

**ÖZ AYARLAMALI PID KONTROLÖR
(PLC UYGULAMASI)**

**YÜKSEK LİSANS TEZİ
Müh. Murat DOĞAN**

**Anabilim Dalı: ELEKTRİK MÜHENDİSLİĞİ
Programı: KONTROL ve OTOMASYON MÜHENDİSLİĞİ**

MART 2006

**ÖZ AYARLAMALI PID KONTROLÖR
(PLC UYGULAMASI)**

**YÜKSEK LİSANS TEZİ
Müh. Murat DOĞAN
504021108**

Tezin Enstitüye Verildiği Tarih : 02 Mart 2006

Tezin Savunulduğu Tarih : 03 Şubat 2006

Tez Danışmanı

Doç. Dr. Salman KURTULAN

Diğer Jüri Üyeleri

Doç. Dr. Fuat GÜRLEYEN (İ.T.Ü)

Yrd. Doç. Dr. Berk ÜSTÜNDAĞ (İ.T.Ü)

ÖNSÖZ

Gelişen teknoloji ile birlikte, endüstriyel otomasyon alanında kullanılan konvansiyonel sistemler yerini PLC gibi mikroişlemci tabanlı gelişmiş sistemlere bırakmaktadır. PLC'ler konvansiyonel sistemlerin açıklarını kapatmakta, kullanıcıya daha geniş, sistem kontrolü imkanı ve hata takibini kolaylaştıran arayüzler sunmaktadır.

PLC gibi sistemlerin endüstriyel otomasyon alanında kullanılmaya başlaması, bu alanda eğitimli bir kadronun gereksinimini de artırmıştır.

Herşeyin otomatik kumanda edildiği bir yapıda, hala kontrolörlerin çalışabilmek için bir ön sistem tanımına ihtiyaç duyması bu konuda ciddi bir eksikliğin olduğunu göstermektedir.

“Öz Ayarlamalı PID Kontrolör” konulu tezimin bu eksikliği gidermede küçük de olsa bir adım olmasını diler, her şeyden önce bana Kontrol Mühendisliği'ni sevdiren ve yol gösteren Sayın Hocam Doç. Dr. Salman Kurtulan'a ve fikirleri ile bana destek olan çalışma arkadaşlarım Gökçen Babaoğlu ve Bahadır Yetik'e teşekkür ederim.

Mart, 2006

Murat DOĞAN

İÇİNDEKİLER

TABLO LİSTESİ	iv
ŞEKİL LİSTESİ	v
SEMBOL LİSTESİ	vi
ÖZET	vii
SUMMARY	viii
1. GİRİŞ	1
1.1 Çalışmanın Amacı	1
2. PID KONTROLÖR	2
2.1 PID Kontrolörün Yapısı	2
2.1.1 P Kontrolör	3
2.1.2 PI Kontrolör	3
2.1.3 PID Kontrolör	4
2.1.4 PID Kontrolörün Sisteme Uygulanışı	4
3. ÖZ AYARLAMALI (SELF TUNING) PID KONTROLÖR	5
3.1 Kullanılabilecek Yöntemler	5
3.1.1 Ziegler – Nicholas (Z-N) Osilasyon Metodu (<i>Oscillation Method</i>)	6
3.1.2 Cohen-Coon Reaksiyon Eğrisi Metodu (<i>Reaction Curve Method</i>)	6
3.1.3 Ziegler – Nicholas (Z-N) Reaksiyon Eğrisi Metodu (<i>Reaction Curve Method</i>)	8
3.1.4 Haalman Yöntemi	9
3.1.5 Genişletilmiş Ziegler-Nicholas Yöntemi	9
3.1.6 Chien, Hrones ve Reswick Yöntemi	11
3.2 Kontrolörün Gerçeklenmesi	12
3.3 Adım 1 – Yumuşatma (Smoothing) Fonksiyonu : fxSmooth	13
3.3.1 fxTriangle – Üçgen Dalga Üretici	15
3.3.2 fxPulse – Kare Dalga Üretici	17
3.4 Adım 2 – PID Fonksiyonu : fxPID	18
3.4.1 fxFOS – 1. Dereceden Sistem Simülasyon Bloğu	21
3.4.2 fxFODS – 1. Dereceden Gecikmeli Sistem Simülasyon Bloğu	23
3.5 Adım 3 – Öz Ayarlamalı PID Fonksiyonu : fxSelfTune	26
3.5.1 fxSelfTune Fonksiyonun Testi	28
4. TEST SONUÇLARI	32
KAYNAKLAR	36

TABLO LİSTESİ

		Sayfa No
Tablo 3.1	Ziegler-Nicholas Osilasyon Metodu	6
Tablo 3.2	Cohen-Coon Reaksiyon Eğrisi Metodu	7
Tablo 3.3	Ziegler-Nicholas Reaksiyon Eğrisi Metodu	8
Tablo 3.4	CHR yöntemi, %0 aşım için	12
Tablo 3.5	CHR yöntemi, %20 aşım için	12
Tablo 4.1	fxSmooth giriş ve çıkışları	14
Tablo 4.2	fxTriangle giriş ve çıkışları	15
Tablo 4.3	fxPulse giriş ve çıkışları	17
Tablo 4.4	fxPID giriş ve çıkışları	19
Tablo 4.5	fxFOS giriş ve çıkışları	22
Tablo 4.6	fxFODS giriş ve çıkışları	24
Tablo 4.7	fxPID giriş ve çıkışları	26
Tablo 4.8	qPhase çıkışı	27

ŞEKİL LİSTESİ

		Sayfa No
Şekil 2.1	PID Kontrolör	4
Şekil 3.1	Cohen-Coon, Sistem cevabı	7
Şekil 3.2	Ziegler-Nicholas, Sistem cevabı	8
Şekil 4.1	fxSmooth fonksiyonu	14
Şekil 4.2	Üçgen Dalga	15
Şekil 4.3	Üçgen Dalganın fxSmooth fonksiyonuna uygulanışı	15
Şekil 4.4	Üçgen Dalga ve yumuşatma işlem sonucu	16
Şekil 4.5	Kare Dalga	17
Şekil 4.6	Kare Dalganın fxSmooth fonksiyonuna uygulanışı	17
Şekil 4.7	Kare Dalga ve yumuşatma işlem sonucu	18
Şekil 4.8	fxPID fonksiyonu CFC görünümü	19
Şekil 4.9	fxFOS fonksiyonu CFC görünümü	21
Şekil 4.10	fxFOS ve fxPID fonksiyonları bağlantı şekli	22
Şekil 4.11	fxFOS-fxPID simülasyon sonucu	23
Şekil 4.12	fxFODS fonksiyonu CFC görünümü	24
Şekil 4.13	fxFODS ve fxPID fonksiyonları bağlantı şekli	25
Şekil 4.14	fxFODS-fxPID simülasyon sonucu	25
Şekil 4.15	fxSelfTune fonksiyonu CFC görünüşü	26
Şekil 4.16a	fxSelfTune – fxPID fonksiyonlarının bağlantı şekli	28
Şekil 4.16b	fxPID – fxFODS fonksiyonlarının bağlantı şekli	29
Şekil 4.17	WinCC test arabirimi	30
Şekil 4.18	Test sonucu, 10. ve 11. aşama	31
Şekil 4.19	Test sonucu, 13. aşama	31
Şekil 5.1	Hesaplanan parametreler ile sistem kontrolü	32
Şekil 5.2	Test sonucu	34
Şekil 5.3	Test sonucu	35

SEMBOL LİSTESİ

		Sayfa No
K_p, K_c	PID Kazanç Katsayısı	2
T_i	PID Integral Katsayısı	2
T_d	PID Türev Katsayısı	2
$m(t)$	Kontrolör Çıkışı	3
$e(t)$	Hata (Referans - Sistem Çıkışı)	3
T_s	Kontrolör örnekleme zamanı	3
P_c	Osilasyondaki sistemin periyodu	6
G_c	Kontrolör transfer Fonksiyonu	9
G_l	Açık çevrim transfer fonksiyonu	9
G_p	Sistem Transfer fonksiyonu	9
L	Sistem gecikme zamanı	9
X_A	Fonksiyon çıkışı (Yumuşatma işlemi sonucu)	13
T_A	Fonksiyon örnekleme zamanı (sn)	13
X_E	Fonksiyon girişi (Sistem geri beslemesi anlık değeri)	13
X_{EA}	Önceki adımlarda hesaplanan referans değeri	13
T_G	Yumuşatma faktörü (sn)	13
X'_{EA}	Sonraki fonksiyon işlemleri için referans	13
$G(s),G(z)$	Sistem çıkışı (s ve z domaini)	21
$y(k)$	k. zamandaki sistem çıkışı	21
K	Sistem kazancı	21
a	Sistem zamanı	21
T	Örnekleme zamanı	21
$r(k)$	k. zamandaki sistem girişi	21

ÖZET

Bu çalışmada ilk önce kontrolörlerden bahsedilmiş, daha sonra ise PID kontrolörün detayı anlatılmıştır. PID kontrolörün işlev hale getirilebilmesi için gerekli olan adımlar anlatılmış, uygulanacak sisteme uygun olarak uyarlanması gereken parametreler detaylı olarak incelenmiştir. Buradan yola çıkılarak, kendi parametrelerini kendi ayarlayabilen PID türünden ve bu konuda yapılan çalışmaların bazılarında (Ziegler-Nicholas Reaksiyon Eğrisi Metodu, Cohen-Coon Reaksiyon Eğrisi Metodu gibi) bahsedilmiştir. Bu çalışmalardan Ziegler-Nicholas Reaksiyon Eğrisi Metodu seçilmiş ve bu yöntemin bir uygulaması olarak, yardımcı fonksiyonlarda göz önünde bulundurularak, Siemens PLC' ler için Öz Ayarlamalı PID Kontrolör kütüphanesi oluşturulmuştur. Kütüphane oluşturulurken endüstriyel sistemler gözönünde bulundurularak, PID uygulaması ve testi için gerekli olan tüm fonksiyonlar (uygulama geliştirme aşamasında ve sisteme uygulanırken) başından sonuna kadar oluşturulmuştur. Kütüphanede sistem analizi, PID, öz ayarlamalı PID ve sistem simülasyon bloklarına (birinci dereceden sistem simülasyon bloğu ve birinci dereceden gecikmeli sistem simülasyon bloğu) yer verilmiştir. Kütüphane fonksiyonları Step-7 (Siemens S7-300 ve S7-400 PLC'leri programlama arayüzü) ve WinCC (Siemens SCADA-kullanıcı kontrol ve izleme programı) gibi gelişkin endüstriyel otomasyon yazılım ortamları kullanılarak gerçekleştirilmiştir. Gerçeklenen tüm fonksiyonlar ayrı ayrı test edilmiş ve test sonuçları verilmiştir. Sonuç olarak, Ziegler-Nicholas Reaksiyon Eğrisi Metodu kullanılarak Öz Ayarlamalı PID kontrolörün PLC ortamında gerçekleştirilebileceği ve yapılan uygulamanın başarılı olduğu gözlenmiştir.

SELF TUNING PID CONTROLLER

SUMMARY

In this study, firstly mentioned about controllers and then PID Controllers are explained in detail. Necessary steps are explained in order to make PID controllers functional, and the PID parameters, which have to be tuned for the applied system, are studied in detail. After all these, mentioned about Self-Tuning PID and some of researches studied on this subject (like Ziegler-Nicholas Reaction Curve Method and Cohen-Coon Reaction Curve Method). A Self-Tuning PID Controller Library is formed for Siemens PLCs' by taking as a reference Ziegler-Nicholas Reaction Curve Method. All needed functions (for application development side and implementation side) for PID application and application tests are implemented from beginning to end by evaluating industrial systems. In this library system analysis, PID application, Self-Tuning PID application and system simulation blocks (first order system simulation block and first order delay system simulation block) are obtained. These library functions are tested by using fully developed industrial automation software environments such as Step-7 (programming interface for Siemens S7-300 and S7-400 PLCs') and WinCC (Siemens SCADA- operator control and monitoring tool). All implemented functions are tested and all the results are explained. As a result, it is observed that Self-Tuning PID Controller functions can be applied to PLCs' by using Ziegler-Nicholas Reaction Curve Method.

1. GİRİŞ

1.1 Çalışmanın Amacı

Aç-Kapa, PID gibi kontrolörler günümüzde endüstriyel sistemlerin vazgeçilmez unsurlarındandırlar. En basit sistemlerden en karmaşık sistemlere kadar geniş bir kullanım alanları bulunmaktadır ve her geçen gün bu kontrolörlerin kullanımı yaygınlaşmaktadır. Bu tür kontrolörlerin yaygınlaşması, bu kontrolörleri kumanda edebilecek ve parametrelerini kullanıldığı sisteme göre uyarlayabilecek eğitimli bir kadroyu da zorunlu kılmıştır.

Orta ölçekteki bir fabrikada ortalama 30 adet Aç-Kapa, PID gibi kontrolörler kullanılmaktadır. Kontrolör parametreleri, sistem devreye alınırken sisteme göre (sistem parametrelerine göre) uyarlanır.

Bilindiği üzere, sistem parametreleri çevre koşullarına göre değişkenlik gösterirler. Örneğin; bir sıcaklık sisteminin parametreleri yaz koşullarında başka, kış koşullarında başka olabilmektedir. Sayısal olarak ifade etmek gerekirse, yazın 1.0 olan sistem kazancı, kışın 0.8' e düşebilmektedir. Bu durum PID gibi endüstriyel kontrolörlerin parametrelerinin belirli aralıklarla gözden geçirilmesini ve değişen sisteme uygun olarak tekrar belirlenmesini zorunlu kılar.

Endüstriyel kontrolörlerin sayıca fazlalığı ve bu kontrolörlerin değişen sistemi takip edebilecek şekilde parametre uyarlamasını yapabilecek eğitimli bir kadronun azlığı düşünüldüğünde, bu tezin konusu olan, kedinen uyarlamalı PID kontrolörlerin gerekliliği ortaya çıkacaktır.

Günümüzde endüstriyel kontrol sistemleri, klasik yöntemler (kontaktör, röle, zaman rölesi,) yerine PLC (Programmable Logic Controller) gibi mikroişlemci tabanlı sistemleri tercih etmektedirler. PLC gibi mikroişlemci tabanlı kontrol sistemleri otomasyon sistemlerini daha basite indirgemekte ve tasarım aşamasını kısaltmaktadır. Bu sebeple uygulama alanı olarak PLC ortamı seçilmiştir. Bu tezin amacı Öz Ayarlamalı PID Kontrolör'ü Siemens S7 PLC'si programlama ortamını

kullanarak gerçeklemek ve sistem kontrolünü kolaylaştırabilecek bir kütüphanenin hazırlanmasıdır.

2. PID KONTROLÖR

PID (Proportional-Integral-Derivative) kontrolör endüstriyel sistemlerde en sık kullanılan kontrolör türüdür. Basit olarak PID kontrolör, sistemde oluşan hataya, hatanın integraline ve türevine göre bir çıkış üretir.

Bu türdeki kontrolörler çok çeşitli sistemlerde kullanılabilir. Örneğin; sıcaklık sistemleri, ağırlık sistemleri, tartım sistemleri, hız kontrol sistemleri, vs... . Kullanıldığı sisteme göre ayarlanması gereken 3 tane parametresi vardır. Bunlar sırası ile;

- K_p - Kazanç Katsayısı,
- T_i - Integral Katsayısı ,
- T_d - Türev Katsayısıdır.

Bu 3 parametreden herhangi birinin değişmesi PID davranışına doğrudan etki eder. Dolayısı ile kontrol edilecek sistem iyi analiz edildikten sonra bu parametrelerin sisteme uygun olarak belirlenmesi gerekir.

2.1 PID Kontrolörün Yapısı

PID kontrolörler standart olarak 3 türde bulunabilirler. Kontrol edilecek sistemin yapısına göre (sistem parametrelerine göre) bu üç türden biri tercih edilir. Bu türler sırası ile şunlardır;

- P Kontrolör
- PI Kontrolör
- PID Kontrolör

Bahsi geçen PID kontrolör türleri aşağıda sırası ile incelenecektir. PID kontrolör türleri anlatılırken, fonksiyonlar iki şekilde verilecektir. Bunlardan biri sürekli form diğeri ise ayrık formdur. PID fonksiyonlarında kullanılacak semboller ve bu sembollerin açıklamaları şöyledir;

m(t) : Kontrolör Çıkışı

e(t) : Hata (Referans - Sistem Çıkışı)

K_c : Kontrolör Kazancı (Proportional)

T_i : Kontrolör Integral Zamanı (s) (Integral Time)

T_d : Kontrolör Türev Zamanı (s) (Derivative Time)

T_s : Kontrolör örnekleme zamanı

2.1.1 P Kontrolör

Sadece kazanç katsayısının aktif olduğu PID kontrolör türüdür. Basit şekilde ifade edilirse; sistemde oluşan hata, PID kazancı ile çarpılır ve kontrolör çıkışı olarak dışarıya verilir.[2,3]

$$m(t) = K_c [e(t)] \quad (2.1.1a)$$

$$m[n] = K_c [e[n]] \quad (2.1.1b)$$

2.1.2 PI Kontrolör

Kazanç ve integral katsayılarının aktif olduğu PID kontrolör türüdür. Basit şekilde ifade edilirse; sistemde oluşan hata, oluşan toplam hatanın integral katsayısına bölümü ile toplanır ve PID kazancı ile çarpılarak kontrolör çıkışı olarak dışarıya verilir.[2,3]

$$m(t) = K_c \left[e(t) + \frac{1}{T_i} \int e(t) dt \right] \quad (2.1.2a)$$

$$m[n] = K_c \left[e[n] + \frac{T_s}{T_i} \sum_{i=0}^n e[i] \right] \quad (2.1.2b)$$

2.1.3 PID Kontrolör

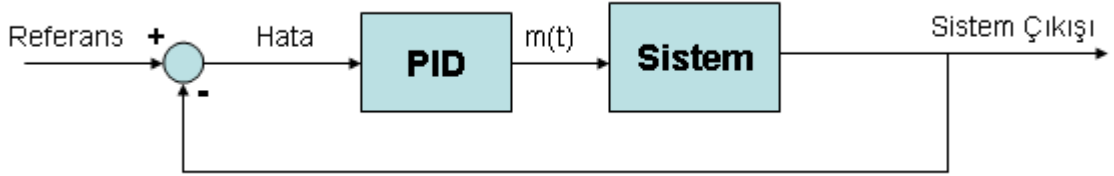
Tüm PID katsayılarının aktif olduğu PID kontrolör türüdür. Sistemde oluşan hata, hatanın toplamı ve hatanın türevi PID katsayıları ile çarpılarak kontrolör çıkışı olarak dışarıya verilir.[2,3]

$$m(t) = K_c \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (2.1.3a)$$

$$m[n] = K_c \left[e[n] + \frac{T_s}{T_i} \sum_{i=0}^n e[i] + \frac{T_d}{T_s} (e[n] - e[n-1]) \right] \quad (2.1.3b)$$

2.1.4 PID Kontrolörün Sisteme Uygulanışı

PID kontrolörün sisteme uygulanış şekli ise şu şekildedir; [1]



Şekil 2.1 : PID Kontrolör

Anlık referans ile sistem geri besleme değerinin farkı alınarak PID kontrolöre giriş olarak verilir. PID çıkışı ise sisteme uygulanır.

3. ÖZ AYARLAMALI (SELF TUNING) PID KONTROLÖR

Bölüm 2’de bahsedildiği gibi PID kontrolörün parametrelerinin sisteme göre uyarlanması gerekmektedir. Bu işlem için çeşitli matematiksel yollar; örneğin “Kök Eğrisi Yöntemi” kullanılmaktadır. PID parametrelerini hesaplamakta kullanılan yöntemler (“Kök Eğrisi Yöntemi” gibi) sistemin matematiksel bir modeline ihtiyaç duyarlar ve ilgili kişilere ağır yükler getirirler. Kaldı ki sistem geri beslemesini hatasız ölçebilmek, sisteme binen bozucu etkileri anlık olarak hesaplayabilmek olanaksızdır. Ayrıca sistem parametrelerinin (sistem kazancı gibi) değişen çevre koşullarına paralel olarak değiştiği de unutulmamalıdır. Bu koşullar göz önünde bulundurulduğunda, kendi parametrelerini kullanıldığı sisteme göre uyarlayabilen ve değişen çevre koşullarına adapte olabilecek bir PID kontrolöre ihtiyaç duyulduğu açıktır.

3.1 Kullanılabilecek Yöntemler

Kendi parametrelerini, önceden belirlenmiş herhangi bir matematiksel yönteme ihtiyaç duymadan hesaplayabilecek PID kontrolöre Öz Ayarlamalı PID Kontrolör denir. Bir PID kontrolörü, Öz Ayarlamalı PID Kontrolör’ e dönüştürebilecek çeşitli yollar literatürde mevcuttur. Bunlardan bazıları şunlardır;

- Ziegler – Nicholas (Z-N) Osilasyon Metodu (*Oscillation Method*)
- Cohen-Coon Reaksiyon Eğrisi Metodu (*Reaction Curve Method*)
- Ziegler – Nicholas (Z-N) Reaksiyon Eğrisi Metodu (*Reaction Curve Method*)
- Haalman Yöntemi

3.1.1 Ziegler – Nicholas (Z-N) Osilasyon Metodu (*Oscillation Method*)

Bu yöntemde sistem osilasyona girinceye kadar sisteme K_p ' si artan bir P kontrolör uygulanır ve sistem çıkışı gözlenerek PID parametreleri hesaplanır.[4] Bu yöntemde izlenecek adımlar şunlardır;

1. K_p ' si olabildiğince küçük olan bir P kontrolör ile başla,
2. Sistem osilasyona girinceye kadar K_p ' yi küçük adımlar ile artır,
3. Osilasyondaki sistemin çıkışını gözleyerek şu parametreleri not et;
 - K_p : Osilasyondaki sistemin P katsayısı
 - P_c : Osilasyondaki sistemin periyodu
4. K_p ve P_c parametrelerini ve aşağıdaki tabloyu kullanarak PID parametrelerini hesapla

Tablo 3.1 : Ziegler-Nicholas Osilasyon Metodu

	K_p	T_i	T_d
P	$0.5K_p$	-	-
PI	$0.45K_p$	$0.833P_c$	-
PID	$0.6K_p$	$0.5P_c$	$0.125P_c$

Pratikte bu yöntemin ne kadar kullanılabilir olduğu tartışılması gereken bir konudur. Sisteme osilasyon yaptırmak veya sistemi osilasyona götürebilmek için sisteme sürekli artan bir giriş vermek sisteme zarar verebilecek unsurlardır. Dolayısı ile bu tezin konusu olan Öz Ayarlamalı PID Kontrolör' ün uygulamasında bu yöntem tercih edilmeyecektir.

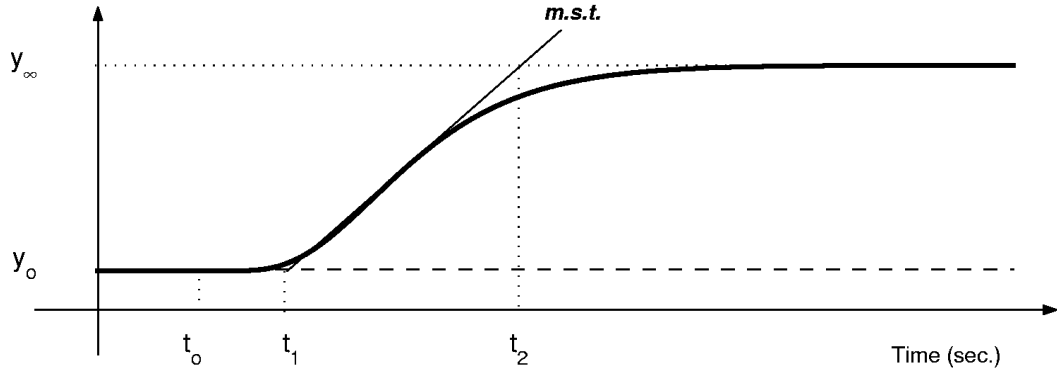
3.1.2 Cohen-Coon Reaksiyon Eğrisi Metodu (*Reaction Curve Method*)

Bu yöntemde sistem davranışını tanıyabilmek için kontrol işlemine açık çevrimde başlanır. Sisteme belli bir büyüklükte giriş uygulanır ve sistem çıkışı gözlenerek sistem parametrelerine ulaşılır.[4] Bu yöntemde izlenmesi gereken adımlar şu şekildedir;

- 1- Sistemi açık çevrim kontrole al,
- 2- Sistem çıkışı sabit hale gelinceye kadar bekle,

- 3- Sistem çıkışı $y(t) = y_0$ ve kontrolör çıkışı $u(t) = u_0$ olarak kaydet,
- 4- t_0 anında sisteme $u(t_0) = u_n$ girişi uygula, (u_n girişinin yeterli büyüklükte, %10' dan büyük, olması gerekir. Küçük bir giriş uygulamak sistem davranışı hakkında yanlış veriler elde edilmesine neden olabilir.)
- 5- Sistem geri besleme değeri sabit hale gelinceye kadar sistem çıkışı gözle
- 6- Sistem çıkışı sabit hale gelince şu parametreleri şekil 3.1 yardımı ile hesapla;

$$K_0 = \frac{y_n - y_0}{u_n - u_0} \quad \tau_0 = t_1 - t_0 \quad v_0 = t_2 - t_1 \quad (3.1.1)$$



Şekil 3.1 : Cohen-Coon, Sistem cevabı

- 7- Aşağıdaki tablo yardımı ile PID kontrolör parametreleri hesaplanır,

Tablo 3.2 : Cohen-Coon Reaksiyon Eğrisi Metodu

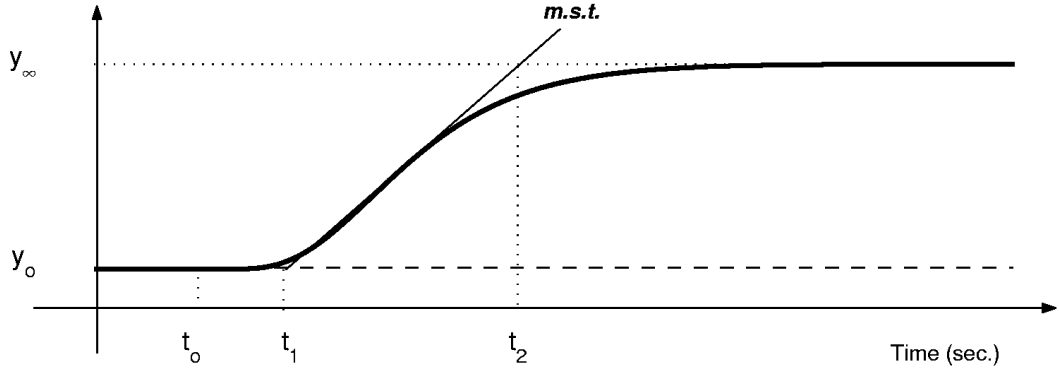
	K_p	T_i	T_d
P	$\frac{v_0}{K_0 \tau_0} \left[1 + \frac{\tau_0}{3v_0} \right]$	-	-
PI	$\frac{v_0}{K_0 \tau_0} \left[0.9 + \frac{\tau_0}{12v_0} \right]$	$\frac{\tau_0 [30v_0 + 3\tau_0]}{9v_0 + 20\tau_0}$	-
PID	$\frac{v_0}{K_0 \tau_0} \left[\frac{4}{3} + \frac{\tau_0}{4v_0} \right]$	$\frac{\tau_0 [32v_0 + 6\tau_0]}{13v_0 + 8\tau_0}$	$\frac{4\tau_0 v_0}{11v_0 + 2\tau_0}$

3.1.3 Ziegler – Nicholas (Z-N) Reaksiyon Eğrisi Metodu (*Reaction Curve Method*)

Bu yöntem Cohen-Coon Reaksiyon Eğrisi Metoduna benzerdir. Cohen-Coon metodunda anlatılan ilk 5 adım aynen tekrar edilir ve aşağıda anlatıldığı şekilde yöneme devam edilir;

6- Sistem çıkışı sabit hale gelince şu parametreleri şekil 3.2 yardımı ile hesapla;

$$K_0 = \frac{y_n - y_0}{u_n - u_0} \quad \tau_0 = t_1 - t_0 \quad v_0 = t_2 - t_1 \quad (3.1.2)$$



Şekil 3.2 : Ziegler-Nicholas, Sistem cevabı

7- Aşağıdaki tablo yardımı ile PID kontrolör parametreleri hesaplanır,

Tablo 3.3 : Ziegler-Nicholas Reaksiyon Eğrisi Metodu

	K_p	T_i	T_d
P	$\frac{v_0}{K_0 \tau_0}$	-	-
PI	$\frac{0.9v_0}{K_0 \tau_0}$	$3\tau_0$	-
PID	$\frac{1.2v_0}{K_0 \tau_0}$	$2\tau_0$	$0.5\tau_0$

Görüldüğü üzere Cohen-Coon Reaksiyon Eğrisi Metodu ile Ziegler-Nicholas Reaksiyon Eğrisi Metodu aynı yolu izlemektedirler. Son aşamada, PID parametre hesabında, farklılıklar göstermektedirler. Bu tezde gerçekleştirilecek olan Öz Ayarlamalı

PID Kontrol için yukarıda anlatılan 3. yöntem (Ziegler-Nicholas Reaksiyon Eğrisi Metodu) kullanılacaktır.

3.1.4 Haalman Yöntemi

Diğer bir yöntemde Haalman yöntemidir.[3] G_l açık çevrim transfer fonksiyonu ise, seçilecek kontrolörün transfer fonksiyonu G_c şöyle olur;[4]

$$G_c = \frac{G_l}{G_p} \quad (3.1.3)$$

G_p sistem transfer fonksiyonudur.

Sistem gecikme zamanı L olan sistemler için, Haalman sistem açık çevrim transfer fonksiyonunu şu şekilde seçilmesini önerir; [4]

$$G_l(s) = \frac{2}{3Ls} e^{-sL} \quad (3.1.4)$$

Eşitlikteki 2/3 oranı referans değerindeki değişikliklerin ortalama kare hatasını azaltmak için kullanılmaktadır. [4]

Haalman yöntemini transfer fonksiyonu

$$G_p(s) = \frac{1}{1+sT} e^{-sL} \quad (3.1.5)$$

Şeklinde olan sisteme uygulanır ise, kontrolör transfer fonksiyonu şu şekilde olur[4];

$$G_c(s) = \frac{2T}{3L} \left(1 + \frac{1}{sT}\right) \quad (3.1.6)$$

Bu da bize kazancı $2T/3L$ ve integral katsayısı T olan PI kontrolörü verir. [4]

3.1.5 Genişletilmiş Ziegler-Nicholas Yöntemi

Açık çevrim sistemin Nyquist eğrisi üzerinde herhangi bir nokta seçelim. Seçilen bu noktayı istediğimiz bir noktaya taşıyan bir kontrolör tasarlamış olalım. Seçilmiş olan noktayı şöyle ifade edelim; [4]

$$A = G_p(iw_0) = r_a e^{i(\pi + \phi_a)} \quad (3.1.7)$$

Bu noktayı şu noktaya götüren kontrolörü seçelim; [4]

$$B = G_l(iw_0) = r_b e^{i(\pi + \phi_b)} \quad (3.1.8)$$

Bu kontrolörün frekans cevabı; [4]

$$G_c(iw_0) = r_c e^{i(\pi + \phi_c)} \quad (3.1.9)$$

ise

$$r_b e^{i(\pi + \phi_b)} = r_a r_c e^{i(\pi + \phi_a + \phi_c)} \quad (3.1.10)$$

olur.

Kontrolör değişkenlerini şöyle seçmemiz gerekir; [4]

$$r_c = \frac{r_b}{r_a} \quad (3.1.11)$$

$$\phi_c = \phi_b - \phi_a \quad (3.1.12)$$

Kontrolörümüzün bir PI kontrolör olduğunu düşünürsek; [4]

$$K = \frac{r_b \cos(\phi_b - \phi_a)}{r_a} \quad (3.1.13)$$

$$T_i = \frac{1}{w_0 \tan(\phi_a - \phi_b)} \quad (3.1.14)$$

Kontrolörümüzün bir PID kontrolör olduğunu düşünürsek; [4]

$$K = \frac{r_b \cos(\phi_b - \phi_a)}{r_a} \quad (3.1.15)$$

$$w_0 T_d - \frac{1}{w_0 T_i} = \tan(\phi_b - \phi_a) \quad (3.1.16)$$

PID kontrol parametrelerinde T_i ve T_d nin birbirine bağımlı olduğu görülmektedir. Denklem için fazladan bir denkleme daha ihtiyaç duyulacaktır. Bu iki parametrenin birbirine şöyle bağlı oldukları kabul edilir; [4]

$$T_d = \alpha T_i \quad (3.1.17)$$

Dikkat edilirse $\alpha = 0,25$ seçildiğinde Ziegler-Nicholas Yöntemi elde edilmektedir.

T_d denklemde yerine konulursa; [4]

$$T_i = \frac{1}{2\alpha\omega_0} \left(\tan(\phi_b - \phi_a) + \sqrt{4\alpha + \tan^2(\phi_b - \phi_a)} \right) \quad (3.1.18)$$

$$T_d = \alpha T_i \quad (3.1.19)$$

Denklemleri elde edilir.

Ziegler-Nicholas yöntemi kullanılarak uygun bir nokta seçildiği düşünülürse; [4]

$$r_a = \frac{1}{K_u} \quad (3.1.20)$$

$$\phi_a = 0 \quad (3.1.21)$$

olduğu görülür. Bu denklemler PID parametrelerinde yerine konulacak olursa; [4]

$$K = K_u r_b \cos(\phi_b) \quad (3.1.22)$$

$$T_i = -\frac{T_u}{2\pi \tan \phi_b} \quad (3.1.23)$$

Elde edilir. Son olarak $\alpha = 0,25$ seçilirse PID parametreleri şöyle olur; [4]

$$K = K_u r_b \cos(\phi_b) \quad (3.1.24)$$

$$T_i = \frac{T_u}{\pi} \left(\frac{1 + \sin \phi_b}{\cos \phi_b} \right) \quad (3.1.25)$$

$$T_d = \frac{T_u}{4\pi} \left(\frac{1 + \sin \phi_b}{\cos \phi_b} \right) \quad (3.1.26)$$

3.1.6 Chien, Hrones ve Reswick Yöntemi

Yine bu yöntemde Ziegler-Nicholas yönteminin geliştirilmiş bir yoludur. Chien, Hrones ve Reswick (CHR) Ziegler-Nicholas yöntemini geliştirerek “aşım olmadan en hızlı cevap” ve “%20 aşım ile en hızlı cevap” kriterlerine ulaşmaya çalışmışlardır. [4]

Kontrolör parametrelerinin CHR yöntemi ile hesaplanabilmesi için sistem modelinde kullanılan a ve L parametrelerine ihtiyaç duyulur. Bu parametreler Ziegler-Nicholas Birim Basamak Yöntemi kullanarak hesaplanır. Daha sonra a ve L parametreleri kullanılarak PID parametreleri hesaplanır. Hesaplama yöntemi (%0 aşım ve %20 aşım için ayrı ayrı) aşağıdaki tablolarda verilmiştir. [4]

Tablo 3.4 : CHR yöntemi, %0 aşım için

	K_p	T_i	T_d
P	0.3/a	-	-
PI	0.6/a	4L	-
PID	0.95/a	2.4L	0.42L

Tablo 3.5 : CHR yöntemi, %20 aşım için

	K_p	T_i	T_d
P	0.7/a	-	-
PI	0.7/a	2.3L	-
PID	1.3/a	2L	0.42L

3.2 Kontrolörün Gerçeklenmesi

Ziegler-Nicholas Reaksiyon Eğrisi Metodu kullanılarak Siemens S7 PLC'leri için Öz Ayarlamalı PID Kontrolör kütüphanesi hazırlanmıştır. Kullanılan yazılımlar aşağıda sıralanmıştır;

- Simatic Manager V5.3 (Program yönetimi ve hazırlanması)
- WinCC V6.0 (Ekran yönetimi)
- S7-PLCSIM V5.2 (Yazılan programların testi için)

Kütüphane hazırlanırken sırası ile şu adımlar izlenmiştir;

- Adım 1-** Sistem geri beslemesi için “Yumuşatma” (Smoothing) fonksiyonunun hazırlanması,
- Adım 2-** PID fonksiyonunun hazırlanması,
- Adım 3-** Öz Ayarlamalı PID Kontrolörün gerçekleştirilmesi,

Bu adımlar sırası ile aşağıda anlatılmıştır.

3.3 Adım 1 – Yumuşatma (Smoothing) Fonksiyonu : fxSmooth

Geri beslemeli kontrol sistemleri için sistem geri beslemesinin ne kadar doğru ölçüldüğü hayati önem taşır. Endüstriyel sistemlerde geri beslemenin elde edilmesi için çeşitli dönüştürücüler kullanılır. Örnek olarak; sıcaklık geri beslemesi için; PT100, Thermocouple vs..., basınç geri beslemesi için basınç dönüştürücüleri vs... gibi. Bu dönüştürücülerin kararlı çıkışlar üretmesi beraberinde PID fonksiyonunun da kararlı davranışını getirecektir.

Sistemde oluşabilecek ani değişimlerden veya dönüştürücü kararsızlığından dolayı PID fonksiyonunun kararsız hale gelmesini önlemek ve daha önemlisi sistemi analiz ederken yanlış sonuçlara varmamızı engellemek için bir “Yumuşatma” fonksiyonu hazırlanmıştır. Bu fonksiyon matematikte trapezoidal diye bilinen yöntemi izleyerek sistem çıkışının (yani kontrolör girişinin) daha kararlı davranmasını sağlayacaktır. Trapezoidal yönteminde şu formüller kullanılarak çıkış üretilir;

$$X_A = \frac{T_A(X_E - X_{EA})}{T_A + 2T_G} + X_{EA} \quad (4.1.1)$$

$$X'_{EA} = \frac{2T_A(X_E - X_{EA})}{T_A + 2T_G} + X_{EA} \quad (4.1.2)$$

X_A : Fonksiyon çıkışı (Yumuşatma işlemi sonucu)

T_A : Fonksiyon örnekleme zamanı (sn)

X_E : Fonksiyon girişi (Sistem geri beslemesi anlık değeri)

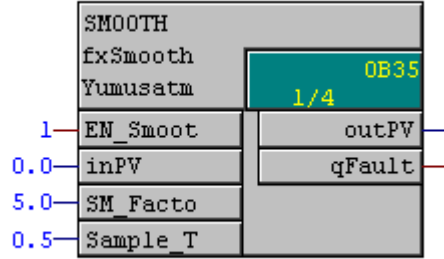
X_{EA} : Önceki adımlarda hesaplanan referans değeri

T_G : Yumuşatma faktörü (sn)

X'_{EA} : Sonraki fonksiyon işlemleri için referans

Yumuşatma faktörü T_G sistem çıkışını ne kadar kararlı görmek istediğimizi belirler birimi saniye olarak belirlenmiştir. Yumuşatma faktörü büyüdükçe PID fonksiyonu girişinde daha kararlı, yavaş hareket eden, bir sistem görecektir. Hazırlanan

yumuşatma fonksiyonu “fxSmooth” ‘un Simatic Manager’in CFC editöründe görünüşü şu şekildedir;



Şekil 4.1 : fxSmooth fonksiyonu

Fonksiyon giriş ve çıkışları aşağıda tablo halinde verilmiştir;

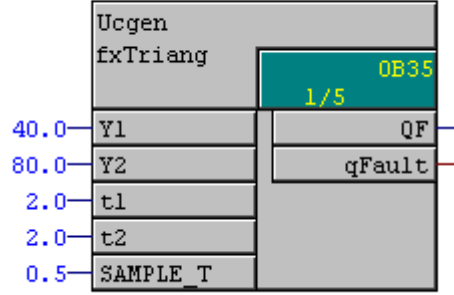
Tablo 4.1 : fxSmooth giriş ve çıkışları

Adı	Tipi	Açıklama
EN_Smooth	<i>BOOL</i>	1-> Yumuşatma işlemi başlat 0-> Yumuşatma işlemi durdur
inPV	<i>REAL</i>	Sistem geri beslemesi
SM_Factor	<i>REAL</i>	Yumuşatma Faktörü (sn)
Sample_T	<i>REAL</i>	Örnekleme Zamanı (sn)
outPV	<i>REAL</i>	İşlem sonucu (PID girişi)
qFault	<i>BOOL</i>	Hata

Fonksiyonu deneyebilmek ve yaptığı işlemin sonucunu görebilmek için 2 adet fonksiyon hazırlanmıştır. Bunlar sırası ile ;

- fxTriangle: İstenilen ölçekte üçgen dalga oluşturacak fonksiyon
- fxPulse :İstenilen ölçekte kare dalga oluşturacak fonksiyon

3.3.1 fxTriangle – Üçgen Dalga Üretici



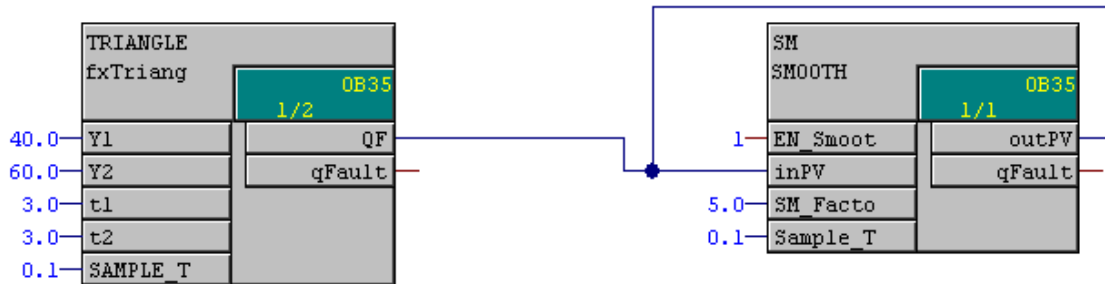
Şekil 4.2 : Üçgen Dalga

Yazılan üçgen dalga üretici fonksiyon “fxTriangle”ın CFC editöründe görünüşü yukarıda verilmiştir. Bu fonksiyon istenilen büyüklükte ve genişlikte üçgen dalgalar oluşturacaktır. Fonksiyon giriş ve çıkışları aşağıdaki tabloda verilmiştir.

Tablo 4.2 : fxTriangle giriş ve çıkışları

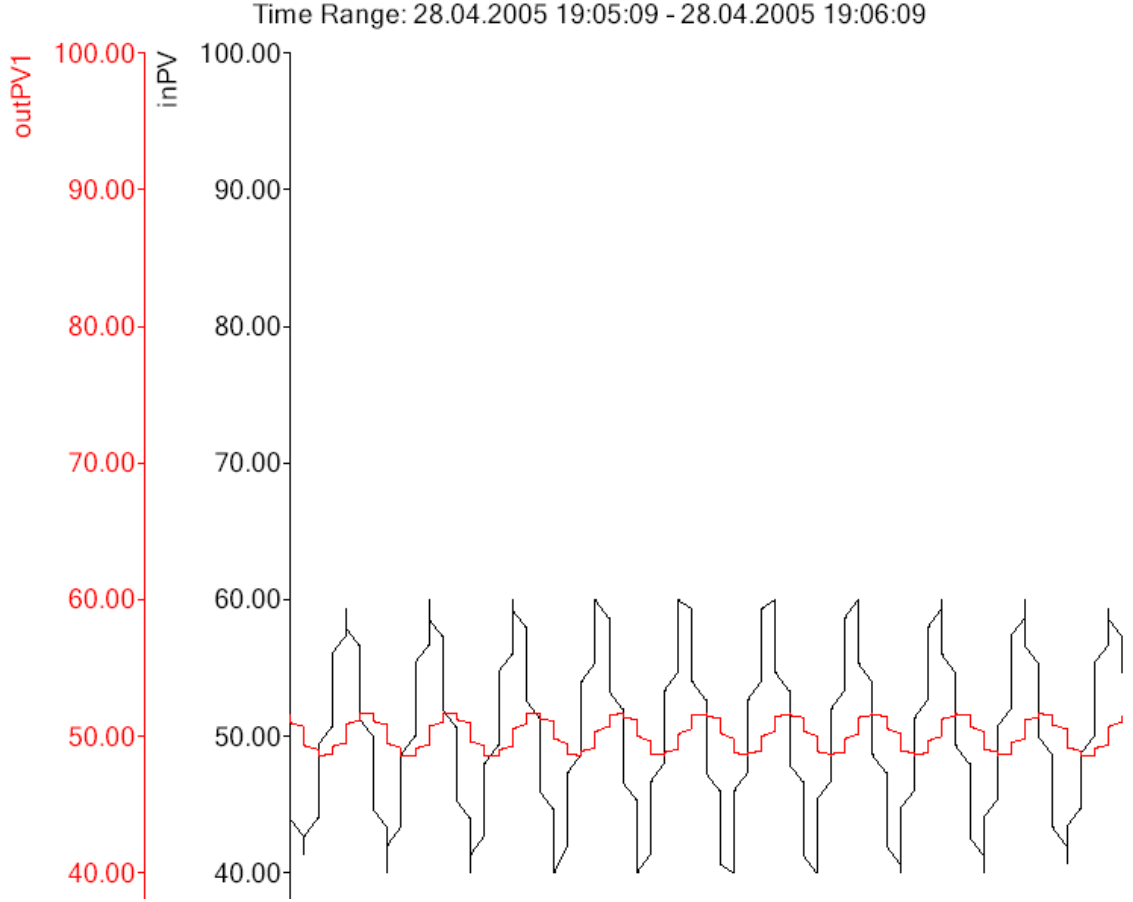
Adı	Tipi	Açıklama
Y1	<i>REAL</i>	Üçgen dalga alt değeri
Y2	<i>REAL</i>	Üçgen dalga üst değeri
t1	<i>REAL</i>	Yükselme Zamanı (sn)
t2	<i>REAL</i>	Alçalma Zamanı (sn)
Sample_T	<i>REAL</i>	Örnekleme Zamanı (sn)
QF	<i>REAL</i>	Fonksiyon Çıkışı
qFault	<i>BOOL</i>	Hata

Yazılan fonksiyonun çıkışı fxSmooth fonksiyonuna giriş olarak verilmiştir. Yapılan işlemin CFC editöründe görünüşü şu şekildedir;



Şekil 4.3 : Üçgen Dalganın fxSmooth fonksiyonuna uygulanışı

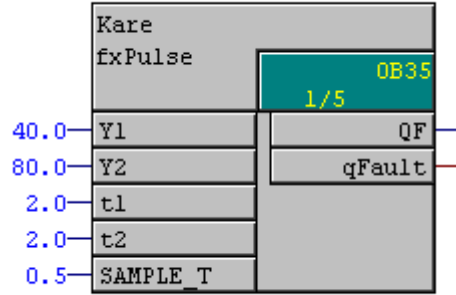
Üçgen dalga üretici fonksiyonu çıkışına yumuşatma işlemi uygulanmış ve sonuç WinCC Eğri Kontrol'ünde (Trend Control) gözlenmiştir. Uygulanan üçgen dalga ve yumuşatma işlem sonucu aşağıdaki şekilde görülmektedir.



Şekil 4.4 : Üçgen Dalga ve yumuşatma işlem sonucu

Görüldüğü gibi sistem geri beslemesi (üçgen dalga üretici çıkışı) %40 ile %60 arasında dalgalanmasına rağmen, 5 saniyelik bir yumuşatma faktörü ile PID girişi (yumuşatma işlemi çıkışı) çok daha karlı bir hale getirilmiştir.

3.3.2 fxPulse – Kare Dalga Üretici



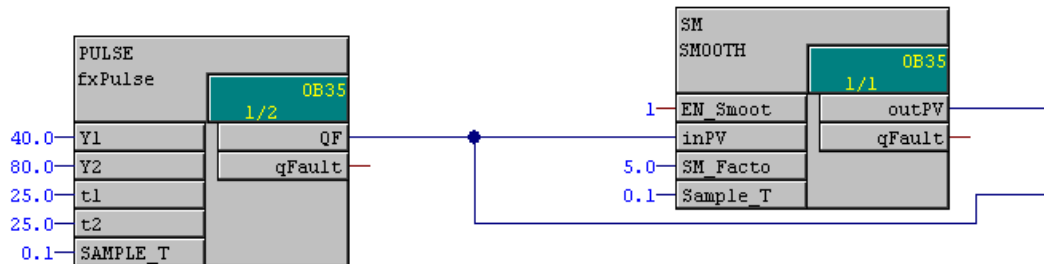
Şekil 4.5 : Kare Dalga

Hazırlanan kare dalga üretici fonksiyon “fxPulse”ın CFC editöründe görünüşü yukarıda verilmiştir. Bu fonksiyon istenilen büyüklükte ve genişlikte kare dalgalar oluşturacaktır. Fonksiyon giriş ve çıkışları aşağıdaki tabloda verilmiştir.

Tablo 4.3 : fxPulse giriş ve çıkışları

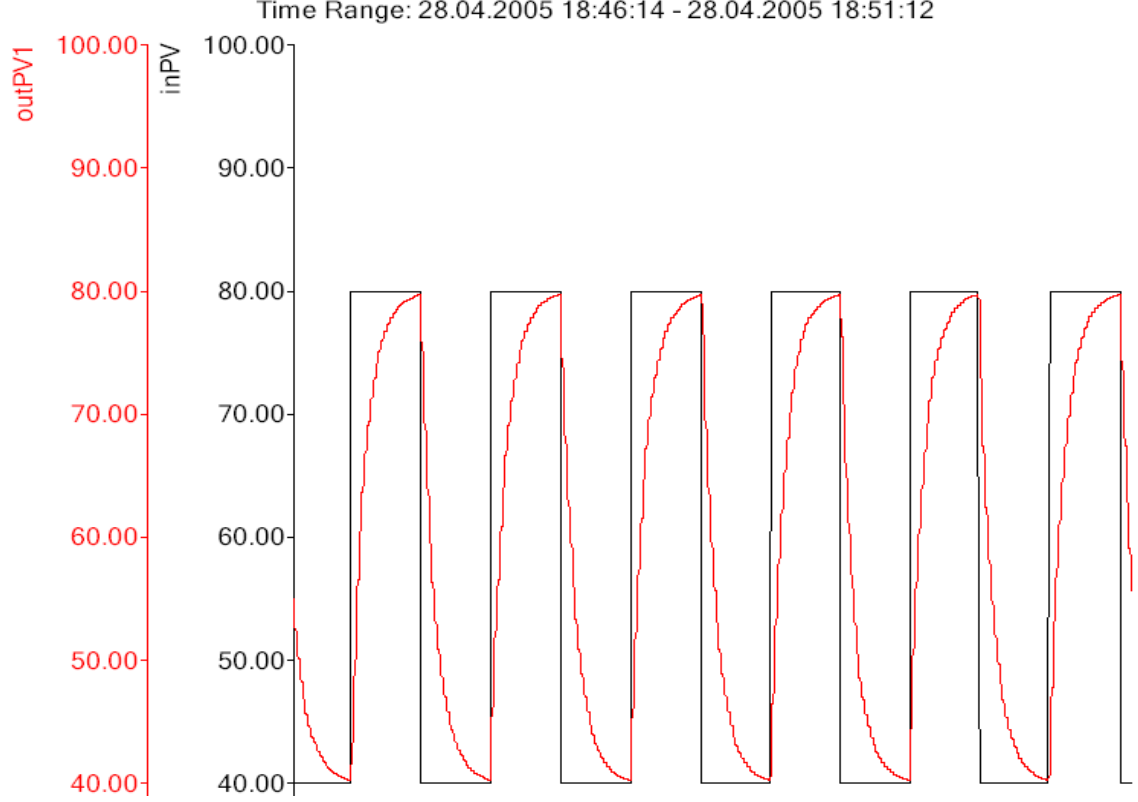
Adı	Tipi	Açıklama
Y1	REAL	Kare dalga alt değeri
Y2	REAL	Kare dalga üst değeri
t1	REAL	Alt değerde kalınacak Zaman (sn)
t2	REAL	Üst değerde kalınacak Zaman (sn)
Sample_T	REAL	Örnekleme Zamanı (sn)
QF	REAL	Fonksiyon Çıkışı
qFault	BOOL	Hata

Yazılan fonksiyonun çıkışı fxSmooth fonksiyonuna giriş olarak verilmiştir. Yapılan işlemin CFC editöründe görünüşü şu şekildedir;



Şekil 4.6 : Kare Dalganın fxSmooth fonksiyonuna uygulanışı

Kare dalga üretici fonksiyonu çıkışına yumuşatma işlemi uygulanmış ve sonuç WinCC Eğri Kontrol'ünde (Trend Control) gözlenmiştir. Uygulanan Kare dalga ve yumuşatma işlem sonucu aşağıdaki şekilde görülmektedir.

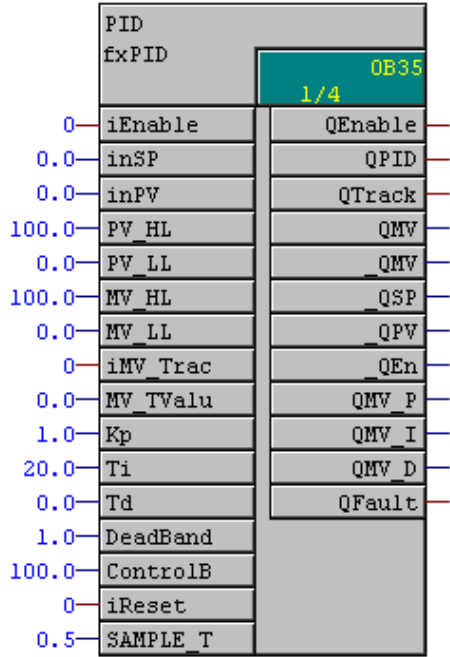


Şekil 4.7 : Kare Dalga ve yumuşatma işlem sonucu

Görüldüğü gibi sistem geri beslemesi (kare dalga üretici çıkışı) çok ani şekilde değişmektedir. 5 saniyelik bir yumuşatma faktörü ile PID girişi (yumuşatma işlemi çıkışı) yumuşatılmıştır.

3.4 Adım 2 – PID Fonksiyonu : fxPID

İkinci adım olarak PID kontrolü gerçekleyen “fxPID” fonksiyonu yazılmıştır. Bu fonksiyon sistem girişi, referans ve PID parametrelerini giriş olarak alıp bir kontrolör çıkışı üretmektedir. “fxPID” fonksiyonunun CFC editöründeki görünüşü aşağıda verilmiştir.



Şekil 4.8 : fxPID fonksiyonu CFC görünümü

Fonksiyon giriş ve çıkışları şöyledir;

Tablo 4.4 : fxPID giriş ve çıkışları

Adı	Tipi	Açıklama
iEnable	<i>BOOL</i>	Fonksiyonu devreye almak için kullanılacak giriş
İnSP	<i>REAL</i>	Referans
İnPV	<i>REAL</i>	Sistem geri beslemesi
PV_HL	<i>REAL</i>	Sistem geri beslemesi üst sınırı
PV_LL	<i>REAL</i>	Sistem geri beslemesi alt sınırı
MV_HL	<i>REAL</i>	PID çıkışı üst sınırı
MV_LL	<i>BOOL</i>	PID çıkışı alt sınırı
İmv_Track	<i>BOOL</i>	PID fonksiyonunu el konumuna almak için giriş
MV_TValue	<i>REAL</i>	PID fonksiyonunu el konumunda iken geçerli olacak PID çıkış değeri
K_p	<i>REAL</i>	PID kazanç katsayısı
T_i	<i>REAL</i>	PID integral katsayısı
T_d	<i>REAL</i>	PID türev katsayısı
DeadBand	<i>REAL</i>	% Ölü bant aralığı
ControlBand	<i>REAL</i>	% Kontrol bant aralığı
iReset	<i>BOOL</i>	Fonksiyonu ilk konumuna getirmek için giriş
Sample_T	<i>REAL</i>	Örnekleme Zamanı (sn)
QEnable	<i>BOOL</i>	Fonksiyonun devrede olduğunu gösteren çıkış
QPID	<i>BOOL</i>	PID fonksiyonunun devrede olduğunu gösteren çıkış
QTrack	<i>BOOL</i>	Fonksiyonun el konumunda olduğunu gösteren çıkış

Adı	Tipi	Açıklama
QMV	REAL	PID çıkışı
_QMV	REAL	PID çıkışı % oranı
_QSP	REAL	Referans % oranı
_QPV	REAL	Sistem geri beslemesi % oranı
_QEN	REAL	Hata % oranı
QMV_P	REAL	PID çıkışı, P
QMV_I	REAL	PID çıkışı, I
QMV_D	REAL	PID çıkışı, D
QFault	BOOL	Hata

iMV_Track:

iMV_Track girişi 1 ve fonksiyon devrede ise (iEnable = 1 ise) PID fonksiyon çıkışı QMV doğrudan MV_TValue girişine eşit olacaktır. Bu girişler daha sonra hazırlanacak Öz Ayarlamalı PID fonksiyonu tarafından kullanılacaktır.

DeadBand:

Ölü bant aralığı, sistem çıkışının tolerans aralığını belirler. Sistem geri besleme değeri ölü bant aralığını girdiğinde PID fonksiyonu çıkışlarını dondurur ve sistemi izlemeye devam eder. Sistem geri beslemesi ölü bant aralığının dışına çıktığında PID fonksiyonu tekrar aktif hale gelerek sistemi kontrol etmeye devam eder. Sistem ölü bant aralığında iken hata sıfır kabul edilir.

ControlBand:

Kontrol aralığını tanımlar. Sistem bu kontrol aralığı dışında iken kontrolör çıkışı doğrudan kontrol çıkışı üst limitine (MV_HL) eşit olur. Böylelikle sistemin daha hızlı hareket etmesini sağlamış oluruz. Sistem kontrol bant aralığında iken, sistem gözlenmeye ve sistem hatası toplanmaya devam edilir.

PID fonksiyonu ve daha sonra hazırlanacak olan Öz Ayarlamalı PID fonksiyonunun testi için 4 adet sistem simülasyon bloğu hazırlanmıştır. Bunlar sırası ile;

- fxFOS – 1.dereceden sistem simülasyonu
- fxFODS - 1. dereceden gecikmeli sistem simülasyonu
- fxSOS – 2. dereceden sistem simülasyonu
- fxSODS – 2.dereceden gecikmeli sistem simülasyonu

3.4.1 fxFOS – 1. Dereceden Sistem Simülasyon Bloğu

Birinci dereceden gecikmesiz bir sistemin transfer fonksiyonu aşağıdaki şekilde olacaktır. [5]

$$G(s) = \frac{K}{as + 1} \quad (4.2.1a)$$

Transfer fonksiyonunun z-dönüşümü ise şöyledir; [5]

$$G(z) = K \frac{1 - e^{-\frac{T}{a}}}{z - e^{-\frac{T}{a}}} \quad (4.2.1b)$$

Fark denklemi; [5]

$$y(k) = e^{-\frac{T}{a}} y(k-1) + K(1 - e^{-\frac{T}{a}}) r(k-1) \quad (4.2.1c)$$

şeklinde olacaktır.

Kullanılan semboller ve açıklamaları şöyledir;

G(s),G(z) : Sistem çıkışı (s ve z domaini)

y(k) : k. zamandaki sistem çıkışı

K : Sistem kazancı

a : Sistem zamanı

T : Örnekleme zamanı

r(k) : k. zamandaki sistem girişi

fxFOS fonksiyonu (4.2.1c) formülasyonuna göre çıkış üretir. Yazılan fonksiyonun CFC editöründeki görünüşü aşağıdaki gibidir.



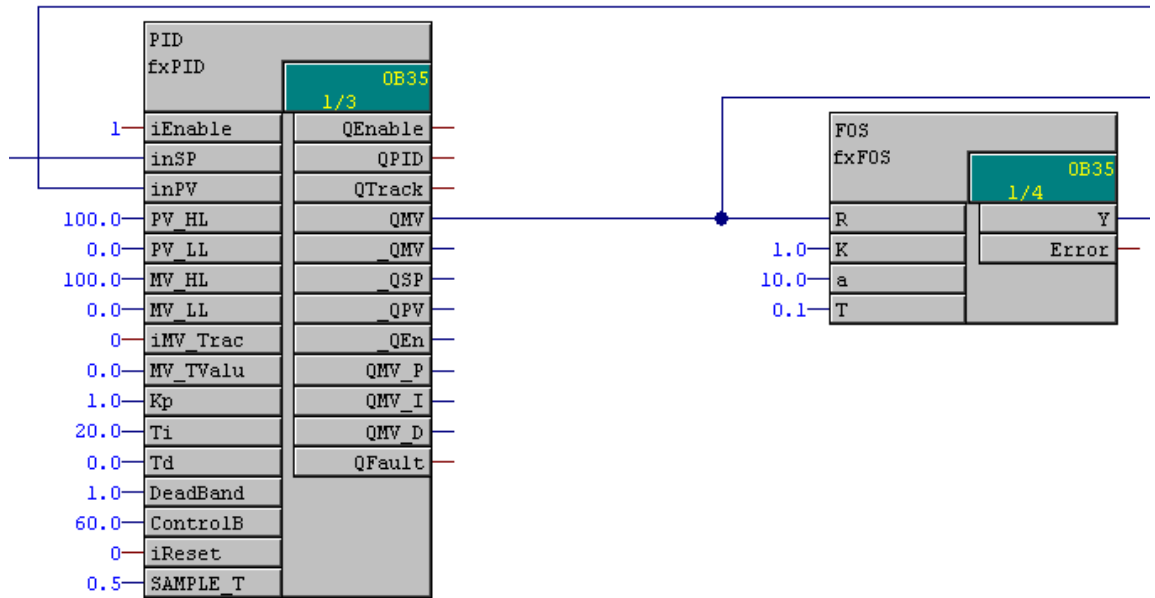
Şekil 4.9 : fxFOS fonksiyonu CFC görünümü

Fonksiyon giriş ve çıkışları şöyledir;

Tablo 4.5 : fxFOS giriş ve çıkışları

Adı	Tipi	Açıklama
R	<i>REAL</i>	Sistem girişi
K	<i>REAL</i>	Sistem kazancı
A	<i>REAL</i>	Sistem zamanı
T	<i>REAL</i>	Örnekleme Zamanı (sn)
Y	<i>REAL</i>	Sistem Çıkışı
Error	<i>BOOL</i>	Hata

fxFOS fonksiyonu 1. dereceden gecikmesiz sistemleri simüle edebilmek için kullanılmıştır. fxFOS fonksiyonunun, fxPID fonksiyonu yapılan bağlantısı ve elde edilen eğriler aşağıda verilmiştir.



Şekil 4.10 : fxFOS ve fxPID fonksiyonları bağlantı şekli

Örnekleme zamanı 100 ms olmak üzere, şu PID ve sistem parametreleri ile simülasyon yapılmıştır;

PID parametreleri;

K_p : 1.0

DeadBand : 1.0

T_i : 20.0

ControlBand : 20.0

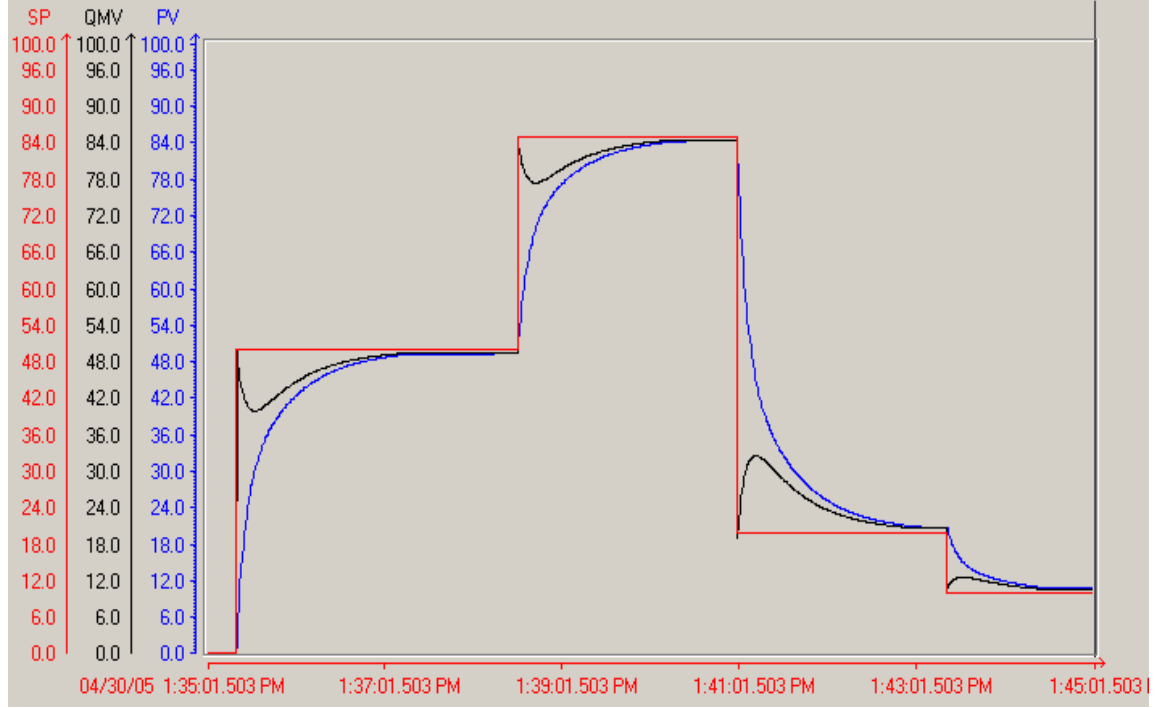
T_d : 0.0

Sistem Parametreleri;

K : 1.0

a : 10

Elde edilen sonuç Şekil 4.11’ de gösterilmiştir.



Şekil 4.11 : fxFOS-fxPID simülasyon sonucu

Şekil 4.11’de kırmızı çizgiler (SP) referansı, siyah çizgiler (QMV) kontrolör çıkışı ve mavi çizgiler (PV) sistem geri besleme değerini göstermektedirler.

3.4.2 fxFODS – 1. Dereceden Gecikmeli Sistem Simülasyon Bloğu

T_d gecikme zamanı olmak üzere, birinci dereceden gecikmeli bir sistemin transfer fonksiyonu aşağıdaki şekilde olacaktır. [5]

$$G(s) = \frac{K}{as+1} e^{-sT_d} \quad (4.2.2a)$$

$d = T_d/T$ olarak tanımlanırsa; transfer fonksiyonunun z-dönüşümü şöyle olur; [5]

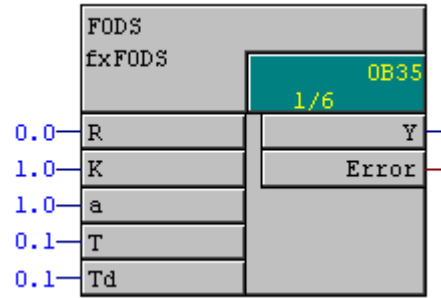
$$G(z) = K \frac{1 - e^{-\frac{T}{a}}}{z - e^{-\frac{T}{a}}} z^d \quad (4.2.2b)$$

Fark denklemi; [5]

$$y(k) = e^{-\frac{T}{a}} y(k-1) + K(1 - e^{-\frac{T}{a}}) r(k-d-1) \quad (4.2.2c)$$

şeklinde olacaktır.

fxFODS fonksiyonu (4.2.2c) formulasyonuna göre çıkış üretir. Yazılan fonksiyonun CFC editöründeki görünüşü aşağıdaki gibidir.



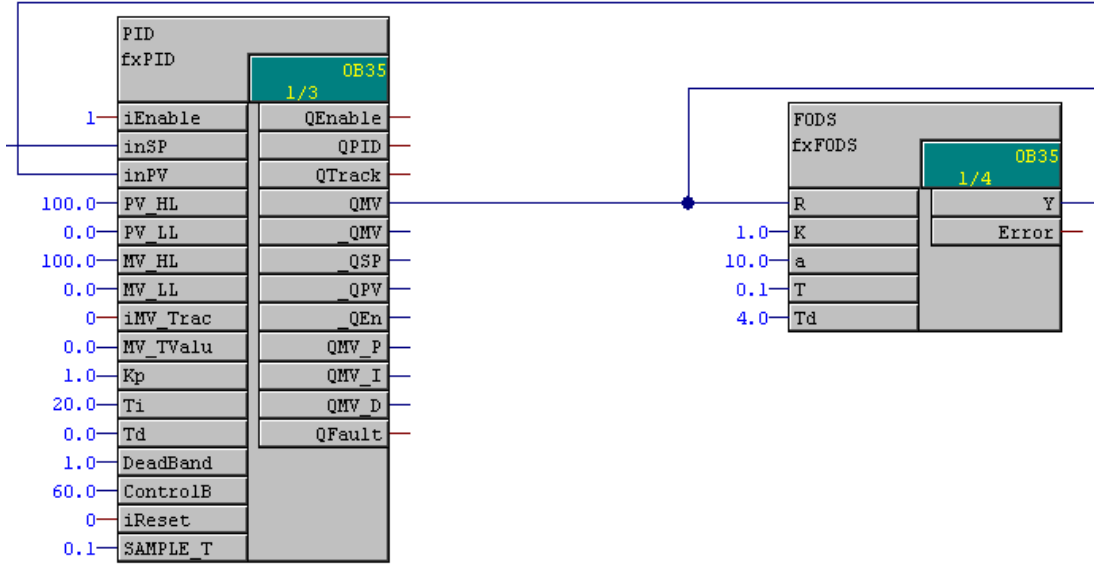
Şekil 4.12 : fxFODS fonksiyonu CFC görünümü

Fonksiyon giriş ve çıkışları şöyledir;

Tablo 4.6 : fxFODS giriş ve çıkışları

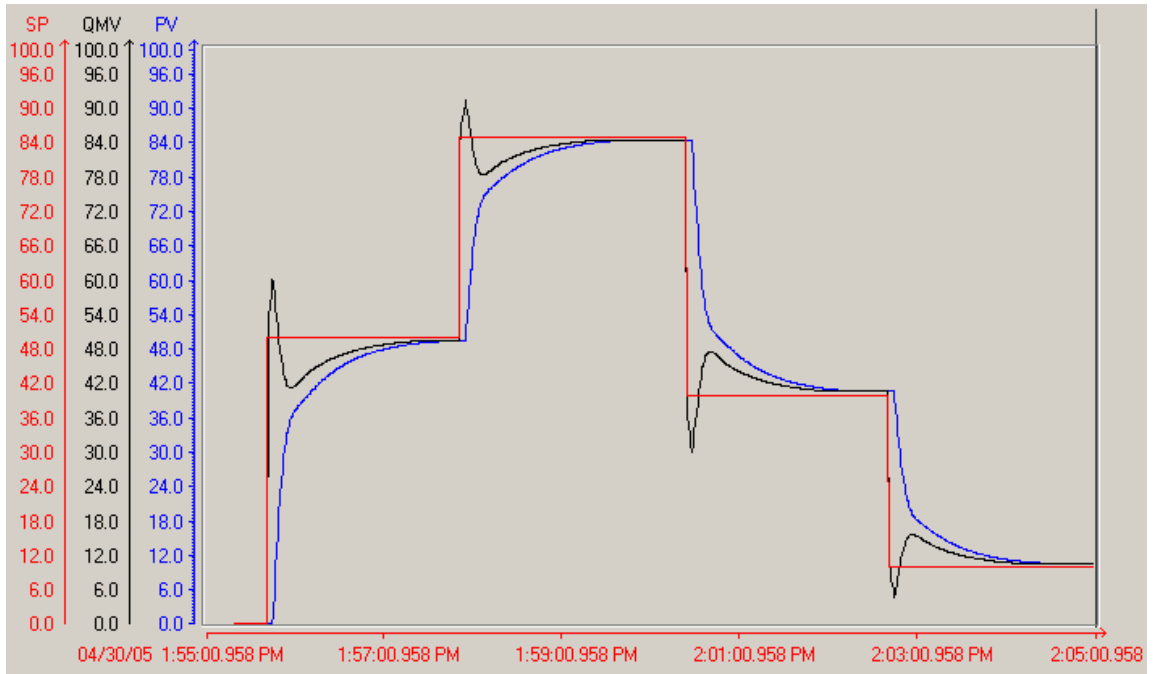
Adı	Tipi	Açıklama
R	<i>REAL</i>	Sistem girişi
K	<i>REAL</i>	Sistem kazancı
a	<i>REAL</i>	Sistem zamanı
T	<i>REAL</i>	Örnekleme Zamanı (sn)
T_d	<i>REAL</i>	Gecikme Zamanı (sn)
Y	<i>REAL</i>	Sistem Çıkışı
Error	<i>BOOL</i>	Hata

fxFODS fonksiyonu 1. dereceden gecikmeli sistemleri simüle edebilmek için kullanılmıştır. fxFODS fonksiyonunun, fxPID fonksiyonu yapılan bağlantısı ve elde edilen eğriler aşağıda verilmiştir.



Şekil 4.13 : fxFODS ve fxPID fonksiyonları bağlantı şekli

4 saniyelik bir sistem gecikmesi için simülasyon yapılmıştır. Elde edilen sonuç Şekil 4.14’de gösterilmiştir.

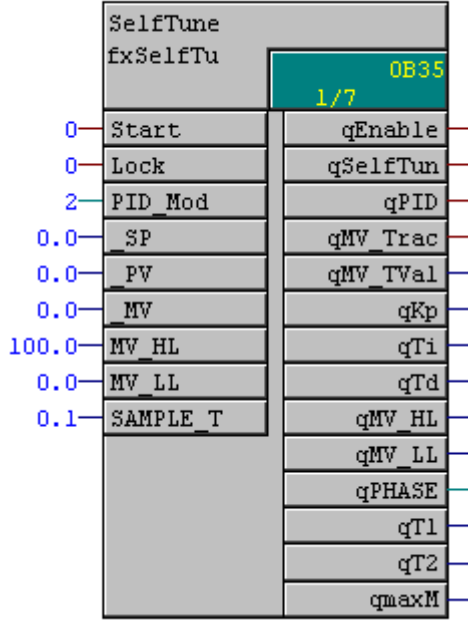


Şekil 4.14 : fxFODS-fxPID simülasyon sonucu

Şekil 4.14’te kırmızı çizgiler (SP) referansı, siyah çizgiler (QMV) kontrol çıkışı ve mavi çizgiler (PV) sistem geri besleme değerini göstermektedirler.

3.5 Adım 3 – Öz Ayarlamalı PID Fonksiyonu : fxSelfTune

Bu adımda Ziegler-Nicholas Reaksiyon Eğrisi Metodu’u kullanılarak Adım-2’de hazırlanan PID fonksiyonunu yönlendirecek ve kontrol edecek Öz Ayarlamalı PID fonksiyonu “fxSelfTune” hazırlanmıştır. Fonksiyonun CFC editöründeki görünüşü aşağıda gösterilmiştir.



Şekil 4.15 : fxSelfTune fonksiyonu CFC görünüşü

Fonksiyon giriş v çıkışları şunlardır;

Tablo 4.7 : fxPID giriş ve çıkışları

Adı	Tipi	Açıklama
Start	<i>BOOL</i>	PID fonksiyonu kontrolüne başla
Lock	<i>BOOL</i>	Çıkışları kitle
PID_Mod	<i>INT</i>	PID Modu
_SP	<i>REAL</i>	% Referans (fxPID fonksiyonunda)
_PV	<i>REAL</i>	% Sistem geri beslemesi (fxPID fonksiyonunda)
_MV	<i>REAL</i>	% Kontrolör çıkışı (fxPID fonksiyonunda)
MV_HL	<i>BOOL</i>	PID çıkışı üst sınırı
MV_LL	<i>BOOL</i>	PID çıkışı alt sınırı
Sample_T	<i>BOOL</i>	Örnekleme zamanı
QEnable	<i>BOOL</i>	fxPID fonksiyonunu devreye almak için kullanılacak çıkış
QSelfTune	<i>BOOL</i>	Fonksiyon Öz Ayarlamalı modda çalışıyor
QPID	<i>BOOL</i>	Fonksiyon PID modunda çalışıyor

Adı	Tipi	Açıklama
qMV_Track	<i>BOOL</i>	fxPID fonksiyonunu el konumuna almak için kullanılacak çıkış
qMV_TVvalue	<i>REAL</i>	fxPID fonksiyonu el konumunda iken kontrolör çıkışı
qK_p	<i>REAL</i>	Hesaplanan PID kazancı
qT_i	<i>REAL</i>	Hesaplanan integral katsayısı
qT_d	<i>REAL</i>	Hesaplanan türev katsayısı
qMV_HL	<i>REAL</i>	PID çıkışı üst sınırı (fxPID fonksiyonuna giriş)
qMV_LL	<i>REAL</i>	PID çıkışı alt sınırı (fxPID fonksiyonuna giriş)
qPhase	<i>INT</i>	Çalışılan faz
qT1	<i>REAL</i>	Hesaplanan sistem gecikmesi
qT2	<i>REAL</i>	Hesaplanan sistem yükselme zamanı
qmaxM	<i>REAL</i>	Hesaplanan en yüksek sistem ivmesi

PID_Mod:

fxSelfTune fonksiyonu 3 farklı modda çalışabilir. Bunlar sırası ile;

- P kontrolör modu (PID_Mod = 1)
- PI kontrolör modu (PID_Mod = 2)
- PID kontrolör modu (PID_Mod = 3)

Sistem parametrelerini hesaplandıktan sonra kullanıcının isteğine göre kontrolör parametreleri üretir. Örneğin PID modu olarak 2 seçilmiş ise, fonksiyon T_d türev parametresini her zaman 0'da tutar.

qPhase:

fxSelfTune fonksiyonunun sistemi kontrol ederken hangi aşamada olduğunu gösterir. qPhase değişkeni şu değerleri alabilir;

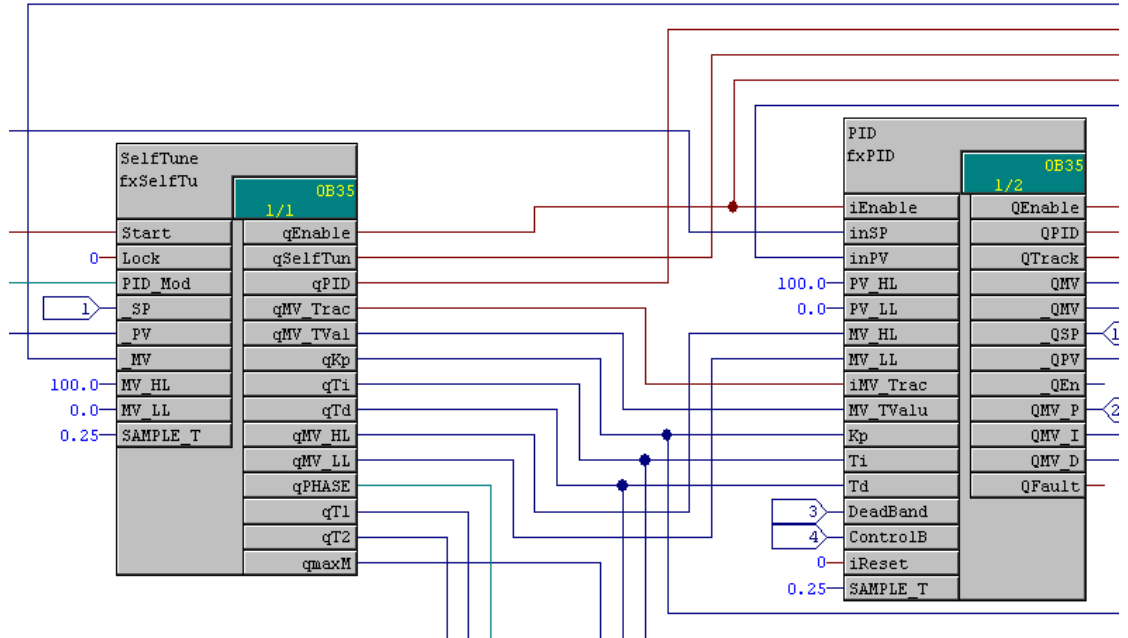
Tablo 4.8 : qPhase çıkışı

qPhase	Açıklama
0	Fonksiyon devre dışı, PID kontrolör çalışmıyor
1	Çıkıştaki PID parametreleri geçerli, PID devrede
10	Sistem analizine başlandı. Sistem çıkışı sabit bir değere ulaşmaya kadar bu aşamada kalınır. Sistem geri besleme değeri sabit hale gelince, 11. aşamaya geçilir.
11	Sisteme %40 kontrolör çıkışı uygulanır ve sistem izlenmeye başlanır. Sistemdeki gecikme hesaplanır, kaydedilir ve 12. aşamaya geçilir
12	Sistem çıkışı sabit bir değere ulaşmaya kadar bu aşamada kalınır. 20 saniyelik aralıklarla sistemden örneklemeler yapılır ve sistemin ne kadar hızlı hareket ettiği hesaplanır. Sistem çıkışı sabit hale gelince, daha

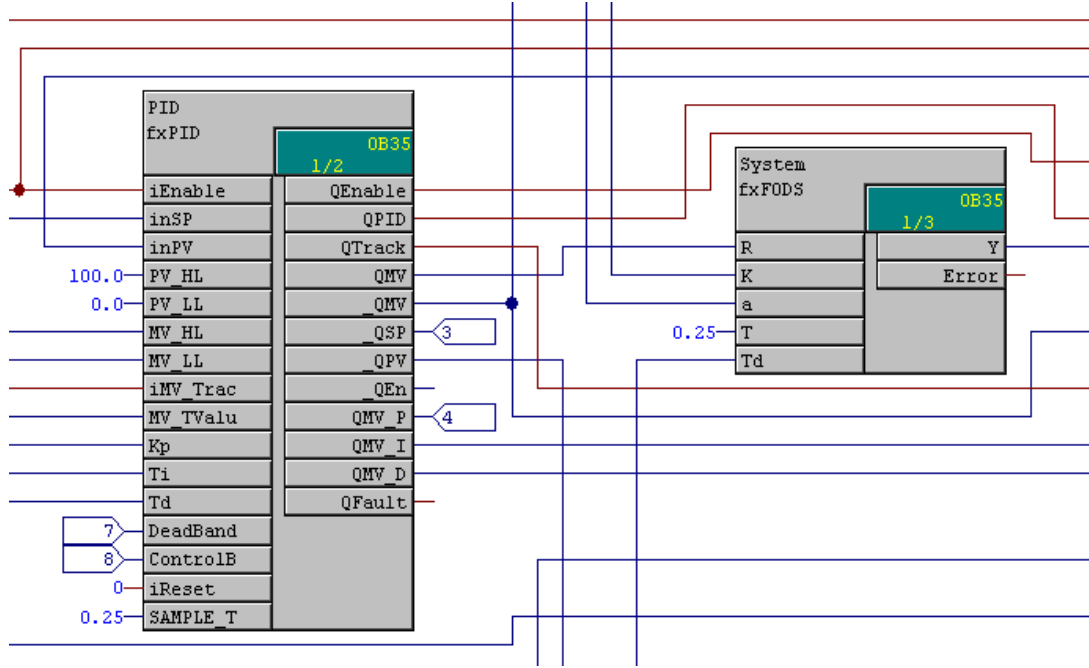
	önceden hesaplanan sistem ivmeleri kullanılarak sistem yükselme zamanı hesaplanır ve 13. aşamaya geçilir.
13	K_o , τ_0 ve ν_0 parametreleri hesaplanır, kaydedilir ve 14. aşamaya geçilir.
14	İstenilen moda uygun olarak PID parametreleri hesaplanır, çıkışlara işlenir ve 15. aşamaya geçilir.
15	fxPID fonksiyonu PID kontrolörü aktif hale getirilir. 20. aşamaya geçilir.
20	Uyarılama işleminin bittiği ve PID parametrelerinin hesaplandığı aşamadır. Bu aşamada fonksiyon yeni bir komut beklemeye başlar.

3.5.1 fxSelfTune Fonksiyonun Testi

fxSelfTune fonksiyonu daha önceki adımlarda yazılan fxPID ve sistem simülasyon blokları ile test edilmiştir. Kurulan düzeneğin CFC editöründeki görünüşü aşağıda verilmiştir.



Şekil 4.16a : fxSelfTune – fxPID fonksiyonlarının bağlantı şekli



Şekil 4.16b : fxPID – fxFODS fonksiyonlarının bağlantı şekli

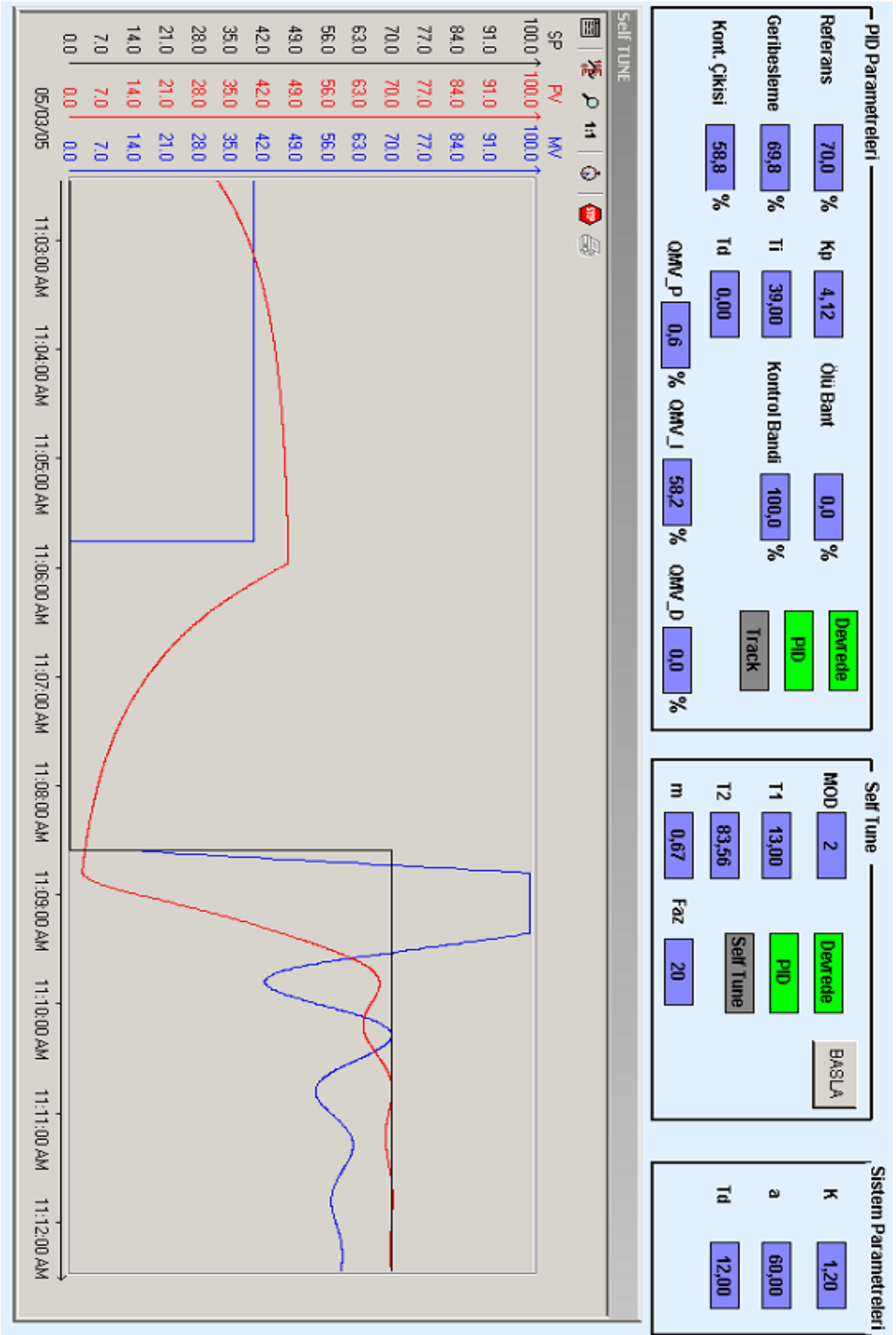
Görüldüğü üzere fxSelfTune çıkışları fxPID fonksiyonunun girişlerine bağlanarak fxPID fonksiyonu kontrol edilmiştir. Fonksiyonun işlev testi için ise sistem simülasyon blokları kullanılmıştır.

WinCC görüntüleme ve kontrol arabirimi kullanılarak düzenek tamamlanmış ve Öz Ayarlamalı PID fonksiyonu test edilmiştir. Hazırlanan test arabirimine tüm değişkenler taşınmıştır. (referans, sistem çıkışı, kontrolör çıkışı, hesaplanan parametreler, vs...)

Test sayfasında 3 bölüm görülmektedir.

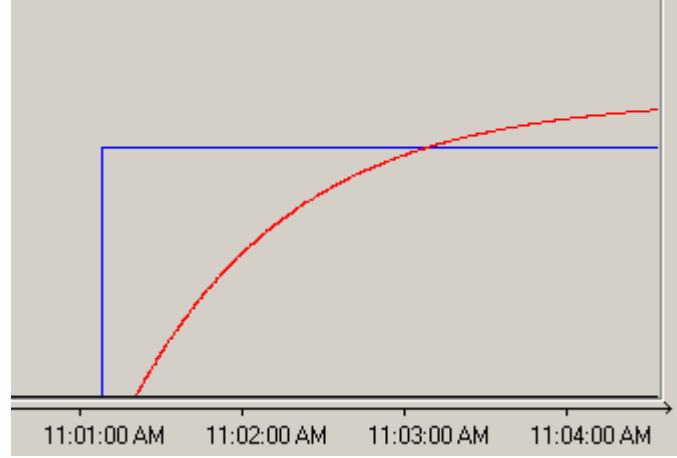
- PID parametreleri
- Öz Ayarlamalı Kontrolör parametreleri ve kontrolü
- Sistem simülasyon parametreleri

Böylelikle yazılan fonksiyonlar etkin şekilde test edilebilmiştir. Test arabirimi görüntüsü bir sonraki sayfada verilmiştir.



Şekil 4.17 : WinCC test arabirimi

Aşama - 10,11 ve 12 : (qPhase=10, qPhase=11, qPhase=12)

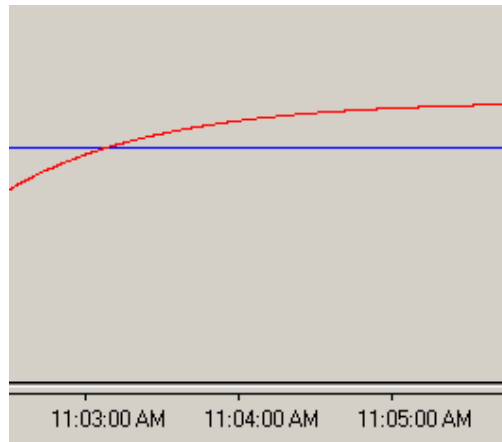


Şekil 4.18 : Test sonucu, 10. ve 11. aşama

Şekilde mavi çizgi ile gösterilen kontrolör çıkışı QMV' yi kırmızı çizgi ile gösterilen ise sistem geri besleme değeri inPV' yi göstermektedir.

Şekil 4.18'de Aşama-10,Aşama-11, Aşama-12 görülmektedir. fxSelfTune fonksiyonu sistem çıkışı sabit hale gelinceye kadar çıkışı gözlemiş, çıkışın sabit olduğunu görünce de sisteme %40' lık bir giriş uygulamıştır. Sistem gecikme zamanı bu aşamada hesaplanmış (T1) ve kaydedilmiştir. Sistem çıkışı yükselme eğiliminde iken fonksiyon 20 saniyelik aralıklarla sistem geri besleme değerini örneklemiş ve sistemin ne kadar hızlı hareket ettiğini hesaplamıştır (maxM). Hesaplanan en yüksek ivme değeri kullanılarak sistem yükselme zamanı hesaplanır.

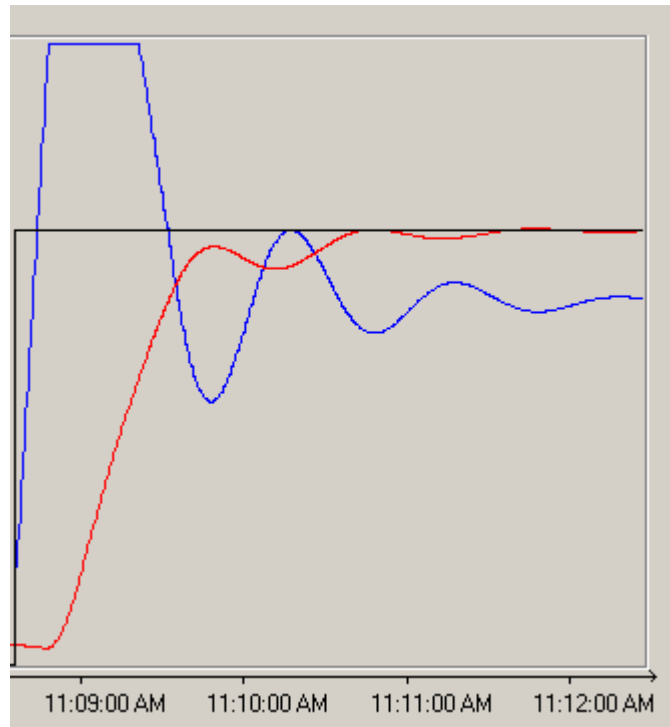
Aşama – 13 : (qPhase = 13)



Şekil 4.19 : Test sonucu, 13. aşama

Şekil 4.19'da Aşama-12' nin sonu ve Aşama-13 görülmektedir. Sistem çıkışı kararlı durumu ulaştınca, sistem çıkış değeri kaydedilmiş ve PID parametrelerinin hesabına başlanmıştır. PID parametreleri hesaplandıktan sonra PID fonksiyonu devreye alınmıştır. (Referans(SP)=0)

4. TEST SONUÇLARI



Şekil 5.1 : Hesaplanan parametreler ile sistem kontrolü

Hesaplanan PID parametreleri ile yapılan sistem kontrolü şekil 5.1'de verilmiştir. Simüle edilen sistemin parametreleri şöyledir;

- K** : 1.2 (Sistem kazancı)
a : 60.0 (Sistem yükselme zamanı)
T_d : 12.0 (Sistem gecikme zamanı)

Yukarıda parametreleri verilen sistem için Öz Ayarlamalı PID fonksiyonu fxSelfTune şu parametreleri hesaplamıştır;

- K_p** : 4.12 (PID kazancı)

T_i : 39.0 (İntegral katsayısı)

T_d : 0.0 (Türev katsayısı)

Şekil 5.2 ve şekil 5.3'de yapılan diğer testler görülmektedir. Yapılan testler için kullanılan ve hesaplanan parametreler şöyledir;

Şekil 5.2 için;

Sistem simülasyonu için kullanılan parametreler;

K : 1.6 (Sistem kazancı)

a : 90.0 (Sistem yükselme zamanı)

T_d : 12.0 (Sistem gecikme zamanı)

Öz Ayarlamalı PID Kontrolör fonksiyonunun hesapladığı PID parametreleri;

K_p : 4.38 (PID kazancı)

T_i : 39.0 (İntegral katsayısı)

T_d : 0.0 (Türev katsayısı)

Şekil 5.3 için;

Sistem simülasyonu için kullanılan parametreler;

K : 1.3 (Sistem kazancı)

a : 50.0 (Sistem yükselme zamanı)

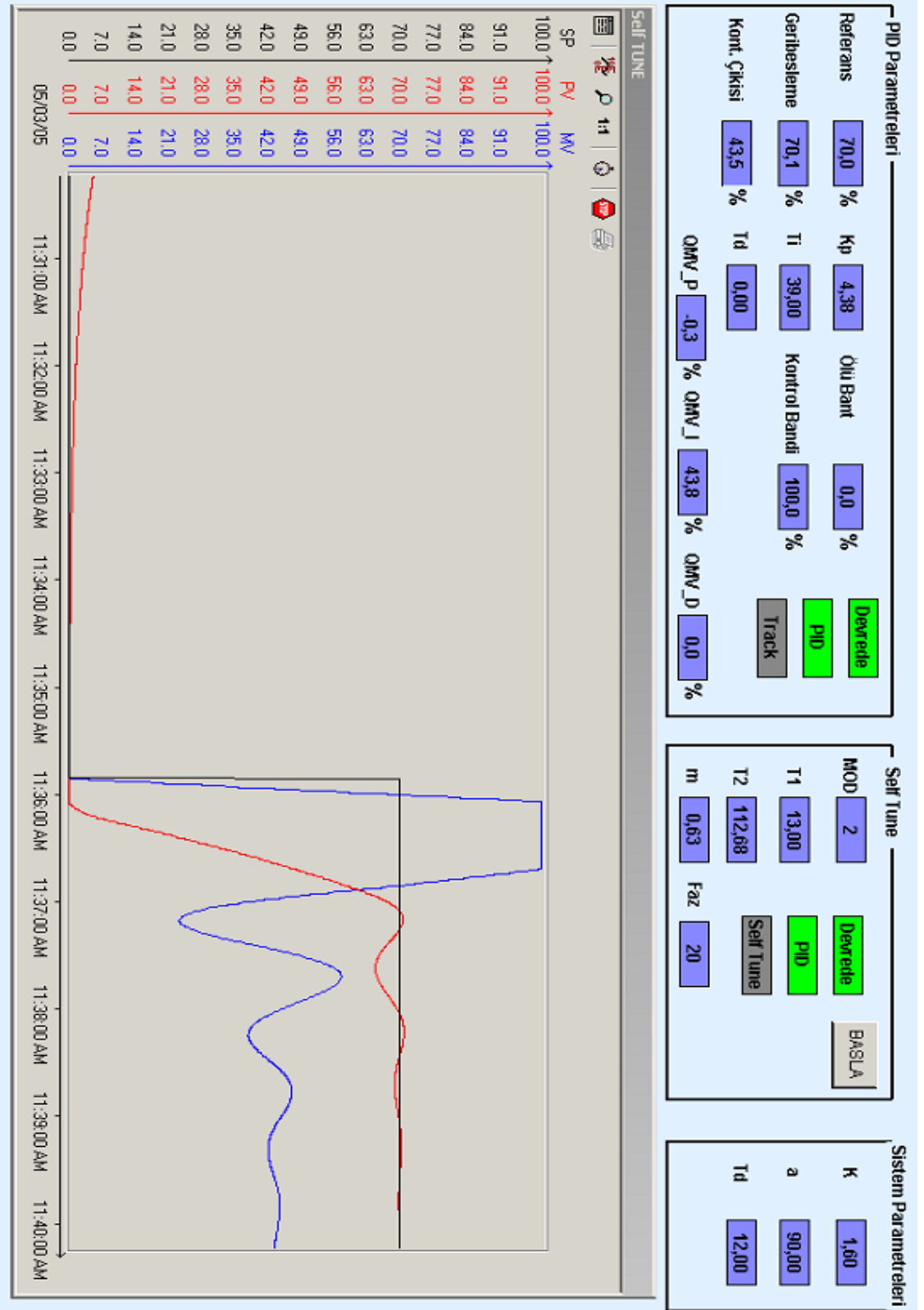
T_d : 0.0 (Sistem gecikme zamanı)

Öz Ayarlamalı PID Kontrolör fonksiyonunun hesapladığı PID parametreleri;

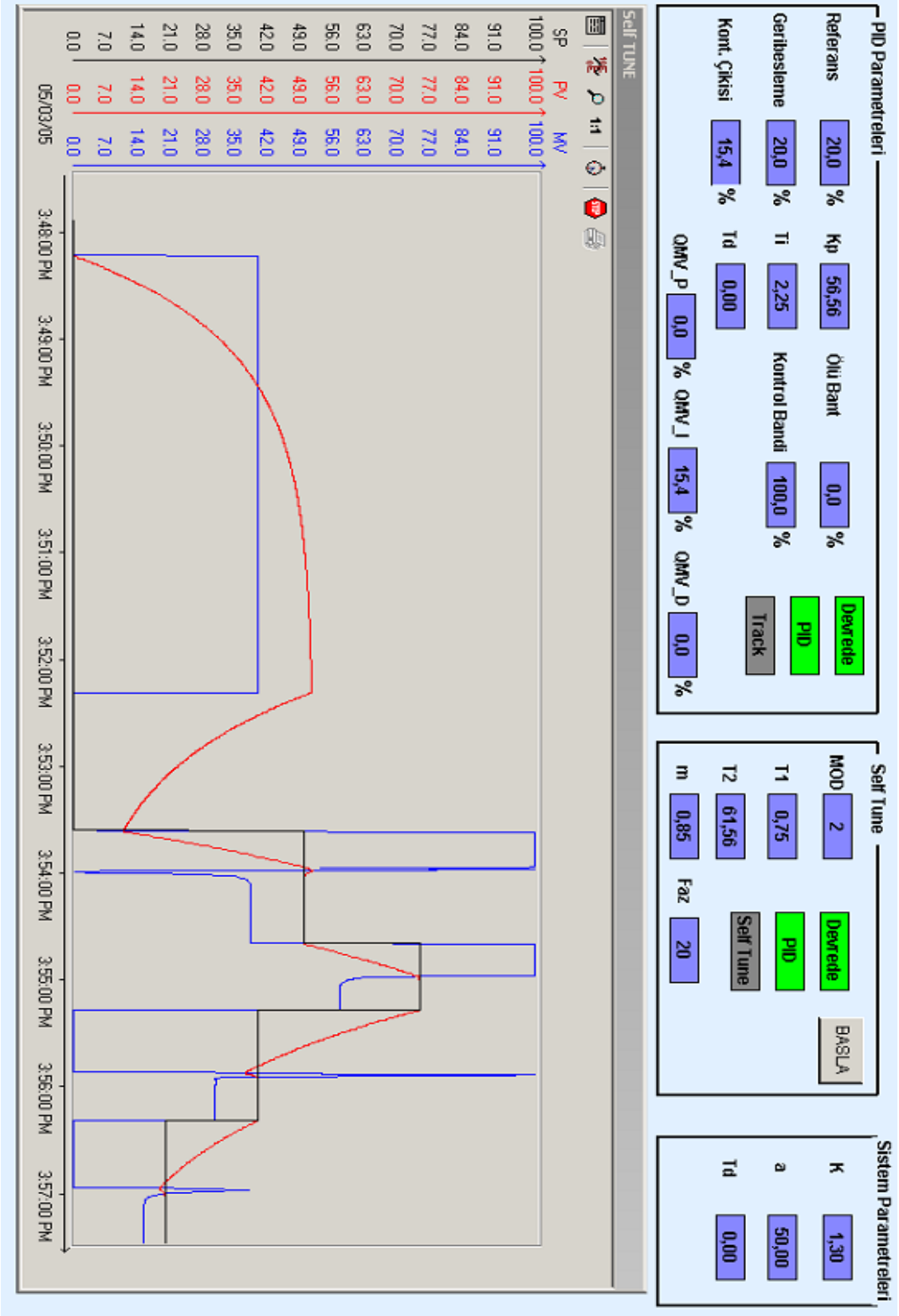
K_p : 56.56 (PID kazancı)

T_i : 2.25 (İntegral katsayısı)

T_d : 0.0 (Türev katsayısı)



Şekil 5.2 : Test sonucu



Şekil 5.3 : Test sonucu

KAYNAKLAR

- [1] **Kuo, Benjamin C.**, 2002. Otomatik Kontrol Sistemleri, Literatür Yayıncılık, İstanbul (Çeviri: Prof. Dr. Atilla Bir)
- [2] **Levine, William S.**, 2000. Control System Fundamentals, CRC Pres,Florida
- [3] **Leigh, James R.**, 1992. Applied Digital Control, Prentice Hall International, Hertfordshire
- [4] **Aström, Karl J. and Hagglund, Tore**, 1995. PID Controllers, 2nd Edition,Instrument Society of America, New York
- [5] **Kurtulan, Salman**, 2001. PLC ile Endüstriyel Otomasyon, Birsen Yayınevi, İstanbul

ÖZGEÇMİŞ

1979, Bingöl, Merkez doğumluyum. İlk ve orta öğrenimimi Bingöl, Merkez’de tamamladım. Fen Lisesi Sınavları’nda Malatya Fen Lisesi’ni okumaya hak kazandım. 1997 yılında Malatya Fen Lisesi’nden mezun oldum. Aynı yıl üniversite sınavlarında İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar Mühendisliği’ni kazandım. 2002 yılında İTÜ Kontrol ve Bilgisayar Mühendisliğinden 3.1 ortalama ile mezun oldum.

Üniversite mezuniyetimden bu yana Endüstriyel Otomasyon alanında faaliyet gösteren çeşitli şirketlerde Proje Mühendisi ve Proje Yöneticisi olarak görev yaptım. Bu süre içerisinde irili ufaklı (sistem otomasyonlarından, komple anahtar teslimi fabrika otomasyonlarına kadar) çeşitli projelerde görev yaptım. Halen Endüstriyel Otomasyon alanında faaliyet gösteren bir firmada Proje Yöneticisi olarak görev yapmaktayım.