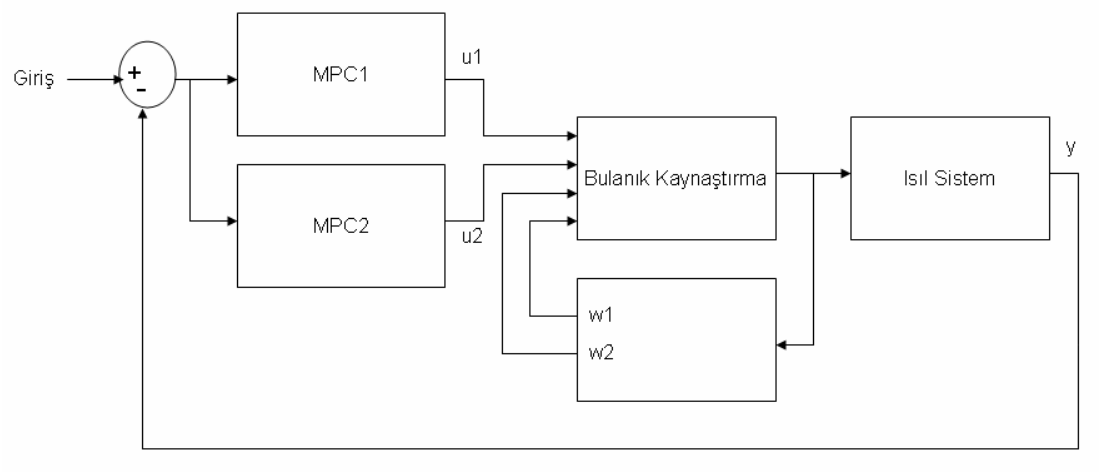
Önerilen yöntem Şekil 5.19' da görüldüğü gibi bulanık olarak kaynaştırılan sistem modeline uygulanmıştır. Modele farklı ayar değerleri verilmiş ve FMPC yapısının ısıl sistem modelini nasıl kontrol ettiği gözlenmiştir.




**Şekil 5.20:** Isıl sistem modeli FMPC kontrolü

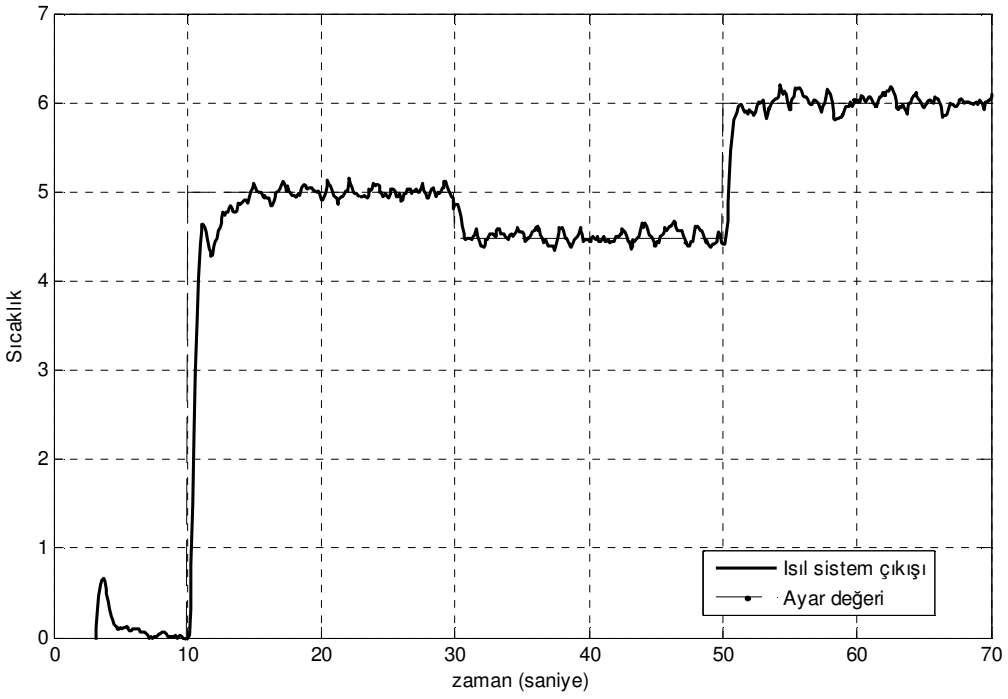
Şekil 5.20' de görüldüğü üzere, FMPC yapısı ısıl sistem modelini istenen ayar değerlerine hızlı bir şekilde ulaştırabilmektedir. Sistemin simülasyonu çalıştıktan sonra, önerilen FMPC yapısı, ısıl sisteme uygulanmıştır. Bu uygulama için MATLAB Simulink' te bulunan gerçek zamanlı analog giriş çıkış kutuları kullanılmıştır. Hesaplanan kontrol işareti analog çıkışa uygulanıp, analog giriş

kutusundan ısıl sistem çıkış değeri alınmış ve sisteme geri beslenmiştir. Simulink' te oluşturulan kontrol blok şeması Şekil 5.21' deki gibidir:




**Şekil 5.21:** Isıl sistem FMPC kontrolü blok şeması

Tasarlanan sisteme farklı ayar değerleri verilmiş ve FMPC yapısının ısıl sistemi nasıl kontrol ettiği gözlenmiştir. FMPC yapısı ile kontrol edilen ısıl sistemin verdiği cevaplar aşağıdaki gibi gözlenmiştir.

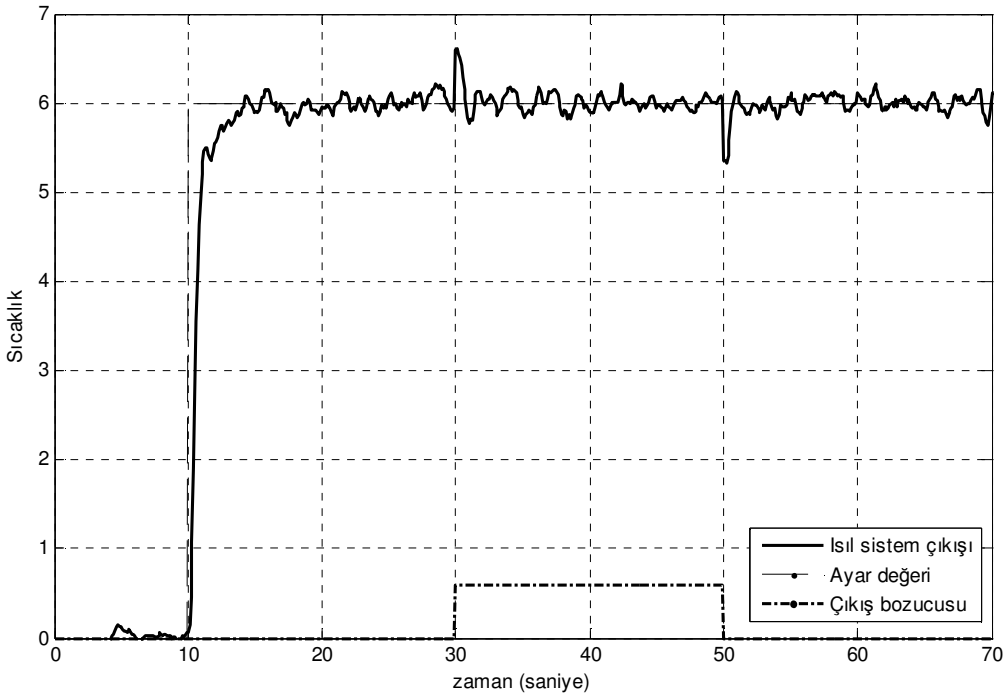



**Şekil 5.22:** Isıl sistem FMPC kontrolü

Şekil 5.22' de görüldüğü üzere FMPC yapısının, ısıl sistemi istenen ayar değerlere 10 saniyeden daha kısa bir sürede ulaştırabildiği, aşımın % 10' u geçmediği ve sistemin ölü zamanlı olmasına rağmen, kontrol işlemini gerçekleştirebildiği gözlenmiştir.

Isıl sistemin FMPC ile kontrolü gerçek zamanlı olarak yapıldıktan sonra, giriş ve çıkış bozucularının sistem üzerindeki etkileri gözlenmiştir.

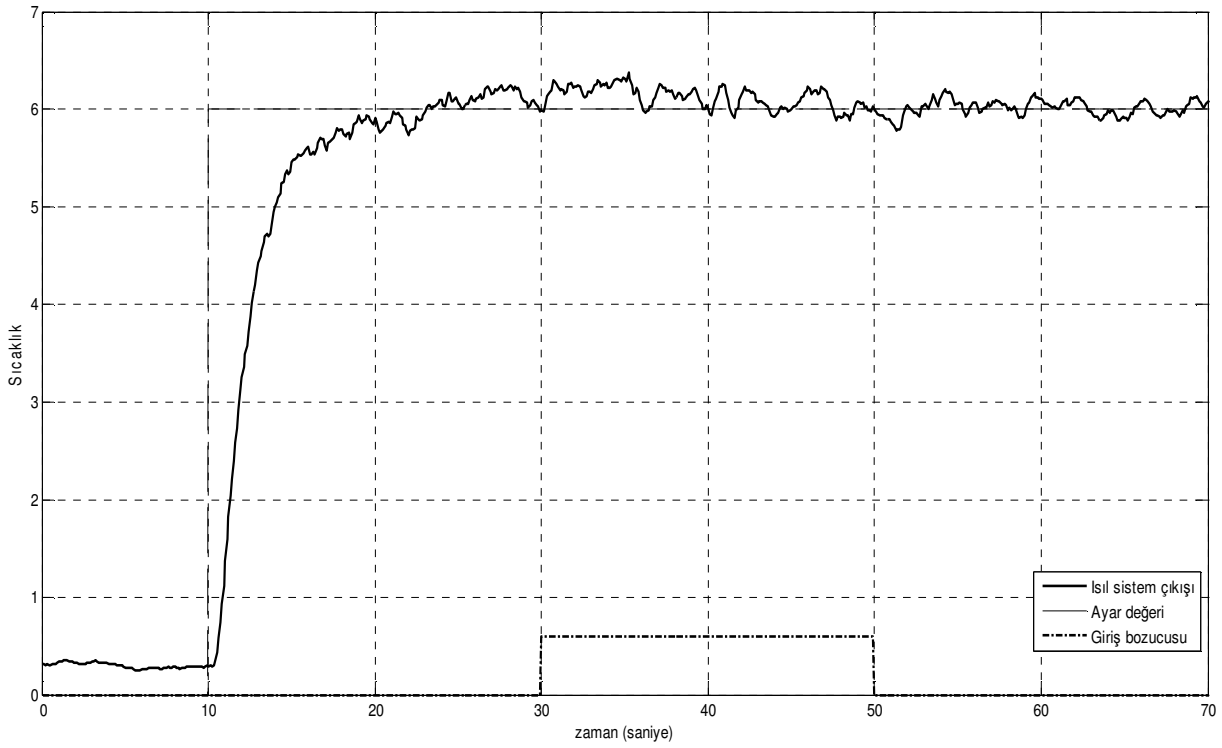
Sisteme ayar değeri olarak 6 Voltluk bir gerilim uygulanmıştır ve bu da yaklaşık olarak $45^{o}$ C' lik bir sıcaklık değerini ifade etmektedir. Daha sonra sistem çıkışına 20 saniye boyunca, ayar değerinin %10' u olan 0.6 V değerinde bir bozucu eklenmiştir ve FMPC yapısının verdiği tepki gözlenmiştir.




**Şekil 5.23:** Isıl sistem FMPC kontrolü (çıkış bozucusu etkisi)

Şekil 5.23' de görüldüğü üzere FMPC çıkış bozucusuna hızlı bir şekilde cevap verebilmekte ve sistemi istenen ayar değerine geri döndürebilmektedir.

Aynı ayar değeri için bu kez sisteme 0.6 V genliğinde bir giriş bozucusu 20 saniye boyunca eklenmiş ve sistemin davranışı gözlenmiştir.




**Şekil 5.24:** Isıl sistem FMPC kontrolü (giriş bozucusu etkisi)

Şekil 5.24' de görüldüğü üzere, FMPC yapısı giriş bozucusundan neredeyse hiç etkilenmemekte ve sistemi verilen ayar değerine çok hızlı bir şekilde oturtabilmektedir.

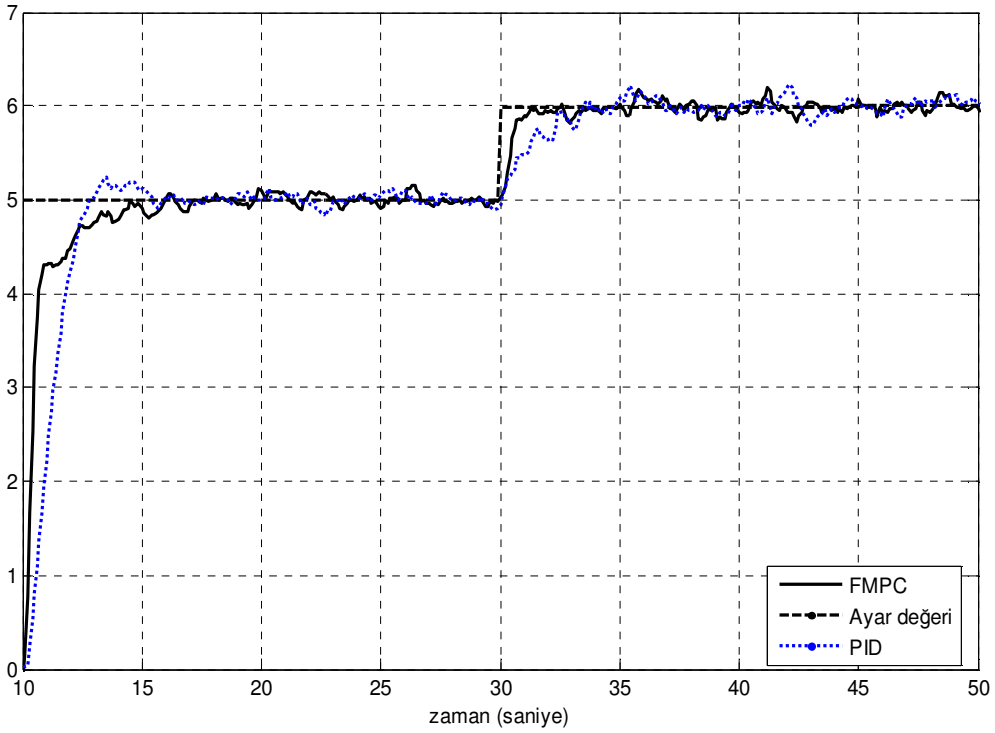
FMPC yapısının giriş ve çıkış bozucularına verdiği tepkileri gözlemledikten sonra, klasik bir kontrol yöntemine kıyasla nasıl davrandığı gözlenmiştir. Bu amaçla bulanık olarak kaynaştırılan sistem modeline uygun bir PID kontrolör tasarlanmıştır. PID kontrolörün katsayıları FMPC yapısının kullandığı hatanın karelerinin toplamını minimize edecek şekilde belirlenmiştir ve katsayılar MATLAB "genetik algoritma" kullanılarak bulunmuştur. Buna göre sistem modeline uygun PID katsayıları aşağıdaki gibidir:

Oransal katsayı = 0.29965

Entegral katsayı = 0.83924

Türev katsayısı = 0.17703

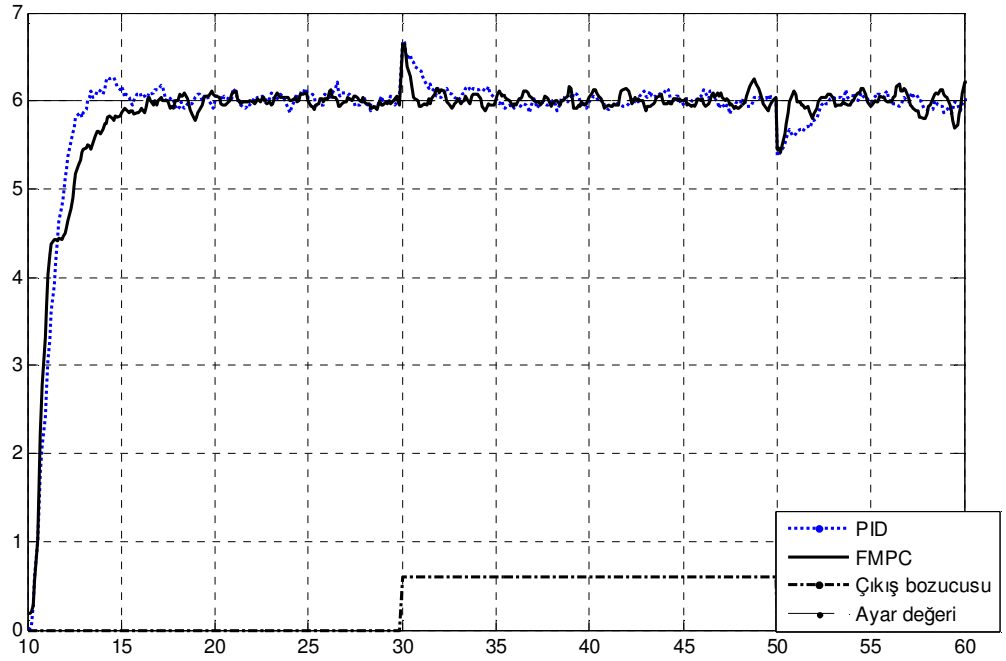
Hesaplanan PID katsayıları ile PID kontrolör tasarlanmış ve ısıl sisteme gerçek zamanlı olarak uygulanmıştır. Farklı ayar değerleri için FMPC yapısı ve ısıl sistem PID kontrol cevabı aşağıdaki gibi gözlenmiştir.



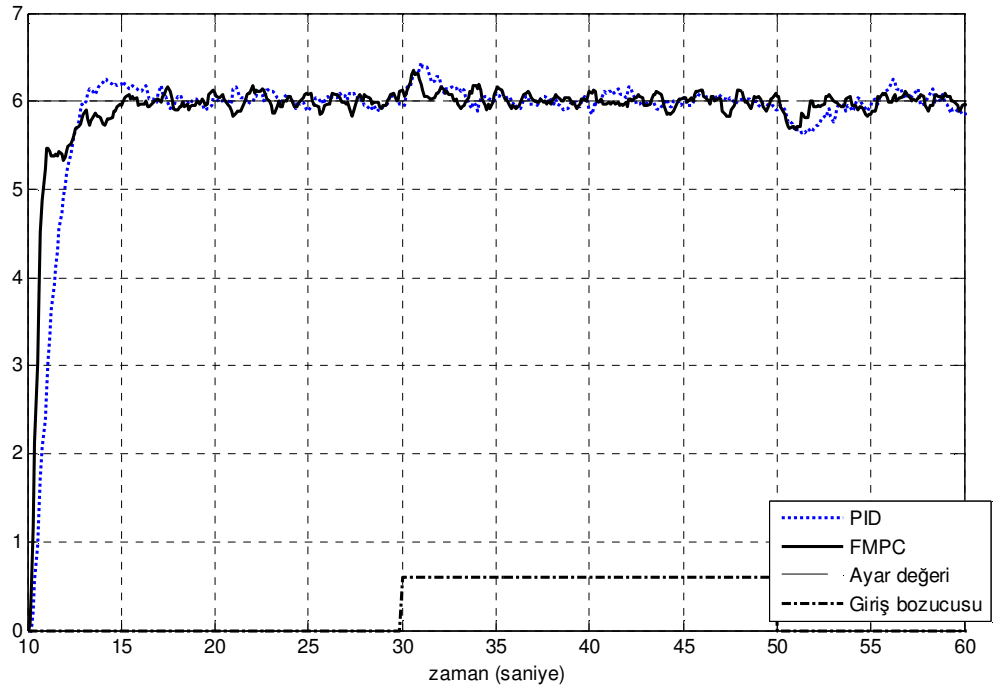

**Şekil 5.25:** Isıl sistem FMPC ve PID kontrolü

Şekil 5.25' de görüldüğü üzere, doğrusal olmayan ısıl sistemin FMPC kontrolü, PID kontrole göre daha hızlı ve aşımsız bir davranış sergilemektedir.

PID kontrolü yapılan ısıl sistemin giriş ve çıkış bozucularına verdiği tepki FMPC yapısı ile karşılaştırılmış ve Şekil 5.26 ve 5.27' deki gibi gözlenmiştir.




**Şekil 5.26:** Isıl sistem FMPC ve PID kontrolü (çıkış bozucusu etkisi)




**Şekil 5.27:** Isıl sistem FMPC ve PID kontrolü (giriş bozucusu etkisi)

Şekil 5.26 ve 5.27' de görüldüğü üzere, FMPC yapısı giriş ve çıkış bozucularına karşı daha hızlı tepkiler vermektedir ve PID kontrolörden daha az aşım yapmaktadır.

Ancak ısıl sistemin yüksek bir doğrusal olmayan yapısı bulunmadığından ve birinci mertebeden sistemlere yakın bir davranış sergilediğinden dolayı, PID kontrolöre göre çok belirgin bir performans farkı görülmemektedir.

# 6. SONUÇ VE ÖNERİLER

Bu çalışmada, son yıllarda endüstride sıkça kullanılmakta olan Model Öngörülü Kontrol yaklaşımı araştırılmıştır. Model Öngörülü Kontrol algoritmaları incelenmiş ve doğrusal olmayan sistemlerin kontrolü için önerilen Bulanık Model Öngörülü Kontrol yapısı irdelenmiştir.

Yöntemin uygulanması için öncelikle doğrusal olmayan sistem, bulanık olarak kaynaştırılabilen doğrusal alt sistemler tarafından modellenmiştir. Alt sistem modellerine uygun model öngörülü kontrolörler tasarlanmıştır.

Bulanık Model Öngörülü Kontrolör tasarımında karşılaşılan en önemli sorun, model öngörülü kontrol için gerekli olan öngörü ve kontrol ufku seçimleri, hatayı ve kontrol artımlarını cezalandıran ağırlık değerlerinin seçimleri ve hata tanımı için gerekli olan üstel fonksiyonun zaman sabitinin seçim kriterleridir.

Bulanık Model Öngörülü Kontrol algoritması MATLAB ortamında gerçeklenmiştir. Doğrusal olmayan bir sisteme Bulanık Model Öngörülü Kontrol yöntemi uygulanmış, değişken ayar değerlerinde sistem davranışı gözlenmiştir. Buna göre sistemin ayar değerini hızlı ve az aşımlı bir şekilde takip edebildiği görülmüştür.

Ayrıca sisteme giriş ve çıkış bozucuları eklenmiş, sistemin kendini hızla toparlayabildiği ve ayar değerine ulaşabildiği gözlenmiştir.

Önerilen kontrolör yapısını, klasik yöntemlerle karşılaştırmak üzere, alt sistemlere uygun PI kontrolörler tasarlanmış ve bu kontrolörler bulanık olarak kaynaştırılıp sistem kontrolü yapılmıştır. PI katsayıları hesaplanırken, hatanın karesi minimize edilmeye çalışılmış, böylece FMPC yönteminin kullandığı eniyileme yöntemi ile aynı yöntemde bir katsayı hesabı gerçekleştirilmiştir. Bu iki yöntem karşılaştırıldığında, önerilen Bulanık Model Öngörülü Kontrol yapısının daha başarılı bir performans sergilediği görülmüştür.

FMPC yapısı, PT326 sıcaklık deney setinden alınan verilere göre de gerçeklenmiştir. Burada, daha önce yapılan uygulamalardan farklı olarak bulanık kaynaşma işlemi sistemin çıkış değerine bakılarak değil, giriş değerine göre uygulanmıştır. Ancak

FMPC yaklaşımı, böyle bir kaynaşma ile de sistemi istenen dinamiğe getirebilmiş ve kararlı hale oldukça hızlı bir şekilde ulaştırabilmiştir.

Gelecekte bu çalışmanın, yüksek derecede doğrusal olmayan bir sürece, örneğin pH süreçlerine uygulanması, klasik kontrol yöntemlerine göre belirgin farklılıklar bulunması ve mevcut yapıya kısıtlar eklenerek kontrolör yapısının incelenmesi amaçlanmaktadır.

## KAYNAKLAR


[1] **Orupke, P. E.,** 2005. *Basics of Model Predictive Control*, ICM, EEE-CAP, Imperial College, London, United Kingdom, 14 April.

[2] **Maciejowski**, **J. M.,** 2002. *Predictive Control with Constraints,* Prentice Hall, Edinburgh Gate, England, pp. 1-80.

[3] **Nikolaou, M.,** *Model Predictive Controllers: "A Critical Synthesis of Theory and Industrial Needs"*, Houston, TX 77204-4792.

[4] **Camacho, E. F. and Bordons, C.,** 1999. *Model Predictive Control*, Springer-Verlag London, UK, pp. 37-42.

[5] **Chau, P. C.,** 2001. *Chemical Process Control: A First Course with MATLAB*, Universtiy of California, San Diego.

[6] **Chen, H. and Allgower, F.,** 1998. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability**,** *Automatica*, **34(10)**, 1205-1217

[7] **Nicolao, L. and Scattolini, R.**, Stability and robustness of nonlinear receiding horizon control*, G. De Progress in Systems and Control Theory*, **26**

[8] **Foss, B. A. and Johansen, T. A. and A. V. Sørensen,** 1995. Nonlinear predictive control using local models — applied to a batch fermentation process, *Control Engineering Practice,* **3(3)**, 389-396

[9] **Takagi, T. and Sugeno, M.**, 1985. Fuzzy identification of systems and its applicationto modeling and control*, IEEE Trans. Syst., Man, Cybern.*, **15**, 116–132

[10] **Nakamori, Y. and Suzuki, K. And Yamanaka, T.,** 1991. Model predictive controlusing fuzzy dynamic models*, in Proc. IFSA'91 Brussels*, **135**, 138

[11] **Huang, Y. L. and Lou, H. H. and Gong, J. P. and Edgar, T. F.,** 2000. Fuzzy model predictive control*, IEE Transactions on Fuzzy Systems*, **8(6)**

[12] **Passino, K. M. and Yurkovich, S.,** 1998. *Fuzzy Control,* Addison-Wesley Longman*,* California, 52.

[13] **Babuska, R. and Verbruggen, H. B.,** 1995. An Overview of Fuzzy Modeling for Control, *Control Engineering Practice*, **4**, 1593-1606

# EKLER

## EK A.1 : İki Alt Sistemin Bulanık Kaynaşması m-file

```
function [yn,w11n,w22n] = modellin( inptn,ykn)

%% Alt Sistem 1 Fark Denklemi%%%%%%%
y1n=0.202*inptn + 0.975*ykn;
%% Alt Sistem 2 Fark Denklemi%%%%%%%
y2n=2.13*inptn+ 0.867*ykn;
%%%%%%A1  Alt Sistem 1 Uyelik Fonksiyonu %%%%%%%
                if ykn < 0
                    w11n= 0;
                elseif ykn>=0 && ykn<3
                    w11n=1;
                elseif ykn>=3 && ykn<15
                    w11n=(-ykn+15)/12;
                else
                    w11n=0;
                end
%%%%%%%%%A2  Alt Sistem 2 Uyelik Fonksiyonu%%%%%%%%%%
                if ykn<3
                    w22n=0;
                elseif ykn>=3 && ykn<15
                    w22n=(ykn-3)/12;
                else
                    w22n=1;
                end
%% Bulanık Kaynastırma %%
yn = w11n*y1n + w22n*y2n;
```

## EK A.2: Model Öngörülü Kontrolör m-file

### Alt Sistem 1:

```
%BASICMPC Basic Predictive Control without constraints. (Script
file)
%
% Kısıtsız MPC yapısı
```

```
%
% asagidaki degiskenler istege gore secilebilir
% Tref: Ustel fonksiyon zaman sabiti
% Ts:   Ornekleme zamanı
% plant: Sistem tanımı  (ayrık zaman SISO LTI )
% model: Model tanımı  (ayrık zaman SISO LTI )
% P: Ongoru ufku
% M: Kontrol ufku
% setpoint: ayar yorungesi
% umpast, uppast, ympast, yppast: Baslangic degerleri plant & model.
%
% Assumes Matlab 5.3. Uses Control Toolbox LTI object class.
% No other toolboxes required.

%% J.M.Maciejowski, 8 March 1999. Revised 22.3.99, 15.12.99.
%% Copyright(C) 1999. All Rights Reserved.
%% Cambridge University Engineering Department.

%%%%%%TWO SUBSYSTEMS WITH TWO MPC CONTROLLERS%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%% PROBLEM TANIMI:
function [uu1x] = ezgibaris3 (setpointx,ypx,kontrolx)
global ympastx
global umpastx
global yppastx
global uppastx
global uux
global d
global yp1x
k1=.3;%% Agırlık katsayisi
Trefx = .5;
Tsx = 0.05;
% Ayrık Zaman SISO plant tanimi 'lti' object 'plant'
%%%%%  Yeni sistem tanimlamak icin burdan itibaren degisiklik
yapilmali
numpx=[0,0.202];
denpx=[1,-0.975];
plantx = tf(numpx,denpx,Tsx);
%%%%%  Yeni sistem tanimlamak icin buraya kadar degisiklik yapilmali
%%%%%
plantx = tf(plantx);  % transfer fonksiyonu bicimine donusturme
numpx = get(plantx,'num'); numpx = numpx{:}; % Pay polinomunu al
denpx = get(plantx,'den'); denpx = denpx{:}; % Payda polinomunu al
nnumpx = length(numpx)-1; % Sistem payı mertebesi
ndenpx = length(denpx)-1; % Sistem paydasi mertebesi

% Ayrık Zaman SISO model tanimi 'lti' object 'model'
% (default model=plant):
%%%%%  Yeni model tanimlamak icin burdan itibaren degisiklik
yapilmali
modelx = plantx;
%%%%%  Yeni model tanimlamak icin buraya kadar degisiklik yapilmali
modelx = tf(modelx);
nummx = get(modelx,'num'); nummx = nummx{:};
denmx = get(modelx,'den'); denmx = denmx{:};
nnummx = length(nummx)-1;
ndenmx = length(denmx)-1;
numpx=[zeros(1,ndenpx-nnumpx-1),numpx];
nummx=[zeros(1,ndenmx-nnummx-1),nummx];
Px = 15; %% Ongoru Ufku
Mx = 10; %% Kontrol Ufku
```

```
% Model birim basamak cevabi hesabi:
steprespx = step(modelx,[0:Tsx:max(Px)*Tsx]);
thetax = zeros(length(Px),Mx);
for jx=1:length(Px),
  thetax(jx,:) = [steprespx(Px(jx):-1:max(Px(jx)-
Mx+1,1))',zeros(1,Mx-Px(jx))];
end
Sx = steprespx(Px);
% Referans hata faktoru hesabi
% Referansin ayar degerine Ustel fonksiyon ile yaklasimi
   errfacx = exp(-Px*Tsx/Trefx);
%% Baslangic degeri verebilmek icin "kontroly" girisine bak
if kontrolx==0;
% Giris cikis sinyallerini tutacak degiskenleri 0' a esitle:
uux = 0; % Giris
ypx = 0; % Sistem cikisi
ymx = 0; %Model cikisi
d=0;% Sistem cikisi, model cikisi arasindaki fark
% Model Cikisi
% Baslangic kosullari
umpastx = zeros(ndenmx,1);
uppastx = zeros(ndenpx,1);
ympastx = zeros(ndenmx,1);
yppastx = zeros(ndenpx,1);
errornowx = setpointx-ypx;
  reftrajx = setpointx - errornowx*errfacx;
% Serbest cevap%
  yfpastx = ympastx;
  ufpastx = umpastx;
  for kkx=1:Px, % Ongoru ufku
    ymfreex(kkx) = nummx(2:nnummx+1)*ufpastx-
denmx(2:ndenmx+1)*yfpastx;
    yfpastx=[ymfreex(kkx);yfpastx(1:length(yfpastx)-1)];
    ufpastx=[ufpastx(1);ufpastx(1:length(ufpastx)-1)];
  end
  dutrajx = thetax\(reftrajx-ymfreex(Px)');
    uu1x = dutrajx(1) + umpastx(1);
    uppastx = [uu1x;uppastx(1:length(uppastx)-1)];
  yp1x = -denpx(2:ndenpx+1)*yppastx+numpx(2:nnumpx+1)*uppastx; %
Simulation
  %Gecmis cikis degerlerini guncelle
  yppastx = [yp1x;yppastx(1:length(yppastx)-1)];
  umpastx = [uu1x;umpastx(1:length(umpastx)-1)];
  ym1x = -denmx(2:ndenmx+1)*ympastx+nummx(2:nnummx+1)*umpastx;  %
Simulation
  ympastx = [ym1x;ympastx(1:length(ympastx)-1)];
  uux=uu1x;
else
  errornowx = setpointx-ypx;
  reftrajx = setpointx - errornowx*errfacx;
% Serbest cevap%
  yfpastx = ympastx;
  ufpastx = umpastx;
  for kkx=1:Px, % Ongoru ufku
    ymfreex(kkx) = nummx(2:nnummx+1)*ufpastx-
denmx(2:ndenmx+1)*yfpastx;
    yfpastx=[ymfreex(kkx);yfpastx(1:length(yfpastx)-1)];
    ufpastx=[ufpastx(1);ufpastx(1:length(ufpastx)-1)];
  end
 %Giris sinyalini hesapla
  d=ypx-yp1x;
```

```
    dutrajx = thetax\(reftrajx-d-ymfreex(Px)');
    uu1x = k1*dutrajx(1) + uux;
  uppastx = [uu1x;uppastx(1:length(uppastx)-1)];
  yp1x = -denpx(2:ndenpx+1)*yppastx+numpx(2:nnumpx+1)*uppastx; %
Simulation
  % Gecmis cikis degerlerini guncelle
  yppastx = [yp1x;yppastx(1:length(yppastx)-1)];
  umpastx = [uu1x;umpastx(1:length(umpastx)-1)];
  ym1x = -denmx(2:ndenmx+1)*ympastx+nummx(2:nnummx+1)*umpastx;  %
Simulation
  ympastx = [ym1x;ympastx(1:length(ympastx)-1)];
  uux=uu1x;
end
```

**Alt Sistem 2:**

```
%BASICMPC Basic Predictive Control without constraints. (Script
file)
%
% Kısıtsız MPC yapısı
%
% asagidaki degiskenler istege gore secilebilir
% Tref: Ustel fonksiyon zaman sabiti
% Ts:   Ornekleme zamanı
% plant: Sistem tanımı  (ayrık zaman SISO LTI )
% model: Model tanımı  (ayrık zaman SISO LTI )
% P: Ongoru ufku
% M: Kontrol ufku
% setpoint: ayar yorungesi
% umpast, uppast, ympast, yppast: Baslangic degerleri plant & model.
%
% Assumes Matlab 5.3. Uses Control Toolbox LTI object class.
% No other toolboxes required.

%% J.M.Maciejowski, 8 March 1999. Revised 22.3.99, 15.12.99.
%% Copyright(C) 1999. All Rights Reserved.
%% Cambridge University Engineering Department.

%%%%%%TWO SUBSYSTEMS WITH TWO MPC CONTROLLERS%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%% PROBLEM TANIMI:
function uu1y = ezgibaris4 (setpointy,ypy,kontroly)
global ympasty
global umpasty
global yppasty
global uppasty
global uuy
global d2
global yp1y
k2=.5; %% Agırlık katsayisi
Trefy = .5;
Tsy = 0.05;
% Ayrık Zaman SISO plant tanimi 'lti' object 'plant'

%%%%%  Yeni sistem tanimlamak icin burdan itibaren degisiklik
yapilmali
numpy=2.13;
denpy=[1,-0.867];
```

```
planty = tf(numpy,denpy,Tsy);
%%%%%  Yeni sistem tanimlamak icin buraya kadar degisiklik yapilmali
%%%%%

planty = tf(planty);  % transfer fonksiyonu bicimine donusturme
numpy = get(planty,'num'); numpy = numpy{:}; % Pay polinomunu al
denpy = get(planty,'den'); denpy = denpy{:}; % Payda polinomunu al
nnumpy = length(numpy)-1; % Sistem payı mertebesi
ndenpy = length(denpy)-1; % Sistem paydasi mertebesi

% Ayrık Zaman SISO model tanimi 'lti' object 'model'
% (default model=plant):
%%%%%  Yeni model tanimlamak icin burdan itibaren degisiklik
yapilmali
modely = planty;
%%%%%  Yeni model tanimlamak icin buraya kadar degisiklik yapilmali
modely = tf(modely);
nummy = get(modely,'num'); nummy = nummy{:};
denmy = get(modely,'den'); denmy = denmy{:};
nnummy = length(nummy)-1;
ndenmy = length(denmy)-1;

numpy=[zeros(1,ndenpy-nnumpy-1),numpy];
nummy=[zeros(1,ndenmy-nnummy-1),nummy];
Py = 15; %% Ongoru Ufku
My = 10; %% Kontrol Ufku
% Model birim basamak cevabi hesabi:
steprespy = step(modely,[0:Tsy:max(Py)*Tsy]);
thetay = zeros(length(Py),My);
for jy=1:length(Py),
  thetay(jy,:) = [steprespy(Py(jy):-1:max(Py(jy)-
My+1,1))',zeros(1,My-Py(jy))];
end
Sy = steprespy(Py);
% Referans hata faktoru hesabi
% Referansin ayar degerine Ustel fonksiyon ile yaklasimi
   errfacy = exp(-Py*Tsy/Trefy);
if kontroly==0; %% Baslangic degeri verebilmek icin "kontroly"
girisine bak
% Giris cikis sinyallerini tutacak degiskenleri 0' a esitle:
uuy = 0; % Giris
ypy = 0; % Sistem cikisi
ymy = 0; %Model cikisi
d=0; % Sistem cikisi, model cikisi arasindaki fark
% Model Cikisi
% Baslangic kosullari
umpasty = zeros(ndenmy,1);
uppasty = zeros(ndenpy,1);
ympasty = zeros(ndenmy,1);  % Model cevabi icin
yppasty = zeros(ndenpy,1);
errornowy = setpointy-ypy;
  reftrajy = setpointy - errornowy*errfacy;
% Serbest cevap%
  yfpasty = ympasty;
  ufpasty = umpasty;
  for kky=1:Py, % Ongoru ufku
    ymfreey(kky) = nummy(2:nnummy+1)*ufpasty-
denmy(2:ndenmy+1)*yfpasty;
    yfpasty=[ymfreey(kky);yfpasty(1:length(yfpasty)-1)];
    ufpasty=[ufpasty(1);ufpasty(1:length(ufpasty)-1)];
  end
```

```
  dutrajy = thetay\(reftrajy-ymfreey(Py)');
    uu1y = dutrajy(1) + umpasty(1);
    uppasty = [uu1y;uppasty(1:length(uppasty)-1)];
  yp1y = -denpy(2:ndenpy+1)*yppasty+numpy(2:nnumpy+1)*uppasty; %
Simulation
  %Gecmis cikis degerlerini guncelle
  yppasty = [yp1y;yppasty(1:length(yppasty)-1)];
  umpasty = [uu1y;umpasty(1:length(umpasty)-1)];
  ym1y = -denmy(2:ndenmy+1)*ympasty+nummy(2:nnummy+1)*umpasty;  %
Simulation
  ympasty = [ym1y;ympasty(1:length(ympasty)-1)];
  uuy=uu1y;
else
  errornowy = setpointy-ypy;
  reftrajy = setpointy - errornowy*errfacy;
% Serbest cevap%
  yfpasty = ympasty;
  ufpasty = umpasty;
  for kky=1:Py, % Ongoru ufku
    ymfreey(kky) = nummy(2:nnummy+1)*ufpasty-
denmy(2:ndenmy+1)*yfpasty;
    yfpasty=[ymfreey(kky);yfpasty(1:length(yfpasty)-1)];
    ufpasty=[ufpasty(1);ufpasty(1:length(ufpasty)-1)];
  end
  %Giris sinyalini hesapla
  d2=ypy-yp1y;
    dutrajy = thetay\(reftrajy-d2-ymfreey(Py)');
    uu1y = k2*dutrajy(1) + uuy;
  uppasty = [uu1y;uppasty(1:length(uppasty)-1)];
  yp1y = -denpy(2:ndenpy+1)*yppasty+numpy(2:nnumpy+1)*uppasty;
  % Gecmis cikis degerlerini guncelle
  yppasty = [yp1y;yppasty(1:length(yppasty)-1)];
  umpasty = [uu1y;umpasty(1:length(umpasty)-1)];
  ym1y = -denmy(2:ndenmy+1)*ympasty+nummy(2:nnummy+1)*umpasty;
  ympasty = [ym1y;ympasty(1:length(ympasty)-1)];
  uuy=uu1y;
end
```

## EK A.3: Isıl Sistem Bulanık Kaynaşması m-file

```
function [y,w1,w2] = modelli2( inpt,yk)
y1=0.05331*inpt + 0.92*yk;
y2=0.06118*inpt+ 0.9048*yk;
                %%%%%%A1  Uyelik Fonksiyonu 1%%%%%%%
                if inpt < 0
                    w1= 0;
                elseif inpt>=0 && inpt<4.5
                    w1=1;
                elseif inpt>=4.5 && inpt<7
                    w1=-0.4*inpt+ 2.8;
                else
                    w1=0;
                end
                %%%%%%%%%A2  Uyelik Fonksiyonu 2%%%%%%%%%%
                if inpt<4.5
                    w2=0;
                elseif inpt>=4.5 && inpt<7
                    w2=0.4*inpt-1.8;
                else
                    w2=1;
```

```
                end
y = w1*y1 + w2*y2;
```

## EK A.4: Isıl Sistem Kullanıcı Tanımlı Fonksiyonlar m-file

### Alt sistem 1:

```
function uu1x   = basicmbpc3(setpointx, ypx)

persistent ympastx
persistent umpastx
persistent yppastx
persistent uppastx
persistent uux
persistent d
persistent yp1x

if isempty (ympastx)
    ympastx=zeros(3,1);
end
if isempty (umpastx)
    umpastx=zeros(3,1);
end

if isempty (yppastx)
    yppastx=zeros(3,1);
end

if isempty (uppastx)
    uppastx=zeros(3,1);
end
if isempty (uux)
    uux=0;
end
if isempty (d)
    d=0;
end
if isempty (yp1x)
    yp1x=0;
end

uu1x=zeros(1);
ymfreex = zeros(15);
k1=0.7;
Trefx = 2.5;
Tsx = 0.1;

numpx=[0,0.05331,0,0];
denpx=[1,-0.92,0,0];

nnumpx = 3;
ndenpx = 3;

nummx=[0,0.05331,0,0];
denmx=[1,-0.92,0,0];
```

```
nnummx=3;
ndenmx=3;

Px=15;
Mx=10;

thetax=[0.459,
0.44097,0.42137,0.40007,0.37691,0.35174,0.32438,0.29464,0.26232,0.22
718];
errfacx = 0.3;

        errornowx = setpointx-ypx;
  reftrajx = setpointx - errornowx*errfacx;
  yfpastx = ympastx;
  ufpastx = umpastx;

  for kkx=1:Px,
      ymfreex(kkx) = nummx(2:nnummx+1)*ufpastx-
denmx(2:ndenmx+1)*yfpastx;
    yfpastx=[ymfreex(kkx);yfpastx(1:length(yfpastx)-1)];
    ufpastx=[ufpastx(1);ufpastx(1:length(ufpastx)-1)];
  end

  d=ypx-yp1x;
dutrajx = (reftrajx-d-ymfreex(Px))/thetax(1);
    uu1x = k1*dutrajx(1) + uux;
    uppastx = [uu1x;uppastx(1:length(uppastx)-1)];
  yp1x = -denpx(2:ndenpx+1)*yppastx+numpx(2:nnumpx+1)*uppastx;
  yppastx = [yp1x;yppastx(1:length(yppastx)-1)];
  umpastx = [uu1x;umpastx(1:length(umpastx)-1)];
  ym1x = -denmx(2:ndenmx+1)*ympastx+nummx(2:nnummx+1)*umpastx;
  ympastx = [ym1x;ympastx(1:length(ympastx)-1)];
  uux=uu1x;
```

## Alt Sistem 2:

```
function uu1y   = basicmbpc4(setpointy, ypy)

persistent ympasty
persistent umpasty
persistent yppasty
persistent uppasty
persistent uuy
persistent dy
persistent yp1y

if isempty (ympasty)
    ympasty=zeros(3,1);
end
if isempty (umpasty)
    umpasty=zeros(3,1);
end

if isempty (yppasty)
    yppasty=zeros(3,1);
end

if isempty (uppasty)
```

```
    uppasty=zeros(3,1);
end
if isempty (uuy)
    uuy=0;
end
if isempty (dy)
    dy=0;
end
if isempty (yp1y)
    yp1y=0;
end
uu1y=zeros(1);
ymfreey = zeros(15);
k2=.9;
Trefy = 2.5;
Tsy = 0.1;
numpy=[0,0.06118,0,0];
denpy=[1,-0.9048,0,0];
nnumpy = 3;
ndenpy = 3;
nummy=[0,0.06118,0,0];
denmy=[1,-0.9048,0,0];
nnummy=3;
ndenmy=3;
Py=15;
My=10;
thetay=[0.48426, 0.4676, 0.44918, 0.42883, 0.40633, 0.38146,
0.35398, 0.32361, 0.29004, 0.25294];
errfacy=0.3;
        errornowy = setpointy-ypy;
  reftrajy = setpointy - errornowy*errfacy;
  yfpasty = ympasty;
  ufpasty = umpasty;

  for kky=1:Py,
      ymfreey(kky) = nummy(2:nnummy+1)*ufpasty-
denmy(2:ndenmy+1)*yfpasty;
    yfpasty=[ymfreey(kky);yfpasty(1:length(yfpasty)-1)];
    ufpasty=[ufpasty(1);ufpasty(1:length(ufpasty)-1)];
  end

  dy=ypy-yp1y;
dutrajy = (reftrajy-dy-ymfreey(Py))/thetay(1);
    uu1y = k2*dutrajy(1) + uuy;
    uppasty = [uu1y;uppasty(1:length(uppasty)-1)];
  yp1y = -denpy(2:ndenpy+1)*yppasty+numpy(2:nnumpy+1)*uppasty;
  yppasty = [yp1y;yppasty(1:length(yppasty)-1)];
  umpasty = [uu1y;umpasty(1:length(umpasty)-1)];
  ym1y = -denmy(2:ndenmy+1)*ympasty+nummy(2:nnummy+1)*umpasty;
  ympasty = [ym1y;ympasty(1:length(ympasty)-1)];
  uuy=uu1y;
```

**ÖZGEÇMİŞ**



**Ad Soyad:** Ezgi Kömürcü

**Doğum Yeri ve Tarihi:** İzmir- 13.06.1984

**Adres:** 35540 Karşıyaka / İzmir

**Lisans Üniversite:** Dokuz Eylül Üniversitesi Elektrik Elektronik Mühendisliği