

İSTANBUL TEKNİK ÜNİVERSİTESİ ELEKTRİK-ELEKTRONİK FAKÜLTESİ

**BİRİNCİ MERTEBEDEN ÖLÜ ZAMANLI
SİSTEMLER İÇİN, KENDİNİ AYARLAYAN BİR
KONTROL SİSTEMİ GERÇEKLENMESİ**

BİTİRME PROJESİ

İREM KOCA OYTUN ERİŞ

040030533 040040452

Anabilim Dalı : KONTROL VE KUMANDA SİSTEMLERİ

Programı : KONTROL MÜHENDİSLİĞİ

MAYIS 2008

ÖNSÖZ

Kontrol Mühendisliđi'ni tanımamıza ve çok sevmemize yardımcı olan tüm Kontrol Mühendisliđi akademik kadrosuna ve lisans hayatımız boyunca bize güvenen, aldığımız her kararda bize destek olup, üzerimizde sonsuz emeđi bulunan Doç. Dr. Salman KURTULAN'a en içten teşekkürlerimizi sunarız.

İÇİNDEKİLER	3
KISALTMALAR	4
ÖZET	5
SUMMARY	6
1. GİRİŞ	7
2. BİRİNCİ MERTEBEDEN ÖLÜ ZAMANLI SİSTEMLER	9
2.1 Matematiksel Model	9
2.2 Ayrık Zaman Modeline İlişkin Transfer Fonksiyonu	10
2.3 Kontrol Kuralının Bulunması	11
2.4 PT326 Sisteminin Tanıtılması	13
3. SİSTEMİN MODELLENMESİ VE KONTROL KURALININ BULUNMASI	15
3.1 Başlangıç Durumundan Sistem Belirleme	15
3.2 Çalışır Durumda Sistem Belirleme	20
4. KONTROL SİSTEMİNİN UYGULANMASI	32
4.1 Başlangıç Durumundan Sistemi Belirleyen Fonksiyon Bloğu ve Kullanımı	32
4.2 Çalışır Durumda Sistemi Belirleyen Fonksiyon Bloğu ve Kullanımı	33
4.3 FB41 PID Fonksiyon Bloğunun Kullanımı	33
5. UYGULAMA SONUÇLARI VE TARTIŞMA	38
5.1 Başlangıç Durumundan Sistem Belirleme ve Kontrol	38
5.2 Çalışan Sistemin Belirlenmesi ve Kontrolü	40
5.3 Uygulama Sonuçlarının Değerlendirilmesi	44
6. SONUÇ	50
KAYNAKLAR	51
ÖZGEÇMİŞ	52

KISALTMALAR

- CPU** : İşlemci (Central Processing Unit)
DB : Veri Bloğu (Data Block)
FB : Fonksiyon Blok (Function Block)
INT : Tam Sayı (Integer)
LMN : Hesaplanan Değer (Manipulated Value)
OB : Organizasyon Blok (Organization Block)
PI : Oransal İntegral (Proportional Integral)
PID : Oransal İntegral Türev (Proportional Integral Derivative)
PLC : Programlanabilir Lojik Kontrolör (Programmable Logic Controller)
PV : Süreç Değeri (Process Value)
SP : Referans Değeri (Set Point)

ÖZET

Bu bitirme çalışmasının kapsamında endüstride sık sık karşılaşılan birinci mertebeden ölü zamanlı sistemlerin otomatik olarak tanınması ve kontrol edilmesi amaçlanmıştır. Bu amaçla yazılan program, endüstride yaygın olarak kullanılan SIEMENS S7-300 ve S7-400 tipi PLC'lere uyum sağlayacak şekilde "SIMATIC MANAGER" programıyla hazırlanmıştır.

Çalışmanın giriş bölümünde projenin amaçları anlatılmış, proje aşamaları ayrıntılı olarak açıklanmıştır.

İlerleyen bölümlerde ise birinci dereceden ölü zamanlı sistemlere uygun kontrolörün nasıl tasarlanacağı ayrıntılı bir şekilde açıklanmış, proje dahilinde deney seti olarak kullanılan PT326 sistemi tanıtılarak gerekli bağlantıların nasıl yapılacağı gösterilmiştir. Sistemi tanıyan ve kontrolörü tasarlayan PLC programı açıklamalarıyla beraber verilmiştir. Aynı zamanda tasarlanan kontrolörün sisteme nasıl uygulanacağı ayrıntılı şekilde gösterilmiştir.

Sonuçlar kısmında ise farklı sistemlerin, farklı yollardan tanınması ve kontrolü ile elde edilen sonuçlar, programın değerlendirilmesi için grafikler halinde verilmiştir.

SUMMARY

In this graduation project, the parameter recognition and the control of the first order dead timed systems are aimed to be done automatically. The program prepared for this purpose is written with SIMATIC MANAGER in order to be compatible with SIEMENS S7-300 and S7-400 type PLC's which are widely used in industrial applications.

In the introduction the aim of the project is explained. In addition to this, the steps of the project are given

The design of the controller parameters for first order plus dead time systems is explained step by step in the further parts of the project. Experiment set PT326 which is used in the project is introduced and its connection to a PLC is described. Also, the PLC program which recognizes the system parameters and designs the proper controller is given with its explanations.

As a conclusion, the graphics of the different systems which are recognized and controlled in different ways are given in order to evaluate the program written.

1. GİRİŞ

Endüstriyel uygulamaların büyük bir bölümü, birinci mertebeden ölü zamanlı sistem olarak tanımlanabilen süreçlerden oluşur. Bu tür süreçler yüksek mertebeden olsalar bile, baskın dinamik davranışları göz önüne alındığında, bu süreçlerin birinci mertebeden sistemler gibi modellenebileceği görülür. Bu modelden yola çıkılarak kontrol yapısı ve kuralı belirlenebilir. Bu çalışmada, matematiksel modeli birinci mertebeden ölü zamanlı sistem olarak verilebilen endüstriyel süreçler için, basit, kolay uygulanabilir ve kendiliğinden ayarlamaya olanak veren bir kontrol kuralı PLC kontrol biriminde gerçekleştirilmiş ve bir süreç uygulanmıştır.

Çalışma 6 bölümden oluşmaktadır. Birinci bölümü izleyen ikinci bölümde sürece ilişkin matematiksel modelin çıkarılması için gerekli işlemler verilmiştir. Bu bölümde birinci mertebeden ölü zamanlı bir sürecin, basamak girişe yanıtından yola çıkılarak model parametrelerinin nasıl bulunacağı anlatılmıştır. İşletme koşulları nedeniyle iki durum göz önüne alınmıştır. Birinci durum ilk koşulları sıfır veya sıfırdan farklı bir sürecin tanınması; ikinci durum ise kapalı çevrim altında devrede olan bir sürecin izin verilen çalışma aralığında belirlenmesidir. Bu bölüm içinde aynı zamanda, PID ile gerçekleştirilecek kontrol kuralı bulunmuştur. Kontrol kuralının bulunmasında, sisteme ilişkin z-tanım bölgesi tasarım yöntemlerinden yararlanılmıştır. Bu yöntemin seçilmesindeki nedenler şu şekilde açıklanabilir:

- Endüstriyel süreçler genellikle ölü zaman içermektedir. Ölü zaman, z-tanım bölgesinde doğrudan modellenebilirken, s-tanım bölgesindeki tasarım yöntemlerinde yaklaşıklıklar (Pade, Tustin, ..) kullanılır .
- PID kontrol kuralı sayısal gerçekleştirildiğinden, kontrol kuralının fark denklemleriyle verilmesi doğrudan uygulamaya olanak sağlar. Ayrıca, örnekleme zamanının etkisi tasarım aşamasında değerlendirilebilir.
- Ölü zamanın modelde yer alması, bu tür sistemler için kararlılık incelemesini kolaylaştırır.

Yapılan tasarımda, sistemin basamak girişe olan kapalı çevrim yanıtının, bu tür endüstriyel süreçler için en uygun çözüm olduğu düşünülen, kritik sönümlü hale getirilmesi benimsenmiştir. Bunun sebebi şöyle açıklanabilir:

- Endüstriyel süreçlerin çalışma şartlarında genellikle sistemin verilen referansı aşması istenmez. Kritik sönümlü davranışta, bozucu etki ya da referans değişikliğinde, sistem verilen referans değerini aşım yapmadan izler.
- Endüstriyel süreçlerde sistemin olabildiğince hızlı şekilde, verilen referans değerine erişmesi istenir. Sistemin aşım yapmadan verilen referans değerine en hızlı eriştiği çözüm, kapalı çevrim sistem yanıtının kritik sönümlü yapılmasıyla bulunur.
- Endüstriyel süreçlerde sisteme uygulanabilecek olan kontrol gücü sınırlı olduğundan kontrol işareti de sınırlıdır. Kritik sönümlü davranışta aşırı bir kontrol gücü gerekmediğinden, kontrol kuralına göre hesaplanan kontrol işareti sınırlandırılmadan sisteme uygulanabilir.

İkinci bölüm içinde son olarak PT326 süreç benzetim düzeneği tanıtılmış, kontrol kuralının gerçekleştirilmesi için kullanılacak olan PLC ile bağlantılarının nasıl yapılması gerektiği ayrıntılı olarak açıklanmıştır.

Üçüncü bölümde, bulunan kontrol kuralının PLC ile gerçekleştirilmesi ve PT326 süreç benzetim düzeneğinde elde edilen farklı sistem modelleri üzerinde denenmesi için yazılan kontrol sistemi açıklamalarıyla beraber verilmiştir. Bu program ile sistem modeline ilişkin

parametrelerin bulunması ve daha önceden verilmiş olan kontrol kuralına göre kontrolör parametrelerinin hesaplanıp sisteme uygulanması hedeflenmiştir.

Dördüncü bölümde, hazırlanmış olan kontrol sisteminin nasıl kullanılması gerektiği şekiller ile ayrıntılı bir biçimde anlatılmıştır. Uygulamalarda SIEMENS firmasının standart PID fonksiyon bloğu (FB41) kullanılmıştır. Bunun nedeni, bu fonksiyon bloğunun endüstride çok kullanılması ve bilinmesidir. Aynı bölüm içinde bu fonksiyon bloğunun kullanımı ve uygulama şeması ayrıntılı olarak anlatılmıştır.

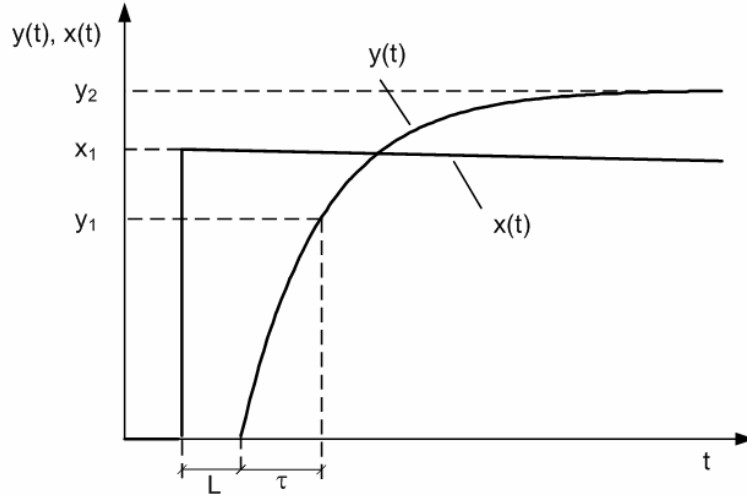
Beşinci bölümde, kontrol sistemi PT326 süreç benzetim düzeneğinde elde edilen farklı sistem modelleri üzerinde denenmiş ve elde edilen sonuçlar grafikler halinde verilmiştir. Bu şekilde kontrol sisteminin farklı sistemler üzerinde olan başarısının karşılaştırmalı olarak değerlendirilmesine olanak tanınmıştır.

Son bölümde ise yapılan çalışma değerlendirilmiş ve sonuçlar tartışılmıştır.

2. BİRİNCİ MERTEBEDEN ÖLÜ ZAMANLI SİSTEMLER

2.1. Matematiksel Model

Basamak girişe yanıtı Şekil 2.1'deki gibi olan sistemler birinci mertebeden ölü zamanlı sistemler olarak tanımlanabilir.



Şekil 2.1 Birinci mertebeden ölü zamanlı sistemin basamak cevabı

Birinci mertebeden ölü zamanlı bir sistemin s-tanım bölgesi transfer fonksiyonu. Y(s) çıkış büyüklüğü, X(s) giriş büyüklüğü olmak üzere

$$G(s) = \frac{Y(s)}{X(s)} = \frac{Ke^{-sL}}{(\tau s + 1)} \quad (2.1)$$

biçiminde verilir. Burada K kazanç, τ zaman sabiti ve L ölü zaman değeri olarak tanımlanır.

Birinci mertebeden ölü zamanlı bir sistemin genliği x_1 olan basamak girişe yanıtının zamana bağlı değişimi, çıkışın başlangıç değeri $y_0 = 0$ için

$$y(t) = Kx_1(1 - e^{-(t-L)/\tau}) \quad (2.2)$$

biçiminde ifade edilir. Burada L parametresi sürecin giriş işaretine tepkisiz olduğu zaman değeri olup ölü zaman olarak tanımlanır. (2.2) ifadesi $t = L + \tau$ için düzenlenirse

$$\begin{aligned} y(L + \tau) &= Kx_1(1 - e^{-(L+\tau-L)/\tau}) = Kx_1(1 - e^{-\tau/\tau}) \\ &= Kx_1(1 - e^{-1}) = Kx_1(1 - 0.368) = 0.6321Kx_1 \end{aligned}$$

$t = L + 5\tau$ için düzenlenirse

$$y(L + 5\tau) = Kx_1(1 - e^{-(L+5\tau-L)/\tau}) = Kx_1(1 - e^{-5}) = 0.9933Kx_1$$

olarak elde edilir. Bu ilişkilere göre birinci mertebeden ölü zamanlı bir sisteme genliği x_1 olan bir basamak giriş işareti uygulanırsa, $L + 5\tau$ süre geçtikten sonra çıkış son değerinin 99% oranına ulaşılır ve kazanç

$$K = \frac{y(L + 5\tau)}{0.9933x_1} \cong \frac{y_2}{0.9933x_1} \cong \frac{y_2}{x_1} \quad (2.3)$$

ifadesine göre bulunur.

Birinci mertebeden ölü zamanlı bir sisteme genliği x_1 olan bir basamak giriş işareti uygulanırsa, $L + \tau$ süre sonra çıkış son değerine 63.21% oranına ulaşılır. Buna göre zaman sabiti τ , çıkış son değerinin 63.21% oranına ulaşıncaya kadar geçen süreden ölü zaman değeri çıkartılarak elde edilir.

Sürece ilişkin modelin, işletme koşulları altında çıkartılmasını gerektiren durumlarda ilk koşullar sıfırdan farklıdır. Bu tür uygulamalarda, yapılan iş gereği kontrol edilen büyüklüğün belirli bir aralıkta tutulması istenir. Birinci mertebeden ölü zamanlı bir sistemin genliği x_1 olan basamak girişe yanıtı $y_1(t)$, genliği x_2 olan basamak girişe yanıtı $y_2(t)$ ve ilk koşul $y_0 \neq 0$ olmak üzere, zamana bağlı ifadeleri

$$y_1(t) = K(1 - e^{-(t-L)/\tau})x_1(t-L) + y_0 e^{-(t-L)/\tau} \quad (2.4)$$

$$y_2(t) = K(1 - e^{-(t-L)/\tau})x_2(t-L) + y_0 e^{-(t-L)/\tau} \quad (2.5)$$

biçiminde yazılabilir. Bu iki ifadeden

$$y_2(t) - y_1(t) = K(1 - e^{-(t-L)/\tau})(x_2(t-L) - x_1(t-L)) \quad (2.6)$$

ilişkisi bulunur.

$$\Delta y = y_2(t) - y_1(t) \quad , \quad \Delta x = x_2(t-L) - x_1(t-L) \quad (2.7)$$

olarak tanımlanırsa

$$\Delta y = K(1 - e^{-(t-L)/\tau})\Delta x \quad (2.8)$$

ifadesi elde edilir. Bu ifadeye göre girişin değişimi ile çıkışın değişimi arasındaki ilişkinin sistemin ilk koşullarından bağımsız olduğu görülür. Bu ifadeden yararlanılarak sisteme ilişkin parametreler bulunur.

2.2 Ayrık Zaman Modeline İlişkin Transfer Fonksiyonu

Endüstriyel sayısal kontrol sistemine ilişkin kontrol kuralını gerçeklemek için kullanılan sayısal kontrol biriminde, belirli zaman dilimlerinde alınan (zamanda ayrık) veriler değerlendirilir. Kontrol kuralının ayrık zaman değerlerine göre işlenmesi nedeniyle sürece ilişkin ayrık zaman modeli 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, sayısal ortamda gerçek zamanlı olarak değerlendirilebilir.

Ayrık zaman modeline ilişkin matematiksel model z-tanım bölgesi transfer fonksiyonu ile verilir. Genel olarak s-tanım bölgesi transfer fonksiyonu $G(s)$ olarak verilen bir sistemin z-tanım bölgesi transfer fonksiyonu; $G_{ZOH}(s)$ sıfırıncı mertebeden tutucu ve T örnekleme zamanı olmak üzere

$$G(z) = Z\{G_{ZOH}(s)G(s)\}, \quad G_{ZOH}(s) = \frac{1 - e^{-sT}}{s} \quad (2.9)$$

ifadesi kullanılarak elde edilebilir.

Sıfırıncı mertebeden tutucu; belirli bir örnekleme zamanında alınan işareti, bir sonraki örnekleme zamanına kadar tutma işlevini görür. Bu projede uygulama örnekleri verilen sayısal kontrol çevre birimleri de bu şekilde çalışmaktadır. Kontrol edilen büyüklüğe ilişkin analog işaret belirli bir örnekleme zamanında okunur ve bir sonraki örnekleme zamanına kadar tutulur. Hesaplanan kontrol işareti analog çıkış birimine kaydedilir ve bir sonraki örnekleme zamanına kadar değişmez. Bu benzerlik nedeniyle z-tanım bölgesi transfer fonksiyonunun elde edilmesinde bu ilişkinin kullanılması, amaca uygun bir modelin elde edilmesine olanak sağlar.

Birinci mertebeden ölü zamanlı sistemlere ilişkin z-tanım bölgesi transfer fonksiyonu, $G_{ZOH}(s)$ sıfırıncı mertebeden tutucuya ilişkin transfer fonksiyonu olmak üzere

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

ifadesinden bulunabilir. Buna göre

$$G(z) = K \frac{(1-e^{-sT})e^{-sL}}{1} Z\left\{\frac{1}{s} - \frac{1}{s+1/\tau}\right\}$$

$z = e^{sT}$, $d = L/T$ ve $A = e^{-T/\tau}$ tanımlanırsa kullanılırsa

$$G(z) = K \frac{(1-z^{-1})z^{-d}}{1} Z\left\{\frac{1}{s} - \frac{1}{s+1/\tau}\right\} = K \frac{(z-1)z^{-d}}{z} Z\left\{\frac{1}{s} - \frac{1}{s+1/\tau}\right\}$$

$$G(z) = K \left(\frac{z-1}{z} z^{-d}\right) \left(\frac{z}{z-1} - \frac{z}{z-e^{-T/\tau}}\right) = K \left(1 - \frac{z-1}{z-A}\right) z^{-d} = K \frac{z-A-z+1}{z-1} z^{-d} \quad (2.10)$$

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

z -tanım bölgesi transfer fonksiyonu elde edilir.

2.3 Kontrol Kuralının Bulunması

Bir endüstriyel sürece ilişkin kontrol sisteminin tasarımında; kararlılık, sürekli durum hatası ve kapalı çevrim sistemin dinamik davranışı ile ilgili özellikler değerlendirilir. Uygulamada; kapalı çevrim sistemi kararlı kılan, sürekli durum hatasını en kısa sürede gideren ve basamak girişe yanıtın aşısız olmasını sağlayan bir kontrol kuralı, çoğu endüstriyel süreç için yeterli sayılır. Bu özellikleri sağlayan bir kontrol sistemi, kontrol gücünün sınırlı olması nedeniyle, genellikle, gerçekleştirilebilir en uygun çözümü verebilir. Bir sistemin kapalı çevrim yanıtının baskın dinamik davranışının kritik sönümlü olmasını sağlayan bir kontrol kuralı bu koşulları sağlar.

Birinci mertebeden ölü zamanlı bir sistem sistemin kapalı çevrim yanıtının kritik sönümlü olmasını sağlayan bir kontrol kuralı PI olarak seçilebilir.

PI kontrolörün transfer fonksiyonu T örnekleme zamanı, K_c oransal katsayı ve T_i integral zaman sabiti olmak üzere

$$G_{PI}(z) = K_c \left(1 + \frac{T}{T_i} \frac{1}{z-1}\right) \quad (2.11)$$

biçimindedir. Bu ifade sıfır-kutup biçiminde yazılıp, düzenlenip

$$G_{PI}(z) = K_c \left(1 + \frac{T}{T_i} \frac{1}{z-1}\right) = K_c \left(\frac{z - \frac{T_i - T}{T}}{z-1}\right)$$

$$\alpha = \frac{T_i - T}{T} \quad ; \quad T_i = \frac{T}{1 - \alpha} \quad (2.12)$$

tanımları yapılırsa

$$G_{PI}(z) = K_c \frac{z - \alpha}{z-1} \quad (2.13)$$

şeklinde elde edilir.

PI kontrolör ve z -tanım bölgesi transfer fonksiyonu (2.10) ifadesi ile verilen birinci mertebeden ölü zamanlı sistemden oluşan sistemin açık çevrim transfer fonksiyonu

$$G(z)G_{PI}(z) = K_c \frac{z - \alpha}{z-1} \frac{K(1-A)}{(z-A)} z^{-d} \quad (2.14)$$

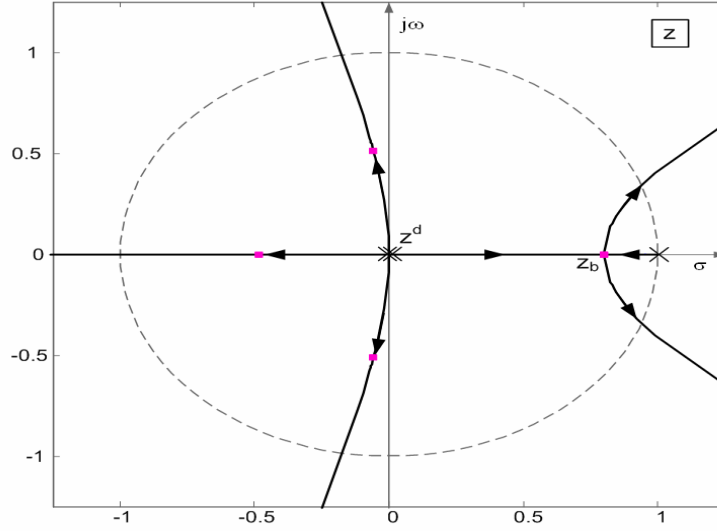
biçiminde yazılır ve $\alpha = A$ seçilirse

$$G(z)G_{PI}(z) = \frac{K_C K(1-A)}{z^d(z-1)} = \frac{K^*}{z^d(z-1)}, \quad K^* = K_C K(1-A) \quad (2.15)$$

elde edilir. Bu durumda kapalı çevrim sistemin transfer fonksiyonu

$$T(z) = \frac{G(z)G_{PI}(z)}{1 + G(z)G_{PI}(z)} = \frac{K^*}{z^d(z-1) + K^*} = \frac{K^*}{z^{d+1} - z^d + K^*} \quad (2.16)$$

şeklinde elde edilir. Bu sistemin $d=4$ için kapalı çevrim kutuplarının K^* kazancına göre değişim eğrisi (köklerin yer eğrisi) Şekil 2.2'deki gibi olacaktır.



Şekil 2.2 Kapalı çevrim kutuplarının yer eğrisi

Kapalı çevrim sistemin basamak girişe yanıtının kritik sönümlü olması için baskın kutupların z_b kopma noktasında bulunması gerekir. Kopma noktası, $\Delta(z)$ karakteristik denkleme ilişkin ifade K^* değişkenine göre düzenlenir ve $\frac{dK^*}{dz} = 0$ koşulundan bulunabilir.

$$\Delta(z) = 1 + G(z)G_{PI}(z) = z^{d+1} - z^d + K^* = 0 \quad (2.17)$$

ifadesinden yol çıkılarak

$$K^* = \frac{1-z}{z^{-d}} \quad (2.18)$$

elde edilir.

$$\frac{dK^*}{dz} = \frac{d}{dz} \left(\frac{1-z}{z^{-d}} \right) = \frac{-z^{-d} - (1-z)(-d)z^{-d-1}}{z^{-2d}} = 0$$

$$-z^{-d} - (1-z)(-d)z^{-d-1} = -z^{-d} + dz^{-d-1} - dz^{-d} = 0$$

$$= -z^{-d} + dz^{-d-1} - dz^{-d} = z^{-d}(-1 + dz^{-1} - d) = 0$$

$$(-1 + dz^{-1} - d) = 0$$

$$dz^{-1} = 1 + d$$

$$z_{b1} = \frac{d}{d+1} \quad (2.19)$$

$$z^{-d} = 0$$

$$z_{b2} = 0 \quad (2.20)$$

z_{b1} ve z_{b2} kopma noktaları bulur. Tasarımda, baskın dinamik davranışa ilişkin z_{b1} noktasındaki kopma noktası değerlendirilecek ve bu nokta $z_b = z_{b1}$ baskın ya da etkin kopma noktası olarak tanımlanacaktır.

Kopma noktasındaki kazanç değeri, (2.18) ifadesinde $z = z_b = z_{b1}$ değeri yazılarak bulunabilir.

$$K^* = \frac{1-z}{z^{-d}} = \frac{1-\frac{d}{d+1}}{\left(\frac{d}{d+1}\right)^{-d}} = \left(\frac{1}{d+1}\right)\left(\frac{d}{d+1}\right)^d = \frac{d^d}{(d+1)^{d+1}} \quad (2.21)$$

Buna göre kontrolör kazancı

$$K^* = K_c K(1-A) \quad (2.22)$$

ifadesinden

$$K_c = \frac{K^*}{K(1-A)} = \frac{d^d}{(d+1)^{d+1}} \frac{1}{K(1-A)} \quad (2.22)$$

şeklinde elde edilmiş olur.

İntegral zaman sabiti T_I , (2.12) ve (2.14) ifadelerinden

$$T_I = \frac{T}{1-\alpha}, \quad T_I = \frac{T}{1-A} \quad (2.23)$$

şeklinde bulunur.

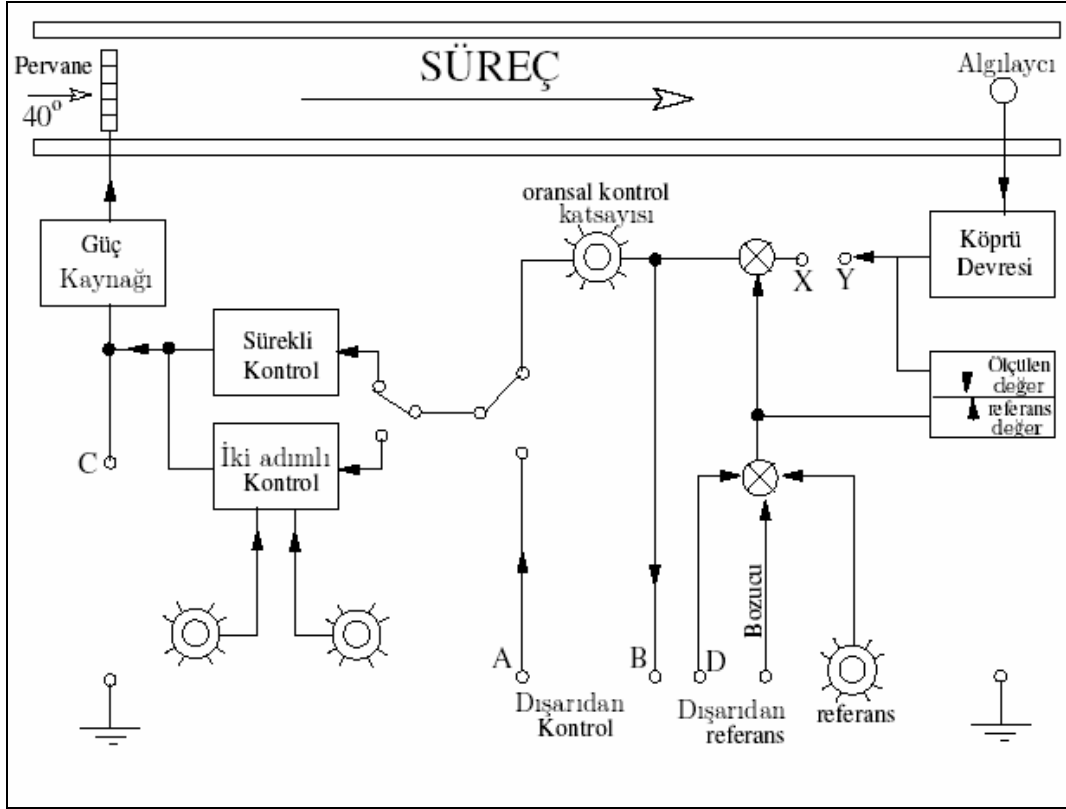
Sonuç olarak, birinci mertebeden ölü zamanlı bir sistemin, basamak girişe kapalı çevrim yanıtının kritik sönümlü olmasını sağlayan PI katsayıları

$$K_c = \frac{d^d}{(d+1)^{d+1} K(1-A)}, \quad T_I = \frac{T}{1-A} \quad (2.24)$$

ifadeleri kullanılarak bulunur.

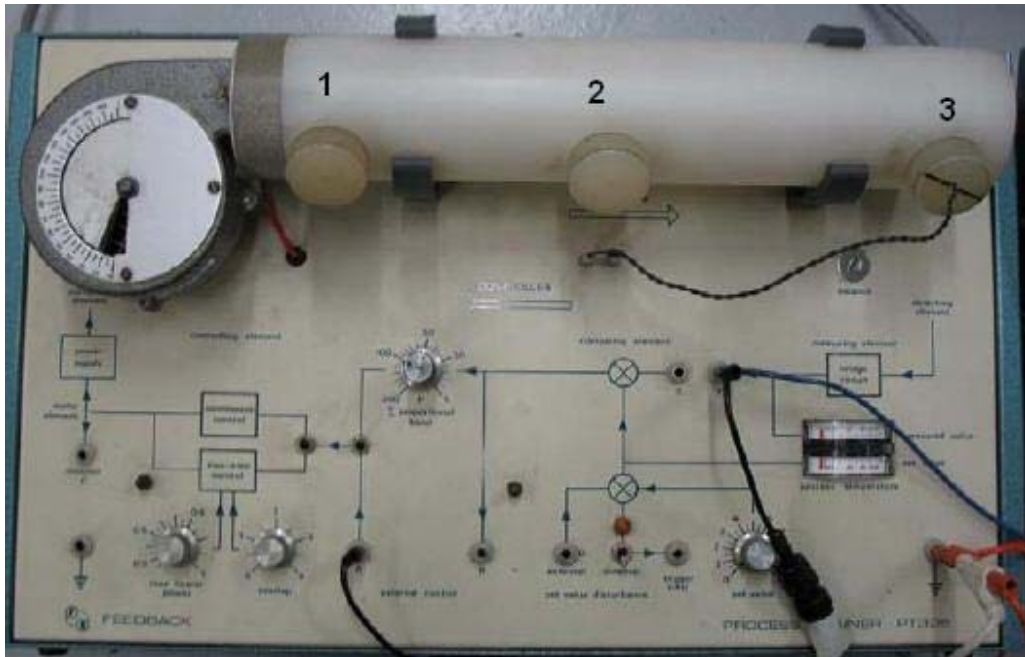
2.4 PT326 Sisteminin Tanıtılması

Yapılan çalışmada birinci mertebeden sistem olarak “Feedback Process Trainer PT326” ısıt sistemi kullanılmıştır. Sisteme PLC üzerinden rahatlıkla analog veri gönderilip, sistemden rahatlıkla analog veri alınabilmesi sistemin en büyük artısıdır. Bunun yanında sistemdeki ölçme gürültüleri sistemin en büyük eksilerini oluşturmaktadır.



Şekil 2.3PT326 ısı benzetim düzeneği şematik yapısı

Sistemin dışarıdan kontrolü için A ucuna PLC'nin birinci analog çıkışı, sistemden gelen verileri almak için Y ucuna PLC'nin birinci analog girişi ve son olarak toprakların ortak olması için sistemin toprağına analog giriş ve çıkışların toprakları bağlanır.



Şekil 2.4 PT326 ısı benzetim düzeneği

Sistemin karakteristiklerini deęiřtirmek iki yolla m¼mk¼nd¼r. İlk olarak pervanenin ¼st¼ndeki kapak belirlenmiř aıclar arasında deęiřtirelerek sistem dinamikleriyle oynanmıř olunur. İkinci bir yol ise algılayıcının, sistemdeki borunun ¼zerinde bulunan delikler arasında konum deęiřtirmesidir. Bu řekilde sistemin ¼l¼ zamanı arttırılıp azaltılabilir.

3. SİSTEMİN MODELLENMESİ VE KONTROL KURALININ BULUNMASI

3.1 Bařlangı Durumundan Sistem Belirleme

Kapalı bir kutu halinde olan sistemin belirlenebilmesi iin en sık bařvurulan y¼ntemlerden biri, sisteme basamak giriř uygulanarak sistem ıkıřının izlenmesidir. Bu y¼ntemi kullanarak verilen sistemi tanıyan ve sisteme daha ¼nce verilen kurallara uygun PI katsayılarını bulan program aıklamasıyla beraber ařaęıda verilmiřtir.

```
A #BASLA
FP #FP_BIT
JCN M001
R #V_ALINDI
R #Bitti
L P#0.0
T #ISARETCI
T #ISARETCI2
T #ISARETCISON
T #ISARETCISON2
L 0
T #ALINAN_V_S
L 0.000000e+000
T #SON_DEGER_REEL
S #V_ALINIYOR
```

Network 1:

Eęer sistem bařtan alıřtırılmak isteniyorsa giriř aktif edildięi anda programın iinde kalan ¼nceki verilerin hepsi temizlenir ve V_ALINIYOR biti setlenerek veri alınma iřlemine bařlandıęı g¼sterilir. Bu řekilde sistemin iinde kalmıř eski deęerler ile yanlıř bir iřlem yapılmasının ¼n¼ne geilir. Eęer sistem bařtan alıřtırılmıyorsa program M001 kısmına atlar.

```
M001: A #BASLA
A #V_ALINIYOR
JCN M002
OPN #D_BLOK
L #STEP_GENLIK
T #STEP_CIKIS
L #INPUT
T DBW [#ISARETCI]
L #ALINAN_V_S
L #V_SAYISI
>=I
R #V_ALINIYOR
S #V_ALINDI
JC M002
L P#2.0
L #ISARETCI
+D
T #ISARETCI
L 1
L #ALINAN_V_S
+I
T #ALINAN_V_S
```

Network 2:

Bařla butonuna basılı iken veri alınma iřlemine bařlanılmıř ise, verilerin kaydedileceęi veri bloęu aıldıktan sonra kullanıcının belirledięi genlikteki basamak giriř sisteme uygulanır ve sistemden kullanıcının belirledięi miktarda veri bir iřareti yardımıyla toplanır. Verilerin veri bloęuna sırasıyla kaydedilmesi iin iřaretinin deęeri her evrimde 2 arttırılır. Bunun sebebi kaydedilecek integer formatındaki verinin 2 byte olmasıdır.

Program hedeflenen veri sayısına ulařtıęında ya da veri toplama iřlemi yapılmadıęında M002 kısmına atlar.

```

M002: A #BASLA
      JC M003
      R #V_ALINIYOR
      R #V_ALINDI
      L 0
      T #STEP_CIKIS
M003: NOP 0

```

```

A #V_ALINDI
JCN SON
A #Bitti
JC M009
L #V_SAYISI
L 2
*D
L 8
-D
SLD 3
T #ISARETCISON2
L #V_SAYISI
L 2
*D
SLD 3
T #ISARETCISON

```

```

OPN #D_BLOK
L P#0.0
T #ISARETCI2
L DBW [#ISARETCI2]
T #ILK_DEGER
M004: L #ISARETCISON2
      T #ISARETCI2
      L DBW [#ISARETCI2]
      ITD
      DTR
      L #SON_DEGER_REEL
      +R
      T #SON_DEGER_REEL
      L P#2.0
      L #ISARETCISON2
      +D
      T #ISARETCISON2
      L #ISARETCISON2
      L #ISARETCISON
      >D
      JCN M004
      L #SON_DEGER_REEL
      L 5.000000e+000
      /R

```

Network 3:

Eğer sistem hala devredeyse program M003 kısmına atlar eğer devrede sisteme gönderilen basamak işaretin genliğini 0 yapar. Bunu sebebi PLC dur konumuna alınsa bile sisteme işaretleri gönderen analog karttaki bilgilerin silinmemesidir.

Network 4:

Verilerin toplanması tamamlandıktan sonra programın veri tablosundaki gereken yerlere gitmesi için işaretçilere gerekli değerlerin verilmesi gerekir. Bu kısımda kullanıcının girdiği veri sayısı 2 ile çarpılıp kapladığı alan byte cinsinden bulunduktan sonra bu sayı işaretçi olarak kullanılmak üzere 3 sola kaydırılır. Bu şekilde işaretçi veri tablosundaki son değeri işaret eder. Aynı şekilde bir hesaplama da son değerden 4 önceki değere gitmek için yapılır ve işaretçiye yüklenir. Eğer veri alınma işlemi bitmemişse program SON kısmına, eğer sistem tanıma işlemi bittiyse program M009 kısmına atlar.

Network 5:

Verilerin saklandığı veri bloğu açıldıktan sonra bir işaretçiye sıfır değeri atanarak, bu işaretçinin yardımıyla veri bloğundan sistemin ilk değeri alınır. Sonrasında daha önceden bulunan, sistemin sondan 4 önceki değerinin olduğu adresi taşıyan işaretçi kullanılarak program döngüye sokulur. Bu döngü ile veri bloğunun sonuna kadar olan 5 değer reel sayıya çevrilerek toplandıktan sonra bu değer 5'e bölünür


```

RND
T #SON_DEGER
L #ILK_DEGER
-I
T #FARK
DTR
T #FARK_REEL
L #STEP_GENLIK
ITD
DTR
L #FARK_REEL
L #STEP_GENLIK_REEL
/R
T #K
L P#0.0
T #ISARETCISON
T #ISARETCISON2

```

```

L #ILK_DEGER
ITD
DTR
T #ILK_DEGER_REEL
L #FARK_REEL
L 1.000000e-002
*R
L #ILK_DEGER_REEL
+R
RND
T #UST_SINIR
L P#0.0
T #ISARETCI2
M005: OPN #D_BLOK
L DBW [#ISARETCI2]
L #UST_SINIR
>I
JC M006
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M005

```

```

M006: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #OLU_ZAMAN

```

Network 5(Devam):

Elde edilen bu değer 5'e bölünerek sistemin son değeri olarak kaydedilir. Bu şekilde sistemdeki gürültülerin etkisiyle son değerın yanlış hesaplanma ihtimali azaltılmış olur. Sistemin son değerinden ilk değeri çıkarılarak sistemdeki değişim bulunmuş olur. Bu değer daha sonra ondalıklı işlemlerin yapımına uygun olması için reel sayıya çevrilir. Sistemin çıkışındaki değişim, girişindeki değişime bölünerek sistemin kazancı bulunur. Sonrasında kullanılan işaretçiler daha sonra tekrardan kullanılmak üzere sıfırlanır.

Network 6:

Sistemin ilk değeri ilerde kullanılmak üzere reel sayıya çevrilerek kaydedilir. Sistemdeki gürültülerin ölü zamanın hesaplanmasına olan etkisini azaltmak amacıyla sistemin çıkışındaki değişimin %1'i bulunarak bir bant yaratılır. Daha sonra bir işaretçinin döngüye sokularak artırılır. Bu döngü ile sistem çıkışının belirlenen bandı geçtiği verinin adresi bulunmuş olur. Adresin bulunması ile beraber program M006 kısmına atlar.

Network 7:

İşaretçinin içeriği 3 sağa kaydırılmak sureti ile işaretçinin adres bilgisi alınarak reel sayıya çevrilir. Adresin kaçınıcı veri olduğunu anlamak için adres 2'ye bölünür. Geçen aralık sayısının bulunması için bu değerden 1 çıkarılır ve örnekleme zamanı ile çarpılarak ölü zaman bulunmuş olur.

```

L #FARK_REEL
L 6.320000e-001
*R
L #ILK_DEGER_REEL
+R
TRUNC
T #TEMP_ZSI
L P#0.0
T #ISARETCI2
M007: L DBW [#ISARETCI2]
L #TEMP_ZSI
>=I
JC M008
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M007
M008: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #T_O
L #OLU_ZAMAN
-R
T #T_O
S #Bitti
L P#0.0
T #ISARETCI2
JU M009
SON: L 0.000000e+000
T #K
T #T_O
T #OLU_ZAMAN
BE
M009: NOP 0

```

Network 8:

Sistemin çıkışının %63.2 si bulunarak ilk değere eklenir ve sistemin çıkışının bu değere ulaştığı zaman aranır. Bu işlemi gerçekleştirmek için daha önce de olduğu gibi bir işaretçi döngüye sokularak sistem çıkışının değeri belirlenen değeri aşana kadar işaretçinin değeri arttırılır. Aranan değer bulunduğu zaman program M008 kısmına atlayarak daha önce olduğu gibi işaretçinin adresinden zamana geçerek sistemin TO'sunu bulur. TO'nun bulunmasının ardından program M009'a atlar. Program daha önceden SON kısmına atlamışsa sistemle ilgili veriler sıfırlanarak fonksiyon bloğu kapatılır.

```

A #Bitti
JCN M010
L #Ts
NEGR
L #T_O
/R
EXP
T #A
L 1.000000e+000
TAK
-R
L #Ts
TAK
/R
T #Ti
L #OLU_ZAMAN
L #Ts
/R
T #d
LN
L #d
*R
EXP
T #Kc
L #d
L 1.000000e+000
+R
T #d1
LN
L #d1
*R
EXP
L #Kc
TAK
/R
T #Kc
L #K
/R
T #Kc
L 1.000000e+000
L #A
-R
L #Kc
TAK
/R
T #Kc
M010: NOP 0

```

Network 9:

Sistemin belirlenmesi işlemi bittiyse sistem daha önceden verilmiş formüllere göre kontrolör kazancını ve integral zaman sabitini bularak belirlenmiş bir veri alanına yazacaktır. Sistem tanıma işleminin bitmemiş olması durumunda program M010 kısmına atladıktan sonra başa dönecektir.

3.2 Çalışır Durumda Sistem Belirleme

Endüstriyel fırınlar gibi bazı süreçlerde, fırın sıcaklığının belirli aralıkların dışına çıkılmasına izin verilmez. Bu durumda sistemin belirlenmesi için izin verilen bir aralıkta sistemin giriş işareti basamak biçiminde artırılır veya azaltılır. Aşağıda bu işlemlerin nasıl yapıldığı PLC program örnekleri verilerek açıklanmıştır.

```
A #BASLA
FP #FP_BIT
JCN M001
R #V_ALINDI
R #Bitti
L P#0.0
T #ISARETCI
T #ISARETCI2
T #ISARETCISON
T #ISARETCISON2
L 0
T #ALINAN_V_S
L 0.000000e+000
T #SON_DEGER_REEL
L #Kontrolor_Cikis
ITD
DTR
T #Kontrolor_Cikis_Reel
S #V_ALINIYOR
```

Network 1:

Eğer sistem baştan çalıştırılmak isteniyorsa giriş aktif edildiği anda programın içinde kalan önceki verilerin hepsi temizlenir ve V_ALINIYOR biti setlenerek veri alınma işlemine başlandığı gösterilir. Bu şekilde sistemin içinde kalmış eski değerler ile yanlış bir işlem yapılmasının önüne geçilir. Çalışmakta olan sistemin o anki kontrol işareti alınarak reel sayıyı çevrilir ve bir veri alanına kaydedilir. Eğer sistem baştan çalıştırılmıyorsa program M001 kısmına atlar.

```
M001: A #BASLA
A #V_ALINIYOR
JCN M002
OPN #D_BLOK
L 1.000000e+002
L #Izin_Verilen_Aralik
-R
L 1.000000e+002
/R
L #Kontrolor_Cikis_Reel
*R
T #Indirimli_Cikis
RND
T #STEP_CIKIS
L #Bastirma_Orani
L #Filtre_Cikis_1
*R
T #Filtre_Cikis
L 1.000000e+000
L #Bastirma_Orani
-R
```

Network 2:

Başla butonuna basılı iken veri alınmaya başlanılmış ise verilerin kaydedileceği veri bloğu açıldıktan sonra kullanıcının belirlediği aralığa uygun olarak basamak girişin genliğinin belirlenir ve sisteme uygulanır. Sistemde oluşabilecek gürültülerin sistemin tanınmasına olan etkilerini azaltmak için sistemin çıkışından gelen veriler bir filtre ile kullanıcının belirlediği bir oranda bastırılır.

```

L #INPUT
ITD
DTR
*R
L #Filtre_Cikis
+R
T #Filtre_Cikis
T #Filtre_Cikis_1
RND
T DBW [#ISARETCI]
L #ALINAN_V_S
L 250
>=I
R #V_ALINIYOR
S #V_ALINDI
JC M002
L P#2.0
L #ISARETCI
+D
T #ISARETCI
L 1
L #ALINAN_V_S
+I
T #ALINAN_V_S

```

```

M002: A #BASLA
JC M003
R #V_ALINIYOR
R #V_ALINDI
L #Kontrolor_Cikis
T #STEP_CIKIS
M003: NOP 0

```

```

A #V_ALINDI
JCN SON
A #Bitti
JC M009
L 250
L 2
*D
L 8
-D
SLD 3
T #ISARETCISON2
L 250
L 2
*D
SLD 3
T #ISARETCISON

```

Network 2(Devam):

Filtreden geçirilmiř 250 veri bir iřaretçi yardımıyla toplanır. Verilerin veri blođuna sırasıyla kaydedilmesi için iřaretçinin deđeri her çevrimde 2 arttırılır. Bunun sebebi kaydedilecek integer formatındaki verinin 2 byte olmasıdır.

Program hedeflenen veri sayısına ulařtıđında yada veri toplama iřlemi yapılmadıđında M002 kısmına atlar.

Network 3:

Eđer sistem hala devredeyse program M003 kısmına atlar eđer devrede deđilse V_ALINIYOR ve V_Alındı bilgilerini temizler. Analog çıkıřa sisteme o andaki kontrol iřareti gönderilerek sisteme bir müdahalede bulunulmamıř olur.

Network 4:

Verilerin toplanması tamamlandıktan sonra programın veri tablosundaki gereken yerlere gitmesi için iřaretçilere gerekli deđerlerin verilmesi gerekir. Bu kısımda veri sayısı 2 ile çarpılıp kapladıđı alan byte cinsinden bulunduktan sonra bu sayı iřaretçi olarak kullanılmak üzere 3 sola kaydırılır. Bu řekilde iřaretçi veri tablosundaki son deđer iřaret eder. Aynı řekilde bir hesaplama da son deđerden 4 önceki deđere gidmek için yapılır ve iřaretçiye yüklenir. Eđer veri alınma iřlemi bitmemiřse program SON kısmına, eđer sistem tanıma iřlemi bittiyse program M009 kısmına atlar.

```

OPN #D_BLOK
L P#0.0
T #ISARETCI2
L DBW [#ISARETCI2]
T #ILK_DEGER
M004: L #ISARETCISON2
T #ISARETCI2
L DBW [#ISARETCI2]
ITD
DTR
L #SON_DEGER_REEL
+R
T #SON_DEGER_REEL
L P#2.0
L #ISARETCISON2
+D
T #ISARETCISON2
L #ISARETCISON2
L #ISARETCISON
>D
JCN M004
L #SON_DEGER_REEL
L 5.000000e+000
/R
RND
T #SON_DEGER
L #ILK_DEGER
TAK
-I
T #FARK
DTR
T #FARK_REEL
L #Kontrolor_Cikis_Reel
L #Indirimli_Cikis
-R
L #FARK_REEL
TAK
/R
T #K
L P#0.0
T #ISARETCISON
T #ISARETCISON2

```

Network 5:

Verilerin saklandığı veri bloğu açıldıktan sonra bir işaretçiye sıfır değeri atanarak, bu işaretçinin yardımıyla veri bloğundan sistemin ilk değeri alınır. Sonrasında daha önceden bulunan, sistemin sondan 4 önceki değerinin olduğu adresi taşıyan işaretçi kullanılarak program döngüye sokulur. Bu döngü ile veri bloğunun sonuna kadar olan 5 değer reel sayıya çevrilerek toplandıktan sonra bu değer 5'e bölünerek sistemin son değeri olarak kaydedilir. Bu şekilde sistemdeki gürültülerin etkisiyle son değerın yanlış hesaplanma ihtimali ortalama alınarak azaltılmış olur.

Sistemin son değerinden ilk değeri çıkarılarak sistemdeki değişim bulunmuş olur. Bu değer daha sonra ondalıklı işlemlerin yapımına uygun olması için reel sayıya çevrilir. Sistemin çıkışındaki değişim girişindeki değişime bölünerek sistemin kazancı bulunduğundan sonra kullanılan işaretçiler tekrardan kullanılmak üzere sıfırlanır.

```
L #ILK_DEGER
ITD
DTR
T #ILK_DEGER_REEL
L #FARK_REEL
L 1.000000e-002
*R
L #ILK_DEGER_REEL
TAK
-R
RND
T #UST_SINIR
L P#0.0
T #ISARETCI2
M005: OPN #D_BLOK
L DBW [#ISARETCI2]
L #UST_SINIR
<I
JC M006
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M005
```

```
M006: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #OLU_ZAMAN
```

Network 6:

Sistemin ilk değeri ilerde kullanılmak üzere reel sayıya çevrilerek kaydedilir. Sistemdeki gürültülerin ölü zamanın hesaplanmasına olan etkisini azaltmak amacıyla sistemin çıkışındaki değişimin %1'i bulunarak bir bant yaratılır. Daha sonra bir işaretçinin döngüye sokularak artırılır. Bu döngü ile sistem çıkışının belirlenen bandın altına indiği verinin adresi bulunmuş olur. Adresin bulunması ile beraber program M006 kısmına atlar.

Network 7:

İşaretçinin içeriği 3 sağa kaydırılmak sureti ile işaretçinin adres bilgisi alınarak real sayıya çevrilir. Adresin kaçınıcı veri olduğunu anlamak için adres 2'ye bölünür. Geçen aralık sayısının bulunması için bu değerden 1 çıkarılır ve örnekleme zamanı ile çarpılarak ölü zaman bulunmuş olur.

```

L #FARK_REEL
L 6.320000e-001
*R
L #ILK_DEGER_REEL
TAK
-R
TRUNC
T #TEMP_ZSI
L P#0.0
T #ISARETCI2
M007: OPN #D_BLOK
L DBW [#ISARETCI2]
L #TEMP_ZSI
<=I
JC M008
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M007
M008: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #T_O
L #OLU_ZAMAN
-R
T #T_O
S #Bitti
L P#0.0
T #ISARETCI2
JU M009
SON: L 0.000000e+000
T #K
T #T_O
T #OLU_ZAMAN
BE
M009: NOP 0

```

Network 8:

Sistemin çıkışının %63.2 si bulunarak ilk değerdne çıkarılır ve sistemin bu değere indiği zaman aranır. Bu işlemi gerçekleştirmek için daha önce de olduğu gibi bir işaretçi döngüye sokularak sistem çıkışının değeri belirlenen değerin altına gelene kadar işaretçinin değeri artırılır. Aranılan değer bulunduğu zaman program M008 kısmına atlayarak daha önce olduğu gibi işaretçinin adresinden zamana geçerek sistemin TO'sunu bulur. TO'nun bulunmasının ardından program M009'a atlar. Program daha önceden SON kısmına atlamışsa sistemle ilgili veriler sıfırlanarak program bloğu kapatılır.


```

A #Bitti
JCN M010
L #Ts
NEGR
L #T_O
/R
EXP
T #A
L 1.000000e+000
TAK
-R
L #Ts
TAK
/R
T #Ti
L #OLU_ZAMAN
L #Ts
/R
T #d
LN
L #d
*R
EXP
T #Kc
L #d
L 1.000000e+000
+R
T #d1
LN
L #d1
*R
EXP
L #Kc
TAK
/R
T #Kc
L #K
/R
T #Kc
L 1.000000e+000
L #A
-R
L #Kc
TAK
/R
T #Kc
M010: NOP 0

```

Network 9:

Sistem belirlenmesi işlemi bittiyse sistem daha önceden verilmiş formüllere göre kontrolör kazancını ve integral zaman sabitini bularak belirlenmiş bir veri alanına yazacaktır. Sistem tanıma işleminin bitmemiş olması durumunda program M010 kısmına atladıktan sonra başa dönecektir.

Bazı durumlarda, ısınma ve soğuma dinamikleri farklı bir fırının soğuma durumuna ilişkin modeline gerek duyulabilir. Bu tür durumlar için kontrol işareti izin verilen belirli bir oranda basamak biçiminde azaltılır ve bu koşullar altında sistem belirlenmeye çalışılır. Aşağıda bu işlemlerin nasıl yapıldığı PLC program örnekleri verilerek açıklanmıştır.

```
A #BASLA
FP #FP_BIT
JCN M001
R #V_ALINDI
R #Bitti
L P#0.0
T #ISARETCI
T #ISARETCI2
T #ISARETCISON
T #ISARETCISON2
L 0
T #ALINAN_V_S
L 0.000000e+000
T #SON_DEGER_REEL
L #Kontrolor_Cikis
ITD
DTR
T #Kontrolor_Cikis_Reel
S #V_ALINIYOR
```

Network 1:

Eğer sistem baştan çalıştırılmak isteniyorsa giriş aktif edildiği anda programın içinde kalan önceki verilerin hepsi temizlenir ve V_ALINIYOR biti setlenerek veri alınma işlemine başlandığı gösterilir. Bu şekilde sistemin içinde kalmış eski değerler ile yanlış bir işlem yapılmasının önüne geçilir. Çalışmakta olan sistemin o anki kontrol işareti alınarak reel sayıyı çevrilir ve bir veri alanına kaydedilir. Eğer sistem baştan çalıştırılmıyorsa program M001 kısmına atlar.

```
M001: A #BASLA
A #V_ALINIYOR
JCN M002
OPN #D_BLOK
L 1.000000e+002
L #Izin_Verilen_Aralik
+R
L 1.000000e+002
/R
L #Kontrolor_Cikis_Reel
*R
T #Indirimli_Cikis
RND
T #STEP_CIKIS
L #Bastirma_Orani
L #Filtre_Cikis_1
*R
T #Filtre_Cikis
L 1.000000e+000
L #Bastirma_Orani
-R
```

Network 2:

Başla butonuna basılı iken veri alınmaya başlanılmış ise verilerin kaydedileceği veri bloğu açıldıktan sonra kullanıcının belirlediği aralığa uygun olarak basamak girişin genliğinin belirlenir ve sisteme uygulanır. Sistemde oluşabilecek gürültülerin sistemin tanınmasına olan etkilerini azaltmak için sistemin çıkışından gelen veriler bir filtre ile kullanıcının belirlediği bir oranda bastırılır.

```

L #INPUT
ITD
DTR
*R
L #Filtre_Cikis
+R
T #Filtre_Cikis
T #Filtre_Cikis_1
RND
T DBW [#ISARETCI]
L #ALINAN_V_S
L 250
>=I
R #V_ALINIYOR
S #V_ALINDI
JC M002
L P#2.0
L #ISARETCI
+D
T #ISARETCI
L 1
L #ALINAN_V_S
+I
T #ALINAN_V_S

```

```

M002: A #BASLA
JC M003
R #V_ALINIYOR
R #V_ALINDI
L #Kontrolor_Cikis
T #STEP_CIKIS
M003: NOP 0

```

```

A #V_ALINDI
JCN SON
A #Bitti
JC M009
L 250
L 2
*D
L 8
-D
SLD 3
T #ISARETCISON2
L 250
L 2
*D
SLD 3
T #ISARETCISON

```

Network 2(Devam):

Filtreden geçirilmiř 250 veri bir iřaretçi yardımıyla toplanır. Verilerin veri blođuna sırasıyla kaydedilmesi için iřaretçinin deđeri her çevrimde 2 arttırılır. Bunun sebebi kaydedilecek integer formatındaki verinin 2 byte olmasıdır.

Program hedeflenen veri sayısına ulařtıđında yada veri toplama iřlemi yapılmadıđında M002 kısmına atlar.

Network 3:

Eđer sistem hala devredeyse program M003 kısmına atlar eđer devrede deđilse V_ALINIYOR ve V_Alındı bilgilerini temizler. Sistem devreden çıkmıřsa analog çıkıřa sisteme o andaki kontrol iřareti gönderilerek sisteme bir müdahalede bulunulmamıř olur.

Network 4:

Verilerin toplanması tamamlandıktan sonra programın veri tablosundaki gereken yerlere gitmesi için iřaretçilere gerekli deđerlerin verilmesi gerekir. Bu kısımda veri sayısı 2 ile çarpılıp kapladıđı alan byte cinsinden bulunduktan sonra bu sayı iřaretçi olarak kullanılmak üzere 3 sola kaydırılır. Bu řekilde iřaretçi veri tablosundaki son deđer iřaret eder. Aynı řekilde bir hesaplama da son deđerden 4 önceki deđere gidmek için yapılır ve iřaretçiye yüklenir. Eđer veri alınma iřlemi bitmemiře program SON kısmına, eđer sistem tanıma iřlemi bittiyse program M009 kısmına atlar.

```

OPN #D_BLOK
L P#0.0
T #ISARETCI2
L DBW [#ISARETCI2]
T #ILK_DEGER
M004: L #ISARETCISON2
T #ISARETCI2
L DBW [#ISARETCI2]
ITD
DTR
L #SON_DEGER_REEL
+R
T #SON_DEGER_REEL
L P#2.0
L #ISARETCISON2
+D
T #ISARETCISON2
L #ISARETCISON2
L #ISARETCISON
>D
JCN M004
L #SON_DEGER_REEL
L 5.000000e+000
/R
RND
T #SON_DEGER
L #ILK_DEGER
-I
T #FARK
DTR
T #FARK_REEL
L #Indirimli_Cikis
L #Kontrolor_Cikis_Reel
-R
L #FARK_REEL
TAK
/R
T #K
L P#0.0
T #ISARETCISON
T #ISARETCISON2

```

Network 5:

Verilerin saklandığı veri bloğu açıldıktan sonra bir işaretçiye sıfır değeri atanarak, bu işaretçinin yardımıyla veri bloğundan sistemin ilk değeri alınır. Sonrasında daha önceden bulunan, sistemin sondan 4 önceki değerinin olduğu adresi taşıyan işaretçi kullanılarak program döngüye sokulur. Bu döngü ile veri bloğunun sonuna kadar olan 5 değer reel sayıya çevrilerek toplandıktan sonra bu değer 5'e bölünerek sistemin son değeri olarak kaydedilir. Bu şekilde sistemdeki gürültülerin etkisiyle son değerın yanlış hesaplanma ihtimali ortalama alınarak azaltılmış olur.

Sistemin son değerinden ilk değeri çıkarılarak sistemdeki değişim bulunmuş olur. Bu değer daha sonra ondalıklı işlemlerin yapımına uygun olması için reel sayıya çevrilir. Sistemin çıkışındaki değişim girişindeki değişime bölünerek sistemin kazancı bulunduğundan sonra kullanılan işaretçiler tekrardan kullanılmak üzere sıfırlanır.

```

L #ILK_DEGER
ITD
DTR
T #ILK_DEGER_REEL
L #FARK_REEL
L 1.000000e-002
*R
L #ILK_DEGER_REEL
TAK
+R
RND
T #UST_SINIR
L P#0.0
T #ISARETCI2
M005: OPN #D_BLOK
L DBW [#ISARETCI2]
L #UST_SINIR
>I
JC M006
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M005

```

```

M006: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #OLU_ZAMAN

```

Network 6:

Sistemin ilk değeri ilerde kullanılmak üzere reel sayıya çevrilerek kaydedilir. Sistemdeki gürültülerin ölü zamanın hesaplanmasına olan etkisini azaltmak amacıyla sistemin çıkışındaki değişimin %1'i bulunarak bir bant yaratılır. Daha sonra bir işaretçinin döngüye sokularak artırılır. Bu döngü ile sistem çıkışının belirlenen bandın üstüne çıktığı verinin adresi bulunmuş olur. Adresin bulunması ile beraber program M006 kısmına atlar.

Network 7:

İşaretçinin içeriği 3 sağa kaydırılmak sureti ile işaretçinin adres bilgisi alınarak reel sayıya çevrilir. Adresin kaçınıcı veri olduğunu anlamak için adres 2'ye bölünür. Geçen aralık sayısının bulunması için bu değerden 1 çıkarılır ve örnekleme zamanı ile çarpılarak ölü zaman bulunmuş olur.

```

L #FARK_REEL
L 6.320000e-001
*R
L #ILK_DEGER_REEL
TAK
+R
TRUNC
T #TEMP_ZSI
L P#0.0
T #ISARETCI2
M007: OPN #D_BLOK
L DBW [#ISARETCI2]
L #TEMP_ZSI
>=I
JC M008
L P#2.0
L #ISARETCI2
+D
T #ISARETCI2
JU M007
M008: L #ISARETCI2
SRD 3
DTR
L 2.000000e+000
/R
L 1.000000e+000
-R
L #Ts
*R
T #T_O
L #OLU_ZAMAN
-R
T #T_O
S #Bitti
L P#0.0
T #ISARETCI2
JU M009
SON: L 0.000000e+000
T #K
T #T_O
T #OLU_ZAMAN
BE
M009: NOP 0

```

Network 8:

Sistemin çıkışının %63.2 si bulunarak ilk değere eklenir ve sistemin bu değere ulaştığı zaman aranır. Bu işlemi gerçekleştirmek için daha önce de olduğu gibi bir işaretçi döngüye sokularak sistem çıkışının değeri belirlenen değer altına gelene kadar işaretçinin değeri artırılır. Aranılan değer bulunduğu zaman program M008 kısmına atlayarak daha önce olduğu gibi işaretçinin adresinden zamana geçerek sistemin TO'sunu bulur. TO'nun bulunmasının ardından program M009'a atlar. Program daha önceden SON kısmına atlamışsa sistemle ilgili veriler sıfırlanarak program bloğu kapatılır.

```

A #Bitti
JCN M010
L #Ts
NEGR
L #T_O
/R
EXP
T #A
L 1.000000e+000
TAK
-R
L #Ts
TAK
/R
T #Ti
L #OLU_ZAMAN
L #Ts
/R
T #d
LN
L #d
*R
EXP
T #Kc
L #d
L 1.000000e+000
+R
T #d1
LN
L #d1
*R
EXP
L #Kc
TAK
/R
T #Kc
L #K
/R
T #Kc
L 1.000000e+000
L #A
-R
L #Kc
TAK
/R
T #Kc
M010: NOP 0

```

Network 9:

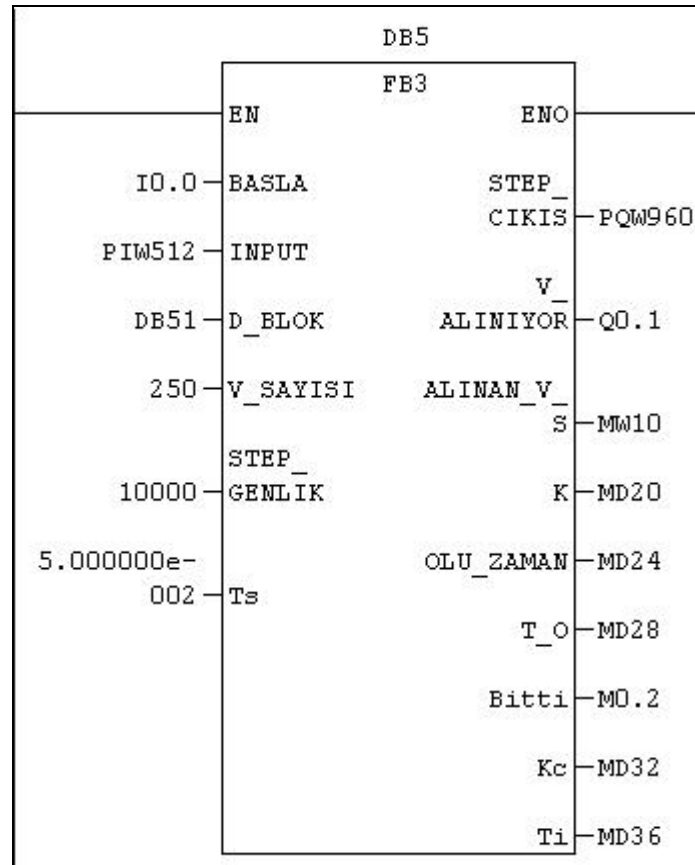
Sistem belirlenmesi işlemi bittiyse sistem daha önceden verilmiş formüllere göre kontrolör kazancını ve integral zaman sabitini bularak belirlenmiş bir veri alanına yazacaktır. Sistem tanıma işleminin bitmemiş olması durumunda program M010 kısmına atladıktan sonra başa dönecektir.

4. KONTROL SİSTEMİNİN UYGULANMASI

Sistemin belirlenmesi ve tasarlanan kontrolörün sisteme uygulanması için belirli bir örnekleme zamanında çalışan bir kesme alt programına ihtiyaç duyulur. Bu projede kullanılan S7-300 tipi PLC'ler de bulunan OB35 bloğu verilen örnekleme zamanı için istenilen kesmeli çalışmayı sağlayacaktır. Sistemi tanıyan fonksiyon bloğu ve kontrolör bloğu OB35 bloğunun içinde aşağıdaki şekilde çağırılarak programın sistemi belirlemesi ve elde edilen kontrol işaretini sisteme uygulaması mümkün olur.

Sistemi kontrol edecek olan kontrolörün yeni bir blokta yazılması yerine, sanayide yaygın olarak bulunan Simatic FB41 hazır PID bloğu kullanılmış, bu şekilde bu alandaki kullanıcıların alışkın olduğu bir kontrol yapısı gerçekleştirilmiştir.

4.1 Başlangıç Durumundan Sistemi Belirleyen Fonksiyon Bloğu ve Kullanımı

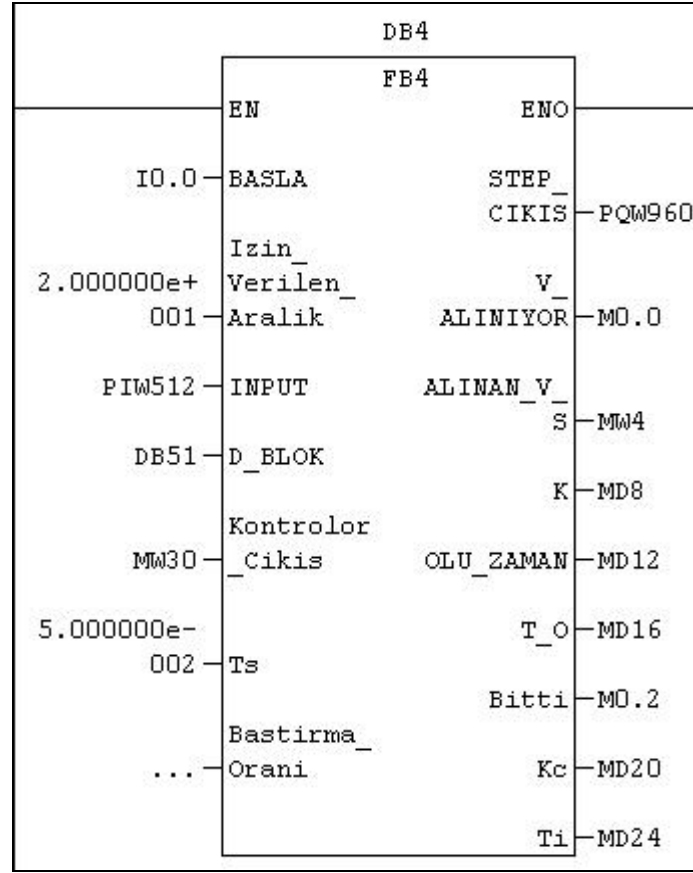


Şekil 4.1Başlangıç durumundan sistemi belirleyen fonksiyon bloğu

Sistemin baştan belirlenmesi için OB35 kesmeli bloğunun içinde FB3 (Baştan Belirleme Fonksiyonu) çağırılır. Bloğun üstündeki yere bloğun kullanacağı Data Block(DB5) yazılır. Simatic Manager'da her fonksiyon bloğunun bir data bloğa bağlanması zorunludur. Daha sonra girişlere kullanıcının belirlediği veri sayısı, basamak girişin genliği ve örnekleme zamanı değerleri girilir. Sistemden alınan veri INPUT kısmına analog adres olarak yazılır (PIW512). Sisteme gönderilecek veri de STEP_CIKIS kısmına analog adres olarak yazılır(PQW960). Verilerin saklanacağı blok D_BLOCK kısmına yazıldıktan sonra çıkışlara boyutlarına göre uygun adresler verilir. Adresler verilirken dikkat edilmesi gereken husus bellek alanlarının çakışmamasıdır. Integer formatındaki veriler bir MW alanına yazılabilir. Bu alanlar 2Byte'lık yer kaplamaktadır. Real formatındaki veriler ise MD alanına yazılabilir. Bu

alanlar 4Byte'lık yer kaplamaktadır. Bool formatındaki veriler içinde 1bitlik bir M adresi yeterli olacaktır.

4.2 Çalışır Durumda Sistemi Belirleyen Fonksiyon Bloğu ve Kullanımı

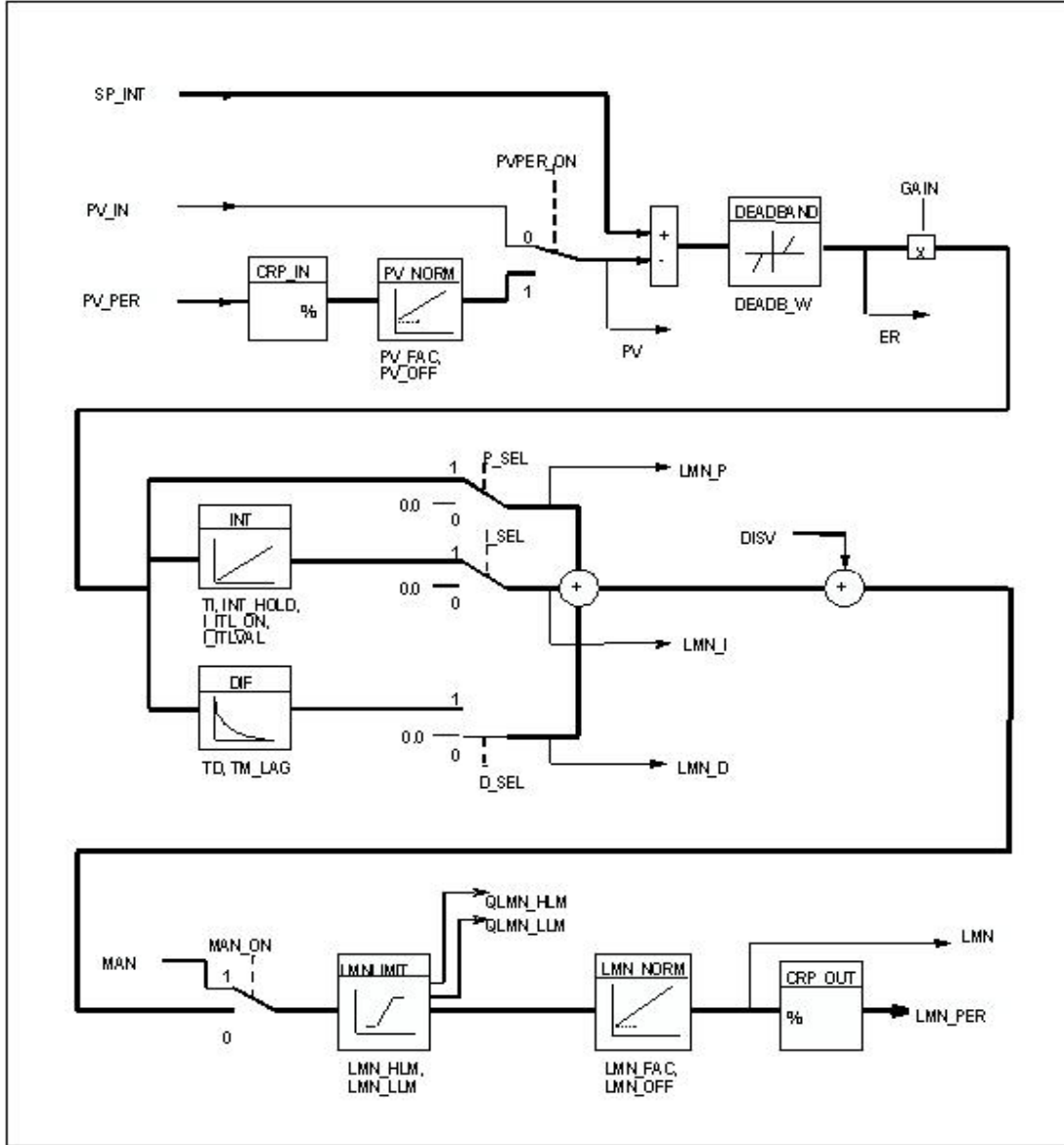


Şekil 4.2 Çalışan sistemi tanıma fonksiyon bloğu

Sistemin belirli bir kontrol işareti ile çalışırken, azaltılan ya da arttırılan bir kontrol işareti ile belirlenmesi için OB35 kesmeli çalışma bloğunun içinde sistem tanıma fonksiyonu (FB4) çağırıldıktan sonra Bölüm 4.1’de yapılan işlemler aynen tekrarlanır. Çalışan sistemin belirlenmesi için yapılması gereken tek farklı işlem, sistemi, o sırada uygulanan kontrol işaretinin adresinin “Kontrolor_Cıkış” kısmına girilmesidir. Eğer bir filtre kullanılmak istenilirse filtrenin bastırma oranı giriş olarak girilebilir. Girilmemesi durumunda bu değer sıfır olarak kabul edilecektir. Program uygulayacağı basamak genliği ve toplayacağı veri sayısını otomatik olarak belirlediğinden çalışan sistemin belirlenmesi işleminde daha önceden kullanılan basamak girişin genliği ve toplanacak veri sayısı değerlerine gerek duyulmamıştır.

4.3 FB41 PID Fonksiyon Bloğunun Kullanımı

Sistem belirlendikten sonra bulunan katsayıların bir PID bloğuna yüklenerek sisteme uygulanması gerekmektedir. Bu amaçla “Simatic Manager” programının standart kütüphanesinde bulunan “FB41 Continuous Control” bloğu kullanılacaktır. Bu bloğun işlevsel uygulama şeması Şekil 4.3’te gösterilmiştir.



Şekil 4.3 FB41 işlevsel uygulama şeması

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

SP_INT: Bu büyüklük -100.0 ile 100.0 arasında değişen sistemin yüzde olarak karşılığıdır. Referans değeri buraya yüzde olarak girilir.

PVPER_ON: Sistemden gelen geri besleme direk olarak analog girişten alınacaksa bu bit aktive edilir.

P_SEL: Aktif olması durumunda kontrolörün kazanç ayağından gelen kontrol işaretinin tüm kontrol işaretine katılmasını sağlar.

I_SEL: Aktif olması durumunda kontrolörün integral ayağından gelen kontrol işaretinin tüm kontrol işaretine katılmasını sağlar.

MAN_ON: Aktifken önceden belirlenen bir kontrol işareti sisteme otomatik olarak verilir. Sürekli bir kontrol işlemi yapılması için bu bit aktif olmamalıdır.

COM_RST: Aktifken kontrolörün içindeki tüm değerleri resetler.

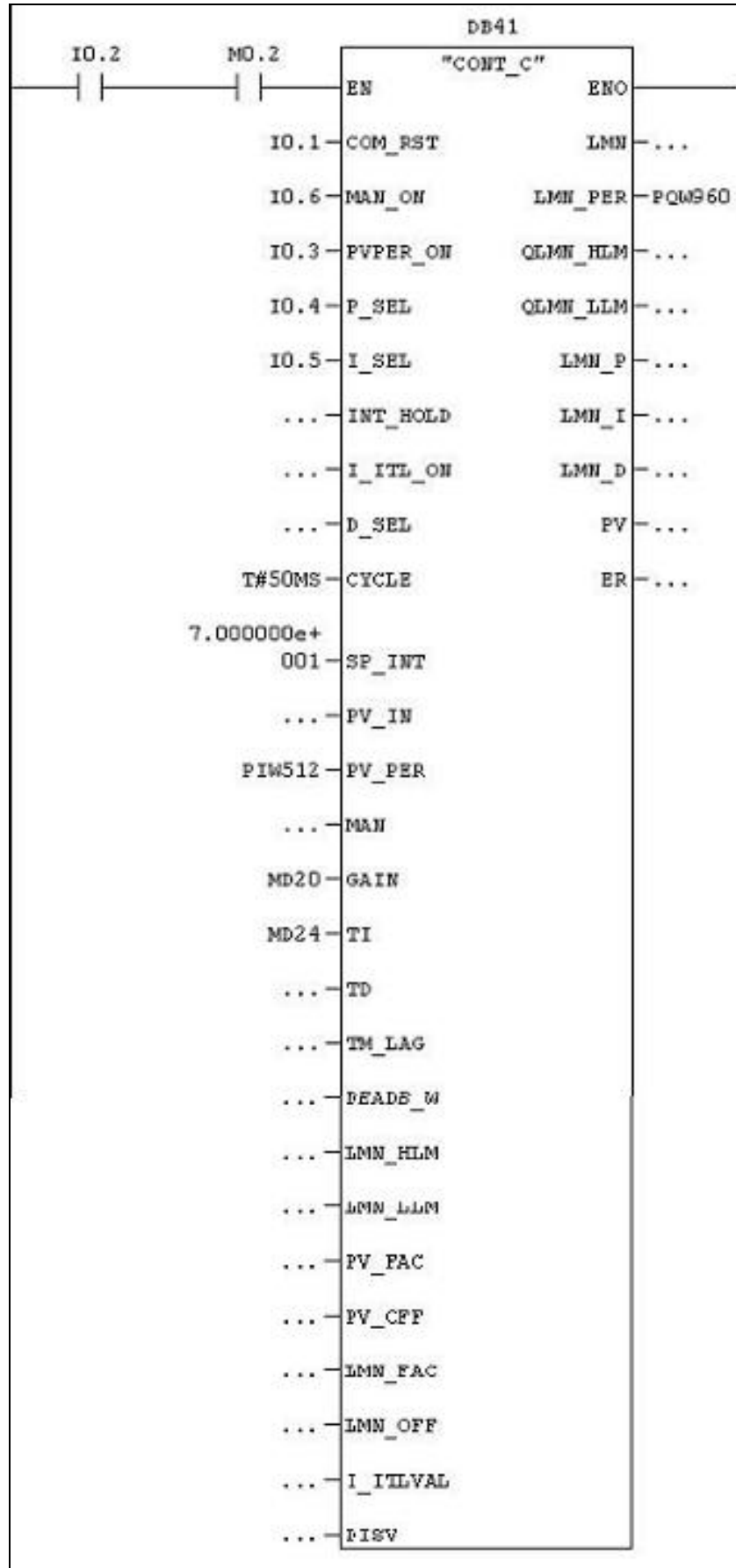
CYCLE: Kontrolörün örnekleme zamanı bu alana T#..MS şeklinde girilir.

PV_PER: Sistemden gelen geribesleme bilgisinin olduğu analog adres bu alana yazılır.

GAIN: Kontrolör kazancı buraya real sayı olarak yada real formatında bir bellek alanı olarak yazılır.

TI: İntegral 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: Kontrol işaretinin sisteme uygulanacağı analog adres bu alana yazılır.



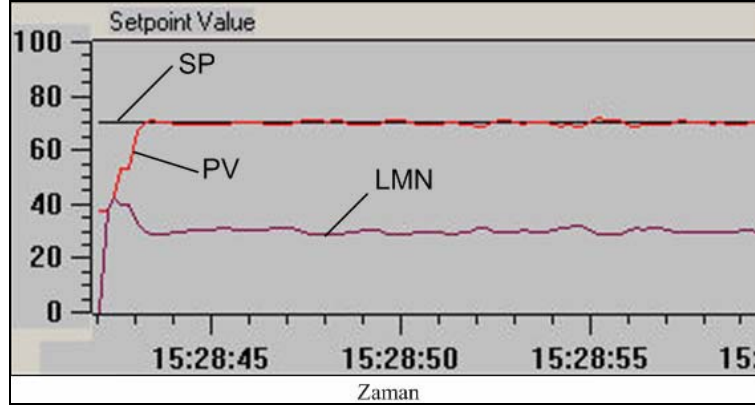
Şekil 4.4 FB41 PID kontrolör fonksiyon bloğu

Bütün bloklar hazırlandıktan sonra tüm bu bloklar PLC'ye eksiksiz olarak yüklenir. Yüklenmemesi durumunda PLC "STOP" konumuna geçecektir. Yapılması gereken önemli bir ayrıntı da OB35 kesmeli bloğunun çağırılma süresinin CPU'ya yüklenmesidir. Bunun için "Hardware" ayarları CPU özelliklerine bölümünde "Cyclic Interrupts" başlığı altındaki OB35 bloğunun çağırılma süresi istenilen süreye göre ayarlanır. Bu değerle kontrolörün örnekleme zamanının aynı olması çok önemlidir.

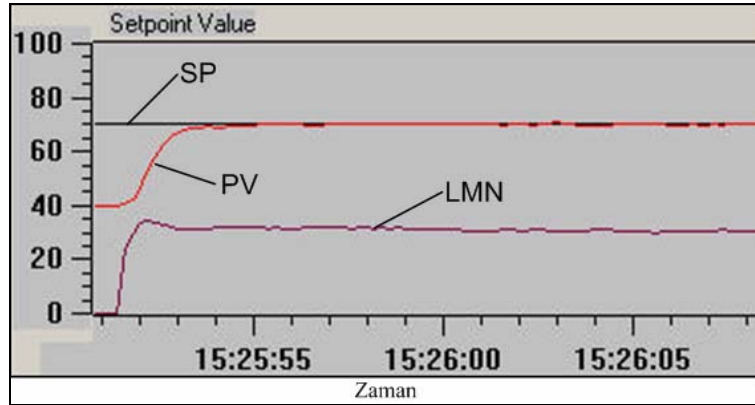
5. UYGULAMA SONUÇLARI VE TARTIŞMA

5.1 Başlangıç Durumundan Sistem Belirleme ve Kontrol

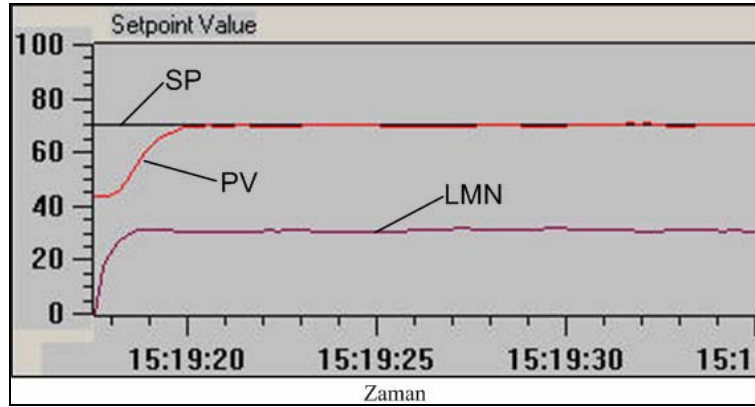
Sistemin ilk değerinden basamak yanıtı ile belirlenmesi ve kontrolü ile ilgili sonuçlar grafikler halinde aşağıda verilmiştir. Programın farklı dinamiklere sahip sistemlere uygulanabilir olduğunu göstermek amacıyla sistemdeki ölçüm cihazının konumu ve sistemin dinamiğini etkileyen kapağın açısı değiştirilerek farklı sistemler elde edilmiştir. Buradaki grafiklerde SP istenen değer, PV sistem çıkışı ve LMN kontrol işaretine ilişkin simgelerdir.



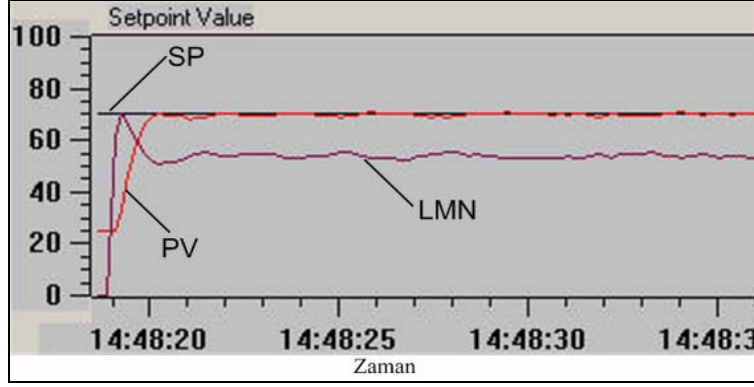
Şekil 5.1 Başlangıç durumundan sistem tanıma (Birinci konum 35 derece)



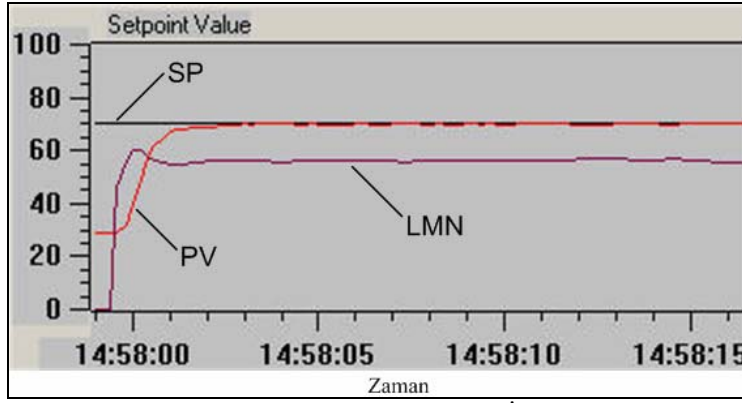
Şekil 5.2 Başlangıç durumundan sistem tanıma (İkinci konum 35 derece)



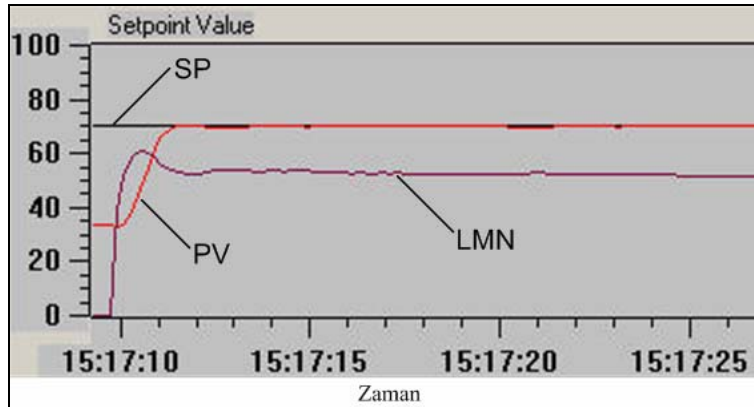
Şekil 5.3 Başlangıç durumundan sistem tanıma (Üçüncü konum 35 derece)



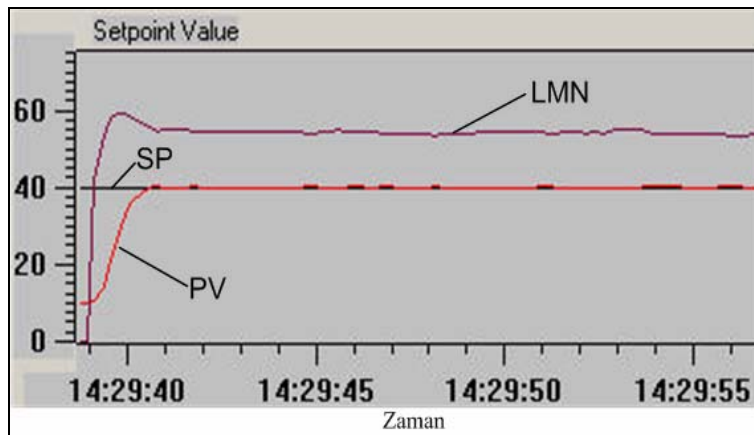
Şekil 5.4 Başlangıç durumundan sistem tanıma (Birinci konum 55 derece)



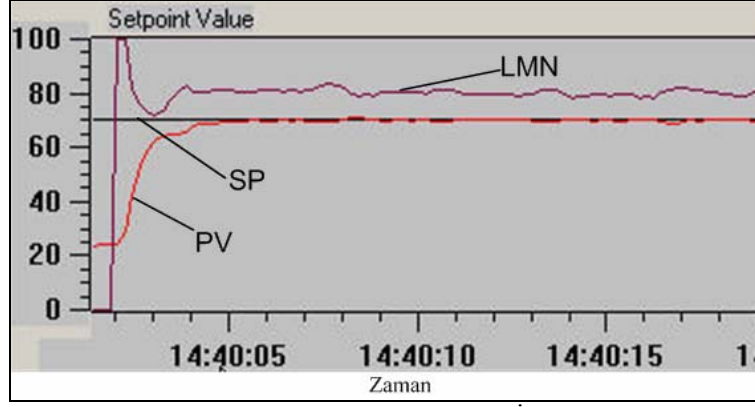
Şekil 5.5 Başlangıç durumundan sistem tanıma (İkinci konum 55 derece)



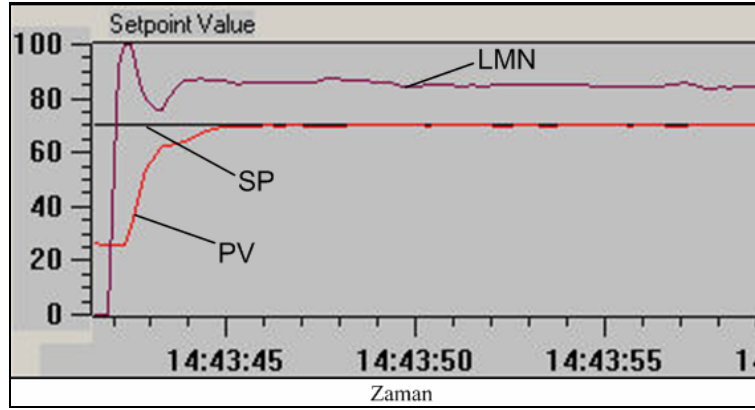
Şekil 5.6 Başlangıç durumundan sistem tanıma (Üçüncü konum 55 derece)



Şekil 5.7 Başlangıç durumundan sistem tanıma (Birinci konum 75 derece)



Şekil 5.8 Başlangıç durumundan sistem tanıma (İkinci konum 75 derece)



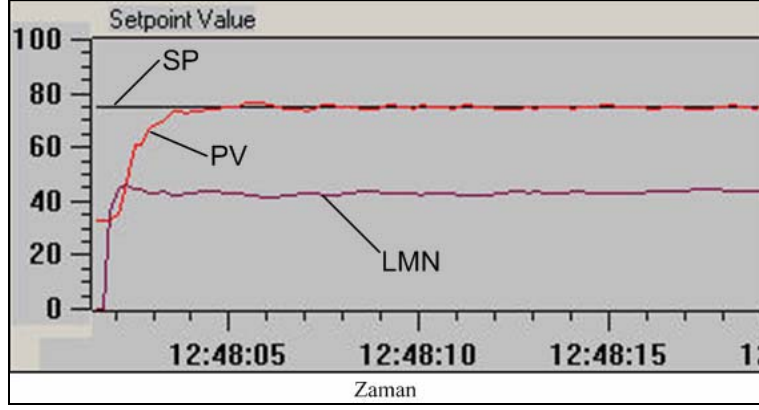
Şekil 5.9 Başlangıç durumundan sistem tanıma (Üçüncü konum 75 derece)

Yukarıdaki şekillerden anlaşılacağı gibi bütün kontrolörler şu ortak özellikleri sağlamıştır.

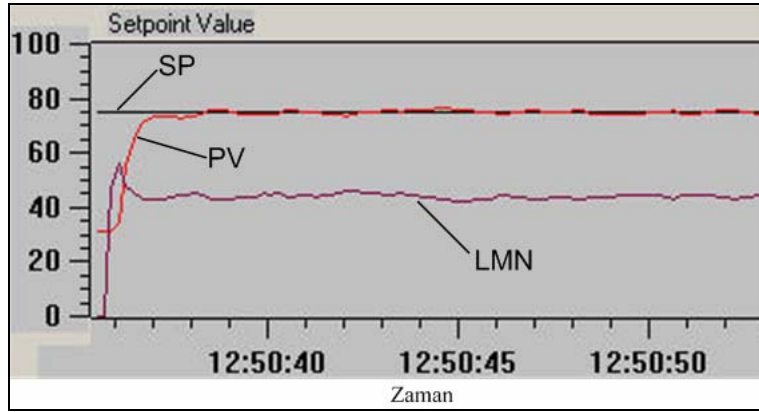
- Bütün sistemler istenilen referansı takip etmiştir
- PI kontrolör kullanılmasının sonucu olarak hiçbir sistemde sürekli hal hatası oluşmamıştır.
- Hiçbir sistemde aşım oluşmamıştır.
- Tüm sistemlerin yerleşme zamanı kabul edilebilir düzeydedir.
- Sistemler bozucu etkilere karşı oldukça dayanıklıdır.

5.2 Çalışan Sistemin Belirlenmesi ve Kontrolü

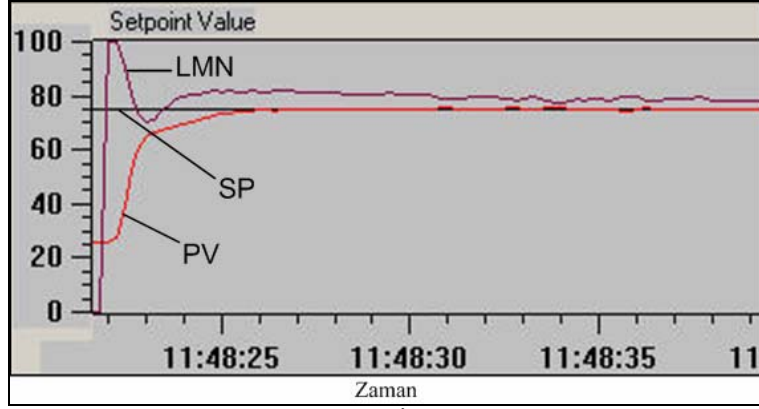
Çalışan sistemin izin verilen bant aralığında artırılarak oluşturulan bir basamak girişe olan yanıtı ile belirlenmesi ve kontrolü ile ilgili sonuçlar grafikler halinde aşağıda verilmiştir. Programın farklı dinamiklere sahip sistemlere de uygulanabilir olduğunu göstermek amacıyla sistemdeki ölçüm cihazının konumu ve sistemin dinamiğini etkileyen kapağın açısı değiştirilerek farklı sonuçlar elde edilmiştir. Sistemin belirlenmesinde ve kontrol edilmesine olan etkisini gözlemlemek amacıyla farklı izin bantları denenmiştir.



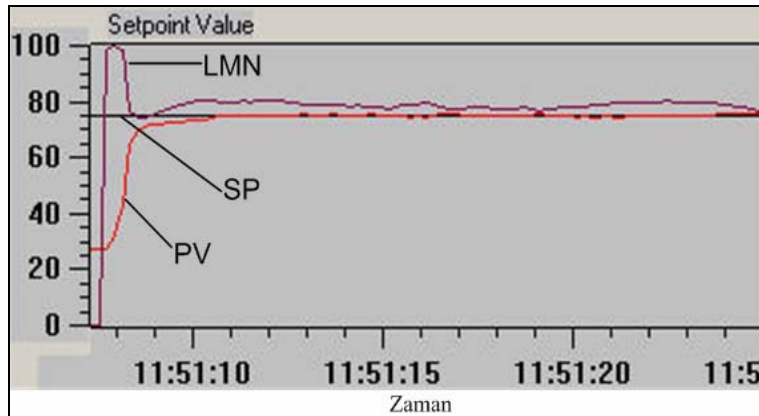
Şekil 5.10 Çalıřan Sistemin Belirlenmesi (Birinci konum 45 derece %25 izin bandı)



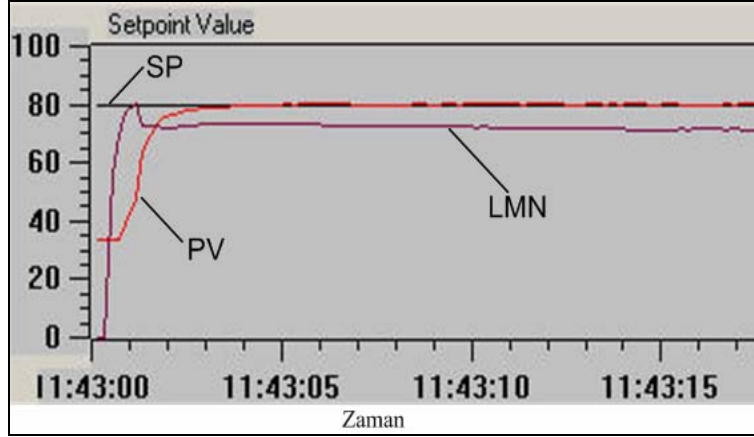
Şekil 5.11 Çalıřan Sistemin Belirlenmesi (Birinci konum 45 derece %50 izin bandı)



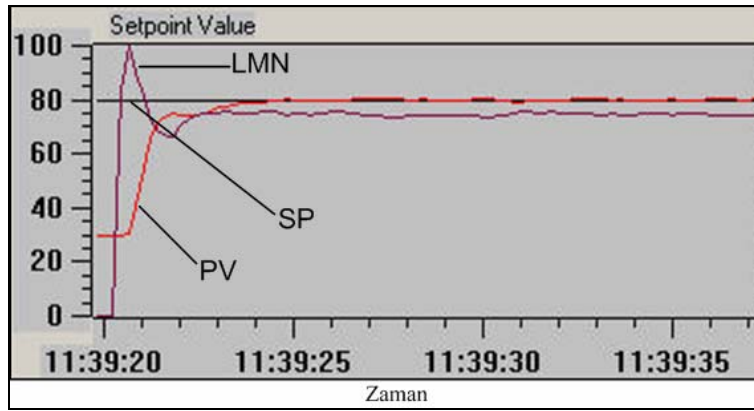
Şekil 5.12 Çalıřan Sistemin Belirlenmesi (İkinci konum 70 derece %25 izin bandı)



Şekil 5.13 Çalıřan Sistemin Belirlenmesi (İkinci konum 70 derece %50 izin bandı)



Şekil 5.14 Çalışan Sistemin Belirlenmesi (Üçüncü konum 55 derece %25 izin bandı)



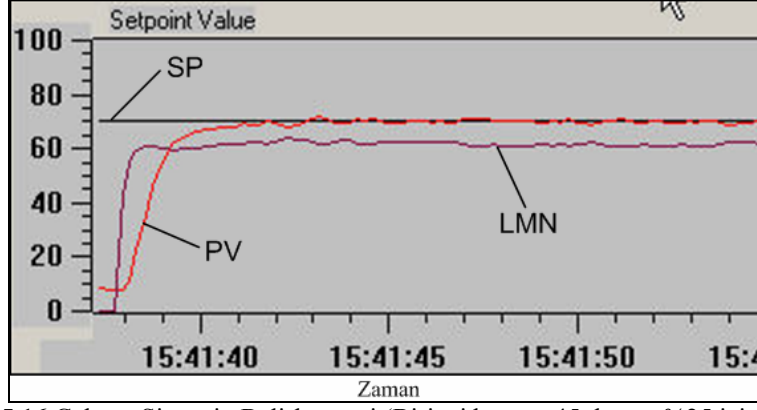
Şekil 5.15 Çalışan Sistemin Belirlenmesi (Üçüncü konum 55 derece %50 izin bandı)

Yukarıdaki şekillerden anlaşılacağı gibi bütün kontrolörler şu ortak özellikleri sağlamıştır.

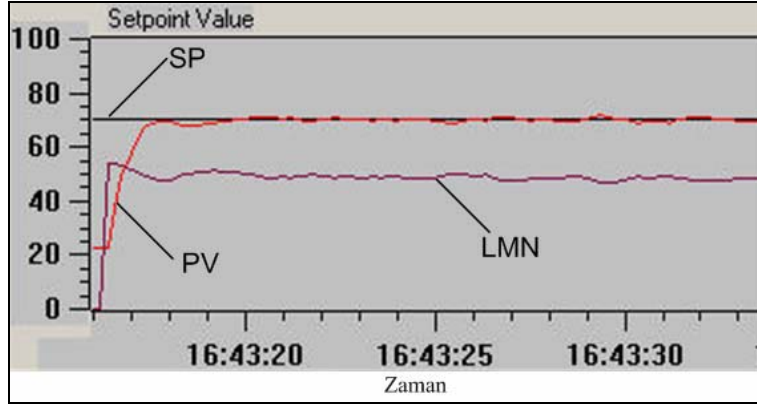
- Bütün sistemler istenilen referansı takip etmiştir
- PI kontrolör kullanılmasının sonucu olarak hiçbir sistemde sürekli hal hatası oluşmamıştır.
- Hiçbir sistemde aşım oluşmamıştır.
- Tüm sistemlerin yerleşme zamanı kabul edilebilir düzeydedir.
- Sistemler bozucu etkilere karşı oldukça dayanıklıdır.

Grafikler incelendiğinde görülebileceği gibi izin bandı sistemin tanınmasında etkili olmaktadır. Görülebileceği gibi izin bandı arttığı zaman yerleşme zamanı daha kısa olan kontrolörler elde edilmiştir. Bunun sebebi sistemdeki büyük gürültülerdir. Sisteme uygulanan basamak girişin genliği artırıldığı zaman, sistemdeki gürültülerin etkisinin giriş işaretine oranı azaldığı için sistem daha iyi tanınmakta ve daha iyi bir kontrolör tasarlanmaktadır.

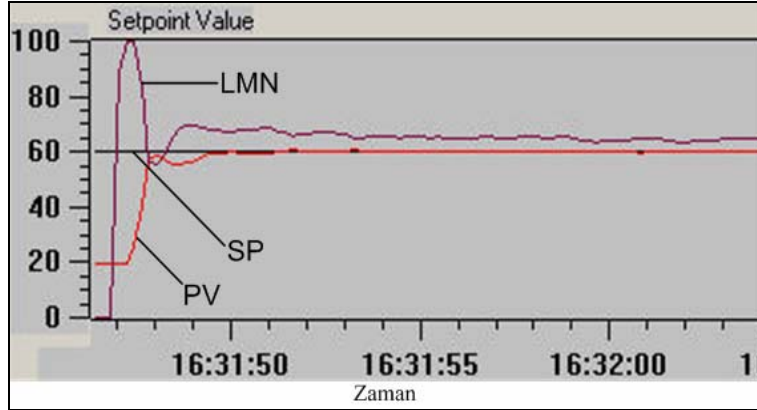
Çalışan sistemin izin verilen bant aralığında azaltılarak oluşturulan bir basamak girişe olan yanıtı ile belirlenmesi ve kontrolü ile ilgili sonuçlar grafikler halinde aşağıda verilmiştir. Programın farklı dinamiklere sahip sistemlere de uygulanabilir olduğunu göstermek amacıyla sistemdeki ölçüm cihazının konumu ve sistemin dinamiğini etkileyen kapağın açısı değiştirilerek farklı sonuçlar elde edilmiştir. Sistemin tanınmasına ve kontrol edilmesine olan etkisini gözlemlemek açısından farklı izin bantları denenmiştir.



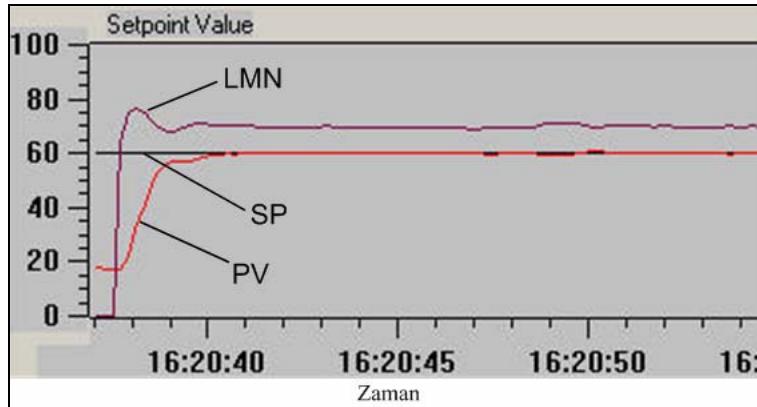
Şekil 5.16 Çalıřan Sistemin Belirlenmesi (Birinci konum 45 derece %25 izin bandı)



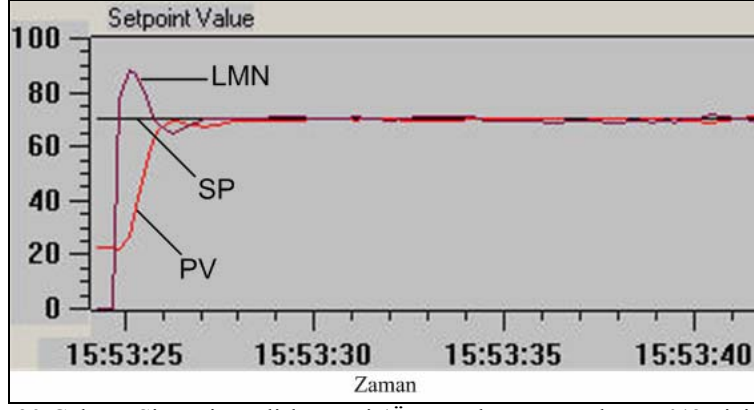
Şekil 5.17 Çalıřan Sistemin Belirlenmesi (Birinci konum 45 derece %50 izin bandı)



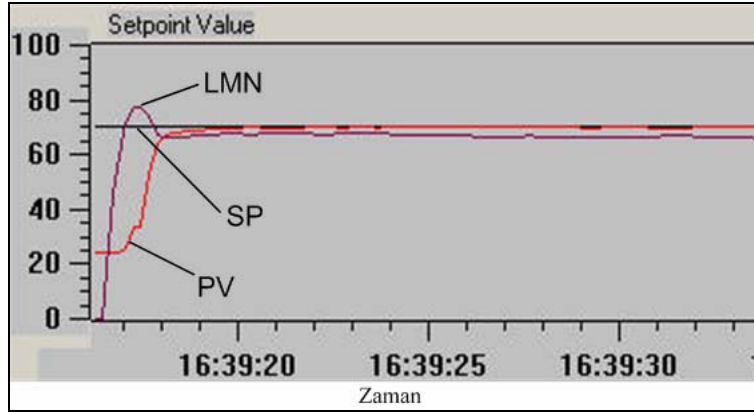
Şekil 5.18 Çalıřan Sistemin Belirlenmesi (İkinci konum 70 derece %25 izin bandı)



Şekil 5.19 Çalıřan Sistemin Belirlenmesi (İkinci konum 70 derece %50 izin bandı)



Şekil 5.20 Çalışan Sistemin Belirlenmesi (Üçüncü konum 55 derece %25 izin bandı)



Şekil 5.21 Çalışan Sistemin Belirlenmesi (Üçüncü konum 55 derece %50 izin bandı)

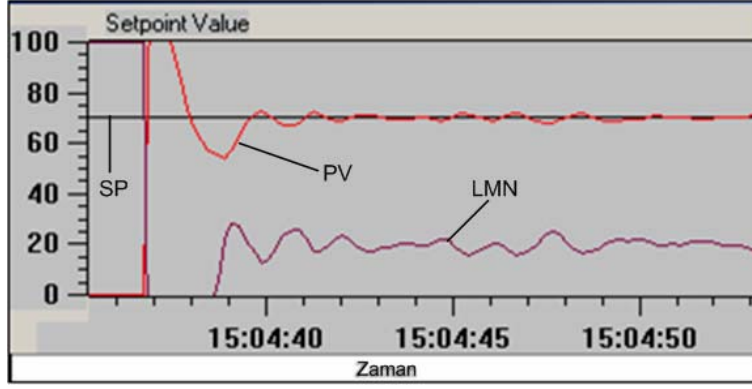
Yukarıdaki şekillerden anlaşılacağı gibi bütün kontrolörler şu ortak özellikleri sağlamıştır.

- Bütün sistemler istenilen referansı takip etmiştir
- PI kontrolör kullanılmasının sonucu olarak hiçbir sistemde sürekli hal hatası oluşmamıştır.
- Hiçbir sistemde aşım oluşmamıştır.
- Tüm sistemlerin yerleşme zamanı kabul edilebilir düzeydedir.
- Sistemler bozucu etkilere karşı oldukça dayanıklıdır.

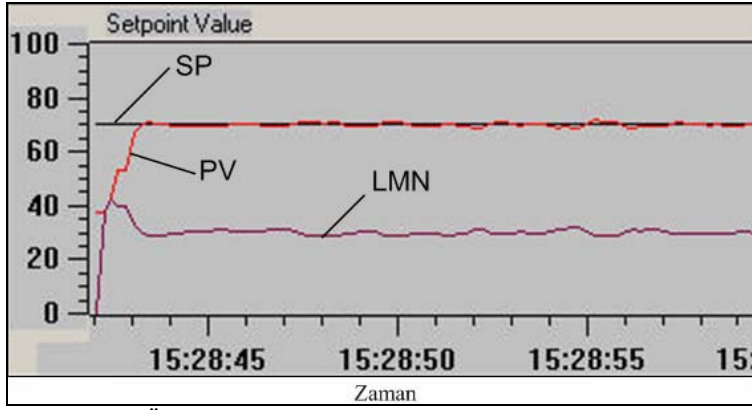
Sistemin yukarıya doğru belirlenmesi kısmında bahsedildiği gibi izin bandı arttıkça sistem daha doğru bir şekilde tanınmakta ve daha iyi bir kontrolör tasarlanılmaktadır.

5.3 Uygulama Sonuçlarının Değerlendirilmesi

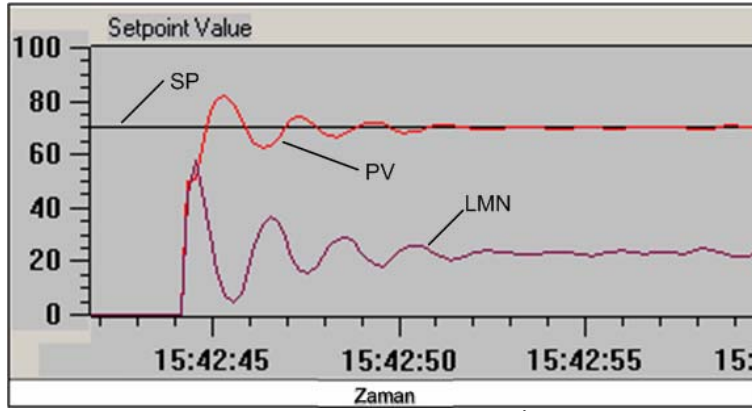
Yapılan çalışmanın değerlendirilmesi amacıyla, elde edilen sonuçlar endüstride yaygın olarak kullanılan Ziegler-Nichols yöntemi ile karşılaştırılmıştır. Önerdiğimiz ve Ziegler-Nichols yöntemi ile bulunan PI parametreleriyle kontrol edilen sistemlerin başarımına ilişkin sonuçlar grafiksel olarak aşağıda verilmiştir.



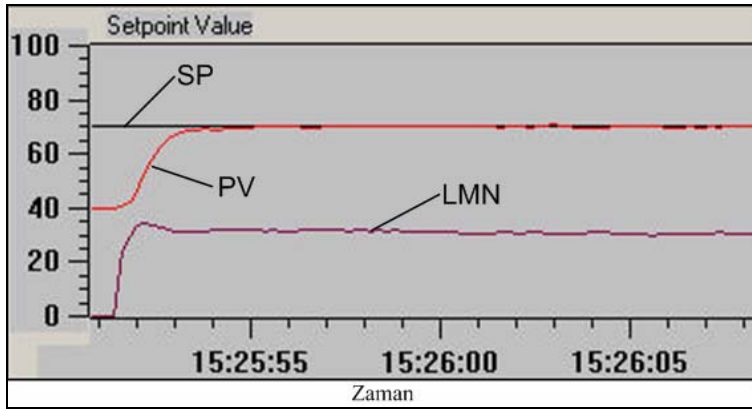
Şekil 5.22 Ziegler-Nichols yöntemine göre yanıt (Birinci konum 35 derece)



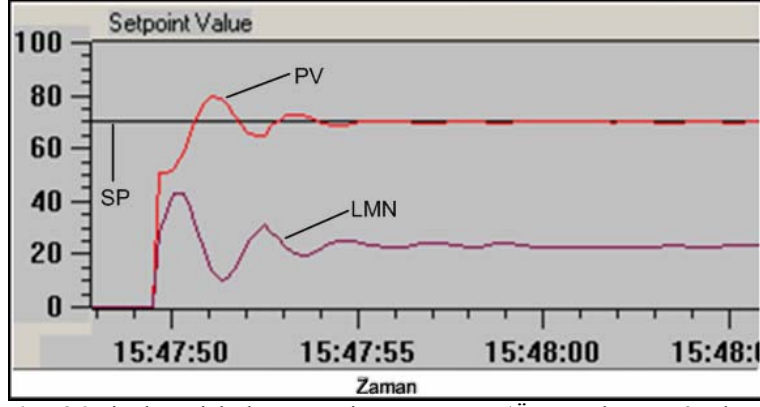
Şekil 5.23 Önerilen yönteme göre yanıt (Birinci konum 35 derece)



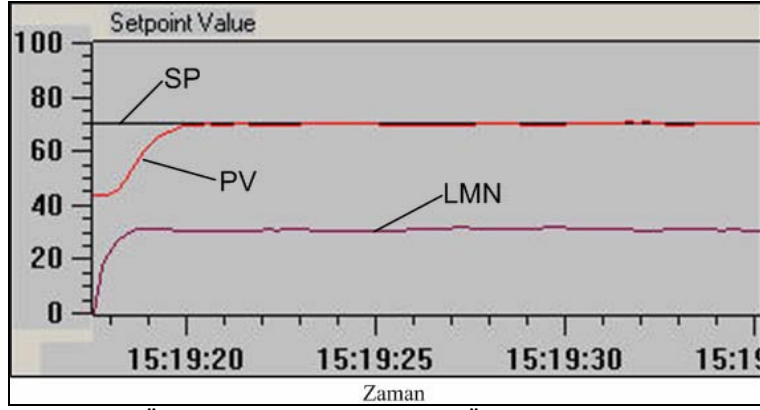
Şekil 5.24 Ziegler-Nichols yöntemine göre yanıt (İkinci konum 35 derece)



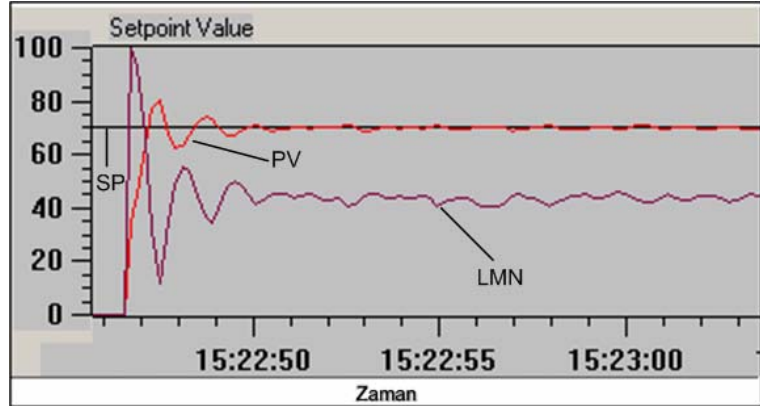
Şekil 5.25 Önerilen yönteme göre yanıt (İkinci konum 35 derece)



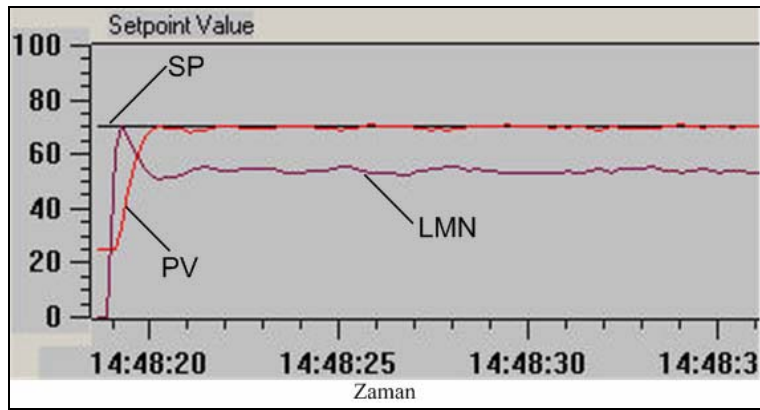
Şekil 5.26 Ziegler-Nichols yöntemine göre yanıt (Üçüncü konum 35 derece)



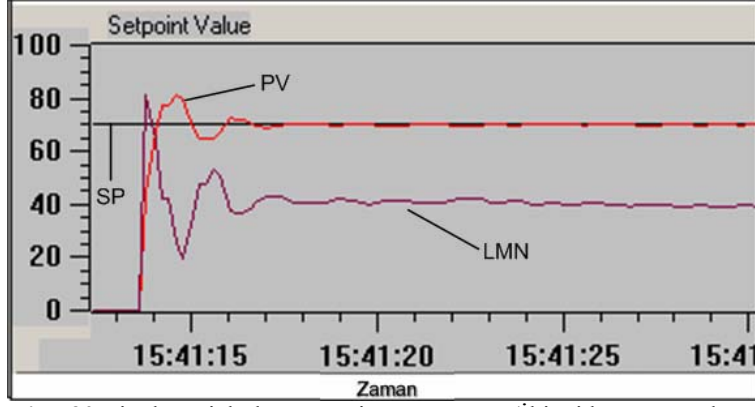
Şekil 5.27 Önerilen yönteme göre yanıt (Üçüncü konum 35 derece)



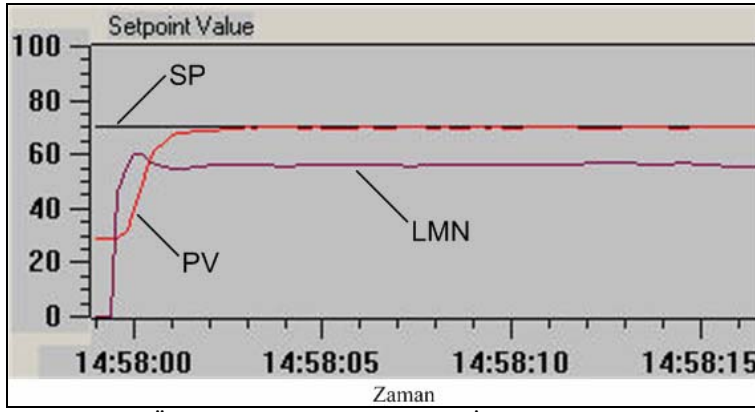
Şekil 5.28 Ziegler-Nichols yöntemine göre yanıt (Birinci konum 55 derece)



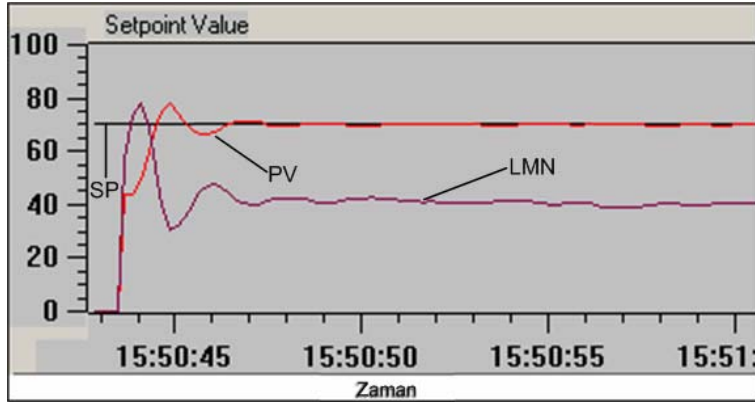
Şekil 5.29 Önerilen yönteme göre yanıt (Birinci konum 55 derece)



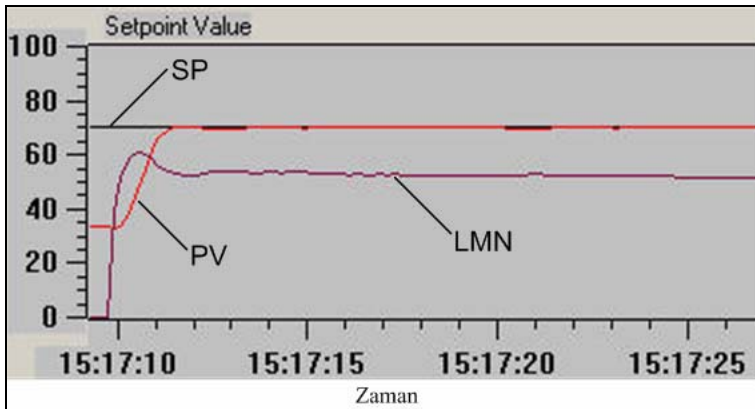
Şekil 5.30 Ziegler-Nichols yöntemine göre yanıt (İkinci konum 55 derece)



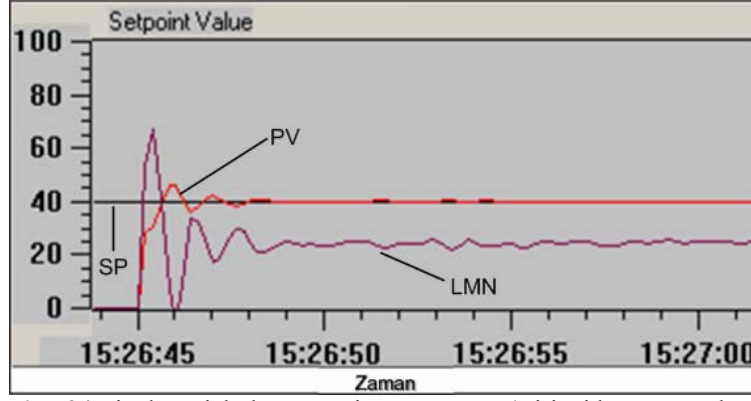
Şekil 5.31 Önerilen yönteme göre yanıt (İkinci konum 55 derece)



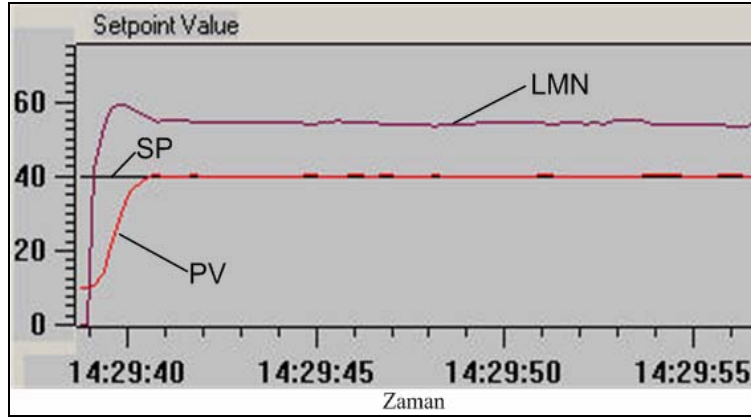
Şekil 5.32 Ziegler-Nichols yöntemine göre yanıt (Üçüncü konum 55 derece)



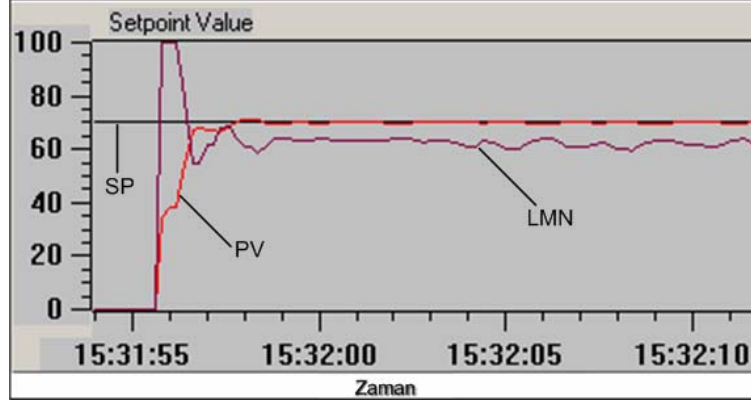
Şekil 5.33 Önerilen yönteme göre yanıt (Üçüncü konum 55 derece)



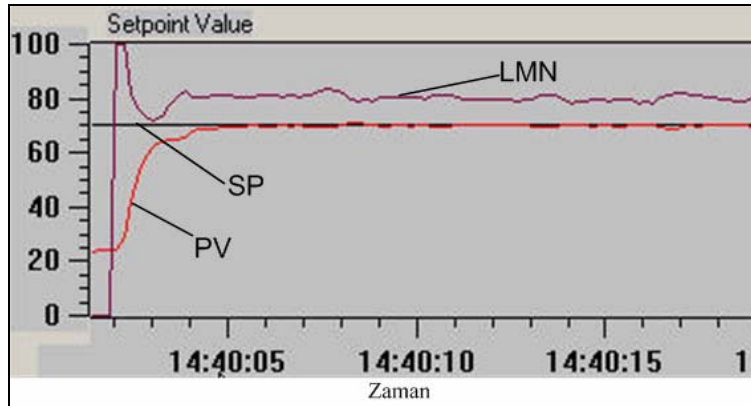
Şekil 5.34 Ziegler-Nichols yöntemine göre yanıt (Birinci konum 75 derece)



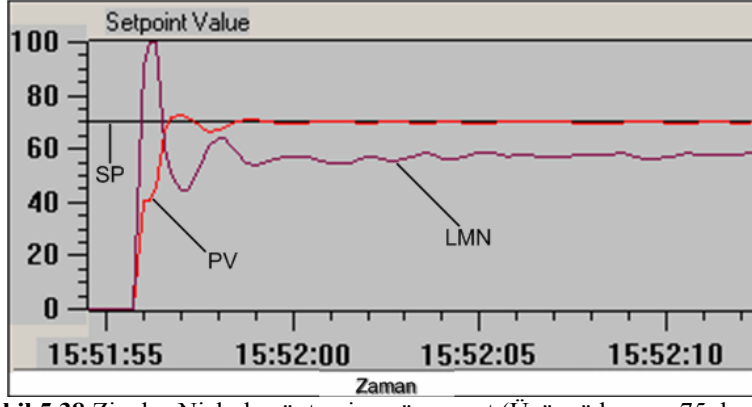
Şekil 5.35 Önerilen yönteme göre yanıt (Birinci konum 75 derece)



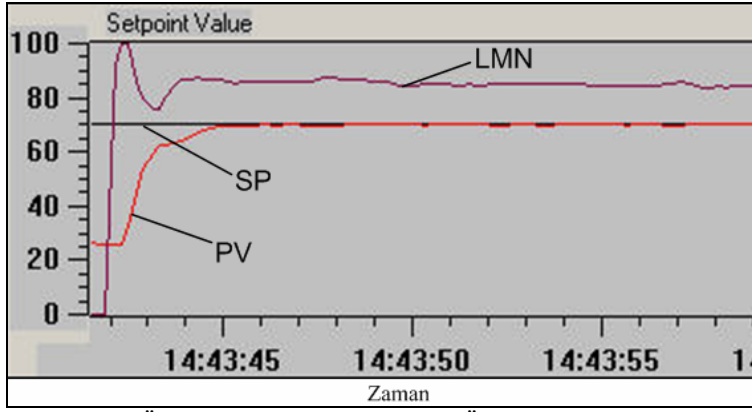
Şekil 5.36 Ziegler-Nichols yöntemine göre yanıt (İkinci konum 75 derece)



Şekil 5.37 Önerilen yönteme göre yanıt (İkinci konum 75 derece)



Şekil 5.38 Ziegler-Nichols yöntemine göre yanıt (Üçüncü konum 75 derece)



Şekil 5.39 Önerilen yönteme göre yanıt (Üçüncü konum 75 derece)

6. SONUÇ

Bu çalışmada, endüstriyel uygulamalarda yaygın olarak karşılaşılan ve birinci mertebeden ölü zamanlı sistem olarak tanımlanabilen süreçler için bir kontrol kuralı elde edilmiş ve sonuçları incelenmiştir. Kontrol edilen sistemin zaman tanım bölgesindeki büyüklüklerine bağlı olarak ifade edilen kontrol kuralının, basit ve kolay uygulanabilir özellikte olması nedeniyle, kendiliğinden ayarlamalı PI olarak da kullanılabilirdiği uygulamalı olarak gösterilmiştir. Bu amaçla Simatic S7 PLC kontrol birimi ile kontrol edilen sistemlerde doğrudan kullanılacak iki fonksiyon bloğu (FB) geliştirilmiştir. Birinci fonksiyon bloğu (FB3), sürecin ilk devreye alınması durumunda, sistemi belirlemekte ve PI katsayılarını bulmakta, ikinci fonksiyon bloğu (FB4) ise devredeki bir süreç için aynı işlemleri gerçekleştirmektedir. Birinci mertebeden ölü zamanlı sistemlerin endüstride yaygın olduğu göz önüne alındığında bu iki fonksiyonun geniş bir uygulama alanı bulacağı düşünülmektedir.

Birinci mertebeden ölü zamanlı sistemler için kendi PI kontrol parametrelerini bulan bu yöntem ile sistem parametrelerinin doğru bulunması durumunda PI katsayıları sorunsuz hesaplanabilmektedir. Burada en önemli sorun sistem modelinin belirlenmesinde yaşanmaktadır. Uygulama yapılan düzeneğin, gerçek endüstriyel sistemlere göre, daha yalın ve sorunsuz olması nedeniyle, sistem belirlemede yakın değerler elde edilmiştir. Ancak, karmaşık ve gürültü kaynaklarının yoğun olduğu endüstriyel süreçlerde, özellikle çalışır durumda sistem belirleme ve kontrol işleminin başarısız olma durumu ile karşılaşılabilir. Bu tür durumlarda, elde edilen kontrol kuralının zamana bağlı büyüklüklere göre tanımlanmış olması nedeniyle, kullanıcı; süreç yanıtlarına bakarak PI parametrelerini ayarlayabilir.

Son söz olarak hazırlanan programın endüstrideki birinci mertebeden ölü zamanlı sistemlerin kontrolü için, kullanım kolaylığı ve elde ettiği sonuçlar açısından bakıldığında uygulanabilir olduğu düşünülmektedir.

KAYNAKLAR

- Bateson N. R. (1996): Control System Technology, Prentice Hall.
- Berger, H. (2003): Automating with SIMATIC, Publicis Corporate Publishing.
- Cheng-Ching Yu (2006): Autotuning of PID Controllers, Springer.
- Haugen F. (2004): PID Control, Tapir Academic Press.
- Johnson M. A. , Moradi M. H. (2005): PID Control, Springer.
- Kontrol Laboratuvarı Ders Notları (2007): Isıl Süreç Kontrol Sistemi
- Kurtulan, S. (2005): PLC İle Endüstriyel Otomasyon, Birsen Yayıncılık.
- Leigh J. R. (1993): Applied Digital Control, Prentice Hall.
- Normey-Rico J. E. (2007): Camacho E. F. Control of Dead-time Processes, Springer.
- Olsson, G. , Piani, G. (1992): Computer Systems for Automation and Control, Prentice Hall.
- Siemens Simatic Manager Help Menu
- Visioli A. (2006): Practical PID Control, Springer.

ÖZGEÇMİŞ

İrem Koca, 29 Temmuz 1985 yılında Eskişehir’de dünyaya geldi. İlk öğrenimini, Eskişehir Adalet İlkokulu’nda, orta ve lise öğrenimini Eskişehir Fatih Anadolu Lisesi’nde tamamladı. 2003 yılında, İstanbul Teknik Üniversitesi Kontrol Mühendisliği Bölümü’ne girmeye hak kazandı. Haziran 2008 döneminde Kontrol Mühendisliği lisans programını tamamladı.

19 Temmuz 1986’da Ankara’da doğan Oytun Eriş, ilköğrenimini Gölcük Piri Reis İlkokulu’nda ve İstanbul Maltepe Anadolu Lisesi’nde tamamladı. Eskişehir Fatih Fen Lisesi’nde başladığı lise eğitimini, burslu olarak okuduğu Özel Marmara Fen Lisesi’nde birincilikle tamamladı. 2004 yılında İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi Kontrol Mühendisliği Bölümü’nde başladığı lisans eğitimini Haziran 2008’de tamamladı.