

**KOCAELİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**YÜKSEK MERTEBEDEN TEORİK BİR SİSTEMİN S7-300/400  
TİPİ PLC İÇİN SAYISAL BENZETİM İLE DİJİTAL PID  
KONTROLÖR TASARIMI**

**EMRE BİROK**

**KOCAELİ 2013**

KOCAELİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

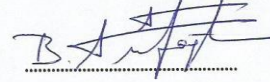
YÜKSEK MERTEBEDEN TEORİK BİR SİSTEMİN S7-300/400  
TİPİ PLC İÇİN SAYISAL BENZETİM İLE DİJİTAL PID  
KONTROLÖR TASARIMI

EMRE BİROK

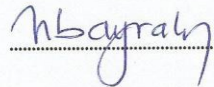
Yrd.Doç.Dr. Birol ARİFOĞLU,  
Danışman, Kocaeli Üniv.

Prof.Dr. Bekir ÇAKIR  
Jüri Üyesi, Kocaeli Üniv.

Doç.Dr. Mehmet BAYRAK  
Jüri Üyesi, Sakarya Üniv.







Tezin Savunulduğu Tarih: 24.07.2013

## **ÖNSÖZ ve TEŞEKKÜR**

Başta, emekleri, yüksek sabrı ve tezle ilgili veya ilgisiz her konudaki desteği için tez danışmanım değerli hocam Yrd. Doç. Dr. Birol ARİFOĞLU'na sonsuz teşekkürlerimi sunarım. Ayrıca bölümdeki tüm hocalarıma ve araştırma görevlilerine en içten dileklerle teşekkürlerimi ve saygılarımı sunarım.

Beni bugünlere getiren, hayatım boyunca hiçbir desteği ve katkıyı esirgemeyen annem Hülya BİROK ve babam Nadir BİROK'a, ayrıca tüm kalbi ve fedakarlığıyla her zaman yanımda olduğu için eşim Sinem AYVER BİROK'a sonsuz minnet duygularımı sunarım.

Mayıs - 2013

Emre BİROK

## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR .....	i
İÇİNDEKİLER.....	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ.....	vi
SİMGELER DİZİNİ VE KISALTMALAR.....	vii
ÖZET .....	viii
ABSTRACT .....	ix
GİRİŞ .....	1
1.GENEL BİLGİLER.....	4
2. PID KONTROL.....	8
2.1. PID Kontrol Tanımı .....	8
2.2. PID Kontrolör Tasarım Türleri .....	10
2.2.1. Oransal (P) kontrolör .....	10
2.2.2. Oransal - integral (PD) kontrolör .....	11
2.2.3. Oransal - türevsel (PI) kontrolör .....	13
2.2.4. Oransal - integral - türevsel (PID) kontrolör .....	14
2.3. Türev Vuruşu .....	15
2.4. İntegral Yığılması.....	15
2.5. PID Parametrelerini Elde Etme Yöntemleri .....	17
2.5.1. Ziegler - Nichols yöntemi .....	17
2.5.1.1. Ziegler - Nichols osilasyon yöntemi .....	18
2.5.1.2. Ziegler - Nichols birim basamak yanıtı yöntemi .....	19
2.5.2. Cohen - Coon yöntemi .....	20
2.5.3. Haalman yöntemi.....	21
2.6. Simulasyon Sonuçları.....	21
3.DİJİTAL PID KONTROL.....	24
3.1. Z-Dönüşümü .....	24
3.2. Fark Denklemleri .....	26
3.3. Endüstride Sistemlerde PID Kontrol Parametreleri ve Fark Denklemlerinin Belirlenmesi .....	27
3.3.1. Birinci mertebeden sistemler.....	27
3.3.1.1. Birinci mertebeden sistemlerde PID katsayılarının bulunması .....	29
3.3.1.2. Birinci mertebeden sistemlerin fark denkleminin bulunması .....	31
3.3.2. İkinci mertebeden sistemler .....	31
3.3.2.1. İkinci mertebeden az sönümlü sistemler .....	32
3.3.2.2. İkinci mertebeden kritik sönümlü sistemler .....	36
3.4. Yüksek mertebeden sistemlerin düşük mertebeden sistemlerle yaklaşık ifadesi .....	40
4. PLC İLE DİJİTAL PID KONTROLÖR PROGRAMLAMA .....	45
4.1. PLC' nin Genel Yapısı .....	46
4.2. SIMATIC S7-300/400 Tipi PLC'lere Genel Bakış.....	47

4.3. SIMATIC S7-300/400 İle PID Kontrol.....	48
5. ÖRNEK SİSTEM İÇİN PLC'DE PID KONTROLÖR TASARIMI.....	53
5.1. Üçüncü Mertebeden Sistemin İkinci Mertebeden Sistem Cinsinden İfadesi .....	53
5.2. Sistem Türünün Belirlenmesi .....	55
5.3. Sayısal Benzetim Parametreleri, Fark Denklemi ve PID Katsayılarının Elde Edilmesi.....	57
5.4. PLC Programının Oluşturulması.....	59
5.5. Programın Uygulanması ve Elde Edilen Sonuçlar .....	61
6. SONUÇLAR VE ÖNERİLER.....	69
KAYNAKLAR.....	71
EKLER.....	73
ÖZGEÇMİŞ.....	83

## ŞEKİLLER DİZİNİ

Şekil 1.1. Bir kontrol sisteminin örneksel birim basamak yanıtı.....	6
Şekil 2.1. P Kontrolörü ifade eden bir blok diyagramı .....	11
Şekil 2.2: P katsayısının birim basamak yanıtındaki etkisi.....	11
Şekil 2.3. PI Kontrolörü ifade eden bir blok diyagramı .....	12
Şekil 2.4. I katsayısının basamak yanıtındaki etkisi .....	12
Şekil 2.5. PD Kontrolörü ifade eden bir blok diyagramı .....	13
Şekil 2.6. D katsayısının basamak yanıtındaki etkisi.....	14
Şekil 2.7. PID Kontrolörü ifade eden bir blok diyagramı .....	14
Şekil 2.8. İşareti sınırlandırılmış bir PI-D kontrolörü yapısı.....	16
Şekil 2.9. İşareti sınırlandırılmış genel bir PID kontrolörü yapısı.....	16
Şekil 2.10. Örnek PID kontrolör şeması .....	17
Şekil 2.11. Ziegler-Nichols osilasyon yönteminin uygulanması.....	18
Şekil 2.12. Ziegler-Nichols birim basamak yönteminin uygulanması.....	19
Şekil 2.13. Ziegler-Nichols osilasyon yöntemi ile P,PI ve PID kontrolör birim basamak yanıtları.....	22
Şekil 2.14. Ziegler-Nichols birim basamak yanıtı yöntemi ile P,PI ve PID kontrolör .....	22
Şekil 2.15. Cohen-Coon yöntemi ile P,PI ve PID kontrolör birim basamak yanıtları .....	23
Şekil 2.16. Haalman yöntemi ile PI kontrolör birim basamak yanıtı .....	23
Şekil 3.1. Birinci mertebeden sistemin basamak girişe yanıtı.....	27
Şekil 3.2. Birinci mertebeden ölü zamanlı sistemin basamak girişe yanıtı.....	28
Şekil 3.3. İkinci mertebeden az sönümlü sistemin basamak girişe yanıtı .....	32
Şekil 3.4. İkinci mertebeden az sönümlü ölü zamanlı sistemin basamak girişe yanıtı .....	33
Şekil 3.5. İkinci mertebeden kritik sönümlü sistemin basamak girişe yanıtı.....	37
Şekil 3.6. İkinci mertebeden kritik zamanlı ölü zamanlı sistemin basamak girişe yanıtı .....	38
Şekil 4.1. PLC genel yapısı .....	46
Şekil 4.2. PLC'nin kontrol birimi olarak kullanıldığı bir kontrol sistemi.....	48
Şekil 4.3. FB41 PID fonksiyon bloğuna ilişkin işlev şeması .....	49
Şekil 4.4. FB41 PID fonksiyon bloğunun programdaki görünümü .....	50
Şekil 4.5. PID fonksiyon bloğu varsayılan değerleri .....	51
Şekil 5.1. İkinci mertebeden sistemin MATLAB devre şeması.....	55
Şekil 5.2. İkinci dereceden sistemin birim basamak yanıtı .....	55
Şekil 5.3. Birim basamak yanıtının düzenlenmiş hali.....	56
Şekil 5.4. Elde edilen sistemin ve üçüncü mertebeden sistemin karşılaştırılması....	56
Şekil 5.5. İkinci mertebeden elde edilen sistemin zaman sabitinin elde edilmesi ....	57
Şekil 5.6. PID kontrolör fonksiyon bloğu FB41 .....	59
Şekil 5.7. Sistemin benzetim fonksiyon bloğu FB1.....	60
Şekil 5.8. $T_1$ ve $T_D$ parametrelerinin doğrudan FB41 fonksiyon bloğunda yerine yazılması için gerekli olan program parçası.....	61
Şekil 5.9. PID benzetim fonksiyonunun online görüntüsü.....	61

Şekil 5.10. FB41 bloğunun online görüntüsü.....	62
Şekil 5.11. PID kontrol ve simulasyon programlarının online görüntüsü .....	63
Şekil 5.12. Sistemin birim basamak yanıtının PID programındaki görüntüsü .....	63
Şekil 5.13. Elde edilen birim basamak yanıtlarının karşılaştırılması a) Benzetim fonksiyonu b) MATLAB .....	64
Şekil 5.14. Sisteme PID kontrol uygulanmış haliyle PID kontrol programındaki görüntü .....	65
Şekil 5.15. Örnekleme zamanı 100ms olduğunda PID programındaki görüntüsü .....	65
Şekil 5.16. Örnekleme zamanına göre zaman tanım bölgesi değerlerine ait grafikler .....	67
Şekil 5.17. 240ms Örnekleme zamanı için elde edilen simulasyon görüntüsü .....	68
Şekil 5.18. Sisteme bozucu uygulanması durumu için simulasyon görüntüsü .....	68

## TABLULAR DİZİNİ

Tablo 2.1. Osilasyon yöntemi katsayı hesap tablosu .....	19
Tablo 2.2. Osilasyon yöntemi için hesaplanan değerler .....	19
Tablo 2.3. Ziegler-Nichols birim basamak yöntemi katsayı hesap tablosu .....	20
Tablo 2.4. Ziegler Nichols birim basamak yöntemi için hesaplanan değerler .....	20
Tablo 2.5. Cohen-Coon yöntemi katsayı hesap tablosu.....	21
Tablo 2.6. Cohen-Coon yöntemi için hesaplanan değerler .....	21
Tablo 2.7. Haalman yöntemi katsayı hesap tablosu .....	21
Tablo 2.8. Haalman yöntemi için hesaplanan değerler .....	21
Tablo 5.1. Farklı örnekleme zamanları ve setpoint değerlerine göre elde edilen zaman tanım bölgesi değerleri ve PID parametreleri .....	66



## SİMGELER VE KISALTMALAR

$c$	: Aşım
$D$	: Türev katsayısı
$e(t)$	: Hata işareti
$I$	: İntegral katsayısı
$K$	: Kazanç
$K_C$	: Toplam kontrolör kazancı
$K_D$	: Türevsel katsayı
$K_I$	: İntegral katsayısı
$K_P$	: Oransal katsayı
$L$	: Ölü zaman gecikmesi
$M_H$	: Yüksek mertebeden sistem transfer fonksiyonu
$M_L$	: Düşük mertebeden sistem transfer fonksiyonu
$P$	: Oransal Katsayı
$T$	: Örnekleme zamanı
$T_D$	: Türev zamanı sabiti
$t_d$	: Gecikme zamanı
$T_I$	: İntegral zaman sabiti
$t_r$	: Yükselme zamanı
$u(t)$	: Kontrol işareti
$\tau$	: Zaman sabiti
$\zeta$	: Sönüm oranı
$w_c$	: Köşe frekansı
$w_s$	: Örnekleme frekansı
$\omega_n$	: Doğal frekans

### Kısaltmalar

PID	: Proportional Integral Derivative (Oransal İntegral Türevsel)
PLC	: Programable Logic Controller (Programlanabilir Mantık Denetleyici)
PV	: Process variable (Süreç Değişkeni)
SP	: Setpoint (Ayarlanan Değer)

# YÜKSEK MERTEBEDEN TEORİK BİR SİSTEMİN S7-300/400 TİPİ PLC İÇİN SAYISAL BENZETİM İLE DİJİTAL PID KONTROLÖR TASARIMI

## ÖZET

Günümüz endüstriyel kontrol uygulamaları içerisinde denetleyici olarak PLC ve kontrol yöntemi olarak da dijital PID kontrol yaygın olarak kullanılmaktadır. Gelişen teknoloji ile birlikte kontrolün en kısa zamanda en az hatayla tasarlanması önem kazanmış, klasik kontrol yöntemlerinin hassasiyetinin düşük kalması, tasarlanma zorluğu gibi olumsuzluklar tasarımcıyı daha kolay ve güvenilir yeni yöntemler bulmaya yöneltmiştir. Tasarımcı bunu gerçekleştirirken tasarım aşamasında bütün deneme yanılmaları bilgisayar ortamında yapmayı tercih etmektedir. Günümüzde benzetim yöntemleri bu konuda kolaylık sağlamaktadır.

Dijital PID kontrolörler birinci ve ikinci mertebeden sistemler ile kolaylıkla tasarlanabilirken, daha yüksek mertebeden sistemlerin dijital olarak tasarlanma gücüğü bulunmaktadır. Yapılan tez çalışmasının amacı, yüksek mertebeden sistemlerin S7-300/400 tipi PLC'ler için sayısal benzetim ile dijital PID kontrolör tasarımını yapmaktır. Bu amaçla öncelikle Ziegler Nichols, Cohen gibi klasik PID kontrolör yöntemleri üçüncü mertebeden örnek bir sistem kullanılarak MATLAB'ta incelenmiştir. Verilen biçimsel yaklaşım yöntemi ile üçüncü mertebeden örnek sistem, dijital olarak tasarlanabilen ikinci mertebeden sistem cinsinden yaklaşık ifade edilmiştir. Yeni sistemin fark denklemi SIMATIC MANAGER'da programlanarak sayısal benzetim fonksiyonu oluşturulmuştur. MATLAB ve SIMATIC MANAGER yazılımlarında yapılan simülasyonlarda elde edilen sonuçlar birbirleriyle karşılaştırılmıştır. Çalışmanın sonucunda benzetim fonksiyonu sistemin kendisi kabul edilerek SIMATIC MANAGER'da S7-300/400 tipi PLC'lerde kullanılabilir, sistem karakteristiğine ait parametrelerin girilmesiyle PID katsayılarını kendisi hesaplayabilen bir dijital PID kontrolör programlanmış ve farklı örnekleme zamanlarına göre simülasyonlar yapılmıştır.

**Anahtar Kelimeler:** Biçimsel Yaklaşım Yöntemi, PID, PLC, Sayısal Benzetim, Yüksek Mertebeden Sistem.

## **PID CONTROLLER DESIGN FOR S7-300/400 TYPE PLC WITH DIGITAL SIMULATIONS OF A THEORICAL HIGH ORDER SYSTEM**

### **ABSTRACT**

Among the industrial control implementations, PLC as monitor and PID as a means of control is widely used today. With the developing technology, the completion of control in shortest time and with least mistake has gained importance; and that the sensitivity of the classic control means has become rather lower and difficulties in projecting have forced the designer to find easier and more reliable methods. Designer prefers to do all the trial and error on electronic environment while performing this process. Today, simulation methods facilitate about this subject.

While digital PID controllers can easily be designed with first and second order systems, higher order systems have difficulties for being digitally designed. The purpose of this thesis is to make PID controller design for S7-300/400 type PLCs with digital simulations. To do this, such classic PID controllers as Ziegler, Nichols, Cohen were analyzed by using a third order sample system in MATLAB. With given formal approaches method, third order sample system is expressed roughly in the sort of second order system which can digitally be designed. The difference equation of the new system has been formed through digital simulation function by programming in SIMATIC MANAGER. The results obtained from simulations made in MATLAB and SIMATIC MANAGER have been compared. At the end of the study, by accepting digital simulation system itself, a digital PID controller which can calculate PID ratios after entering parameters belonging to system characteristics and which can be used S7-300/400 type PLCs in SIMATIC MANAGER is programmed and simulations has been made according to different sample surveys.

**Keywords:** Formal Approach Method, PID, PLC, Digital Simulation, High Order System.

## GİRİŞ

Son yıllarda otomatik kontrol sistemleri insanlığın ve uygarlıkların gelişmesi için önemli roller oynayan bir bilim dalı haline gelmiş ve büyük bir hızla gelişmektedir. Bu sistemler günlük hayattaki basit otomatik makine hatlarından silah sistemlerine, insanın yerini alan robot uygulamalarından uzay teknolojilerine kadar birçok alanda uygulanmaktadır. İyi bir otomatik kontrol sisteminde giriş büyüklüğünün değişmesiyle çıkış büyüklüğü de en kısa sürede istenilen değere ulaşması, bu süre içinde meydana gelen bozucu etkilerin ise kendi kendine giderilmesi hedeflenir. Örneğin 100 metre koşan bir atlet bu mesafeyi mümkün olabilecek en kısa sürede koşmayı amaçlar. Öte yandan bir maraton koşucusu mesafeyi en kısa sürede koşmanın yanı sıra, enerji tüketimini kontrol etmesi, kendisi için en uygun yarış stratejisini izlemesi gerekir. Böyle bir hedefe ulaşabilmede ise devreye denetleyici girer.

Endüstriyel otomasyonda kullanılan en yaygın denetleyici, Programlanabilir Mantık Denetleyici (Programable Logic Controller – PLC ) ‘dir. İlk olarak 60’lı yılların sonlarında ortaya çıkan PLC, genel olarak endüstri alanında kullanılmak üzere tasarlanmış, dijital prensiplere göre yazılan fonksiyonu gerçekleyen, bir sistemi ya da sistem gruplarını, giriş çıkış kartları ile denetleyen, içinde barındırdığı zamanlama, sayma, saklama ve aritmetik işlem fonksiyonları ile genel kontrol sağlayan elektronik bir cihazdır. Bakım maliyetlerinin elektromekanik röleli kontrol sistemlerine göre oldukça ucuz olması, yapısının basitliği, sayısız özel uygulamaya ve sistemlerin uzantılarına cevap verecek biçimde çalışması, yaptığı işe göre kapladığı alanın düşüklüğü gibi avantajlara sahip olması, PLC’nin endüstride vazgeçilmez olmasını sağlamış ve ileriye dönük yeni güncelleştirmelerin gerekliliğini ortaya çıkarmıştır.

Oransal-İntegral-Türevsel (Proportional Integral Derivative - PID) kontrol ise, endüstriyel süreçlere ilişkin kontrol sorunlarının çözümünde önemli bir yere ve öneme sahip bir kontrol yöntemidir. Endüstriyel uygulamalara ilişkin kontrol yapıları genellikle PID kontrol çevrimlerinden oluşur. PID kontrol, hata işaretinin oransa

değeri (P), zamana göre integrali (I) ve zamana göre türevine (D) bağı olarak ifade edilir. Kontrol edilen işaret alınır ve referans işaretiyle karşılaştırılır. Giriş ve çıkış işaretinin arasındaki fark hatası ile orantılı işaret, PID denetleyici tarafından değerlendirilerek hatayı gidermek üzere sistem için en uygun çıkış işreti üretilip sisteme uygulanır. Sistemin istenilen şekilde çalışabilmesi için PID parametrelerinin ayarlanması gerekmektedir. Bunun için manuel ayarlama ve parametrelerin hesaplanması (Ziegler-Nichols) gibi çeşitli yöntemlerin dışında otomatik ayarlama yöntemleri de söz konusudur.

Analog ve Dijital PID kontrol olmak üzere iki çeşit PID kontrol yönteminden söz edilebilir. Analog PID kontrol, sayısal gürültüyü gidermesi, dijital-analog ya da analog-dijital dönüştürücü kullanım gerekliliği olmaması, geniş bant aralığı ve yüksek hız sağlaması gibi avantajlara sahip olmasına rağmen geniş bir kullanım alanı bulamamıştır. Bunun nedeni, dijital kontrolörlerin elektronik olarak yeniden kolayca konfigüre edilebilmesi ve güvenilir olmasıdır. Bu sebeplerden dolayı analog kontrol yerine dijital kontrol yaygın olarak kullanılmaktadır.

Dijital kontrol sistem tasarımı, analog kontrol sistem tasarımıyla prensipte benzerdir. Kontrol edilen sistem genelde aynıdır sadece dijital kontrolör örneklenmiş veriye yani dijital bilgiye göre tasarlanır. Bu durumda da analog kontrol sistemlerinde olduğu gibi zaman ve frekans tanım bölgesinde inceleme yapılabilir. Ancak analog kontrol sistemleri için s-tanım bölgesindeki tasarım kriterleri, dijital kontrolde z-tanım bölgesine uyarlanır.

Tasarım kriterleri genellikle sistemin ne yapması gerektiğini belirtmek ve nasıl yaptığını değerlendirmek için kullanılır. Bu kriterler her bir uygulamaya özgü farklıdır ve genellikle görece kararlılık, kararlı hal hatası, geçici yanıt ve frekans yanıtı özellikleri ile ilgili kısımlardan oluşur [5].

Doğrusal kontrol sistemlerinin tasarımı zaman ya da frekans tanım bölgesinde gerçekleştirilebilir. Örneğin kararlı hal hatası genellikle birim basamak, rampa ya da parabolik giriş için tanımlanır. Belirli tasarım kriterleri zaman tanım bölgesinde çok daha kolay değerlendirilebilir. En büyük aşım, yükselme zamanı ve yerleşme zamanı gibi birim basamak giriş için tanımlanan kriterler genellikle zaman bölgesi

tasarımında kullanılır. Kazanç payı, faz payı ve rezonans tepesi (Mr) gibi büyüklükler ise frekans tanım bölgesinde kullanılır [5].

Tasarımın zaman ya da frekans tanım bölgesinde yapılması kararı genellikle tasarımcının seçimine bağlı kalmış bir konudur. Doğrusal kontrol sistemlerinin tasarımı tarihsel olarak Bode diyagramı, Nyquist yer eğrisi, genlik-faz eğrisi ve Nichols abağı gibi frekans tanım bölgesinde oluşturulan bir dizi güçlü grafiksel yöntemlerle başlamıştı. Bu yöntemlerin yararı yaklaşık çizimlerde ayrıntıya gerek duyulmamasıdır. Böylece tasarımcı kazanç payı, faz payı, Mr vs. gibi frekans tanım bölgesi kriterlerinden yararlanarak yüksek mertebeden sistemleri bile tasarlayabilir. Böylece bazı kontrolörler için frekans tanımı bölgesinde gerekli deneme sına adımları en az sayıya indirilebilir. Zaman tanım bölgesinde ise yükselme zamanı, gecikme zamanı, yerleşme zamanı, aşım, vs. gibi davranış kriterleri sadece birinci ve ikinci mertebeden sistemlerde analitik tasarlanabilir ya da bu sistemler cinsinden yaklaşık ifade edilebilir. İkinci mertebeden daha yüksek sistemlerde zaman tanımı bölgesi için geçerli bir tasarım yöntemi vermek zordur. Deneyimsiz tasarımcı kazanç ve faz payı, rezonans tepesi gibi frekans tanım bölgesi büyüklüklerinin gerçek sistem davranışı üzerindeki etkisini anlamakta zorlanır. Ayrıca anlamlı frekans tanım bölgesi koşullarının deneme sına ile belirlenmesi, kendisi de bir deneme sına gerektiren tasarım işleminin bir ön evresini oluşturur. Frekans tanım bölgesi tasarım yöntemlerinin tarihsel üstünlüğü bilgisayar yazılım paketlerinin geliştirilmesi ile değerini yitirmiştir [5].

Yukarıda bahsedilen hususlar göz önüne alındığında tez çalışmasında ikinci mertebeden sistemin dijital kontrol tasarımı yapılacağından tasarım yöntemi olarak zaman tanım bölgesi seçilecektir. Dolayısıyla sonraki kısımda zaman tanım bölgesinden bahsedilecektir.

## 1.GENEL BİLGİLER

Kontrol sistemlerinde zaman genellikle bağımsız bir değişken görevini üstlenir ve bu nedenle durum ve çıkış büyüklüklerinin zamana göre değişimi ya da zaman yanıtları genel ilgi alanını oluşturur. Analiz problemlerinde sisteme referans giriş işaretleri uygulanır ve bu işaretlere verilen yanıtlar incelenerek sistemlerin davranışı değerlendirilmeye çalışılır. Bir kontrol sisteminde eğer çıkış işaretinin giriş işaretini belirli koşullar altında takip etmesi isteniyor ise, giriş ve çıkış işaretleri zaman fonksiyonu olarak karşılaştırılır. Bu nedenle kontrol sistemi davranışlarının son değerlendirmesi genellikle hep zaman yanıtı üzerinde yapılır [5].

Bir kontrol sisteminin zaman yanıtı genellikle iki kısımdan oluşur: geçici hal yanıtı ve sürekli hal yanıtı. Eğer  $y(t)$  bir sürekli sistem yanıtını ifade ediyorsa  $y_t(t)$  geçici yanıt ve  $y_{ss}(t)$  sürekli yanıtı ifade etmek üzere genelde,

$$y(t) = y_t(t) + y_{ss}(t) \quad (1.1)$$

yazılabilir.

Kontrol sistemlerinde geçici hal yanıtı, sistem yanıtının, zaman ilerledikçe sıfıra doğru giden kısmı olarak tanımlanır. Buna göre  $y_t(t)$ ,

$$\lim_{t \rightarrow \infty} y_t(t) = 0 \quad (1.2)$$

Sürekli hal yanıtı ise geçici hal yanıtı söndükten sonra zaman yanıtının geriye kalan kısmıdır. Bir sistemin sürekli hal yanıtı da çok önemlidir, çünkü zaman ilerledikçe sistem yanıtının nerede sona erdiğini belirtir. Genelde çıkışın sürekli hal yanıtı istenen referans değeriyle uyuşmadığında sistemde kararlı hal hatası bulunduğu anlamına gelir [5].

Doğrusal kontrol sistemlerinde geçici hal yanıtının değerlendirilmesi genellikle birim basamak yanıtından ( $u_s(t)$ ) yararlanılarak yapılır. Kontrol sistemlerinin birim basamak girişe cevabı birim basamak yanıtı olarak adlandırılır. Birim basamak yanıtı ile ilişkili olarak kontrol sistemlerinin zaman tanım bölgesi özellikleri aşağıdaki davranış kriterleriyle değerlendirilir [5].

- En büyük aşım:  $y(t)$  birim basamak yanıtı olmak üzere  $y(t)$ 'nin en büyük değeri  $y_{mak}$  ve sürekli hal değeri  $y_{ss}$  ile belirlenmiş olsun ( $y_{mak} \geq y_{ss}$ ).  $y(t)$ 'nin en büyük aşımı,

$$\text{en büyük aşım} = y_{mak} - y_{ss} \quad (1.3)$$

olarak tanımlanır. En büyük aşım genellikle basamak yanıtı son değerinin yüzdesi ile ifade edilir,

$$\% \text{ en büyük aşım} = \frac{\text{en büyük aşım}}{y_{ss}} \times 100 \quad (1.4)$$

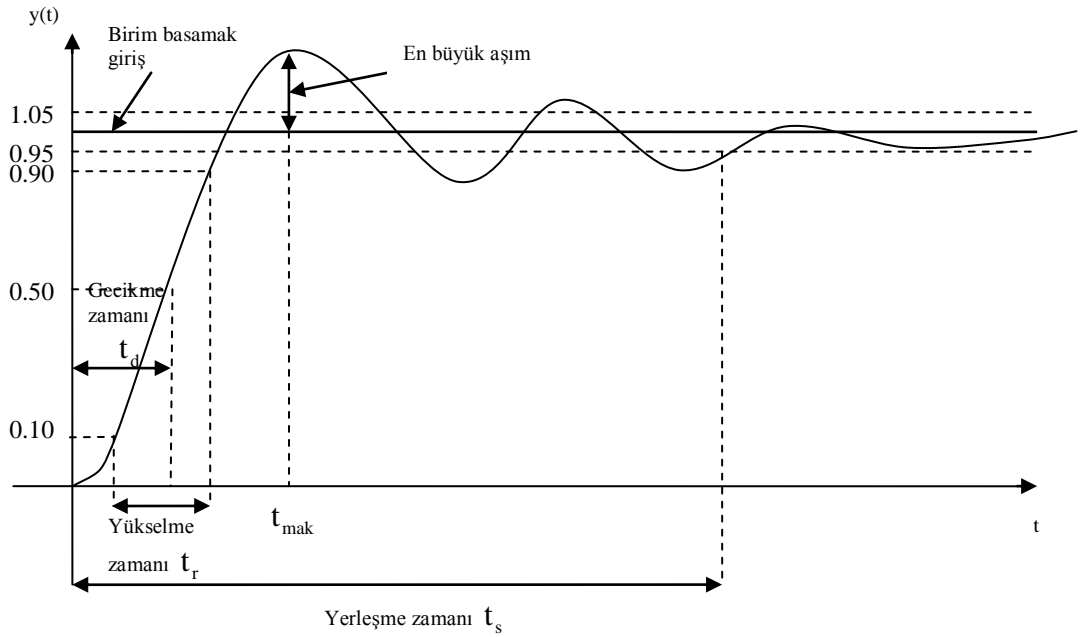
En büyük aşım genellikle bir kontrol sisteminin görelî kararlılığını değerlendirme ölçüsü olarak kullanılır. Genellikle sistemde aşımın büyük olması istenmez. Tasarımda en büyük aşım bir zaman tanım bölgesi ölçüsü olarak verilir. Şekil 1.1'de birim basamak yanıtında en büyük aşımın birinci aşımında gerçekleştiği görülür. Bazı sistemlerde en büyük aşım sonraki tepe değerlerinde oluşabilir [5].

- Gecikme zamanı: Gecikme zamanı  $t_d$  basamak yanıtının son değerinin %50 değerine erişim zamanı olarak tanımlanır. Bu zaman Şekil 1.1'de gösterilmiştir.

- Yükselme zamanı  $t_r$ , Şekil 1.1'de görüldüğü gibi, basamak yanıtının son değerinin %10 değerinden %90 değerine ulaşma zamanı olarak tanımlanır. Yükselme zamanının ayrıca, son değer %50 değerinde basamak yanıtı teğetinin tersi olarak ifade edildiği, başka bir tanımı da vardır.

- Yerleşme zamanı: Şekil 1.1'de görüldüğü gibi, basamak yanıtı son değerinin belirli bir yüzdesine kadar azalması ve bu değer altında kalması için geçmesi gereken zaman olarak tanımlanır. %5 çok sık kullanılan bir değerdir.





Şekil 1.1. Bir kontrol sisteminin örneksel birim basamak yanıtı

Yukarıda verilen dört büyüklük, birim basamak yanıtına bağlı olarak kontrol sisteminin doğrudan geçici hal davranışına ilişkin ölçüleri tanımlar. Basamak yanıtı Şekil 1.1'deki gibi tanımlandığında bu zaman tanım bölgesi kriterleri görece kolay ölçülür. Bu değerlerin üçüncü mertebenin altındaki basit sistemler dışında, analitik elde edilmeleri çok zordur [5].

Yapılan tez çalışmasının sonraki bölümü olan ikinci bölümde PID kontrol yöntemi ile ilgili genel bilgiler verilecek ve üçüncü mertebeden örnek bir sistem kullanılarak klasik PID kontrolör tasarım türleri incelenecektir.

Üçüncü bölümde birinci ve ikinci mertebeden endüstriyel sistemler ele alınarak bu sistemlerin fark denklemleri ve PID parametrelerinin matematiksel olarak elde edilmeleri anlatılacaktır. Ayrıca bu bölümde yüksek mertebeden sistemlerin düşük mertebeden sistemler cinsinden yaklaşık ifade edilmesine ilişkin bir yöntem verilecektir.

Dördüncü bölümde PLC'nin genel yapısı anlatılacak, PLC'de PID kontrolör tasarımına dair bilgiler verilecektir.

Beşinci bölümde ise öncelikle ikinci bölümde örnek olarak alınan üçüncü mertebeden sistem, üçüncü bölümde verilen yöntemle ikinci mertebeden sistem cinsinden yaklaşık ifade edilecektir. İfade edilen sistemin birim basamak yanıtı MATLAB programında elde edilerek sistem türü belirlenecektir. Belirlenen sistem türüne göre üçüncü bölümde verilen denklemler yardımıyla dijital PID kontrolör yazılımı için gerekli olan fark denklemi elde edilecektir. Fark denklemi sayısal benzetim yöntemiyle Simatic Manager programında programlanacak ve sonuçta S7-300/400 tipi PLC'lerde çalışabilecek, sistem karakteristiğine ait parametrelerin girilmesiyle PID katsayılarını kendisi hesaplayan bir dijital PID kontrolör tasarlanmış olacaktır. Bölümün sonunda PID kontrolör tasarım aşamasında örnekleme zamanının seçiminin etkisi araştırılacak ve uygun bir örnekleme zamanı kullanılarak oluşturulan bir kontrolöre bozucu uygulanması durumu incelenecektir.

Tezin amacı; yüksek mertebeden (üçüncü mertebeden) bir sistemle karşılaştırılması durumunda bu sistemin biçimsel yaklaşım yöntemi kullanılarak düşük mertebeden (ikinci mertebeden) sistem cinsinden dijital olarak programlanabilmesine yönelik teorik çalışma yapmak, bu çalışmayı yaparken S7-300/400 tipi PLC'ler için, SIMATIC MANAGER programında, sahada çalışmak yerine doğrudan bilgisayar ortamında tasarım aşamasında her türlü değişikliğe ve analize imkan verebilen sayısal benzetim yöntemi uygulanan, ve sistem karakteristiğine ait parametrelerin girilmesiyle PID katsayılarını kendisi hesaplayabilen ikinci mertebeden sistemlere ait bir dijital PID kontrolör tasarlamaktır.

Yapılan tezin literatürdeki diğer çalışmalardan farkı, yüksek mertebeden sistemin dijital kontrolör tasarımının incelenmesi, MATLAB karşılaştırılması verilerek sayısal benzetimin yapılabilirliğinin kanıtlanması ve SCL (Structured control language) kodu kullanılarak aynı PLC program bloğunda hem sayısal benzetim yapılması hem de PID katsayılarının hesaplanmasıdır.

## 2. PID KONTROL

### 2.1. PID Kontrol Tanımı

PID, endüstriyel kontrol sistemlerinde özel bir yere ve öneme sahip, geri beslemeli bir kontrol yöntemidir. Bir PID kontrolör, ölçülen proses değişkeni ile arzu edilen referans değeri arasındaki hatayı hesaplayıp, işlemi, hızlıca ayarlanabilen bir düzeltme işareti ile sisteme vererek, hatayı en az seviyede tutmak amacıyla düzeltmeye çalışır.

PID kontrol kuralının basit olması ve en çok üç parametre ayarı gerektirmesi, ilgili kontrol işaretinin uygulanabilir değerler arasında tutulmasının kontrol kuralını fazla etkilememesi, sistem parametrelerinin değişimine karşı duyarlılığının düşük olması gibi nedenler, PID kontrolün önemli sayılabilecek özellikleridir.

PID kontrol, üç ayrı parametre içerir: oransal (P,proportional), integral (I,Integral) ve türevsel (D,Derivative). Oransal parametre geçerli hataya karşı tepkiyi, integral parametresi son hataların toplamına bağlı tepkiyi, türevsel parametre ise hatanın değişmekte olduğu orana bağlı tepkiyi ifade eder [2,10]. PID kontrolörü ile hata, hatanın zamana bağlı integrali, ve hatanın değişimine bağlı bir kontrol işareti üretilir. Bu tür bir kontrolöre ilişkin matematiksel ifade genel olarak,

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2.1)$$

biçiminde veya endüstriyel uygulamalarda yaygın olarak,

$$u(t) = K_c \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_D \frac{de(t)}{dt} \right) \quad (2.2)$$

biçiminde verilir. Burada;

$K_P$ : Oransal katsayı

$K_I$ : İntegral katsayısı

$K_D$ : Türevsel katsayı

$K_C$ : toplam kontrolör kazancı

$T_I$ : integral zaman sabiti

$T_D$ : türev zamanı sabiti

$e(t)$ : hata işareti

$u(t)$ : kontrol işareti

olarak tanımlanır.

PID kontrol kuralına göre;

$u(t)$ = kontrol işareti,

$P(t) = K_C e(t)$  oransal terim,

$I(t) = \frac{K_C}{T_I} \int e(t) dt$  integral terimi,

$D(t) = K_C T_D \frac{de(t)}{dt}$  türev terimi

olmak üzere,

$$u(t) = P(t) + I(t) + D(t) \quad (2.3)$$

biçiminde yazılabilir [3,4].

Bu ifadelerle göre oransal terim  $e(t)$  hata işareti değerinin bir oran katsayısı ( $K_C$ ,  $K_P$ ) ile çarpımı kadar, hata sıfırdan büyükse kontrol işaretini arttırıcı yönde, hata sıfırdan küçükse eksiltici yönde etkiler.  $I(t)$  integral teriminin kontrol işaretini arttırıcı veya

eksiltici yönde etkilemesi hatanın zamana göre toplamına bağlıdır. D(t) türevsel terim ise hatanın değişim oranına bağlı olarak kontrol işaretini etkiler [3,4].

## 2.2. PID Kontrolör Tasarım Türleri

Sabit K kazançlı bir kontrol işleminde, kontrol işareti kontrolör çıkışına sabit bir oranla aktarıldığından, bu kontrol oransal kontrol olarak adlandırılır. Sezgisel olarak, oransal işleve ek olarak, giriş işaretinin türevinden ya da integralinden de yararlanılabileceği düşünülebilir. Buna göre, içinde toplayıcı (toplama veya çıkarma), kuvvetlendirici, zayıflatıcı, türev ve integral alıcı elemanlar bulunan, daha genel bir kontrolör göz önünde bulundurulabilir. Tasarımcının görevi bu elemanlardan hangilerinin, hangi oranda ve şekilde bağlanarak kullanılması gerektiğini belirlemektir [5].

PID kontrolörler standart olarak dört türde bulunabilirler. Kontrol edilecek sistemin yapısına göre (sistem parametrelerine göre) bu dört türden biri tercih edilir. Bu türler sırası ile şunlardır:

- P Kontrolör
- PI Kontrolör
- PD Kontrolör
- PID Kontrolör

Konunun devamında yukarıda bahsedilen kontrolör türleri incelenecektir.

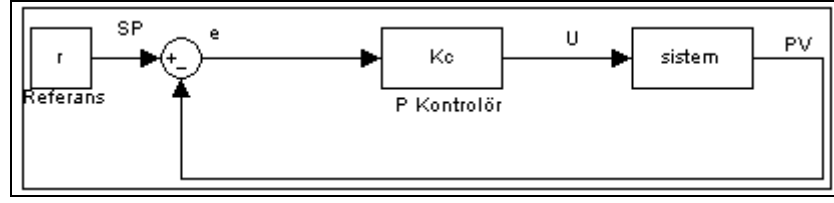
### 2.2.1. Oransal (P) kontrolör

Sadece oransal katsayının aktif olduğu PID kontrolör türüdür. Kontrol kuralı,

$$U = K_c (SP - PV) \quad (2.4)$$

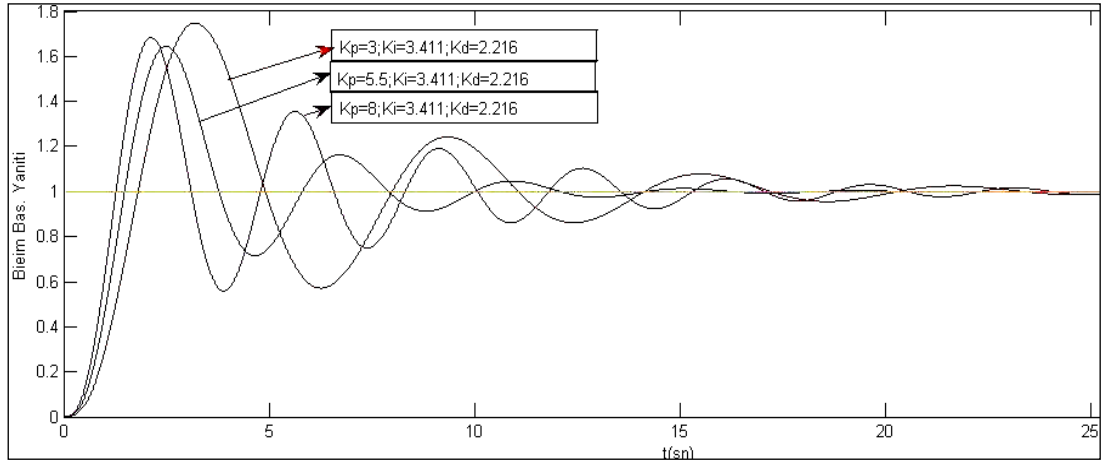
biçiminde ifade edilen bir kontrol yöntemi olarak tanımlanır. Burada, SP (setpoint) kontrol edilen büyüklüğe ilişkin istenen değer, PV (process variable) kontrol edilen

büyükliğün gerçek değeri ve  $K_c$  ( $K_p$ ) oransal katsayıdır [6]. P kontrolörü ifade eden bir blok diyagramı Şekil 2.1.'de görülmektedir.



Şekil 2.1. P Kontrolörü ifade eden bir blok diyagramı

Oransal kazancın yüksek olması, kontrolör çıkışında büyük değişikliğe yol açar. Oransal kazanç çok yüksek ise sistem kararsızlaşabilir. Oransal kazancın düşük olması ise yüksek giriş hatasına yanıtın zayıf ve kontrolör hassasiyetinin düşük olmasına neden olur [2]. Şekil 2.2'de sabit türev ve integral katsayıları için oransal kazancın birim basamak yanıtına etkisi görülmektedir.



Şekil 2.2. P katsayısının birim basamak yanıtındaki etkisi

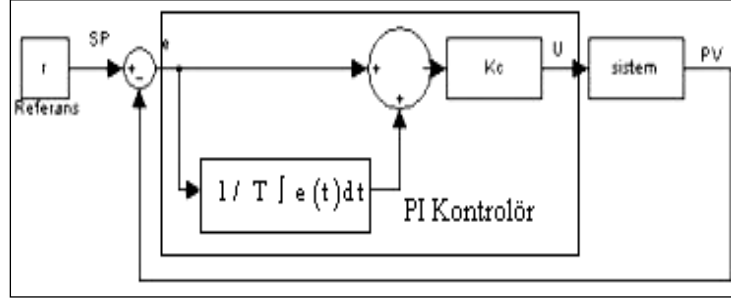
Oransal kontrolde çıkış hiçbir zaman istenilen seviyeye ulaşmaz, kararlı hal hatası oluşur. Kontrol elemanının %100 çıkış vermesi için gerekli olan hatadaki % değişim oransal bant olarak tanımlanır. Oransal bantı küçültmek sistem kazancını arttırmak anlamına gelir [7].

### 2.2.2. Oransal-İntegral (PI) kontrolör

Oransal ve integral katsayılarının aktif olduğu PID kontrolör türüdür. PI kontrol kuralı;  $u$  kontrol işareti,  $e$  hata işareti,  $K_c$  oransal katsayı ve  $T_I$  integral zaman sabiti olmak üzere,

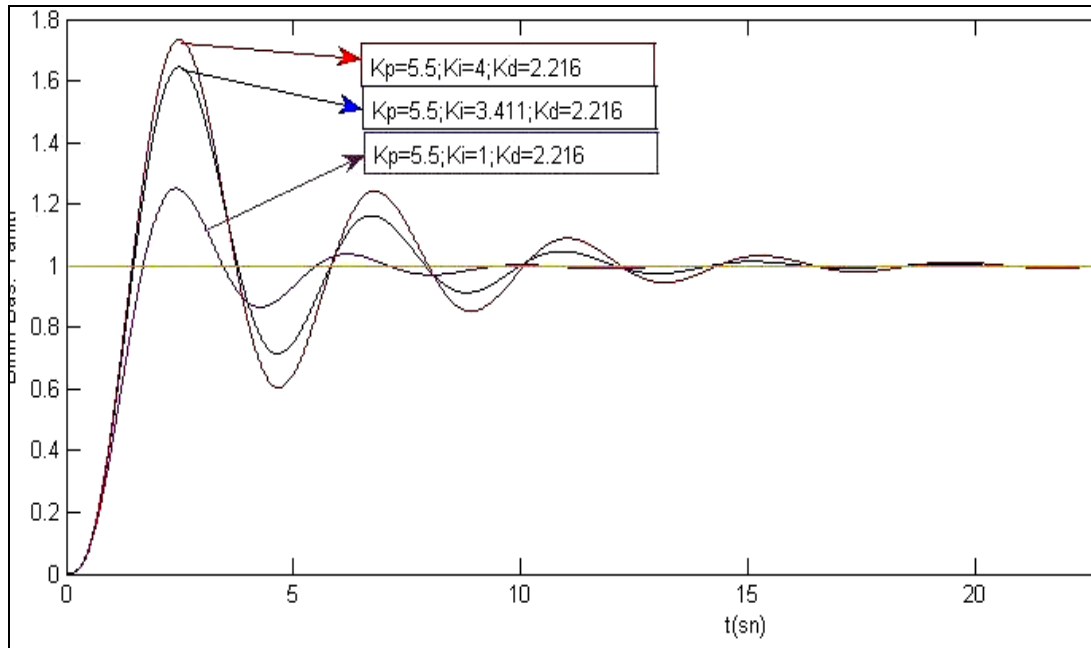
$$U(t) = K_c[e(t) + 1/T_i \int e(t)dt] \quad (2.5)$$

Bu biçimde ifade edilir [6]. PI kontrolörü ifade eden bir blok diyagramı Şekil 2.3'de görülmektedir.



Şekil 2.3. PI Kontrolörü ifade eden bir blok diyagramı

Oransal kontrolde kararlı hal hatası kontrol işaretinin değişmediği denge noktasına erişince oluşur. Kontrolör hata sıfır olmadığı sürece bir artan işaret üretirse, kararlı hal hatası giderilebilir. Bu, integral kontrolörün prensibidir. İntegral kontrolör sistemin yükselme zamanını arttırır, fakat geçici cevabı iyileştirecek yönde etkilemez. Bu nedenle oransal kontrolör ile birlikte kullanılması yararlı olur [7]. Şekil 2.4'te sabit oransal ve türev katsayılarında integral parametresinin değiştirilmesi sonucu birim basamak yanıtındaki değişim gözlenebilir.



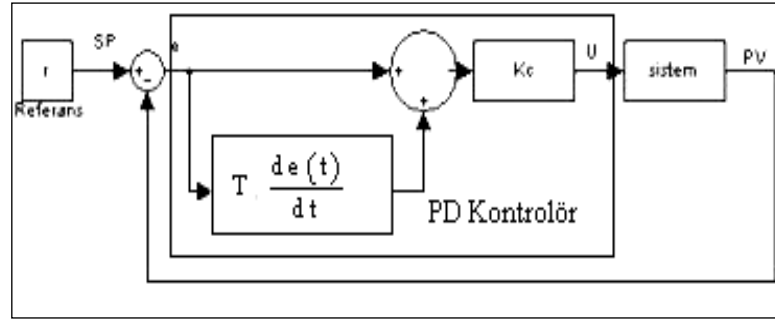
Şekil 2.4. I katsayısının basamak yanıtındaki etkisi

### 2.2.3 Oransal-Türevsel (PD) kontrolör

Oransal ve türevsel katsayıların aktif olduğu PID kontrolör türüdür. PD kontrol kuralı; u kontrol işareti, e hata işareti,  $K_C$  oransal katsayı ve  $T_D$  türev zaman sabiti olmak üzere,

$$U(t) = K_C \left( e(t) + T_D \frac{de(t)}{dt} \right) \quad (2.6)$$

biçiminde ifade edilir [6]. PD kontrolörü ifade eden bir blok diyagramı Şekil 2.5'te gösterilir.

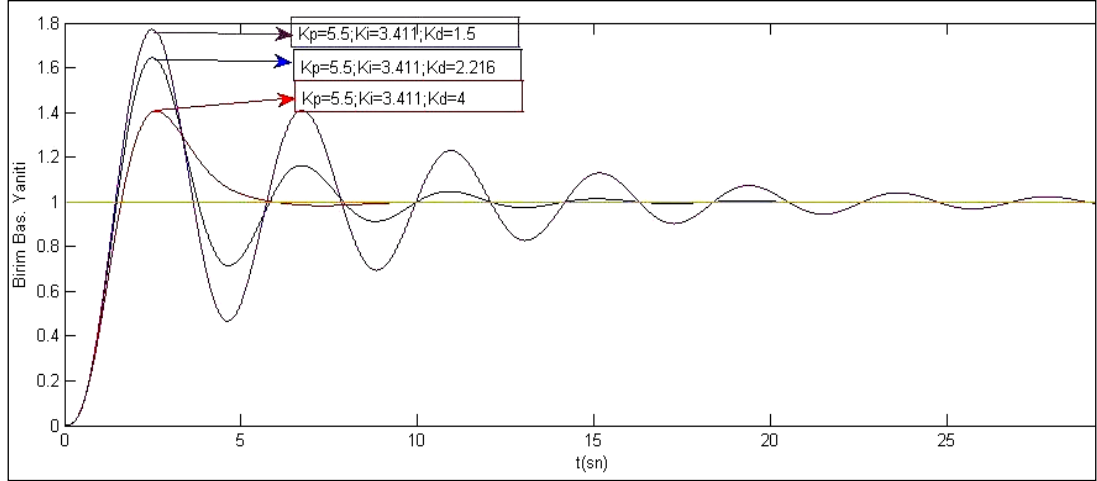


Şekil 2.5. PD Kontrolörü ifade eden bir blok diyagramı

İntegral kısmı hata kalmadığı halde bile kontrol işareti üretmeye devam eder, bunun sonucu olarak sistemde salınımlar oluşur. Salınımların önüne geçmek için kontrolöre hatanın sıfıra yaklaştığı iletilmelidir. Bu, hatanın türevi alınarak yapılabilir. Türev ancak oransal kontrolle beraber kullanılabilir [7].

Türevsel terim kontrolör çıkışındaki değişim oranını yavaşlatır ve bu etki kontrolör setpoint değerine en yakın olandır. Böylece, türevsel kontrol integral bileşeninin ürettiği aşım miktarını azaltır ve kontrolör işlem kararlılığını artırırken sinyal farklılaşması gürültüyü artırır. Böylece kontrolördeki bu terim, hata değerindeki gürültüye karşı çok hassas olur. Bu durum, gürültünün ve/veya türevsel terimin çok büyük olması halinde işlemin kararsızlaşmasına neden olabilir [2]. Şekil 2.6'da sabit oransal ve integral katsayılarında türev parametresinin değiştirilmesi sonucu birim basamak yanıtındaki değişim gözlenebilir.



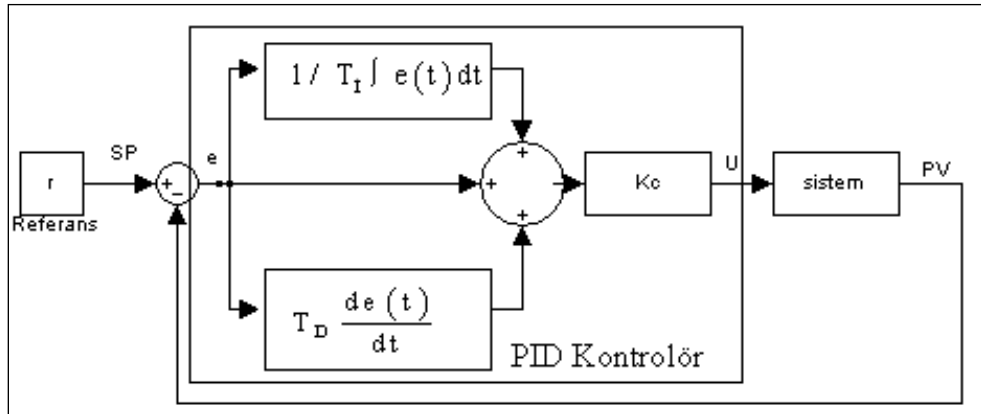


Şekil 2.6. D katsayısının basamak yanıtındaki etkisi

## 2.2.4 Oransal-İntegral-Türevsel (PID) kontrolör

Tüm PID katsayılarının aktif olduğu PID kontrolör türüdür. PD kontrolörün sisteme hassaslık getirdiği ancak sistemin kararlı hal davranışını etkilemediği belirtilmiştir. PI kontrolörün ise, göreceli kararlılığı ve aynı zamanda kararlı hal hatalarını düzelttiği, ancak yükselme zamanını arttırdığı belirtilmiştir. Bu sonuçlar bizi, PI ve PD kontrolörlerinin iyi yönlerinden yararlanmayı sağlayan, PID kontrolörünü kullanmaya yöneltir [5]. PID kontrolün endüstriyel uygulamalarda kullanılan ifadesi Denklem (2.2)' de verilmiştir.

PID kontrolörü ifade eden bir blok diyagramı Şekil 2.7'de belirtilmiştir.



Şekil 2.7. PID Kontrolörü ifade eden bir blok diyagramı

### 2.3 Türev Vuruşu

Çoğu endüstriyel uygulamada PI-D olarak adlandırılan ve,

$$e(t) = SP(t) - PV(t) \quad (2.7)$$

kontrol kuralı ifadesi Denklem (2.2)'de yerine koyulursa,

$$u(t) = K_c \left( SP(t) - PV(t) + \frac{1}{T_I} \int [SP(t) - PV(t)] dt + T_D \frac{d[SP(t) - PV(t)]}{dt} \right) \quad (2.8)$$

$SP(t)=SP$  sabit değer için sabitin türevi sıfır olacağından,

$$u(t) = K_c \left( SP(t) - PV(t) + \frac{1}{T_I} \int [SP(t) - PV(t)] dt - T_D \frac{dPV(t)}{dt} \right) \quad (2.9)$$

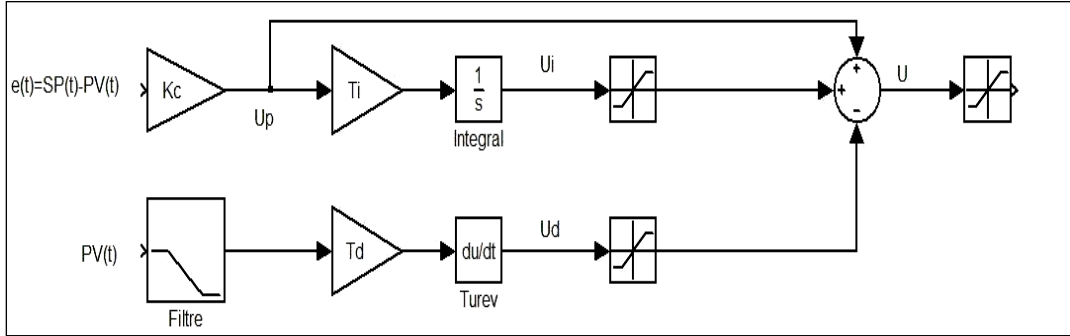
biçiminde yazılan ifadenin dijital olarak gerçekleşmesi ile elde edilen bir kontrolör türü kullanılır. Bu tür kullanım türev ya da referans vuruşu (derivative kick, set point kick) ve integral sarması (integral yığılması) olarak bilinen olayları önlemek içindir [6]. Referansın sık değiştiği durumlarda, bu şekilde anlık büyük türev değerleri engellenmiş olur.

### 2.4. İntegral Yığılması

Pratik uygulamalarda hiçbir sistemin girişine sonsuz büyüklükte bir işaret uygulanamamaktadır. Bütün sistemlerin girişine uygulanabilen kontrol işaretlerinin belli bir alt ve üst limitleri bulunmaktadır. Örneğin motorların belli hız limitleri bulunmaktadır ya da valflerin tamamen açık ve tamamen kapalı pozisyonları bulunmaktadır. Hesaplanan dijital kontrol işaretinin bu değerlerin dışına taşması durumunda bir sınırlama işlemi yapılmalıdır. Kapalı çevrim sistemlerde kullanılan bu sınırlayıcılar, doyma limitlerine ulaştıkları anda kontrolör girişine hala fark işareti geliyorsa kontrolör işlem yapmaya devam edecektir. Özellikle integratör tipli işlemciler kontrolörlerin yapısında bulunuyorsa doyma anında integral toplaması devam edeceği için bu terimler aşırı büyüyerek kontrolörün verimsiz çalışmasına neden olacaktırlar [10].

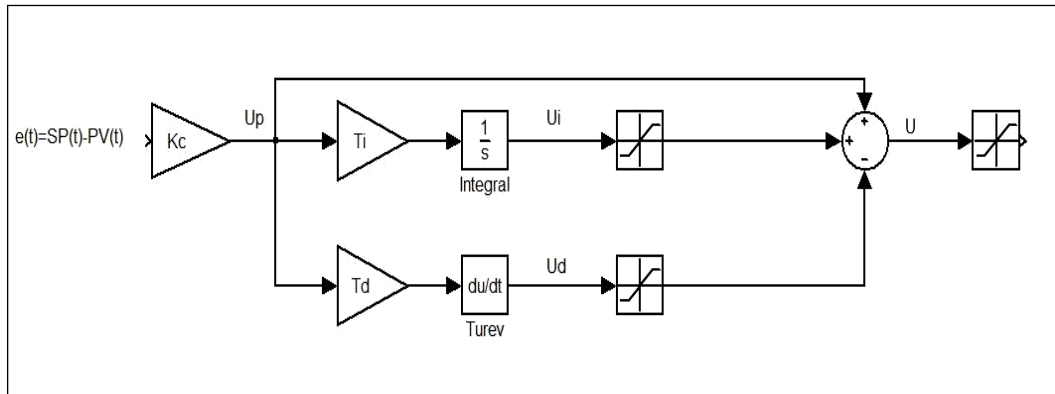
İntegral yığılması sorunu önlemek için çok çeşitli metotlar vardır. Üretici firmalar ürünlerinde çeşitli yollar kullanmalarına rağmen bunları bir imalat sırrı olarak saklamaktadırlar. Sorunu önlemenin bilinen en kolay yolu, kontrolör çıkışını sınırlandırmaktır, öyle ki kontrol işareti asla cihazın (motor vb.) sınır giriş değerini aşmamalıdır. Diğer bir metot ise kontrol işaretinin sürekli artıp doymaya ulaşması durumunda integral işlemini durdurmaktır. Çeşitli koşullara bağlı olarak integral işleminin devreye alınıp çıkarılmasını temel alan birçok farklı metot bulunmaktadır [2].

Şekil 2.8’de İntegral yığılması ve türev vuruşu olaylarını gidermesi için önerilen, işareti sınırlandırılmış bir PID kontrolör şeması verilmiştir.



Şekil 2.8. İşareti sınırlandırılmış bir PID kontrolörü yapısı

Endüstriyel uygulamalarda genel olarak Şekil 2.9’da verilen PID kontrolör yapısı kullanılır ve dijital PID yazılımı bu yapıya göre gerçekleştirilir.



Şekil 2.9. İşareti sınırlandırılmış genel bir PID kontrolörü yapısı

Bu tezde kullanılacak olan ve Bölüm 3’ te detaylı olarak inceleyeceğimiz dijital PID kontrolör formu, türev vuruşu ve integral yığılmasını kendisi önlemektedir [15].

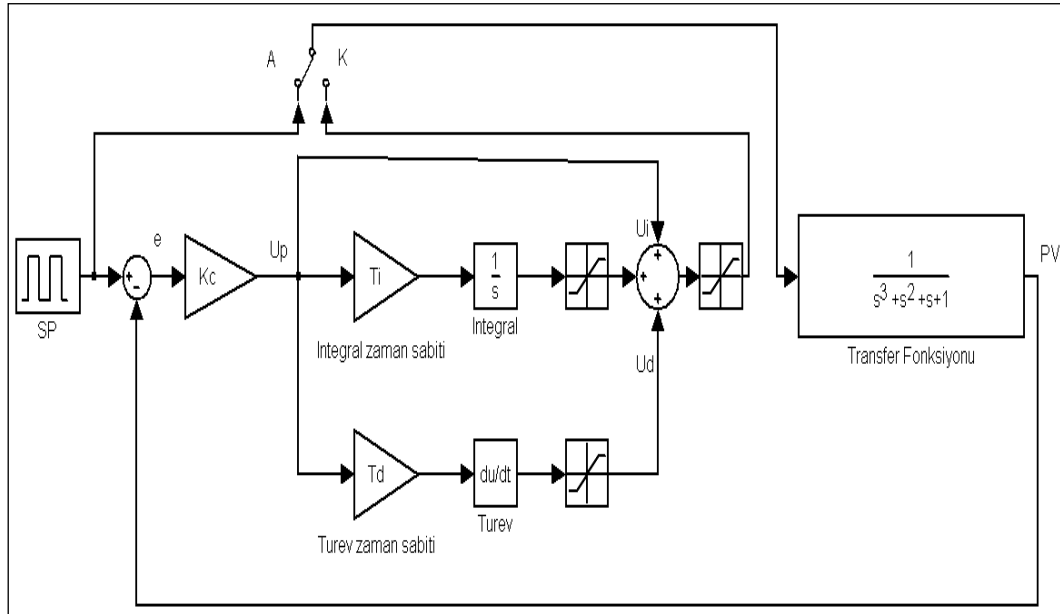
## 2.5. PID Parametrelerini Elde Etme Yöntemleri

PID kontrol parametrelerinin elde edilmesine ilişkin birçok yöntem mevcuttur. Yapılan çalışmada bu yöntemlerden endüstride en çok kullanılanlar incelenecektir.

Konuya ışık tutması için transfer fonksiyonu,

$$G(s) = \frac{1}{s^3 + 3s^2 + 3s + 1} \quad (2.10)$$

olan ve Şekil 2.10'da PID devre şeması gösterilen bir sistem ele alınacak, bu sistemin gösterilecek yöntemlere göre PID katsayıları elde edilecek, sisteme P,PI ve PID kontrolör uygulanıp sonuçları yorumlanacaktır. PD kontrolör; az sönümlü ya da kararsız sistemlerde etkili olmaması, pek çok durumda öngörülen hedefleri sağlamaması [5]. ve katsayılarının elde edilmesi amaçlı bir yöntemde sahip olunmaması gibi nedenlerden dolayı sonraki konularda ele alınmayacaktır.



Şekil 2.10. Örnek PID kontrolör şeması

### 2.5.1. Ziegler - Nichols yöntemi

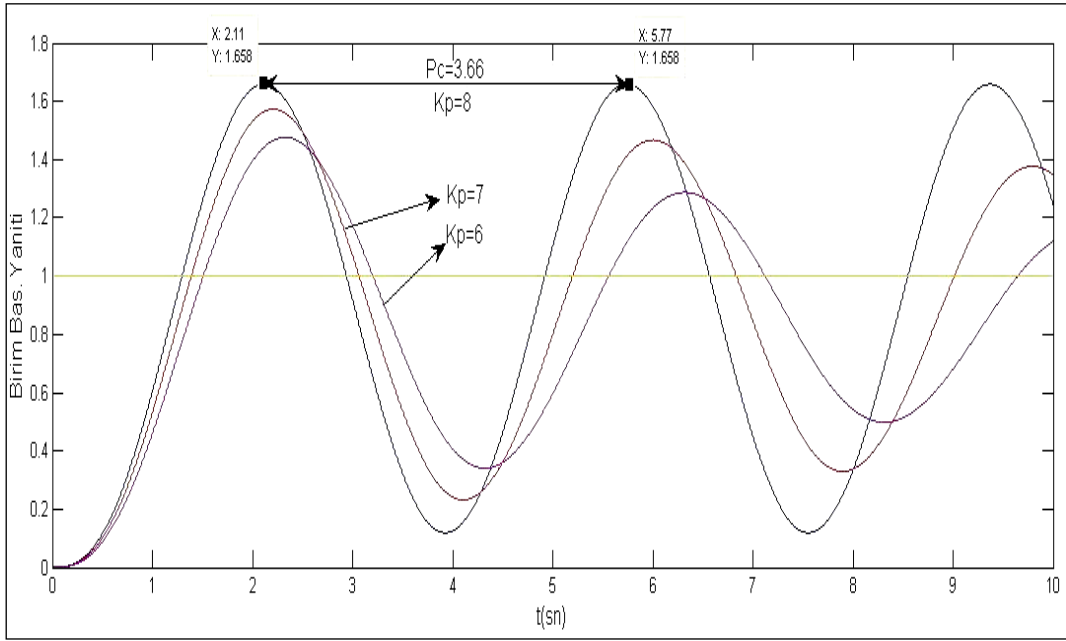
Ziegler-Nichols yöntemi PID kontrolörlerin parametrelerinin belirlenmesinde sıklıkla kullanılan yöntemlerden birisi olup sistem modeli gerektirmemesi, uygulananın basit olması ve yük bozuculara karşı dayanıklı olması, bu yöntemin sıklıkla

kullanılma nedenleri olarak gösterilebilir [11,12]. Ziegler-Nichols yöntemi osilasyon yöntemi ve birim basamak yanıtı yöntemi olmak üzere ikiye ayrılır [10].

### 2.5.1.1. Ziegler - Nichols osilasyon yöntemi

Ziegler - Nichols osilasyon yöntemi literatürde frekans yanıtı yöntemi, kapalı çevrim yöntemi ve sürekli döngü yöntemi olarak da geçmektedir. Yöntemi adım adım uygularsak:

1. Sisteme Şekil 2.11'deki gibi bir oransal kontrolör uygulanır.
2. Sistemin cevap eğrisi osilasyona girinceye kadar oransal kazanç ( $K_p$ ) sıfırdan başlanarak küçük adımlarla artırılır.



Şekil 2.11. Ziegler-Nichols osilasyon yönteminin uygulanması

3. Osilasyondaki sistemin oransal kazancı ( $K_U$ ) ve osilasyondaki sistemin periyodu ( $P_C$ ) kaydedilir. Verilen örnek için Şekil 2.11'de da görüldüğü gibi  $K_U=8$  ve  $P_C=3,66$  olarak hesaplanır.
4.  $K_U$  ve  $P_C$  parametreleri kullanılarak Tablo 2.1 yardımıyla PID parametreleri hesaplanır. Hesaplanan değerler Tablo 2.2'de belirtilmiştir.

Tablo 2.1. Osilasyon yöntemi katsayı hesap tablosu

	$K_P$	$T_I$	$T_D$
P	$0,5 \cdot K_U$	$\infty$	0
PI	$0,45 \cdot K_U$	$0,833 \cdot P_C$	0
PID	$0,6 \cdot K_U$	$0,5 \cdot P_C$	$0,125 \cdot P_C$

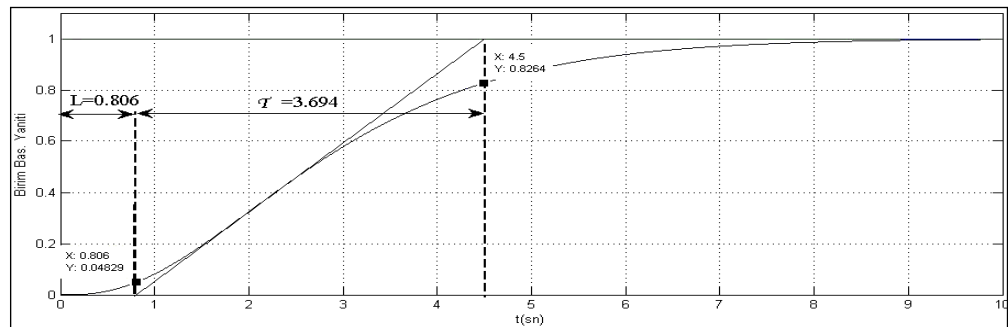
Tablo 2.2. Osilasyon yöntemi için hesaplanan değerler

	$K_P$	$T_I$	$T_D$	$K_I (K_P/T_I)$	$K_D (K_P \cdot T_D)$
P	4	$\infty$	0	0	0
PI	3,6	3,05	0	1,18	0
PID	4,8	1,83	0,45	2,62	2,16

### 2.5.1.2. Ziegler - Nichols birim basamak yanıtı yöntemi

Ziegler-Nichols birim basamak yanıtı yöntemi literatürde reaksiyon eğrisi yöntemi, açık çevrim yöntemi ve süreç eğrisi yöntemi olarak da geçmektedir. Bu yöntem, sistemin açık çevrim birim basamak yanıtı incelenerek,  $\tau$  (zaman sabiti) ve  $L$  (zaman gecikmesi) parametrelerinin elde edilmesine dayanır [10]. Yöntemi adım adım uygularsak:

1. Sistemin açık çevrim birim basamak yanıtı incelenerek  $\tau$  (zaman sabiti) ve  $L$  (zaman gecikmesi) parametreleri elde edilir. Bunun için birim basamak yanıtı eğrisinin eğiminin en yüksek olduğu noktaya ait Şekil 2.10'daki gibi bir doğru çizilir. Verilen örnek için Şekil 2.12'de görüldüğü gibi  $\tau = 3.694$ ,  $L = 0.806$  olarak hesaplanmıştır.



Şekil 2.12. Ziegler-Nichols birim basamak yönteminin uygulanması

2. Bulunan bu parametreler Tablo 2.3'teki formüllerde yerlerine konularak Tablo 2.4'teki PID katsayıları elde edilir.

Tablo 2.3. Ziegler-Nichols birim basamak yöntemi katsayı hesap tablosu

	$K_p$	$T_I$	$T_D$
P	$\frac{\tau}{K.L}$	$\infty$	0
PI	$0,9 \frac{\tau}{K.L}$	3.L	0
PID	$1,2 \frac{\tau}{K.L}$	2.L	0.5.L

Tablo 2.4. Ziegler Nichols birim basamak yöntemi için hesaplanan değerler

	$K_p$	$T_I$	$T_D$	$K_I$	$K_D$
P	4,583	$\infty$	0	0	0
PI	4,125	2,418	0	1,706	0
PID	5,5	1,612	0,403	3,411	2,216

### 2.5.2 Cohen - Coon yöntemi

Cohen-Coon yöntemi, Ziegler-Nichols birim basamak yanıtı yöntemine benzer ancak Cohen-Coon yönteminde PID katsayıları Tablo 2.5 yardımıyla Tablo 2.6'daki gibi elde edilir.

Tablo 2.5. Cohen-Coon yöntemi katsayı hesap tablosu

	$K_p$	$T_I$	$T_D$
P	$\frac{\tau}{K.L} (1 + \frac{L}{3.\tau})$	$\infty$	0
PI	$\frac{\tau}{K.L} (0,9 + \frac{L}{12.\tau})$	$L \frac{(30.\tau + 3.L)}{(9.\tau + 20.L)}$	0
PID	$\frac{\tau}{K.L} (\frac{4}{3} + \frac{L}{4.\tau})$	$L \frac{(32.\tau + 6.L)}{(13.\tau + 8.L)}$	$\frac{4.L.\tau}{11.\tau + 2.L}$

Tablo 2.6. Cohen-Coon yöntemi için hesaplanan değerler

	$K_p$	$T_I$	$T_D$	$K_I$	$K_D$
P	4,916	$\infty$	0	0	0
PI	4,21	1,85	0	2,27	0
PID	6,36	1,82	0,28	3,49	1,78

### 2.5.3 Haalman yöntemi

Haalman methodunda da Ziegler-Nichols birim basamak yanıtı yönteminde elde edilen parametreler Tablo 2.7'deki formüller yardımıyla kullanılır ve sisteme uygun PI kontrolör katsayıları Tablo 2.8'deki gibi elde edilir.

Tablo 2.7. Haalman yöntemi katsayı hesap tablosu

	$K_p$	$T_I$
PI	$\frac{2.\tau}{3.K.L}$	$\tau$

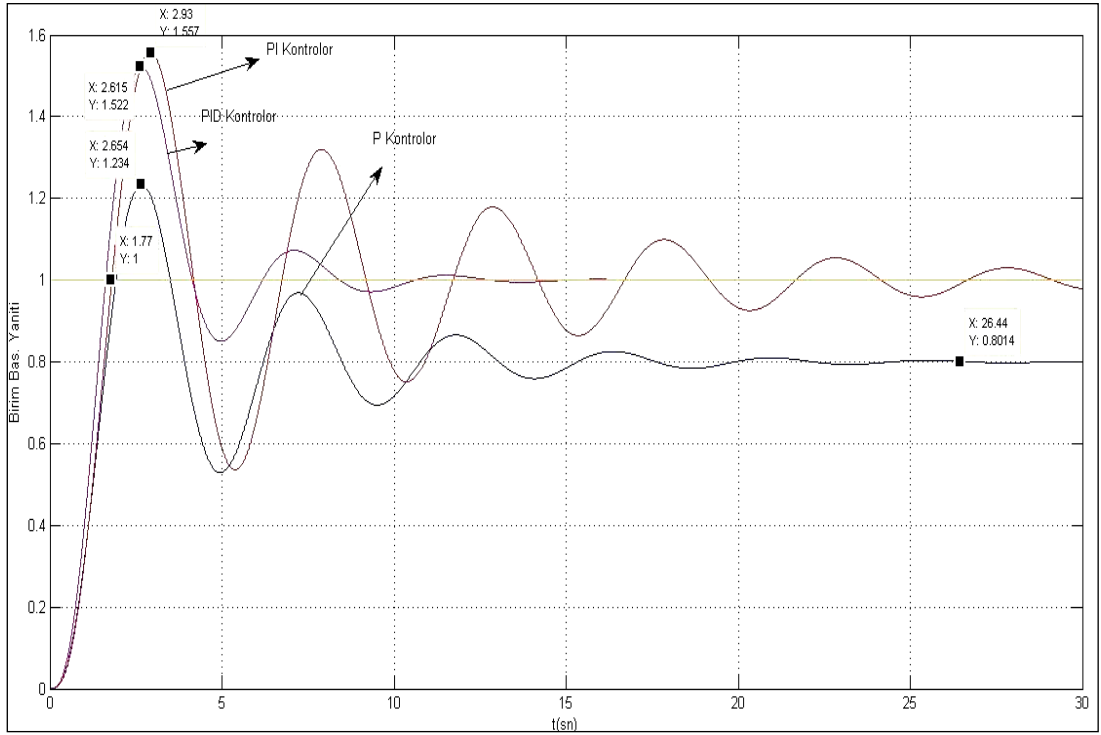
Tablo 2.8. Haalman yöntemi için hesaplanan değerler

	$K_p$	$T_I$	$K_I$
PI	3,1	3,752	0,83

### 2.6.Simulasyon Sonuçları

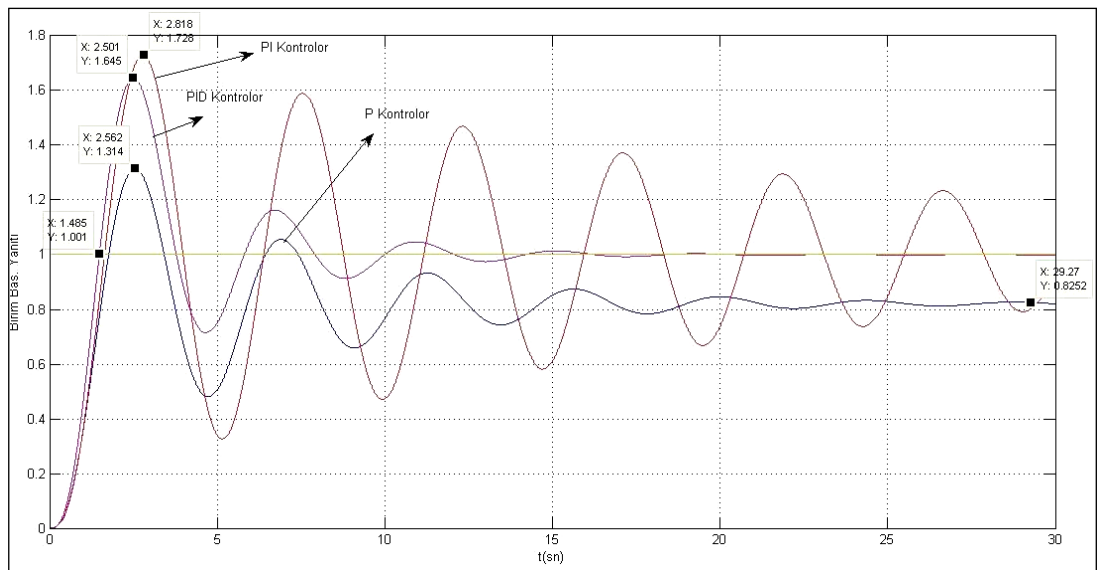
Şekil 2.13, Şekil 2.14, Şekil 2.15 ve Şekil 2.16'da (2.7) ile verilen örnek sistem için PID katsayılarını elde etme yöntemlerine ait P, PI ve PID kontrolör çıkışları belirtilmiştir. Bu sonuçlara göre Ziegler - Nichols osilasyon yönteminde her üç kontrolör türünde de diğer yöntemlere göre en büyük aşım miktarlarının düşük olduğu, PID kontrolörde oturma zamanının kısa olduğu ancak yükselme zamanının uzun olduğu Şekil 2.13'de görülebilir. Ancak bu yöntemde sistemin osilasyona girmesi her durumda mümkün olmadığından yöntemin kullanımı sınırlıdır.





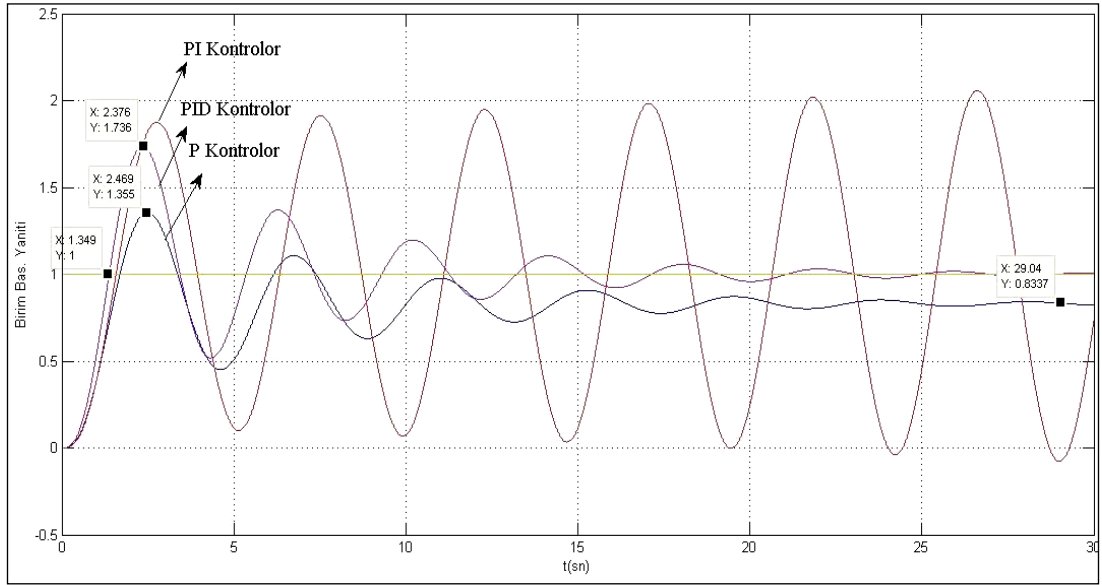
Şekil 2.13. Ziegler-Nichols osilasyon yöntemi ile P, PI ve PID kontrolör birim basamak yanıtları

Birim basamak yönteminde ise P kontrolördeki kazancın yüksek olduğu Şekil 2.14'de görülebilir. PID kontrolörde oturma zamanı osilasyon yöntemindekine yakın bir değerdedir. Bu yöntemde gözlenen olumsuz durum ise PI kontrolörde oturma zamanının çok yüksek olmasıdır.



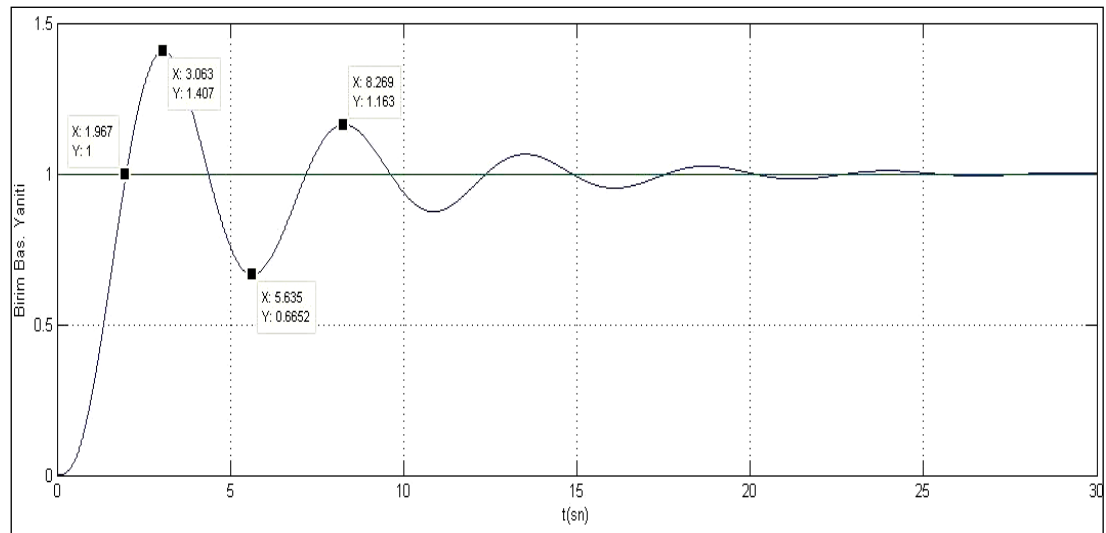
Şekil 2.14. Ziegler-Nichols birim basamak yanıtı yöntemi ile P, PI ve PID kontrolör birim basamak yanıtları

Cohen-Coon yönteminde her üç kontrolör türünde de diğer yöntemlere göre yükselme zamanının kısa olduğu ve P kontrolör kazancının yüksek olduğu Şekil 2.15'te görülebilir. Ancak her üç kontrolör türünde de en büyük aşım miktarı diğer yöntemlerdekinden fazladır. PI kontrolör türünde elde edilen PI kontrolör katsayıları uygulandığında ise sistem kararsızlaşmıştır.



Şekil 2.15. Cohen-Coon yöntemi ile P, PI ve PID kontrolör birim basamak yanıtları

Haalman yönteminde Şekil 2.16'te görüldüğü gibi, diğer yöntemlerdeki PI kontrolörlere göre oldukça iyi bir sonuç elde edilmiştir. Ancak yine de diğerlerine kıyasla yükselme zamanının daha uzun olduğunu belirtmek gerekir.



Şekil 2.16. Haalman yöntemi ile PI kontrolör birim basamak yanıtı

### 3. DİJİTAL PID KONTROL

Dijital kontrol sistemlerinin tasarımı analog kontrol sistemlerinin tasarım prensibine eşdeğerdir. Burada da hedef, sistemi kriterlere uygun bir şekilde davranmaya yönelten, bir kontrolör tasarlamaktır. Gerçekte, kontrol edilen sistem hep aynıdır, aradaki tek fark dijital sistemlerde kontrolörün dijital veriyi işleyebilecek yetenekte olmasıdır [5].

Endüstriyel dijital kontrol sistemine ilişkin kontrol kuralını gerçeklemek için kullanılan dijital kontrol biriminde, belirli zaman dilimlerinde alınan (ayrık zamanlı) veriler değerlendirilir. Kontrol kuralının ayrık zaman değerlerine göre işlenmesi nedeniyle sürece ilişkin ayrık zaman modeli (z-tanım bölgesi) üzerinde çalışmak daha uygundur. Özellikle, endüstriyel süreçlerde yaygın olarak karşılaşılan ölü zamanlı sistemlere ilişkin matematiksel model, hiçbir yaklaşıklık yapılmadan, doğrudan ayrık zamanlı olarak verilebilir. Bu şekilde, örnekleme zamanının hem süreç hem de kontrolöre etkisi, tasarım aşamasında değerlendirilebilir. Ayrıca kontrol edilen sisteme ilişkin gerçek zamanlı benzetim, ayrık zamanlı modelden yararlanılarak doğrudan gerçekleştirilebilir. Böylece kontrol sisteminin başarımı, dijital ortamda gerçek zamanlı olarak değerlendirilebilir.

Daha önce belirtilen (2.1) ve (2.2) denklemleri, analog PID kontrolörleri ifade etmektedir. Dijital PID kontrol bu denklemlerin z-tanım bölgesi ve fark denklemleri kavramlarından yararlanılarak ayrıklaştırılmasıyla sağlanır. Bu nedenle z-dönüşümü ve fark denklemleri kavramlarının açıklanmasında fayda vardır.

#### 3.1. Z-Dönüşümü

$y(k)$ 'nın  $k=0,1,\dots$  için, bir değerler dizisini ifade ettiğini varsayalım. z-dönüşümü, z gerçek ve sanal kısımları olan karmaşık bir değişken olmak üzere,

$$Y(z) = y(k)'nın z-dönüşümü = Z[y(k)] = \sum_{k=0}^{\infty} y(k)z^{-k} \quad (3.1)$$

olarak tanımlanır. (3.1) ifadesinin en önemli özelliği z-dönüşümünün gerçek tanım

bölgesindeki bir değerler dizisini, karmaşık z-tanım bölgesine dönüştürdüğüdür.

Analog kontrolörlerde kullanılan Laplace dönüşümleri ile dijital kontrolörlerde kullanılan z-dönüşümleri arasındaki ilişki aşağıdaki şekilde ifade edilebilir.

$k = 0, 1, 2, \dots$  için  $y(kT)$  değer dizisi,  $t$  zaman aralıkları ile ayrılmış, bir impuls dizisi olarak değerlendirilebilir.  $T$  zamanı örnekleme periyodu olarak bilinir.  $k$ 'nci andaki  $\delta(t-kT)$  impulsu  $y(kT)$ 'nin değerini taşır. Dijital kontrol sistemlerinde,  $y(t)$  işaretinin  $T$  saniye aralıklarla örneklenmiş haline karşı gelen bu zaman dizisine çok sık rastlanır. Buna göre  $y(kT)$  dizisi,

$$y^*(t) = \sum_{k=0}^{\infty} y(kT)\delta(t-kT) \quad (3.2)$$

Şeklinde bir işaretle ilişkilendirilebilir. Denklem (3.2)'nin Laplace dönüşümü alınırsa

$$Y^*(s) = \mathcal{L}[y^*(t)] = \sum_{k=0}^{\infty} y(kT)e^{-kTs} \quad (3.3)$$

elde edilir. Denklem (3.3), Denklem (3.1) ile karşılaştırılırsa  $z$  ve Laplace dönüşümlerinin birbirleriyle

$$z = e^{Ts} \quad (3.4)$$

ifadesi ile ilişkilendirilebileceği görülür. Gerçekten de (3.4) tanımında  $z$ -dönüşümünün  $T=1$  özel durumuna karşı düştüğü görülür. (3.4) ile verilen  $z$ -dönüşüm tanımı bize örneklenmiş sistemleri işlemeye ve analog sistemlerin dijital karşılıklarını elde etmeye olanak sağlar. Sonuç olarak  $z$ -dönüşüm tanımı,

$$Y(z) = \mathcal{Z}[y(kT)] = \mathcal{Z}[y^*(t)] = \mathcal{Z}[Y^*(s)] = Y^*(s) \Big|_{z=e^{Ts}} \quad (3.5)$$

şeklinde özetlenebilir ya da,

$$y(z) = \mathcal{Z}[y(t)] = \mathcal{Z}Y(s) \quad (3.6)$$

yazılabilir; buna göre  $y(t)$  işaretinin,  $z$ -dönüşümü alınmadan önce  $y^*(t)$ 'yi elde etmek üzere, örneklendiği ya da ayrıklaştığı anlaşılır [5].

Örnek:  $y(t) = e^{-\alpha t} U_s(t)$  zaman fonksiyonunu ele alalım.  $y(t)$ 'nin z-dönüşümü aşağıdaki basamaklar aşılarak elde edilir:

1.  $y(t)$ 'nin,  $k=0,1,2,\dots$ , olmak üzere,  $t=kT$  anlarındaki değerlerini veren  $y^*(t)$  fonksiyonunu oluşturur:

$$y^*(t) = \sum_{k=0}^{\infty} e^{-\alpha kT} \delta(t - kT) \quad (3.7)$$

2. (3.7) ilişkisinin Laplace dönüşümü alınır:

$$Y^*(s) = \sum_{k=0}^{\infty} e^{-\alpha kT} e^{-kTs} = \sum_{k=0}^{\infty} e^{-(s+\alpha)kT} \quad (3.8)$$

3. (3.4) uygulanarak  $Y^*(s)$ 'in z-dönüşümü

$$Y(z) = \frac{z}{z - e^{-\alpha T}} \quad (3.9)$$

elde edilir.

N pozitif bir tamsayı olmak üzere, eğer iki fonksiyondan biri,  $e^{-NTs}$  geciktirme fonksiyonu ise,

$$Z[e^{-NTs} Y(s)] = Z[e^{-NTs}] Z[Y(s)] = z^{-N} Y(z) \quad (3.10)$$

ilişkisi yazılabilir [5]. Bu ilişki ilerleyen konularda ölü zamanlı sistemler için kullanılacaktır.

### 3.2. Fark Denklemleri

Dijital kontrolörlerin kullanıldığı kontrol sistemlerinde dijital ve analog işaretleri ilişkilendiren denklemlerden yararlanmak gerekir. Diferansiyel denklemlerin analog sistemleri belirlediği gibi, fark denklemleri de, dijital sistemleri belirler. Fark denklemleri, bilgisayarda daha kolay programlanıp çözülebildikleri için, ayrıca diferansiyel denklemleri yaklaşık ifade etmede de kullanılır [5].

Genelde n'inci mertebeden sabit katsayılı doğrusal bir fark denklemi,

$$y(k) = a_1y(k-1) + a_2y(k-2) + \dots + a_ny(k-n) + b_1u(k-1) + b_2u(k-2) + \dots + b_nu(k-n) \quad (3.11)$$

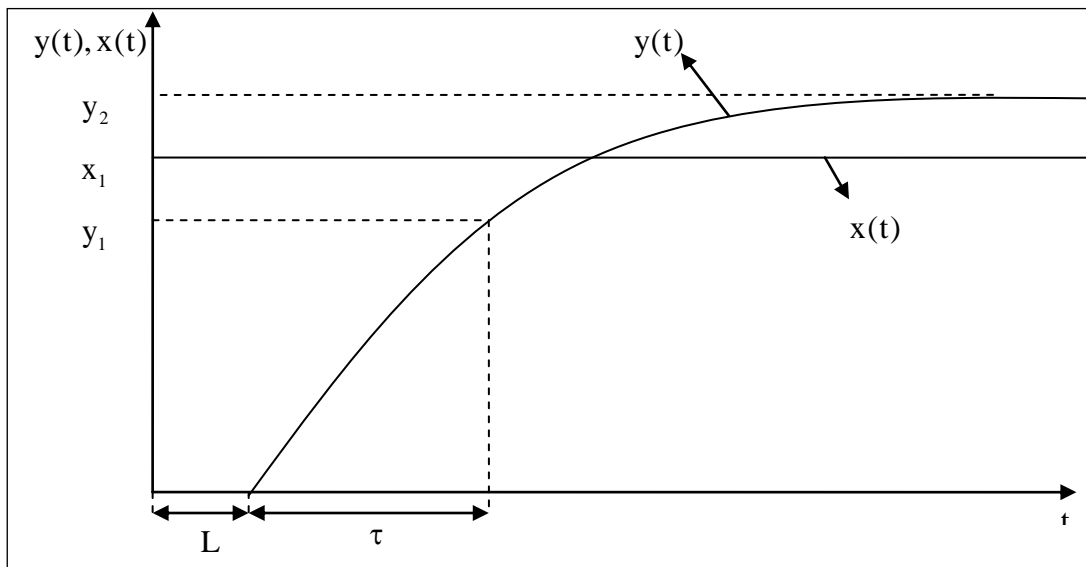
şeklinde yazılabilir; burada özellikle bağımsız değişkenin zaman olması nedeniyle  $i=k, k-1, \dots, k-n$  için,  $y(i)$  bağımlı değişkeni  $y$ 'nin  $i$  anındaki değerini ifade eder. Genelde bağımsız değişken herhangi bir gerçek büyüklük olabilir.

### 3.3. Endüstride Sistemlerde PID Kontrol Parametreleri ve Fark Denklemlerinin Belirlenmesi

Sistemler mertebesine göre düşük mertebeden ve yüksek mertebeden sistemler olarak iki şekilde ifade edilebilir. Düşük mertebeden sistemler de kendi içinde 1.mertebeden ve 2.mertebeden olarak ikiye ayrılır. Zaman gecikmesine göre de sistemler ölü zamanlı ve ölü zamansız olarak ikiye ayrılır. Bu bölümde endüstride en çok kullanılan sistemler ve bu sistemlerin PID kontrol parametrelerinin ve fark denklemlerinin elde edilmesi ile yüksek mertebeden sistemlerin düşük mertebeden sistem cinsinden ifade edilme yöntemi anlatılacaktır.

#### 3.3.1. Birinci mertebeden sistemler

Birim basamak girişi uygulandığında çıkışın zamana göre değişimi şekil 3.1'de verilen sistemlerdir.

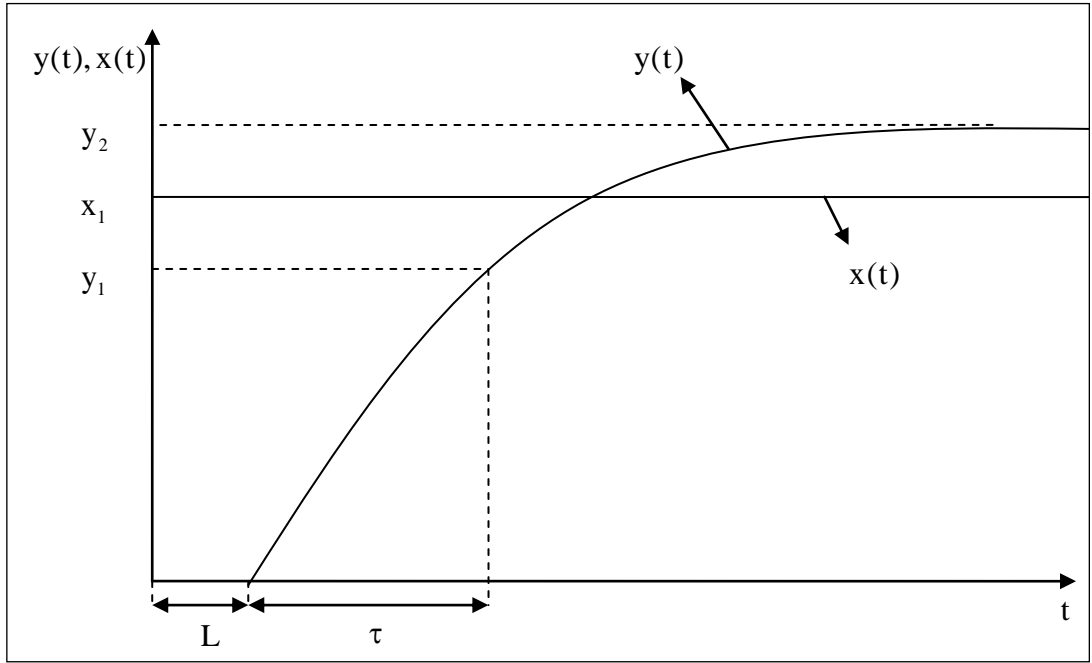


Şekil 3.1. Birinci mertebeden sistemin basamak girişe yanıtı

Birinci mertebeden bir sisteme ilişkin s-tanım bölgesi transfer fonksiyonu

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K}{(\tau s + 1)} \quad (3.12)$$

Biçiminde verilir. Birçok endüstriyel sürecin basamak girişe yanıtı, ölçme veya sürücü düzeneklerinden kaynaklanan gecikmeler nedeniyle Şekil 3.2'deki gibi değişir. Bu tür sistemler birinci mertebeden ölü zamanlı sistem olarak tanımlanır [6].



Şekil 3.2. Birinci mertebeden ölü zamanlı sistemin basamak girişe yanıtı

Birinci mertebeden ölü zamanlı bir sisteme ilişkin s-tanım bölgesi transfer fonksiyonu,

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K}{(\tau s + 1)} e^{-sL} \quad (3.13)$$

olarak verilir. Denklem (3.12) ve Denklem (3.13) ifadelerinde K kazanç,  $\tau$  zaman sabiti ve L sürecin giriş işaretine tepkisiz olduğu ölü zaman değeri ya da kısaca ölü zaman olarak tanımlanır. Kazanç büyüklüğü,

$$K = \frac{y_2}{y_1} \quad (3.14)$$

oranından elde edilir. Zaman sabitinin elde edilmesinde ise,

$$y(L + \tau) = y_1 = 0,6321y_2 \quad (3.15)$$

ifadesinden yararlanılır. Buna göre sisteme genliği  $x_1$  olan bir basamak giriş işareti uygulanırsa,  $L + \tau$  süre sonra çıkış son değerinin %63,21 oranına ulaşılır. Buna göre zaman sabiti  $\tau$ , çıkış son değerinin %63,21 oranına ulaşılmaya kadar geçen süreden ölü zaman değeri çıkartılarak elde edilir [6]. Sistem ölü zamansız ise ölü zaman değeri  $L=0$  alınır.

### 3.3.1.1. Birinci mertebeden sistemlerde PID katsayılarının bulunması

Birinci mertebeden bir sistemin açık çevrim davranışı ile kapalı çevrim davranışının aynı ve sürekli durum hatasının sıfır olmasını sağlayan bir kontrol kuralı PI kontrolör ile gerçekleştirilebilir. Bunun için öncelikle örnekleme zamanı  $T$  seçilir.

Örnekleme zamanı  $T$ , açık çevrim sisteminin köşe frekansındaki ( $w_c$ ) genliğinin 40db/dec zayıfladığı frekansın iki katı örnekleme frekansı ( $w_s$ ) olarak seçilirse birinci mertebeden sistemler için,

$$w_s = 2w_c = 2 \times \frac{1}{\tau} \times 100 = \frac{200}{\tau} \quad (3.16)$$

olur ve örnekleme zamanı,

$$T = \frac{2\pi}{w_s} = \frac{2\pi}{200} \tau \cong 0,0314\tau, \quad T \cong \frac{\tau}{30} \quad (3.17)$$

olarak seçilebilir. Ancak uygulamalarda,

$$\frac{\tau}{30} \leq T \leq \frac{\tau}{5} \quad (3.18)$$

arasındaki değerler kullanılabilir. Örnekleme zamanının alt sınırını PID kontrol algoritmasının işletildiği organizasyon bloğunun (OB30,...,OB38) çevrim süresi ve PLC tarama çevrim süresi belirler [6,14].



S-tanım bölgesi transfer fonksiyonu Denklem (3.12) ifadesinde verilen birinci mertebeden sistemin sıfırıncı mertebeden tutucu ile elde edilen ayrık zaman modeline ilişkin z-tanım bölgesi transfer fonksiyonu, T örnekleme zamanı olmak üzere,

$$G(z) = Z\{G_{\text{zoh}}(s)G(s)\} = Z\left\{\frac{1-e^{-sT}}{s} \frac{K/\tau}{s+1/\tau}\right\} = \frac{1-e^{-sT}}{1} Z\left\{\frac{K/\tau}{s(s+1/\tau)}\right\} \quad (3.19)$$

ifadesinden,

$$G(z) = \frac{Y(z)}{X(z)} = \frac{K(1-e^{-T/\tau})}{z-e^{-T/\tau}} \quad (3.20)$$

olarak elde edilir. Burada örnekleme zamanına bağlı olarak,

$$A = e^{-T/\tau} \quad (3.21)$$

tanımlaması yapılırsa denklemin son hali,

$$G(z) = \frac{Y(z)}{X(z)} = \frac{K(1-A)}{z-A} \quad (3.22)$$

olarak elde edilir. Sistem ölü zamanlı ise Denklem (3.4) ve (3.10) eşitlikleri kullanılarak ölü zaman değerinin örnekleme zamanına oranı,

$$d = \frac{L}{T} \quad (3.23)$$

olmak üzere,

$$e^{-sL} = (e^{sT})^{-d} = z^{-d} \quad (3.24)$$

ifadesi Denklem (3.22) ifadesi ile çarpılarak ölü zamanlı sistemin z-tanım bölgesi transfer fonksiyonu,

$$G(z) = \frac{Y(z)}{X(z)} = \frac{K(1-A)}{z-A} z^{-d} \quad (3.25)$$

olarak elde edilir.

Bu katsayılar yardımıyla PI kontrolör kuralını oluşturacak olan integral sabiti  $T_I$  ve kazanç sabiti  $K_C$  parametreleri,

$$A_c = \frac{d}{d+1} \quad (3.26)$$

olmak üzere,

$$T_I = \frac{T}{1-A} \quad (3.27)$$

$$K_C = \frac{(1-A_c)A_c^d}{K(1-A)} \quad (3.28)$$

ifadeleri kullanılarak elde edilir [6,14].

### 3.3.1.2. Birinci mertebeden sistemlerin fark denklemlerinin bulunması

Denklem (3.36) ifadesi yeniden düzenlenirse,

$$G(z) = \frac{Y(z)}{X(z)} = \frac{K(1-A)}{z-A} z^{-d} \quad (3.29)$$

olarak elde edilir. Bu denklemden yola çıkarak birinci mertebeden ölü zamanlı sistemler için fark denklemi,

$$Y(z) = \frac{K(1-A)}{z-A} z^{-d} X(z) = \frac{K(1-A)}{1-Az^{-1}} z^{-d-1} x(z)$$

$$Y(z) = Az^{-1}Y(z) + K(1-A)z^{-d-1}X(z)$$

$$y(k) = Ay(k-1) + K(1-A)x(k-1-d) \quad (3.30)$$

olarak elde edilir. Denklem (3.30) ifadesinde  $d=0$  yapılırsa ölü zamansız sistem için geçerli olan fark denklemi elde edilir [4,6,14].

### 3.3.2. İkinci mertebeden sistemler

İkinci mertebeden sistemler, s-tanım bölgesi transfer fonksiyonu,

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.31)$$

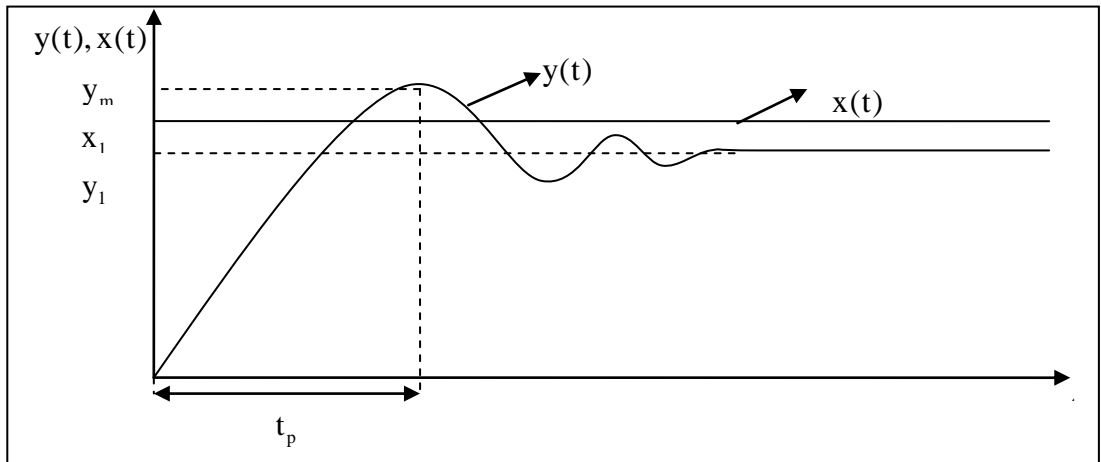
biçiminde verilen sistemler olarak tanımlanır. Burada  $\zeta$  : sönüm oranı,  $\omega_n$  : doğal frekans olarak tanımlanır. Bu tür sistemler  $\zeta$  parametresine göre az sönümlü, kritik sönümlü, aşırı sönümlü ve sönümsüz sönümlü sistemler olarak sınıflandırılır. Sönüm oranı  $0 < \zeta < 1$  değerinde olan sistemler az sönümlü, sönüm oranı  $\zeta = 1$  değerinde olan sistemler kritik sönümlü, sönüm oranı  $\zeta > 1$  değerinde olan sistemler aşırı sönümlü ve sönüm oranı  $\zeta = 0$  olan sistemler sönümsüz olarak tanımlanır [6]. Bu çalışmada uygulamalarda en çok kullanılan az sönümlü ve kritik sönümlü sistemler anlatılacaktır.

### 3.3.2.1. İkinci mertebeden az sönümlü sistemler

İkinci mertebeden az sönümlü bir sisteme ilişkin s-tanım bölgesi transfer fonksiyonu,

$$G(s) = \frac{Y(s)}{R(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (3.32)$$

biçiminde verilir. Bu tür sistemlerin birim basamak girişine yanıtı Şekil 3.3'te verildiği gibi değişir ve sistem yanıtından alınan  $y_m$ ,  $y_1$  ve  $t_p$  değerleri kullanılarak K kazanç,  $\zeta$  sönüm oranı ve  $\omega_n$  doğal frekans parametreleri Şekil 3.3' teki gibi bulunabilir.



Şekil 3.3. İkinci mertebeden az sönümlü sistemin basamak girişe yanıtı

Aşım c,

$$c = \frac{y_m - y_1}{y_1} \quad (3.33)$$

olarak tanımlanır ve bu büyüklüğe bağlı olarak,

$$\delta = \ln(c), \quad 0 < y_1 < y_m < 2y_1 \quad (3.34)$$

olarak tanımlanırsa sönüm oranı,

$$\zeta = \frac{\delta}{\sqrt{\pi^2 + \delta^2}} \quad (3.35)$$

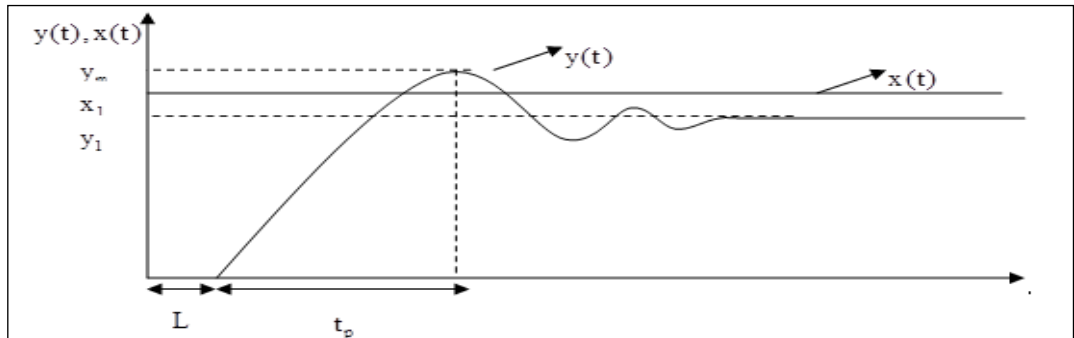
ifadesi kullanılarak bulunur.

Doğal frekans  $\omega_n$  ,

$$\omega_n = \frac{\pi}{t_p \sqrt{1 - \zeta^2}}, \quad 0 < \zeta < 1 \quad (3.36)$$

olarak elde edilir. K kazanç büyüklüğü ise Denklem (3.14) ifadesinde olduğu gibi çıkış işaretinin son değerinin basamak giriş işaretine oranından elde edilir [6].

İkinci mertebeden az sönümlü sistemde ölü zaman bulunması durumu için s-tanım bölgesi transfer fonksiyonu Denklem (3.37) ifadesinde belirtilmiştir. Ölü zamanlı sistemin birim basamak yanıtı Şekil 3.4'te gösterilmiştir. L, ölü zaman değerini ifade etmektedir.



Şekil 3.4. İkinci mertebeden az sönümlü ölü zamanlı sistemin basamak girişe yanıtı

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-sL} \quad (3.37)$$

İkinci mertebeden az sönümlü sistemler için PID katsayıları aşağıdaki adımlar uygulanarak bulunabilir.

İkinci mertebeden bir sistemin açık çevrim davranışı ile kapalı çevrim davranışının aynı ve sürekli durum hatasının sıfır olmasını sağlayan bir kontrol kuralı PID kontrolör ile gerçekleştirilebilir. Bunun için öncelikle örnekleme zamanı  $T$  seçilir.

Örnekleme zamanı  $T$ , açık çevrim sisteminin köşe frekansındaki ( $\omega_c$ ) genliğinin 40db/dec zayıfladığı frekansın iki katı örnekleme frekansı ( $\omega_s$ ) olarak seçilebilir. İkinci mertebeden bir sistemin genliği doğal frekansın yaklaşık 10 katı bir değerde 40db/dec zayıflar. Bu değerın 2 katı  $\omega_s$  olarak tanımlanırsa ikinci mertebeden sistemler için [6,14],

$$\omega_s = 2 \times 10 \times \omega_n = 20\omega_n \text{ rad / sec} \quad (3.38)$$

olur ve örnekleme zamanı,

$$T = \frac{2\pi}{\omega_s} = \frac{2\pi}{20\omega_n} = \frac{\pi}{10\omega_n} \cong \frac{0,0314}{\omega_n} \quad (3.39)$$

olarak elde edilir.

İkinci mertebeden az sönümlü bir sistemin ayrık zaman modeline ilişkin sıfırıncı mertebeden tutucu ile elde edilen z-tanım bölgesi transfer fonksiyonu Denklem (3.4) eşitliği kullanılarak,

$$G(z) = Z\{G_{\text{zOH}}(s)G(s)\} = K(1-z^{-1})Z\left\{\frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)}\right\} \quad (3.40)$$

İfadesinden hareketle,

$$G(z) = K \frac{(1 - a \cos \theta - ab \sin \theta)z + a^2 - a \cos \theta + ab \sin \theta}{z^2 - (2a \cos \theta)z + a^2} \quad (3.41)$$

biçiminde elde edilir. Bu ifadede  $a$ ,  $b$  ve  $\theta$  büyüklükleri aşağıda belirtilmiştir.

$$a = e^{-\zeta\omega_n T} \quad (3.42)$$

$$b = \frac{\zeta}{\sqrt{1-\zeta^2}} \quad (3.43)$$

$$\omega = \omega_n \sqrt{1-\zeta^2} \quad (3.44)$$

$$\theta = \omega T \quad (3.45)$$

(3.41) ifadesi yeniden düzenlenirse,

$$G(z) = K(1 - a \cos \theta - ab \sin \theta) \frac{z + \frac{a^2 - a \cos \theta + ab \sin \theta}{1 - a \cos \theta - ab \sin \theta}}{z^2 - (2a \cos \theta)z + a^2} \quad (3.46)$$

olarak yazılır ve,

$$b_0 = \frac{a^2 - a \cos \theta + ab \sin \theta}{1 - a \cos \theta - ab \sin \theta} \quad (3.47)$$

$$K_z = K(1 - a \cos \theta - ab \sin \theta) \quad (3.48)$$

$$a_1 = -2a \cos \theta \quad (3.49)$$

$$a_0 = a^2 \quad (3.50)$$

tanımları yapılırsa,

$$G(z) = \frac{K_z(z + b_0)}{z^2 - a_1 z + a_0} \quad (3.51)$$

olarak elde edilir. Sistem ölü zamanlı ise birinci mertebeden sistemde olduğu gibi Denklem (3.23) ve (3.24) ifadeleri kullanılarak ölü zamanlı sistemin z-tanım bölgesi transfer fonksiyonu,

$$G(z) = \frac{K_z(z + b_0)}{z^2 - a_1 z + a_0} z^{-d} \quad (3.52)$$

biçiminde elde edilir.

$$b_1 = \frac{b_0(d+2) - d}{(d+1)} \quad (3.53)$$

$$z_b = \frac{-b_1 + \sqrt{b_1^2 + 4b_0}}{2} \quad (3.54)$$

$$K^* = \frac{z_b^{d+1}(1-z_b)}{z_b + b_0} \quad (3.55)$$

olmak üzere PID katsayıları,

$$T_d = \frac{a^2}{1-a^2} T \quad (3.56)$$

$$T_I = \frac{1-a^2}{a^2 - 2a \cos \theta + 1} T \quad (3.57)$$

$$K_C = \frac{K^* T}{K_Z(T + T_d)} \quad (3.58)$$

ifadeleri kullanılarak elde edilir [6,14].

İkinci mertebeden az sönümlü sistemin fark denklemi Denklem (3.52) ifadesi kullanılarak,

$$G(z) = \frac{Y(z)}{X(z)} = \frac{K_Z(z+b_0)}{z^2 - a_1 z + a_0} z^{-d} = \frac{K_Z(z^{-1} + b_0 z^{-2}) z^{-d}}{1 + a_1 z^{-1} + a_0 z^{-2}} = \frac{K_Z(z^{d-1} + b_0 z^{d-2})}{1 + a_1 z^{-1} + a_0 z^{-2}} \quad (3.59)$$

$$(1 + a_1 z^{-1} + a_0 z^{-2}) Y(z) = K_Z(z^{d-1} + b_0 z^{d-2}) X(z) \quad (3.60)$$

$$Y(z) = -a_1 z^{-1} Y(z) - a_0 z^{-2} Y(z) + K_Z[z^{d-1} X(z) + b_0 z^{d-2} X(z)] \quad (3.61)$$

$$y(k) = -a_1 y(k-1) - a_0 y(k-2) + K_Z[x(k-d-1) + b_0 x(k-d-2)] \quad (3.62)$$

biçiminde bulunur. Bu ifadede  $d=0$  yapılırsa ölü zamansız sistem için geçerli olan fark denklemi bulunur [6].

### 3.3.2.2. İkinci mertebeden kritik sönümlü sistemler

İkinci mertebeden kritik sönümlü bir sisteme ilişkin s-tanım bölgesi transfer fonksiyonu Denklem (3.45) ifadesinde  $\zeta=1$  yazılarak,

$$G(s) = \frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{K\omega_n^2}{s^2 + 2\omega_n s + \omega_n^2} = \frac{K\omega_n^2}{(s + \omega_n)^2} \quad (3.63)$$

biçiminde veya,

$$\omega_n = \frac{1}{\tau} \quad (3.64)$$

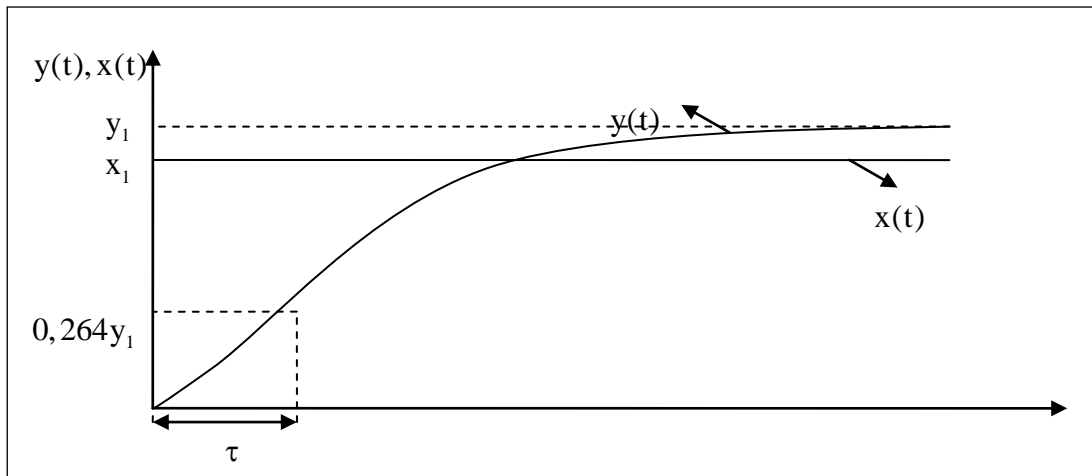
olarak alınıp Denklem (3.63) ifadesinde yerine konulursa zaman sabiti ve kazanç parametrelerine bağlı olarak,

$$G(s) = \frac{K}{(\tau s + 1)^2} \quad (3.65)$$

biçiminde tanımlanabilir. Bu tür sistemlerin birim basamak girişine yanıtı Şekil 3.5'te verildiği gibi değişir ve Şekil 3.5'teki  $y(t)$  eğrisinden K kazanç ve  $\tau$  zaman sabiti parametreleri elde edilir. K sabiti yine Denklem (3.31) ifadesindeki gibi çıkış işaretinin son değerinin basamak giriş işaretine oranından elde edilir.  $\tau$  sabiti ise [6],

$$y(t) = y(\tau) = 0,264y_1 \quad (3.66)$$

İlişkisinden elde edilir. Buna göre  $\tau$  zaman sabiti, çıkış işaretinin  $0,264y_1$  değerine ulaştığı andaki süredir.

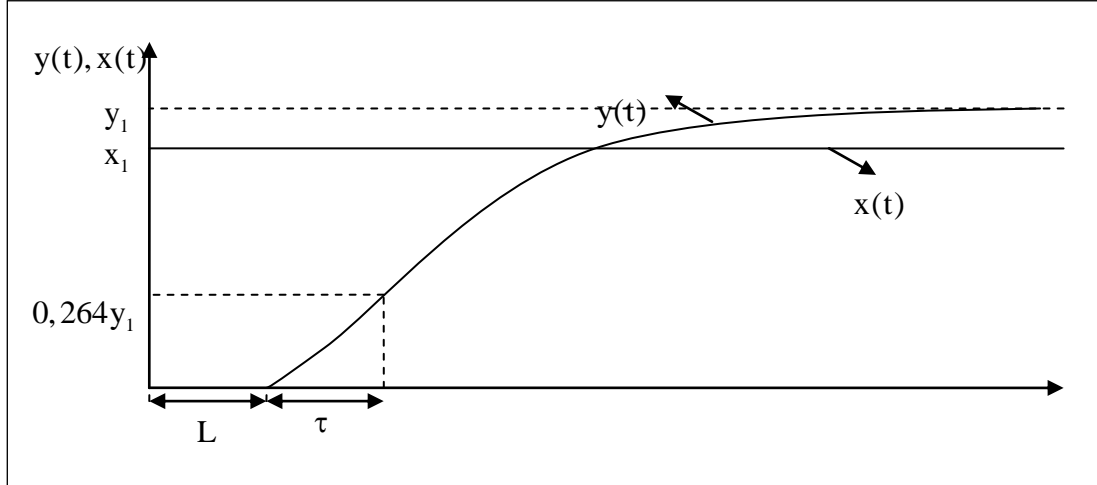


Şekil 3.5. İkinci mertebeden kritik sönümlü sistemin basamak girişe yanıtı



İkinci mertebeden az sönümlü sistemde ölü zaman bulunması durumu için s-tanım bölgesi transfer fonksiyonu Denklem (3.67) ifadesinde belirtilmiştir. Ölü zamanlı sistemin birim basamak yanıtı Şekil 3.6'da gösterilmiştir. L, ölü zaman değerini ifade etmektedir.

$$G(s) = \frac{K}{(\tau s + 1)^2} e^{-sL} \quad (3.67)$$



Şekil 3.6. İkinci mertebeden kritik zamanlı ölü zamanlı sistemin basamak girişe yanıtı

Kritik sönümlü ölü zamanlı sistemde  $\tau$  zaman sabitinin elde edilmesi için ise Denklem (3.68) kullanılır. Buna göre  $\tau$  zaman sabiti, çıkış işaretinin  $0,264y_1$  değerine ulaştığı süreden L ölü zaman süresinin çıkarılması ile elde edilir [6].

$$y(t) = y(L + \tau) = 0,264y_1 \quad (3.68)$$

İkinci mertebeden kritik sönümlü bir sistemin PID katsayıları aşağıdaki adımlar uygulanarak bulunabilir.

İkinci mertebeden kritik sönümlü bir sistem için önceki konularda anlatılan sistemlerde olduğu gibi öncelikle örnekleme zamanı T seçilir.

Örnekleme zamanı T, Denklem (3.38) ifadesinde  $w_n$  doğal frekans,  $\tau$  zaman sabiti cinsinden yazılırsa,

$$T = \frac{\pi}{10\omega_n} = \frac{\pi}{10} \tau \cong 0,314\tau \quad (3.69)$$

olarak elde edilir.

İkinci mertebeden kritik sönümlü bir sistemin ayrık zaman modeline ilişkin sıfırıncı mertebeden tutucu ile elde edilen z-tanım bölgesi transfer fonksiyonu Denklem (3.4) eşitliği kullanılarak,

$$G(z) = Z\{G_{zOH}(s)G(s)\} = Z\left\{\frac{1-e^{-sT}}{s} K \frac{1}{(\tau s + 1)^2}\right\} = K(1-z^{-1})Z\left\{\frac{1}{s(\tau s + 1)^2}\right\} \quad (3.70)$$

ifadesinden hareketle,

$$G(z) = K \frac{(1-A-AT/\tau)z + A^2 - A + TA/\tau}{z^2 - 2zA + A^2} \quad (3.71)$$

biçiminde elde edilir. Bu ifadede,

$$A = e^{-\frac{T}{\tau}} \quad (3.72)$$

olarak tanımlanan sabittir.

Denklem (3.71) yeniden düzenlenirse,

$$G(z) = K(1-A-AT/\tau) \frac{z + \frac{A^2 - A + TA/\tau}{(1-A-AT/\tau)}}{(z^2 - 2zA + A^2)} \quad (3.73)$$

olarak yazılır ve,

$$b_0 = \frac{A^2 - A + TA/\tau}{(1-A-AT/\tau)} \quad (3.74)$$

$$K_z = K(1-A-AT/\tau) \quad (3.75)$$

$$a_1 = -2A \quad (3.76)$$

$$a_0 = A^2 \quad (3.77)$$

tanımları yapılırsa Denklem (3.51) ifadesinde olduğu gibi benzer şekilde,

$$G(z) = \frac{K_z(z + b_0)}{z^2 - a_1z + a_0} \quad (3.78)$$

olarak elde edilir. Sistem ölü zamanlı ise Denklem (3.23) ve (3.24) ifadeleri kullanılarak ölü zamanlı sistemin z-tanım bölgesi transfer fonksiyonu

$$G(z) = \frac{K_z(z + b_0)}{z^2 - a_1z + a_0} z^{-d} \quad (3.79)$$

biçiminde elde edilir. Denklem (3.69) , (3.70) ve (3.71) ifadelerinden yararlanılarak PID katsayıları,

$$T_d = \frac{A^2}{1 - A^2} T \quad (3.80)$$

$$T_I = \frac{1 + A}{1 - A} T \quad (3.81)$$

$$K_C = \frac{K^* T}{K_z(T + T_d)} \quad (3.82)$$

ifadeleri kullanılarak ile elde edilir [6,14].

İkinci mertebeden kritik sönümlü ölü zamanlı bir sistemin fark denklemi, (3.62) ifadesindeki fark denkleminin Denklem (3.74), (3.75), (3.76) ve (3.77) ifadelerindeki sırasıyla  $b_0$ ,  $K_z$ ,  $a_1$  ve  $a_0$ , katsayılarına göre düzenlenmesiyle elde edilir. Burada da  $d=0$  yapılırsa ölü zamansız sistem için geçerli olan fark denklemi bulunur.

#### **3.4. Yüksek Mertebeden Sistemlerin Düşük Mertebeden Sistemlerle Yaklaşık İfadesi: Biçimsel Yaklaşım**

Yüksek mertebeden bir sistem verildiğinde, analiz ve sentez işlemlerini basitleştirmek için, sistem genellikle mümkün olduğu sürece daha düşük mertebeden yaklaşık sistemlerle ifade edilmek istenir. Bu kısımda, yüksek mertebeden bir

$M_H(s)$  transfer fonksiyonu verildiğinde düşük mertebeden bir  $M_L(s)$  transfer fonksiyonunun yaklaşık ifade edilmesiyle ilgili bir yöntem verilecektir.

Yüksek mertebeden sistem,  $n \geq m$  olmak üzere,

$$M_H(s) = K \frac{1 + b_1s + b_2s^2 + \dots + b_ms^m}{1 + a_1s + a_2s^2 + \dots + a_ns^n} \quad (3.83)$$

transfer fonksiyonu ile ifade edilmiş olsun.  $n \geq p \geq q$  için düşük merteben yaklaşık sistem transfer fonksiyonu,

$$M_L(s) = K \frac{1 + c_1s + c_2s^2 + \dots + c_qs^q}{1 + d_1s + d_2s^2 + \dots + d_ps^p} \quad (3.84)$$

şeklinde tanımlayalım. Sıfır frekansı için ( $s=0$ ) her iki transfer fonksiyonuna ilişkin  $K$  kazancının aynı olduğunu belirtelim. Buna göre yüksek mertebeden sistemin kararlı hal davranışı düşük merteben sistemde aynen korunmuş olur. Ayrıca, kararsız sistemlerle ilgilenilmediği için,  $M_H(s)$  ve  $M_L(s)$  transfer fonksiyon kutuplarının sol yarı  $s$ - düzleminde bulunduğunu varsayalım. Genelde  $M_H(s)$  ve  $M_L(s)$  transfer fonksiyonları kapalı çevrimli sistemlere ilişkindir, ancak gerektiğinde açık çevrim transfer fonksiyonlarını da ifade edebilir.

$M_H(s)$  verildiğinde daha düşük mertebeden  $M_L(s)$ ' nin belirlenmesine ilişkin kriter aşağıdaki ilişkinin mümkün olduğu kadar gerçekleşmesine bağlıdır.

$$0 \leq \omega \leq \infty \text{ için, } \frac{|M_H(j\omega)|^2}{|M_L(j\omega)|^2} = 1 \quad (3.85)$$

bu koşul, her iki sistemin frekans tanım bölgesinde ( $s = j\omega$ ) genlik karakteristiğinin benzer olması anlamına gelir. Bununla her iki sistemde benzer zaman yanıtları elde edilebileceği umulur.

Yaklaşıklık yöntemi aşağıdaki adımlardan oluşur:

1.  $M_L(s)$  için uygun bir  $q$  pay ve  $p$  payda mertebesi seçilir.

2.  $i=1,2,\dots,q$  için  $c_i$  ve  $j=1,2,\dots,p$  için  $d_j$  katsayıları (3.85) koşulu gerçekleşecek şekilde seçilir.

(3.83) ve (3.84) ilişkilerinden yararlanılarak  $M_H(s)$  'nin  $M_L(s)$  'ye oranı,  $u=m+p$  ve  $v=n+q$  olmak üzere,

$$\frac{M_H(s)}{M_L(s)} = \frac{(1 + b_1s^2 + \dots + b_ms^m)(1 + d_1s^2 + \dots + d_ps^p)}{(1 + a_1s^2 + \dots + a_ns^n)(1 + c_1s^2 + \dots + c_qs^q)} = \frac{1 + m_1s + m_2s^2 + \dots + m_us^u}{1 + l_1s + l_2s^2 + \dots + l_vs^v} \quad (3.86)$$

şeklinde elde edilir. Ayrıca (3.83) ilişkisi,

$$\frac{|M_H(jw)|^2}{|M_L(jw)|^2} = \frac{M_H(s)M_H(-s)}{M_L(s)M_L(-s)} \Big|_{s=jw} \quad (3.87)$$

şeklinde yazılabilir, burada  $M_H(s)M_H(-s)$  ve  $M_L(s)M_L(-s)$  ifadeleri  $s$  'ye göre çift polinomlardır (diğer bir deyişle  $s$  sadece çift üslerden oluşur). Buna göre Denklem (3.85) ifadesi,

$$\frac{|M_H(jw)|^2}{|M_L(jw)|^2} = \frac{1 + e_2s^2 + e_4s^4 + \dots + e_{2u}s^{2u}}{1 + f_2s^2 + f_4s^4 + \dots + f_{2v}s^{2v}} \Big|_{s=jw} \quad (3.88)$$

şeklinde yazılabilir. Denklem (3.88)'de sağ tarafta pay paydaya bir kez bölünürse,

$$\frac{|M_H(jw)|^2}{|M_L(jw)|^2} = 1 + \frac{(e_2 - f_2)s^2 + (e_4 - f_4)s^4 + \dots}{1 + f_2s^2 + f_4s^4 + \dots + f_{2v}s^{2v}} \Big|_{s=jw} \quad (3.89)$$

elde edilir.  $u=v$  için, Denklem (3.89) ifadesinde son terim  $(e_{2u} - f_{2u})s^{2u}$  olur. Ancak uygulamada genellikle beklendiği gibi  $u < v$  ise, Denklem (3.88) ilişkisinin paydasında,  $(e_{2u} - f_{2u})s^{2u}$  terimleri ötesinde ayrıca,

$$-f_{2(u+1)}s^{2(u+1)} - \dots - f_{2v}s^{2v} \quad (3.90)$$

terimleri de yer alır.

Denklem (3.85) koşulunu sağlayan bir ifade Denklem (3.89) ilişkisinden,  $u=v$  için,

$$\begin{aligned}
e_2 &= f_2 \\
e_4 &= f_4 \\
e_6 &= f_6 \\
&\cdot \\
&\cdot \\
&\cdot \\
e_{2u} &= f_{2u}
\end{aligned} \tag{3.91}$$

olarak elde edilir.  $u < v$  için türetilen daha düşük mertebeden modelin hatası,

$$|\varepsilon| = \frac{|M_H(j\omega)|^2}{|M_L(j\omega)|^2} - 1 = \frac{-f_{2(u+1)}s^{2(u+1)} - \dots - f_{2v}s^{2v}}{1 + f_2s^2 + f_4s^4 + \dots + f_{2v}s^{2v}} \Big|_{s=j\omega} \tag{3.92}$$

şeklindedir.

$M_H(s)$  verildiğinde  $M_L(s)$ ' ye ilişkin bilinmeyen katsayıların belirlenmesinde Denklem (3.91) koşullarından yararlanır. Bunun için,

$$\frac{M_H(s)M_H(-s)}{M_L(s)M_L(-s)} = \frac{(1+m_1s+m_2s^2+\dots+m_us^u)[1-m_1s+m_2s^2-\dots+(-1)^u m_us^u]}{(1+l_1s+l_2s^2+\dots+l_vs^v)[1-l_1s+l_2s^2-\dots+(-1)^v l_vs^v]} = \frac{1+e_2s^2+e_4s^4+\dots+e_{2u}s^{2u}}{1+f_2s^2+f_4s^4+\dots+f_{2v}s^{2v}} \tag{3.93}$$

ifadesi yazılır. Denklem (3.93) ifadesinde her iki taraf eşitlenirse  $e_2, e_4, \dots, e_{2u}$  terimleri  $m_1, m_2, \dots, m_u$  cinsinden, benzer şekilde  $f_2, f_4, \dots, f_{2v}$  terimleri  $l_1, l_2, \dots, l_v$  cinsinden elde edilir.

Örnek olarak  $u=8$  için (3.93) ilişkisinden,

$$\begin{aligned}
e_2 &= 2m_2 - m_1^2 \\
e_4 &= 2m_4 - 2m_1m_3 + m_2^2 \\
e_6 &= 2m_6 - 2m_1m_5 + 2m_2m_4 - m_3^2 \\
e_8 &= 2m_8 - 2m_1m_7 + 2m_2m_6 - 2m_3m_5 + m_4^2 \\
e_{10} &= 2m_2m_8 - 2m_3m_7 + 2m_4m_6 - m_5^2 \\
e_{12} &= 2m_4m_8 - 2m_3m_7 + m_6^2 \\
e_{14} &= 2m_6m_8 - m_7^2 \\
e_{16} &= m_8^2
\end{aligned} \tag{3.94}$$

elde edilir. Genelleştirilirse,  $m_0 = 1$  ve  $x=1, 2, \dots, u$  için,

$$e_{2x} = \sum_{i=0}^{x-1} (-1)^i 2m_i m_{2x-i} + (-1)^x m_x^2 \quad (3.95)$$

ve benzer şekilde,  $l_0 = 1$  ve  $y=1, 2, \dots, v$  için,

$$f_{2y} = \sum_{i=0}^{y-1} (-1)^i 2l_i l_{2y-i} + (-1)^y m_y^2 \quad (3.96)$$

Yazılarak biçimsel yaklaşım yöntemi tamamlanmış olur [5]. Verilen biçimsel yaklaşım yöntemi denklemleri Bölüm-5'te üçüncü mertebeden örnek sistemin ikinci mertebeden sistem cinsinden yaklaşık ifadesinde kullanılacaktır

#### **4. PLC İLE DİJİTAL PID KONTROLÖR PROGRAMLAMA**

Programlanabilir Lojik Kontrolörler probleme bağlı olmaksızın üretilmiş kumanda ve kontrol elemanlarıdır. Bütün kumanda problemlerinin çözümünde mantık işlemleri, bellek fonksiyonları, zaman ve sayıcı gibi elemanlara ihtiyaç vardır. Bunlar PLC'lerde üretici firmalar tarafından hazır olarak sunulmuş durumdadır. Basit bir programlama ile bütün bu imkanlar problemin çözümünde bir araya getirilebilirler [8]. Genel olarak PLC'ler şu özelliklere sahiptir:

- PLC ile yapılan çözümde kumanda devresi yazılımla sağlandığından daha kolay ve güvenilirlerdir,
- Daha az yer tutar ve daha az arıza yaparlar,
- Yeni bir uygulamaya daha çabuk adapte olurlar,
- Kötü çevre şartlarından kolay kolay etkilenmezler,
- Daha az kablo bağlantısı isterler,
- Hazır fonksiyonları kullanma imkanı vardır,
- Giriş ve çıkışların durumları izlenebilir

Bir PLC ile kontrol sistemlerinin oluşturulması:

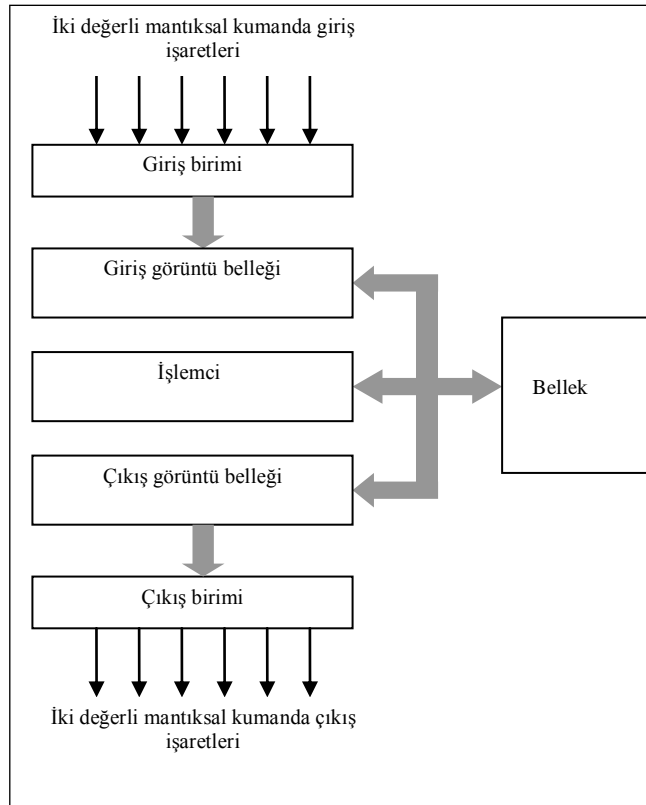
1. Kontrol probleminin ifade edilmesiyle sorunun kağıda dökülmesi ve senaryonun yazılması,
2. Sorunun çözümü için gerekli program veya fonksiyonların belirlenmesi,
3. Programın diyagrama aktarılması (LADDER STL, SCL, FBD)
4. Programın yazılması



olarak sıralanabilir. En yaygın programlama dili olarak merdiven (LADDER) kullanılır. Fakat kompleks uygulamalarda ve yoğun matematiksel ve sisteme ilişkin blok yazılımı gerektiren programlarda STL daha ön plana çıkmaktadır [15].

#### 4.1. PLC' nin Genel Yapısı

Bir PLC en genel anlamda işlevsel üç temel birimden oluşur: Yazılımın saklandığı bellek, bu yazılım içindeki komutları çalıştırıp giriş-çıkış birimleri ile haberleşen CPU ve PLC 'nin çevresel cihazlar ile iletişimini sağlayan giriş-çıkış birimleri. Giriş birimi çeşitli elektriksel özelliklerdeki kumanda ya da geri besleme işaretlerinin PLC'de işlenecek sayısal büyüklüklere dönüştürülmesini, çıkış birimi ise PLC'de işlenen verilere ilişkin sayısal değerlerin kontrol edilen sisteme uygun elektriksel işaretlere dönüştürülmesini sağlayan birimdir. İşlemci birimi, PLC belleğindeki sistem programına göre çalışmayı düzenleyen ve kullanıcı programını yürüten en önemli birimdir. PLC'ye ilişkin genel yapı Şekil 4.1'deki gibi gösterilebilir.



Şekil 4.1. PLC genel yapısı

Bellek, işletim sistemi programının bulunduğu sistem program belleği, kullanıcı programının bulunduğu program belleği ve veri belleği gibi bölümlerden oluşur.

Sistem belleđi ve PLC'ye iliřkin deđiřtirilmeyen veriler iin salt okunur kalıcı bellek; program belleđi ve veri belleđi iin rastgele eriřimli bellek kullanır. Veri belleđi giriř-ıkıř iřaret durumları, sayıcı ve zamanlayıcı ierikleri, özel amalı kaydedici ierikleri, analog iřaretlere iliřkin sayısal deđerlerin tutulduđu eřitli blmlerden oluřur.

Giriř grnt belleđi, programın yrtlmesi srecinde, giriř birimindeki iki deđerli iřaretlerin mantıksal durumlarının (0-1) saklandıđı bellek alanı; ıkıř grnt belleđi ise kullanıcı programının yrtlmesi srecinde, ıkıř noktalarına iliřkin hesaplanan mantıksal deđerlerin saklandıđı bellek alanıdır. Siemens tarafından retilen PLC'lerde giriř grnt belleđi iin I0.0, I0.1,I0.2... gibi, ıkıř grnt belleđi iin ise Q0.0, Q0.1, Q0.2... gibi adresler kullanılır. PLC'lerde programda hesaplanan ara deđerlerin saklandıđı "merker", "flag", "internal output", "auxiliary relay" gibi isimler verilen bellek alanları da kullanılır. Bu adres alanları iin F,M,V gibi harflerle bařlayan F0.0, F0.1...;M0.0,M0.1...;V0.0,V0.1... gibi adresler kullanılır. Bu tr verilere genellikle 1 bit, 8 bit, 16 bit ya da 32 bit olarak eriřilebilir [3,6].

#### **4.2. SIMATIC S7-300/400 Tipi PLC'lere Genel Bakıř**

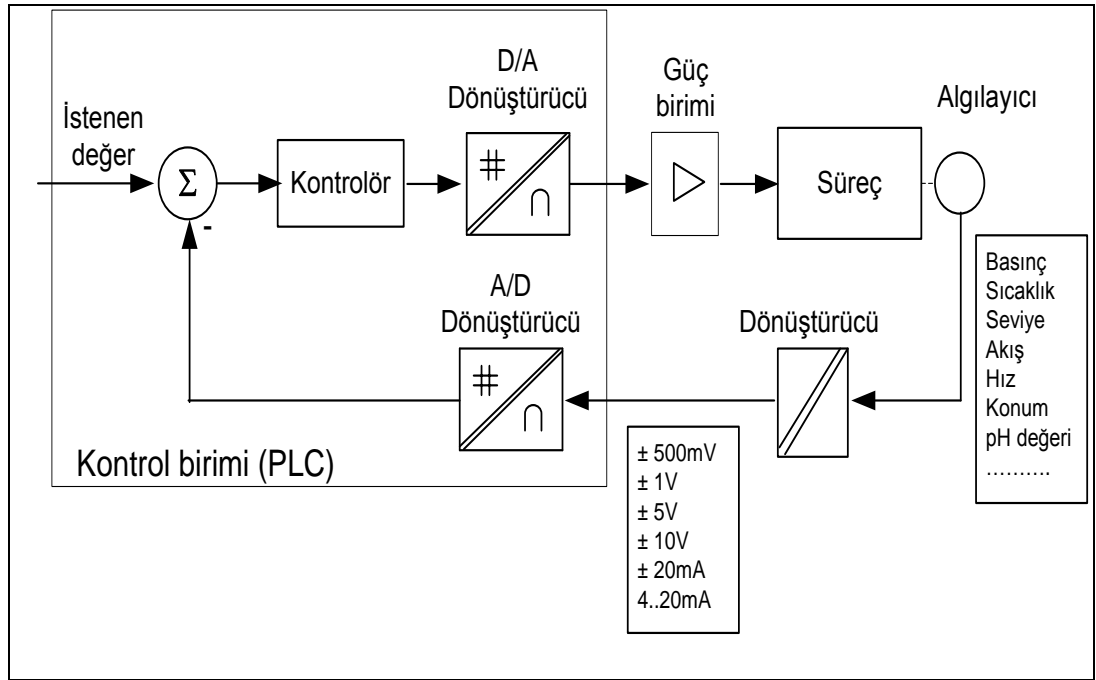
SIMATIC S7-300/400 tipi PLC'ler genel olarak ařađıdaki zelliklere sahiptir [13,14],

- Dřk ve orta performanslı prosesler iin geliřtirilmiřtir,
- Farklı otomasyon problemlerine cevap verebilecek nitelikte zengin rn eřitliliđine sahiptir,
- Proseste gerekebilecek bir geliřtirmeye olanak sađlar,
- MPI, PROFIBUS ve Endstriyel Ethernet gibi haberleřme ađlarına bađlanabilme olanađına sahiptir,
- SCL, GRAPH gibi st dzey programlama teknikleriyle programlayabilme olanađı sađlar.

SIMATIC S7-300/400 ailesindeki iřlemcilerin donanım ve haberleřme ayarlarının yapılması ve programlanması SIMATIC MANAGER yazılımıyla sađlanır.

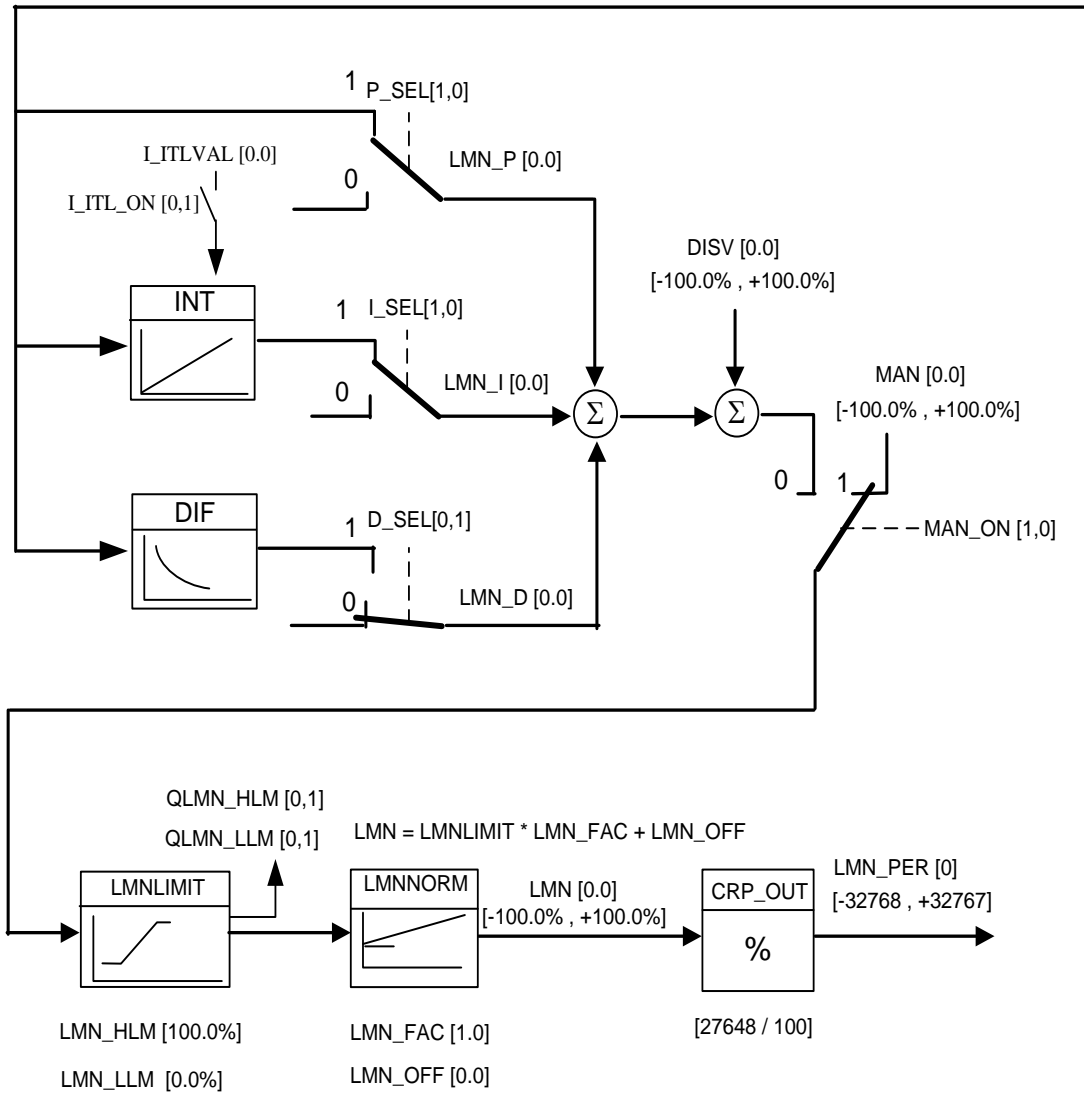
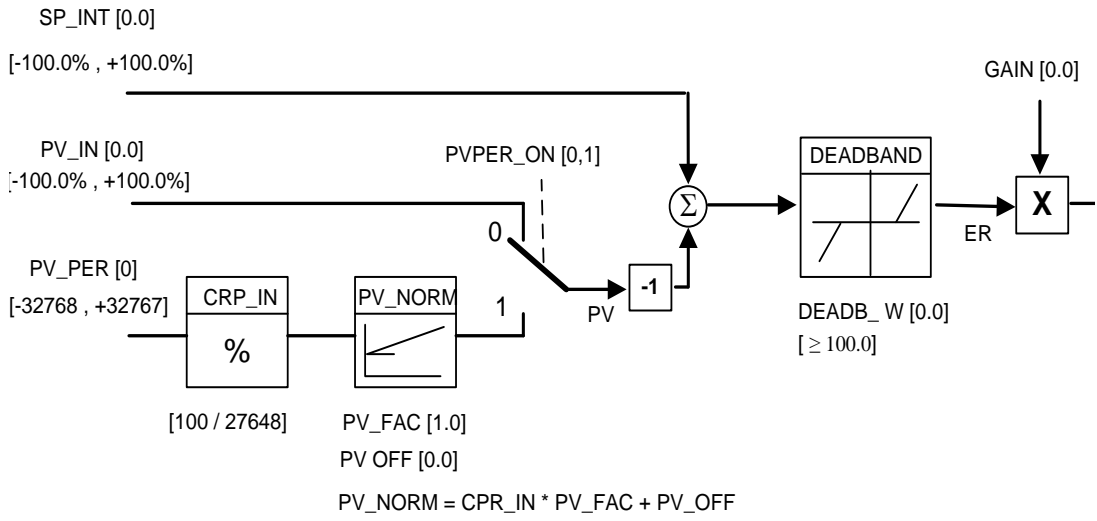
### 4.3. SIMATIC S7-300/400 İle PID Kontrol

Günümüz endüstriyel kumanda sistemlerinin gerçekleşmesinde yaygın olarak kullanılan çoğu PLC, dijital kontrol sistemlerinin gerçekleşmesi için gerekli donanım ve yazılım özelliklerini de taşır. PLC'nin kontrol birimi olarak kullanıldığı bir kontrol sistemi Şekil 4.2'deki gibi gösterilebilir. Analog giriş-çıkış birimleri ile donatılmış ve kontrol kurallarının yazılımı için gerekli matematiksel işlem komutları olan herhangi bir PLC ile dijital kontrol sistemleri başarıyla gerçekleştirilebilir. Günümüzde çoğu PLC, kontrol sistemlerinin gerçekleşmesinde kullanılan hazır fonksiyonları da içerir. Örneğin, birçok PLC'de endüstriyel uygulamalarda yaygın olarak kullanılan PID kontrol işlevini sağlayan komut ya da fonksiyonlar bulunur. Kontrol sistemlerinin PLC ile gerçekleşmesine ilişkin bu kısımda bu tür komut ya da fonksiyonların nasıl kullanıldığı açıklanacaktır [4].

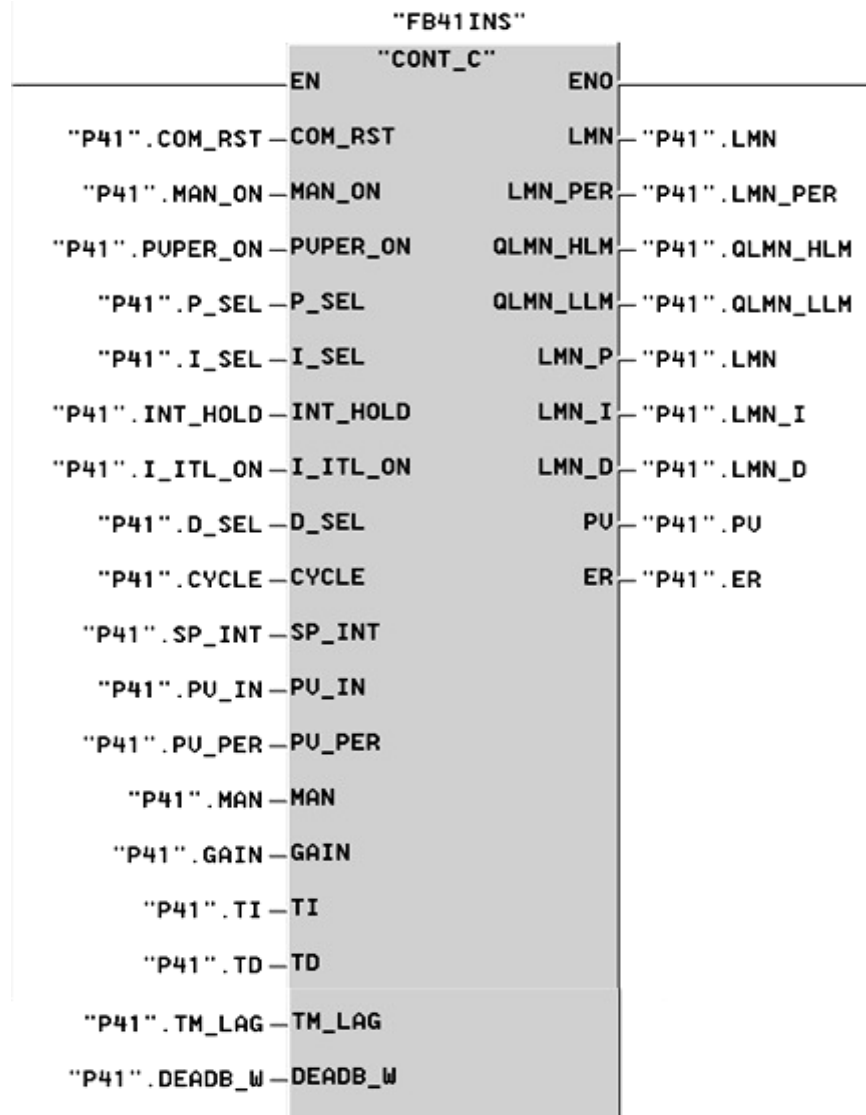


Şekil 4.2. PLC'nin kontrol birimi olarak kullanıldığı bir kontrol sistemi

S7-300/400 tipi PLC uygulamalarında PID işlevini yerine getiren birçok fonksiyon blok bulunur. Şekil 4.3'de yapılan tez çalışmasında kullanılacak FB41 PID fonksiyon bloğuna ilişkin işlev şeması, Şekil 4.4'te programdaki görünümü verilmiştir [4].



Şekil 4.3. FB41 PID fonksiyon bloğuna ilişkin işlev şeması



Şekil 4.4. FB41 PID fonksiyon bloğunun programdaki görünümü

FB41 kontrol bloğunda birçok özellik bulunmakla beraber tasarlanan kontrolörün sisteme uygulanması için bu özelliklerin hepsine gerek yoktur. Kullanılması gereken temel özellikler aşağıdaki gibi verilebilir.

SP\_INT (Internal Setpoint): Bu büyüklük -100.0 ile 100.0 arasında değişen sistemin yüzde olarak karşılığıdır. Referans değeri buraya yüzde olarak girilir.

PUPER\_ON (Process variable peripherie on): Sistemden gelen geri besleme direkt olarak analog girişten alınacaksa bu bit aktive edilir.

P\_SEL (Proportional action on): Aktif olması durumunda kontrolörün kazanç ayağından gelen kontrol işaretinin tüm kontrol işaretine katılmasını sağlar.

I\_SEL (Integral action on): Aktif olması durumunda kontrolörün integral ayağından gelen kontrol işaretinin tüm kontrol işaretine katılmasını sağlar.

D\_SEL (Derivative action on): Aktif olması durumunda kontrolörün türev ayağından gelen kontrol işaretinin tüm kontrol işaretine katılmasını sağlar

MAN\_ON (Manual value on): Aktif olması durumunda önceden belirlenen bir kontrol işareti sisteme otomatik olarak verilir. Analog bir kontrol işlemi yapılması için bu bit aktif olmamalıdır.

COM\_RST (Complete restart): Aktif olması durumunda kontrolörün içindeki tüm değerler resetlenir.

CYCLE (Sample time): Kontrolörün örnekleme zamanı bu alana T#.MS şeklinde girilir.

PV\_PER (Process variable peripherie): Sistemden gelen geri besleme bilgisinin olduğu analog adres bu alana yazılır.

GAIN (Proportional gain): Kontrolör kazancı buraya reel sayı olarak ya da reel formatında bir bellek alanı olarak yazılır.

TI (Reset time): İntegral zaman sabiti buraya zaman formatında girilmelidir. Eğer hesaplanan değer direkt olarak gönderilmek isteniliyorsa katsayı milisaniyeye çevrilip tam sayı haline getirilerek buraya yazılmalıdır.

TD (Derivative time): Türev zaman sabiti buraya zaman formatında girilmelidir. Eğer hesaplanan değer direkt olarak gönderilmek isteniliyorsa katsayı önce milisaniyeye çevrilip tam sayı haline getirilerek buraya yazılmalıdır

LMN\_PER (Manipulated value peripherie): Kontrol işaretinin sisteme uygulanacağı analog adres bu alana yazılır.

PV (Process variable): Bu değer ile setpointe ne kadar yaklaşıldığı görülür.

ER (Error signal): SP ile PV arası hatanın % değeridir.

PID fonksiyon bloğunda girişlere bir atama yapılmadığı durumdaki varsayılan ilk değerler ile ilgili data blok tablosu ile Şekil 4.5'te verilmiştir.

	Address	Declaratio	Name	Type	Initial value	Comment
1	0.0	in	COM_RST	BOOL	FALSE	complete restart
2	0.1	in	MAN_ON	BOOL	TRUE	manual value on
3	0.2	in	PV_PER_ON	BOOL	FALSE	process variable peripherie on
4	0.3	in	P_SEL	BOOL	TRUE	proportional action on
5	0.4	in	I_SEL	BOOL	TRUE	integral action on
6	0.5	in	INT_HOLD	BOOL	FALSE	integral action hold
7	0.6	in	I_ITL_ON	BOOL	FALSE	initialization of the integral action
8	0.7	in	D_SEL	BOOL	FALSE	derivative action on
9	2.0	in	CYCLE	TIME	T#1S	sample time
10	6.0	in	SP_INT	REAL	0.000000e+000	internal setpoint
11	10.0	in	PV_IN	REAL	0.000000e+000	process variable in
12	14.0	in	PV_PER	WORD	W#16#0	process variable peripherie
13	16.0	in	MAN	REAL	0.000000e+000	manual value
14	20.0	in	GAIN	REAL	2.000000e+000	proportional gain
15	24.0	in	TI	TIME	T#20S	reset time
16	28.0	in	TD	TIME	T#10S	derivative time
17	32.0	in	TM_LAG	TIME	T#2S	time lag of the derivative action
18	36.0	in	DEADB_W	REAL	0.000000e+000	dead band width
19	40.0	in	LMN_HLM	REAL	1.000000e+002	manipulated value high limit
20	44.0	in	LMN_LLM	REAL	0.000000e+000	manipulated value low limit
21	48.0	in	PV_FAC	REAL	1.000000e+000	process variable factor
22	52.0	in	PV_OFF	REAL	0.000000e+000	process variable offset
23	56.0	in	LMN_FAC	REAL	1.000000e+000	manipulated value factor
24	60.0	in	LMN_OFF	REAL	0.000000e+000	manipulated value offset
25	64.0	in	I_ITLVAL	REAL	0.000000e+000	initialization value of the integral action
26	68.0	in	DISV	REAL	0.000000e+000	disturbance variable
27	72.0	out	LMN	REAL	0.000000e+000	manipulated value
28	76.0	out	LMN_PER	WORD	W#16#0	manipulated value peripherie
29	78.0	out	QLMN_HLM	BOOL	FALSE	high limit of manipulated value reached
30	78.1	out	QLMN_LLM	BOOL	FALSE	low limit of manipulated value reached
31	80.0	out	LMN_P	REAL	0.000000e+000	proportionality component
32	84.0	out	LMN_I	REAL	0.000000e+000	integral component
33	88.0	out	LMN_D	REAL	0.000000e+000	derivative component
34	92.0	out	PV	REAL	0.000000e+000	process variable
35	96.0	out	ER	REAL	0.000000e+000	error signal

Şekil 4.5. PID fonksiyon bloğu varsayılan değerleri

Sistemin belirlenmesi ve tasarlanan kontrolörün sisteme uygulanması için belirli bir örnekleme zamanında çalışan bir kesme alt programına ihtiyaç duyulur. Bu çalışmada kullanılan S7-300/400 tipi PLC'lerde bulunan OB35 bloğu verilen örnekleme zamanı için istenilen kesmeli çalışmayı sağlayacaktır.

Yukarıdaki verilen bilgiler ışığında S7-300/400 tipi PLC nin ve FB41 PID bloğunun endüstriyel kontrol için esnek ve yeterli bir yapıya sahip olduğu görülebilmektedir. Bu nedenle yapılan tez çalışmasında bahsedilen PLC tipi ve bloğunun kullanımına karar verilmiştir.

## 5. ÖRNEK SİSTEM İÇİN PLC'DE PID KONTROLÖR TASARIMI

Bu bölümde üçüncü bölümde anlatılan yüksek mertebeden sistemin düşük mertebeden ifadesi yöntemi kullanılarak ve Step-7 Simatic Manager programından yararlanılarak, ikinci bölümde örnek olarak kullanılan üçüncü mertebeden sistem için PID kontrolör tasarımı ele alınacaktır. Gerçek PLC yerine S7-PLCSIM Simulasyon programı kullanılacak, gerçek sistem yerine ise Simatic Manager programında benzetim fonksiyonu yazılacaktır.

### 5.1. Üçüncü Mertebeden Sistemin İkinci Mertebeden Sistem Cinsinden İfadesi

Bu kısımda ikinci bölümde örnek olarak kullanılan Denklem (2.10) ile belirtilen sistemin ikinci mertebeden sistemle yaklaşık ifadesi elde edilecektir.

Denklem (2.10)'daki sistemi Denklem (3.83) için yeniden düzenlersek,

$$G(s) = \frac{1}{s^3 + 3s^2 + 3s + 1} = M_H(s) = K \frac{1 + b_1s + b_2s^2 + \dots + b_ms^m}{1 + a_1s + a_2s^2 + \dots + a_ns^n} = \frac{1}{1 + 3s + 3s^2 + s^3} \quad (5.1)$$

olarak elde edilir. İkinci mertebeden elde edilecek sistemin transfer fonksiyonu ise,

$$M_L(s) = K \frac{1 + c_1s + c_2s^2 + \dots + c_qs^q}{1 + d_1s + d_2s^2 + \dots + d_ps^p} = \frac{1}{1 + d_1s + d_2s^2} \quad (5.2)$$

olarak alınabilir. Denklem (3.86) ilişkisinden,

$$\frac{M_H(s)}{M_L(s)} = \frac{\frac{1}{1 + 3s + 3s^2 + s^3}}{\frac{1}{1 + d_1s + d_2s^2}} = \frac{1 + d_1s + d_2s^2}{1 + 3s + 3s^2 + s^3} = \frac{1 + m_1s + m_2s^2}{1 + l_1s + l_2s^2 + l_3s^3} \quad (5.3)$$

yazılabilir. Buna göre,

$$l_1 = 3, \quad l_2 = 3, \quad l_3 = 1, \quad d_1 = m_1, \quad d_2 = m_2 \quad (5.4)$$

olmalıdır. Denklem (3.93) ilişkisinden yararlanılırsa,



$$\frac{M_H(s)M_H(-s)}{M_L(s)M_L(-s)} = \frac{1 + e_2s^2 + e_4s^4}{1 + f_2s^2 + f_4s^4 + f_6s^6} \quad (5.5)$$

yazılabilir. Denklem (3.91) ve (3.94) ilişkilerinden,

$$\begin{aligned} e_2 = f_2 = 2m_2 - m_1^2 = 2d_2 - d_1^2 \\ e_4 = f_4 = 2m_4 - 2m_1m_3 + m_2^2 = d_2^2 \end{aligned} \quad (5.6)$$

ve benzer biçimde Denklem (3.96) ilişkisinden,

$$\begin{aligned} f_2 = 2l_2 - l_1^2 \\ f_4 = 2l_4 - 2l_1l_3 + l_2^2 \\ f_6 = 2l_6 - 2l_1l_5 + 2l_2l_4 - l_3^2 \end{aligned} \quad (5.7)$$

Yazılabilir. Buna göre,

$$f_2 = -3, \quad f_4 = 3, \quad f_6 = -3 \quad (5.8)$$

olarak hesaplanır. Denklemler çözüldürse,

$$d_2^2 = f_4 = 3$$

$$d_2 = \pm\sqrt{3}$$

$M_L(s)$  'in kararlı olabilmesi için pozitif değerler alınırsa,

$$d_2 = 1,732 \quad (5.9)$$

$$d_1^2 = 2d_2 - f_2 = 6,464$$

$$d_1 = \pm\sqrt{6,464}$$

Aynı şekilde  $M_L(s)$  'in kararlı olabilmesi için pozitif değerler alınırsa,

$$d_1 = 2,542 \quad (5.10)$$

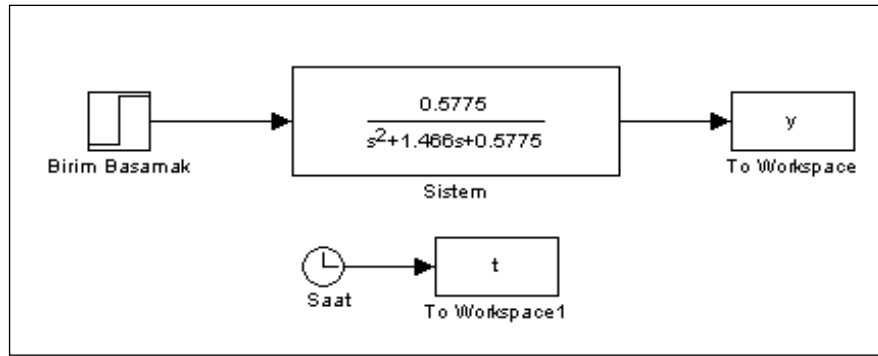
olarak hesaplanır. Buna göre ikinci mertebeden transfer fonksiyonu,

$$M_L(s) = \frac{1}{1 + 2,542s + 1,732s^2} = \frac{0,5775}{s^2 + 1,466s + 0,5775} \quad (5.11)$$

olarak elde edilir ve böylece üçüncü mertebeden sistem, ikinci mertebeden bir sistemle ifade edilmiş olur. Sonraki kısımda, elde edilen ikinci mertebeden sistemin türü belirlenerek, PID katsayılarını kendisi hesaplayan bir benzetim fonksiyonu oluşturulacaktır. Bu fonksiyon için gereken parametrelerin elde edilmesinde Matlab Simulink programından yararlanılacaktır.

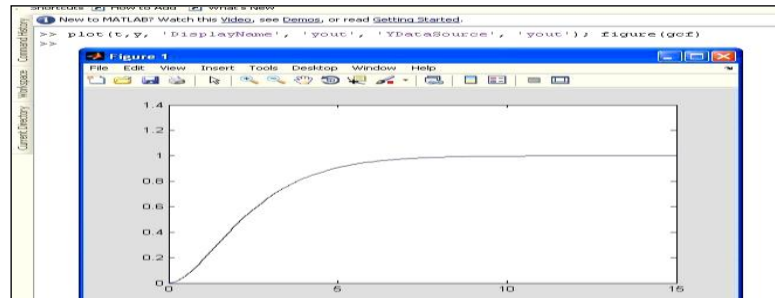
## 5.2. Sistem Türünün Belirlenmesi

Elde edilen ikinci mertebeden sistemin Matlab devre şeması Şekil 5.1’de gösterilmiştir.



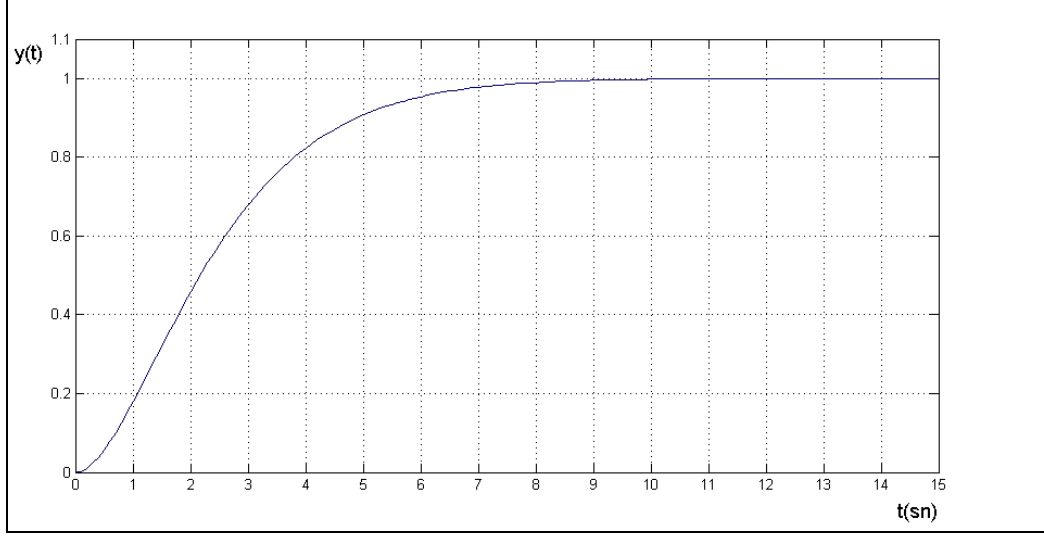
Şekil 5.1. İkinci mertebeden sistemin MATLAB devre şeması

MATLAB komut satırına `>> plot(t,y, 'DisplayName', 'yout', 'YDataSource', 'yout');` `figure(gcf)` komutu yazıldığında sistemin birim basamak yanıtı Şekil 5.2’deki gibi elde edilir.



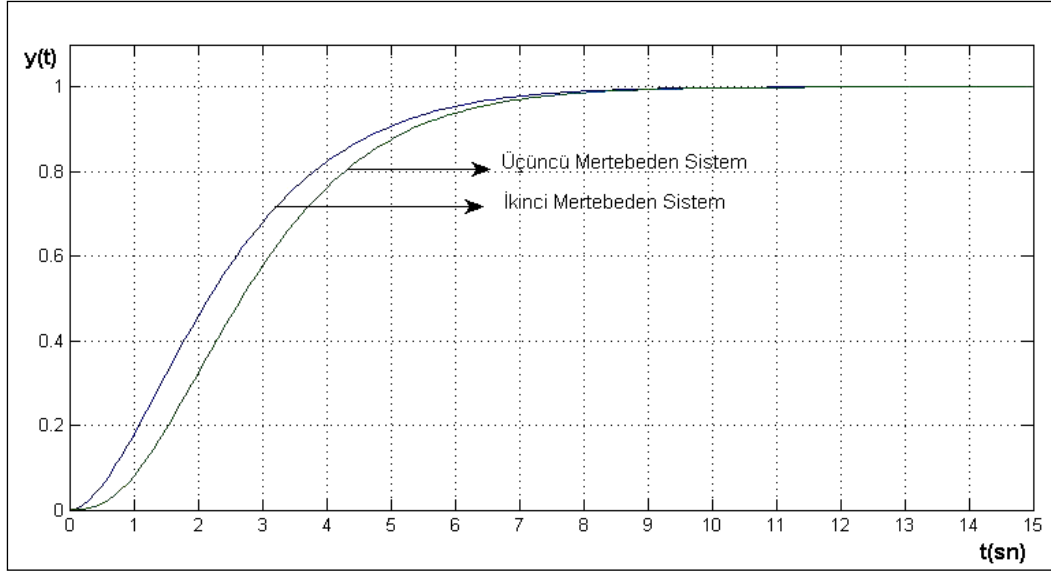
Şekil 5.2. İkinci dereceden sistemin birim basamak yanıtı

Grafiğin düzenlenmiş hali şekil 5.3’te gösterilmiştir.



Şekil 5.3. Birim basamak yanıtının düzenlenmiş hali

İkinci mertebeden elde edilen sistem ile üçüncü mertebeden sistemin Bölüm-3'te elde edilen birim basamak yanıtlarının karşılaştırılması Şekil 5.4'te gösterilmiştir.



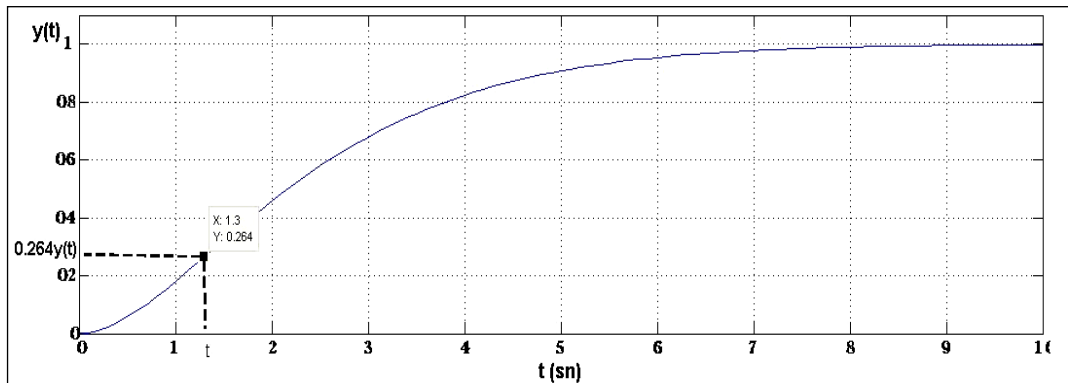
Şekil 5.4. Elde edilen sistemin ve üçüncü mertebeden sistemin karşılaştırılması

Sistemin birim basamak yanıtının elde edilmesinden sonra ilk olarak sistemin türünün belirlenmesi gereklidir. Bunun için Bölüm-3'te anlatılan sistemlerle elde ettiğimiz birim basamak yanıtını kıyaslırsak, elde edilen birim basamak yanıtı Şekil 5.3'teki gibi olduğundan sistemin kritik sönümlü ölü zamansız olduğu söylenebilir.

### 5.3. Sayısal Benzetim Parametreleri, Fark Denklemi ve PID Katsayılarının Elde Edilmesi

Sayısal benzetim, bir sistemin dinamik davranışına ilişkin sayısal verilerin üretilmesi olarak tanımlanabilir. Bir sisteme ilişkin sayısal benzetimden yararlanılarak, bir kontrol yönteminin amaca uygunluğu ya da sistem davranışı üzerindeki etkisi yazılım aşamasında değerlendirilebilir. Bir sistemin dinamik davranışını tanımlayan matematiksel ifadelerin sayısal çözümünde dijital kontrol sistemlerinin modellenmesinde kullanılan z-tanım bölgesi transfer fonksiyonlarından elde edilen fark denklemleri kullanılır [6]. Bu kısımda sayısal benzetim için gerekli parametreler ve fark denklemi elde edilecektir. Ayrıca PID katsayıları da bir sonraki konuda oluşturulacak olan PLC programının sağlamasını yapmak üzere bu kısımda elde edilecektir.

Denklem (3.66) ifadesinden zaman sabiti  $\tau = 1,3$  olarak elde edilir. İlgili grafik Şekil 5.5'te gösterilmiştir.



Şekil 5.5. İkinci mertebeden elde edilen sistemin zaman sabitinin elde edilmesi

Kazanç değeri K,

$$K = \frac{y_2}{y_1} = 1 \quad (5.12)$$

Örnekleme zamanı T,

$$T = 0,314\tau = 0,408 \quad (5.13)$$

olarak elde edilir. Sistemin z-tanım bölgesi transfer fonksiyonundaki katsayılar ise,

$$A = e^{-\frac{T}{\tau}} = 0,7306 \quad (5.14)$$

$$b_0 = \frac{A^2 - A + TA/\tau}{(1 - A - AT/\tau)} = 0,811 \quad (5.15)$$

$$K_z = K(1 - A - AT/\tau) = 0,04 \quad (5.16)$$

$$a_1 = -2A = -1,461 \quad (5.17)$$

$$a_0 = A^2 = 0,533 \quad (5.18)$$

olarak hesaplanır. Buna göre z-tanım bölgesi transfer fonksiyonu,

$$G(z) = \frac{K_z(z + b_0)}{z^2 + a_1z + a_0} = \frac{0,04(z + 0,811)}{z^2 - 1,461z + 0,533} \quad (5.19)$$

olarak elde edilir. PID katsayıları ise,

$$T_d = \frac{A^2}{1 - A^2} T = 0,467 \quad (5.20)$$

$$T_I = \frac{1 + A}{1 - A} T = 2,621 \quad (5.21)$$

$$K_C = \frac{K^* T}{K_z(T + T_d)} = 2,305 \quad (5.22)$$

olarak hesaplanır. Sistemin fark denklemi, hesaplanan değerler Denklem (3.62) ifadesinde yerine konulduğunda,

$$y(k) = -1,461y(k-1) - 0,533y(k-2) + 0,04[x(k-1) + 0,811x(k-2)] \quad (5.23)$$

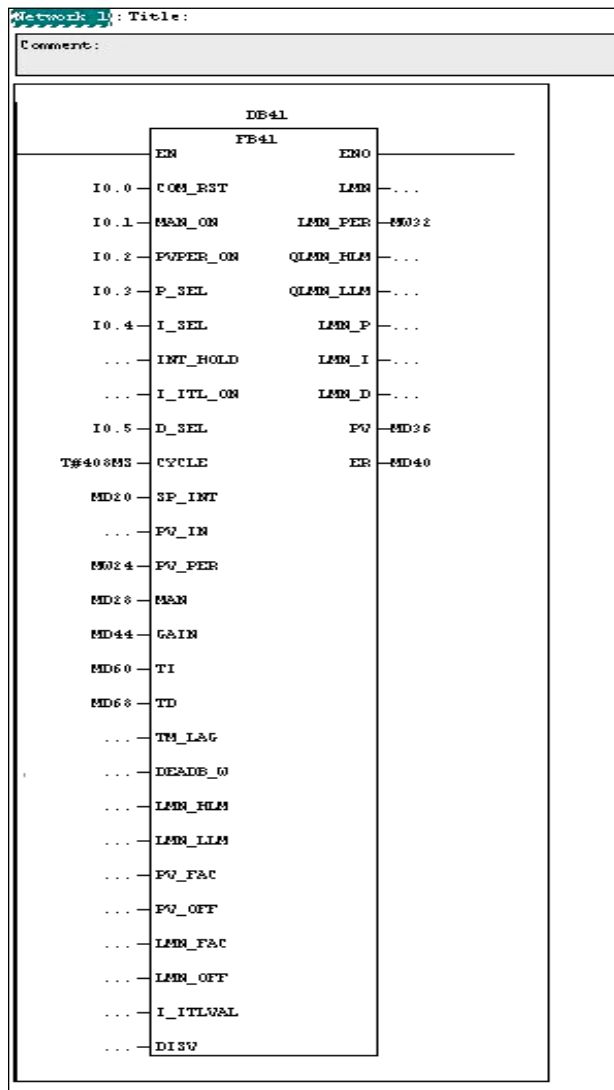
ya da,

$$y(k) = -1,461y(k-1) - 0,533y(k-2) + 0,04x(k-1) + 0,032x(k-2) \quad (5.24)$$

şeklinde elde edilir. Artık bu fark denklemi programlanarak benzetim fonksiyonu elde edilebilir.

## 5.4. PLC Programının Oluşturulması

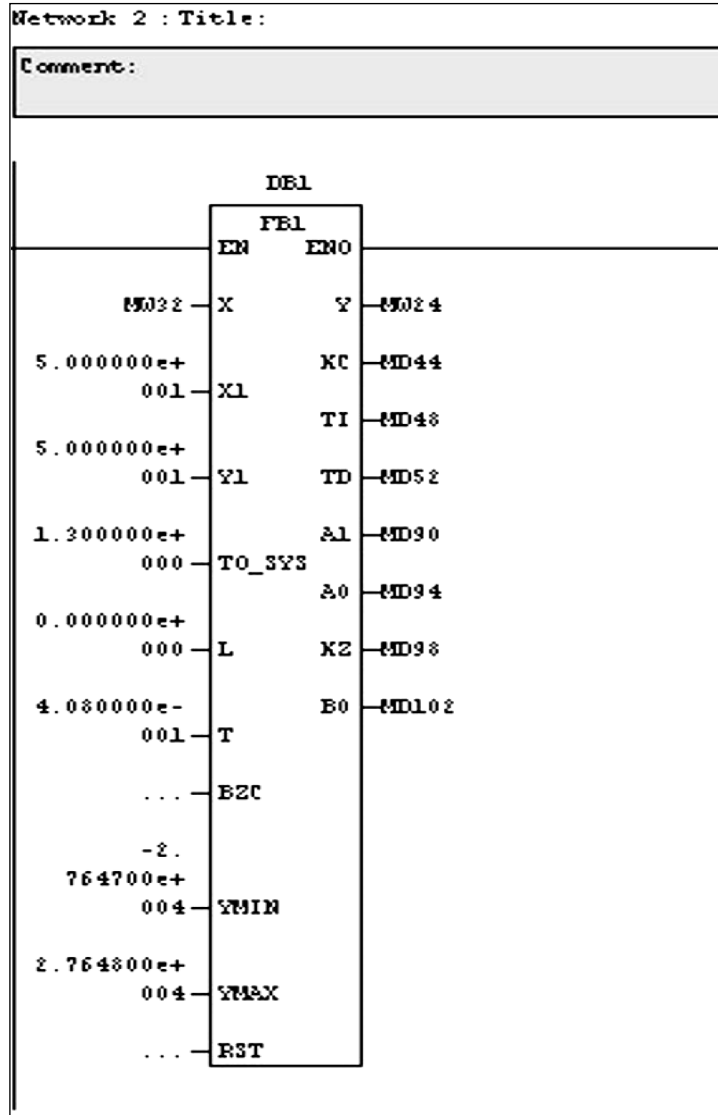
PID kontrolörünün PLC’de programlanması için belirli bir örnekleme zamanında çalışan bir kesme alt programına ihtiyaç duyulur. OB35 bloğu verilen örnekleme zamanı için istenilen kesmeli çalışmayı sağlayacaktır. Sistem türü belirlendikten sonra bulunan katsayıların bir PID bloğuna yüklenerek sisteme uygulanması gerekmektedir. Bu amaçla “Simatic Manager” programının standart kütüphanesinde bulunan ve Bölüm-4’te de anlatılan “FB41 Continuous Control” bloğu kullanılacaktır. Bu bloğun uygulamadaki görüntüsü Şekil 5.6’da gösterilmiştir.



Şekil 5.6. PID kontrolör fonksiyon bloğu FB41

PID kontrolör bloğundan sonra sistemin benzetim fonksiyonu ve PID katsayılarını hesaplamak için FB1 bloğu oluşturulmuştur. Oluşturulan FB1 bloğunun

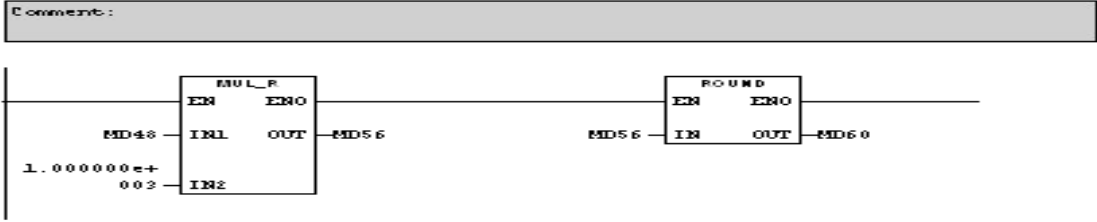
uygulamadaki görüntüsü Şekil 5.7’de gösterilmiştir. Bu bloğun SCL dilinde yazılmış hali EK-A’da verilmiştir.



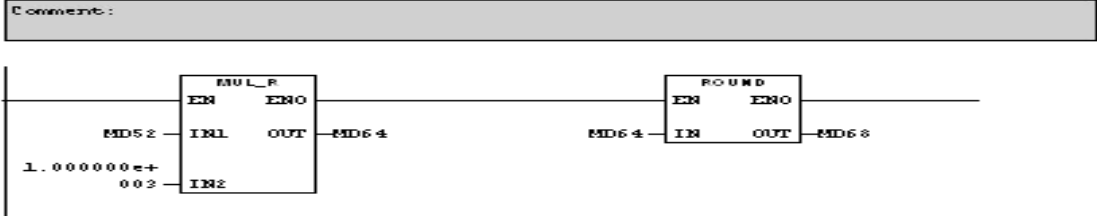
Şekil 5.7. Sistemin benzetim fonksiyon bloğu FB1

Hesaplanan  $T_1$  ve  $T_D$  parametrelerinin doğrudan FB41 fonksiyon bloğunda yerine yazılması için gerekli olan program parçasının uygulamadaki görüntüsü Şekil 5.8’de gösterilmiştir.

Network 3 : T<sub>I</sub> katsayısını fb41 e yazdırma



Network 4 : T<sub>D</sub> katsayısını fb41 e yazdırma



Şekil 5.8. T<sub>I</sub> ve T<sub>D</sub> parametrelerinin doğrudan FB41 fonksiyon bloğunda yerine yazılması için gerekli olan program parçası

## 5.5. Programın Uygulanması ve Elde Edilen Sonuçlar

PID kontrolör programı yazıldıktan sonra simülasyon programına yüklenir ve simülasyondaki CPU RUN konumuna alınır. Benzetim fonksiyonunu içeren ve PID katsayılarını hesaplayan FB1 bloğunun online görüntüsü Şekil 5.9'da gösterilmiştir. Buna göre Denklem (5.15), (5.16), (5.17), (5.18), (5.19), (5.20), (5.21) ve (5.22) ifadelerinde hesaplanan katsayıların programda hesaplananlarla aynı olduğu görülebilir.

Network 2 : Title:

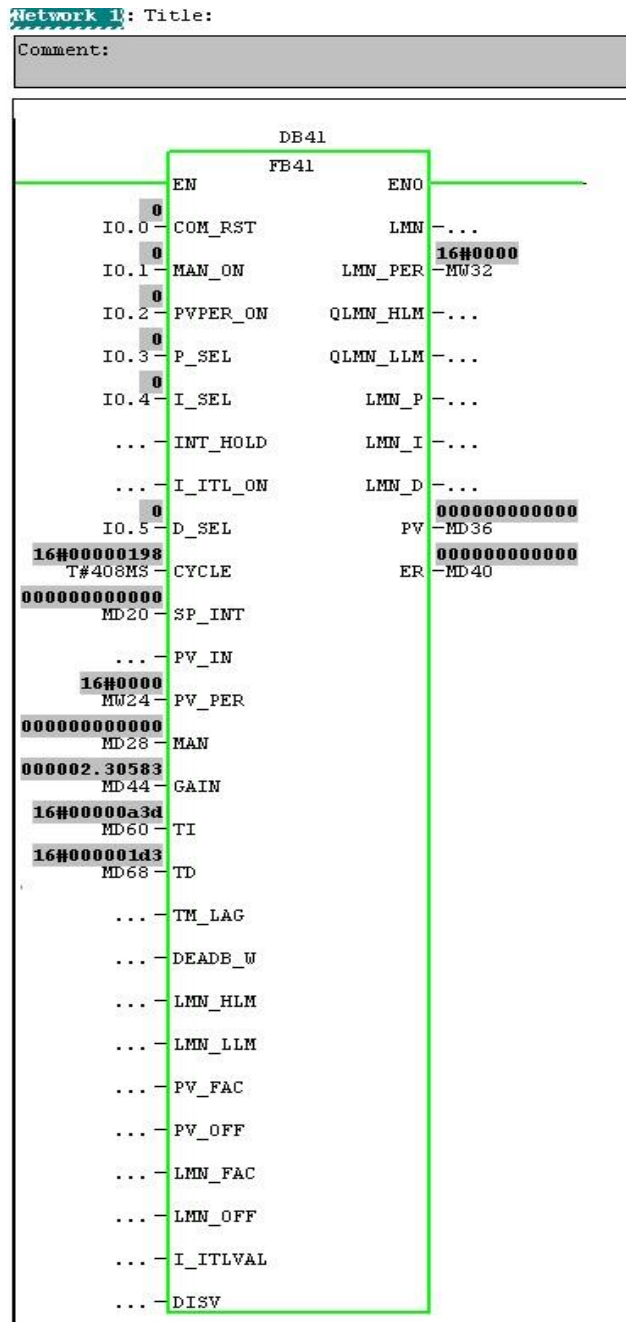
Comment:

DB1	
EN	ENO
MW32 0 X	Y MW24 0
5.000000e+001 X1	KC 000002.30583
5.000000e+001 Y1	TI MD44 000002.62131
1.300000e+000 TO_SYS	TD MD48 00000.467203
0.000000e+000 L	TD MD52 -00001.46126
4.080000e-001 T	A1 MD100 00000.533822
...	AO MD104 0000.0400627
-2.764700e+004 YMIN	KZ MD108 00000.811146
2.764800e+004 YMAX	B0 MD112
...	BZC
...	RST

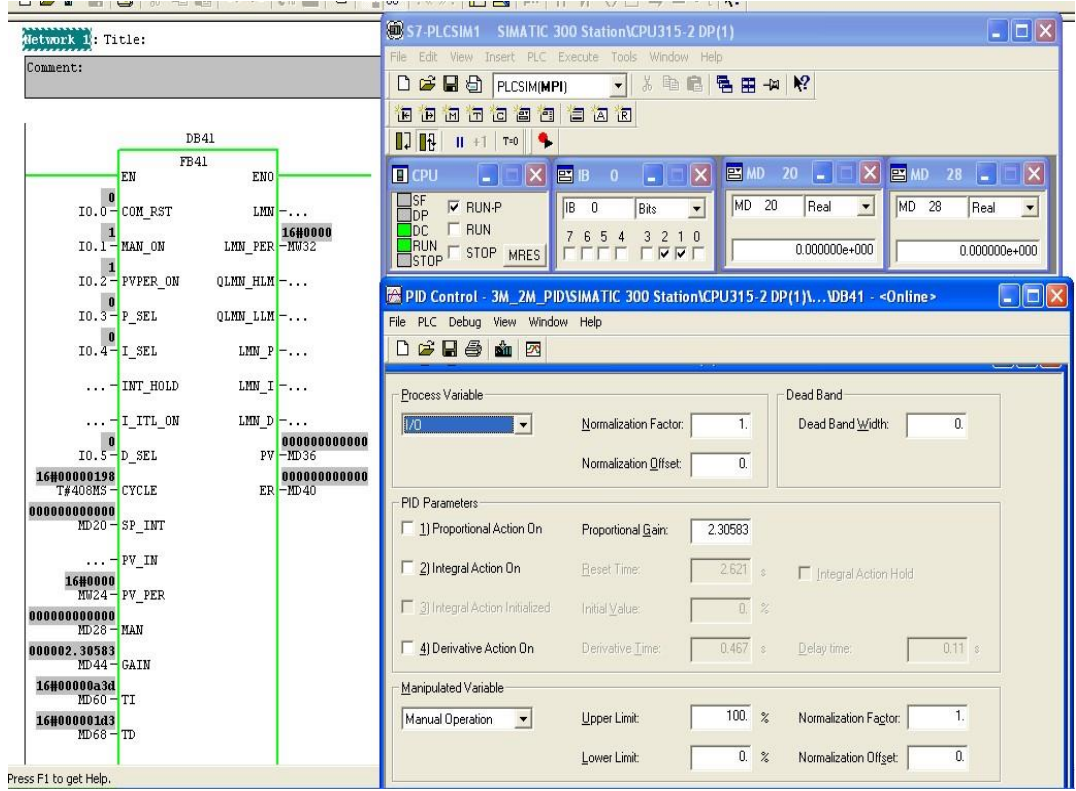
Şekil 5.9. PID benzetim fonksiyonunun online görüntüsü



FB41 bloğunun online görüntüsü Şekil 5.10’da gösterilmiştir. Bloğun devreye alınabilmesi için PVPER\_ON girişi, yani simülasyon programında I0.2 biti aktif yapılmalıdır. Sistemin birim basamak yanıtını gözlemleyebilmek için ise MAN\_ON girişi yani I0.1 biti aktif yapılmalıdır. Birim basamak yanıtını izlemek için PID Control Parameter Assignment programı kullanılacaktır. Son durumdaki FB41 bloğunun Simatic Manager ve PID Control programındaki online görüntüsü ile simülasyon programının online görüntüsü Şekil 5.11’de gösterilmiştir.

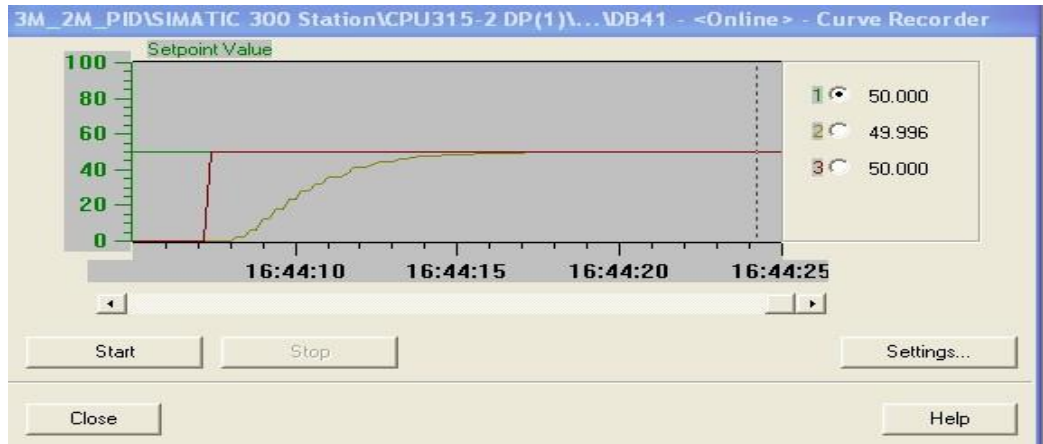


Şekil 5.10. FB41 bloğunun online görüntüsü



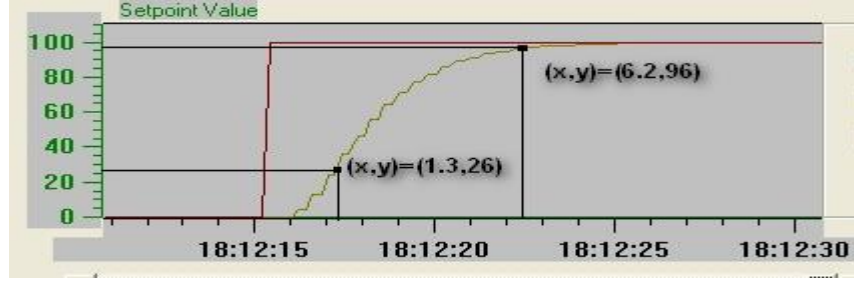
Şekil 5.11. PID kontrol ve simülasyon programlarının online görüntüsü

MAN\_ON girişi aktif yapıldıktan sonra birim basamak yanıtının izlenmesi için PID programında GRAFİK butonuna ve açılan sayfada START butonuna basılır. Daha sonra FB41 bloğundaki MAN girişine yani bu giriş için tanımlanan MD28 adresine istenen setpoint değeri girilir. Simülasyon programında setpoint değeri olarak 50.0 değeri girilmiştir. PID programında oluşan görüntü birim basamak yanıtıdır. Bu görüntü Şekil 5.12’de gösterilmiştir.

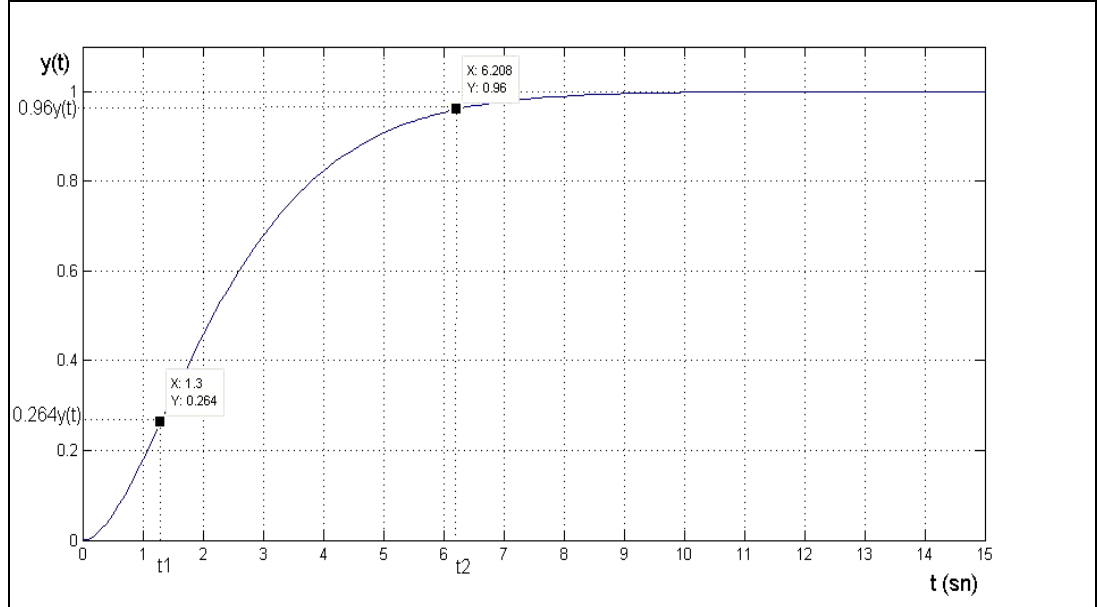


Şekil 5.12. Sistemin birim basamak yanıtının PID programındaki görüntüsü

Matlab programıyla ve benzetim fonksiyonuyla elde edilen birim basamak yanıtlarının yaklaşık olarak aynı olduğu Şekil 5.13'deki karşılaştırmalarla anlaşılabilir. Burada karşılaştırmayı daha iyi görebilmek için setpoint olarak 100.0 değeri girilmiştir. Benzetim fonksiyonuyla elde edilen sonuçta bir ölü zaman süresinin oluştuğu görülebilir.



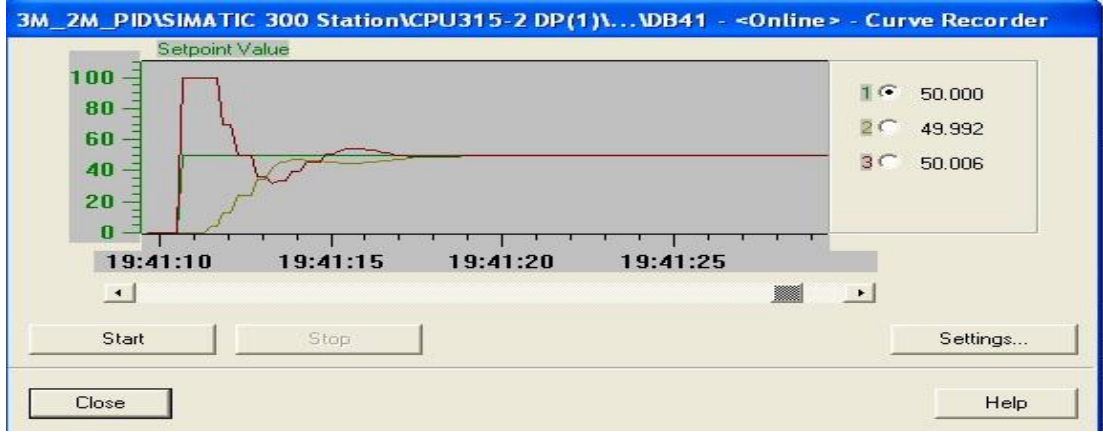
(a)



(b)

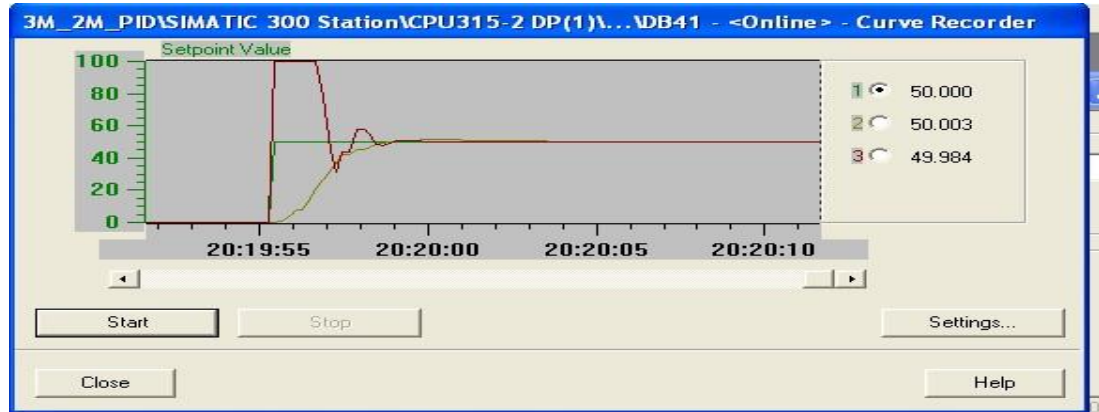
Şekil 5.13. Elde edilen birim basamak yanıtlarının karşılaştırılması a) Benzetim fonksiyonu birim basamak yanıtı b) MATLAB birim basamak yanıtı

Sistemin birim basamak yanıtı da gözlemlendikten sonra PID kontrolörün devreye alınması işlemine geçilebilir. Bunun için öncelikle MAN\_ON girişi yani I0.1 biti pasif yapılmalıdır. Daha sonra sisteme PID kontrol uygulanacağı için P\_SEL , I\_SEL ve D\_SEL girişleri yani sırasıyla I0.3 , I0.4 ve I0.5 bitleri aktif yapılmalıdır. Bu bitler aktif yapıldıktan sonra PV\_PER girişine yani MD20 adresine SETPOINT değeri girilir. Girilen değer yine 50.0 olacaktır. PID kontrol uygulanmış haliyle PID kontrol programındaki görüntü Şekil 5.14'te gösterilmiştir.



Şekil 5.14. Sisteme PID kontrol uygulanmış haliyle PID kontrol programındaki görüntü

Şekil 5.14'e bakarak uygulanan PID kontrolörün cevabında dalgalanmalar olduğu görülebilir. Bunun nedeninin örnekleme zamanının seçimiyle ilgili olup olmadığını anlayabilmek için yeni bir örnekleme zamanı alınarak bir uygulama daha yapılacaktır. Örnekleme zamanı olarak 100ms değerini alıp Hardware ile FB41 ve FB1 bloklarındaki yerlerine yazdığımızda çıkan sonuç Şekil 5.15'te gösterilmiştir. Yeni durumda yükselme zamanı ve oturma zamanı kısalırken aşım oluştuğu görülmektedir.

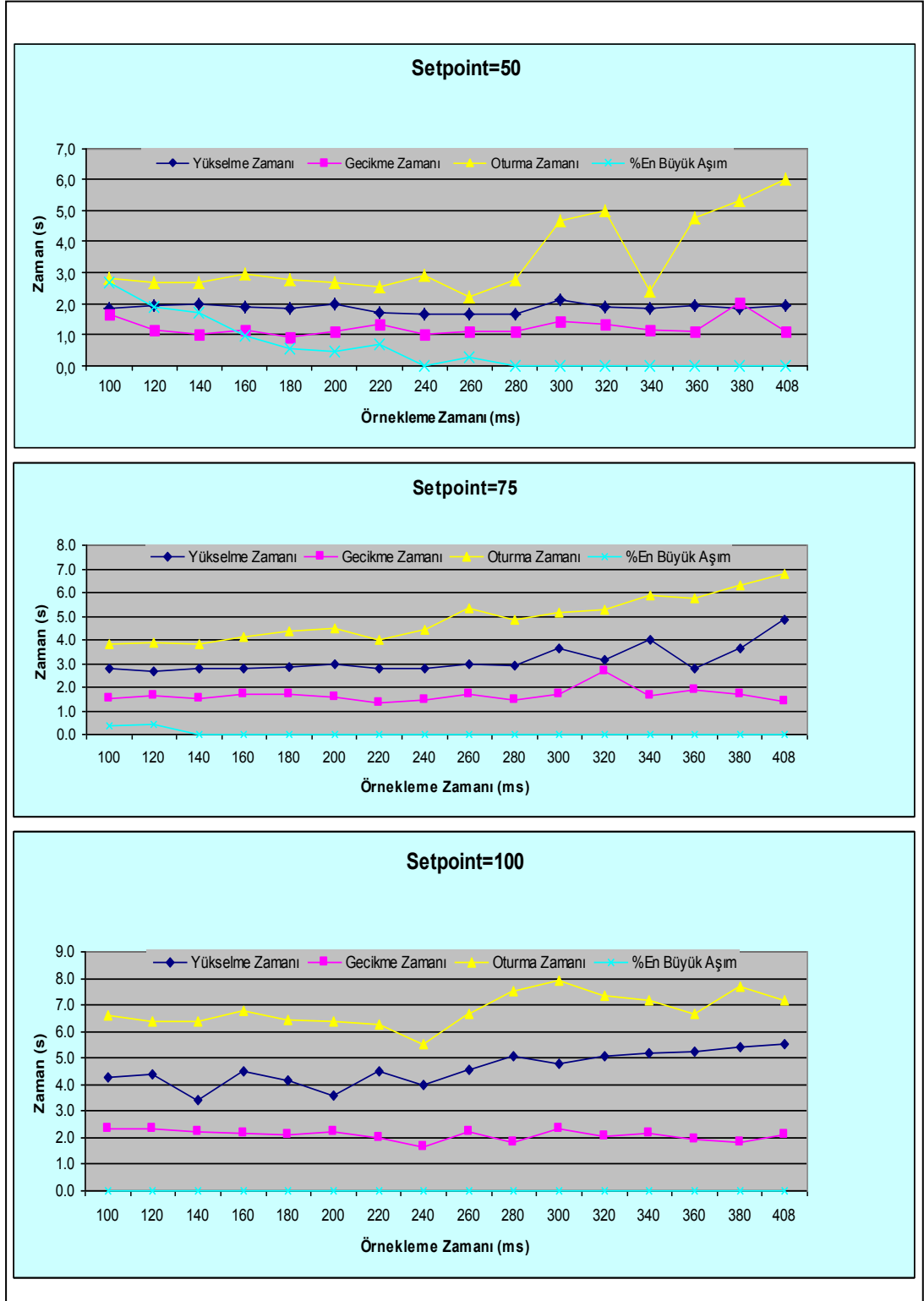


Şekil 5.15. Örnekleme zamanı 100ms olduğunda PID programındaki görüntüsü

Farklı örnekleme zamanları ve setpoint değerlerine göre elde edilen zaman tanım bölgesi değerleri ve PID parametreleri Tablo 5.1'de gösterilmiştir. Örnekleme zamanına göre bu değerlerin değişimi Şekil 5.16'da gösterilmiştir.

Tablo 5.1. Farklı örnekleme zamanları ve setpoint değerlerine göre elde edilen zaman tanım bölgesi değerleri ve PID parametreleri

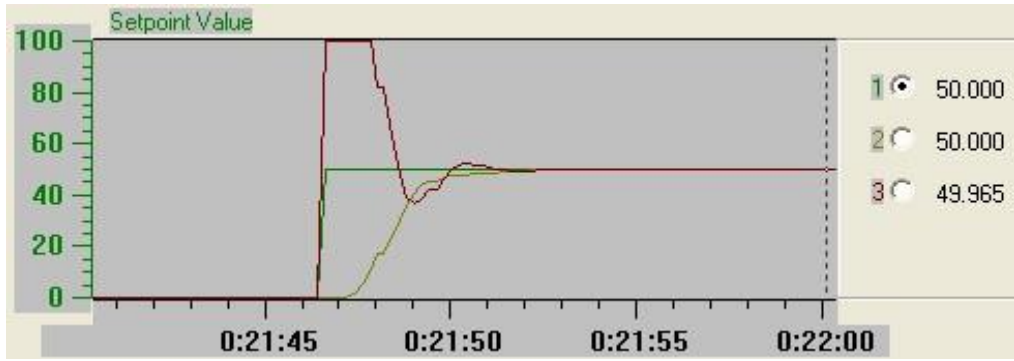
Örnekleme Zamanı (ms)	Setpoint	Yükselme Zamanı	Gecikme Zamanı	Oturma Zamanı	%Aşım	P	T <sub>I</sub>	T <sub>D</sub>
100	50	1,8	1,7	2,8	2,7	9,022	2,601	0,601
	75	2,8	1,5	3,8	0,4			
	100	4,3	2,3	6,6	0			
120	50	1,9	1,2	2,7	1,9	7,536	2,601	0,591
	75	2,7	1,6	3,9	0,4			
	100	4,4	2,3	6,4	0			
140	50	2,0	1,0	2,7	1,7	6,475	2,602	0,582
	75	2,8	1,5	3,8	0,1			
	100	3,4	2,2	6,4	0			
160	50	1,9	1,2	2,9	1,0	5,580	2,603	0,573
	75	2,8	1,7	4,1	0			
	100	4,5	2,2	6,8	0			
180	50	1,8	0,9	2,8	0	5,061	2,604	0,564
	75	2,8	1,7	4,4	0			
	100	4,2	2,1	6,4	0			
200	50	2,0	1,1	2,7	0,5	4,567	2,605	0,555
	75	2,9	1,6	4,5	0			
	100	3,6	2,2	6,4	0			
220	50	1,7	1,3	2,6	0,7	4,163	2,606	0,546
	75	2,8	1,3	4,0	0			
	100	4,5	2,0	6,3	0			
240	50	1,7	1,0	2,9	0	3,823	2,607	0,537
	75	2,8	1,4	4,4	0			
	100	4,0	1,7	5,5	0			
260	50	1,7	1,1	2,2	0,3	3,541	2,608	0,528
	75	3,0	1,7	5,3	0			
	100	4,6	2,2	6,7	0			
280	50	1,7	1,1	2,8	0	3,297	2,610	0,520
	75	2,9	1,4	4,8	0			
	100	5,1	1,8	7,5	0			
300	50	2,1	1,4	4,7	0	3,086	2,611	0,511
	75	3,7	1,7	5,2	0			
	100	4,8	2,3	7,9	0			
320	50	1,9	1,3	5,0	0	2,901	2,613	0,503
	75	3,2	2,7	5,3	0			
	100	5,1	2,1	7,3	0			
340	50	1,8	1,2	2,4	0	2,739	2,614	0,494
	75	4,0	1,6	5,9	0			
	100	5,2	2,2	7,2	0			
360	50	1,9	1,1	4,8	0	2,594	2,616	0,486
	75	2,8	1,9	5,8	0			
	100	5,2	2,0	6,7	0			
380	50	1,8	2,1	5,3	0	2,465	2,618	0,478
	75	3,6	1,7	6,3	0			
	100	5,4	1,8	7,7	0			
408	50	1,9	1,1	6,0	0	2,305	2,621	0,467
	75	4,8	1,4	6,8	0			
	100	5,6	2,1	7,2	0			



Şekil 5.16. Örnekleme zamanına göre zaman tanım bölgesi değerlerine ait grafikler

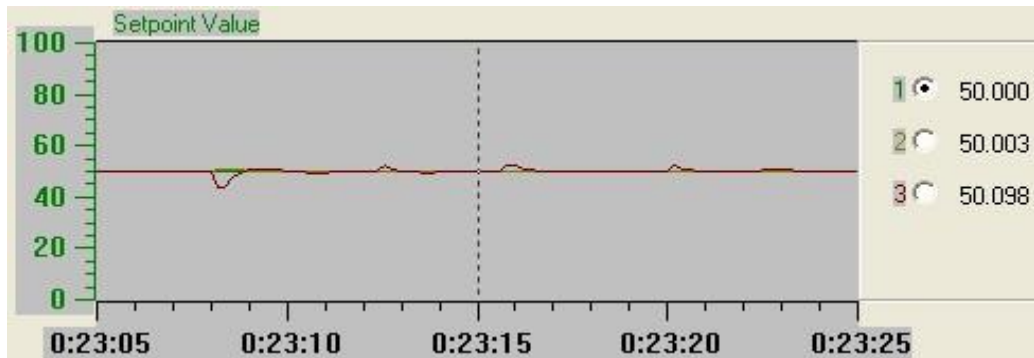
Elde edilen sonuçlardan, yükselme, gecikme ve oturma zamanlarının en düşük, aşımın en az olduğu örnekleme zamanı değerinin 240ms-280ms aralığında olduğu görülebilir. Ancak örnekleme zamanı ne kadar yüksek olursa verilerin alınma

sıklığının süresi de uzayacağından örnekleme zamanı olarak mümkün olan en düşük değerin seçilmesi gerekliliği unutulmamalıdır. EK-B’de verilen, örnekleme zamanlarına göre elde edilen simülasyon grafiklerinden örnekleme zamanı arttıkça ölü zaman değerinin arttığı ve eğrinin dalgalandığı görülebilir. Tüm hususlar göz önünde bulundurulduğunda ölü zaman artışı göz ardı edildiğinde örnekleme zamanı olarak 240ms seçilmesinin en iyi sonucu sağlayacağı söylenebilir. Örnekleme zamanı olarak 240ms seçilmesi durumu için elde edilen simülasyon görüntüsü Şekil 5.17’de gösterilmiştir.



Şekil 5.17. 240ms örnekleme zamanı için elde edilen simülasyon görüntüsü

Sistem üzerinde bozucun etkisini görebilmek için sisteme bozucu işareti uygulanır, kontrolörün meydana gelen değişimlere karşı gösterdiği tepki ile sistem çıkış işareti incelenir. Sisteme bozucu uygulanması durumu için elde edilen simülasyon görüntüsü şekil 5.18’de gösterilmiştir.



Şekil 5.18. Sisteme bozucu uygulanması durumu için simülasyon görüntüsü

## 6. SONUÇLAR VE ÖNERİLER

Bu çalışmada üçüncü mertebeden teorik bir sistem, verilen yöntemle ikinci mertebeden bir sistem cinsinden yaklaşık ifade edilmiş, ifade edilen yeni sistemin MATLAB programında birim basamak yanıtı parametreleri ve sistem türü belirlenerek, bunların zaman tanım bölgesi denklemlerinde kullanılmasıyla fark denklemi elde edilmiştir. SIMATIC MANAGER programında SCL programlama dilinde bu fark denklemi kullanılarak sistemin sayısal benzetimini ve PID katsayı hesaplarını yapan bir blok oluşturulmuştur. PID kontrolör bloğu olarak SIMATIC MANAGER kütüphanesinde hazır olarak bulunan FB41 bloğu kullanılarak gerekli adreslemeler yapılmış ve program yazılımı tamamlanmıştır. Program PLC-SIM programında çalıştırıldığında sistem ölü zamansız olduğu halde bir ölü zaman oluştuğu gözlenmiştir. Ölü zaman oluşumunun örnekleme zamanıyla ilişkisini ortaya koymak için farklı örnekleme zamanlarında simülasyonlar yapılmış ve örnekleme zamanı süresi düşürüldüğünde sistem yanıt süresinin kısaldığı, örnekleme zaman süresi arttırıldığında ise yanıt süresinin geciktiği tespit edilmiştir. Zaman tanım bölgesi kriterlerine göre en iyi sonuç olarak 240ms örnekleme zamanı kullanılması önerilmiş, bu örnekleme zamanı kullanılması durumu için PLC-SIM programında simülasyon sonuçları verilmiştir. Daha sonra sisteme bozucu uygulanması durumu incelenmiş, simülasyon sonucunda bozucu uygulandığında bile girilen setpoint değerinin hızlı bir şekilde yakalandığı görülmüştür.

Yapılan çalışmanın S7-300/400 PLC tipleri için sayısal benzetim yöntemi ile PID kontrolör tasarımı yapılması sayesinde tasarım aşamalarının kolaylaştırılması konusunda yenilik getireceği, aynı zamanda yüksek mertebeden sistemle karşılaşılması durumunda dijital PID kontrolörün bu tip PLC'lerde nasıl tasarlanabileceği, örnekleme zamanının seçimi ve PID katsayılarının yazılımla otomatik hesaplanabilmesi konularında sonraki benzer çalışmalara ışık tutabilmesi düşünülmektedir.



Yapılan bu çalışma sayesinde S7-300/400 PLC tipleri için sayısal benzetim yöntemi kullanılarak PID kontrolör tasarımı yapılması gerçekleştirilmiştir. Böylece gerçek sisteme ihtiyaç olmadan PID katsayıları kolay ve güvenli bir şekilde belirlenebilmiştir. Bu da kontrol mühendisinin işini kolaylaştırarak hızlı tasarım yapmasına olanak sağlayacaktır. Ayrıca tasarımın başlangıcında gerçek sistemin kurulumuna gerek olmaması tasarım maliyetlerini önemli ölçüde düşürecektir.

Elde edilen sonuçların oluşturduğu temel doğrultusunda PID katsayılarını otomatik ayarlayabilen (self tuning) sistem üzerinde çalışmalar yapmak mevcut çalışmayı daha ileriye taşıyacaktır.

## KAYNAKLAR

- [1] Aggarwal V., Mao M., O'Reilly U., A self-tuning analog proportional integral derivative (PID) controller, *Adaptive Hardware and Systems*, DOI: 10.1109/AHS.2006.12.
- [2] Markarođlu H., İntegral yığılması problemine karşı yeni bir yaklaşım, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2006, 221961.
- [3] Kurtulan S., *PLC ile endüstriyel otomasyon*, 4. Basım, Birsen Yayınevi, İstanbul, 2005
- [4] Siemens Sitrain otomasyon ve kontrol eğitim bölümü, SIMATIC S7-300/S7-400 ile PID kontrol, İstanbul, 2011.
- [5] Kuo B. C., *Otomatik kontrol sistemleri*, 7.Basım, Literatür Yayınları, İstanbul, 2006.
- [6] Kurtulan S., *PLC ile endüstriyel otomasyon*, 5. Basım, Birsen Yayınevi, İstanbul, 2008
- [7] Hurma H. N., PID kontrolör ve PLC uygulaması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 1998, 75219.
- [8] Eminođlu Y., PLC programlama ve S7-300 ders notları, Kocaeli Anadolu Teknik Lisesi Otomatik Kumanda Bölümü, Kocaeli, 2003.
- [9] [http://www.20sim.com/webhelp/library/signal/control/pid\\_control/antiwindup.htm](http://www.20sim.com/webhelp/library/signal/control/pid_control/antiwindup.htm) (Ziyaret Tarihi: 21 Temmuz 2011)
- [10] Astrom K. J., Hagglung T., *PID Controllers: Theory, design and tuning*, 2nd Edition, Instrument Society of America, New York, 1995
- [11] Subaşı Ö., Öz ayarlamalı ve sıçramasız geçişli kontrol sistemi tasarımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2007, 223559.
- [12] Coşkun İ., Terziođlu H., Hız performans eğrisi kullanarak kazanç (PID) parametrelerinin belirlenmesi, *Teknik-Online Dergi*, 2007, **6**, 180-205.
- [13] Bilgiç Ş., PLC'ye giriş , Elsim eğitim semineri, İstanbul, 2009.
- [14] Siemens Sitrain otomasyon ve kontrol eğitim bölümü, SIMATIC S7-300/400 Sistem 1, İstanbul, 2008.

- [15] Yersel M. A., PID yönteminin PLC’de yazılarak gereklenmesi: ift cidarlı reaktör tank sıcaklık kontrolü örneđi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2007, 201169.

## **EKLER**

## **EK-A**

### **SCL PROGRAMI**

```
FUNCTION_BLOCK FB1
VAR_INPUT
X:INT; //GIRIS
X1:REAL;
Y1:REAL;
TO_SYS:REAL;
L:REAL;
T:REAL;
BZC:INT;
YMIN:REAL;
YMAX:REAL;
RST:BOOL;
END_VAR
VAR_OUTPUT
Y:INT;//TAMSAYI SISTEM CIKISI
KC:REAL;
TI:REAL;
TD:REAL;
END_VAR
VAR_TEMP
DX:INT;
a_1,a_0,b_1,b_0:REAL;
index:int;
A:REAL;
K_SYS:REAL;
BZC_R:REAL;
BZC_Y:REAL;
D:REAL;
ZB:REAL;
KY:REAL;
B0:REAL;
B1:REAL;
KZ:REAL;
END_VAR
VAR
Y_R:ARRAY[-3..0] OF REAL;
X_R:ARRAY[-100..1] OF REAL;
END_VAR

BEGIN
OK:=TRUE;
```

```

IF TO_SYS >0.0 THEN
A:=EXP(-T/TO_SYS);
K_SYS:=Y1/X1;
b_1:=1-A-A*T/TO_SYS;
b_0:=A*A-A+T*A/TO_SYS;
a_1:=-2*A;
a_0:=A*A;
D:=L/T;
TD:=A*A*T/(1.0-A*A);
TI:=(1.0+A)*T/(1.0-A);
B0:=A*(A-1.0+T/TO_SYS)/(1.0-A-A*T/TO_SYS);
KZ:=K_SYS*(1.0-A-A*T/TO_SYS);
B1:=(B0*(D+2.0)-D)/(D+1.0);
ZB:=(-B1+SQRT(B1*B1+4*B0))/2.0;
KY:=(ZB**(D+1.0)*(1.0-ZB))/(ZB+B0);
KC:=KY*T/(KZ*(T+TD));

ELSE
b_1:=0.0;
b_0:=0.0;
a_1:=0.0;
a_0:=0.0;
OK:=false;
RETURN;
END_IF;

DX:=REAL_TO_INT (L/T);

IF DX >= 100 THEN
  DX:=100;
END_IF;

IF RST THEN
Y_R[-2]:=0.0;
Y_R[-1]:=0.0;
X_R[-DX-1]:=0.0;
X_R[-DX-2]:=0.0;
END_IF;

Y_R[0]:=-a_1*Y_R[-1]-a_0*Y_R[-2]+K_SYS*(b_1*X_R[-DX-1]+b_0*X_R[-DX-2]);

IF Y_R[0] > 32000.0 THEN
  Y_R[0]:=32000.0;
END_IF;

IF Y_R[0] < -32000.0 THEN
  Y_R[0]:=-32000.0;
END_IF;

```

```

Y_R[-2]:=Y_R[-1];
Y_R[-1]:=Y_R[0];

X_R[0]:=INT_TO_REAL(X);

FOR index:= (-DX-1) TO 0 BY 1 DO
X_R[index-1]:=X_R[index];
END_FOR;

BZC_R:=INT_TO_REAL(BZC);

BZC_Y:=Y_R[0]-BZC_R;

IF BZC_Y <= YMIN THEN
    BZC_Y:= YMIN;
END_IF;

IF BZC_Y >= YMAX THEN
    BZC_Y:= YMAX;
END_IF;

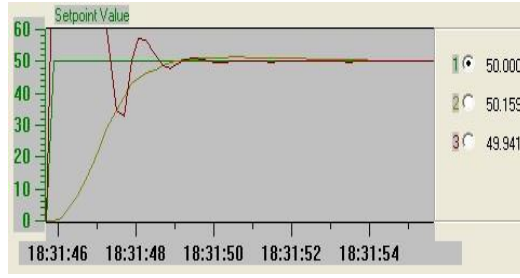
Y:=REAL_TO_INT(BZC_Y);

END_FUNCTION_BLOCK

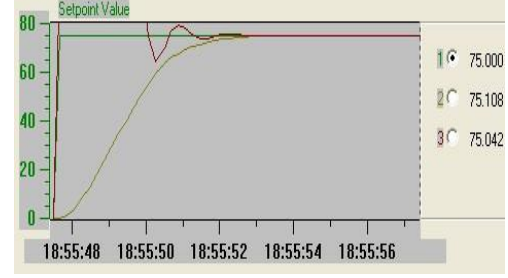
```

## EK-B

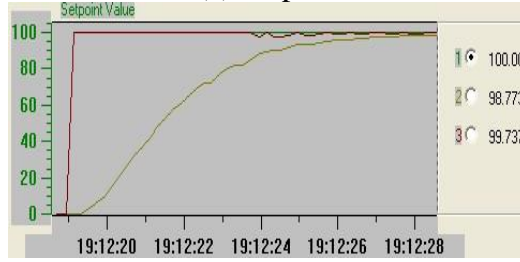
### ÖRNEKLEME ZAMANLARINA VE SETPOINT DEĞERLERİNE GÖRE ELDE EDİLEN SİMÜLASYON GRAFİKLERİ



(a) Setpoint=50

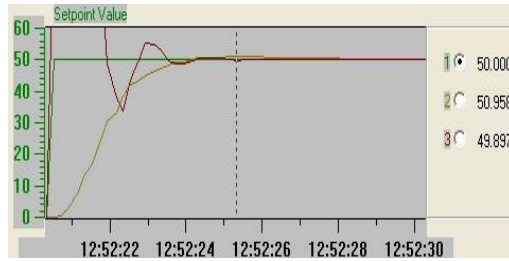


(b) Setpoint=75

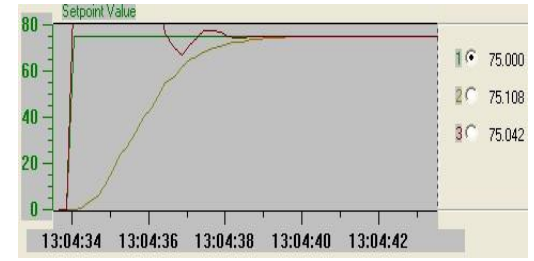


(c) Setpoint=100

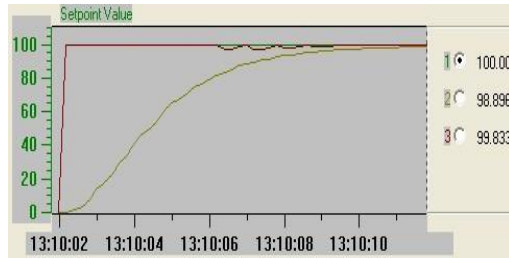
Şekil B.1: Örnekleme zamanı=100ms için simülasyon görüntüleri



(a) Setpoint=50



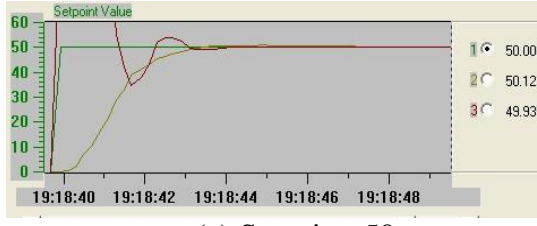
(b) Setpoint=75



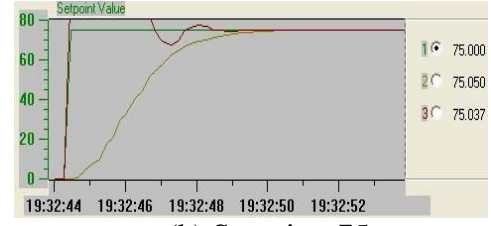
(c) Setpoint=100

Şekil B.2: Örnekleme zamanı=120ms için simülasyon görüntüleri

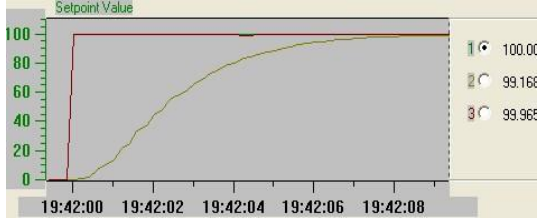




(a) Setpoint=50

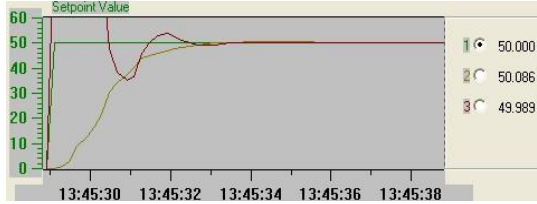


(b) Setpoint=75



(c) Setpoint=100

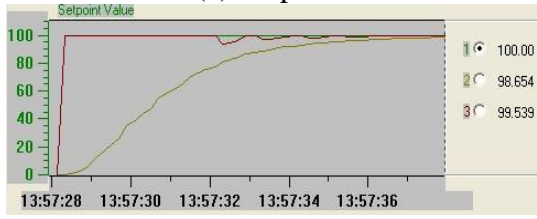
Şekil B.3: Örnekleme zamanı=140ms için simülasyon görüntüleri



(a) Setpoint=50

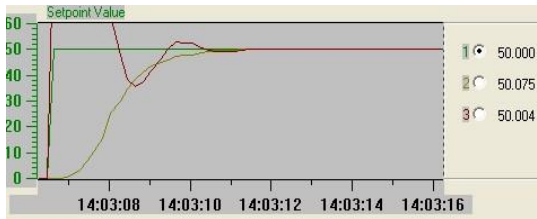


(b) Setpoint=75

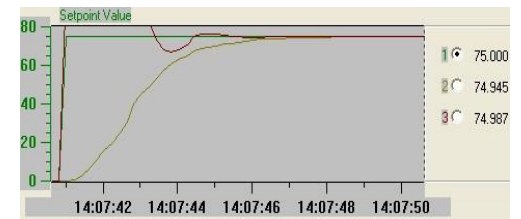


(c) Setpoint=100

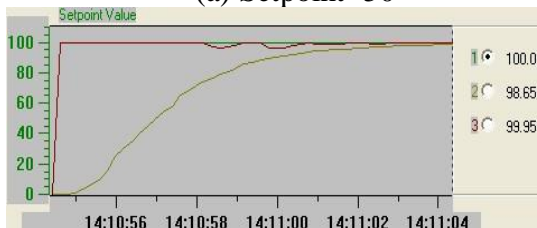
Şekil B.4: Örnekleme zamanı=160ms için simülasyon görüntüleri



(a) Setpoint=50

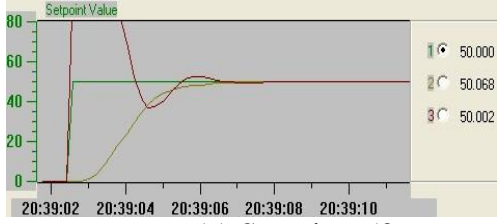


(b) Setpoint=75

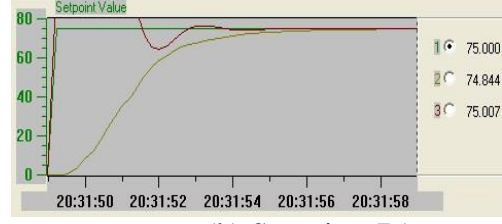


(c) Setpoint=100

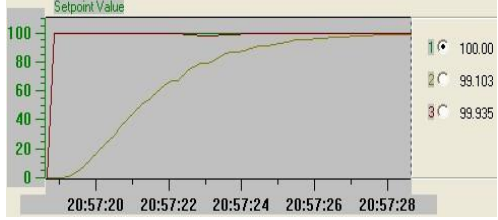
Şekil B.5: Örnekleme zamanı=180ms için simülasyon görüntüleri



(a) Setpoint=50

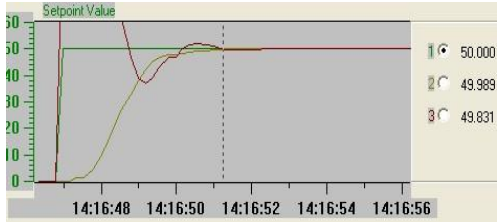


(b) Setpoint=75

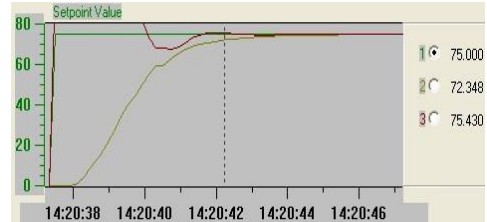


(c) Setpoint=100

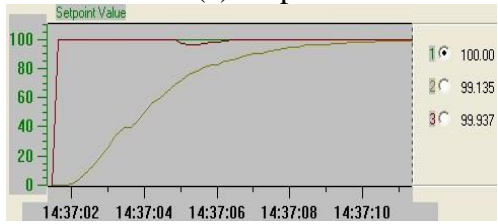
Şekil B.6: Örnekleme zamanı=200ms için simülasyon görüntüleri



(a) Setpoint=50

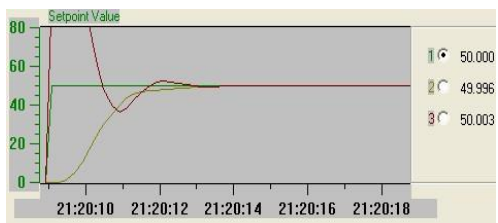


(b) Setpoint=75

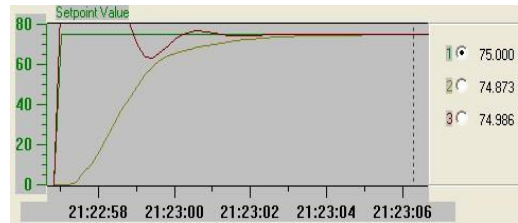


(c) Setpoint=100

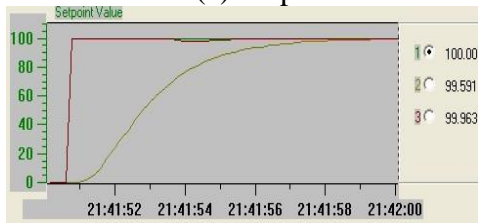
Şekil B.7: Örnekleme zamanı=220ms için simülasyon görüntüleri



(a) Setpoint=50

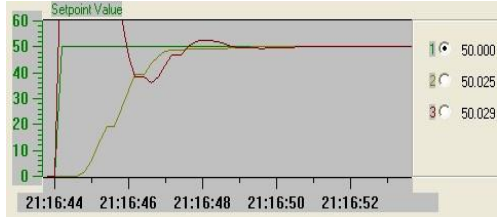


(b) Setpoint=75

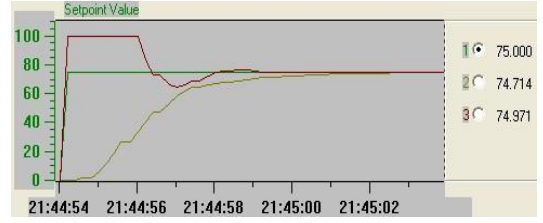


(c) Setpoint=100

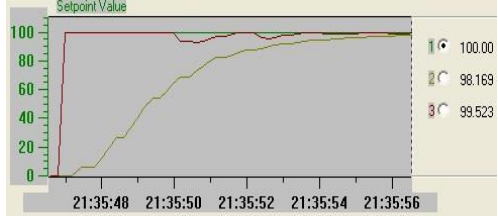
Şekil B.8: Örnekleme zamanı=240ms için simülasyon görüntüleri



(a) Setpoint=50

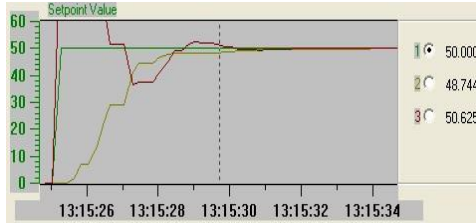


(b) Setpoint=75

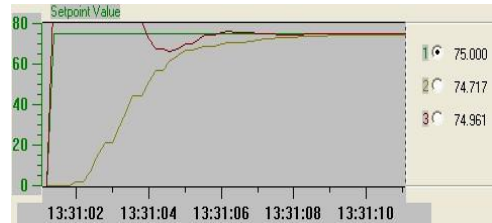


(c) Setpoint=100

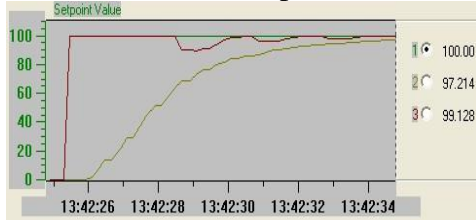
Şekil B.9: Örnekleme zamanı=260ms için simulasyon görüntüleri



(a) Setpoint=50

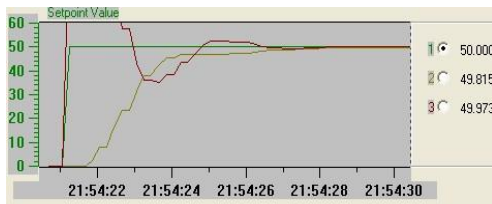


(b) Setpoint=75

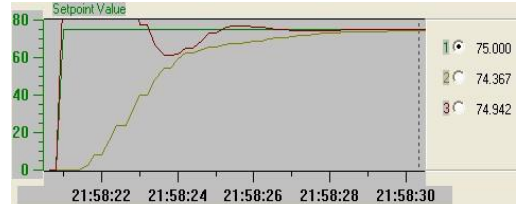


(c) Setpoint=100

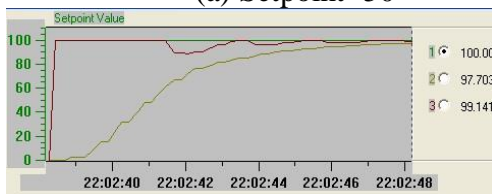
Şekil B.10: Örnekleme zamanı=280ms için simulasyon görüntüleri



(a) Setpoint=50

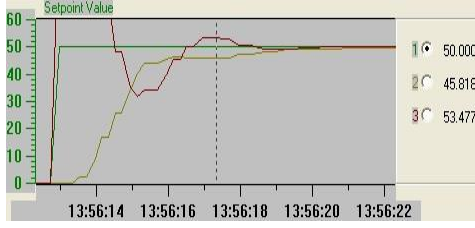


(b) Setpoint=75

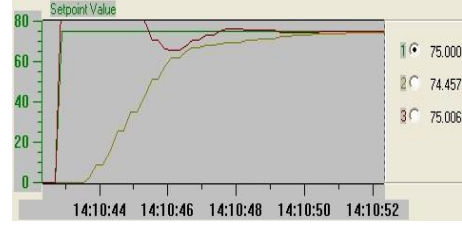


(c) Setpoint=100

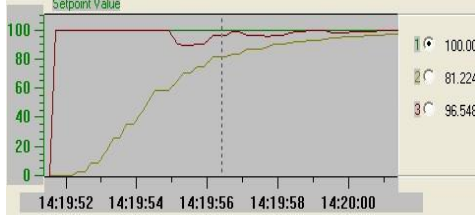
Şekil B.11: Örnekleme zamanı=300ms için simulasyon görüntüleri



(a) Setpoint=50

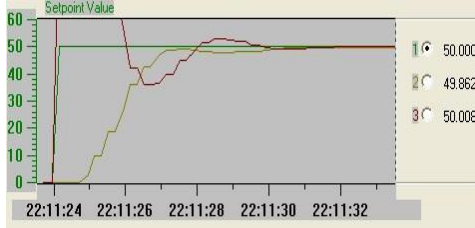


(b) Setpoint=75

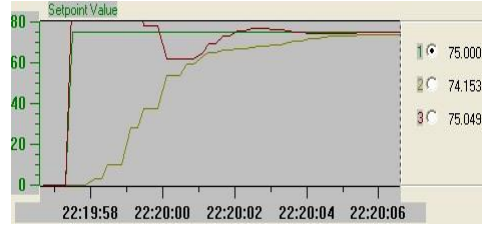


(c) Setpoint=100

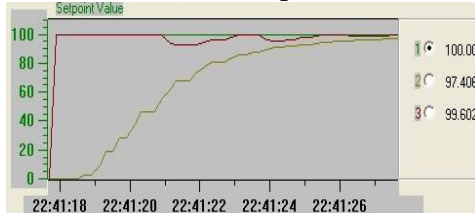
Şekil B.12: Örnekleme zamanı=320ms için simülasyon görüntüleri



(a) Setpoint=50

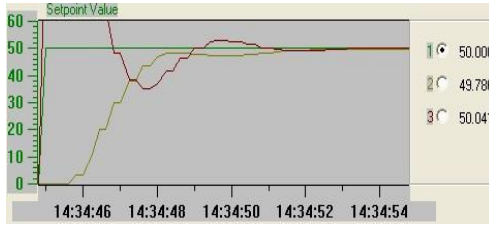


(b) Setpoint=75

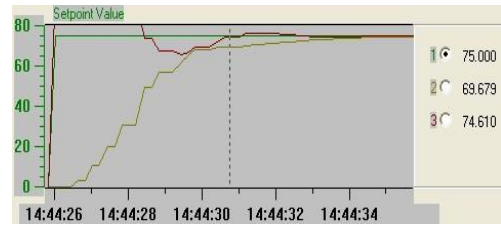


(c) Setpoint=100

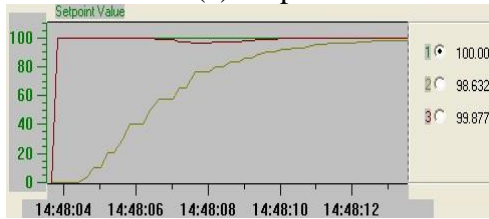
Şekil B.13: Örnekleme zamanı=340ms için simülasyon görüntüleri



(a) Setpoint=50

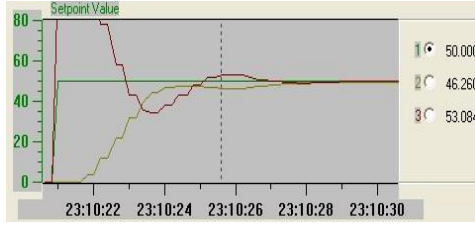


(b) Setpoint=75

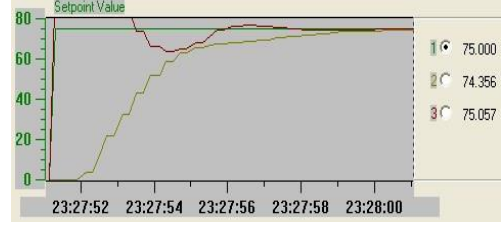


(c) Setpoint=100

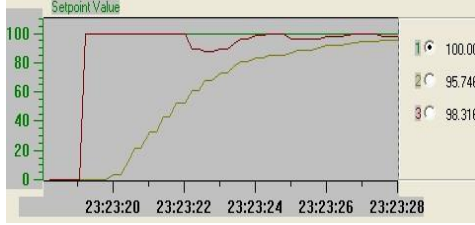
Şekil B.14: Örnekleme zamanı=360ms için simülasyon görüntüleri



(a) Setpoint=50

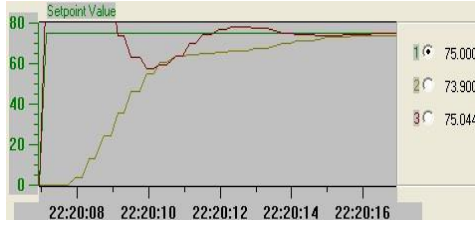


(b) Setpoint=75

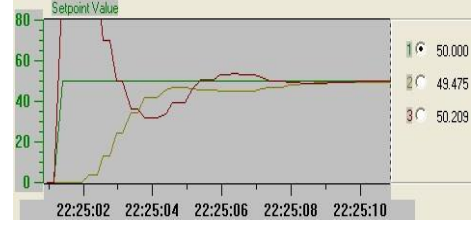


(c) Setpoint=100

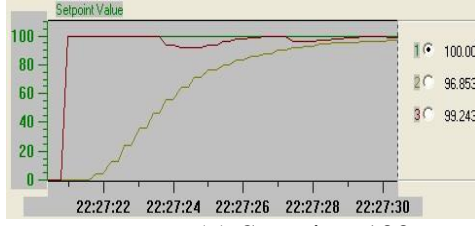
Şekil B.15: Örnekleme zamanı=380ms için simülasyon görüntüleri



(a) Setpoint=50



(b) Setpoint=75



(c) Setpoint=100

Şekil B.16: Örnekleme zamanı=408ms için simülasyon görüntüleri

## **ÖZGEÇMİŞ**

1985 yılında İzmit'te doğdu. İlk, orta ve lise öğrenimini İzmit'te tamamladı. 2003 yılında girdiği Kocaeli Üniversitesi Elektrik Mühendisliği Bölümü'nden 2008 yılında mezun oldu. Mart 2008 tarihinden itibaren Kalibre Boru A.Ş.'de "Bakım Mühendisi" olarak çalışmaya başladı. 2008-2009 güz döneminde Kocaeli Üniversitesi Elektrik Mühendisliği Bölümü'ne yüksek lisans eğitimi için kabul edildi. Şubat 2011 tarihinden itibaren Kalibre Boru A.Ş.'de "Bakım Şefi" olarak çalışmaya devam etmektedir. İzmit'te ikamet etmektedir.